RĪGAS TEHNISKĀ UNIVERSITĀTE

PROMOCIJAS DARBS

2013

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

Lietišķo datorsistēmu institūts

ILZE BIRZNIECE

INTERAKTĪVAS UZ INDUKTĪVO APMĀCĪBU BALSTĪTAS KLASIFIKĀCIJAS SISTĒMAS MODEĻA IZSTRĀDE

Promocijas darbs

Zinātniskā vadītāja

Dr.sc.ing., profesore

Mārīte Kirikova

Rīga 2013

Šis darbs izstrādāts ar Eiropas Sociālā fonda atbalstu projektā „Atbalsts RTU doktora studiju īstenošanai”.

ANOTĀCIJA

Palielinoties informācijas apjomam pasaulē, daudzās sfērās pieaug nepieciešamība pēc datorizētas dažādu objektu klasifikācijas. Tāpēc arvien aktuālāka ir automātisku datu apstrādes tehniku izmantošana, kurās ir iesaistīta mašīnapmācība. Klasifikācija ir viens no mašīnapmācības uzdevumiem, kura mērķis ir noteikt objekta piederību noteiktai klasei, balstoties uz klasifikācijas algoritmam sniegtiem faktiem par zināma skaita objektu atbilstību šīm klasēm. Par *automātisku klasifikāciju* promocijas darbā tiek saukts datorizēts klasifikācijas process, kurā no klasifikatora apmācības brīža ar sagatavotiem apmācības datiem līdz lēmuma pieņemšanai par jaunu objektu klasifikāciju netiek iesaistīts sistēmas lietotājs vai eksperts. Risināmajiem uzdevumiem un apstrādājamajiem datiem kļūstot arvien sarežģītākiem, pilnībā automātiskas klasifikācijas pieejas ne vienmēr sniedz vēlamo rezultātu. Tādēļ promocijas darbs ir veltīts *automatizēta* jeb *daļēji automātiska* klasifikācijas risinājuma izveidei, kas izmanto gan mašīnapmācības sniegtās iespējas, gan interaktīvu sadarbību ar jomas ekspertu savu rezultātu uzlabošanai klasifikatora lietošanas laikā, ja klasifikators sastopas ar objektu, ko tas nespēj klasificēt vai nav pārliecināts par sava lēmuma pareizību.

Viena no mašīnapmācībā izmantotajām klasifikācijas metožu grupām ir induktīvā apmācība. Induktīvajā apmācībā iegūtie rezultāti ir saprotami ne tikai datorsistēmai, bet arī tās lietotājam. Šī ir būtiska induktīvās apmācības priekšrocība, salīdzinājumā ar citām klasifikācijas metodēm. Lai pilnvērtīgi iesaistītu ekspertu, klasifikācijas sistēma balstās uz induktīvās apmācības izmantošanu cilvēkam saprotamu klasifikācijas likumu iegūšanai. Promocijas darbā ir izstrādāts interaktīvas uz induktīvo apmācību balstītas klasifikācijas sistēmas (*InClaS*) modelis, kas apvieno algoritmus, arhitektūras un vadlīnijas, kuras ļauj izstrādāt interaktīvu klasifikācijas sistēmu. Šādas sistēmas izstrādes mērķis ir samazināt nepareizi klasificēto objektu skaitu, salīdzinot ar „tradicionālu" automātisku klasifikācijas sistēmu. Modelis ir speciāli izstrādāts lietošanai jomās, kur objekts var piederēt vienlaicīgi vairākām klasēm (daudzkategoriju klasifikācija). *InClaS* modelis ir aprobēts divās problēmsfērās – izglītībā un medicīnā –, kas pierāda, ka nepareizi klasificēto objektu skaitu ir iespējams samazināt, ja klasifikatoram neskaidrie (neklasificētie un nepārliecinoši klasificētie) objekti tiek atlasīti un nodoti ekspertam izvērtēšanai.

Promocijas darba rezultāti ir atspoguļoti 13 zinātniskajos rakstos un par tiem ir ziņots 12 dažādās starptautiskās konferencēs. Darbs sastāv no ievada, 6 nodaļām, galvenajiem rezultātiem un secinājumiem. Tajā ir 160 lappuses, 47 attēli un 34 tabulas pamattekstā, 139 nosaukumu literatūras sarakstā un 11 pielikumi.

ANNOTATION

Growing amount of information in the world has increased the need for computerized classification of different objects. Therefore, more important have become automatic data processing techniques which make use of machine learning. Classification is one of the machine learning tasks where the program learns to classify new instances from the provided facts. In the thesis the term *automatic classification* is used to denote a computerized classification process which excludes the user or expert involvement starting from the classifier's training with provided data set till applying it for new instance classification. Application domains and data are getting more complex leading to inability for automatic classification approaches to always reach the desired result. Thereof, the thesis is devoted to the development of *automated* or *semi-automatic* classification solution which incorporates both machine learning facilities and interactive involvement of a domain expert in the classifier's applying stage for improving its results if the classifier makes uncertain classification.

One of the classification method groups used in machine learning is inductive learning. Results obtained from the inductive learning methods are interpretable not only for machines, but also for their users. This is a fundamental advantage over other classification methods. To fully utilize interactivity with an expert, the classification system is based on inductive learning for extracting human-readable classification rules. The doctoral thesis provides interactive inductive learning based classification system's (*InClaS*) model which gathers algorithms, architectures and guidelines for developing an interactive classification system. The aim of developing this system is to decrease the number of misclassified instances regarding to the "traditional" automatic classification system. This model is particularly intended to be applied in domains with a multi-label class membership. The *InClaS* model has been approbated in two problem domains – education and medicine – which demonstrate the ability to reduce the number of misclassified instances supposing that instances with uncertain classification (unclassified and classified with low confidence) are detected and transferred to the expert for assessment.

The results of the thesis are published in 13 international scientific publications and presented in 12 international conferences.

The doctoral thesis includes introduction, 6 sections, main results and conclusions section. It consists of 160 pages, 47 figures and 34 tables in the main text, 11 appendices. The bibliography contains 139 references.

PATEICĪBA

Vēlos pateikties visiem, kuri ir palīdzējuši promocijas darba tapšanā – darba vadītājai profesorei Mārītei Kirikovai par gatavību vienmēr veltīt savu laiku un uzmanību, studentiem, kuriem esmu vadījusi bakalaura darbus, par ieguldījumu tēmas teorētiskā un praktiskā risinājuma papildināšanā, Sistēmu teorijas un projektēšanas katedras kolēģiem par radošām sarunām, Datorzinātnes un informācijas tehnoloģijas fakultātes mācībspēkiem par sniegto izglītību un ikvienam, kurš apzināti vai neapzināti virzījis manu un šī darba attīstību.

Vislielākais paldies manai mammai un vīram, bez kuru atbalsta promocijas darba izstrāde būtu bijusi neizmērojami grūtāka.

Saturs

IEVADS

Pieaugošais informācijas daudzums pasaulē rada nepieciešamību pēc datu apstrādes tehnikām, kas spētu samazināt cilvēka veiktās rutīnas aktivitātes. Šādas iespējas piedāvā mākslīgā intelekta nozare mašīnapmācība (ang. v. - *machine learning*). Mašīnapmācība sniedz datorprogrammai spēju apmācīties, balstoties uz pagātnes pieredzi, un uzlabot savu sniegumu [1]. Klasifikācija ir viens no mašīnapmācības uzdevumiem, kur klasifikators apgūst noteikt objekta klases piederību, balstoties uz iepriekš iegūtiem faktiem konkrētā problēmsfērā (ang. v. - *domain*). Ar jēdzienu „problēmsfēra" var apzīmēt jebkuru sistēmu vai darbības jomu. Būtiskākie darbā lietotie jēdzieni un termini ir skaidroti darba 1. pielikumā atrodamajā terminu sarakstā.

Nepieciešamība pēc dažādu objektu klasificēšanas ir sastopama daudzās sfērās, piemēram, medicīnas diagnostikā, kredītņēmēju vērtēšanā, attēlu apstrādē, mārketingā, dokumentu organizēšanā utt. [2]. Mašīnapmācībā tiek plaši izmantotas daudzas klasifikācijas pieejas. Tajā pat laikā palielinās arī problēmu loks, kam būtu lietderīgi izmantot mašīnapmācību [3], tomēr ir problēmsfēras, kur prakse pierāda, ka pilnībā automātiski risinājumi nav piemēroti. Par *automātisku klasifikāciju* promocijas darbā tiek saukts datorizēts klasifikācijas process, kurā no klasifikatora apmācības brīža (neskaitot apmācības datu sagatavošanu un klasifikācijas algoritma parametru iestatīšanu) līdz lēmuma pieņemšanai par jaunu objektu (jeb no apmācības puses skatoties - piemēru) klasifikāciju netiek iesaistīts sistēmas lietotājs vai eksperts. Nepiemērotība automātiskam klasifikācijas risinājumam var rasties vairāku iemeslu dēļ [4, 5]:

- problēmsfēra ir grūti formāli atspoguļojama, jo satur ne tikai skaitlisku vai nominālu, bet arī konceptuāli sarežģītu informāciju;
- ir maz datu, uz ko balstīt apmācību;
- eksperti, it sevišķi sarežģītās problēmsfērās, neuzticas automātiskiem risinājumiem.

Visos šajos gadījumos var palīdzēt *automatizēti* jeb *daļēji automātiski* risinājumi, iesaistot datorizētā klasifikācijas procesā sistēmas lietotāju – padarot klasifikācijas procesu interaktīvu. Tā, piemēram, pastāv viedoklis [6], ka klasifikācijai jābūt interaktīvai gan tādēļ, lai izmantotu eksperta zināšanas, gan, lai saņemtu eksperta apstiprinājumu sistēmas iegūtajam rezultātam. Klasifikācija nekad nav bijusi izolēta no cilvēka, jo eksperts ir tas, kurš definē klases un izvēlas raksturīgās pazīmes, ko izmantot klasifikācijā. Tātad cilvēka iesaistīšana dažās klasifikācijas procesa fāzēs nav jāuzskata par problēmu vai pieejas trūkumu [6], bet dabisku sastāvdaļu. Eksperta zināšanas par problēmsfēru var un vajag izmantot, lai uzlabotu klasifikācijas rezultātus un palīdzētu sistēmai strādāt labāk. Tā ir iespēja cilvēkam un tehnoloģijām mijiedarboties, un

palīdzot datorsistēmai, eksperts nodrošina to, ka sistēma varēs dot vairāk labuma un pilnvērtīgāk palīdzēt viņam pašam. Ja kādā problēmsfērā automātiska klasifikācija sniedz neapmierinošus klasifikācijas rezultātus, savukārt ir pieejams eksperts, kurš var veikt klasifikāciju šīs sfēras ietvaros, tad daļēji automātiskas jeb interaktīvas klasifikācijas izmantošana ļauj iegūt kompromisu starp eksperta ieguldītā darba apjomu un klasifikācijas rezultātu kvalitāti.

Promocijas darbs ir veltīts *automatizēta* jeb *daļēji automātiska* klasifikācijas risinājuma izveidei, kas izmanto gan mašīnapmācības sniegtās iespējas, gan interaktīvu sadarbību ar jomas ekspertu klasifikatora lietošanas laikā, ja klasifikators sastopas ar objektu, kura klasifikācija tam ir neskaidra. *Klasifikatoram neskaidrs objekts* jeb *neskaidra klasifikācija* (ang. v. – *uncertain classification*) šī darba kontekstā ietver gan *neklasificētus* (ang. v. – *unclassified*), gan *nepārliecinoši klasificētus* (ang. v. – *low confidence of classification*) objektus. Šie termini sīkāk ir apskatīti darba 3. un 4. nodaļā.

Tēmas aktualitāte

Galvenā mašīnapmācības būtība ir *mācīšanās*. Spēja mācīties ir sistēmas intelekta pazīme. Mācīšanās ir iespējama, ja pētāmajā vidē, problēmā vai datos ir regularitātes un sakarības. Ja sistēma spēj šīs sakarības atklāt, tad tā var izdarīt secinājumus un nākotnē darboties efektīvāk vai nepieļaut atkārtotas kļūdas. Viena no klasifikācijas problēmām ir nespēja iegūt apmierinošus automātiskas klasifikācijas rezultātus visās jomās, kurās mašīnapmācības sniegtās iespējas varētu būt noderīgas. Tā kā mācīšanās notiek no eksperta vai ārējās vides piegādātiem datiem, tad tiem ir ļoti liela nozīme iegūstamā apmācības rezultāta kvalitātē. Ir naivi gaidīt, ka no neprecīziem, nepilnīgiem vai nepietiekama apjoma datiem būs iespējams iegūt akurātu klasifikatoru, kurš veiksmīgi spēs veikt klasifikāciju iepriekš apmācības laikā neredzētiem datiem. Datu kvalitāti nosaka gan procesi to iegūšanā un sagatavošanā, gan sfēras īpatnības, ko tie raksturo. Ne visās dzīves sfērās iespējams iegūt vienlīdz piemērotus datus klasifikācijas uzdevumiem. Klasifikācijas algoritmi izmanto skaitliskus vai nominālus datus, kuri ir strukturēti. *Strukturēti dati* ir sadalīti nelielās, diskrētās vienībās. Katra datu vienība attiecas uz vienu konceptu, piemēram, objekta krāsu. Strukturēti dati parasti tiek glabāti tabulās vai citās viegli izgūstamās formālās struktūrās. Tā, piemēram, signālu apstrādē jāsastopas ar skaitliskiem datiem, bet lietotāju profilēšana e-komercijā saistās vairāk ar nomināliem jeb kategoriju veida datiem. Informācija var glabāties arī *nestrukturētu datu veidā*, kad dati, piemēram, teksts, ir bez jebkādām iezīmēm. No šādiem datiem, piemēram, dokumentiem, kas jāorganizē atbilstoši to tēmām un sākotnēji ir atspoguļoti kā nestrukturēts teksts, lai tos izmantotu mašīnapmācībā, tiek

iegūti strukturēti dati, visbiežāk izveidojot vārdu vektorus, kas raksturo dažādu vārdu biežumu tekstā.

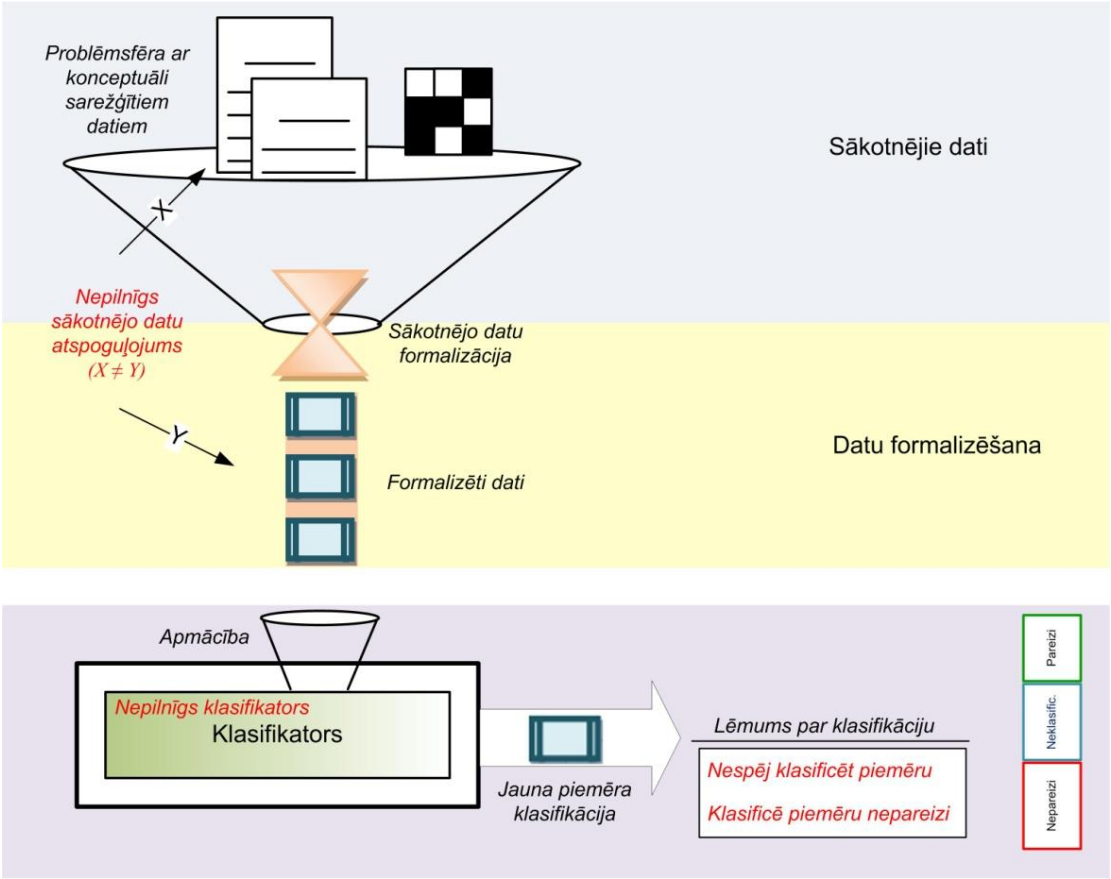
Daudzās jomās praktiski pieejamie dati ir *daļēji strukturēti*, un arī tos ir sarežģīti pilnvērtīgi organizēt noteiktās struktūrās. Tas apgrūtina iespēju izmantot tradicionālās mašīnapmācības metodes automātiski un izslēdz no analīzes lielu daudzumu pieejamo datu. Piemēram, dati no sociālajiem medijiem ir arvien nozīmīgāks informācijas avots, bet tie ir daudz grūtāk apstrādājami un analizējami neviennozīmīgās strukturētības, dabīgās valodas lietojuma un lielā datu apjoma dēļ. Ir vairākas iespējas, kā rīkoties ar daļēji strukturētiem datiem [7]: (1) ignorēt tos, (2) pārvērst strukturētos datos, neskatoties uz ierobežojumiem un nepilnībām, vai arī (3) atrast citu glabāšanas un izmantošanas mehānismu.

Lai lietotu mašīnapmācību plašākā cilvēku darbības sfēru lokā nekā šobrīd tiek lietotas automātiskās klasifikācijas metodes, jāņem vērā šādi apstākļi:

1. zināšanas par problēmsfēru tiek atspoguļotas dažādās formās un strukturētības pakāpēs;
2. esošajiem apmācības algoritmiem klasifikācijas veikšanai nepieciešami formāli atspoguļoti (strukturēti) dati.

Pārveidojot pieejamos datus no sākotnēji daļēji strukturēta vai nestrukturēta formāta uz klasifikācijas algoritmiem lietojamu – strukturētu formātu, var pazaudēt daļu būtiskas informācijas vai atspoguļot to neprecīzi, tādējādi iegūstot apmācības kopu, kas nepilnīgi raksturo pētāmo problēmsfēru. Ja šos pārveidotos datus tālāk izmanto mašīnapmācībai un klasifikatora iegūšanai, pastāv iespēja, ka klasifikators būs nepilnīgs un nespēs noteikt klasi jauniem piemēriem vai noteiks to nepareizi. Šāda situācija ir atspoguļota 1. attēlā. Paskaidrojot atšķirību starp dabā lietotajiem terminiem „klasifikators" un „klasifikācijas sistēma", jāmin, ka klasifikators ir daļa no klasifikācijas sistēmas. Klasifikators ir likumu kopa vai cita veida klasifikācijas modelis, kas kalpo jaunu piemēru klases (klašu) noteikšanai konkrētā problēmsfērā. Savukārt klasifikācijas sistēma ir programmatūra, kas ietver (apvieno) klasifikatoru, lietotāja saskarni un citas saistītās komponentes, piemēram, pirms un pēcapstrādi. Principā klasifikators var būt arī indukcijas rezultātā iegūts likumu saraksts uz papīra, pēc kura vadīties jauna piemēra klasifikācijas gaitā. Klasifikācijas sistēma ir klasifikators un tā perifērija, kas kopumā nodrošina datorizētu klasifikācijas procesu.

1. attēls demonstrē, ka iespējamie klasifikācijas rezultāti vispārīgā gadījumā ietver (1) piemērus, ko klasifikators ir klasificējis pareizi, (2) piemērus, kam tas nespēj noteikt klasi, balstoties uz savu klasifikācijas modeli, iegūstot neklasificētus piemērus, un (3) piemērus, kam klase ir noteikta nepareizi.



1. att. Automātiska klasifikācija nepilnīgu apmācības datu gadījumā

Promocijas darbā īpaša uzmanība ir pievērsta vienai no cilvēku darbības sfērām, kam raksturīgi konceptuāli sarežģīti dati, proti, studiju priekšmetu salīdzināšanai augstākās izglītības jomā. Salīdzināt studiju programmas un atsevišķus priekšmetus ir laikietilpīgs darbs, ja to veic tikai manuāli. Studiju programmu un priekšmetu apraksti, kas tiek izmantoti priekšmetu atbilstības noteikšanai, parasti ir daļēji strukturētu tekstu veidā, kas var saturēt dažādi nosauktas un konceptuāli atšķirīgas sadaļas un atrodas, piemēram, mācību iestāžu mājas lapās internetā. Formalizēt un strukturēt šo informāciju traucē neskaidrās objektu attiecības un dabīgā valodā veidotie apraksti, kā arī mašīnapmācībai pieejamais relatīvi nelielais piemēru skaits (eksperta veiktie salīdzinājumi, kas sistēmai kalpo par pieredzi, no kuras izdarīt secinājumus). Par šo problēmsfēru, tajā pastāvošajiem datorizētajiem risinājumiem un prasībām pret nepieciešamo mašīnapmācības risinājumu studiju priekšmetu salīdzināšanas atbalstam sīkāk tiks stāstīts darba 1.2. apakšnodaļā.

Problēmas nostādne

Ikdienas darbu atvieglo dažādi datorizēti risinājumi, tajā skaitā, dažādu objektu klasifikācija iepriekš noteiktās kategorijās jeb klasēs. Tomēr automātiski risinājumi ne vienmēr

sniedz vēlamo rezultātu un pareizi nosaka objektu klases piederību jauniem objektiem, jo iegūtais klasifikators dažādu iemeslu dēļ var nebūt pilnīgs un precīzs (skat. 1. att.). Tā vietā, lai šādos gadījumos atteiktos no mašīnapmācības sniegtajām iespējām vispār, var atkāpties no uzstādījuma iegūt pilnīgi automātisku risinājumu, bet izveidot automatizētu jeb daļēji automātisku sistēmu, ļaujot klasifikatoram izpildīt daļu darba un saglabājot arī eksperta iesaisti. Tādējādi eksperts, no vienas puses, palīdzētu pilnveidot klasifikatoru, papildinot to ar savām zināšanām par konkrēto sfēru, no otras puses, varētu caurskatīt klasifikācijas rezultātus un vairotu savu pārliecību par sistēmas darba rezultātu uzticamību. Lai strādātu ar klasifikācijas sistēmu un caurskatītu iegūtos rezultātus, sistēmas lietotājam nav jābūt problēmsfēras ekspertam, taču, lai sniegtu savas zināšanas un pilnveidotu klasifikatoru, ir nepieciešama padziļināta izpratne par attiecīgo jomu. Tādēļ cilvēks vienmēr var būt klasifikācijas sistēmas lietotājs, bet ne vienmēr ir arī eksperts attiecīgajā jomā.

Būtisks parametrs, kas nosaka, vai klasifikācijas rezultāti kādā sfērā ir uzticami un praktiskai lietošanai pieņemami, ir nepareizi klasificēto objektu skaits, tādēļ tas arī izvēlēts kā sasniedzamā rezultāta novērtējuma mērs. Nepareizi klasificēto objektu skaita samazināšana ir vēlama jebkurā klasifikācijas gadījumā, tomēr vēl jo būtiskāka tā ir kritisku lēmumu gadījumā, piemēram, medicīnas sfērā, un situācijās, kad sākotnēji iegūstamie automātiskas klasifikācijas rezultāti ir neapmierinoši, līdz ar to – nebūtu izmantojami reālai ieviešanai.

Darba mērķis un uzdevumi

Darba mērķis ir izstrādāt automatizētas klasifikācijas sistēmas modeli, kas pieļauj interaktivitāti ar ekspertu klasifikatora lietošanas laikā, ja klasifikators sastopas ar objektu, ko tas nespēj klasificēt vai nav pārliecināts par sava lēmuma pareizību.

Izvirzītā darba mērķa sasniegšanai ir jāīsteno šādi **darba uzdevumi:**

- Veikt izglītības dokumentu datorizētas salīdzināšanas risinājumu analīzi un identificēt risināmos uzdevumus.
- Veikt klasifikācijas uzdevuma mašīnapmācībā izpēti.
- Veikt esošo interaktīvo klasifikācijas risinājumu izpēti.
- Veikt klasifikācijas sistēmu arhitektūru analīzi interaktīvas klasifikācijas sistēmas izstrādei.
- Izstrādāt interaktīvas klasifikācijas sistēmas modeli, kas apvieno interaktīvas klasifikācijas sistēmas radīšanai nepieciešamās komponentes (algoritmus, metodes, pieejas un arhitektūras).

- Izstrādāt interaktīvas klasifikācijas sistēmas modeļa papildinājumu, kas apvieno interaktīvas daudzkategoriju klasifikācijas sistēmas radīšanai nepieciešamās komponentes.
- Realizēt interaktīvas klasifikācijas sistēmas prototipu, kas ievieš izstrādāto modeli.
- Pārbaudīt izstrādātā modeļa lietderību un prototipa lietojamību, veicot praktiskus eksperimentus.

Pieņēmumi un ierobežojumi

Izstrādājamā interaktīvā klasifikācijas sistēma ir paredzēta situācijām, kurās pastāv šādi *nosacījumi*:

- klasifikatora apmācībai izmantojamie dati ir ekspertam saprotami:
  - pēc savas būtības un struktūras;
  - pēc to apjoma (objekta apraksta apjoms nav *pārāk* liels).
- cilvēks-eksperts ir pieejams.

Tātad datiem, ar kuriem strādās klasifikācijas sistēma, ir jābūt tādiem, kurus eksperts spēj interpretēt. Tas nozīmē, ka ekspertam ir jāsaprot sākotnējo vai apstrādājamo datu jēga, lai viņš varētu klasificēt jaunu objektu, ja sistēma to lūgs. Ja datiem ir veikta būtiska priekšapstrāde, kas nepieciešama vienotas struktūras ieviešanai, bet kas padara objektu aprakstu lietotājam neinterpretējamu, tad jāsaglabā sasaiste ar sākotnējo datu avotu. Šāds gadījums, piemēram, ir teksta dokumentu pārvēršana vārdu vektoros – eksperts var noteikt dokumenta tematu, redzot tā pilnu tekstu, bet daudz grūtāk tas ir izdarāms pēc datu priekšapstrādes veikšanas, kurā tiek iegūts dažādu vārdu parādīšanās biežums sākotnējā tekstā. Eksperts labāk saprot pilno tekstu, bet klasifikators – vārdu vektorus, tāpēc ir vēlams saglabāt saiti starp oriģinālo un pārveidoto datu formātu. Savukārt, sensoru dati ciparu formātā cilvēkam var būt pilnībā nesaprotami, līdz ar to viņš nevarēs palīdzēt sistēmai noteikt klasi jauniem objektiem šajā situācijā. Cits apstāklis ir viena objekta aprakstīšanai izmantoto datu apjoms. Ja tas ir pārāk liels, tad eksperts nespēs sniegtos datus operatīvi saprast un, attiecīgi, arī klasificēt. Konkrēts slieksnis, līdz kuram aprakstošo datu apjoms ir pieņemams, būs atkarīgs no problēmas specifikas, tās sarežģītības, kā arī tās strukturēšanas un uzskatāmas attēlošanas iespējām. Visbeidzot, ekspertam, kas risināmajā jomā ir spējīgs sniegt sistēmai padomu, vispār ir jābūt pieejamam. Ja šāda eksperta nav, vai jau iepriekš zināms, ka eksperta nodarbināšana ir pārāk dārga, tad interaktīvam risinājumam nav jēgas. Tātad tiek pieņemts, ka interaktīva klasifikācijas sistēma tiek lietota sfērā, kur augstāk minētos nosacījumus ir iespējams ievērot.

No klasifikācijas pieejām darbā ir izvēlēta induktīvā apmācība, jo tās sniegtie rezultāti (iegūtais klasifikators jeb klasifikācijas modelis) ir caurskatāmi. Induktīvās apmācības algoritmi savus rezultātus atspoguļo lēmumu koka vai likumu saraksta veidā, kas ir lasāmi un saprotami ne tikai datorsistēmai, bet arī lietotājam. Sistēmas interaktivitāte un klasifikatora caurskatāmība palīdz lietotājam un ekspertam gan uzlabot klasifikatoru, gan veicina uzticēšanos iegūtajam rezultātam, jo klasifikācijas process ir kontrolējams un sistēma var pamatot savus spriedumus.

Promocijas darbā tiek apskatītas problēmsfēras, kurām ir raksturīga objektu daudzkategoriju piederība, tas ir, objekts dabiski var vienlaicīgi piederēt vairākām klasēm. Šāda situācija ir, piemēram, ziņu organizēšanā, jo viens raksts var atbilst vairākām tēmām, piemēram, sports un tūrisms. Šis nosacījums ir radies tādēļ, ka daudzkategoriju klasifikācija ir raksturīga primārajā izskatāmajā problēmsfērā - studiju priekšmetu salīdzināšanā.

Pētījuma objekts, priekšmets un izmantotās metodes

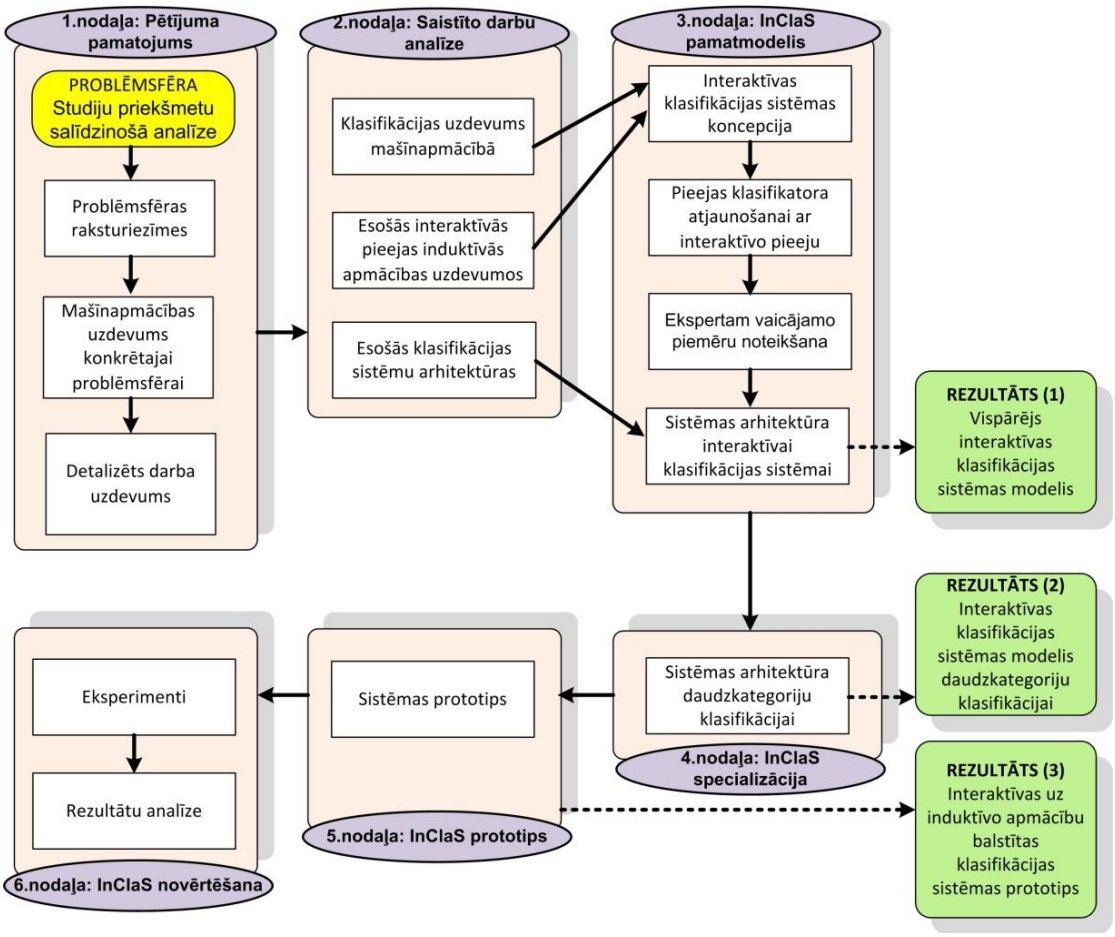
Pētījumu objekts ir klasifikācijas uzdevums mašīnapmācībā.

Pētījuma priekšmets ir klasifikācijas sistēmas lietotāja - jomas eksperta iesaistīšana klasifikācijas rezultātu uzlabošanā.

Darba izstādes laikā galvenie pētāmie un izstrādājamie aspekti, kā arī to savstarpējā saistība, grafiski ir atspoguļota 2. attēlā. Kā promocijas darba galvenie rezultāti jāmin vispārējs interaktīvas klasifikācijas sistēmas modelis, šī modeļa specializācija daudzkategoriju klasifikācijas uzdevumiem un modeļa realizācija sistēmas prototipa veidā.

Darba izstrādē ir izmantota projektēšanas zinātniskā metode. Teorētiskie pētījumi galvenokārt pamatojas uz literatūras analīzi. Eksperimentālie pētījumi pamatojas uz induktīvās apmācības metožu izmantošanu klasifikatoru veidošanā un programmatūras *Weka* un bibliotēkas *Mulan* izmantošanu.

Pētījuma veids ir lietišķs un eksperimentāls. Darba aprobācija ir veikta eksperimentāli – iegūtie rezultāti ar izstrādāto automatizētas (interaktīvas) klasifikācijas sistēmas prototipu ir pārbaudīti kontrolētos un atkārtojamos eksperimentos, salīdzinot interaktīvu sistēmu pret automātisku divām dažādām problēmsfērām. Darbā ir veikta arī konkrētas problēmas risināšana – balstoties uz izstrādāto modeli ir sniegts risinājums, kas atvieglo universitāšu studiju priekšmetu savstarpējās atbilstības noteikšanu (salīdzināšanu).



2. att. Darba izstrādes plāns un galvenie rezultāti

Aizstāvamās tēzes

Darbā tiek izvirzītas šādas aizstāvamās tēzes.

T1 Klasifikācijas sistēma, kas realizē interaktīvas klasifikācijas sistēmas modeli eksperta iesaistīšanai klasifikatora lietošanas posmā, ļauj samazināt nepareizi klasificēto objektu skaitu, salīdzinot ar automātisku klasifikāciju.

T2 *Piemērotākā pārliecības sliekšņa lieluma noteikšanas metode* palīdz atrast klasifikatora pārliecības slieksni, ar kuru nepareizi klasificēto piemēru skaits N ir minimāls pie izvirzītajiem eksperta ieguldāmā darba ierobežojumiem.

T3 Universitāšu studiju priekšmetu salīdzināšanā ir lietderīgi izmantot *uz induktīvo apmācību balstītu, interaktīvu, daudzkategoriju klasifikācijas sistēmu*.

Zinātniskais jaunieguvums un praktiskā vērtība

Darbs ietver interaktīvas klasifikācijas sistēmas modeļa izstrādi, kas ir balstīts uz teorētiskiem pētījumiem un pārbaudīts eksperimentāli. Tas ir izmantojams esošo klasifikācijas algoritmu papildināšanai un darbības uzlabošanai (nepareizi klasificēto piemēru skaita samazināšanai) daudzkategoriju klasifikācijas gadījumā.

Darba galvenais **zinātniskais jauniegumums** ir šāds:

- Izstrādāts interaktīvas klasifikācijas sistēmas *InClaS* (ang. v. - ***In****teractive Inductive Learning based* ***Cla****ssification* ***S****ystem*) modelis, kas apvieno interaktīvas klasifikācijas sistēmas radīšanai nepieciešamās komponentes.
- Izstrādāts *InClaS* modeļa papildinājums, kas apvieno interaktīvas daudzkategoriju klasifikācijas sistēmas radīšanai nepieciešamās komponentes.

Zinātniskais jaunieguvums ietver vairākus pakārtotus **teorētiskos rezultātus**:

- Izstrādāta klasifikācijas sistēmā ieviešamā interaktivitātes shēma.
- Izstrādāta interaktīvas klasifikācijas sistēmas uzbūve – klasifikācijas sistēmas funkcionālie moduļi, to īpašības un sasaistes.
- Izstrādātas divas klasifikatora atjaunošanas (papildināšanas) shēmas pēc eksperta veiktas klasifikācijas.
- Izstrādāts algoritms klasifikatoram neskaidru piemēru noteikšanai daudzkategoriju klasifikācijas gadījumā.
- Izstrādāta metode atbilstošākā pārliecības sliekšņa noteikšanai, pie kura klasifikatora klasificētos piemērus atzīt par nepārliecinoši klasificētiem un nodot eksperta pārziņā.
- Veikts interaktīvas daudzkategoriju klasifikācijas sistēmas projektējums sistēmu veidojošo moduļu, to ieeju un izeju apraksta veidā.
- Veikta literatūras analīze un iegūti sistematizēti apkopojumi par klasifikācijas un izglītības dokumentu salīdzināšanas tēmām.

Izstrādājot darbu, ir iegūti šādi **praktiskie rezultāti**:

Izstrādāts interaktīvas klasifikācijas sistēmas *InClaS* prototips daudzkategoriju klasifikācijas uzdevumiem. Prototips ir īpaši pielāgots studiju priekšmetu salīdzināšanai lietotājam ērtākas saskarnes nodrošināšanai.

Kā papildu rezultāts darba gaitā ir izveidota programma daudzkategoriju datu pārveidošanai dažādos atspoguļojuma formātos (atbilstoši atšķirīgajām ieejas datu formāta prasībām programmatūrā *Weka* un bibliotēkā *Mulan*).

Par darba rezultātiem ir ziņots šādās **konferencēs**:

- 2012. gada 18. – 23. novembrī. *The Fifth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services* ar referātu „Architecture of an Interactive Classification System”. Lisabona, Portugāle.

- 2012. gada 16. – 20. jūlijā. *International Conference on Machine Learning and Data Mining* ar stenda referātu „Machine Learning Approach for Study Course Comparison". Berlīne, Vācija.
- 2012. gada 16. – 18. maijā. *Sixth International IEEE Conference on Research Challenges in Information Science* ar referātu „Interactive Use of Inductive Approach for Analyzing and Developing Conceptual Structures". Valensija, Spānija.
- 2011. gada 6. – 8. oktobrī. *10th International Conference on Perspectives in Business Informatics Research* ar referātu „Artificial Intelligence in Knowledge Management: Overview and Trends". Rīga, Latvija.
- 2011. gada 13. – 16. oktobrī. RTU 52. starptautiskā zinātniskā konference, sekcija „Datorzinātne" ar referātu „Interaktīvas klasifikācijas sistēmas arhitektūra". Rīga, Latvija.
- 2011. gada 24. – 26. jūlijā. *Intelligent Systems and Agents* ar referātu „Interactive Inductive Learning based Classification System". Roma, Itālija.
- 2011. gada 7. – 10. martā. *Rethinking Education in the Knowledge Society* ar referātu „Interactive Inductive Learning Based Study Course Comparison". Monte Verita, Šveice.
- 2010. gada 11. – 15. oktobrī. RTU 51. starptautiskā zinātniskā konference, sekcija „Datorzinātne" ar referātu „Interaktīva induktīvā apmācība: pielietojums izglītības jomā". Rīga, Latvija.
- 2010. gada 5. – 7. jūlijā. *Ninth International Baltic Conference on Databases and Information Systems* ar referātu „Interactive Inductive Learning System: The Proposal". Rīga, Latvija.
- 2010. gada 27. – 28. maijā. *19th Annual Machine Learning Conference of Belgium and The Netherlands* ar referātu „Interactive Inductive Learning Service for Indirect Analysis of Study Subject Compatibility". Leuvena, Beļģija.
- 2010. gada 22. – 23. aprīlī. *16th International Conference on Information and Software Technologies* ar referātu „The Use of Inductive Learning in Information Systems". Kauņa, Lietuva.
- 2009. gada 12. – 16. oktobrī. RTU 50. starptautiskā zinātniskā konference, sekcija „Datorzinātne" ar referātu „No induktīvās apmācības uz interaktīvu induktīvo apmācību". Rīga, Latvija.

Darba rezultāti ir publicēti šādos **izdevumos**:

- Birzniece I. Architecture of an Interactive Classification System // The Fifth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC 2012), Lisabona, Portugāle, 18. – 23. novembris, 2012. IARIA, - 91.-100. lpp. Iekļauts ThinkMind datu bāzē.
- Birzniece I. Machine Learning Approach for Study Course Comparison // International Conference on Machine Learning and Data Mining (MLDM 2012), Berlīne, Vācija, 15. – 20. jūlijs, 2012. IBai publishing - 1.-13. lpp. **Iegūts atzinības raksts par labāko stenda referātu.**
- Birzniece I. Interactive Use of Inductive Approach for Analyzing and Developing Conceptual Structures // Sixth International Conference on Research Challenges in Information Science (RCIS 2012): Conference Proceedings, Spānija, Valensija, 16. –18. maijs, 2012. IEEE - 129.-134. lpp. Iekļauts **Scopus**, IEEE Xplore, DBLP datu bāzēs.
- Birzniece I. Interactive Inductive Learning Based Study Course Comparison // Proceedings of the Red-Conference: Rethinking Education in the Knowledge Society (RED 2011), Šveice, Ascona, 7. – 10. marts, 2011. Università della Svizzera italiana - 339.-347. lpp.
- Birzniece I., Kirikova M. Interactive Inductive Learning: Application in Domain of Education // RTU zinātniskie raksti. 5. sēr., Datorzinātne. - 47. sēj., 2011, RTU izdevniecība - 57.-64. lpp. Iekļauts DBLP, EBSCO datu bāzēs.
- Birzniece I. Artificial Intelligence in Knowledge Management: Overview and Trends // RTU zinātniskie raksti. 5. sēr., Datorzinātne. - 46. sēj., 2011, RTU izdevniecība - 5.-11. lpp. Iekļauts DBLP, EBSCO, io-port.net datu bāzēs.
- Birzniece I. Interactive Inductive Learning Based Classification System // Proceedings of the IADIS International Conference Intelligent Systems and Agents (ISA 2011), Itālija, Roma, 24. – 26. jūlijs, 2011. IADIS - 112.-116. lpp. Iekļauts **Scopus** datu bāzē.
- Birzniece I., Rudzājs P. Machine Learning Based Study Course Comparison // IADIS Conference on Intelligent Systems and Agents (ISA 2011), Itālija, Roma, 24. – 26. jūlijs, 2011. IADIS - 107.-111. lpp. Iekļauts **Scopus** datu bāzē.
- Birzniece I. The Use of Inductive Learning in Information Systems // Proceedings of 16th International Conference on Information and Software Technologies (IT 2010), Lietuva, Kauņa, 22. – 23. aprīlis, 2010. Technologija Kaunas - 95.-101. lpp. Iekļauts **Thomson Reuters Web of Science** datu bāzē.

- Birzniece I. From Inductive Learning Towards Interactive Inductive Learning // RTU zinātniskie raksti. 5. sēr., Datorzinātne. - 43. sēj., 2010, RTU izdevniecība - 106.-112. lpp. Iekļauts VERSITA, DBLP, io-port.net, EBSCO datu bāzēs.
- Birzniece I. Interactive Inductive Learning System // Frontiers of AI and Applications. Databases and Information Systems VI, Vol. 224, Selected Papers of Baltic DB&IS, 2011. IOS Press - 380.-393. lpp. Iekļauts ACM, DBLP, io-port.net datu bāzēs.
- Birzniece I., Kirikova M. Interactive Inductive Learning Service for Indirect Analysis of Study Subject Compatibility // Proceedings of the BeneLearn 2010, Beļģija, Leuven, 27. – 28. maijs, 2010. Katholieke Universiteit Leuven - 1.-6. lpp.
- Birzniece I. Interactive Inductive Learning System: The Proposal // Proceedings of the Ninth International Baltic Conference on Databases and Information Systems (Baltic DB&IS 2010), Latvija, Rīga, 5. – 7. jūlijs, 2010. University of Latvia Press - 245.-260. lpp.

Darba struktūra

Darbs sastāv no ievada, 6 nodaļām, rezultātu un secinājumu apkopojuma, literatūras avotu saraksta un pielikumiem. Ievadā ir pamatota risināmā problēma, definēts darba mērķis, uzdevumi un aizstāvamās tēzes. Tāpat ievadā ir izklāstīts arī darba izpildes process, galvenie rezultāti un promocijas darba saturs.

Pirmā nodaļa "Pētījuma pamatojums" apraksta līdzšinējos centienus datorizētā studiju programmu un priekšmetu salīdzināšanā, izvēršot aktuālās problēmas un definējot promocijas darba ietvaros risināmos uzdevumus un darba mērogu.

Otrajā nodaļā "Saistīto darbu analīze: iestrādnes un pastāvošās problēmas" atspoguļoti un apkopoti līdzšinējie pētījumi par darbam aktuālām tēmām. 2.1. apakšnodaļa apskata klasifikācijas uzdevumu mašīnapmācībā, galveno uzmanību pievēršot induktīvās apmācības problēmu aprakstam, daudzkategoriju klasifikācijai, tās novērtējuma mēriem un klasifikācijā risināmajām problēmām, piemēram, nespējai klasificēt jaunus piemērus. 2.2. apakšnodaļa veltīta interaktīvajām pieejām klasifikācijā, bet 2.3. apakšnodaļa apkopo līdzšinējo veikumu klasifikācijas sistēmu arhitektūru izstrādē.

Trešā nodaļa "Interaktīvas uz induktīva apmācību balstītas klasifikācijas sistēmas *InClaS*) pamatmodelis" izklāsta izstrādātās interaktīvās klasifikācijas sistēmas modeļa galvenās komponentes – ekspertam vaicājamo piemēru noteikšanu, eksperta sniegto zināšanu iekļaušanu klasifikatorā un sistēmas uzbūvi.

Ceturtā nodaļa "*InClaS* modelis daudzkategoriju klasifikācijas uzdevumam" paplašina sistēmas pamatmodeli ar daudzkategoriju klasifikācijai nepieciešamajām komponentēm, definējot algoritmu neskaidri klasificēto piemēru noteikšanai un metodi piemērotākā pārliecības sliekšņa atrašanai. Šeit ir detalizēti ar sistēmas projektēšanu un realizācijas detaļām saistītie lēmumi studiju priekšmetu salīdzināšanas uzdevumam.

Piektajā nodaļā "*InClaS* prototips" apkopotas izstrādātās modeļa komponentes un aprakstīta to realizācija sistēmas prototipa veidā. Nodaļā paskaidrotas izstrādātā *InClaS* prototipa atšķirības no citiem klasifikācijā izmantotiem rīkiem un sniegts ieskats prototipa funkcijās un lietotāja saskarnē.

Sestā nodaļa "*InClaS* modeļa novērtējums" apraksta eksperimentu plānu un iegūtos rezultātus izglītības un medicīnas jomās, salīdzinot promocijas darbā piedāvātās interaktīvās un klasiskās automātiskās klasifikācijas sniegtos rezultātus un pārbaudot izstrādātā *InClaS* modeļa lietderību.

Darbu noslēdz "Galvenie rezultāti un secinājumi", kas sniedz darba teorētisko un praktisko rezultātu, iegūto atziņu un turpināmo darbu aprakstu.

Promocijas darbam ir 11 pielikumi.

Darba 1. pielikumā ievietots svarīgāko darbā lietoto terminu skaidrojumu saraksts.

Darba 2. pielikumā aprakstīta veiktā priekšapstrāde studiju priekšmetu pilno aprakstu izmantošanai klasifikācijā – vārdu vektoru iegūšana ar programmatūru *Weka*.

Darba 3. pielikumā sniegta forma, ar kuras palīdzību tiek iegūti dati no eksperta studiju priekšmetu salīdzināšanai ar Eiropas e-kompetenču ietvara starpniecību.

Darba 4. pielikumā demonstrēta studiju priekšmetu formālo aprakstu iegūšana tiešajā un netiešajā salīdzināšanā.

Darba 5. pielikums definē rezultātus, kas iegūstami ar utilītprogrammu datņu pārveidošanai starp dažādiem daudzkategoriju atspoguļošanas formātiem.

Darba 6. pielikums sniedz induktīvās apmācības algoritmu iedalījumu.

Darba 7. pielikumā apkopotas pieejas klasifikācijas sistēmu projektēšanai.

Darba 8. pielikumā apkopoti klasifikācijas sistēmu uzbūves modeļi.

Darba 9. pielikums satur promocijas darba eksperimentos izmantoto klasifikācijas metožu un algoritmu nosaukumu saīsinājumu skaidrojumu.

Darba 10. pielikumā ievietoti iegūstamo klasifikācijas modeļu atspoguļojuma formāti dažādiem algoritmiem.

Darba 11. pielikumā atrodams pilns eksperimentu rezultātu atspoguļojums piemērotākā sliekšņa lieluma noteikšanai studiju priekšmetu salīdzināšanas uzdevumā.

1. PĒTĪJUMA PAMATOJUMS

Šajā nodaļā tiks izklāstīta problemātika, kas pamato promocijas darbā risināmo uzdevumu un nosaka nepieciešamos pētījumus un izstrādnes. Vispirms izklāstīti automātiskās klasifikācijas ierobežojumi lietošanai sfērās, kuras ir grūti formāli definējamas un nav aprakstītas ar plašu piemēru kopu (nesniedz reprezentatīvu izlasi). Kā nozīmīga darbības sfēra, kurā parādās nepieciešamība pēc automatizēta, uz induktīvo apmācību balstīta klasifikācijas risinājuma, analizēta izglītības joma, kur ir aktuāla dažādu studiju priekšmetu savstarpējas atbilstības noteikšana. Pamatojoties uz šo problēmsfēru, izvirzītas prasības izstrādājamajam risinājumam. Tomēr kā pierādījums tam, ka izstrādājamais risinājums neatbalsta tikai vienu konkrētu lietošanas jomu, tiks aprakstītas arī *konceptuāli līdzīgas* problēmas medicīnas jomā un noskaidrota plānotā risinājuma atbilstība lietošanai datiem arī šajā nozarē.

1.1. Automātiskas klasifikācijas ierobežojumi

Mašīnapmācības pieejas saskaras ar izaicinājumu jomās, kurām būtu lietderīgi izmanot mašīnapmācību tādēļ, ka tajās pastāv laikietilpīgas cilvēka veiktas aktivitātes, bet kuras neatbilst tipiskiem parametriem mašīnapmācības piemērošanai datu apjoma un strukturētības ziņā. Grūtības piemērot automātiskus klasifikācijas risinājumus galvenokārt rada divu veidu iemesli – *iekšējie* un *ārējie* (grafiski atspoguļots 1.1. attēlā).

1.1. att. Iemesli automātiskas klasifikācijas grūtībām

Iekšējos iemeslus nosaka nepilnības pašā klasifikācijas sistēmā un pieejamo apmācības datu īpatnības. Ja sākotnējie dati ir daļēji strukturētā vai citādi nepilnīgi izmantojamā formā, tad apmācībai izmantotie dati nespēs nodrošināt klasifikatoram labu vispārināšanas spēju (tiks iegūts nepilnīgs klasifikators, kas neapraksta daļu būtisku pētāmās problēmsfēras aspektu). Ja turklāt šo datu ir maz, tad klasifikatoram nepietiek avotu, kur smelties pieredzi. Šo apstākļu

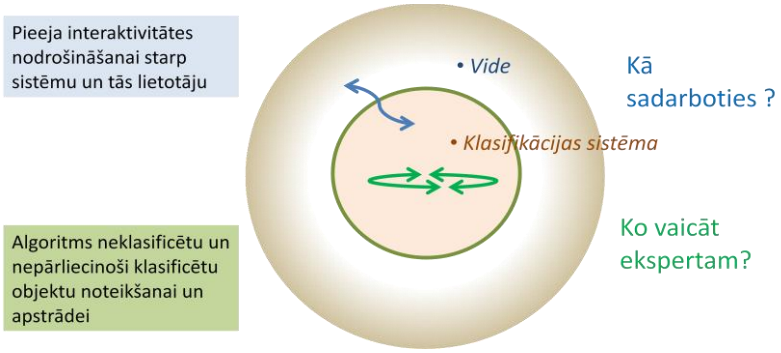
kopums arī var izraisīt neapmierinošu klasifikatora darbību – daudz nepareizi klasificētu objektu.

Ārējie iemesli, kas traucē izmantot automātiskus risinājumus, ir noteikti sistēmas lietošanas un vides faktori. Jomas eksperti, kas labi pārzina konkrētās problēmsfēras sarežģītību, parasti neuzticas pilnībā automātiskām pieejām [5]. Ja konkrētajā situācijā klasifikatora sniegtie rezultāti turklāt ir kļūdaini apmācībai izmantoto datu dēļ, tad lietotājam tiešām nav pamata uzticēties šādai sistēmai. Toties jāņem vērā aspekts, ka lietotāji ir gatavi ieguldīt zināmas pūles klasifikatora pilnveidošanā, lai iegūtu piemērotāku risinājumu [5]. Šie apstākļi rada motivāciju daļēji automātiska (automatizēta) jeb interaktīva klasifikācijas risinājuma izveidei, kas izmantotu gan mašīnapmācības sniegtās iespējas, gan ņemtu vērā iespējamos problēmsfēras radītos apgrūtinājumus un iesaistītu ekspertu sistēmas darbībā. Šī iesaiste, no vienas puses, palīdzētu pilnveidot klasifikatoru, papildinot to ar eksperta zināšanām par konkrēto sfēru, no otras puses, ļautu caurskatīt klasifikācijas rezultātus, tādējādi arī veicinot uzticēšanos sistēmai un tās sniegtajiem rezultātiem. Tādā veidā tiktu paplašinātas mašīnapmācības izmantošanas jomas un dota iespēja atvieglot cilvēka darbu tur, kur automātisko klasifikatoru nepiemērotības dēļ pagaidām prevalē manuāli risinājumi.

Detalizējot darba mērķi (*izstrādāt automatizētas klasifikācijas sistēmas modeli, kas pieļauj interaktivitāti ar ekspertu klasifikatora lietošanas laikā, ja klasifikators sastopas ar objektu, ko tas nespēj klasificēt vai nav pārliecināts par sava lēmuma pareizību*), ir redzams, ka izmaiņas līdzšinējā automātiskā klasifikācijas pieejā ir nepieciešamas gan iekšējo, gan ārējo iemeslu ietekmēšanai:

- klasifikatora darbībā, atrodot un apstrādājot klasifikatoram neskaidrus objektus;
- veidā, kā klasifikācijas sistēma mijiedarbojas ar ārējo vidi (tas ir, sistēmas lietotāju).

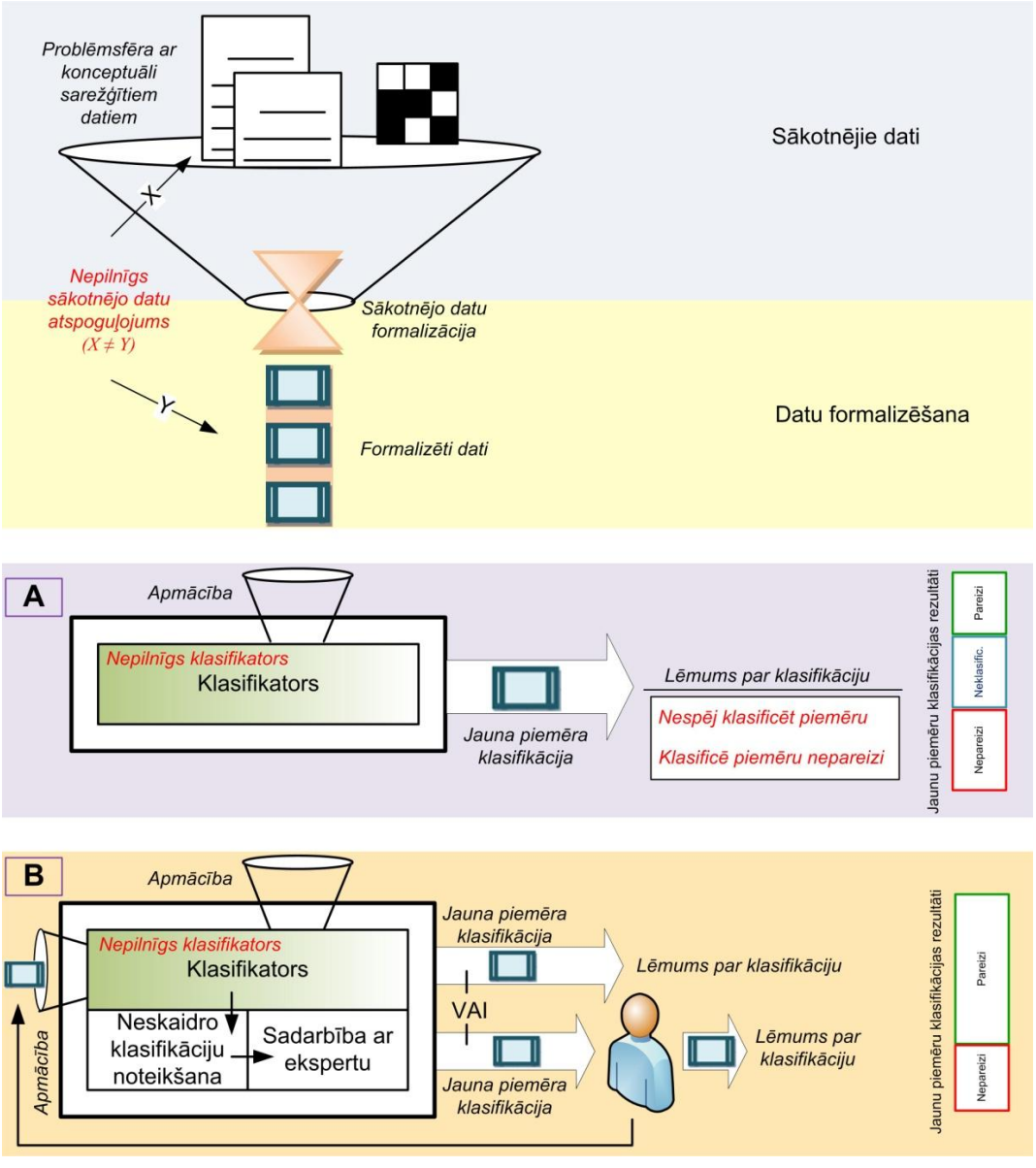
Shematiski tas ir parādīts 1.2. attēlā.



1.2. att. Izmaiņu aspekti pārejai no automātiskas klasifikācijas uz interaktīvu

Galvenie jautājumi, uz kuriem ir jāatbild interaktīvas klasifikācijas sistēmas izveidošanā, ir "Kā sadarboties klasifikācijas sistēmai ar tās lietotāju?" un "Kā noskaidrot, ko vaicāt lietotājam-ekspertam?". Tas norāda uz nepieciešamību (1) izstrādāt atbilstošu interaktivitātes pieeju un (2) algoritmu neskaidro objektu noteikšanai, kuru klasificēšanu būtu lietderīgi uzticēt ekspertam.

1.3. attēlā parādīts, ka, ja tradicionālo automātisko klasifikācijas sistēmu (attēla A daļa) aizvieto ar daļēji automātisku (automatizētu, interaktīvu) klasifikācijas sistēmu (attēla B daļa), kurā klasifikators ir papildināts ar elementiem (1) neskaidro klasifikāciju noteikšanai un (2) saziņai ar ārēju ekspertu, tad jauno piemēru klasifikāciju veic pati sistēma vai, ja tā nav droša par savu spēju pieņemt lēmumu, jomas eksperts, tādējādi likvidējot neklasificētus piemērus un samazinot nepareizi klasificēto piemēru skaitu.



1.3. att. Automātiska (A) vai interaktīva (B) klasifikācija nepilnīgu apmācības datu gadījumā

Klasifikators arī iegūst papildu apmācības iespējas, jo eksperta klasificētie piemēri var kalpot par jaunu pieredzi. Ar šādu interaktīvu klasifikācijas sistēmu tiktu ietekmēti gan iekšējie, gan ārējie iemesli automātisku klasifikācijas sistēmu lietošanas problēmām. Interaktivitāte palīdz lietotājam gan uzlabot klasifikatoru, gan veicina uzticēšanos, jo klasifikācijas process ir kontrolējams un sistēma var paskaidrot savus iegūtos rezultātus, pamatojot tos ar klasifikatorā esošajiem likumiem.

Pirms interaktīvas klasifikācijas sistēmas idejas tālākas attīstīšanas jāpaplašina pamatojums vajadzībai pēc šādas sistēmas. Nākamajā apakšnodaļā tiks detalizēti apskatīta konkrēta problēmsfēra – līdzšinējā automatizētu risinājumu pieredze un problemātika ar augstāko izglītību saistītā sfērā, lai demonstrētu pamatojumu daļēji automātisku klasifikācijas pieeju nepieciešamībai. 1.2.6. sadaļā tiks īsi analizēta problemātika arī medicīnas jomā.

1.2. Klasifikācijas uzdevumi izglītības jomā

Studiju programmu un atsevišķu studiju priekšmetu salīdzināšana ir aktuāla dažādos ar augstākās izglītības vadību saistītos uzdevumos. Sabiedrības globalizācija, pieaugošā migrācija, starptautisko studiju popularitāte, studentu apmaiņas programmas un pieaugušo apmācība ir daži no faktoriem [8], kas rada nepieciešamību salīdzināt un izvērtēt dažādu izglītības dokumentu savstarpēju atbilstību. Galvenie uzdevumi šajā jomā ir ekvivalentu priekšmetu un kredītpunktu pārskaitīšanas iespēju noteikšana starp dažādām studiju programmām, tālākizglītības kursu ieteikšana, balstoties uz studenta iepriekšējo izglītību, starptautisko grādu atzīšana, ņemot vērā programmu līdzību, priekšmetu satura, ne tikai nosaukumu, atbilstība utt.

Augstākās izglītības studiju programmu (ang.v. - *curriculum*) veido studiju priekšmetu (ang. v.- *courses*) kopums. Dažādās izglītības programmās pastāv arī moduļa (ang. v.- *module)* jēdziens, kas vai nu apvieno vairākus studiju priekšmetus vienotā idejiskā blokā, vai arī ir granularitātes ziņā mazāka vienība par studiju priekšmetu. Studiju priekšmetus iedala obligātajos un izvēles priekšmetos, dažreiz arī sīkāk, piemēram, Rīgas Tehniskajā universitātē (RTU) ir ierobežotās izvēles un brīvās izvēles priekšmetu sadaļas. Studiju priekšmeta apjomu parasti mēra kredītpunktos, kas norāda uz laiku, kas ir jāpatērē priekšmeta apguvei, un semestros. Skaidrības labad šajā darbā ir ieviestas šādas ar izglītības jomu saistītas **terminu definīcijas.**

- Izglītības (akadēmiskie) dokumenti – dažāda veida materiāli, kas raksturo augstākās izglītības saturu un novērtējumu, tai skaitā mācību materiāli, lekciju konspekti, praktisko uzdevumu apraksti, studiju plāni, mācību priekšmetu apraksti, kopsavilkumi, izglītību apliecinoši diplomi un to pielikumi.

- Mācību materiāli – izglītības dokumentu daļa, kas attiecas uz studiju priekšmetu apguvi, piemēram, lekciju konspekti.
- Izglītības diplomi – izglītības dokumentu daļa, kurā ietilpst izglītību apliecinoši diplomi un to pielikumi.

1.2.1. Nepieciešamība pēc studiju programmu un priekšmetu salīdzināšanas

Darba apjoms, ar ko jāsastopas universitāšu un citu izglītības institūciju darbiniekiem, karjeras konsultantiem (ang. v. - *academic advisors*), izglītības dokumentu salīdzināšanas institūcijām (ang. v. - *academic credential evaluation services*) vai universitāšu lietvedībām ir būtiski pieaudzis [8]. Praktiskas problēmas studiju procesā gan studentiem, gan universitāšu lietvedībām sagādā kredītpunktu pārskaitīšana par priekšmetiem, kas apgūti, atrodoties apmaiņas programmās. Studentu apmaiņas programmas sniedz vērtīgu pieredzi, ļaujot vienu vai divus semestrus studēt ārvalstu universitātē, tādējādi paplašinot studenta profesionālo un personīgo redzesloku. Tomēr izvēloties studiju posmu partneraugstskolā, ir jārēķinās, ka grādu piešķir studiju pamatinstitūcija, līdz ar to studentam jāapgūst šajā programmā paredzētais saturs. Tādēļ ir jāraugās, lai apmaiņas procesā apgūtie mācību priekšmeti vismaz daļēji atbilstu priekšmetiem, kas prombūtnes laikā būtu jāapgūst savā studiju programmā [9]. Šis uzdevums prasa priekšmetu salīdzināšanu.

Cits motivātors studiju priekšmetu salīdzināšanai ir jaunu studiju programmu izstrāde. Lai iesniegtu jaunu augstākās izglītības studiju programmu, ir nepieciešams veikt salīdzinošo analīzi ar vairākām citām līdzīgām programmām no universitātēm visā pasaulē, savukārt pilnvērtīga studiju programmu salīdzināšana prasa iedziļināšanos arī studiju priekšmetu saturā. Šis uzdevums ir ārkārtīgi laikietilpīgs, ja ir jāveic vienīgi cilvēka – programmas direktora vai lietveža – spēkiem bez atbilstošu tehnoloģiju atbalsta.

Par arvien populārāku tendenci pasaulē kļūst mūžizglītība, kas arī prasa novērtēt studēt gribētāja līdzšinējo izglītību, lai noteiktu esošās priekšzināšanas un salīdzinātu tās ar prasībām potenciālajiem tālākizglītības priekšmetiem. Iepriekšējās izglītības novērtēšana un atzīšana (ang. v. - *Prior Learning Assessment and Recognition* (PLAR)) saistās ar daudziem risināmajiem uzdevumiem [10], piemēram, iegūto kompetenču novērtēšanu, izlīdzinājumu starp atšķirīgiem augstākās izglītības modeļiem, neformālās izglītības pielīdzināšanu akadēmiskajai. Elektroniskas iepriekšējās izglītības un kompetenču novērtēšanas sistēmas vēl ir jauns un neizstrādāts virziens izglītībā, sevišķi, automatizēta novērtēšana, kas būtu balstīta uz dažādiem pieejamajiem dokumentiem. Gan iepriekšējās izglītības dokumentu, gan kompetenču novērtēšana ietilpst iepriekšējās izglītības novērtēšanas un atzīšanas kontekstā, par ko aktīvi tiek

diskutēts izglītības un zinātnes aprindās, galvenokārt, pateicoties mūžizglītības vecināšanai [10]. Sevišķi aktuāli šie jautājumi tiekot risināti Ziemeļamerikā.

Nedaudz cits salīdzināšanas aspekts ir mācību materiālu kategorizēšana, kas ir aktuāla e-apmācības sistēmās. Interneta piekļuves un e-apmācības sistēmu izplatīšanās ļauj lietot elektroniskas apmācības formas un veicina dažādu izglītībai paredzētu materiālu pieejamību. Viens no grūtākajiem un dārgākajiem uzdevumiem e-apmācības nodrošināšanā ir mācību materiālu radīšana [11]. Tomēr atšķirīgu standartu, izglītības modeļu un kultūras kontekstu dēļ mācību materiālu izplatīšana plašai izmantošanai bieži vien ir neefektīva [12]. Tieši atkārtota izmantojamība tiek norādīta kā galvenais ceļš e-apmācības sistēmu satura papildināšanai un uzturēšanai, kas aiztaupītu pūles jaunu materiālu radīšanai, pielāgošanai un izplatīšanai [11]. Daudzām izglītības institūcijām ir izstrādāts liels daudzums elektronisku mācību materiālu. Tomēr, lai šos materiālus varētu izmantot citi pasniedzēji un studenti, ir jānodrošina gan to pieejamība, gan atpazīstamība. Atpazīstamība šajā ziņā nozīmē to, ka, meklējot materiālus par kādu konkrētu tēmu, ir nodrošināta automatizēta dokumentu organizācija, balstoties uz to saturu, apskatītajām tēmām vai citiem parametriem, kas atvieglo nepieciešamo materiālu identificēšanu un atrašanu.

Augstākās izglītības sadarbība ar industriju ir cits ļoti plašs darba lauks. Nepastāvot nekādiem servisiem, kas nodrošinātu izglītības pieprasījuma un piedāvājuma monitoringu [13], aktuālā priekšstata uzturēšana par darba devēju vajadzībām un tam atbilstošu izmaiņu vai papildinājumu ieviešanu izglītības programmās ir ārkārtīgi apgrūtināta. Šajā gadījumā izskatāmo un salīdzināmo dokumentu daudzums un heterogenitāte ir vēl lielāka, iekļaujot darba devēju izsludinātās vakances, profesionālās kvalifikācijas kursu aprakstus, izglītības dokumentus, studiju priekšmetu aprakstus, profesiju standartus, amatu aprakstus utt. [14].

No apskatītā var secināt, ka studiju programmu un priekšmetu salīdzināšana ir aktuāla šādos gadījumos [8-11, 15]:

- studentu apmaiņas programmu (piemēram, ERASMUS) ietvaros;
- tālākizglītības kursu ieteikšanai;
- jaunu studiju programmu izveidē;
- partnerprogrammu izvēlē;
- studiju priekšmetu izstrādē;
- augstākās izglītības sadarbībai ar industriju;
- iepriekšējās izglītības novērtēšanai un atzīšanai;
- atkārtotas izmantojamības nodrošināšanai studiju materiāliem e-apmācībā.

Lai arī e-apmācības sistēmu izplatība un informācijas pieejamība internetā ir paplašinājusi tehnoloģiju atbalstu izglītības procesā gan studentiem, gan izglītības iestāžu administrācijai, gan pasniedzējiem, tomēr akadēmisko dokumentu caurskatīšana ir un paliek laikietilpīgs un apgrūtinošs pienākums [8]. Tāpēc dažādu izglītības dokumentu salīdzināšanas procesu ir nepieciešams atvieglot ar datorizētu risinājumu palīdzību. Mūsdienīgai sistēmai, kas spētu atbalstīt augstākās izglītības vajadzības, ir plašs attīstības potenciāls [16], gan paplašinot studentu tālākās izglītības iespējas, gan atvieglojot akadēmiskā personāla darbu izglītības dokumentu salīdzināšanā. Tehnoloģijas, kas spētu nodrošināt dažādu studiju programmu un priekšmetu atbilstības noteikšanu, ļautu sadarboties dažādām institūcijām, neskatoties uz to dažādību. Tādējādi tiktu panākts, ka atšķirīgie izglītības satura un formas standarti nekļūtu par nepārvaramu šķērsli un neierobežotu attīstību [17].

1.2.2. Problēmas sarežģītība

Augstākās izglītības sistēmā pasaulē valda liela dažādība. Dažādas ir mācību programmas gan pēc to uzbūves, gan dokumentācijas. Šī dažādība gan palīdz uzturēt unikalitāti, gan rada ierobežojumus un grūtības dažādu izglītības veidu salāgošanā. Piemēram, ja students no Ukrainas vēlas pārskaitīt studijās iegūtos kredītpunktus un Kanādā turpināt nepabeigtās bakalaura studijas, jāsastopas ar dažādiem sarežģījumiem [10]. Pirmkārt, viens studiju priekšmets no studiju programmas Ukrainā var ietvert vairākus priekšmetus studiju programmā Kanādā. Atpazīt konkrētu priekšmetu atbilstību ne vienmēr ir viegli. Otrkārt, vienā priekšmetā apgūtās tēmas var būt izkārtotas daudzos dažādos priekšmetos otrā studiju programmā. Kā noteikt priekšmetu atbilstību šajā gadījumā? Treškārt, lai salīdzinātu arī iegūtos zināšanu novērtējumus, ir jānosaka atbilstība starp dažādām vērtēšanas sistēmām. Piemēram, Ziemeļamerikā lieto atzīmju sistēmu no F līdz A, Eiropā ir dažādas skalas, gan 1 – 10, gan 2 – 5, gan apgrieztā 5 – 1 skala, kur 1 ir augstākā atzīme. Šajās sistēmās arī ir atšķirīgi sliekšņi, pie kuriem vērtējums tiek uzskatīts par sekmīgu. Visi šie apstākļi norāda, ka izglītības dokumentu salīdzināšana ir sarežģīts process, tādēļ tehnoloģiju atbalsts ir ļoti nepieciešams procesā iesaistīto cilvēku darba atvieglošanai [10].

Eiropā standartizācijas problēma jau ir apskatīta un tiek veikti dažādi pasākumi tās mazināšanai. Viens no 1999. gadā aizsāktā Boloņas procesa galvenajiem mērķiem ir vienota augstākās izglītības modeļa veicināšana Eiropā [18]. Neraugoties uz šiem centieniem, joprojām nepastāv vienots standarts mācību priekšmetu aprakstiem visās Eiropas universitātēs. Šis fakts rada galvenos sarežģījumus automātisku risinājumu ieviešanai priekšmetu salīdzināšanā. 1.4.,

1.5. un 1.6. attēlā ir parādīti fragmenti no līdzīgu studiju priekšmetu aprakstiem dažādu universitāšu *Biznesa informātikas* programmās (izmantots ar zināšanu vadību saistīta priekšmeta apraksts, kas pieejams attiecīgās universitātes mājas lapā internetā).

RTU Course "Knowledge Management Systems"

12307 Sistēmu teorijas un projektēšanas katedra

General data

Code	DSP701
Course title	Knowledge Management Systems
Course status in the programme	Compulsory/Courses of Limited Choice
Course level	Post-graduate Studies
Course type	Academic
Field of study	Computer Science
Responsible instructor	Kirikova Mārīte
Academic staff	Apšvalka Dace
Volume of the course: parts and credits points	1 part, 4.0 Credit Points, 6.0 ECTS credits
Language of instruction	LV, EN
Possibility of distance learning	Not planned
Abstract	In this course students will learn about the concepts of organisational learning and knowledge, essential factors of organisational learning, knowledge flow and networks and technologies supporting them. Human-computer interaction and interface design will be discussed. Students will learn to define knowledge management strategy, to design knowledge management systems, to plan the development of these systems and will be familiar with different knowledge management technologies.
Goals and objectives of the course in terms of competences and skills	Successful completion of this course will provide students with the content and skills necessary to: explain the impact of the nature of knowledge on the management of knowledge; understand and interpret the concept and objectives of knowledge management in terms of advanced business practices and technologies; analyse knowledge processes within an organisation in terms of organisational performance and development; identify approaches (tools and techniques) that organisations may take to make a contribution to organisation's knowledge processes; understand the need for equal consideration of technological, human and organisational aspects; identify and define the best approach of knowledge.
Structure and tasks of independent studies	In individual assignments students will explore and analyse knowledge management solutions

1.4. att. Studiju priekšmeta apraksts Rīgas Tehniskajā universitātē [19]

MBI 665 Knowledge Management and Decision Support

This course introduces students to knowledge management practices and the technologies collectively called decision support systems. To cover the most current topics affecting how individuals and organizations use computerized support in making decisions. Business applications of data warehouses, online analytical processing, group support systems, knowledge acquisition and representation, knowledge management, knowledge-based decision support and intelligent systems will be explored.
PREREQ: MBI 625 MBI 625

1.5. att. Studiju priekšmeta apraksts Ziemeļkentuki universitātē [20]

Wissensmanagement

Wirtschaftsinformatik (Master) Semester 2

Art:	Vorlesung	**Credits**	3	**Dozenten:**	Friedl
Umfang:	2 SWS	**Fachtyp:**	Pflichtfach		

Lernziele:
Die turbulenten Entwicklungen der Märkte, beispielsweise in Form einer immer weiter zunehmenden Bedeutung von Wissen für die inner- und überbetriebliche Leistungserstellung, zeigt den Trend zur Wissensgesellschaft und wissensintensiven Industrien auf.
Um der Herausforderung der zunehmenden Datenfluten Herr zu werden ist es von besonderer Bedeutung, schnell und bedarfsgerecht heterogene Informationsquellen auszuwerten, aufzubereiten und als handlungsorientiertes Wissen bereit zu stellen.
Für die dafür benötigten Wissensmanagement-Systeme sollen die Teilnehmer in der Lage sein, eine entsprechende Architektur konzipieren zu können.
Da ein integriertes Wissensmanagement nicht nur aus der technologischen Infrastruktur (oder einer Sammlung von Dokumenten) besteht, werden darauf aufbauend aktuelle Trends des Wissensmanagements diskutiert.

Inhalt
Herausforderung Wissensmanagement
Wissensbasis des Unternehmens
Bausteine des Wissensmanagements
Architektur für integrierte Wissensmanagement-Systeme
Theorien und Kodierung von Wissen
Aktuelle Trends im Wissensmanagement

Weitere Informationen/Literatur:
Probst, G./Raub, S./Romhardt, K.: Wissen managen, 5. Aufl., 2006, Gabler-Verlag.
Lehner, F.: Wissensmanagement, 1. Aufl., 2006, Hanser-Verlag.
Riempp, G.: Integrierte Wissensmanagement-Systeme, 1. Aufl., 2004, Springer-Verlag.
Daconta, M. et al.: The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management, 1. Aufl., 2003, Wiley & Sons Verlag.
Hannig, U.: Knowledge-Management und Business Intelligence, 2002, Springer-Verlag.

1.6. att. Studiju priekšmeta apraksts Rāvensburgas-Vaingartenas lietišķo zinātņu augstskolā [21]

Galvenie salīdzināšanā izmantojamie dokumenti ir studiju programmu un priekšmetu apraksti, parasti daļēji strukturēta (skat. 1.4. att. un 1.6. att.) vai nestrukturēta (skat. 1.5. att.) teksta veidā universitāšu mājas lapās internetā *html, pdf* vai teksta dokumentu formātā. Unikālais daļēji strukturētais formāts, atbilstošu gatavu apstrādes rīku trūkums un fakts, ka nepastāv vienots standarts dokumentu aprakstā ne satura, ne formas, un pat ne valodas ziņā, definē nepieciešamību pēc jaunas, problēmsfērai specifiskas pieejas [8].

Studiju priekšmetu salīdzināšana nav triviāls process ne no loģiskās izpildes, ne automatizācijas viedokļa. Šī procesa veikšana prasa intelektu, jo nepastāv iepriekš uzrakstīta instrukciju kopa, ko izmantot visiem salīdzināšanas parametriem. Tātad var runāt par dažādu mākslīgā intelekta metožu iesaistīšanu.

Vienkāršākais veids, kā skatīties uz priekšmetu salīdzināšanu, ir pieņemt priekšmetus aprakstošos dokumentus kā nestrukturētus tekstus un izmantot automātiskas teksta klasifikācijas metodes, lai noteiktu dokumentu līdzību. Tomēr studiju priekšmetu aprakstu gadījumā teksts parasti satur dažādas sadaļas, piemēram, „nepieciešamās priekšzināšanas", „priekšmeta mērķi",

„sasniedzamie mācību rezultāti" utt., un šīs sadaļas ir nepieciešams nošķirt. Turklāt daļēji strukturēts teksts var sniegt vairāk noderīgas informācijas nekā iespējams izmantot, lietojot teksta klasifikācijas tehnikas, kas uztver to kā pilnīgi nestrukturētu dokumentu [22, 23]. Piemēram, priekšmetam norādītais studiju līmenis jau spēj pateikt ļoti daudz, ja ir zināms pieņēmums, ka bakalaura līmeņa priekšmets nevar aizvietot maģistra līmeņa priekšmetu. Līdz ar to ir nepieciešams izmantot sarežģītākas tehnikas, lai izgūtu daļēji strukturētā dokumenta daļas un noteiktu to nozīmi (piemēram, priekšmeta apgūšanas priekšnosacījumi vai apskatītās tēmas), ko vēlāk izmantot salīdzināšanā.

Jārēķinās, ka „jebkura pilnībā automātiska informācijas izgūšanas sistēma, kas strādā ar nestrukturētiem vai daļēji strukturētiem dokumentiem, nespēs sniegt pilnīgi precīzus rezultātus" [8]. Turklāt cilvēki, kas ir pazīstami ar problēmsfēras sarežģītību, bieži vien nemaz neuzticas automātiskiem risinājumiem. Viņi ir gatavi ieguldīt arī savas pūles sistēmas darbībā, bet sagaida, ka ieguldījums tiks efektīvi izmantots [5], piemēram, sistēmas apmācībā un pilnveidošanā. Tātad runa varētu būt par automatizētas jeb daļēji automātiskas sistēmas nepieciešamību izglītības dokumentu salīdzināšanas atbalstam.

Kopumā studiju priekšmetu salīdzināšanas sarežģītību raksturo šādi apstākļi, kas ir arī saistīti savā starpā:

- nav vienota aprakstošo dokumentu formāta;
- izglītības dokumenti ir daļēji strukturēti;
- informācijas izgūšanas metodes iegūst nepilnīgus rezultātus;
- pastāv neviennozīmīga studiju priekšmetu savstarpējā atbilstība;
- salīdzināšanu ir grūti formāli definēt;
- salīdzināšanas rezultātam ir jābūt uzticamam;
- jomas pazinēji neuzticas pilnīgi automātiskiem risinājumiem.

Līdzšinējie centieni radīt sistēmas dažādu augstākās izglītības dokumentu salīdzināšanas atbalstam tiks apskatīti nākamajā apakšnodaļā.

1.2.3. Līdzšinējie risinājumi

Universitāšu studiju programmu savstarpējās atbilstības automatizēta noteikšana ir dažādās valstīs pētīts temats [8, 10-12, 15]. Viena no pieejām ir izstrādāta Latvijas (Rīgas Tehniskā universitāte) un Francijas (Monpeljē II universitāte) Osmozes programmas sadarbības projekta "Servisi mācību programmu salīdzināšanai" (*SECC: Services for Curricula Comparison*) ietvaros. Projekta mērķis bija radīt metodoloģiju un realizēt tīmekļa servisu

prototipus datorizētai studiju programmu salīdzināšanai. Automatizēta salīdzināšana tiek veikta, analizējot un atspoguļojot jēdzienu kartes, kas iegūtas no dokumentiem elektroniskā formā [15]. Studiju programmas tiek atspoguļotas jēdzienu karšu veidā, kuras tiek salīdzinātas ar shēmu savstarpējās atbilstības noteikšanas [24] tehnikām. Projekta ietvaros ir izstrādāta sistēma studiju programmu struktūru salīdzināšanai, specificējot programmas līdz atsevišķu mācību priekšmetu nosaukumiem, bet neapskatot to saturu. Ir veikta gan tiešā programmu salīdzināšana, gan netiešā salīdzināšana, izmantojot standartizētu kompetenču ietvaru kā starpslāni [15]. Tomēr, lai varētu veikt studiju programmu pilnvērtīgu salīdzināšanu, ir jāskatās arī uz programmā iekļauto priekšmetu saturu, kas šī projekta ietvaros nav darīts.

Brunsvikas universitātē (*University of New Brunswick*) Kanādā tiek pastāvīgi strādāts izglītības programmu salīdzināšanas jomā, konkrēti, pie izglītības diplomu atzīšanas, elektroniskiem padomdevējiem (ang. v. - *e-advising*) un mācību materiālu klasifikācijas. Šajā jomā aktīvi darbojas Biletskis (*Biletskiy*). Viņa un kolēģu agrākie darbi ir saistīti ar mašīnapmācības stratēģijas izstrādi studiju priekšmetu klasificēšanai noteiktās apakšjomās (priekšmetos) kādas jomas (studiju programmas) ietvaros [12]. Autori apraksta metodoloģijas izstrādi tādu mācību materiālu klasifikācijai, kas doti dažādās formās bez labi definētiem metadatiem. Viņu pieeja definē divas galvenās fāzes; pirmkārt, informācijas izgūšanu, otrkārt, paša klasifikatora veidošanu. Izgūtā informācija tiek glabāta noteiktas formas *XML* dokumentā. Klasifikācijas posms ietver apakšjomu (klašu) identificēšanu, priekšmetu aprakstu izvēli apmācības kopas veidošanai, kā arī pieeju normalizētu atslēgvārdu biežuma tabulu ģenerēšanai katrai no klasēm. Tādējādi jauna mācību dokumenta klasifikācija tiek veikta, atrodot mazāko attālumu līdz kādai no klasēm (autori tos sauc par klasteriem), t.i., apakšjomai. Pētnieki norāda to, cik svarīgi ir izgūt un saglabāt mācību dokumentu jēgpilnā formātā, piemēram, *XML*, tā, lai to vēlāk varētu piesaistīt vajadzīgajam kontekstam, izmantojot automātiskas vai daļēji automātiskas metodes [12]. Piedāvātā mašīnapmācības pieeja mācību materiālu klasificēšanai sastāv no šādiem soļiem:

1. apakšjomu identificēšana vienā vai vairākās sfērās, piemēram, studiju priekšmetu izdalīšana vienas studiju programmas ietvaros;
2. manuāla priekšmetu aprakstu un studiju programmas aprakstu izvēle apmācības veikšanai;
3. mašīnapmācības metožu izmantošana, lai iegūtu atslēgvārdu biežuma bibliotēku katram priekšmetam un studiju programmai kopumā;
4. atslēgvārdu biežuma tabulas normalizēšana, lai atšķirīgs atslēgvārdu skaits sfērā un apakšjomās neietekmētu rezultātus;

5. normalizēto atslēgvārdu tabulas iegūšana jauniem nejauši izvēlētiem priekšmetu aprakstiem;
6. īsākā attāluma atrašana starp nejaušajiem priekšmetu aprakstiem un apakšjomām (aprēķinot vidējo kvadrātisko novirzi), izmantojot normalizēto atslēgvārdu tabulu.

Jaunākos pētījumos šie autori ir pievērsušies akadēmiskai elektroniskai padomdošanai (ang. v. - *academic e-advising*) un karjeras konsultāciju atbalstam [8]. Šim uzdevumam ir nepieciešama vienota datu bāze ar visu izglītības institūciju piedāvātajiem studiju priekšmetiem un to aprakstiem, bet tāda šobrīd nav izveidota. Toties informācija, ko šādai datu bāzei vajadzētu saturēt, ir pieejama gandrīz visu izglītības institūciju mājas lapās internetā akadēmiskā kalendāra vai priekšmetu aprakstu veidā. Tādējādi autori uzskata, ka, ja esošos dokumentus būtu iespējams izmantot kopīgas datu bāzes izveidei, tad elektronisks daļēji automātisks akadēmiskais padomdevējs varētu tikt ieviests.

Kā galvenā problēma šajā ceļā tiek uzsvērts priekšmetu apraksta formāts, kurš ir ārkārtīgi atšķirīgs, līdz ar to apgrūtina būtiskās informācijas atrašanu un izgūšanu. Lai informācija būtu mašīnlasāma, tai jābūt atspoguļotai, piemēram, *XML* formā, bet vairums priekšmetu aprakstu mājas lapās ir daļēji strukturētā *HTML* formātā. Pētījums noved pie priekšmetu aprakstu datu izguvēja (ang. v. - *course outline data extractor* (*CODE*)) izstrādes – sistēmas, kas spētu *HTML* dokumentus pārvērst *XML* formā. Šī sistēma sastāv no četrām galvenajām fāzēm:

1. **priekšapstrāde**;
2. ***HTML* analizēšana un dokumentu objektu modeļa (*DOM*) izveide**. Šī posma mērķis ir pārveidot priekšmetu aprakstu tā, lai būtu ērti darboties ar informācijas izgūšanas metodēm nākamajā solī. *DOM* veidošana ļauj izvairīties no teksta apstrādes, pārveidojot aprakstu koka struktūrā, kur katra *HTML* iezīme un teksta lauks ir mezgla punkts;
3. **informācijas izgūšana**. Tā kā priekšmetu aprakstu struktūra ir tik unikāla, nav iespējams izgūt visu būtisko informāciju tikai ar vienu izguves metodi, tāpēc tiek pielietota virkne metožu, kuras var iedalīt trīs kategorijās:
    a. vispārējā satura izgūšana;
        i. atslēgvārdu identificēšana, izmantojot bibliotēku un terminu vārdnīcu;
        ii. tipveida vērtību (ang. v. - *regular expressions*) atpazīšana.
    b. kopsavilkuma (piemēram, apraksta) izgūšana. Šeit [8] autori izmanto pieņēmumu, ka *DOM* mezglu punkts, kurš satur visvairāk sfērai specifisko atslēgvārdu, ir priekšmeta kopsavilkums;

c. priekšmetā apskatīto tematu izgūšana. Tematu izgūšana nav vienkārša tajos gadījumos, kad tie nav uzskaitīti strukturēta saraksta vai tabulas veidā;

4. **apakšjomas noteikšana.** Šis solis balstās uz pieņēmumu, ka priekšmetus var uzskatīt par ekvivalentiem, ja tiem ir līdzīgs saturs un tie pieder vienai apakšjomai. Apakšjomas noteikšanai tiek izmantota iepriekš [12] aprakstītā pieeja.

Lai arī apakšjomas noteikšana ietver klasifikāciju, tomēr šo autoru darbs galvenokārt ir vērsts uz nepieciešamās informācijas izgūšanu un organizēšanu tālākai izmantošanai, nevis salīdzināšanas mehānismu definēšanu un pašu priekšmetu atbilstības noteikšanu.

Viens no iespējamajiem scenārijiem plānotās elektroniskās padomdošanas sistēmas lietošanā ir šāds. Studēt gribētājs iesniedz savas iepriekšējās izglītības dokumentu kopijas. Akadēmiskais padomdevējs piedāvā programmu tālākai izglītībai. Gan iepriekšējās, gan iespējamās tālākās izglītības dokumentos minētie mācību priekšmeti tiek sameklēti vienotajā priekšmetu datu bāzē. Jā kāda no priekšmetiem tur vēl nav, to ir iespējams norādīt ārējā resursā, un *CODE* sistēma dokumentu apstrādās un pievienos datu bāzei. Kad visa būtiskā informācija ir norādīta, priekšmeti tiek salīdzināti pēc vairākiem kritērijiem (nosaukuma, studiju līmeņa, apraksta, tematiem un apakšjomas) un papildu zināšanām (ontoloģijām un likumiem). Katram kritērijam ir savs svars, kuru var mainīt un pielāgot atkarībā no pieejamās informācijas. Salīdzināšanas rezultātā potenciālās mācību programmas priekšmeti tiek iedalīti trīs kategorijās. Pilnīgi saskanošie priekšmeti norāda iespēju pārskaitīt kredītpunktus, nedaudz saskanošie nosaka atbilstošas priekšzināšanas priekšmeta apguvei, bet neskaidras saskanības gadījumā ir nepieciešama eksperta līdzdalība sīkākai analīzei [8].

Nākamajā sistēmas attīstības solī tiek piedāvāta pieeja ekspertu sistēmas izveidošanā akadēmisko diplomu un kompetenču elektroniskai novērtēšanai (ang. v. - *e-Assessment*) [10]. Kompetenču noteikšana ir samērā jauns virziens iepriekšējās izglītības novērtēšanas laukā. Līdz ar mūžizglītības aktualizāciju un centieniem atzīt neformālās izglītības veidā iegūtās prasmes, praktiskās darbības ceļā iegūto kompetenču pielīdzināšana formālās izglītības iegūstamajiem mācību rezultātiem arī tiek ietverta šīs sistēmas redzeslokā. Šī pieeja izmanto semantiskā tīmekļa tehnoloģijas, lai pārveidotu akadēmiskos diplomus starp dažādām institūcijām un veidotu sakarības starp personas kompetencēm un iegūtajiem diplomiem, ko pēc tam var izmantot iespējamās tālākizglītības noteikšanai. Šis sarežģītais process ir pieskaitāms jomai, kas nodarbojas ar strukturāli un semantiski heterogēnu informācijas avotu apstrādi un piegādāšanu lietotājiem (padomdevējiem, izglītības jomas ekspertiem un studentiem).

Elektroniskās novērtēšanas ekspertu sistēmas arhitektūrā ir jārisina vairāki jautājumi tās komponenšu realizēšanai. Galvenā sistēmas sastāvdaļa ir ontoloģija, kas atspoguļo mācību

programmu, kredītpunktu, vērtēšanas un kompetenču shēmas un ievieš sistēmā semantiku. Ontoloģiju ir jāspēj arī radīt no dokumentiem, ko iegūst no mācību iestādēm un studentiem. Lai varētu salīdzināt atšķirīgas studiju programmas un to sastāvdaļas, ir nepieciešams radīt un uzturēt pārvēršanas likumus, kā arī vaicājumu mehānismu izglītības dokumentu pārvēršanai starp dažādiem standartiem un vērtēšanas skalām.

Lai arī sistēmas mērķi ir plaši, tās autoru piedāvātajā variantā [10] elektroniskā novērtēšanas sistēma salīdzināšanā izmanto tikai studiju priekšmetu nosaukumus, kredītpunktu skaitu un atzīmes, kā arī vienkāršu vārdu līdzības mēru, kas, kā atzīst arī paši autori, nenodrošina pietiekamu precizitāti un uzticamību veiktajiem salīdzinājumiem.

Ir precedenti arī samērā vienkāršu teksta analīzes tehniku veiksmīgam lietojumam izglītības dokumentu salīdzināšanā. Mācību materiālu organizēšanai Alvess un Figeira [11] ir piedāvājuši izmantot klasterizācijas pieeju, lai organizētu dokumentus semantiski līdzīgās grupās. Šis mehānisms ir papildināts arī ar sociālās klasifikācijas palīdzību, izmantojot iezīmes (atslēgvārdus). Piedāvātais mehānisms organizē klasteros dažāda formāta dokumentus no e-apmācības sistēmas, izmantojot k-vidējo klasterizāciju, kas darbojas ar vektoru telpas modeli un gudro atlasi (ang. v. - *smart seeding*), piešķirot vārdiem dažādu nozīmību.

Pieejas pārbaude ir veikta šādiem eksperimentiem:

1. manuāla dokumentu organizācija loģiskās grupās (atbilstoši studiju priekšmetiem), iegūstot *k* klasterus;
2. šo pašu dokumentu automātiska klasterizācija, balstoties uz vektoru telpas modeli, norādot klasteru skaitu *k;*
3. automātiskās klasterizācijas rezultātu salīdzināšana ar eksperta veiktās grupēšanas rezultātiem.

Sākotnējie sistēmas pārbaudes rezultāti ir norādījuši samērā labu klasterizācijas kvalitāti, kā arī to, ka papildus iezīmju lietošana nav uzlabojusi darbības rādītājus.

Lai atspoguļotu līdzšinējos risinājumus studiju programmu un priekšmetu salīdzināšanas jomā no dažādiem skatu punktiem, promocijas darbā ir sastādītas apkopojošas tabulas. Apskatītas risinājumu variācijas pēc to mērķa lietojuma (1.1. tabula), galvenajām izmantotajām tehnikām (1.2. tabula) un akadēmisko dokumentu salīdzināšanā izmantotajiem parametriem (1.3. tabula).

1.1 . tabula

Izglītības dokumentu salīdzināšana - variācijas pēc mērķa lietojuma

Mērķa lietojums	Avots
Studiju programmu salīdzināšana	[15]
E-apmācības sistēmā esošo dokumentu grupēšana pa studiju priekšmetiem	[11]
Priekšmetu aprakstu klasifikācija pa sfērām (apakšnozarēm)	[12]
Iepriekšējās izglītības novērtēšanas elektronisks padomdevējs	[8, 10]
Dažādu izglītības dokumentu un kompetenču atbilstības noteikšana, ietver priekšmetu savstarpēju atbilstības noteikšanu	[10]

1.2. tabula

Izglītības dokumentu salīdzināšana - variācijas pēc galvenajām izmantotajām tehnikām

Tehnika	Avots
Jēdzienu kartes	[15]
XML shēmu saskanības noteikšana (*YAM++* un *WebSmatch*)	[15]
Ontoloģijas (*KIVIS*, *Protege*, *OWL*)	[15]
Klasterizācija(*k-means++* algoritms, vidējās kvadrātiskās kļūdas minimizēšana)	[8, 11, 12]
Teksta klasifikācija balstoties uz normalizētu biežāk sastopamo atslēgvārdu vai vārdu biežumu	[8, 11, 12]
CODE - problēmsfērai specifisks parsētājs no *HTML* uz *XML*	[8]
Ekspertu sistēma, semantiskais tīmeklis, ontoloģijas, pārveidošanas likumi	[10]

1.3. tabula

Izglītības dokumentu salīdzināšanā izmantotie parametri

Parametrs	Avots
Studiju programmas uzbūve	[15]
Galvenie priekšmeta temati (ang. v. - *topics*)	[11]
Viss dokumenta saturs (nešķirojot sadaļas un arī pašus dokumentus)	[11]
Priekšmeta apraksts	[12]
Priekšmeta nosaukums, studiju līmenis, apraksts, temati un apakšjomas, ontoloģijas un likumi	[8]
Priekšmeta nosaukums, kredītpunkti, atzīmes	[10]

1.2.4. Pastāvošās problēmas un pilnveidošanas iespējas

Līdz šim aprakstītie piemēri pierāda, ka attīstības potenciāls un nepieciešamība pēc dažādu izglītības dokumentu salīdzināšanas pastāv. Tomēr praktisko sistēmu realizāciju apgrūtina daudzi pilnveidojami vai neatrisināti jautājumi.

Fakts, ka studiju priekšmeti ir aprakstīti daļēji strukturētu dokumentu veidā un šiem aprakstiem nav vienotas sistēmas dažādu izglītības institūciju starpā, ir novedis pie divu galēju pieeju lietošanas:

- nestrukturēta teksta apstrādes metožu lietošana dokumentu klasifikācijai, pilnīgi atsakoties no teksta strukturēšanas;

- sarežģītu un nepilnīgu manipulāciju ieviešana formālu struktūru izgūšanai, cenšoties saglabāt visu dokumenta kontekstu.

Pirmajā gadījumā tiek zaudēts daudz potenciāli vērtīgas salīdzināšanas informācijas, savukārt otrajā gadījumā lielākā uzmanība tiek pievērsta informācijas izgūšanai, nevis salīdzināšanas mehānismiem. Lai arī informācijas izgūšana ir svarīga un sarežģīta priekšmetu salīdzināšanas uzdevuma daļa, ir būtiski pilnveidot arī pašu salīdzināšanas mehānisma kodolu.

Elektroniskās padomdošanas sistēmas autori [8] norāda, ka sistēmas rezultātus noteikti varētu uzlabot, palielinot apmācības kopu, kā arī iesaistot ekspertu. Piemēram, ieviešot vienkāršu mehānismu manuālai izgūto datu pārbaudei un papildināšanai, būtu iespējams uzlabot tālākā darbībā izmantotās informācijas kvalitāti. Klasifikācijai izmantojamās datu kopas palielināšana arī prasa manuālu darbu, jo sagatavot apmācības piemērus var tikai problēmsfēras eksperts, līdz ar to pieejamo paraugu daudzums vienmēr būs ierobežots.

Kā apstiprina [8], izglītības dokumentu salīdzināšana ir sarežģīts uzdevums gan ekspertiem, gan datorsistēmām, tāpēc šī procesa automatizācija prasa specifiskas pieejas un eksperta līdzdalību. Tas tiks ņems vērā, izstrādājot promocijas darbu.

1.2.5. Promocijas darbā risināmais uzdevums

Analīze apliecina, ka izglītības dokumentu salīdzināšanā, lai iegūtu uzticamu rezultātu, risinājumam ir jābūt automatizētam, bet ne automātiskam, jo bez eksperta līdzdalības nevar iegūt pieņemamu rezultātu. Ir vērts pievērst uzmanību arī faktam, ka, neskatoties uz to, ka starp dažādām studiju programmām nepastāv viennozīmīga priekšmetu savstarpējā atbilstība, ko apstiprina arī pētītās literatūras autori [10, 12], studiju priekšmetu salīdzināšana līdzšinējos risinājumos ir veikta, meklējot 1 pret 1 atbilstību. Nevienā no realizētajām sistēmām līdz šim nav apskatītas iespējas studiju priekšmetu vienā studiju programmā attiecināt vienlaicīgi uz vairākiem studiju priekšmetiem citā programmā jeb izmantot daudzkategoriju (ang. v. - *multi-label*) klasifikāciju.

Priekšmetu salīdzināšanā bieži izmantots mērs ir leksikografiskā (vārdiskā) līdzība. Šāds salīdzināšanas veids ir nepilnīgs, sevišķi, ja tiek izmantots kā vienīgais līdzības kritērijs. Vārdnīcu un sinonīmu izmantošana terminu salīdzināšanā ļauj rezultātus ievērojami uzlabot, tomēr to joprojām nevar saukt par priekšmetu salīdzināšanu pēc to būtības. Eksperts, veicot salīdzināšanu, pievērš uzmanību studiju priekšmetā apskatītajiem tematiem (ja tādi ir pieejami), sasniedzamajiem mācību rezultātiem, kredītpunktu skaitam un citiem lielumiem, no kuriem tikai daži automātiskā veidā ir pilnvērtīgi apstrādājami ar leksikografisko salīdzināšanu. Lai sistēmai iemācītu salīdzināt studiju priekšmetus, ir jāpiekļūst tuvāk priekšmeta būtībai.

Lai arī ir uzsvērta cilvēka iesaistīšanās nepieciešamība, līdzšinējos risinājumos izmantotās mašīnapmācības metodes neveicina iesaistītās personas izpratnes pilnveidošanu un atgriezeniskās saites uzlabošanu. Priekšmetu salīdzināšanā izmantotās metodes, piemēram, klasterizācija, nesniedz lietotājam saprotamu sistēmas iegūtā modeļa skaidrojumu. Ja sistēma jauna mācību dokumenta līdzību kādam citam pamato ar attālumu līdz klasterim, kas lietotājam ir tikai abstrakts jēdziens, tad lietotājam ir grūti gan novērtēt šī risinājuma pareizību, gan uzticēties tam. Lai palīdzētu klasifikatoram savu darbību pilnveidot, lietotājam vai ekspertam ir nepieciešams saprotamāks paskaidrojums, piemēram, likumu formā. Šādu iespēju sniedz klasifikācijā izmantojamās induktīvās apmācības algoritmi.

Izdarot secinājumus par šīs sfēras problemātiku, promocijas darbā tiks izskatīts uzdevums par

- induktīvās apmācības algoritmos balstītu;
- interaktīvu;
- daudzkategoriju

studiju priekšmetu salīdzināšanas atbalsta sistēmas kodola – automatizēta priekšmetu salīdzināšanas risinājuma – izstrādi.

Izglītības dokumentu salīdzināšana ir tikai viena no daļēji strukturētu dokumentu izmantošanas jomām. Daļēji strukturētu tekstu apstrāde nav problēma tikai izglītības dokumentu sakarā. Izstrādājot pieeju izglītības dokumentu salīdzināšanai, būtu iespējams iegūt labumu arī citās jomās, tādēļ tiks piedāvāta koncepcija vispārīgam lietojumam, to pārbaudot studiju priekšmetu salīdzināšanas problēmai.

Informācijas izgūšana no daļēji strukturētiem dokumentiem ir disciplīna, kas ir cieši saistīta ar studiju priekšmetu salīdzināšanu, jo priekšmetu apraksti ir organizēti daļēji strukturētu dokumentu veidā. Izstrādājot risinājumu studiju priekšmetu salīdzināšanu, jāpatur prātā problemātika praktiskai informācijas izgūšanai no daļēji strukturētiem dokumentiem. Tomēr šī darba ietvaros tiks apskatītas priekšmetu salīdzināšanas iespējas ar jau atlasītu informāciju, neiedziļinoties informācijas iegūšanas tehnoloģijās no daļēji strukturētiem dokumentiem.

Promocijas darba rezultātā nav plānots iegūt sistēmu, kas spētu viennozīmīgi un patstāvīgi noteikt, vai studiju priekšmeti ir savstarpēji aizvietojami. Būtisku lēmumu pieņemšanu nevar pamatot tikai ar datorsistēmas veiktu „vienīnieciņu un nullīšu salīdzināšanu", bet, ja ekspertam pašam būtu jāizpēta liels daudzums dokumentu, tad konkrētu virzienu norādīšana un caurskatāmā materiāla būtiska samazināšana var izrādīties lietderīgs pienesums, ko var sniegt automatizēts risinājums [25]. Lai atvieglotu ikdienas darbu ir nepieciešama

datorizētu rīku ieviešana lietvežu, pasniedzēju vai citu sfēras ekspertu veiktas priekšmetu salīdzināšanas atbalstam, uzlabojot un papildinot esošos automātiskos risinājumus šim uzdevumam.

Apkopojot iepriekš minētos apstākļus, studiju priekšmetu salīdzināšanas uzdevumu kā prasības pret mašīnapmācības risinājumu definē šādas ***raksturīgās īpašības***:

- iegūtie rezultāti ir jāsaprot klasifikatora lietotājam un ekspertam;
- pieejamā apmācības kopa ir maza;
- sākotnējie dati ir daļēji strukturēti vai nestrukturēti;
- problēmsfērā ir raksturīgas daudzas klases, kuras sastopamas vienlīdz bieži;
- objekts var piederēt vienlaicīgi vairākām klasēm.

Tāpat kā jebkura problēmsfēra, arī studiju priekšmetu salīdzināšana definē savas prasības un ierobežojumus, kas jāņem vērā risinājuma izvēlē. 1.7. attēlā atspoguļots izskatāmais risinājuma apgabals. Tas nosaka:

- problēmas un ierobežojumus, kas jāņem vērā risinājuma izvēlē;
- risinājumu koncepcijas, kas saskan ar pastāvošajiem ierobežojumiem.

Pētījumu gaitā katrai risinājuma komponentei ir jānosaka iespējamie risinājuma varianti, jāizanalizē tie un jāizvēlas atbilstošākais. Risinājumu koncepciju izpēte ļauj izvirzīt šādus esošo risinājumu veidus katrai komponentei.

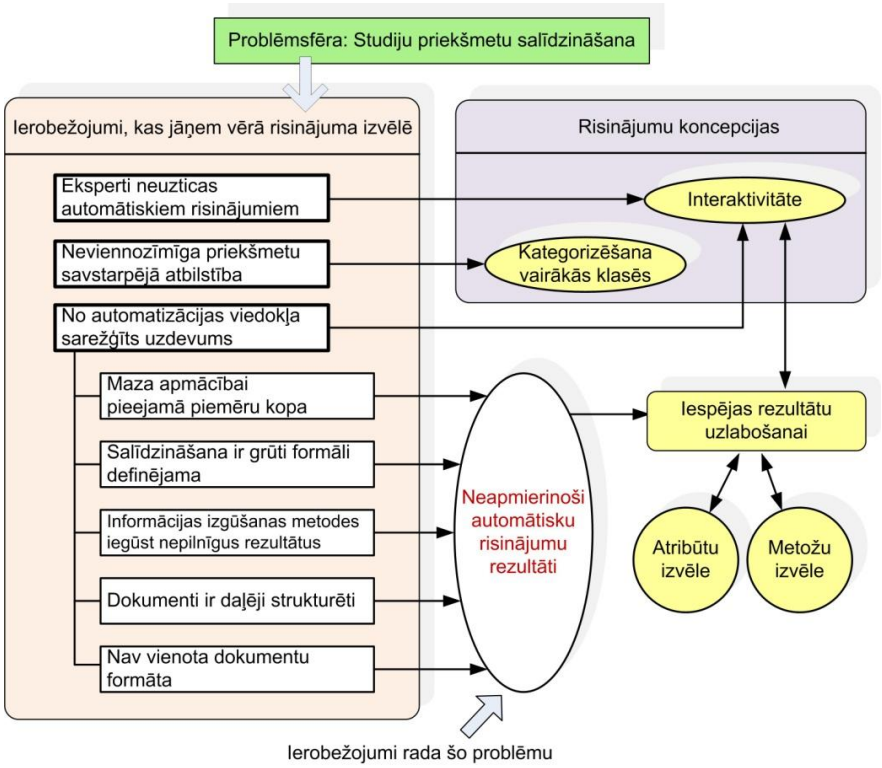
Daudzas klases (sīkāk apskatīts darba 2.1. nodaļā):

- daudzkategoriju (ang. v. - *multi-label*) klasifikācija;
- izplūdusī (ang. v. - *fuzzy*) klasifikācija;
- pārliecības (ang.v - *credal*) klasifikācija.

Interaktivitāte (sīkāk apskatīts darba 2.2. nodaļā):

- dažādas esošās pieejas eksperta iesaistīšanai klasifikācijā;
- aktīvā mācīšanās (ang. v. - *active learning*);
- *Ripple down* likumi.

Lai **uzlabotu klasifikācijas rezultātus**, nepieciešams gan izvēlēties piemērotākos sfēru aprakstošos atribūtus un klasifikācijas metodes, gan atkāpties no automātiska risinājuma lietošanas, ieviešot interaktivitāti. Darbs balstās uz pieņēmumu, ka studiju priekšmetu salīdzināšanai ir nepieciešams automatizēts risinājums - pusceļš starp manuālu un automātisku pieeju.



1.7. att. Studiju priekšmetu salīdzināšanas problēmu un risinājumu apgabals

Tā kā studiju priekšmetu salīdzināšanas problēma no mašīnapmācības viedokļa ir definēta ar *raksturīgajām īpašībām*, tad darbā risināmais uzdevums var tikt abstrahēts no konkrētās studiju priekšmetu problēmsfēras, nosakot, ka tiek meklēts risinājums visām sfērām, kurās pastāv šādas īpašības. Lai pārliecinātos, ka pastāv šādas konceptuāli līdzīgas problēmas un izglītības joma nav unikāls gadījums, nākamajā darba sadaļā īsi apskatīta situācija medicīnas jomā.

1.2.6. Līdzīgi klasifikācijas uzdevumi medicīnas jomā

Izglītības sfēra ar nepieciešamību salīdzināt studiju programmas un priekšmetus nav vienīgā, kam ir tādas raksturīgās īpašības, lai būtu ieteicams klasifikācijā izvēlēties interaktīvu pieeju. Konceptuāli līdzīgi uzdevumi eksistē arī medicīnā. Medicīnā sastopamie klasifikācijas uzdevumi ir dažādi – gan dažādu slimību konstatēšana, balstoties uz pacientu simptomiem, gan medicīnisko attēlu analīze, lai identificētu anomālijas, gan medicīnas dokumentu organizācija. Medicīnas diagnostika, sava augstā atbildības sloga dēļ, ir pieskaitāma pie tiem klasifikācijas uzdevumiem, kur klasifikatora lēmuma pieņemšanas caurskatāmība medicīnas speciālistam ir būtiska. Tas ir pirmais apstāklis, kas norāda uz nepieciešamību lietot induktīvās apmācības algoritmus, ja tiek izmantota automātiska vai daļēji automātiska klasifikācija. Tiek runāts par

lietotājam labi lasāmiem likumiem, kas nozīmē gan likumu formāta uztveramību (piemēram, IF – THEN formā), gan likumu skaitu, kurš nedrīkst būt pārāk liels [26].

Medicīnas diagnostikā realitātē ir iespējams saskarties ar vairākām diagnozēm vai slimībām vienam pacientam. Tomēr vairums radīto risinājumu apskata klasifikācijas uzdevumu, kura mērķis ir noteikt tikai vienas slimības esamību vai neesamību, ignorējot iespējamās blakus slimības. Literatūrā sastopami nedaudzi piemēri vienlaicīgi vairāku diagnožu uzstādīšanai vienam pacientam vai medicīnas artefaktam (formāli - vairāku klašu piešķiršanai vienam piemēram). Tā, piemēram, [27] apraksta metodi lielākā medicīnas rakstu repozitorija MEDLINE izmantošanai, ņemot vērā tā rakstu klasifikāciju vienlaicīgi vairākās kategorijās. Iespējamā saistība starp klasēm medicīnas uzdevumā tiek izmantota pieejā [28], kas lieto Beijesa teorēmu daudzkategoriju klasifikatora izveidē. Pacientu diagnožu noteikšana atbilstoši ICD-9 diagnožu kodiem, balstoties uz medicīnas kabinetā izdarītajiem audioierakstiem pacienta vizītes laikā, kas apskatīta [29], ir dažādā ziņā sarežģīts uzdevums. Pirmkārt, teksti ir brīvas formas piezīmes par pacientu stāvokli, izmeklēšanu, veiktajām procedūrām utt., ko ierakstījuši dažādas kvalifikācijas mediķi. Otrkārt, katram tekstam var būt piešķirti vairāki kodi. Veiktajos eksperimentos iegūti 978 ieraksti un piešķirti pavisam 140 dažādi diagnožu kodi, pie tam, lielākajai daļai diagnožu ir pavisam maz atbilstošo piemēru. Līdz ar to apmācības kopa šādam klašu skaitam ir samērā maza. Autori arī atzīst, ka risinājumam, iespējams, jābūt daļēji automātiskam, un esošie risinājumi ne vienmēr ņem vērā problēmsfēras daudzkategoriju dabu. Medicīnas datu kopa, kas raksturo pacientu vizīšu laikā iegūtajos tekstuālajos aprakstos biežāk sastopamos vārdus un piešķirtās diagnozes, 2007. gadā arī tika izmantota *Skaitļošanas medicīnas centra dabiskās valodas apstrādes konkursā* un šobrīd tiek lietota kā datu kopa dažādu metožu salīdzināšanai un pārbaudei [30]. Daudzkategoriju klasifikācijas uzdevumu risināšanas nepieciešamību medicīnas diagnostikā norāda arī disertācija [31], kas veltīta klīnisko tekstu analīzei ar mašīnapmācības palīdzību. Savukārt avotā [32] medicīnas sfērā tiek izmantota klasifikatoru kombinācija, lai sniegtu lietotājam lēmuma atbalsta pieeju un parādītu 20 visticamākās atrastās klases konkrētajam piemēram. Uzdevuma nostādne iekļauj tekstuālus un neviennozīmīgi strukturētus sākotnējos apmācībai izmantojamos datus, kas sniegti dabīgajā valodā, daudzkategoriju klasifikāciju, un nepieciešamību pēc daļēji automatizētas pieejas. Tomēr šajā gadījumā apmācības kopa ir lielāka, netiek prasīta klasifikatora caurskatāmība, un tiek uzskatīts, ka pastāv viena ‘galvenā’ klase. Medicīnas datu heterogenitāti un sarežģītumu, sevišķi atkarību no atšķirīgām ārstu interpretācijām, kas veido medicīnisko uzdevumu unikalitāti, uzsver arī avotā [33].

Kopumā var secināt, ka klasifikācijas uzdevumos medicīnas jomā ir sastopamas visas *raksturīgās īpašības*, kas iepriekš definētas studiju priekšmetu salīdzināšanai un uzstādītas par prasībām darbā izstrādājamam mašīnapmācības risinājumam. Līdz ar to darba aprobācija, papildus galvenajai problēmsfērai, proti, izglītībai, var tikt veikta arī medicīnas jomā esošajām problēmām.

1.3. Izglītības jomas uzdevuma interpretācija mašīnapmācības kontekstā

Šī apakšnodaļa sniedz skaidrojumu promocijas darbā risinātajai augstākās izglītības studiju priekšmetu salīdzināšanas problēmai no mašīnapmācības puses. Tajā tiks paskaidrotas daļēji strukturēto studiju priekšmetu aprakstu transformēšanas metodes klasifikācijas algoritmiem izmantojama datu formāta iegūšanai. Tas nozīmē strukturētu priekšmetus aprakstošo *atribūtu* izgūšanu un *klašu* jeb kategoriju, kurās klasificējamie priekšmeti jāiedala, definēšanu. Problēmsfēru raksturojošo atribūtu izvēle un iegūšana ir būtisks aspekts. Studiju priekšmetu salīdzināšanas gadījumā atribūtu izvēle nav triviāls uzdevums un veido arī daļu no uzdevuma sarežģītības. Lai arī vairākos līdzšinējos aprakstītajos risinājumos ir piedāvāts traktēt studiju priekšmetu salīdzināšanu kā teksta klasifikācijas uzdevumu (izmantojot pilnus pieejamos priekšmetu aprakstus un nemēģinot izgūt no tiem semantiski atšķirīgas sadaļas, bet pārvēršot apraksta tekstu vārdu vektoros), jāatzīmē, ka šāds traktējums rada bažas par iegūstamo klasifikācijas rezultātu kvalitāti. Studiju priekšmetu apraksti parasti satur dažādas atšķirīgas semantiski nozīmīgas sadaļas, piemēram, „nepieciešamās priekšzināšanas" un „sasniedzamie mācību rezultāti", kuras ir būtiski nošķirt. Turklāt daļēji strukturētajam priekšmetu apraksta formātam ir bagātāka un sarežģītāka struktūra kā nestrukturētam tekstam, un to nav iespējams pilnībā izmantot, ja tiek lietotas tikai teksta klasifikācijas pieejas [22, 23]. Līdz ar to promocijas darbā plānots izmantot arī citu pieeju atribūtu definēšanai, proti, formalizētu semantiski nozīmīgu atribūtu izgūšanu no studiju priekšmetu aprakstiem.

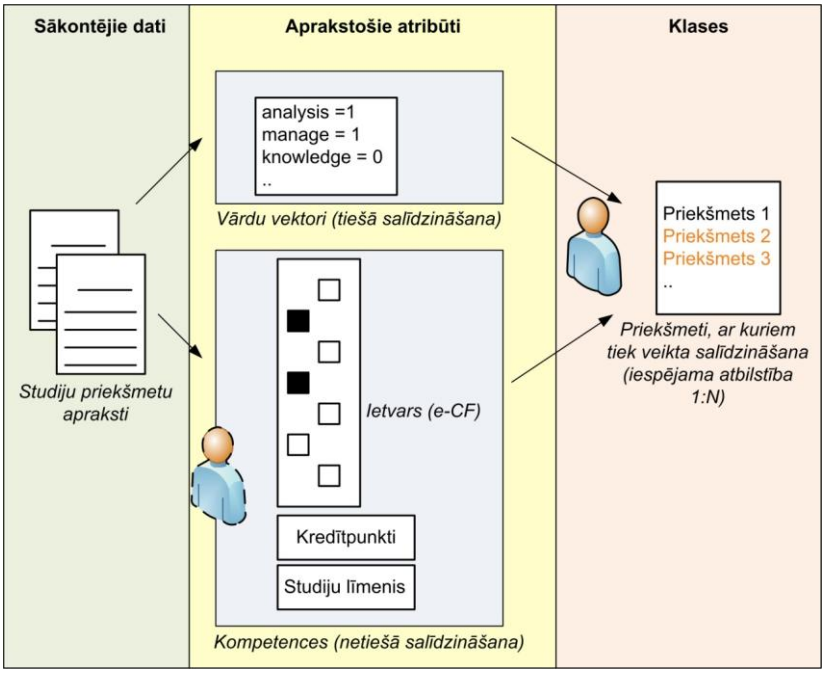
Apmācības datu iegūšana un klašu definēšana shematiski parādīta 1.8. attēlā. Sākotnējie dati ir studiju priekšmetu apraksti tādā formā, kādā izglītības institūcija tos ir publicējusi. Lai iegūtu uz atribūtu-vērtību pāriem balstītu atspoguļojumu, ko izmantot induktīvās apmācības procesā, sākotnējais apraksta veids ir jāformalizē, nosakot klases, definējot un izgūstot raksturīgos atribūtus un to vērtības. Priekšmetu apraksta formalizācija eksperimentu nolūkos tiek veikta divos neatkarīgos veidos:

- tekstu salīdzināšana neizgūstot no tiem semantiski jēgpilnas daļas (iegūstot vārdu vektorus angļu valodā);

- semantiski nozīmīgu daļu salīdzināšana (atspoguļojot sasniedzamos mācību rezultātus pret vienotu standartizētu ietvaru un izmantojot citus eksperta izvēlētus atribūtus).

Jāņem vērā, ka atribūtu izvēle šajā uzdevumā nav predefinēta un tiek veikts hipotētisks pieņēmums, ka izvēlētie atribūti spēj raksturot pētāmo konceptu – par to varēs pārliecināties praktiskajos eksperimentos.

Klases tiek izvēlētas atbilstoši klasifikācijas mērķim. Ja mērķis ir noteikt priekšmetu savstarpēju atbilstību, tad klases veido priekšmetu kopa, pret kuru notiek salīdzināšana. Klasifikatora apmācības kopas iegūšanā, iepazīstoties ar jauna priekšmeta aprakstu, piešķiramās klases šim priekšmetam nosaka eksperts, turklāt viņš var piešķirt vairākas klases vienam priekšmetam, jo neviennozīmīgās priekšmetu atbilstības dēļ viens priekšmets savā saturā var pārklāties ar vairākiem citiem.



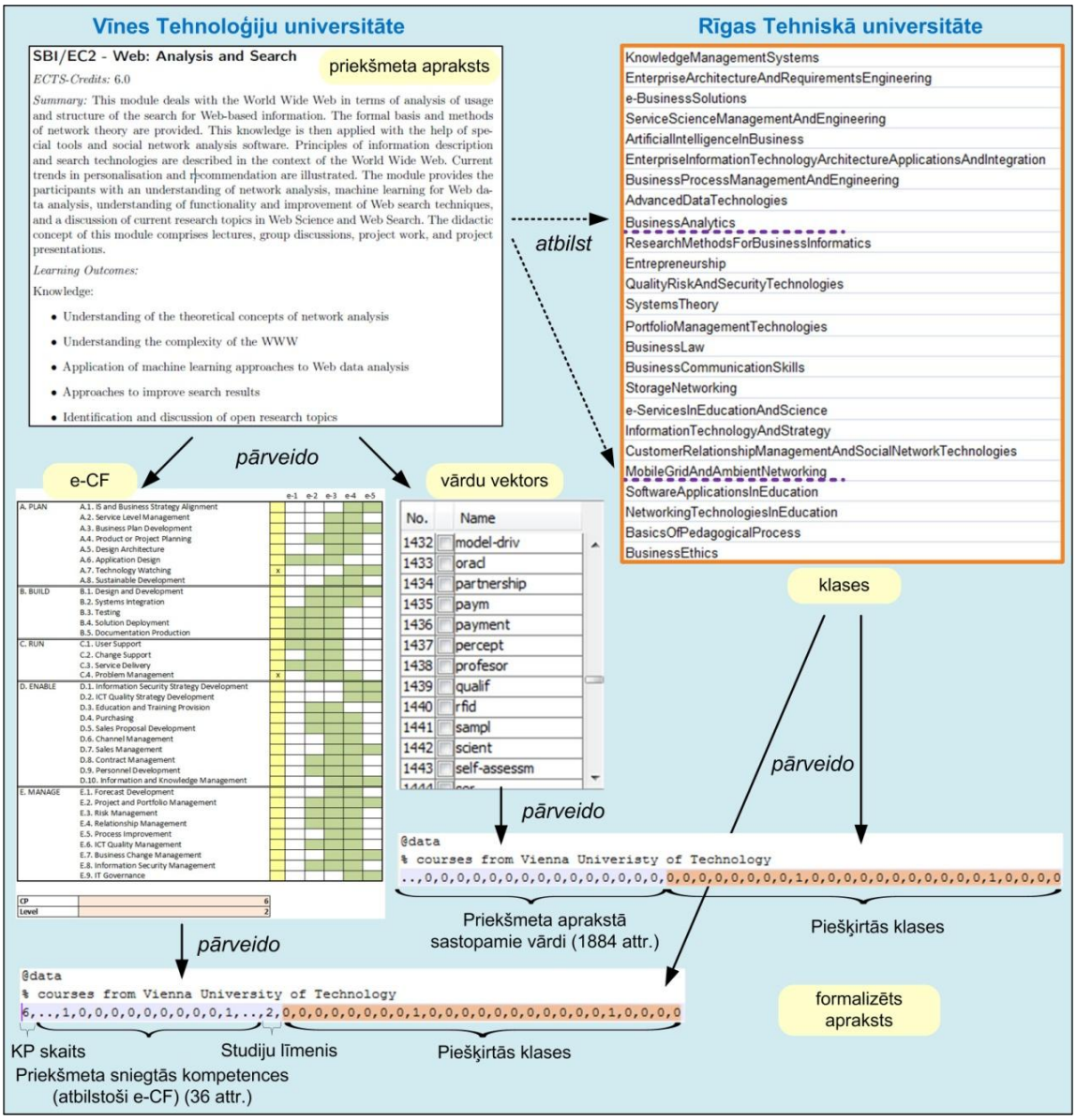
1.8. att. Priekšmetu aprakstu formalizēšanas koncepcija

Priekšmetu klasifikācija, izmantojot vārdu vektoru iegūšanu no priekšmetu aprakstiem, tiek saukta par tiešo salīdzināšanu, jo, lai iegūtu aprakstā izmantotos vārdus, tiek veikta tikai sintaktiska sākotnējo datu apstrāde. Priekšmetu apraksti, kas sākotnēji nav angļu valodā, tiek tajā pārtulkoti. Datu kopā iekļautie priekšmetu apraksti vācu valodā no Rostokas universitātes ir tulkoti angliski ar rīka *Google Translate* [34] palīdzību. Vārdu vektors satur vārdus, kas sastopami tekstos apmācības datu kopā, un konkrētam studiju priekšmetam tiek norādīts, kuri no vārdiem šī priekšmeta aprakstā ir iekļauti. Algoritms, pēc kura tiek veikta tekstu priekšapstrāde vārdu vektoru iegūšanai, kā arī praktiskās realizācijas detaļas sniegtas darba 2. pielikumā.

Netiešajā salīdzināšanā no studiju priekšmetu aprakstiem eksperts izsecina iegūstamās kompetences, atspoguļojot tās vienotā ietvarā (izvēlēts Eiropas e-kompetenču ietvars (*e-CF*) [35]), tādējādi sākotnējie dati tiek pārveidoti arī semantiski. Kā galvenie klasifikācijas atribūti tiek izmantotas konkrētas kompetences (to esamība vai neesamība) un priekšmeti tiek salīdzināti pastarpināti, par starpslāni izmantojot kompetenču ietvaru. Datu ieguves forma un aprakstošie atribūti netiešās salīdzināšanas gadījumā ir detalizēti darba 3. pielikumā. 4. pielikumā atrodams tiešās un netiešās salīdzināšanas formālo aprakstu sistemātisks iegūšanas ceļš, bet 5. pielikumā raksturota izstrādātā datu priekšapstrādes procesu atbalstošā utilītprogramma.

Balstoties uz veikto literatūras analīzi, tiek izteikts pieņēmums, ka priekšmetu tiešā salīdzināšana sniegs sliktākus klasifikācijas rezultātus kā netiešā; to ir paredzēts pārbaudīt eksperimentāli.

Konkrēts piemērs formalizējot un klasificējot Vīnes Tehnoloģiju universitātes priekšmeta „Tīmekļa analīze un meklēšana" aprakstu atspoguļots 1.9. attēlā. Šajā uzdevumā citu universitāšu studiju priekšmeti tiek salīdzināti ar RTU studiju programmas *Biznesa informātika* 25 priekšmetiem, līdz ar to iegūstot 25 klases. Vispirms eksperts ir definējis kompetences, kredītpunktu skaitu un studiju līmeni katram no šiem klases raksturojošajiem priekšmetiem. Tālāk veicot sveša priekšmeta attiecināšanu pret klasēm, eksperts nosaka, ka konkrētais Vīnes Tehnoloģiju universitātes priekšmets atbilst RTU priekšmetiem „Biznesa analītika" un „Mobilā, režģiskā un aptverošā tīklošana". Formalizējot priekšmeta aprakstā sniegto informāciju un iegūstot tālāk izmantojamos atribūtus, *e-CF* ietvarā tiek atzīmētas kompetences, kuras, pēc eksperta domām, students iegūst, apgūstot šo priekšmetu, tāpat tiek noteikts kredītpunktu skaits Eiropas kredītpunktu sistēmā (šajā piemērā – "6") un studiju līmenis ("2", kas apzīmē maģistra studijas), bet priekšmeta aprakstā izmantoto vārdu noteikšanai ar programmatūras palīdzību tiek iegūts vārdu vektors. Iegūtie rezultāti katrā no atspoguļojuma veidiem ir demonstrēti ar atbilstošu datu kopas fragmentu atribūtu un klašu vērtību aprakstīšanai *.arff* formātā. Attiecīgi, vērtība "0" norāda, ka atbilstošais atribūts vai klase šim piemēram nav piešķirta, bet "1" – ka ir piešķirta. Katrs atspoguļojuma veids tiek izmantots atsevišķa klasifikatora izveidei un tiek izmantots neatkarīgi viens no otra. Sīkāks formālā apraksta iegūšanas ceļš *.arff* formā katrā salīdzināšanas veidā sniegts darba 4. pielikumā.



1.9. att. Priekšmetu aprakstu formalizēšanas piemērs

Sīkāka analīze studiju priekšmetu salīdzināšanas problēmas risinājumam tiks veikta darba 4.4. apakšnodaļā saistībā ar klasifikācijas sistēmas projektēšanu studiju priekšmetu salīdzināšanas atbalstam. Formāli definēti klasifikācijas uzdevumā izmantotie atribūti un klases tiks sniegti praktiski veikto eksperimentu kontekstā darba 6. nodaļā.

1.4. Ar interaktīvu induktīvo apmācību risināmo problēmu loks

Apkopojot ievadā un 1. nodaļā aprakstītās problēmas un to risinājuma iespējas, izveidots to problēmsfēru raksturojošo īpašību apraksts, kuras paredzēts ņemt vērā darbā izstrādātajā risinājumā. Tiek izteikts pieņēmums, ka plānotais risinājums – interaktīva uz induktīvo apmācību balstīta klasifikācijas pieejai – var būt piemērotāks nekā tradicionālās automātiskas klasifikācijas pieejas uzdevumos, kuros raksturīgi šādi apstākļi.

Nosacījumi pieejas lietošanas iespējamībai (prasības pret problēmsfēru):

- **Cilvēks – eksperts konkrētajā problēmsfērā ir pieejams**, jo nav vērts domāt par interaktīvu sistēmu, ja kādu iemeslu dēļ nav iespējams nodrošināt eksperta pieejamību, kurš spētu palīdzēt sistēmai veikt piemēru klasifikāciju.
- **Problēmas definēšanai netiek izmantots *pārāk liels* atribūtu skaits** – tāds, kādu eksperts spēj uztvert un apstrādāt. Lielāka atribūtu skaita gadījumā iespējams definēt kādas tehnikas atribūtu grupēšanai, lai atvieglotu uztveramību lietotājam.
- **Problēmsfēras aprakstīšanai tiek izmantoti galvenokārt kvalitatīvi, nevis kvantitatīvi atribūti.** Šajā ziņā galvenais ir tas, lai eksperts spētu tos saprast.

Indikatori pieejas lietošanas nepieciešamībai (norādes, ka šis risinājums var būt piemērotāks par klasisku automātisku klasifikācijas pieeju):

- **Problēmsfērā ir svarīgi iegūt pareizu klasifikāciju** pēc iespējas vairāk klasificējamiem piemēriem. Interaktīvas induktīvās apmācības pieejā ieguldītais eksperta laiks tiek "atmaksāts" ar vairāk pareizi klasificētiem piemēriem (ja pieņemam, ka eksperts var sniegt pareizu klasifikāciju).
- **Sagaidāmā eksperta iesaistīšana nav *pārāk bieža***. Šis mērs ir subjektīvs un ir cieši saistīts ar eksperta gatavību iesaistīties piemēru klasificēšanā un sistēmas apmācībā. Ja sistēmas jautājumi apgrūtina ekspertu tik lielā mērā, ka eksperts vairs nav ar mieru šo sistēmu apkalpot, tad interaktīvas sistēma lietošana nenesīs vēlamo efektu.
- **Problēmsfērā ir grūti izgūt vai definēt raksturīgās iezīmes**, tādēļ pastāv aizdomas, ka atribūti problēmsfēru nedefinē pilnīgi. Iespējams, ka problēmsfēru aprakstošie dati ir heterogēni un mainīgi.
- **Ir pieejama tikai neliela sākotnējā apmācības kopa**, un pastāv aizdomas, ka tā nav reprezentabla. Tas nozīmē, ka klasifikatoram ir lielāka iespējamība sastapties ar piemēriem, ko tas nespēs klasificēt, jo pētāmais koncepts nav vispusīgi aprakstīts.

1.5. Nodaļas kopsavilkums

Šajā nodaļā tika pamatota promocijas darbā risināmā problēma, parādītas konkrētas cilvēku darbības jomas, kurās šī problēma pastāv, un apskatīts risinājumu apgabals, kurš tālākajā darbā jāattīsta. Kā galvenā joma, kurā neder esošie automātiskās klasifikācijas risinājumi, analizēta izglītība. Augstākās izglītības dokumentu salīdzināšanas nepieciešamība parādās vairākās formās un ir nosacīti iedalāma trīs kategorijās.

- Studiju priekšmetu salīdzināšana apmaiņas programmām, tālākizglītībai, izglītības dokumentu pielīdzināšanai u.c. vajadzībām. Galvenokārt tiek izmantoti dokumenti, kas apraksta studiju programmu un priekšmetu saturu un apliecina iegūto izglītību.
- Studiju programmu izstrāde, kas iesaista salīdzinājuma veikšanu ar citām līdzīgām izglītības programmām. Galvenokārt tiek izmantoti dokumenti, kas apraksta studiju programmu un priekšmetu saturu.
- Mācību materiālu dalīšana kategorijās, piemēram, e-apmācības sistēmās, lai interesentam tiktu piedāvāti atbilstoši mācību materiāli.

Apkopotajos informācijas avotos ir sekojošas norādes par apstākļiem studiju satura salīdzināšanas procesā:

- studiju priekšmetu apraksti ir ļoti dažādi, tādēļ ar automātiskiem līdzekļiem nevar sasniegt perfektus rezultātus;
- daļēji strukturētais dokumentu formāts prasa specifisku pieeju un dažādu informācijas izguves metožu lietošanu;
- vajadzība pēc eksperta iesaistes parādās dažādās procesa fāzēs;
- procesa uzlabošanas galvenais ierobežojums ir darba apjoms, ko eksperts var ieguldīt.

Nepieciešamība atbalstīt dažādu izglītības dokumentu salīdzināšanas procesu ar datoru palīdzību ir pamatota literatūrā un izteikta no augstākās izglītības jomā strādājošo puses. Līdzšinējo darbu analīze norāda nepieciešamos uzlabojumu virzienus un pastāvošos ierobežojumus, lai akadēmisko dokumentu salīdzināšanā varētu ieviest zināmu automatizāciju. Izdarot secinājumus par šīs sfēras problemātiku, promocijas darbā tiks izskatīts uzdevums par **uz induktīvo apmācību balstītu**, **interaktīvu** un **daudzkategoriju** klasifikācijas sistēmas izstrādi studiju priekšmetu salīdzināšanas atbalstam.

Formalizējot studiju priekšmetu salīdzināšanas uzdevumu kā prasības pret mašīnapmācības risinājumu, ir definētas šādas ***raksturīgās īpašības***:

- iegūtie rezultāti ir jāsaprot klasifikatora lietotājam un ekspertam;
- pieejamā apmācības kopa ir maza;
- sākotnējie dati ir daļēji strukturēti vai nestrukturēti;
- problēmsfērā ir raksturīgas daudzas klases, kuras sastopamas vienlīdz bieži;
- objekts var piederēt vienlaicīgi vairākām klasēm.

Izpētot problēmas medicīnas jomā, ir secināts, ka līdzīgas sfēras īpašības ir sastopamas arī šeit, līdz ar to darbā risināmais uzdevums nav uzskatāms tikai par vienas konkrētas problēmsfēras vajadzību. Ir definēti arī papildus nosacījumi sfērām, kurās plānotais risinājums var būt piemērotāks nekā tradicionālās klasifikācijas pieejas.

- Nosacījumi pieejas lietošanas iespējamībai (prasības pret problēmsfēru):
    - cilvēks – eksperts konkrētajā problēmsfērā ir pieejams;
    - problēmas definēšanai netiek izmantots *pārāk liels* atribūtu skaits;
    - problēmsfēras aprakstīšanai tiek izmantoti galvenokārt kvalitatīvi, nevis kvantitatīvi atribūti.
- Indikatori pieejas lietošanas nepieciešamībai (norādes, ka šis risinājums ir atbilstošāks par klasisku klasifikācijas pieeju):
    - problēmsfērā ir svarīgi iegūt pareizu klasifikāciju;
    - sagaidāmā eksperta iesaistīšana nav *pārāk bieža*;
    - problēmsfērā ir grūti izgūt vai definēt raksturīgos atribūtus;
    - ir pieejama tikai neliela sākotnējā apmācības kopa.

Sākot interaktīvas klasifikācija sistēmas izveidi, jāapskata līdzšinējie risinājumi automātiskā klasifikācijā, sadarbībā ar sistēmas lietotāju un klasifikācijas sistēmu arhitektūru izstrādē, lai apzinātu problēmas, izmantotu jau esošus un pārbaudītus risinājumus, ja tādi ir, un pārņemtu labo praksi. Tam tiks veltīta darba 2. nodaļa.

2. SAISTĪTO DARBU ANALĪZE: IESTRĀDNES UN PASTĀVOŠĀS PROBLĒMAS

Šī nodaļa apkopo līdzšinējos pētījumus un teorētisko bāzi dažādos klasifikācijas aspektos. 2.1. apakšnodaļā tiks apskatīts automātiskās klasifikācijas uzdevums, sevišķu uzmanību pievēršot induktīvās apmācības pieejai, daudzkategoriju klasifikācijai un tās novērtēšanas mēriem, kā arī problēmām jaunu piemēru klasificēšanā. 2.2. apakšnodaļa ir veltīta dažāda veida līdzšinējiem interaktīviem risinājumiem induktīvajā apmācībā un citās klasifikācijas pieejās. 2.3. apakšnodaļa sniegs klasifikācijas sistēmu arhitektūru apkopojumu, lai pamatotu interaktīvas klasifikācijas sistēmas izstrādes stūrakmeņus.

2.1. Klasifikācijas uzdevums mašīnapmācībā

Mašīnapmācība var tikt definēta kā datorprogrammu spēja uzlabot savu darbību, balstoties uz savu pagātnes pieredzi, kā arī spēja definēt jaunas, no iepriekšējām atšķirīgas, datu struktūras [1]. Mašīnapmācības algoritmus var iedalīt pārraudzītajos un nepārraudzītajos [1, 36]. Pie pārraudzītās apmācības pieder klasifikācija, pie nepārraudzītās apmācības pieder klasterizācija. Klasifikācija ir objekta piederības noteikšana kādai no iepriekš definētām grupām jeb klasēm. Klasterizācija ir datu objektu apvienošana pa grupām jeb klasteriem, balstoties uz šo objektu līdzībām, kuras nosaka pēc objektu atribūtu vērtībām. Klasterizācijas gadījumā grupas iepriekš nav zināmas. Šī darba ietvaros turpmāk tiks apskatīts klasifikācijas uzdevums, jo iepriekš definētā darbā risināmā problēmas nostādne paredz konkrētu klašu piederības noteikšanu interesējošajiem objektiem.

Lai veiktu klasifikāciju, ir nepieciešams realizēt kāda veida spriešanu. Viens no spriešanas veidiem ir indukcija. Indukcija ir process, kurā konkrēti fakti tiek pārvērsti vispārīgās likumsakarībās. Svešvārdu vārdnīca [37] indukciju definē šādi: „Slēdziena veids un metode, kurā no vairākiem atsevišķiem gadījumiem tiek izsecināts vispārējais (pretstats dedukcijai)”. Šī darba kontekstā galvenā uzmanība tiek pievērsta induktīvās spriešanas izmantošanai datu automātiskā vai daļēji automātiskā apstrādē, tādēļ darba ietvaros tiek lietots termins „induktīvā apmācība”. Indukcija ir viena no vecākajām un svarīgākajām apmācības problēmām datorzinātnē [38]. Induktīvā apmācība datorzinātnē nozīmē mācīšanos no piemēriem, kad sistēma cenšas inducēt vispārīgu likumu no doto piemēru kopas [39]. Ar piemēru saprot atsevišķu, veselu datu vienību, eksemplāru, kurš apraksta kādu reālās pasaules objektu. Induktīvā apmācība ietver klasifikāciju – atsevišķu objektu piederības noteikšanu kādai klasei.

Klasifikācijas uzdevuma formāla definīcija:

$K = \{k_1, ..., k_j\}$: klašu kopa, j: klašu skaits

$X = \{x_1, ..., x_i\}$: datu kopa, i: piemēru skaits

$x_i = \{(a_1, v_{a1}), ..., (a_n, v_{an})\}$: datu kopas objekts (piemērs), atribūtu-vērtību pāru vektors,

n – atribūtu skaits, v_a – atribūta a vērtība

Klasifikācijas uzdevums ir izveidot atspoguļojumu $c: X \rightarrow K$ jeb atrast *koncepta aprakstu*.

$c = h(X, K)$: koncepta apraksts (klasifikators)

$K_c = f(X, c)$: klasifikatora prognozētais klašu vektors balstoties uz koncepta aprakstu c

Tradicionālajā vienas kategorijas klasifikācijā katrs piemērs l ir saistīts ar **1 klasi** k no nepārklājošos klašu kopas K: $l = (x, k)$

$l \in L$: apmācības piemērs no apmācības piemēru kopas

$L = \{(x_1, k_1), ..., (x_i, k_b)\}$: apmācības piemēru kopa

Daudzkategoriju klasifikācijā katrs piemērs l ir saistīts ar **klašu kopas apakškopu** Y, $Y \subseteq K$: $l = (x, Y)$

$L = \{(x_1, Y_1), ..., (x_i, Y_d)\}$: apmācības piemēru kopa

Šīs apakšnodaļas turpinājumā tiks analizēti aspekti klasifikācijas metodes izvēlē (2.1.1. sadaļā), apskatītas induktīvās apmācības pamatnostādnes un process (2.1.2. sadaļā), pārejot pie daudzkategoriju klasifikācijas iespēju skaidrojuma (2.1.3. sadaļā) un klasifikācijas rezultātu novērtēšanas (2.1.4. sadaļā). 2.1.5. sadaļa sniedz promocijas darba uzdevumos ietilpstošo klasifikācijas un induktīvās apmācības problēmu analīzi.

2.1.1. Klasifikācijas metodes izvēle

Klasifikācijā var izmantot dažādas mašīnapmācības tehnikas, piemēram, lēmumu kokus un likumus ģenerējošos algoritmus, neironu tīklus, ģenētiskos algoritmus, Beijesa teorēmā balstītus algoritmus u.c. Turklāt citās sfērās sastopami citi nosaukumi pašam klasifikācijas uzdevumam - statistikā klasifikāciju sauc par diskriminantu analīzi, inženierijā, par tēlu pazīšanu. Literatūrā ir sastopami daudzi pārskati gan par metožu grupām, gan konkrētiem algoritmiem, piemēram, [2, 4, 40] kā arī šī darba 6. pielikumā.

Saskaņā ar ikgadējo datizraces lietošanas pārskatu, ko veic „*Rexer Analytics*", visbiežāk izmantotā metode 2010. gadā ir bijusi lēmumu koki [41]. To izmantojuši 69 % respondentu no aptaujātajiem 735 datizraces speciālistiem 60 valstīs [41]. No lēmumu kokiem populārākais algoritms ir bijis *C4.5* – Kvinlana (*Quinlan*) klasiskā *ID3* algoritma uzlabota versija. Lēmumu

koki, tāpat kā likumus ģenerējošie algoritmi pieder induktīvās spriešanas algoritmiem, tādēļ tos ir lietderīgi apskatīt vienotā kontekstā.

Jāņem vērā, ka neviena klasifikācijas tehnika vai algoritms nav viennozīmīgi labāks par kādu citu [42]. To, kādu tehniku labāk izmantot, nosaka problēmas tips, iepriekšējās zināšanas un citi faktori. Katras konkrētas problēmas risināšanā klasifikatoru veikums atšķiras, un ir iespējams izdarīt tikai pieņēmumus par klasifikatora piemērotību pirms tā reālās izmantošanas. Nav konteksta un pielietojuma neatkarīgu iemeslu, kas ļautu apgalvot, ka kāda metode ir labāka par citām, lai sasniegtu labu vispārināšanu. Turklāt katrā klasifikācijas situācijā ir citi aspekti, kuri ir jāņem vērā – apmācības datu pieejamība, izmaksas, iepriekšējo zināšanu izmantošana u.c. Dažas metodes spēj veikt teicamu apmācību pie liela datu daudzuma, savukārt ir vājas, ja pieejama tikai neliela apmācības piemēru kopa. Nepietiek tikai ar teorētiskām zināšanām par algoritma darbību; klasifikācija ir empīriska disciplīna, kur piemērotākos klasifikācijas algoritmus konkrētā sfērā var noskaidrot, tikai veicot praktiskus mēģinājumus.

Izvēle par labu kādai no klasifikācijas metodēm ir atkarīga no prasībām, konkrētās problēmsfēras, datu apjoma un citiem lielumiem. Uzsverot induktīvās apmācības priekšrocības, jāmin, ka tās iegūtie klasifikācijas rezultāti parāda spriešanas ceļu, kas ir neatsverami sistēmās, kam jāapstrādā klasifikācijas rezultāti, jāpārbauda tie un jāizmanto turpmāk [43]. Tādas, piemēram, ir ekspertu sistēmas, kuru zināšanu bāzēs var izmantot induktīvās apmācības iegūtos likumus. Turklāt, ja darbības joma ir atbildīga, lietotājam ir nepieciešams izsekot spriešanas gaitai klasifikatorā, lai uzticētos tā pieņemtajam lēmumam [44]. Nav svarīgi, ka kāda metode ir ļoti ātrdarbīga un veiksmīga – ja tā nespēj paskaidrot savus iegūtos lēmumus, daudzu problēmu risināšanai tā vienkārši nav piemērota.

Būtisks parametrs klasifikācijas metodes izvēlē ir problēmsfēras objektiem piešķiramo klašu jeb kategoriju skaits. Klasifikācijas uzdevumos parasti ir nepieciešams noteikt objekta **piederību tikai vienai klasei** (ang. v. - *single-label*). Tas ir, katrs piemērs ir saistīts ar vienu klasi k no nepārklājošos klašu kopas K, $|K| > 1$. Piemēram, elektroniskā pasta vēstule ir klasificējama kā lietderīgs pasts vai surogātpasts. Tomēr ir arī sfēras, kurās **objekti var piederēt vienlaicīgi vairākām klasēm**, piemēram, žurnāla raksts var attiekties uz vairākām tēmām vai viens studiju priekšmets var atbilst vairākiem citiem priekšmetiem. Šajā gadījumā runa ir par daudzkategoriju (ang. v. – *multi-label*) klasifikāciju, un katrs piemērs ir saistīts ar apakškopu $Y \subseteq K$. Latviešu valodā darba autore izvēlējusies tulkojums „daudzkategoriju klasifikācija" tādēļ, ka it kā loģiski tuvākais jēdziens „daudzklašu klasifikācija" angļu valodā sasaucas ar „*multiclass clasification*", kas nozīmē ko citu. Vienkategorijas klasifikācijā tas nozīmē klasifikācijas

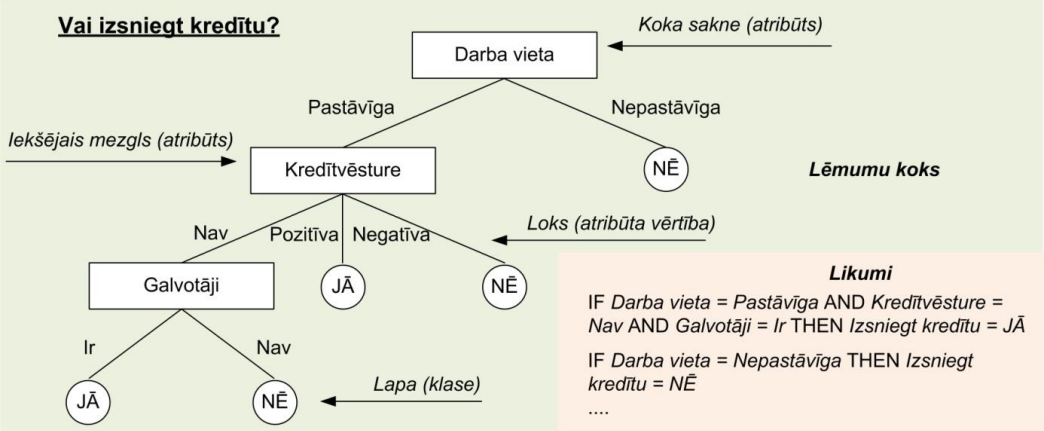
uzdevumu, kur jāizvēlas starp vairāk nekā divām piešķiramajām klasēm, pretstatā binārajai klasifikācijai, kurā ir tieši divas klases.

Daudzkategoriju klasifikācija ir aktuāla tādās sfērās kā bioinformātika, teksta kategorizēšana un medicīnas diagnostika [45]. Kategorizēšana ir sinonīms vārdam klasifikācija, kurš parasti tiek lietots tieši tekstu un dokumentu apstrādes kontekstā. Tas, vai piederība vairākām klasēm vienlaicīgi ir iespējama vai nevēlama parādība, ir atkarīgs no risināmās problēmas dabas. Ja nav speciāli norādīts, ka uzdevumā iespējama vairāku klašu piešķiršana katram piemēram, tad klasifikāciju pēc noklusējuma uzskata par uzdevumu, kurā klases ir savstarpēji izslēdzošas. Sīkāks apraksts par daudzkategoriju klasifikāciju tiks sniegts 2.1.3. sadaļā.

2.1.2. Induktīvās apmācības pamatnostādnes

Literatūrā induktīvajai apmācībai pieskaitāmo algoritmu loks variējas. Dažādiem autoriem ir atšķirīgi uzskati par to, kur robežojas induktīvā apmācība. Netiek apšaubīts, ka induktīvajiem algoritmiem pieder lēmumu koki un likumus ģenerējošie algoritmi. Tomēr daļa literatūras avotu, piemēram, [46-48], par induktīvām uzskata arī tādas metodes kā *k* – tuvāko kaimiņu klasifikators, Beijesa klasifikators, atbalsta vektoru mašīnas, ģenētiskie algoritmi un dažreiz arī mākslīgo neironu tīklus. Principiālo atšķirību starp šiem diviem zskatiem nosaka tas, vai par induktīvām apmācības metodēm tiek atzītas tādas metodes, kuras tikai spēj klasificēt jaunus datus uz iepriekš doto piemēru pamata, vai tādas, kuras izsaka hipotēzi par datu struktūru un sniedz vispārinošu modeli, ar kura palīdzību klasificēt jaunos datus. Otrajā gadījumā metožu loks tiek ierobežots, jo ‘vienkāršu’ klasifikāciju spēj veikt plašāks pieeju klāsts. Kā pamato Bhavsars [49], vispārināšanu induktīvās apmācības kontekstā nevar nošķirt no mācīšanās. Sistēma, kas nespēj vispārināt, nav uzskatāma par spējīgu apmācīties [49]. Vienkāršākos vārdos induktīvā apmācība tās šaurākajā nozīmē tiek raksturota šādi: novērojumu un iepriekšējo zināšanu apgūšana, vispārinot likumus un zināšanas [50]. Vispārināšana var tikt uzskatīta par metodi, kas veido jaunas, vienkāršāk atspoguļotas zināšanas no jau esošajām, joprojām nezaudējot iepriekšējās zināšanas. Induktīvā apmācība dod iespēju identificēt regularitātes un tēlus iepriekšējās zināšanās vai apmācības datos un izgūt tos kā vispārinātus likumus. Tādēļ, piemēram, *k* – tuvāko kaimiņu klasifikators nevar tikt pieskaitīts induktīvajai apmācībai, jo tas neveic nekādu zināšanu kompresiju, bet visu laiku strādā ar visiem apmācības kopas piemēriem. Promocijas darba autore arī pieturēsies pie principa, ka pie induktīvās apmācības algoritmiem pieskaitāmi lēmumu kokus veidojošie un likumus ģenerējošie algoritmi, kā pieņem arī liela daļa apskatītās literatūras autoru [49, 51-56].

Induktīvās apmācības algoritmi ir salīdzinoši vienkārši un uzticami, turklāt ļauj veidot saprotamus, caurskatāmus klasifikācijas modeļus, kas nav iespējams ar visām klasifikācijas metodēm. Tos var attēlot IF – THEN likumu vai lēmumu koku veidā. 2.1. attēlā sniegts lēmumu koka un dažu tam atbilstošu likumu piemērs kredīta izsniegšanas uzdevumā.



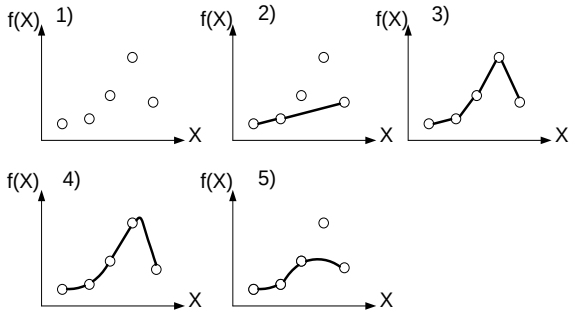
2.1. att. Lēmumu koka un likumu piemērs

Likumu indukcija no empīriskiem datiem ir efektīva tehnika automātiskai zināšanu iegūšanai [43]. Daudzās lietojumsfērās klasifikācijas koki sniedz līdzīgu klasifikācijas precizitāti kā neironu tīkli un tuvāko kaimiņu klasifikators, sevišķi gadījumos, kad trūkst zināšanu par labāko iespējamo klasifikatora formu [42]. Lēmumu kokus veidojošie klasifikatori ir īpaši lietderīgi nominālu datu gadījumā. Lēmumu koku metodes balstās uz pārbaužu sēriju, kas rezultātā nosaka klasifikāciju. Piemēru klasificēšana ar lēmumu koku var būt samērā vienkārša, turklāt, pateicoties to strukturālajai vienkāršībai, kokus ir viegli interpretēt. Var tikt izmantots arī skaitlisku un nominālu atribūtu salikums. Lēmumu koki ir liela nelineāro klasifikatoru daļa [57]. Problēmsfērās ar daudziem atribūtiem induktīvās metodes praktiskos ierobežojumus apiet, sākotnējo problēmu sadalot vairākās apakšproblēmās un lēmumu procesu vienkāršojot [58].

Induktīvā apmācība sniedz šādas priekšrocības:

- nelineāras klasifikācijas iespējas [57];
- caurskatāmu rezultāta iegūšanas aprakstu, kas ir salīdzināms ar eksperta spriešanas procesu [43];
- jauna objekta klasifikācijai var būt nepieciešamas tikai dažas pārbaudes [57];
- induktīvās apmācības algoritmi ir neatkarīgi no apskatāmās darbības sfēras [43];
- nav nepieciešams izdarīt kādus pieņēmumus par modelējamās sistēmas struktūru [43];
- iegūtie likumi ir labi lietojami ekspertu sistēmās [43].

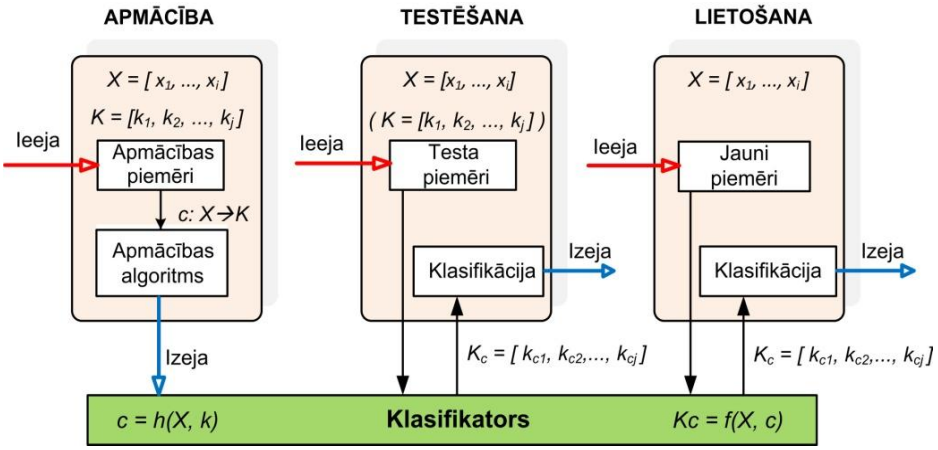
2.2. attēlā parādīts induktīvās apmācības uzdevums kā hipotētiskās funkcijas atrašana piemēru kopā (ilustrācijai izmantots avots [59]).



2.2. att. Funkcijas meklēšana piemēru kopā [59]

Matemātiskā formā koncepta apraksta iegūšanu var traktēt kā funkcijas atrašanu piemēru kopā. Katrs pāris ***(x, f(x))*** veido piemēru, kur ***x*** ir ieeja, bet ***f(x)*** izeja. Tā kā īstā funkcija ***f***, ko apraksta doto datu kopa, nav zināma, tad indukcijas uzdevums ir iegūt hipotēzi ***h***, kas būtu tuvināta nezināmajai īstajai funkcijai. 2.2. attēla 2. līdz 5. grafikā uzskatāmi redzams, ka dažādi algoritmi var atrast dažādas hipotēzes vienā un tajā pašā piemēru kopā.

Tradicionālā induktīvās apmācības procesā var izšķirt vairākus posmus, kas grafiski attēloti 2.3. attēlā. Apmācības jeb klasifikatora veidošanas laikā apmācības piemēri tiek izmantoti, lai ar konkrēta apmācības algoritma palīdzību inducētu klasifikatoru. Testēšanas laikā klasifikatora pārbaudei tiek izmantoti testa piemēri, kuru klases piederība ir zināma, bet kuri iepriekš nav izmantoti apmācībai, līdz ar to var izdarīt secinājumus par klasifikatora precizitāti un citiem veiktspējas parametriem. Ja šie parametri uzrāda apmierinošus rezultātus, klasifikatoru var sākt lietot jaunu un iepriekš neredzētu piemēru klases piederības noteikšanai.



2.3. att. Induktīvās apmācības posmi

Klasifikatora izveidošanai var izmantot dažādus apmācības algoritmus, par kuru veidiem un konkrētu algoritmu piemēriem sīkāk ir stāstīts darba 6. pielikumā, kur darba autore ir veikusi induktīvās apmācības algoritmu iedalījumu pēc dažādām pazīmēm – pēc realizācijas un attēlošanas formas, pēc apmācības veida un pēc šķeļošo plakņu novietojuma. Sagatavotie

apkopojumi pirmo reizi ir publicēti autores bakalaura [60] un maģistra darbā [61], kur arī atrodami vairāku algoritmu detalizēti apraksti ar izskaidrojošiem piemēriem. Promocijas darbā ir sīkāk aprakstīts likumus ģenerējošais algoritms *Ripper*, jo tas ir (1) samērā jauns un maz izskaidrots un (2) praktiskajos eksperimentos pierādījis sevi gan no veiktspējas, gan likumu saprotamības puses (ar eksperimentālajiem rezultātiem var iepazīties darba 6. nodaļā).

Induktīvās apmācības algoritms Ripper

Ripper ir likumus veidojošs induktīvās apmācības algoritms, kura autors ir Koens (ang. v. – *Cohen*) [62]. Katrs likums ir nosacījumu konjunkcija, kur atsevišķu nosacījumu apraksta formā $a = v$ (ja a ir nomināls atribūts) vai formā $a \leq v$ vai $a \geq v$ (ja a ir skaitlisks atribūts), ar v norādot atribūta vērtību. Koena *Ripper* algoritms balstās uz iepriekšējiem darbiem lēmumu koku un likumu indukcijas rezultātu uzlabošanai - atzarošanu kļūdas samazināšanai (ang. v. - *reduced error pruning – REP*) un inkrementālo atzarošanu kļūdas samazināšanai *IREP*. Šīs pieejas pamatojas uz principu, ka koks vai likums tiek saīsināts, ja tādējādi samazinās klasifikācijas kļūda iepriekš neredzētiem piemēriem.

Apmācības kopa tiek sadalīta likumu veidošanas (audzēšanas) un atzarošanas (ang.v. – *pruning*) kopās. Likums tiek veidots, pievienojot jaunus nosacījumus, un pēc tam atzarots (saīsināts). Likumu veidošana turpinās, līdz ir sasniegts kāds no apstāšanās kritērijiem – kļūdas līmenis (ang. v. – *error rate*) vai apraksta garums (ang. v. – *description length*). Tam seko likumu optimizācija.

Ripper algoritms ir samērā sarežģīts un satur dažādas variācijas iespējas lietotajās metrikās. Šeit dotas algoritma galvenās sastāvdaļas:

1. Inicializēt likumu sarakstu $LS = \{ \}$ un katrai no klasēm, sākot ar mazāk pārstāvēto līdz plašāk pārstāvētajai, izpildīt soļus 2- 5. Iedalīt piemērus likumu veidošanas un atzarošanas kopās pozitīvajos (klasei piederošajos) un negatīvajos (klasei nepiederošajos) piemēros.
2. Atkārtot 3. – 4. soli līdz sasniegts kāds no apstāšanās kritērijiem:
    a. Apraksta garums *AG* ir par 64 bitiem (*IREP** versijā [63]) garāks par īsāko līdz šim atrasto apraksta garumu. *AG* ir skaitliska vērtība, kas raksturo likuma sarežģītību, ņemot vērā likuma garumu un piemērus, ko likums pārklāj gan pozitīvo, gan negatīvo piemēru kopā;
    b. Nav pozitīvu piemēru;
    c. Kļūdas līmenis pārsniedz 50%.

3. Likumu veidošana – balstoties uz likumu audzēšanas piemēru kopu, audzēt likumu, pievienojot tajā nosacījumus, līdz sasniegta 100% precizitāte. Šajā solī tiek pārbaudīta katra atribūta visas iespējamās vērtības un izvēlēts nosacījums ar lielāko informācijas ieguvumu (ang. v. – *information gain*), kuru aprēķina šādi :

$$Ieguvums\ (R\text{^}, R) = p \times \left(\log\frac{p}{t} - \log\frac{P}{T}\right),$$

kur R – oriģinālais likums,

R^ – kandidāta likums pēc nosacījuma pievienošanas,

p – piemēru skaits pozitīvo piemēru kopā, ko pārklāj likums R^,

t – piemēru skaits, kurus apskata R^,

P – piemēru skaits pozitīvo piemēru kopā, ko pārklāj likums R,

T – piemēru skaits, kurus apskata R.

4. Likumu atzarošana – balstoties uz atzarošanas piemēru kopu, secīgi atzarot katru likumu, izmantojot metriku (*IREP** versijā [63]):

$$\frac{p-n}{p+n},$$

kur p – piemēru skaits pozitīvo piemēru kopā, ko pārklāj likums,

n – piemēru skaits negatīvo piemēru kopā, ko pārklāj likums

Pievienot iegūto likumu R_i sarakstam *LS* un izņemt piemērus, ko pārklāj likums, no pozitīvo un negatīvo piemēru kopas.

5. Optimizācijas posms - katram likumam $R_i \in LS$ ģenerēt un atzarot divus likuma variantus, izmantojot 3. un 4. soli. Likuma variants R_i' tiek veidots sākot ar tukšu nosacījumu konjunkciju, bet R_i'' – pievienojot nosacījumus oriģinālajam R_i likumam. Par gala likumu tiek izvēlēts variants R_i, R_i' vai R_i'' ar mazāko apraksta garumu *AG*. Ja pēc visu *LS* likumu pārbaudes ir palikuši nepārklāti piemēri pozitīvo piemēru kopā, tiek ģenerēti jauni likumi saskaņā ar 3. un 4. soli.

Ripper priekšrocības ir viegli interpretējamā likumu kopa un labie rezultāti nesabalansētu datu gadījumā, kad klases raksturojošie apmācības piemēri nav līdzīgās proporcijās [31]. Īpaši šī algoritma priekšrocības esot jūtamas lielām un trokšņainām datu kopām [63].

Programmatūrā *Weka* algoritms *Ripper* realizēts ar nelielām izmaiņām un atrodams ar nosaukumu *JRip* (turpmāk darbā arī tiks izmantots šis nosaukums).

Sīkāka informācija par *Ripper*, kā arī *REP* un *IREP* algoritmiem atrodama [62, 63]. *Ripper* izskaidrots piemērs latviešu valodā atrodams bakalaura darbā [64].

2.1.3. Klasifikācija daudzkategoriju gadījumā

Daudzkategoriju gadījumā, atšķirībā no tradicionālās vienkategoriju klasifikācijas, objekti var piederēt vienlaicīgi vairākām klasēm - katrs objekts ir saistīts ar apakškopu $Y \subseteq K$, kur K ir visa klašu kopa dotajā problēmsfērā.

Lai parādītu atšķirības starp klasifikāciju vienkategorijas un daudzkategoriju gadījumā, 2.1. tabulā sniegts salīdzinājums iespējamajam klasifikācijas rezultātam vienam un tam pašam klasificējamajam objektam dažādiem klasifikācijas veidiem. Termins „piešķirtās klases" nozīmē klasifikatora prognozētās klases dotajam objektam, turpretī „īstās klases" apzīmē objektam patiešām zināmās piederīgās klases (parasti – apmācības un testēšanas laikā).

Nedaudz sīkāk aprakstot apmācības uzdevumu daudzkategoriju datiem, jādefinē divi virzieni – **daudzkategoriju klasifikācija** un **kategoriju ranžēšana**. Pēdējā laikā parādās arī šo virzienu apvienojums jeb daudzkategoriju ranžēšana. Daudzkategoriju klasifikācijas rezultāts jaunam piemēram ir divas klašu kopas – piederīgās un nepiederīgās klases [65]. Kategoriju ranžēšanas gadījumā tiek iegūts klašu saraksts tādā secībā, ka piemērs ir vairāk piederīgs pirmajām klasēm un aizvien mazāk saistīts ar tālākajām.

2.1. tabula

Klasifikācijas piemērs vienas kategorijas un daudzkategoriju klasifikācijas gadījumā

Pieeja	**Objekts (aprakstošie atribūti)**	**Piešķirtās klases**				**Secinājums**
		A	**B**	**C**	**D**	
Vienkategorijas klasifikācija	a1 = 1, a2 = 1, a3 = 0	**1**	**0**	**0**	**0**	Objekts pieder klasei A
Daudzkategoriju klasifikācija		**1**	**0**	**1**	**0**	Objekts pieder klašu kopai $\{A, C\}$, nepieder klašu kopai $\{B, D\}$
Kategoriju ranžēšana		**1**	**3**	**2**	**4**	Ranžēts klašu saraksts: A, C, B, D

Daudzkategoriju datu apstrādi var veikt divos veidos:

- problēmas transformācija,
- algoritmu adaptācija.

Problēmu transformācijas metodes ir neatkarīgas no tālāk izmantotajiem algoritmiem. Šajā gadījumā apmācības uzdevums tiek pārvērsts vienā vai vairākos bināros vienkategorijas klasifikācijas uzdevumos, kuriem ir plašs izmantojamo algoritmu klāsts (visi darba 6. pielikumā minētie klasifikācijas algoritmi ir paredzēti vienkategorijas klasifikācijai). Problēmu

transformācijas metodes var uzskatīt (1) katru objektam piešķirto klašu kombināciju par jaunu klasi, veidojot klašu kopas (ang. v. - *label powerset*), tādējādi uzreiz iegūstot vienkategorijas klasifikācijas uzdevumu, vai (2) izveidot q binārus klasifikatorus, pa vienam katrai no q klasēm datu kopā ar binārās saistības (ang. v. - *binary relevance*) metodi, un gala klasifikāciju jaunam piemēram piešķirt, apvienojot visu vienkategorijas klasifikatoru lēmumu. Klašu kombināciju izmantošana, lai veidotu jaunu klasi, ir lietderīga tad, ja kombinācijām ir semantiska jēga un apmācības piemēru skaits to atļauj (katra klašu kombinācija ir raksturota ar vairākiem piemēriem).

Sekojošajā piemērā demonstrēts, kā daudzkategoriju datu kopu pārveido abas pieminētās problēmu transformācijas metodes. 2.2. tabulā atspoguļota daudzkategoriju datu kopa ar trīs objektiem (piemēriem) un četrām iespējamajām klasēm.

2.2. tabula

Sākotnējā daudzkategoriju datu kopa

	Klases			
Objekts	**A**	**B**	**C**	**D**
1	**1**	0	0	**1**
2	0	**1**	**1**	0
3	**1**	0	0	0

2.3. tabulā parādīts, kā klašu kopas veidojošā metode ir pārveidojusi datu kopu, radot tik klašu, cik dažādu klašu kombināciju ir sākotnējos datos, un iegūstot vienu vienkategorijas klasifikācijas uzdevumu ar jaunajām klasēm.

2.3. tabula

Klašu kopas veidojošās metodes izveidotā datu transformācija

Objekts	**A**	**A∧D**	**B∧C**
1	0	**1**	0
2	0	0	**1**
3	**1**	0	0

Savukārt binārās saistības metode ir ieguvusi četrus vienkategorijas uzdevumus (skat. 2.4. attēlu a – d) binārai klasifikācijai – katrs atsevišķais klasifikators noteiks, vai objekts pieder vai nepieder konkrētajai klasei.

Abu transformācijas veidu gadījumā vienkategorijas klasifikatoru iegūtais rezultāts jāpārvērš atpakaļ daudzkategoriju formā. Klašu kopu gadījumā klases ir jāsadala atpakaļ

atbilstoši oriģinālajam datu formātam, bet binārās saistības metodei atsevišķo klasifikatoru rezultāti jāapvieno, lai uzzinātu visas objektam noteiktās klases.

Objekts	A	¬A
1	1	0
2	0	1
3	1	0

(a)

Objekts	B	¬B
1	0	1
2	1	0
3	0	1

(b)

Objekts	C	¬C
1	0	1
2	1	0
3	0	1

(c)

Objekts	D	¬D
1	1	0
2	0	1
3	0	1

(d)

2.4. att. Binārās saistības metodes izveidotās četras datu kopas

Algoritmu adaptācijas pieeja paredz algoritmu pielāgošanu tā, lai daudzkategoriju uzdevumus varētu risināt tiešā veidā. Būtībā tas nozīmē jaunu algoritmu radīšanu; vairāki šādi algoritmi jau eksistē, piemēram, *MLkNN* [66], kas ir *k*-tuvāko kaimiņu klasifikatora daudzkategoriju versija.

Ja konkrētajā problēmas definējumā klases savstarpēji ir sakārtotas hierarhiskā struktūrā, tad veidojas jauna uzdevumu grupa, ko sauc par hierarhisku daudzkategoriju klasifikāciju [67, 68]. Daudzkategoriju klasifikācijas uzdevums ir sarežģītāks pats par sevi (salīdzinot ar uzdevumiem, kur piemērs pieder tikai vienai klasei), bet pievienojot hierarhisku klašu struktūru, risinājuma sarežģītība vēl pieaug.

Vairāk informācijas par dažādiem daudzkategoriju apmācības aspektiem var atrast [45, 65, 67-69].

Par sava veida daudzkategoriju klasifikāciju var uzskatīt **izplūdušo klasifikāciju** (ang. v. – *fuzzy classification*). Izplūdušās klasifikācijas pamati balstās uz Zadē izstrādāto izplūdušo kopu teoriju [70] un izplūdušo loģiku, kas operē ar jēdzieniem, kuru robežas nav precīzi definētas. Izplūdušās klasifikācijas gadījumā objekti var piederēt dažādām klasēm ar noteiktu piederības pakāpi. Tomēr, lai arī gan daudzkategoriju klasifikācija, gan izplūdusī klasifikācija ļauj definēt piederības funkciju vienlaicīgi vairākām klasēm, abu pieeju mērķi un risinātās problēmas ir atšķirīgas [71]. Uz izplūdušo loģiku balstītā klasifikācija ir līdzeklis neskaidrības kliedēšanai starp klases aprakstošajiem atribūtiem un var tikt uzskatīta kā sagatavošanās bloks pirms klasifikācijas, lai nošķirtu dažādas klases. Pēc klasifikācijas posma parasti seko solis

izplūdušā lēmuma pārvēršanai gala klasifikācijas lēmumā (ang. v. – *de-fuzzification*), piemēram, izvēloties klasi, kurai ir visaugstākā piederības vērtība. Izplūdušās piederības vērtības pēc normalizācijas iegūst summāro vērtību "1", kamēr daudzkategoriju klasifikācijā vairākas vai pat visas klases var saņemt piederības vērtību "1" (tādējādi summārai piederības vērtībai visām klasēm nav jābūt "1")[72].

Komplicētāka pieeja klašu piešķiršanai vienas kategorijas uzdevumos ir **naivais pārliecības klasifikators** [73] (ang. v. - *Naive Credal Classifier*), kurš ņem vērā neprecizitāti, kas radusies no nepārliecības par patieso sadalījumu datos. Klasifikators piešķir objektam vairākas klases, norādot uz nepārliecību par vienu vienīgo klasi. Par pamatu naivajam pārliecības klasifikatoram ir naivā Beijesa klasifikatora paradigma, un pieejas autors apgalvo, ka pieeja ir piemērota mazām un nepilnīgām datu kopām [73]. Atšķirībā no daudzkategoriju uzdevuma, kur arī tiek piešķirtas vairākas klases, naivais pārliecības klasifikators ir paredzēts problēmsfērām, kurās semantiski katrs objekts pieder tikai vienai klasei.

2.1.4. Klasifikatora veiktspējas novērtēšana

Lai varētu spriest, vai apmācības laikā iegūtais klasifikators ir derīgs jaunu piemēru klases piederības noteikšanai, tas ir jānovērtē. Viens no būtiskākajiem klasifikatora raksturlielumiem ir tā precizitāte. Lai noteiktu izveidotā klasifikatora korektumu jaunu ieejas datu analīzē, nepieciešams veikt validāciju. Validācija ir iegūtā klasifikācijas modeļa pārbaude ar testa datiem, kuriem ir zināma klase, bet kuri nav izmantoti klasifikatora apmācībai [74]. Praksē liela nozīme ir sākotnējās datu kopas sadalīšanai apmācības un testa kopā. Divas plašāk izmantotās pieejas, ar kuru palīdzību dati tiek dalīti šajās funkcionālajās kopās, ir novilcināšana (ang. v.- *holdout*) un šķērsvalidācija (ang. v. - *cross-validation*) [75].

Novilcināšanas metode realizē vienreizēju proporcionālu visas datu kopas dalījumu apmācības un testa kopā. Metode ir vienkārša, bet tai ir savi trūkumi [57]. Pirmkārt, sadalot datu kopu divās daļās, tiek samazināts gan apmācībai izmantojamo, gan testēšanai pieejamo piemēru skaits, turklāt apmācības un testa kopās ir jānodrošina vienlīdzīgs gadījumizlases sadalījums (ang. v. - *random sampling*). Tas ir sevišķi būtiski gadījumos, kad pieejams maz piemēru, tomēr rada sarežģījumus arī lielas datu kopas gadījumā. Šī iemesla dēļ novilcināšanas metodi nav vēlams izmantot reālos problēmu risinājumos. Otrkārt, nav zināms optimālais procentuālais sadalījums starp apmācības un testa kopām. Praksē visbiežāk lieto sadalījumu 70 % - apmācībai, 30 % - pārbaudei [75].

Ir vairāku tipu **šķērsvalidācijas metodes**. Piemēram, *K*-kārtu (angļu v. - *K-fold*) šķērsvalidācija sadala sākotnējos datus *K* apakškopās (skaitu *K* nosaka lietotājs) [57, 75]. *K*-1

apakškopa tiek izmantota apmācībai un 1 kopa - testēšanai. Šķērsvalidācijas process tiek izpildīts *K* reizes, katru reizi ņemot 1 no *K* apakškopām par pārbaudes kopu. Iegūtie rezultāti tiek kombinēti, lai iegūtu galīgo novērtējumu. Speciāls šķērsvalidācijas gadījums ir saknēšana (ang. v. – *bootstrapping*).

Izplatītākie klasifikatora novērtēšanas mēri uzdevumos, kur objekts pieder tikai vienai klasei, ir klasifikatora precizitāte (ang. v. - *precision*), atsaukums (ang. v. - *recall*), f-mērs (ang. v. - *f-measure*), kopējā klasifikatora precizitāte (ang. v - *accuracy*), nepareizi klasificēto piemēru īpatsvars (ang. v. - *error rate*), laukums zem ROC līknes (ang. v. - *Area Under the ROC Curve*), kā arī citi. Šie mēri ir plaši aprakstīti līdzšinējos darbos, piemēram [75, 76]. Būtisks klasifikatora veiktspējas rādītājs ir arī neklasificēto piemēru īpatsvars. Šī mēra izmantojamība ir atkarīga no klasifikācijas algoritma; daudzi algoritmi ‘uz āru’ neparāda nespēju noteikt klases piederību, bet labāk vai sliktāk veic klasifikāciju paši - klasifikatora iekšienē vai pievienojot klasifikāciju beigās. Sīkāk par šo problēmu tiks diskutēts 2.1.5.3. sadaļā.

Turpmāk šajā darba sadaļā uzmanība ir veltīta galvenajiem klasifikatora novērtēšanas mēriem daudzkategoriju uzdevumos, jo šī darba fokuss saistās ar daudzkategoriju klasifikāciju. Klasifikatoriem, kas darbojas ar daudzkategoriju datiem, ir savas specifiskas novērtēšanas metodes - gan klasifikācijas, gan klašu ranžēšanas novērtēšanai. Šeit apskatītās metrikas ir aprakstītas avotos [65, 77-79] un sīkāk tajos arī ir analizētas.

Klašu piešķiršanas (ang. v. - bipartitions) novērtējums

Klasifikācijas uzdevumos, kuros zināms, ka katrs objekts pieder tieši vienai klasei, labi noder precizitātes mēri, kas vērtējumu izsaka striktos "1" vai "0" novērtējumos. Daudzkategoriju gadījumā vairāk par šādiem mēriem izsaka zaudējuma funkcijas. Tas ir tādēļ, ka vairāku klašu piešķiršanas gadījumā pilnīgi precīzus risinājumus parasti iegūt neizdodas, tomēr daži risinājumi joprojām ir labāki (vēlamāki) par citiem – to arī nosaka zaudējuma funkcija. Galvenais, lai zaudējuma mērs izsaka to, kas patiešām ir svarīgs novērtējumā. Piemēram, ja objektam atbilst četras klases, tad klasifikators, kurš būs pareizi noteicis trīs no tām, ir strādājis labāk nekā tas, kurš noteicis tikai vienu klasi. Striktu precizitātes mēru lietošanas gadījumā abi klasifikatori tiktu novērtēti ar lietderību "0", jo neviens nav perfekti atradis visas klases.

Mērus var iedalīt uz piemēriem un klasēm balstītajos.

Uz piemēriem balstītie mēri:

Haminga zaudējums (ang. v. - *Hamming loss*) raksturo zaudējumu, kas saistīts ar Haminga attālumu. Tas ir attālums starp bināriem vektoriem, kas skaita, cik klašu nesakrīt (cik spriedumu

ir nepareizi) - būtībā nosaka simetrisko starpību starp objektam prognozētajām klasēm un tā īstajām klasēm. Tādu pašu rezultātu sniedz arī bināro klasifikatoru vidējā kļūda.

$$Haminga\ zaudējums = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \Delta Z_i|}{M}, \tag{2.1}$$

kur Δ nozīmē divu kopu simetrisko starpību vai XOR operatoru Būla loģikā,

m – apmācības piemēru skaits,

Yi – īstā klašu kopa *i*-tajam piemēram,

Zi – prognozētā klašu kopa *i*-tajam piemēram.

Klasifikācijas precizitāte vai **apakškopu precizitāte** (ang. v. - *classification accuracy* or *subset accuracy*) ir ļoti strikts mērs, kas prasa pilnīgu prognozētās un īstās klašu kopas sakrišanu, lai piešķirtu vērtību "1", pretējā gadījumā novērtējot ar "0".

$$Klasifikācijas\ precizitāte = \frac{1}{m}\sum_{i=1}^{m} I(Z_i = Y_i) \tag{2.2}$$

Uz klasēm balstītie mēri:

Šeit var lietot jebkuru mēru, kas paredzēts klasifikatoru binārai novērtēšanai. Mēri tiek pārveidoti, lietojot divas vidējo noteikšanas operācijas, makro-vidējais (ang. v. - *macro-averaging*) un mikro-vidējais (ang. v. - *micro-averaging*). F-mērs, kas lietots kopā ar mikro vai makro vidējo, ir viens no biežāk izmantotajiem daudzkategoriju klasifikācijas rezultātu novērtēšanā [77].

$$B_{makro} = \frac{1}{q}\sum_{\lambda=1}^{q} B\,(tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda), \tag{2.3}$$

$$B_{mikro} = B\left(\sum_{\lambda=1}^{q} tp_\lambda,\ \sum_{\lambda=1}^{q} fp_\lambda,\ \sum_{\lambda=1}^{q} tn_\lambda,\ \sum_{\lambda=1}^{q} fn_\lambda\right), \tag{2.4}$$

kur tp_λ - patiesi pozitīvo klasifikāciju skaits klasei λ,

fp_λ - nepatiesi pozitīvo klasifikāciju skaits klasei λ,

tn_λ – patiesi negatīvo klasifikāciju skaits klasei λ,

fn_λ - nepatiesi negatīvo klasifikāciju skaits klasei λ,

q – klašu skaits,

B – kāds novērtējuma mērs, piemēram, (2.5), (2.6) vai (2.7).

$$Precizitāte = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap Z_i|}{|Z_i|}, \tag{2.5}$$

$$Atsaukums = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap Z_i|}{|Y_i|}, \tag{2.6}$$

$$F\ mērs = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}, \quad (2.7)$$

kur ∩ nozīmē divu kopu šķēlumu,

∪ nozīmē divu kopu apvienojumu,

m – apmācības piemēru skaits,

Yi – īstā klašu kopa *i*-tajam piemēram,

Zi – prognozētā klašu kopa *i*-tajam piemēram.

*Sakārtojuma (*ang. v. - *ranking) novērtējums*

Sakārtojuma jeb ranga novērtēšana sniedz ieskatu tajā, cik pareizi klasifikators noteicis objekta klases pēc to atbilstības.

Vienas kļūdas (ang. v. - *One-error*) novērtējums nosaka, cik reizes visaugstāk vērtētā jeb par visatbilstošāko prognozētā klase (sakārtotu klašu gadījumā) nav viena no piemēra īstajām klasēm.

$$Viena\ kļūda = \frac{1}{m}\sum_{i=1}^{m}\delta\left(arg\min_{\lambda \in L} r_i(\lambda)\right), \quad (2.8)$$

kur $\delta(\lambda) = \begin{cases} 1, ja\ \lambda\ \notin Y_i \\ 0, ja\ \lambda\ \in Y_i \end{cases}.$

Kļūdas kopas izmērs (ang. v. - e*rror set size*) ir zaudējuma funkcija, kas atgriež klašu pāru skaitu, kas nav piešķirti pareizā secībā.

Pārklājums (ang. v - *coverage*) ir novērtējums, cik vidēji tālu ir jāiet sakārtotajā sarakstā, lai pārklātu visas objektam piederošās klases.

$$Pārklājums = \frac{1}{m}\sum_{i=1}^{m}\max_{\lambda \in Y_i} r_i(\lambda) - 1 \quad (2.9)$$

Sakārtojuma zaudējums (ang. v. - *ranking loss*) definē, cik reizes piemēram nepiederīgās klases ir novērtētas augstāk nekā īstās klases.

Vidējā precizitāte (ang. v. - *average precision*) aprēķina katrai īstajai klasei, cik procenti no klasēm, kas atrodas sarakstā virs tās, ir citas īstās klases, un iegūst vidējo vērtību visām īstajām klasēm.

Daļa no šiem mēriem tiks izmantota rezultātu novērtēšanā promocijas darba praktiskajos eksperimentos.

Nākamā darba sadaļa pievēršas klasifikācijas uzdevumu, sevišķi induktīvās apmācības, problemātikai un tiem risināmajiem jautājumiem, kas tiks izskatīti šajā promocijas darbā.

2.1.5. Problēmas klasifikācijā un induktīvajā apmācībā

Galvenās problēmas, kas jārisina klasifikatora izveidošanā, ir līdzīgas visām klasifikācijas metodēm un ir saistītas ar datiem un to apstrādi. Dati var būt nepareizi savākti vai apkopoti, trokšņaini, ar trūkstošām vērtībām. Tomēr vispirms dati ir jāiegūst. Novērojumu ceļā iegūtie dati parasti tiek glabāti un izgūti no kāda veida datu bāzes. Datu var būt neaptverami daudz vai nepietiekami maz. Reālajā pasaulē dati bieži vien jāiegūst no vairākiem avotiem, kas prasa datu integrāciju un iespējamo konfliktu risināšanu. Konflikti var būt saistīti ar dažādu kodējumu un attēlojumu [75].

Problēmas, kas ir saistītas ar datu priekšapstrādi, ir plaši apskatītas literatūrā [74, 75, 80-83] un netiks padziļināti analizētas. Ir zināms, ka datu priekšapstrāde var aizņemt pat 80% no kopējā datu analīzes un klasifikatora veidošanas laika un ietver tādus etapus kā datu atlasīšana, attīrīšana un transformēšana, nepiederošo (ang.v.- *outlier*) un trokšņaino (ang. v. – *noisy*) datu izķeršana, datu kodēšana un raksturīgo iezīmju jeb atribūtu izvēle.

Cita klasifikatoriem raksturīga problēma ir pārapmācība (ang. v. – *overfitting*). Tā ir algoritma īpatnība, kas piešķir nozīmīgumu datu novirzēm, pieņemot tās par svarīgām datu kopas iezīmēm [74]. No pārapmācības lēmumu kokos var izvairīties, pārtraucot koka palielināšanu brīdī, kad no koka tālākas izvēršanas nevar vairs iegūt nozīmīgu informāciju [83]. Līdzīgu pieeju var izmantot arī likumu ģenerēšanas metodēs. Lēmumu kokos izmantota pārapmācības novēršanas metode ir atzarošana – tiek izslēgti dalījumi koka zemākajos līmeņos [74]. Parasti tiek noteikts kāds sliekšņa lielums, kurš raksturo, pie kāda ierakstu skaita tālāka koka zarošana netiek veikta.

Sīkāk tiks apskatīti citi risināmie jautājumi, kas rada problēmas induktīvās apmācības metodēm un klasifikatoriem kopumā. Klasifikācija mazas apmācības kopas gadījumā un nespēja klasificēt jaunu piemēru ir problēmas, kuras definētas darbā risināmā uzdevuma nostādnē, tāpēc tiks izskatītas nākamajās darba sadaļās.

2.1.5.1 Atkarība no apmācības kopas apjoma

Dažādas problēmas ir saistītas ar pieejamo datu daudzumu. Mašīnapmācības metodēm ir atšķirīgas sekmes darbojoties ar ļoti lielu vai mazu apmācības datu apjomu. Atsevišķos uzdevumos apstrādājamo piemēru un atribūtu skaits ir kritisks parametrs klasifikācijas metodes izvēlē.

Liela apmācības kopa

Milzīgi datu apjomi (ang. v. - *big data*) ir viena no būtiskākajām datu analīzes problēmām, ar ko cīnās mūsdienu praksē [2]. Lielu datu apjomu apstrādes iespējas kā kritisku

nepieciešamību atzīst jau visā pasaulē - gan Eiropas Komisija 7. ietvara programmas uzsaukumi [84], gan *IBM* [85], gan *The Wall Street Journal* savos rakstos [86], kā arī citi nozīmīgi avoti. Tā ir atsevišķa un plaša sfēra, tādēļ šajā darbā lielu datu apjomu apstrādes problēmai uzmanība pievērsta netiks.

Neliela apmācības kopa

Ne visās jomās ir jāsaskaras ar lielu apstrādājamo datu daudzumu. Dažās sfērās problēma ir pilnīgi pretēja – nepieciešamo datu ieguve ir apgrūtināta [69]. Datus iegūt var būt dārgi, sarežģīti, bīstami vai neiespējami. Apmācības piemēru skaits, ko dēvē par mazu, atšķiras dažādu uzdevumu gadījumā, piemēram, avota [87] pētījumos par klasifikāciju mazu datu kopu apstākļos variē ar apmācības kopu 5-70 ierakstu apjomā. Teksta klasifikācijas uzdevumos, kur atribūtu skaits ir milzīgs, par mazu apmācības piemēru skaitu tiek uzskatīti pāris tūkstoši ierakstu (piemēram, 1800 piemēri ar 5171 atribūtiem un 36 klasēm [88]).

Hamalainens un Vinni [89] apliecina, ka galvenā problēma, piemēram, izglītības datu apstrādē, ir datu trūkums. Apskatītajā sistēmā tiek izmantoti studentu dati, lai prognozētu, vai students noliks eksāmenu. Mērķis ir to noteikt pēc iespējas agrākā mācību fāzē. Izglītības jomā iegūstamais ierakstu skaits parasti ir 50 līdz 100 piemēri vienai analizējamajai parādībai. Tas ir maz, ja ņem vērā, ka parasti mašīnapmācības uzdevumos tiek izmantoti vairāki tūkstoši apmācības piemēru. Turklāt pieejamo atribūtu ir samērā daudz un ne visi ir reprezentatīvi. Cits būtisks apstāklis šajā klasifikācijas uzdevumā ir izglītības datu heterogēnais veids – dati ir gan nomināli, gan skaitliski. Tomēr datiem izglītības jomā ir arī kāda priekšrocība, salīdzinot ar daudzām citām problēmsfērām - datu kopas parasti ir diezgan uzticamas, t.i., dati ir korekti un nesatur trokšņus, kas iegūti novērojumu laikā.

Čangs apskata dažādas iespējas mazu datu kopu gadījumā, piemēram, piedāvā mākslīgi palielināt datu kopu, ģenerējot papildu apmācības piemērus ar matemātisku funkciju palīdzību [87]. Tomēr Čanga piedāvātā mega-izpludināšanas (ang. v. - *mega-fuzzification*) metode, tāpat kā izplūdušais neironu tīkls [90], ir derīga problēmsfērās ar ierobežotu atribūtu skaitu (piemēram, jau 15 atribūtu gadījumā tas vairs nestrādā).

Jāņem vērā arī iespēja, ka limitēta datu apjoma gadījumā var trūkt kāda būtiska informācija par problēmsfēru [87]. Nepilnīgas apmācības kopas apstākļos inducētais klasifikators arī var būt nepilnīgs, līdz ar to apgrūtinot jaunu piemēru klasifikāciju. Papildus jārēķinās, ka klašu sadalījums apmācības kopā var neatbilst īstajam klašu sadalījumam problēmsfērā [88].

2.1.5.2 Problēmas jaunu piemēru klasifikācijā

Dažādas problēmas ir jārisina brīdī, kad klasifikators ir izveidots un tiek lietots jaunu objektu klasificēšanai [91]. Jauna piemēra klasifikācija notiek, meklējot šim piemēram atbilstošu likumu vai koka zaru klasifikācijas modelī. Šajā procesā var atgadīties šādi konflikti [92]:

1. Vairāk nekā viens likums (vai koka zars) var klasificēt piemēru, un katrs no tiem paredz atšķirīgu klasi;
2. Neviens likums (vai koka zars) neklasificē piemēru.

Pirmajā gadījumā ir vairākas metodes konflikta risināšanai, kas plašāk ir aprakstītas [83, 93]. Piemēram, induktīvās apmācības algoritms *AQR* starp pretrunīgi klasificējošajiem likumiem izvēlas to, kas prognozē klasi, kura ir biežāk sastopama starp apmācības piemēriem [83]. Cits bieži sastopams, vienkāršs un efektīvs veids, kā atrisināt konfliktu, ir labākā likuma stratēģija, kas izmanto likumu saskanību un pilnību, lai izvēlētos kvalitatīvāko likumu [92, 93].

Otrā gadījuma risināšanā visbiežāk tiek izmantots noklusētais likums. Literatūrā nav sastopamas plašas diskusijas par citu pieeju lietošanu situācijās, kad klasifikators nav spējīgs noteikt klases piederību jaunajam piemēram. Tomēr noklusētā likuma lietošana nav piemērota visās situācijās. Tādēļ šī problēma tiks sīkāk analizēta darba nākamajā sadaļā.

2.1.5.3 Neklasificēti piemēri

Pat tad, ja klasifikatora precizitāte testa piemēriem ir bijusi apmierinoša, mēdz gadīties, ka klasifikators nevar piešķirt jaunajam piemēram klasi. Iespējams, ka klasifikācijai tiek nodots kāds unikāls vai izņēmuma gadījums vai arī klasifikators nepārklāj visus problēmsfēras aspektus. Var būt arī, ka izmantotie atribūti nepilnīgi apraksta problēmsfēru un ir jāsaskaras ar slēpto kontekstu.

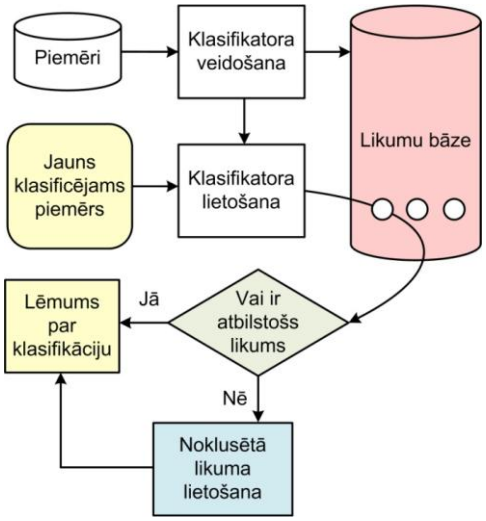
Skaidrības labad jāpaskaidro turpmāk izmantotais terminu lietojums.

*Piemērs, kurš vēl nav klasificēts, jo nav padots klasifikatoram klasificēšanai, promocijas darbā tiek saukts par **jaunu** vai **klasificējamu** piemēru, par **neklasificētu** piemēru saucot tādu piemēru, kuram klasifikators nav spējis noteikt klases piederību pēc klasifikatora lietošanas.*

CN2 induktīvās apmācības algoritms popularizē noklusētā likuma izmantošanu, kas nozīmē, ka likumu saraksta beigās ir papildu likums, kurš nosaka, ka visiem tiem piemēriem, kam neatbilda neviens no iepriekšējiem likumu bāzes likumiem, jāpiešķir viena konkrēta klase [83]. Tā parasti ir visbiežāk sastopamā klase apmācības piemēru kopā. Šādi pati pieeja lietota arī *AQ* algoritmā [83], noklusēto likumu lieto arī *C4.5rules* algoritms [94] un tā papildinātā

komerciālā versija *C5.0*, kā arī skudru kolonijas algoritms *Ant-Miner* [95]. 2.5. attēlā ir atspoguļots klasisks klasifikācijas process ar noklusētā likuma lietošanu neklasificētu piemēru gadījumā.

Klasifikatora veidošanas bloks iekļauj gan apmācības, gan testēšanas posmus. Vienkāršības labad*, izmantojot terminu „likumi", turpmāk darbā tiks apzīmēts klasifikācijas modelis gan IF- THEN likumu, gan lēmumu koka veidā*.



2.5. att. Klasifikācija ar noklusētā likuma lietošanu

Ja jauna piemēra klasificēšanas laikā likumu bāzē tiek atrasts atbilstošs likums, tad lēmumu par klasifikāciju var pieņemt uzreiz. Ja atbilstoša likuma nav, ir jāizdara ticams minējums par piemēra klasi. Parasti tas nozīmē noklusētā likuma lietošanu.

Noklusētā likuma lietošana ir viegli saprotama un realizējama pieeja neklasificētu piemēru klases piederības noteikšanai, tomēr tā nav piemērota visos gadījumos. Tā, piemēram, ja problēmsfērā ir izdalāmas daudzas klases, turklāt, ja tās ir vienlīdz plaši izplatītas un nevienā no tām piemēri nav skaitliskā pārsvarā, tad vienas konkrētas klases piešķiršana visiem neklasificētajiem piemēriem nevar nodrošināt augstu klasifikatora precizitāti neklasificētajiem piemēriem. Tas ir uzskatāmi, ka noklusētā likuma lietošana problēmsfērā ar 10 klasēm, kuru pārstāvji parādās caurmērā vienlīdz bieži, nodrošina tikai 10% precizitāti neklasificētajiem piemēriem, pareizi klasificējot tikai katru desmito piemēru. Šāds rezultāts ir neapmierinošs.

Ir problēmsfēras, piemēram, medicīnas diagnostika, kur jāņem vērā, ka klašu nepareizas noteikšanas izmaksas (ang.v. - *misscalsification costs*) nav vienādas. Vienas klases nepareiza noteikšana ir ļoti būtiska (tas ir, ja slimība ir, bet tā netiek konstatēta); šaubu gadījumā var tikt piešķirta svarīgākā klase, lai nepieļautu nozīmīgu kļūdu. Slimības konstatēšanas gadījumā rezultātu pacientiem ar pozitīvu diagnozi ir iespējams vēl pārbaudīt, tāpēc ir drošāk piemēru

klasificēt kā „nepatiesi pozitīvu", nevis „nepatiesi negatīvu", tādējādi palaižot garām slimības gadījumu [96]. Jārēķinās, ka nesabalansētas datu kopas, kur viena klase parādās daudz biežāk nekā cita, ir plaši sastopamas praktiskos lietojumos, tādēļ arī šajos gadījumos būtu jānodrošina pēc iespējas precīzāka klasifikācija [97].

Kā atzīts [98], noklusētā likuma pievienošana likumu kopai ir nevēlama arī tādēļ, ka noklusētais likums nesniedz citu semantiski lietderīgu interpretāciju kā vien „neviens cits likums nederēja". Šajā gadījumā ir būtiski izšķirt noklusētos likumus, kas tiek pievienoti likumu kopas beigās un tādus, kas ir paredzēti apmācībās algoritmā kā pamatsastāvdaļa. Lai arī nosaukums ir viens, to būtība ir atšķirīga. Tādi induktīvās apmācības algoritmi kā, piemēram, *Ridor* [99] un *Ripper* [62] savu darbību sāk ar noklusēto likumu, kuram veido izņēmumus. Šajā gadījumā noklusētajam likumam ir lielāka saistība ar sakarībām datu kopā, lai arī blakusefekts saglabājas tāds pats, kā beigās pievienoto noklusēto likumu situācijā - jebkuram jaunajam objektam, ko klasifikatoram uzdots klasificēt, tiks noteikta klase, neatkarīgi no klasifikatora pārliecības par piešķirtās klases pareizību.

Literatūrā ir piedāvāta arī cita iespēja to piemēru klasificēšanā, kam netiek uzreiz noteikta klases piederība. Dinamiskā likumu apmācības pieeja skaitliskiem datiem, kas balstās uz piemēru attāluma jēdzienu, ir aprakstīta [100]. Šī apmācības sistēma lieto divu veidu likumus. Saskanīgie likumi klasificē jaunus piemērus, meklējot atbilstošu pārklājošu likumu, kamēr nesaskanīgie likumi klasificē piemērus pēc attāluma, kā to dara k-tuvāko kaimiņu algoritms. Piemērs, kuru nevar klasificēt uzticamie saskanīgie likumi, var tikt klasificēts ar nesaskanīgo likumu, piešķirot tādu klases vērtību, kāda ir tuvākajam kaimiņam. Tomēr šī pieeja ir specifiska un nestrādā datiem ar nomināliem atribūtiem, starp kuriem attālumu nevar aprēķināt.

Klasifikācijas uzdevumiem kļūstot arvien sarežģītākiem un mēģinot paplašināt ar induktīvo apmācību risināmo problēmu loku, gadījumu skaits, kad klasifikators nespēj klasificēt jaunu piemēru, tāpat kā nepareizi klasificēto piemēru skaits, pieaug. Atstāt lēmuma pieņemšanu kādam predefinētam algoritmam ne vienmēr ir labākā izvēle. Daļa mašīnapmācības sistēmu cenšas mazināt eksperta vai lietotāja iesaistīšanu apmācības procesā, kamēr citas ievieš sadarbības mehānismus starp sistēmu un tās lietotāju. Situācijā, kad klasifikators nespēj viennozīmīgi klasificēt jauno piemēru, sadarbība starp datorsistēmu un sistēmas lietotāju (problēmsfēras ekspertu) būtu lietderīga. Šāda interaktīva induktīvās apmācības pieeja tiks piedāvāta darba 3. nodaļā, bet nākošajā apakšnodaļā tiks apskatīti dažādi interaktivitātes veidi, kas ir sastopami klasifikācijas sistēmās šobrīd.

2.2. Interaktivitāte klasifikācijā un induktīvajā apmācībā

Šī apakšnodaļa iepazīstina ar interaktīvas induktīvās apmācības jēdzienu un tā dažādajām izpratnēm, kā arī esošajām interaktīvajām klasifikācijas pieejām. Interaktivitāti klasifikācijas procesā tiešā veidā skar arī aktīvās mācīšanās pieeja un *Ripple down* likumi, kuri tiks apskatīti atsevišķi, jo šie ir samērā patstāvīgi mašīnapmācības un klasifikācijas virzieni.

Pēdējos divdesmit gados literatūrā jēdziens „interaktīva induktīvā apmācība" ir traktēts ļoti dažādi, tāpat kā iemesli lietotāja vai eksperta iesaistīšanai induktīvas apmācības procesā. Jomas eksperts savas zināšanas izmanto jau risinājuma apgabala definēšanā, sniedzot pārmeklēšanas heiristiku vai savas zināšanas par problēmsfēru, piemēram, definējot piemērotu atribūtu kopu [101]. Dažādās sfērās definētās sistēmas piedāvā dažāda līmeņa komunikāciju ar lietotāju, iesaistot to klasifikācijas posmos. Promocijas darba autore ir apkopojusi [102] dažādu avotu sniegtos aprakstus [6, 103-107], kuros iespējams izdalīt šādus interaktivitātes **veidus**:

1. sistēmas, kurās tiek prasīta atgriezeniskā saite no eksperta, lai novērtētu iegūtos rezultātus;
2. sistēmas, kas mācās atpazīt konceptu, balstoties uz eksperta sniegto klasifikāciju;
3. sistēmas, kur eksperts vispirms sniedz savas zināšanas sistēmai un pēc tam apstiprina sistēmas inducētos likumus;
4. eksperts novērtē un atlasa sistēmas inducētos likumus klasifikatora veidošanas posmā;
5. apmācības sistēmas, kur mācās lietotājs, un sistēmai ir jāspēj komunicēt lietotājam draudzīgā formā.

Lai izskaidrotu katru no interaktivitātes veidiem, sīkāk tiks apskatītas konkrētas sistēmas un lietojumi. **1. veidam** atbilst Okabes un Jamadas [103] piedāvātā sistēma, kas uzlabo globālā tīmekļa meklētāja rezultātus. Globālā tīmekļa meklētāji parasti meklēšanas rezultātos parāda arī daudz nesvarīgu un neatbilstošu lapu, jo lietotāja ievadītais vaicājums nav bijis gana specifisks, tāpēc vaicājumu specificēšanai ir piedāvāts izmantot interaktīvu procesu ar atgriezenisko saiti no lietotāja. Vaicājumam tiek iegūts specifisks filtrs, kas sastāv no likumu kopas, kuri meklētājam palīdz noteikt to, vai lapu atspoguļot rezultātos, vai nē. Filtru veido predikātu loģikā sakņots induktīvās loģiskās programmēšanas (ILP) algoritms *FOIL*. Pēc meklēšanas rezultātu iegūšanas lietotājam tiek prasīts novērtēt to atbilstību un svarīgumu. Lietotāja novērtētās lapas tiek saglabātas apmācībai, analizētas, izmantotas filtrēšanas likumu ģenerēšanai, beigās atkārtojot meklēšanu, šoreiz jau lietojot arī likumu filtru.

Tanumara, Ksī un Au [104] apraksta pieeju, kurā datorsistēma apgūst dažādu krāsu konceptus, kurus vienu pēc otra ir klasificējis eksperts (interaktivitātes **2. veids**). Sistēmai pašai

nav iespēja izveidot vai mainīt kategorijas. Šajā gadījumā komunikācija ar ekspertu notiek nepārtraukti sistēmas apmācības laikā. Sistēma mācās klasificēt krāsas, veidojot neironu tīklu. Lai arī klasifikators nav balstīts uz induktīvo spriešanu, tas nemaina iespējamo sadarbības mehānismu. Demonstrētās sistēmas rezultāts liecina, ka iespējams iegūt cilvēka intelektam tuvinātu krāsu uztveres mehānismu, nepielietojot sarežģītu cilvēka uztveres imitāciju [104].

Ar interaktivitāti ir papildināta arī datorsistēmu apmācība ILP tehnikā [108], ko sistēmā *CLINT* ieviesuši Vongs un Leungs [106]. ILP izmanto zināšanas par problēmsfēru un piemēru kopu, kas ir atspoguļota kā faktu datu bāze. Papildu citiem ieejas datiem, interaktīvā ILP sistēma *CLINT* var saņemt arī atbildes no eksperta uz pašas sistēmas ģenerētajiem jautājumiem - klasifikācijas pieprasījumiem (**2. veids**). Sistēma pārbauda arī integritātes nosacījumus starp esošajām un jaunajām zināšanām.

Buntine un Stirlings [6] argumentē, ka indukcijai būtu jābūt interaktīvai gan tādēļ, lai klasifikators iegūtu vairāk subjektīvas informācijas, gan, lai indukcijas rezultāts gūtu apstiprinājumu no eksperta (**3. veids**). Informācija klasifikatoram tiek iegūta no eksperta un arī indukcija kopumā netiek skatīta kā automātisks process. Autori skaidro, ka indukcija nekad nav bijusi izolēta no cilvēka darbībām, jo jebkurā gadījumā eksperts ir tas, kurš definē klases un izvēlas raksturīgos atribūtus. Buntine un Stirlings neuzskata, ka lietotāja vai eksperta iesaistīšana dažās induktīvās apmācības fāzēs radītu problēmas vai būtu definējama kā sistēmas trūkums.

Hadjimičels un Vasilevska [105] piedāvā varbūtisku induktīvās apmācības sistēmu, kurā eksperts ir cieši iesaistīts, vispirms sniedzot sistēmai nosacījumus, un pēc tam atlasot turpmāk izmantojamos nosacījumus no sistēmas piedāvātajiem. Šajā pieejā lietotāja loma ir līdzīga lēmumu koku atzarošanas jeb vispārināšanas tehnikām. Sistēma izvada visus ģenerētos likumus, un eksperts izvēlas, kurus no tiem saglabāt un kurus atmest (**4. veids**). Atšķirībā no automātiskas koku atzarošanas, eksperts var izvērtēt pilnu likumu spektru, pirms daļa informācijas tiek zaudēta.

Sistēmās, kur lietotāja – datora sadarbība ir daļa no cilvēka mācību procesa, kā, piemēram, visaptverošajā intelektuālajā apmācībā (ang.v. - *ambient intelligent learning*) [107], mijiedarbībai starp apmācāmajiem un datorsistēmu ir jābūt tik dabiskai, lai lietotāji neizjustu datortehnoloģiju klātbūtni kā traucēkli mācību procesā (**5. veids**).

Nākamajā sadaļā tiks analizēti dažādie interaktivitātes jēdzieni vienotā kontekstā.

2.2.1. Esošo interaktīvo pieeju analīze

Iepriekš tika aprakstītas dažādas induktīvās sistēmas, kas iesaista cilvēku savā darbībā. Tagad šīs sistēmas tiks analizētas attiecībā uz iespēju tās izmantot neklasificēto piemēru apstrādē.

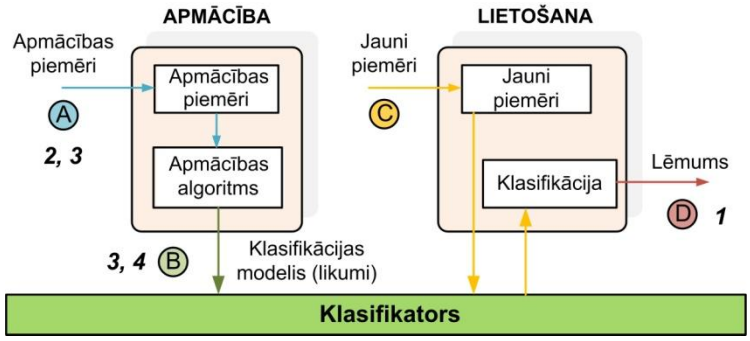
Sistēma [103], kas lieto nozīmības atgriezenisko saiti (ang. v. – *relevance feedback*), mijiedarbojas ar lietotāju pašā pēdējā klasifikatora lietošanas brīdī. Lai risinātu neklasificēto piemēru problēmu induktīvajā apmācībā, šis brīdis jau ir par vēlu, jo lietotājs tiek iesaistīts tad, kad visa klasifikācija procedūra jau ir noslēgusies un lēmumi par klases piešķiršanu ir klasifikatorā pieņemti.

Hadjimičela un Vasilevskas piedāvātā sistēma [16], kas vispusīgi iesaista ekspertu klasifikatora veidošanā, ir daudzsološa, tomēr problēmas ar tās lietošanu sākas tad, kad klasifikācijas modelis, ko apraksta apmācības piemēri, ir sarežģīts, inducētie likumi ir gari un/vai daudzi. Ekspertam būtu grūti salīdzināt dažus desmitus likumu, nemaz nerunājot par simtiem. Buntinem un Stirlingam [6] neapšaubāmi ir taisnība tajā ziņā, ka eksperta intuīcija un problēmsfēras zināšanas ietekmē klasifikatora veidošanu, un to nemaz nevar izslēgt no apmācības procesa. Tomēr sistēmas izstrāde pēc šo autoru ieteikumiem prasa ļoti daudz eksperta pūļu, ja tajā pat laikā klasifikatoru ir iespējams iegūt ātrāk un automatizētāk. To pašu varētu teikt par sistēmu, kas apgūst krāsu konceptus no eksperta nepastarpināti demonstrētiem piemēriem [104].

Vonga un Leunga sistēma [106] *CLINT* ir diezgan tuvu koncepcijai par eksperta palīdzību sarežģītāko piemēru klasificēšanā. Tomēr šī sistēma vaicā eksperta viedokli apmācības laikā, kas nekādi nenovērš neklasificētu piemēru problēmu sistēmas lietošanas laikā.

Tādas sistēmas kā [107] aprakstītā visaptverošā intelektuālā apmācība nav attiecināmas uz risināmo problēmu, jo tajās lietotāja – datora sadarbība tiek skatīta kā regulāra darbība un mācīšanās virziens ir pretējais, jo mācās cilvēks, nevis tiek apmācīta datorsistēma.

No analīzes var secināt, ka sadarbība ar sistēmas lietotāju izpētītajās sistēmās notiek dažādos klasifikatora apmācības posmos. Balstoties uz fāzi, kurā interaktivitāte ar ekspertu ir sagaidāma, 2.6. attēlā ir atspoguļots abstrakts apkopojums par dažādiem līdz šim piedāvātajiem interaktivitātes modeļiem. Aplīši ar burtiem 2.6. attēlā norāda konkrēto fāzi, kurā ir sagaidāma sadarbība ar lietotāju vai ekspertu, bet skaitļi pie burtiem apzīmē nodaļas sākumā izdalītos interaktivitātes veidus, kuri izpaužas kā sadarbība ar lietotāju atbilstošās fāzēs.



2.6. att. Interaktivitātes brīži starp sistēmu un lietotāju

Klasifikatora veidošanas posmā tiek padoti apmācības un testa piemēri, izejā iegūstot likumus. Klasifikatora lietošanas posmā klasifikatoram tiek dots jauns piemērs, kuram nav zināma klasifikācija, un tiek sagaidīts lēmums par klases piederību šim piemēram. Tādejādi klasifikācijas procesā var izdalīt šādas fāzes:

A klasifikatora apmācības datu veidošana, datu atlase;

B likumu izgūšana, apstrāde un atlase;

C darbs ar jauniem, vēl neklasificētiem piemēriem;

D lēmuma apstrāde pēc jauna piemēra klasificēšanas.

Lietotāja iesaistīšana klasifikatora apmācības laikā (fāze A 2.6. att.) ir praktizēta divos veidos – kā mācīšanās tikai no eksperta demonstrētiem piemēriem [104, 109] vai mācīšanās no eksperta sniegtām atbildēm uz sistēmas ģenerētiem jautājumiem [106]. Sistēmas, kas aprakstītas [6, 105] iesaista ekspertu gan apmācības datu sagatavošanā, gan apmācības rezultātu novērtēšanā (fāzes A un B). Sistēmā [103] atgriezeniskā saite no lietotāja tiek gaidīta pēc klasifikācijas veikšanas, lai uzlabotu nākamās meklēšanas rezultātus (fāze D).

Neviena no iepriekš apskatītajām metodēm nepiedāvā atbilstošu interaktivitātes modeli induktīvās apmācības problēmas risināšanai – eksperta iesaistīšanai tādu piemēru apstrādē, kam likumu bāzē nav atrasts atbilstošs likums. Esošās lietotāja iesaistes notiek vai nu par ātru, vai par vēlu, kad lēmums jau ir pieņemts. Tādēļ, apkopojot esošos piedāvājumus un novērtējot to trūkumus, ir redzama nepieciešamība definēt interaktīvu induktīvās apmācības pieeju, kur eksperts tiek iesaistīts fāzē C. Tālāk šis piedāvājums tiks attīstīts darba 3. nodaļā.

Eksperta iesaistīšana mācīšanās procesā klasifikācijas uzdevumos ir aktuāla arī citās pieejās, kuras atsevišķi tiks apskatītas nākamajās divās darba sadaļās.

2.2.2. Aktīvā mācīšanās

Aktīvā mācīšanās ir pārraudzītās mašīnapmācības apakšjoma, kurā apmācības sistēma var uzdot jautājumus. Tā balstās uz hipotēzi, ka apmācības algoritms sasniegs labākus

klasifikācijas rezultātus ar mazāku apmācības piemēru kopu, ja pats varēs izraudzīties piemērus, no kuriem mācīties. Lai varētu veikt aktīvo mācīšanos, ir jānodrošina divi apstākļi [110]:

1. sistēmai jābūt iespējai uzdot jautājumus ekspertam;
2. jauni piemēri ir brīvi pieejami vai viegli iegūstami.

Jautājumu uzdošana izpaužas kā sistēmas vaicājums ekspertam jauna piemēra klases piederības noteikšanai. Kā uzsver Setls [110], visās situācijās, kur izmantojama pārraudzītā apmācība, ir iespējama arī aktīvā mācīšanās. Aktīvās mācīšanās sistēmas ir sevišķi aktuālas jomās, kur piemēri bez klasifikācijas ir viegli pieejami, bet klases noteikšana ir dārga, sarežģīta vai laikietilpīga. Aktīvā mācīšanās notiek kā iteratīvs process. Sistēma sāk apmācību ar nelielu daudzumu klasificētu apmācības piemēru. Tad, balstoties uz kādu *iepriekš noteiktu kritēriju*, tiek izvēlēts jauns, vēl neklasificēts piemērs, kurš, ja tam būtu zināma klase, sniegtu vislielāko ieguvumu klasifikatora uzlabošanai. Šis piemērs tiek rādīts ekspertam, kurš tam piešķir klasi, un pievienots apmācības kopai. Klasifikators tiek atjaunots, un process turpinās. Kritēriju, kas nosaka, vai piemērs ir lietderīgs vaicāšanai, var izvēlēties dažādos veidos. Pamatā šim kritērijam ir potenciāli vaicājamā piemēra informatīvuma novērtējums. Viena no vienkāršākajām un biežāk lietotajām stratēģijām informatīvuma novērtēšanai ir nedrošo gadījumu izlase (ang. v. - *uncertainty sampling*). Šajā gadījumā vaicāšanai tiek izvēlēti tie piemēri, par kuru klases piederību klasifikators ir visnedrošākais. Citas stratēģijas ir apkopotas un sīkāk aprakstītas [110]. Ir trīs galvenie scenāriji, kā tiek radīti vai izvēlēti piemēri vaicāšanai:

1. piemēri var tikt mākslīgi sintezēti;
2. ja jaunie piemēri bez klasifikācijas parādās dinamiski, pa vienam, tad algoritms izvēlas, vai piemērs ir lietderīgs vaicāšanai vai nē;
3. ja jauni piemēri ir pieejami plašā skaitā uzreiz, tad tie tiek izanalizēti un atlasīti vaicāšanai vēlamie piemēri.

Daudzās reālās problēmsituācijās pirmais variants nav lietderīgs, jo šādi var tikt veidoti neiespējami apmācības piemēri. Piemēram, veidojot klasifikatoru rakstzīmju atpazīšanai, tika konstatēts, ka algoritms sintezē neeksistējošas rakstzīmes uz esošo rakstzīmju elementu pamata [111]. Praksē biežāk izplatīts ir otrais un trešais scenārijs [112].

2.2.3. *Ripple down* likumi

Ripple down likumi (*Ripple-Down Rules*, turpmāk tekstā - *RDR*) ir inkrementāla pieeja zināšanu iegūšanai, kas aptver vairākas tehnikas. *RDR* ietvaru izstrādāja Komptons (*Compton*) un Jansens (*Jansen*), lai risinātu problēmas, kuras tika atklātas medicīnas ekspertu sistēmas *CARVEN-ES1* uzturēšanā [113]. Viens no galvenajiem novērojumiem bija tāds, ka eksperti

nekad nedeva plašu skaidrojumu savu lēmumu pieņemšanā. Tie labprātāk pamatoja, ka secinājumi ir pareizi, un pamatošana tika saistīta ar kontekstu, kādā informācija tika pasniegta. Sākumā *RDR* mērķis bija nodrošināt zināšanu iegūšanu un pārvaldības risinājumus, kas pienācīgi tiktu galā ar zināšanu kontekstuālo daļu. *RDR* gadījumā, datorsistēma palīdz izgūt zināšanas par problēmsfēru no eksperta, ļaujot tam darboties kontekstā, nodrošina zināšanu glabāšanu likumu veidā un veido datu struktūru neatkarīgi – bez eksperta iesaistīšanas. *RDR* ir izmantoti, piemēram, tādās problēmsfēras kā personalizēta e-pastu filtru veidošana [114] un medicīnas dokumentu komentāru automātiska veidošana [115].

Galvenās *RDR* pazīmes ir šādas [116]:

- sistēma organizē un kontrolē katru jaunu jebkāda veida zināšanu pievienošanu likumu bāzei;
- pirms izmaiņu ieviešanas sistēma veic pārbaudi, lai nodrošinātu to, ka jaunās zināšanas nodrošina zināšanu sistēmas pieaugumu un negatīvi neietekmē iepriekšējās zināšanas;
- lai pievienotu jaunas zināšanas, ekspertam ir jāidentificē pazīmes katram piemēram, kas to atšķir no citiem piemēriem, kuri ir iegūti iepriekš.

Fakts, ka likumi nekad netiek dzēsti vai pārveidoti, tikai papildināti, nodrošina vienkāršu likumu bāzes saskanības nodrošināšanu un pārvaldīšanu [117]. *Ripple down* likumus veido interaktīvi, un tie tiek formulēti šādi:

```
IF nosacījums THEN secinājums [BECAUSE piemērs] EXCEPT
IF ....
        ELSE IF.....
```

Pati likumu veidošana notiek pakāpeniski. Sākotnēji *RDR* veido noklusējuma (standarta) likumu, piemēram:

```
IF A THEN 1 | likums 0
```

Ja atklātos gadījums, kurā šis nosacījums nav patiess, ja spēkā ir B, tad likumu bāze būtu jāatjauno. *RDR* gadījumā netiek mainīti esošie likumi, bet gan radīti izņēmumi, kuri apmierina jaunos nosacījumus. Līdz ar to likumu bāzi var papildināt šādi:

```
IF A THEN 1 EXCEPT
IF B THEN 0 | likums 1
```

Ja atklājas jauni nosacījumi, pie kuriem jāparedz vērtība "1", bet nav spēkā A, tos iekļauj likumu kopā ar ELSE palīdzību. Piemēram:

IF A THEN 1 EXCEPT
IF B THEN 0
ELSE IF C AND D THEN 1 | likums 2

To, kādus nosacījumus pievienot likumiem, nosaka lietotājs-eksperts, bet datorsistēma viņam šajā procesā var palīdzēt, analizējot saglabātos piemērus un sakarības tajos. Šīs metodes ietvaros neatņemama nozīme ir ekspertam, jo tieši viņš definē visus nepieciešamos atribūtus likumam, pirms šo likumu pievieno likumu bāzei.

RDR pieejai ir arī savi trūkumi. Tā kā šī pieeja pilnībā balstās uz eksperta ieteiktiem likumiem, tad tā prasa vairāk lietotāja laika kā automātiskās vai daļēji automātiskās metodes. Cits aspekts ir izveidotās likumu kopas caurskatāmība un kompaktums. Ļoti svarīga ir pirmā likuma izvēle; no tā lielā mērā būs atkarīgs turpmākais likumu un izņēmumu izvērsums. Ja eksperts nebūs izvēlējies labu sākuma likumu, tālākā likumu struktūra kļūs grūti uzskatāma, prasīs daudz izņēmumu, no kuriem varētu izvairīties, izvēloties citu sākuma likumu. Tomēr balstoties uz pieejas pamatprincipu – nevienu likumu nedzēst, tikai veidot specifiskākus izņēmumus – atkāpšanās atpakaļ vairs nav iespējama. Tas noved arī pie atkārtotām zināšanām dažādās likumu bāzes vietās [117].

Gan aktīvās mācīšanās, gan *RDR* pieeja ir nedaudz plašāk apkopota bakalaura darbā [64].

Darba turpinājumā tiks apskatīta klasifikācijas sistēmu izstrāde no to projektēšanas un uzbūves puses.

2.3. Klasifikācijas sistēmu arhitektūra

Šī apakšnodaļa apkopo literatūrā aprakstītās klasifikācijas sistēmu arhitektūras, nepretendējot uz pilnīgu pārskatu, bet orientējoties uz kopīgo un atšķirīgo iezīmju identificēšanu, kas ļautu iegūt nepieciešamo pamatu interaktīvas klasifikācijas sistēmas izstrādei. Izklāstītos rezultātus darba autore pirmo reizi ir publicējusi [118].

Ar sistēmas arhitektūru parasti saprot sistēmas komponentes un to savstarpējo saistību, kā arī sistēmas funkcionēšanas shēmu. Klasifikācijas sistēmu kontekstā sistēmas arhitektūra literatūrā ir traktēta arī kā sistēmas projektēšanas posmi un dažādu iespējamo alternatīvu analīze sistēmas projektēšanas gaitā. Darbā ir apskatīti abi šie aspekti, tos nodalot un atsevišķi apskatot klasifikācijas sistēmas arhitektūru kā (1) sistēmas projektēšanas posmus (izstrādes procesa aprakstu) un (2) sistēmas komponenšu un funkcionēšanas modeli (sistēmas uzbūvi).

Atkarībā no mašīnapmācības sistēmas lietošanas jomas un veida atšķiras šo sistēmu nosaukumi. Literatūrā ir sastopamas tēlu pazīšanas sistēmas [42], klasifikācijas sistēmas [57],

induktīvās apmācības sistēmas [119], induktīvās apmācības tehniku lietojumi [119] vai vienkārši apmācības sistēmas [4]. Tomēr visām šīm sistēmām ir kopīgi principi, kas tiek ievēroti izstrādes procesā [119], līdz ar to tās var aplūkot vienā kontekstā. Šīs sistēmas iekļaujas arī intelektuālu sistēmu kategorijā, jo par intelektuālu tiek dēvēta sistēma, kas „savas eksistences laikā mācās" [120]. Tātad uz klasifikācijas sistēmām ir attiecināmas arī intelektuālu sistēmu īpašības. Šajā darbā tiks lietots termins „klasifikācijas sistēma", ja vien apskatītās literatūras autori nav pamatojuši cita termina lietojumu.

Nākamajās divās darba sadaļās ir apkopotas dažādu autoru piedāvātās projektēšanas un sistēmu funkcionālās shēmas - klasifikācijas sistēmu projektēšanas cikli un sistēmu funkcionēšanas modeļi.

2.3.1. Klasifikācijas sistēmu projektēšana un uzbūve

Projektēšanas problēmu risināšanai ir sastopami daudzi projektēšanas tipi [121, 122]. Vispārīgā veidā projektēšanas uzdevumi tiek iedalīti trijās klasēs.

- **Rutīnas**. Iepriekš ir zināmi gan nepieciešamo zināšanu avoti, gan problēmu risināšanas stratēģijas.
- **Inovatīvā**. Iepriekš ir zināmi tikai zināšanu avoti. Šo uzdevumu var uzskatīt arī par esošo komponentu oriģinālu kombināciju.
- **Kreatīvā**. Nav zināmi ne zināšanu avoti, ne problēmu risināšanas stratēģijas, kā rezultātā tiek iegūts nozīmīgs jauninājums vai pilnīgi jauns produkts.

Pastāv arī iedalījums četrās klasēs [123], papildu izdalot pārprojektēšanu (ang.v. - *redesign*). Iedalījums šajās klasēs ir diezgan vienkāršots [121]. Tuvāk realitātei būtu sistēmas novērtējums koordinātu telpā, kur uz vienas ass ir attēlota pāreja no rutīnas uz kreatīvu projektēšanu un uz otras ass – virzība no konceptuālas uz parametrisku projektēšanu.

Eksistē arī citi rutīnas, inovatīvās un kreatīvās projektēšanas raksturojumi, un ir skaidrs, ka robežas starp šiem veidiem nav tik viegli nosakāmas. Rozenmans un Gero [124] rutīnas projektēšanu definē kā iepriekš jau zināma tipa sistēmas realizāciju, atlasot elementus konkrētajai situācijai, izvēloties mainīgo vērtības no zināma apgabala. Inovatīvā projektēšana rada kādas sistēmas jaunu apakštipu, savukārt kreatīvā projektēšana rada jauna tipa izgudrojumu.

Induktīvās apmācības klasifikācijas sistēmas izstrāde vairāk atbilst rutīnas projektēšanai, savukārt interaktīvas uz induktīvo apmācību balstītas klasifikācijas sistēmas izveidošana, kur cilvēks tiktu iesaistīts konkrētas apmācības problēmas risināšanā, jau būtu inovatīvas projektēšanas rezultāts.

Pētījumā apskatītās sistēmu projektēšanas pieejas pilnā apjomā ir raksturotas darba 7. pielikumā, darba pamattekstā atspoguļojot tikai iegūtos secinājumus. Izpētītās klasifikācijas sistēmu projektēšanas pieejas apkopo Čerkaskija un Muliera [80], Dovdija un Verdena [125], Mičela [4], Dudas un Hārta [42], Teodora un Kotrumbas [57], Vardeniusa un Somerena [119], Brodleja un Smita [126], Bielavska un Levanda [127] un *Lielbritānijas Tirdzniecības un industrijas departamenta* [128] veikumu.

Ja projektēšana apraksta, kā izveidot klasifikācijas sistēmu un kādi lēmumi jāpieņem šī procesa gaitā, tad klasifikācijas sistēmas uzbūve raksturo, kā sistēma funkcionē un no kādām komponentēm sastāv. Konkrētie apskatītie klasifikācijas sistēmu uzbūves piemēri ir aprakstīti darba 8. pielikumā. Izpētītās pieejas apkopo Hana un Kembera [129], Čerkaskija un Muliera [80], Mičela [4], Dudas un Hārta [42] dažāda abstrakcijas līmeņa piedāvātos modeļus.

2.3.2. Esošo klasifikācijas sistēmu arhitektūru analīze

Izdarot secinājumus par apkopotajām klasifikācija sistēmu arhitektūrām, klasifikācijas sistēmu projektēšana un uzbūve literatūrā ir diezgan plaši apskatīta. Atšķirības starp aprakstītajām arhitektūrām galvenokārt nosaka fokuss, risinājuma mērogs un lietojuma sfēra. Neskatoties uz atšķirībām, var secināt, ka starp dažādajām arhitektūrām nepastāv būtiskas pretrunas.

Sistēmu projektēšanas procesā lielākajai daļai apskatīto darbu ir atrodamas vairākas kopīgas iezīmes – tās vienotajā modeļu atspoguļojumā arī ir iekrāsotas dzeltenā un zaļā krāsā (skat. 7. pielikumu).

Pirmkārt, lielākā daļa modeļu pievērš būtisku nozīmi **sistēmas izstrādes sākotnējam posmam**. Tas tiek saukts vai nu par *problēmas nostādni*, *lietojuma identificēšanu*, *problēmsfērai specifisko faktoru analīzi*, *problēmas identificēšanu* vai kā savādāk, bet nemainīgi tiek uzsvērts, ka problēmsfēras izpēte un problēmas formulēšana ir ļoti būtisks posms pirms jebkādas apmācības tehniku lietošanas. Saprātīga problēmas nostādne un ieejas dati veido pamatu visai tālākajai sistēmas izstrādei. Ja nopietnas kļūdas būs šajos posmos, viss tālākais darbs nesniegs vajadzīgos rezultātus. Dažādos klasifikācijas sistēmu projektēšanas modeļos šim uzdevuma ir veltīts viens vai vairāki posmi, dažviet tam piešķirot vislielāko lomu visā projektēšanas gaitā.

Otrkārt, **sistēmas projektēšana ietver labākā risinājuma meklēšanu.** Tas var tikt darīts dažādos veidos – analītiski, eksperimentējot vai radot sistēmas prototipu. Klasifikācijas sistēmas veidošana konkrētai problēmsfērai reti kad ir vienvirziena programmatūras izstrādes process. Klasifikācijas sistēmas izstrādes laikā ir jāizdara daudz izvēļu, piemēram, jāizvēlas

apmācības metožu klase, konkrēts klasifikācijas algoritms, algoritma parametri. Uz šo nepieciešamību norāda arī atgriezeniskās saites daļai no apskatītajiem projektēšanas modeļiem.

Apmācības sistēmu uzbūve literatūrā ir aprakstīta retāk un sniegta abstraktu komponenšu veidā; granularitātes līmenis šiem sistēmas modeļiem ir diezgan augsts. No tā var secināt, ka detalizētākas klasifikācijas sistēmu arhitektūras ir specifiskas katram lietojumam, ar ierobežotu atkārtotu izmantojamību, līdz ar to maz aprakstītas zinātniskajā literatūrā.

2.4. Nodaļas kopsavilkums

Šajā nodaļā tika apskatīti saistītie darbi par tām tēmām, kuru izpēte ir būtiska promocijas darbā piedāvātā risinājuma izstrādei. Šīs tēmas ir (1) klasifikācijas uzdevums mašīnapmācībā, (2) interaktīvie risinājumi induktīvajā apmācībā un (3) klasifikācijas sistēmu arhitektūra no to uzbūves un projektēšanas puses.

Klasifikācija ir būtiska daudzos problēmu risināšanas uzdevumos. Klasifikācijas uzdevumi atšķiras pēc objektiem piešķiramo kategoriju skaita, ko nosaka problēmsfēra. Visbiežāk ir nepieciešams noteikt objekta piederību tikai vienai klasei. Tas ir, katrs piemērs ir saistīts vienu klasi k no nepārklājošos klašu kopas K, $|K| > 1$. Tomēr ir arī sfēras, kurās objekti var piederēt vienlaicīgi vairākām klasēm. Tādā gadījumā runa ir par daudzkategoriju klasifikāciju, un katrs piemērs ir saistīts ar klašu apakškopu $Y \subseteq K$. Šajā promocijas darbā **uzsvars ir likts uz daudzkategoriju klasifikācijas uzdevumu**, jo galveno darbā risināmo problēmsfēru – studiju priekšmetu salīdzināšanu – raksturo iespēja, ka viens mācību priekšmets ir līdzīgs vai atbilstošs vairākiem citiem.

Klasifikācijai gan vienas kategorijas, gan daudzkategoriju gadījumā, ir izmantojams plašs metožu loks, tomēr induktīvās apmācības algoritmiem ir priekšrocības tajās sistēmās, kur nepieciešams tālāk apstrādāt iegūtos spriedumus un izprast lēmuma pieņemšanas ceļu. Daudzas problēmas, ar kurām jāsaskaras induktīvās apmācības pielietošanas gaitā, tiek sekmīgi risinātas, piemēram, datu pirmapstrāde. Tomēr **jaunu piemēru klasificēšanā ir trūkumi, kuri pagaidām nav novērsti.** Piemēram, metodes klasifikācijas nodrošināšanai gadījumos, kad klasifikatora likumu bāze nespēj nodrošināt jauna piemēra klasificēšanu, nav pilnīgas un piemērotas visām dzīves situācijām. Lai risinātu šo problēmu, algoritmos *AQ* un *CN2* tika piedāvāts lietot noklusēto likumu, kurš piešķir apmācības kopā visizplatītāko klasi tiem piemēriem, kurus klasifikators nav varējis klasificēt [83]. Lai arī metode ir vienkārša un labi pamatojama, ne vienmēr tā strādā pieņemami. Piemēram, situācijā, kad klašu ir daudz un tās visas parādās caurmērā vienlīdz bieži, šāda pieeja nedod labu klasifikācijas rezultātu. Tādēļ var secināt, ka promocijas darba mērķis - eksperta iesaistīšana neklasificēto un nepārliecinoši klasificēto piemēru klases piederības

noteikšanai - ir jauns un potenciāli lietderīgs ieviesums klasifikācijas rezultātu uzlabošanā, tādējādi arī paplašinot induktīvās apmācības lietošanas iespējas jaunās jomās.

Pirms piedāvāt jaunu interaktīvu risinājumu induktīvajā apmācībā, ir apkopotas literatūrā minētās **līdzšinējās pieejas interaktīvai induktīvajai apmācībai**, kā arī citām klasifikācijas metodēm. Apkopojuma rezultātā ir izstrādāta shēma, kurā izdalītas tās fāzes klasifikatora veidošanā un lietošanā, kurās saskarsme ar ekspertu, saskaņā ar dažādām pieejām, ir nepieciešama vai vēlama. Klasifikatora veidošanas posmā izdalīta apmācības datu sagatavošana un atlase (A) un izveidoto likumu atlase un apstrāde (B). Klasifikatora lietošanas posmā izdalīta jauno klasificējamo piemēru apstrāde (C) un pieņemtā lēmuma apstrāde (D). Dažādās līdz šim aprakstītajās sistēmās sadarbība ar lietotāju izpaužas posmā A vai gan posmā A, gan B, vai arī posmā D, bet ne posmā C. Esošās lietotāja iesaistes notiek vai nu par ātru, vai par vēlu, kad lēmums jau ir pieņemts. Apskatīti arī speciāli interaktīvas klasifikācijas gadījumi – aktīvā mācīšanās un *Ripple down* likumi.

Izvērtējot esošo sistēmu trūkumus un konstatējot to neatbilstību promocijas darbā risināmajai problēmai, tiek **apstiprināta nepieciešamība pēc jauna tipa interaktīvas uz induktīvo apmācību balstītas klasifikācijas sistēmas izveides**, lai risinātu neklasificētu piemēru problēmu.

Lai izstrādātu interaktīvas klasifikācijas sistēmas arhitektūru, vispirms ir **apskatītas dažādas esošās klasifikācijas sistēmu arhitektūras** – to uzbūve un projektēšanas posmi. Atšķirības starp aprakstītajām arhitektūrām galvenokārt nosaka fokuss, risinājuma mērogs un pielietojuma sfēra. Detalizētāka klasifikācijas sistēmas uzbūve ir specifiska katram lietojumam, ar mazu atkārtotu izmantojamību, kas ļauj secināt, ka arī interaktīvas sistēmas uzbūve būs unikāla un veidojama no jauna. Savādāk ir ar pašu sistēmas projektēšanas procesu; lielākā daļa no 9 apskatītajiem projektēšanas modeļiem satur kopīgas iezīmes – uzsverot sākotnējo problēmsfēras izpētes nozīmību un labākā risinājuma meklēšanu kā daļu no projektēšanas aktivitātēm. Tādējādi sistēmas projektēšanas modeļi ir atkārtoti izmantojami, un interaktīvas klasifikācijas sistēmas arhitektūrai netiks radīts jauns projektēšanas cikls, bet lietots kāds no jau līdz šim aprakstītajiem un praksē pārbaudītajiem.

Balstoties uz veikto literatūras analīzi un konstatētajiem pastāvošo pieeju trūkumiem, nākošajā nodaļā tiks aprakstīts piedāvātais interaktīvas uz induktīvo apmācību balstītas klasifikācijas sistēmas (angļu valodā - ***In****teractive Inductive Learnign based* ***Cla****ssification* ***S****ystem* un turpmāk lietojot saīsinājumā - *InClaS*) pamatmodelis un tā komponentes.

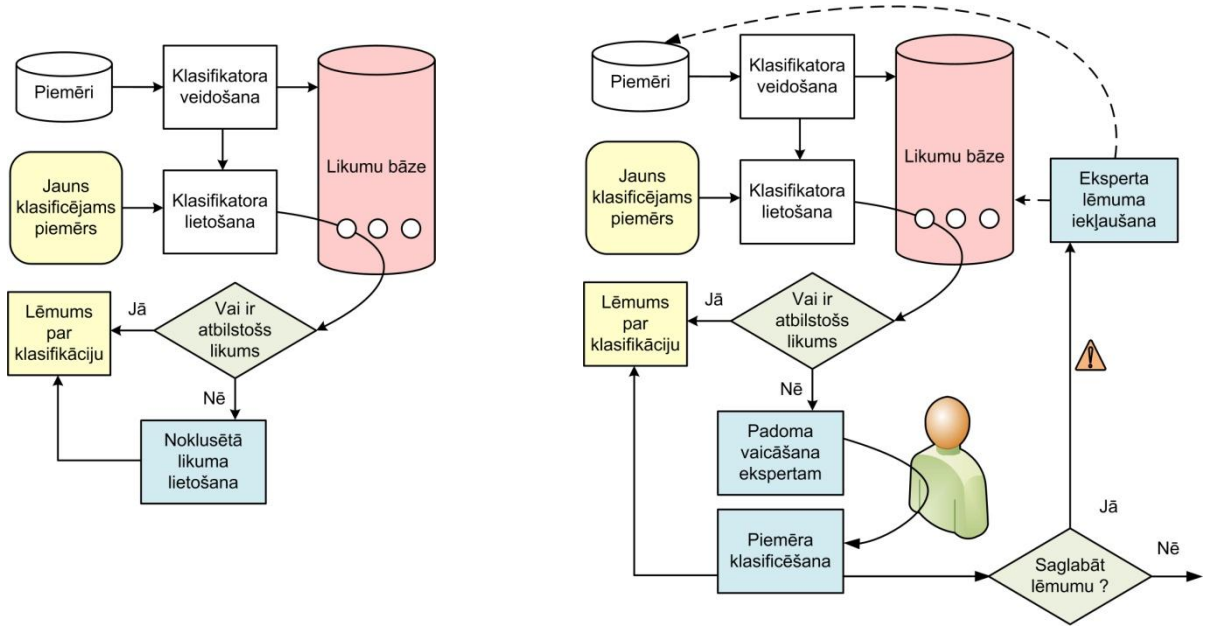
3. INTERAKTĪVAS UZ INDUKTĪVO APMĀCĪBU BALSTĪTAS KLASIFIKĀCIJAS SISTĒMAS (*INCLAS*) PAMATMODELIS

Šajā nodaļā tiks izklāstīta promocijas darba autores izstrādātā interaktīvas klasifikācijas pieeja, kura iesaista ekspertu klasifikatoram neskaidro piemēru klases piederības noteikšanā. Vispirms tiks izskaidrots, ar ko šī pieeja atšķiras no citām interaktīvajām klasifikācijas pieejām un kā klasifikācijas shēma izmainās attiecībā pret noklusētā likuma lietošanu klasifikācijā. Tam veltīta 3.1. apakšnodaļa. 3.2. apakšnodaļa sniedz interaktīvās klasifikācijas sistēmas projektēšanas soļu un uzbūves aprakstu, balstoties uz 2.3. apakšnodaļā veikto klasifikācijas sistēmu arhitektūru apskatu. 3.3. apakšnodaļā tiek skaidrots, kādi piemēri tiek saukti par klasifikatoram neskaidriem un tiek nodoti eksperta klasifikācijai. 3.4. apakšnodaļā ir analizēti jautājumi, kas saistīti ar eksperta sniegto zināšanu iekļaušanu klasifikatorā, jo interaktīvā sistēma mācās un papildina savu likumu bāzi, tādējādi pilnveidojot klasifikatoru. Izklāstītās komponentes veido elementus kopējā interaktīvās klasifikācijas sistēmas *InClaS* modelī, kurš apkopots 3.5. apakšnodaļā.

3.1. Izstrādātā interaktīvā klasifikācijas pieeja līdzšinējo metožu kontekstā

Esošo interaktīvo klasifikācijas pieeju analīze darba 2.2. apakšnodaļā liecina, ka neviena no apskatītajām interaktīvajām pieejām pilnībā nepiedāvā tādu interaktivitātes shēmu, kas ļautu risināt neklasificēto piemēru problēmu. Vistuvākais virziens problēmas nostādnē izvirzītajam interaktivitātes modelim no 2.2. apakšnodaļā apskatītajām interaktīvajām pieejām ir aktīvā mācīšanās. Būtībā promocijas darbā piedāvāto interaktivitātes pieeju var uzskatīt par aktīvās mācīšanās paveidu. Kopīgs šīm pieejām ir tas, ka eksperts klasificē *kādā ziņā* vērtīgu neklasificētu piemēru un sistēma to izmanto klasifikatora papildināšanai. Atšķirību nosaka apstākļi, ka aktīvās mācīšanās gadījumā ekspertam dodamo piemēru izvēlas algoritms no plaša piemēru skaita, kam vēl nav noteikta klasifikācija, lai mērķtiecīgi uzlabotu savu klasifikatoru, savukārt interaktīvā klasifikatora gadījumā piemēru neizvēlas algoritms, un to vispirms tiek mēģināts klasificēt paša klasifikatora spēkiem. Teorētiski ir iespējams abas pieejas arī apvienot, taču ne visās problēmsfērās izpildās aktīvās mācīšanās otrais nosacījums, t.i., „piemēri bez klasifikācijas ir brīvi pieejami vai viegli iegūstami". Tā tas ir arī risināmajā problēmsfērā - studiju priekšmetu salīdzināšanā. Turklāt neklasificēto piemēru noteikšanu un interaktivitāti ar lietotāju ir nepieciešams integrēt klasifikācijas sistēmas patstāvīgas izmantošanas posmā, ne tikai apmācības laikā.

Šajā darbā piedāvāta sistēma iesaista ekspertu tad, ja jaunais klasificējamais piemērs nevar tikt klasificēts, balstoties uz klasifikatora likumu bāzi, tādējādi aizstājot noklusētā likuma lietošanu standarta klasifikācijas pieejās. 3.1. attēlā redzams uz induktīvo apmācību balstīta klasifikatora darbības cikls noklusētā likuma lietošanas gadījumā (attēla A daļa) un piedāvātajā interaktīvajā pieejā (attēla B daļa).



A **B**

3.1. att. Klasifikācija ar noklusēto likumu (A) un eksperta iesaistīšanu (B)

Piedāvātā interaktīvā pieeja cenšas nodrošināt sistēmas neatkarību, jo:

- klasifikācijas sistēmas darbs nav atkarīgs no eksperta, principā tā var strādāt patstāvīgi;
- eksperts nav piesaistīts sistēmai, jo sistēmas darbā nav jāiesaistās sistemātiski.

Galvenās jaunās pieejas priekšrocības neklasificēto piemēru gadījumā, salīdzinot ar citām 2.2. apakšnodaļā analizētajām interaktīvajām pieejām, ir šādas:

- eksperts netiek apgrūtināts ar pārāk biežu vēršanos pie viņa, kā tas ir pieejās, kur apmācība notiek tikai ar eksperta iesaistīšanos;
- palīdzība tiek meklēta tieši tad, kad tā ir nepieciešama - kad klasifikatoram neskaidrais piemērs ir jāklasificē - un tādējādi netiek nelietderīgi tērēts eksperta laiks.

Interaktīvā pieeja (3.1. B. att.) paredz, ka eksperta zināšanas var tikt izmantotas ne tikai piemēra klasifikācijas noteikšanai. Eksperta lēmums par piemēra klasi var tikt izmantots arī klasifikatora papildināšanai un uzlabošanai, ja eksperts vēlas, lai tas tiktu darīts. Tomēr, ja eksperta sniegtās zināšanas tiek izmantotas likumu bāzes atjaunošanai, ir jārūpējas par

klasifikatora nepretrunību un zināšanu saskanību starp esošajiem likumiem un izmaiņām, kas tiek veiktas jauno zināšanu ietekmē. Ir vairāki veidi, kā to nodrošināt. Šīs iespējas promocijas darba autore ir analizējusi publikācijā [91], un par to tiks runāts arī darba 3.4. apakšnodaļā.

3.2. Interaktīvas klasifikācijas sistēmas arhitektūra

Apkopojot secinājumus no klasifikācijas sistēmu arhitektūru apskata (2.3. apakšnodaļā), galvenās atziņas, kas tiks izmantotas interaktīvas klasifikācijas sistēmas izstrādē, ir šādas:

- liela nozīme ir sistēmas izstrādes sākotnējam posmam - problēmas nostādnei, lietojuma identificēšanai un līdzīgi nosauktiem posmiem ar šo pašu funkciju;
- sistēmas projektēšana ietver labākā risinājuma meklēšanu – analītiski, eksperimentējot vai radot sistēmas prototipu;
- detalizēta klasifikācijas sistēmas arhitektūra ir specifiska katram lietojumam.

3.2.1. Interaktīvas klasifikācijas sistēmas projektēšana

Interaktīvas klasifikācijas sistēmas projektēšanas process savā būtībā neatšķiras no tādas klasifikācijas sistēmas projektēšanas, kura nav interaktīva. Sistēmas izveides soļi nav atkarīgi no šī jauninājuma. Atšķirības parādās prasībās, ierobežojumos un pieņemtajos lēmumos. Gandrīz jebkurš no 2.3.1. sadaļā pieminētajiem un darba 7. pielikumā apskatītajiem sistēmu projektēšanas modeļiem ir derīgs interaktīvas sistēmas projektēšanā, tomēr ir izvēlēta Bielavski un Levanda piedāvātā piecu soļu procedūra intelektuālu sistēmu projektēšanai [127]. Šī procedūra izvēlēta vairāku iemeslu dēļ.

1. Tā neuzliek pārāk stingrus ierobežojumus projektējamai sistēmai un nenosaka ierobežojumus lietojuma sfērai, kas šajā situācijā ir svarīgi, jo vispirms tiek izstrādāta vispārīga arhitektūra, kura vēlāk tiks detalizēta konkrētam sistēmas lietojumam.
2. Tā ir vienkārša, vispārīga, bet aptver visus nepieciešamos soļus.
3. Tā iekļauj aspektus, kas atzīti par kopīgiem daudzām klasifikācijas sistēmu projektēšanas pieejām, respektīvi, problēmas analīzi un labākā risinājuma meklēšanu (šajā gadījumā – prototipējot).

Projektēšanas soļi, atbilstoši [127] sniegtajai vispārīgajai procedūrai, ir analizēti 3.1. tabulā. Jāņem vērā, ka sistēmas projektēšana parasti jāveic, rēķinoties ar konkrēto tās lietošanas jomu, tādēļ vispārīgas interaktīvās sistēmas projektēšanas gadījumā projektēšanas arī soļus iespējams norādīt tikai vispārīgi. Katrā no soļiem var būt citas papildu aktivitātes, ja tās ir nepieciešamas konkrētajā problēmsfērā vai tādēļ, lai sistēma iegūtu vairāk funkcionalitātes.

Vispārīgas projektēšanas galvenā nozīme ir iespējama norādīt jautājumus, par kuriem jādomā, un lēmumus, kas jāpieņem sistēmas ieviešanas laikā reālā problēmsfērā. Interaktīvas klasifikācijas sistēmas projektēšana konkrētai problēmsfērai - studiju priekšmetu salīdzināšanai - atbilstoši šai procedūrai tiks detalizēta darba 4.4. apakšnodaļā.

3.1. tabula

Interaktīvas klasifikācijas sistēmas projektēšanas soļi

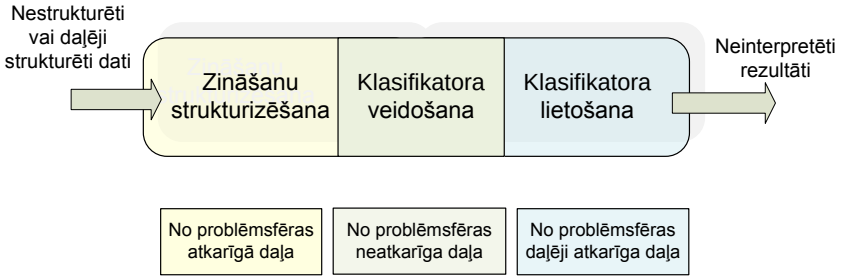
Intelektuālas sistēmas projektēšanas soļi [127]	**Soļa interpretācija interaktīvas klasifikācijas sistēmas izstrādē**
1. Problēmas identificēšana	Problēmas identificēšana ir cieši saistīta un daļēji pārklājas ar nākamo soli. Problēmsfērā ir jānosaka, par kādiem procesiem vai objektiem apmācība būtu lietderīga un varētu tikt veikta, jāidentificē, kas tiks klasificēts un kādi parametri spēj šo klasifikāciju nodrošināt.
2. Zināšanu iegūšana un attēlošana	Šis solis ietver dažādus aspektus. Par būtiskiem atzītos atribūtus no problēmsfēras ir jāvar iegūt un apkopot. Induktīvās apmācības algoritmi savās ieejās vispārīgā gadījumā saprot tikai tabulas tipa datus, kur ir definēti ieraksti ar atribūtiem un to vērtībām, kā arī klasēm (ja notiek apmācība). Reālās problēmsfērās bieži vien informācija ir strukturēta sarežģītākās formās, piemēram, nestrukturēta vai daļēji strukturēta teksta veidā, attēlos, grafos. Lai šādus datus varētu izmantot induktīvās apmācības algoritmi, ir jāveic dažādas priekšapstrādes, piemēram, būtisko datu nošķiršana no fona, datu pārveidošana u.c. darbības.
3. Rīku izvēle	Šajā gadījumā par rīkiem nav jāuzskata tikai programmatūras produkti, ar kuru palīdzību realizēt sistēmu, bet arī paši klasifikācijas algoritmi, kuri tiks izvēlēti klasifikatora veidošanai un jaunu piemēru klasificēšanai. Tā kā neeksistē no problēmsfēras neatkarīgi kritēriji, kas noteiktu, kura metode vai algoritms būtu jāizmanto, tad šis uzdevums ir pietiekami sarežģīts un atbildīgs, tādēļ var nākties to atkārtot vairakkārt.
4. Prototipēšana un izstrāde	Prototipēšana daļēji sākas jau pirmajā solī, kad tiek atlasīti problēmu raksturojošie atribūti. Lai paustu lielāku pārliecību atribūtu izvēlē, jau laicīgi ir jāveic eksperimenti, kas apstiprina vai noliedz izvēlēto atribūtu spēju raksturot un nošķirt dažādu klašu pārstāvjus. Šādā veidā tiek iegūts klasifikatora prototips. Induktīvās apmācības kontekstā „izstrāde" ir konkrēta algoritma (algoritmu) implementācija, datu kopas sadalīšana apmācības un testa kopās un apmācības veikšana klasifikatora iegūšanai.
5. Testēšana un uzturēšana	Testēšanas posmā tiek noteikta klasifikācijas sistēmas precizitāte (vai citi parametri), balstoties uz testa datu kopu, kurā objektiem ir zināma klasifikācija, bet kas netika izmantota apmācības laikā. Tas var tikt uzskatīts par izstrādes posma beigām. Uzturēšanas un ekspluatācijas laikā sistēmai ir jābūt spējīgai klasificēt jaunus objektus un, ja tas ir paredzēts, arī papildināt esošo klasifikatoru ar jaunām zināšanām.

3.2.2. Interaktīvas klasifikācijas sistēmas uzbūve

Šī sadaļa apraksta interaktīvas klasifikācijas sistēmas uzbūvi no dažādiem skatu punktiem.

Galvenās interaktīvas klasifikācijas sistēmas sastāvdaļas

Galvenie uzdevumi, kas jāveic klasifikācijas sistēmā ar sarežģītiem datiem (ang. v. – *complex data types*), kuri pieprasa atbilstošu priekšapstrādi un strukturizēšanu, ir parādīti 3.2. attēlā. Dažādiem posmiem ir atšķirīgs neatkarības līmenis, tas ir, dažu uzdevumu realizācija ir atkarīga no konkrētas problēmsfēras, kurā klasifikators tiks ieviests, kamēr citos posmos iespējams definēt plašāk izmantojamu risinājumu.



3.2. att. Galvenie posmi klasifikācijas procesā nestrukturētiem vai daļēji strukturētiem ieejas datiem

Zināšanu strukturizēšana no sistēmas izstrādes viedokļa ir no problēmsfēras atkarīgs uzdevums. Metodes, kas nepieciešamas datu sagatavošanai to tālākai lietošanai klasifikācijā, ir atkarīgas no konkrētajiem datiem un to sākotnējā atspoguļojuma. Datu glabāšanas struktūrām var būt nepieciešama specifiska priekšapstrāde un atribūtu izgūšana.

Klasifikatora veidošanu var uzskatīt par no problēmsfēras relatīvi neatkarīgu uzdevumu un to var specificēt jau vispārīgā sistēmas arhitektūrā. Klasifikatoru veidošanas principi ir daudz aprakstīti literatūrā un praktiski plaši lietoti. Tomēr konkrētas apmācības pieejas, metodes un parametru izvēle ir cieši sasaistīta ar izmantojamajiem datiem.

Klasifikatora lietošana ir no problēmsfēras daļēji atkarīgs uzdevums. Tehniskie aspekti klasifikācijā var tikt uzskatīti par universāliem visām problēmsfērām, bet rezultātu atspoguļošanas formātu ietekmē sākotnējie dati un tālākās rezultātu izmantošanas un apstrādes vajadzības, piemēram, uzskatāmība, demonstrējot rezultātus lietotājam. Klasifikācijas rezultāts ir neinterpretēti rezultāti, kuri, piemēram, var tikt padoti augstākstāvošai lēmumu atbalsta sistēmai, kas nodrošinātu problēmsfēras specifisku interpretāciju vai tālāku datu apstrādi.

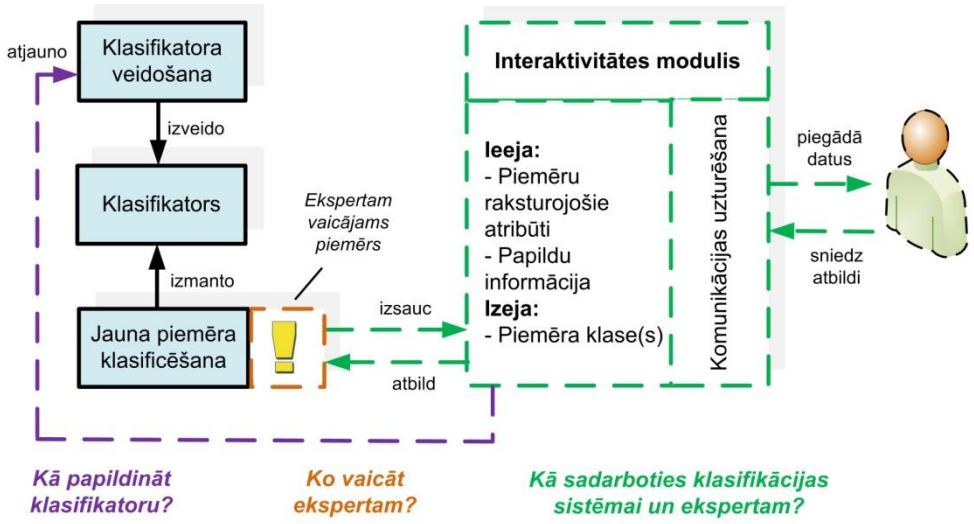
Ņemot vērā iepriekš minēto, var secināt, ka sākotnējo datu apstrādes specificēšanu var veikt tikai saskaņā ar konkrētu problēmsfēru, bet daļu lēmumu par klasifikatora veidošanu un

lietošanu var pieņemt jau iepriekš. No problēmsfēras mazāk atkarīgās daļas var projektēt agrākos sistēmas arhitektūras izstrādes posmos nekā atkarīgās.

Vispārējā ieviešamā interaktivitātes shēma

Interaktivitātei ir jāatspoguļojas sistēmas arhitektūrā. Interaktīvas klasifikācijas sistēmas mērķis nav kāda konkrēta induktīvās apmācības algoritma uzlabošana, bet gan klasifikācijas iespēju paplašinājuma izstrāde, kas ļautu lietot interaktīvu pieeju visiem tiem induktīvās apmācības algoritmiem, kuros nav definēti citi mehānismi neklasificēto piemēru klasificēšanai (piemēram, noklusētais likums), vai šie mehānismi var tikt aizstāti, neizdarot lielas izmaiņas pašā algoritmā. Šī pieeja klasifikācijas procesā maina veidu, kā klasifikators tiek lietots jaunu piemēru klases piederības noteikšanai, nevis to, kā klasifikators tiek veidots.

Nepieciešamo izmaiņu apjoms interaktīvā klasifikācijas sistēmā, salīdzinājumā ar „klasisko" neinteraktīvo pieeju, ir atkarīgs no apmācības algoritma un tā implementācijas. Ja informācija par neklasificētajiem piemēriem ir pieejama uzreiz pēc mēģinājuma tos klasificēt, neklasificēto piemēru apstrāde var tikt pievienota kā ārējs paplašinājums, bez nepieciešamības modificēt sākotnējo klasifikācijas procesu. Pretējā gadījumā jaunu piemēru klasifikācijas procedūra ir jāpapildina ar iespēju izsekot neklasificētos piemērus. 3.3. attēlā ir parādīts, kā interaktivitātes modulis papildina tradicionālo klasifikācijas procesu, iegūstot interaktīvas uz induktīvo apmācību balstītas klasifikācijas sistēmas, saīsināti ***InClaS*** (ang. v. - ***In****teractive Inductive Learning Based* ***Cla****ssification* ***S****ystem*), darbības vispārēju shēmu.



3.3. att. Interaktivitātes iekļaušana vispārīgajā klasifikācijas modelī

3.3. attēlā redzami klasifikācijas sistēmas pamatelementi un saites (ar nepārtrauktu līniju), kā arī elementi un saites, kas nodrošina interaktivitātes ieviešanu (ar raustītu līniju) neklasificēto piemēru klases piederības noteikšanai. Paplašinājums iekļauj šādas funkcijas:

1. klasifikācijas posmā tiek "pārtverts" piemērs, kuram klasifikators nespēj noteikt klasi;
2. klasifikatoram neskaidrais piemērs, kā arī papildu informācija, ja tāda ir, tiek sniegta ekspertam;
3. eksperta sniegtā atbilde tiek apstrādāta;
4. balstoties uz jaunajām zināšanām, kas saņemtas no eksperta, tiek atjaunots klasifikators.

Iepriekš 1.1. sadaļā tika noskaidroti galvenie jautājumi, uz kuriem ir jāatbild interaktīvas klasifikācijas sistēmas izveidošanā, proti, "Kā noskaidrot, ko vaicāt lietotājam-ekspertam?" un "Kā sadarboties klasifikācijas sistēmai ar tās lietotāju?". Sīkāk analizējot aspektus interaktīvas klasifikācijas sistēmas izstrādē, risināmo jautājumu loks tiek detalizēts un paplašināts. 3.3. attēls parāda jaunu jautājumu – „Kā papildināt klasifikatoru?", kas praktiskā veidā saista abus sākotnējos jautājumus. Tātad interaktivitātes ieviešanai klasifikācijas sistēmā tiek apskatīti šādi aspekti:

- ekspertam vaicājamo piemēru „pārtveršana" klasifikatora lietošanas posmā. Lai to realizētu, svarīgi ir definēt, kādi piemēri tiek atzīti par klasifikatoram neskaidriem (tas tiks skaidrots 3.3. apakšnodaļā);
- klasifikatora atjaunošana pēc eksperta sniegtas klasifikācijas (ar to saistītie apsvērumi tiks apskatīti 3.4. apakšnodaļā);
- klasifikācijas sistēmas un eksperta sadarbības nodrošināšanu nosaka visa klasifikācijas sistēmas uzbūve (sistēmas arhitektūras specificēšana tiek atspoguļota šajā nodaļā kopumā).

Interaktīvas klasifikācijas sistēmas moduļi

Balstoties uz 3.3. attēlā redzamo interaktīvās klasifikācijas sistēmas vispārīgo shēmu un iepriekš 2.3. nodaļā sniegto klasifikācijas sistēmu arhitektūru apkopojumu un analīzi, projektējamajā sistēmā ir izdalītas komponentes, kas atbild par dažādām funkcijām. Sistēmai ir izvēlēta modulāra arhitektūra. Izvēli par labu modulārai arhitektūrai nosaka vairāki aspekti. Pirmkārt, moduļi savā starpā ir relatīvi neatkarīgi un izmaiņas viena moduļa iekšienē neskar citus moduļus, kā tas būtu integrētas arhitektūras gadījumā. Otrkārt, „Galvenās interaktīvas klasifikācijas sistēmas sastāvdaļas" sadaļā tika secināts, ka klasifikācijas sistēmai ir no problēmsfēras atkarīgas un neatkarīgas daļas, līdz ar to daļa moduļu var būt nemainīgi dažādām

lietošanas sfērām, kamēr citi tiek mainīti un pielāgoti konkrētām vajadzībām. Moduļu mērķis, funkcionalitāte un saistība ar citiem moduļiem ir sniegta 3.2. tabulā.

3.2. tabula

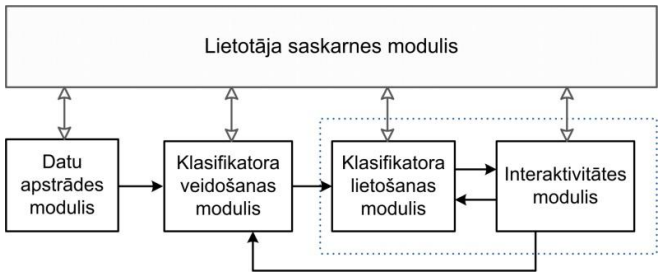
Interaktīvas klasifikācijas sistēmas moduļi

Lietotāja saskarnes modulis
Nodrošina lietotājam draudzīgu saskarni starp sistēmu un tās lietotāju: - Atspoguļo datus. - Nodrošina lietotāja pieprasījumu izpildi, izsaucot funkcijas cits sistēmas moduļos. Tieša saziņa ar citiem moduļiem: - Datu apstrādes modulis - Klasifikatora veidošanas modulis - Klasifikatora lietošanas modulis - Interaktivitātes modulis
Datu apstrādes modulis
Nodrošina datu attēlošanu dažādos formātos: - Nodrošina sistēmas lietotājam iespēju pievienot datus dažādos formātos, palīdzot strukturētu datu iegūšanā. - Nodrošina sistēmas lietotājam iespēju apskatīt apmācības datus un klasifikācijas likumus dažādos formātos. - Nodrošina datu transformāciju moduļu iekšējiem un savstarpējiem procesiem. Tieša saziņa ar citiem moduļiem: - Lietotāja saskarnes modulis - Klasifikatora veidošanas modulis - Klasifikatora lietošanas modulis - Interaktivitātes modulis
Klasifikatora veidošanas modulis
Ģenerē klasifikācijas modeli dotajai datu kopai. Klasifikators sistēmas iekšējā struktūrā tiek glabāts sistēmai specifiskā formātā. Ja apmācības algoritma attēlojuma forma ir likumi, tad no šī formāta var izgūt IF – THEN likumus. Šis modulis ir balstīts uz jau implementētu induktīvās apmācības algoritmu bibliotēku izmantošanu. Tieša saziņa ar citiem moduļiem: - Lietotāja saskarnes modulis - Datu apstrādes modulis
Klasifikatora lietošanas modulis
Izmanto iepriekš izveidotu klasifikatoru, lai noteiktu klašu piederību jauniem piemēriem, iegūst klasifikatora pārliecības par pieņemto lēmumu. Šis modulis izmanto gatavas implementētas apmācības metodes, kas ir papildinātas ar spēju pārtvert piemērus, kurus nodot izvērtēšanai ekspertam. Šajā gadījumā tiek izsaukts interaktivitātes modulis. Tieša saziņa ar citiem moduļiem: - Lietotāja saskarnes modulis - Datu apstrādes modulis - Interaktivitātes modulis
Interaktivitātes modulis
Nodrošina saziņu ar sistēmas lietotāju, sagatavojot un apstrādājot nepieciešamo informāciju. Cieši saistīts ar klasifikatora lietošanas moduli. - Nodrošina neskaidro piemēru un papildu informācijas sagatavošanu atspoguļošanai ekspertam, kā arī saņem

atbildi. Papildu informācija ir, piemēram, neklasificētajam piemēram vistuvākie likumi vai piemēru paplašināts apraksts.
- Iniciē piemēru bāzes atjaunošanu pēc eksperta atbildes saņemšanas.
- Nodrošina likuma atspoguļošanu pēc lietotāja pieprasījuma.

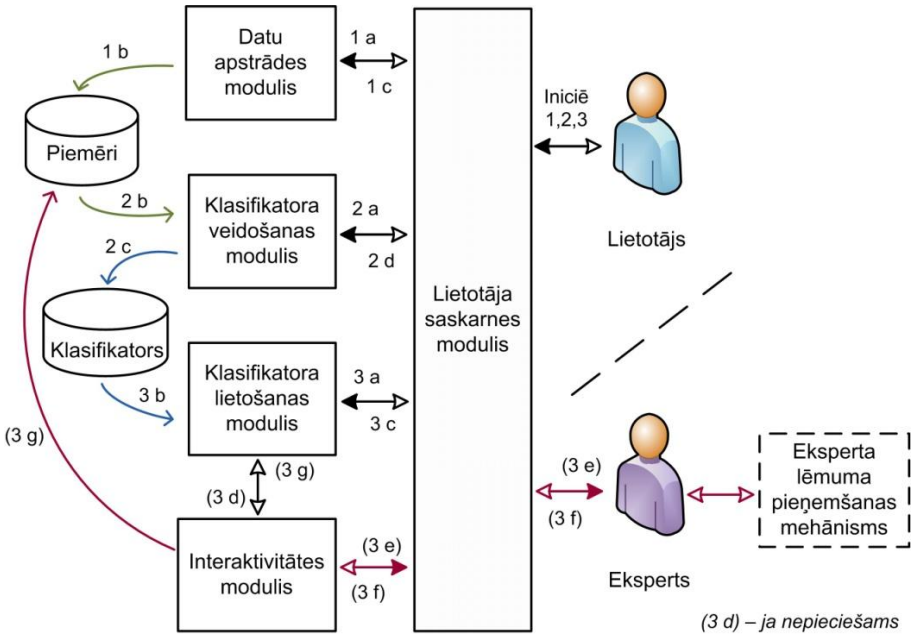
Tieša saziņa ar citiem moduļiem:
- Lietotāja saskarnes modulis
- Klasifikatora veidošanas modulis
- Datu apstrādes modulis

3.4. attēlā ir parādīti interaktīvas klasifikācijas sistēmas moduļi un galvenās saites starp tiem.



3.4. att. Interaktīvas klasifikācijas sistēmas moduļi un to loģiskā saistība

Fiziski moduļi savā starpā komunicē savādāk. Apmācības piemēri un klasifikators, piemēram, likumu formā, tiek glabāti atsevišķās datu bāzēs un konkrētas moduļu funkcijas aktivizē lietotājs. 3.5. attēls demonstrē datu plūsmas un procesu inicializāciju sistēmā, iekļaujot arī sistēmas lietotāju un ekspertu, kurš šajā gadījumā arī uz uzskatāms par funkcionējošas klasifikācijas sistēmas daļu.



3.5. att. Interaktīvas klasifikācijas sistēmas funkcionēšana

3.5. attēls atspoguļo tipisku sistēmas lietošanas scenāriju (bet ne vienīgo iespējamo), neaprakstot iekšējos procesus moduļos. Caur lietotāja saskarni sistēmas lietotājs sniedz apmācības piemērus (3.5. attēlā solis *1a*), kurus datu apstrādes modulis sagatavo iekšējā formātā un saglabā *Piemēru krātuvē* (*1b*). Par uzdevuma izpildi lietotājs saņem atgriezenisko saiti (*1c*). Kad lietotājs iniciē klasifikatora izveidi (*2a*), *Klasifikatora veidošanas modulis* uz *Piemēru bāzē* esošo datu pamata (*2b*) inducē klasifikācijas modeli, kas tiek saglabāts *Klasifikatorā* (*2c*), darbības pabeigšanu apstiprinot lietotājam (*2d*). Lai noteiktu klases piederību jaunam piemēram vai piemēriem, lietotājs izsauc *Klasifikatora lietošanas moduli* (*3a*), kurš, izmantojot *Klasifikatoru*, veic piemēru klases piederības noteikšanu (*3b*). Ja klasifikācija ir veiksmīga, t.i., klasifikators spēj atrast atbilstošus likumus, rezultāti tiek demonstrēti lietotājam (*3c*). Ja piemērs netiek klasificēts vai tiek klasificēts nepārliecinoši, *Klasifikatora lietošanas modulis* sūta pieprasījumu *Interaktivitātes modulim* risināt situāciju (*3d*). Interaktivitātes modulis lūdz neklasificētā piemēra klasifikāciju veikt sistēmas lietotājam – ekspertam (*3e*); šajā gadījumā darbība tiek iniciēta no sistēmas puses, nevis no lietotāja puses, kā iepriekšējos gadījumos. Saņemot lietotāja atbildi (*3f*), *Interaktivitātes modulis* sniedz atbildi *Klasifikatora lietošanas modulim* un papildina *Piemēru bāzi* (*3g*), kā rezultātā arī iespējama *Klasifikatora* atjaunošana, sazinoties ar *Klasifikatora veidošanas moduli (2b, 2c)*. Eksperta lēmuma pieņemšana ir ārpus šī darba apskatāmo jautājumu loka. Klasifikācijas sistēmā tiek sagaidīts viens eksperta lēmums par piemēra klases piederību.

Lai definētu mehānismu, kā noteikt ekspertam nododamos piemērus, vispirms jānoskaidro, kā šie piemēri tiek identificēti. Tas tiks iztirzāts darba nākamajā apakšnodaļā.

3.3. Ekspertam vaicājamo piemēru noteikšana

Iepriekš 2.1.5.3. sadaļā tika apspriesta viena no klasifikācijas uzdevumos sastopamām problēmām – nespēja noteikt klases piederību jaunam piemēram – un esošās risinājumu pieejas šajā situācijā. Pirms detalizēt darbības, ko tālāk veikt ar neklasificētu piemēru, vispirms jādefinē, kādos gadījumos piemēru atzīst par neklasificētu.

Tiešākajā nozīmē neklasificēts piemērs ir tāds, kuram nav noteikta klase pēc klasifikatora lietošanas – neviens likums vai zars klasifikācijas kokā nav atbilstošs. Tomēr ne vienmēr ir nepieciešams ‘piespiest’ klasifikatoru veikt klasifikāciju ‘par katru cenu’ – svarīgi ir saglabāt rezultātu precizitāti un, līdz ar to, klasifikatora uzticamību. Klasifikatoram būtu jāspēj noteikt, ka tas vairs nevar pieņemt uzticamu lēmumu, tas ir, *jāapzinās sava nezināšana*. Tādēļ darba sākotnējais uzstādījums – izmantot interaktīvu pieeju, lai dotu ekspertam apstrādāt piemērus, ko klasifikators nav spējis klasificēt, – ir paplašināms arī uz nepārliecinoši klasificētu

piemēru noteikšanu. Viens no iemesliem šādam paplašinājumam ir tas, ka tiešā veidā nespēju klasificēt piemēru atzīst maza daļa induktīvās apmācības algoritmu, un tie praktiski ir novecojuši, piemēram, *CN2*, *Prism*. Par to liecina šo algoritmu minimāla izmantošana zinātniskajās publikācijās atspoguļotajos praktiskajos lietojumos un izņemšana no aktīvi lietotām programmām, piemēram, *Weka* (sākot no versijas 3.7.3. *Prism* ir atrodams tikai *Simple educational shemes* bibliotēkā un nav atrodams grafiskajā *Weka Explorer*, tāpat kā, piemēram, populārais, bet praksē vairs nelietotais *ID3* algoritms). Daļa likumus veidojošie algoritmi, piemēram, *Ripper*, vispār neparedz iespēju piemēru atstāt neklasificētu, jo tas pamatā piešķir noklusēto likumu, ja vien nav spēkā kāds no izņēmumiem. Tas nozīmē, ka jebkurā gadījumā piemēram tiks piešķirta klase, lai cik maza pārliecība par to būtu.

Klasifikators lēmuma pieņemšanā var ņemt vērā pārliecību, kura tiek noteikta, balstoties uz piemēru sadalījumu apmācības kopā, kas tika izmantota klasifikatora izveidošanai. Piemēram, ja likums, kurš nosaka klases *A* piešķiršanu, pārklāj 3 piemērus ar klasi *A* un 2 piemērus ar klasi *B*, tad klašu sadalījums ar šo likumu klasificētam piemēram ir 0.6 klasei *A* un 0.4 klasei *B*, iegūstot klasifikatora pārliecību - 0.6 (par piešķiramo klasi - *A*). Promocijas darbā par ***nepārliecinoši klasificētu piemēru*** jeb piemēru ar zemu klasifikācijas pārliecību (ang. v. – *low confidence of classification*) tiek saukts tāds piemērs, kam klasifikatora iegūtā klases pārliecība ir pārāk zema, lai piešķirtu klasi. Dažādas klasifikācijas metodes un algoritmi lieto atšķirīgus pārliecības noteikšanas veidus. Standarta klasifikatori, kuros tiek izmantota pārliecība klasifikācijas lēmumu pieņemšanā, izmanto slieksni 0.5. Ja pārliecība ir lielāka vai vienāda ar 0.5, tad klase tiek piešķirta, ja mazāka par 0.5, tad netiek. Tomēr dažādās problēmsfērās un lietojumos pārliecības sliekšņa lielumu var diferencēt, pieprasot no klasifikatora lēmumus, kuri ir ar lielāku pārliecības pakāpi, tādējādi cenšoties iegūt vairāk pareizu lēmumu. Ja klasifikācijas mērķis ir iegūt pēc iespējas korektākus rezultātus, pieļaujot, ka daļa piemēru tiek atzīti par neskaidriem, jo klasifikators nav pārliecināts par sava lēmuma pareizumu, tad ir iespējams noteikt citu pārliecības slieksni, pie kura klasifikators pieņem lēmumu.

Tādējādi klasifikatora sniegto rezultātu uzlabošanai un nepareizi klasificēto piemēru skaita samazināšanai tiek noteikti un nodoti ekspertam ne tikai **neklasificēti**, bet arī **nepārliecinoši klasificēti** piemēri, kas promocijas darbā kopā tiek definēti kā **klasifikatoram neskaidri piemēri** jeb neskaidra klasifikācija (ang. v. – *uncertain classification*).

Neskaidro piemēru identificēšana un piemērotākā pārliecības sliekšņa noteikšana daudzkategoriju klasifikācijā tiks sīkāk apskatīta darba 4.1., 4.2. un 4.3. apakšnodaļā.

3.4. Eksperta sniegto zināšanu iekļaušana klasifikatorā

Viens no galvenajiem interaktīva algoritma uzdevumiem ir eksperta atgriezeniskās saites pieņemšana un klasifikatora atjaunošana jauno zināšanu rezultātā. Šī apakšnodaļa sniedz darba autores izstrādātos risinājumus, kas sākotnēji atspoguļoti publikācijās [91, 130]. Pirms lietotāja klasificētā piemēra izmantošanas klasifikatora papildināšanai ir jānoskaidro, vai (1) likumu bāzi ir nepieciešams atjaunot [131], un, ja ir nepieciešams, tad (2) kā to darīt. Ir iespējams izmantot dažādas apmācības stratēģijas. Sadaļā 3.4.1. tiks izklāstītas dažādas pieejas eksperta lēmuma apstrādei, kā arī šo pieeju sekas.

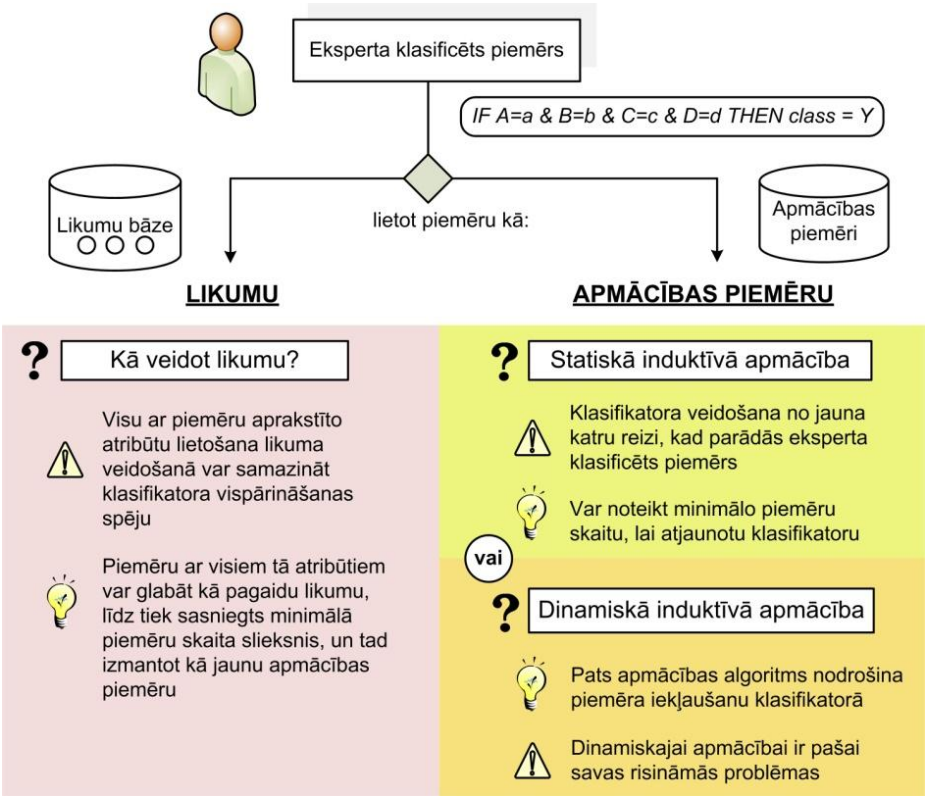
Vienmēr, kad jāapvieno zināšanas no dažādiem informācijas avotiem, ir jāparedz mehānisms savstarpējo konfliktu novēršanai [93]. Šajā gadījumā eksperta spriedums un klasifikatorā esošā zināšanu bāze ir dažādi avoti. Konflikti, kas var atgadīties jauna piemēra klasificēšanas laikā, jau tika apskatīti 2.1.5.2. sadaļā. Konflikti, kas jārisina jauna likuma ieviešanā, tiks iztirzāti 3.4.2. sadaļā, savukārt 3.4.3. sadaļa tiks veltīta paskaidrojošam piemēram.

3.4.1. Zināšanu apvienošanas pieejas

Klasifikatora zināšanas ir jāatjauno tad, kad klasifikators nav spējis noteikt klases piederību jaunam piemēram un to pēc tam ir izdarījis eksperts. Eksperts var arī nevēlēties, lai viņa lēmums tiktu izmantots klasifikatora papildināšanai (piemēram, ja viņš nav drošs par lēmuma pareizību). Ja eksperts vēlas, lai viņa lēmumu par neklasificētā piemēra klasi izmanto klasifikatora uzlabošanai, jāatbild uz svarīgu jautājumu – vai eksperta lēmumu uzreiz uzskatīt par jaunu likumu vai tikai par papildu apmācības piemēru? Dažādās šī jautājuma risinājumu sekas ir attēlotas 3.6. attēlā.

Ja eksperta lēmumu uztver kā jaunu likumu, tad jāizlemj, kā noteikt likumā izmantojamos atribūtus. Ir jārēķinās, ka piemēra aprakstīšanai var tikt izmantoti daudzi atribūti. Veidot un glabāt likumu ar visiem pieejamajiem atribūtiem un to vērtībām ir neefektīvi gan no vispārināšanas, gan izmantotās atmiņas viedokļa. Ilgtermiņā šāda pieeja samazinās klasifikatora vispārināšanas spēju, tādējādi samazinot prognozēšanas precizitāti jauniem piemēriem.

Ja eksperta klasificētais piemērs tiek uztverts kā jauns apmācības piemērs, tālākās darbības ir atkarīgas no tā, vai klasifikators ir balstīts uz statisko vai dinamisko induktīvās apmācības pieeju.



3.6. att. Iespējamās pieejas un to problēmas eksperta veiktās klasifikācijas tālākai izmantošanai

Ja ir izmantotas statiskās apmācības metodes, tad eksperta klasificētais piemērs ir jāiekļauj sākotnējā piemēru kopā un apmācība visam klasifikatoram jāveic no jauna. Šāda pieeja prasa palielinātus laika un skaitļošanas resursus un ir neefektīva salīdzinot ar dinamiskajām metodēm [82]. Tomēr šis apstāklis var nebūt kritisks, ja sistēmai nav jādarbojas reālajā laikā. Tas, vai atkārtota klasifikatora apmācība ir pieņemama, ir atkarīgs no problēmsfēras un apmācības uzdevuma. Risinājums skaitļošanas apjoma samazināšanai ir gaidīt vairākus apmācības kopā iekļaujamos piemērus un tikai tad atjaunot klasifikatoru, nevis to darīt ar katru piemēru atsevišķi. Piemēru skaitu, kas jāgaida, var noteikt ar sliekšņa lielumu. Kamēr slieksnis nav sasniegts, ir iespēja apvienot abas pieejas – eksperta klasificētos piemērus uz laiku pievienot likumu bāzei, izmantojot pilnu piemēra garumu. Pēc vajadzīgā sliekšņa sasniegšanas, pagaidu likumus no likumu kopas izņem un visus piemērus izmanto klasifikatora atjaunināšanai. Sliekšņa lieluma noteikšana ir atkarīga no (1) laika, kāds nepieciešams, lai izveidotu klasifikatoru no apmācības piemēriem un (2) eksperta klasificēto piemēru parādīšanās biežuma. Abi šie parametri kopā raksturo to, kāda slodze ir sistēmai, atjaunojot klasifikatoru. Jo augstākas ir izmaksas klasifikatora atjaunošanai, jo augstāks minimālais piemēru skaits tiks uzlikts atkārtotas apmācības veikšanai.

Dinamiskās apmācības (ang. v. - *incremetnal learning*) algoritmu izmantošanai ir savas priekšrocības un trūkumi. Dinamisku algoritmu izmantošana ir vēlama tādēļ, ka par eksperta sniegtā apmācības piemēra pievienošanu klasifikācijas modelim, kā arī klasifikatora saskanību rūpēsies pats apmācības algoritms, kuram jaunu piemēru parādīšanās ir plānota pēc būtības. No otras puses, dinamiskās apmācības algoritmiem ir pašiem savas problēmas, kas jārisina to konstruēšanas gaitā, piemēram, glabājamo apmācības piemēru skaita noteikšana [132], līdz ar to izvēle par labu dinamiskās apmācības algoritmam ir vēlama tad, ja statisko algoritmu sniegums konkrētā problēmsfērā neapmierina.

Apkopojot apskatītās iespējas, autore ir izstrādājusi divus ieteicamos veidus, kā atjaunot klasifikatoru pēc tam, kad eksperts ir sniedzis savu viedokli par neklasificētā piemēra klases piederību.

Uz slieksni balstīta statiskās apmācības pieeja, kas iekļauj šādus soļus:

1. Sliekšņa uzstādīšana – pozitīva skaitļa izvēle, kas atspoguļo skaitu, cik eksperta klasificētu piemēru jāgaida, lai veiktu klasifikatora atjaunošanu.
2. Eksperta klasificēto piemēru pievienošana likumu kopai ar visiem piemēra atribūtiem, kamēr netiek sasniegts slieksnis.
3. Statiskās induktīvās apmācības metodes lietošana, lai izveidotu klasifikatoru, balstoties uz visiem sākotnējiem apmācības un eksperta klasificētajiem piemēriem.
4. Iepriekšējā klasifikatora aizstāšana ar jauno.
5. Atgriešanās pie 2. soļa.

Dinamiskās apmācības pieeja, kas iekļauj šādus soļus:

1. Sniegt eksperta klasificēto piemēru apmācības algoritmam un ļaut klasifikatoru atjaunot tādā veidā, kā to paredz pats algoritms.
2. Lietot atjaunoto klasifikatoru.

Dinamiskās induktīvās apmācības algoritmi iedalās algoritmos bez piemēru atmiņas, ar daļēju piemēru atmiņu un pilnu piemēru atmiņu (sīkāk apskatīts darba 6. pielikumā). 3.3. tabulā ir parādītas induktīvo apmācības algoritmu stiprās (+) un vājās (-) puses, pieņemot, ka problēmsfērā eksperta klasificēti piemēri parādās ne pārāk bieži. Ar „+” tiek saprasta augstāka klasifikatora precizitāte, mazākas atkārtotas apmācības izmaksas, mazākas datu glabāšanas izmaksas, labāka pielāgošanās koncepta izmaiņām laika gaitā. Tabula balstīta uz datiem, kas sniegti [82, 100, 132-134].

3.3. tabula

Dažādu induktīvās apmācības metožu tipu raksturojums

	Statiskās metodes	Dinamiskās metodes bez piemēru atmiņas	Dinamiskās metodes ar daļēju piemēru atmiņu	Dinamiskās metodes ar pilnu piemēru atmiņu
Klasifikatora precizitāte	+	-	-	+
Atkārtotas apmācības izmaksas	-	+	+	+
Datu glabāšanas izmaksas	-	+	+	-
Koncepta izmaiņu ievērošana	-	+	+	-

Saskaņā ar [132] teikto, statiskās apmācības algoritmi var sniegt augstāku klasifikācijas precizitāti kā dinamiskie algoritmi ar daļēju vai bez piemēru atmiņas (ja problēmsfērā nav novērojamas lielas koncepta izmaiņas laikā). Ja ir sagaidāms, ka jaunpienākušo apmācības piemēru būs daudz, tad ir vērts apsvērt dinamiska algoritma lietošanu. Šī izvēle būs atkarīga arī no nepieciešamā un pieejamā atmiņas apjoma, vēlamās precizitātes un citiem apsvērumiem, kurus vislabāk var novērtēt eksperimentāli.

Apmācības stratēģijas izvēle kļūst par daudzkriteriālu uzdevumu, kurā jāsabalansē klasifikatora prognozēšanas precizitāte, atkārtotas apmācības izmaksas, datu glabāšanas izmaksas un citi parametri, kas tiktu atzīti par būtiskiem konkrētajam uzdevumam. Jāņem vērā prasības pret risinājumu, ierobežojumi un zināšanas par problēmsfēru.

3.4.2. Likumu bāzes saskanības nodrošināšana

Šajā sadaļā tiks aplūkots klasifikatorā esošās likumu bāzes saskanības jeb integritātes jēdziens, lai definētu klasifikācijas sistēmā nodrošināmos kritērijus un izskaidrotu, kā iepriekšējā darba sadaļā aprakstītās klasifikatora atjaunošanas pieejas tos spēj ievērot. Lai klasifikatora papildināšana neizraisītu pretrunas ar esošajām zināšanām [135], ir nepieciešams definēt integritātes nodrošināšanas metodes. Vispārīgi datu bāzu integritātes nosacījumi ir aprakstīti literatūrā, piemēram, [135, 136]. Viens no svarīgiem uzdevumiem datu bāzes projektēšanā ir identificēt nodrošināmos integritātes nosacījumus, kuri jāuztur, lai nodrošinātu datu bāzes nepretrunīgumu [136]. Induktīvās apmācības sistēmā klasifikators (likumu kopa) var tikt uzskatīts par datu bāzi, līdz ar to uz to var attiecināt datu bāzu integritātes nosacījumus.

Jēdzienam ‘nepretrunīgums’ vai ‘konsekvence’ (ang. v. – *consistency*) literatūrā ir piedāvātas vairākas definīcijas [131]. Nepretrunīgums var attiekties uz algoritmu, kas producē klasifikatoru [131], apmācības piemēriem, atsevišķu likumu un likumu bāzi. Induktīvās apmācības kontekstā ir apskatāmi šādi izteikumi par pretrunīgumu un konsekventumu.

- Divi *piemēri* ir pretrunīgi, ja tos apraksta vienas un tās pašas atribūtu vērtības, bet tiem ir atšķirīgas klases [133].
- *Likums* ir nepretrunīgs, ja apmācības kopas ietvaros tas pārklāj tikai prognozētās klases piemērus. Tas gan nenozīmē, ka likums ir saskanīgs ar visiem testa kopas piemēriem, kas vēl nav redzēti. Šis princips klasifikatoros bieži netiek ievērots labākas klasifikatora vispārināšanas labad. Atkāpjoties no nosacījuma radīt tikai nepretrunīgus likumus (jeb nodrošināt 100 % klasifikatora pārliecību par likuma piešķirto klasi), tiek samazināta klasifikatora pārapmācība un vairota noturība pret nepiederošiem un kļūdainiem piemēriem. Piemēram, iepriekš 2. nodaļā aprakstītais klasifikācijas algoritms *Ripper* neveido konsekventus likumus, bet atspoguļo pārliecību par likuma pareizību.
- *Apmācības algoritms* ir uzskatāms par konsekventu, ja dažādās apmācības reizēs (bet vienai un tai pašai apmācības kopai) tas ģenerē klasifikatoru, kurš vienādi klasificē jaunus, iepriekš neredzētus piemērus [131]. Tas nenozīmē, ka klasifikatori ir identiski, bet tie pieņem vienādus lēmumus.
- Nepretrunīgums ir *algoritma* spēja iegūt līdzvērtīgu klasifikācijas precizitāti dažādu apmācības sesiju laikā [131].

Pastāv konsekvences mēri, kas orientējas uz apmācības algoritma spēju radīt līdzīgi klasificējošu likumu kopas dažādos apmācības gadījumos, piemēram, mērs, kas aprakstīts [131]. Metodes, piemēram, avotā [135] aprakstītā, kas var nodrošināt integritātes uzturēšanu datu bāzēs, un arī induktīvās apmācības radītajās likumu bāzēs, ir pazīstamas un tiek plaši lietotas [136].

Ņemot vērā iepriekš izteiktos apgalvojumus, induktīvās apmācības klasifikatorā jānodrošina šādi integritātes nosacījumi [131]:

- likumiem jābūt savstarpēji izslēdzošiem, nepārklājošiem;
- likumi, kas satur vienādus atribūtu-vērtību pārus (vienādus izteikumus nosacījumu daļā), nedrīkst piešķirt dažādas klašu vērtības.

Apskatīsim, kā abas piedāvātās klasifikatora atjaunošanas pieejas nodrošina integritātes nosacījumus. Pirmais nosacījums paredzēts minimāli nepieciešamā likumu skaita uzturēšanai un nevajadzīgu likumu izslēgšanai. Ja klasifikācijas algoritms, kas tiek izmantots klasifikatora iegūšanai, nodrošina nepārklājošos un nepretrunīgu likumu veidošanu, tad abas klasifikatora atjaunošanas pieejas neizmaina šos principus tādēļ, ka jauns likums tiks pievienots tikai tādā gadījumā, ja neviens esošais likums nespēja klasificēt piemēru. Tomēr tas neizslēdz iespējamību, ka likumi, kas līdz šim bija saskanīgi, tādi vairs nav pēc jauna klasificējama

piemēra parādīšanās. Piemēram, likumu bāze satur likumus "*IF A=a AND B=b THEN klase = 0*" un "*IF C=c AND D=d THEN klase = 1*". Parādās jauns piemērs "*A=a AND B=b AND C=c AND D=d*". Abi likumi klasificē piemēru atšķirīgi. Šī situācija ir jārisina ar kādu no 2.1.5.2. sadaļā aprakstītajām metodēm un to nerisina jauna likuma pievienošanas laikā.

3.4.3. *Uz slieksni balstītās statiskās apmācības pieejas* demonstrācija

Ar vienkārša piemēra palīdzību tiks demonstrēta iepriekš 3.4.1. sadaļā aprakstītā darba autores izstrādātā *Uz slieksni balstītā statiskās apmācības pieeja*. *Dinamiskās apmācības pieeja* netiek izvērsta tādēļ, ka pastāv daudzas dinamiskā apmācības metodes, kuras veic piemēra iekļaušanu saskaņā ar savu algoritmu (šīs metodes ir apskatītas darba 6. pielikumā). Savukārt, uz *slieksni balstītā statiskās apmācības pieeja* nekur iepriekš nav aprakstīta.

3.4. tabula satur sākotnējos apmācības piemērus vienkāršotā problēmsfērā, kas raksturo dažādas dabas parādības konkrētā gadalaikā. Problēmsfēru apraksta trīs nomināli atribūti - temperatūra, koku lapas un laikapstākļi. Klases vērtības definē gadalaiku.

3.4. tabula

Apmācības piemēri gadalaika klasificēšanai

	Temperatūra	Koki	Laiks	**Gadalaiks**
1	vidēja	dzelteni	lietains	**rudens**
2	vidēja	kaili	saulains	**pavasaris**
3	augsta	zaļi	saulains	**vasara**
4	zema	kaili	saulains	**ziema**

Likumu ģenerēšanai ir izmantots statiskās apmācības algoritms *RULES-3* [137]. Šis algoritms darbojas viegli saprotamā veidā un sniedz pārskatāmus rezultātus likumu formā.

3.5. tabula

Sākotnējā likumu kopa

Likumu kopa
IF koki = dzelteni THEN gadalaiks = rudens
IF temperatūra = vidēja AND koki = kaili THEN gadalaiks = pavasaris
IF temperatūra = augsta THEN gadalaiks = vasara
IF temperatūra = zema THEN gadalaiks = ziema

Demonstrācijai ir izvēlēti divi sliekšņa lielumi – viens (sl = 1) un trīs (sl = 3). Tas nozīmē, ka pirmajā gadījumā eksperta klasificētais piemērs uzreiz tiks iekļauts sākotnējā apmācības kopā un klasifikators ģenerēts no jauna, bet otrajā gadījumā tiks gaidīti 3 eksperta klasificēti piemēri līdz klasifikatora atjaunošanai. Pārbaudīsim, kā algoritms strādā abos šajos

gadījumos, secīgi pienākot 3 jauniem klasificējamiem piemēriem, un pieņemot, ka mums ir divi klasifikatori ar vienādu sākotnējo likumu kopu, bet atšķirīgiem sliekšņiem.

Pirmais piemērs, kam jānosaka klases piederība, ir "*koki = zaļi AND temperatūra = vidēja AND laiks = lietains*". Neviens no esošajiem likumiem (skat. 3.5. tabulu) nevar klasificēt šo piemēru. Piemēru nododot klasificēšanai ekspertam, tiek saņemta atbilde "pavasaris". Klasifikatorā ar sl = 1 klasifikators tiek ģenerēts no jauna, sākotnējiem apmācības piemēriem (3.4. tabula) pievienojot jauniegūto piemēru, kas der apmācības papildināšanai „*koki = zaļi AND temperatūra = vidēja AND laiks = lietains THEN gadalaiks = pavasaris*". Tā rezultātā ir iegūts jauns likums "*IF koki = zaļi AND temperatūra = vidēja THEN gadalaiks = pavasaris*", kas tiek pievienots likumu kopai. Klasifikatorā ar sl = 3, likumu kopai tiek pievienots pagaidu likums pilna piemēra garumā, bet esošā likumu kopa netiek pārskatīta (skat. 3.6. tabulu).

3.6. tabula

Likumu bāzes pēc pirmā klasificētā piemēra

Klasifikators 1 (sl = 1)	Klasifikators 2 (sl = 3)
IF koki = dzelteni THEN gadalaiks = rudens	IF koki = dzelteni THEN gadalaiks = rudens
IF temperatūra = vidēja AND koki = kaili THEN gadalaiks = pavasaris	IF temperatūra = vidēja AND koki = kaili THEN gadalaiks = pavasaris
IF temperatūra = augsta THEN gadalaiks = vasara	IF temperatūra = augsta THEN gadalaiks= vasara
IF temperatūra = zema THEN gadalaiks = ziema	IF temperatūra = zema THEN gadalaiks = ziema
IF temperatūra = vidēja AND koki = zaļi THEN gadalaiks = pavasaris	**IF temperatūra = vidēja AND koki = zaļi AND laiks = lietains THEN gadalaiks = pavasaris**

Kā otrais piemērs klasifikatoriem parādās šis: "*koki = zaļi AND temperatūra = vidēja AND laiks = mākoņains*". Klasifikators ar sl = 1 var klasificēt šo piemēru ar nupat radīto likumu. Otrs klasifikators padod šo piemēru ekspertam, kurš nosaka klasi "pavasaris". Šim klasifikatoram arī tiek pievienots jauns pagaidu likums (skat. 3.7. tabulu).

3.7. tabula

Likumu bāzes pēc otrā klasificētā piemēra

Klasifikators 1 (sl = 1)	Klasifikators 2 (sl = 3)
IF koki = dzelteni THEN gadalaiks = rudens	IF koki = dzelteni THEN gadalaiks = rudens
IF temperatūra = vidēja AND koki = kaili THEN gadalaiks = pavasaris	IF temperatūra = vidēja AND koki = kaili THEN gadalaiks = pavasaris
IF temperatūra = augsta THEN gadalaiks= vasara	IF temperatūra = augsta THEN gadalaiks= vasara
IF temperatūra = zema THEN gadalaiks = ziema	IF temperatūra = zema THEN gadalaiks = ziema
IF temperatūra = vidēja AND koki = zaļi THEN gadalaiks = pavasaris	**IF temperatūra = vidēja AND koki = zaļi AND laiks = lietains THEN gadalaiks = pavasaris**

	IF temperatūra = vidēja AND koki = zaļi AND laiks = mākoņains THEN gadalaiks = pavasaris

Trešais klasificējamais piemērs ir "*koki = kaili AND temperatūra = ļoti zema AND laiks = mākoņains*". Neviens no klasifikatoriem tam nevar noteikt klasi, un eksperts definē to kā ziemu. Klasifikatora atjaunošana tagad ir jāveic abiem klasifikatoriem, jo ir līdz ar trešā eksperta klasificētā piemēra parādīšanos ir sasniegts slieksnis arī otrajam klasifikatoram (skat. 3.8. tabulu).

3.8. tabula

Likumu bāzes pēc trešā klasificētā piemēra

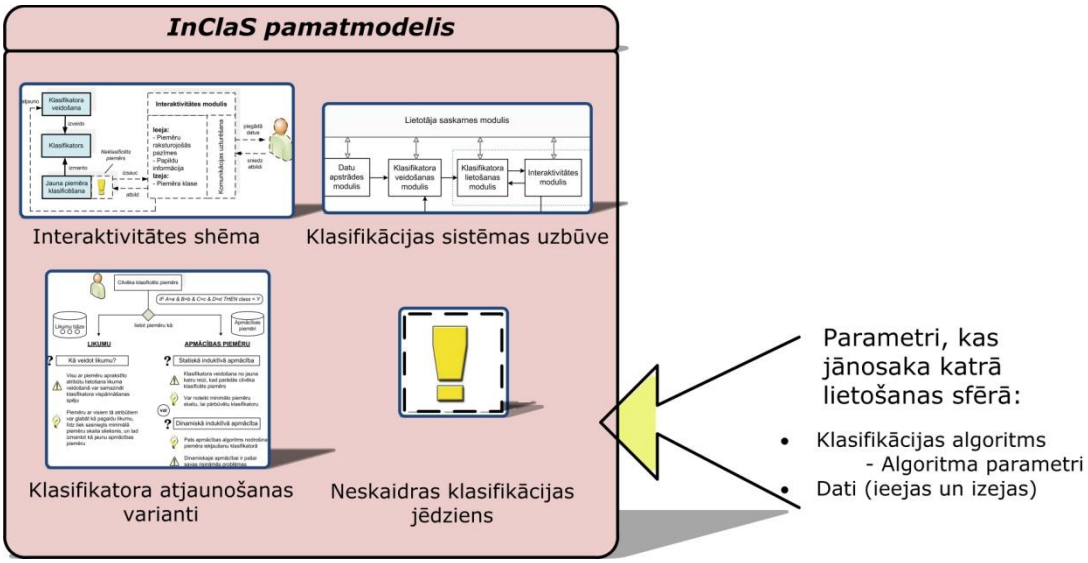
Klasifikators 1 (sl = 1)	Klasifikators 2 (sl = 3)
IF koki = dzelteni THEN gadalaiks = rudens	IF koki = dzelteni THEN gadalaiks = rudens
IF temperatūra = vidēja AND koki = kaili THEN gadalaiks = pavasaris	IF temperatūra = vidēja AND koki = kaili THEN gadalaiks = pavasaris
IF temperatūra = augsta THEN gadalaiks= vasara	IF temperatūra = augsta THEN gadalaiks= vasara
IF temperatūra = zema THEN gadalaiks = ziema	IF temperatūra = zema THEN gadalaiks = ziema
IF temperatūra = vidēja AND koki = zaļi THEN gadalaiks = pavasaris	**IF temperatūra = vidēja AND koki = zaļi THEN gadalaiks = pavasaris**
IF temperatūra = ļoti zema THEN gadalaiks = ziema	**IF temperatūra = ļoti zema THEN gadalaiks = ziema**

Abi klasifikatori tagad izskatās vienādi, jo pagaidu likumi ir atmesti un veikta abu klasifikatoru atkārtota apmācība. Šis piemērs rāda, ka algoritms ar zemāku sliekšņa lielumu traucēja ekspertu retāk (sl = 1 to darīja divas reizes, bet sl = 3 - trīs reizes). Klasifikators ar zemāko slieksni ātrāk spēja vispārināt un lietot eksperta sniegtās zināšanas, līdz ar to viens piemērs jau tika klasificēts ar jauniegūto likumu, nevis vēršoties pie eksperta. Tomēr, lai arī algoritms ar zemāku slieksni dod ātrākus rezultātus vispārināšanas ziņā, tas prasa vairāk skaitļošanas. Klasifikatoram ar slieksni, kas augstāks par viens, vispārināšanas spēja pirms klasifikatora atjaunošanas, ieviešot pagaidu likumu, tiek palielināta minimāli. Tomēr pagaidu likums palīdz gadījumā, ja vēlāk parādās tieši tāds pats piemērs kā ieviestais pagaidu likums. Līdz ar to apskatītais piemērs ļauj izteikt pieņēmumu, ka, izmantojot jebkuru sliekšņa lielumu, paredzamā nepieciešamība pēc eksperta iesaistīšanas ir mazāka, ja tiek izmantoti pagaidu likumi, nekā, ja šādi likumi netiek lietoti. Protams, jo mazāks sliekšņa lielums, jo mazāka sagaidāmā nepieciešamība vērsties pie eksperta nepamatoti.

3.5. *InClaS* pamatmodeļa komponentes

Šajā apakšnodaļā tiek apkopotas visas darba 3. nodaļā izstrādātās un aprakstītās komponentes, kuras veido *InClaS* pamatmodeli – interaktīvas klasifikācijas sistēmas kodolu. Nodaļa kopumā definē nepieciešamos elementus *InClaS* modeļa izveidei, izstrādājot un pamatojot:

- vispārējo ieviešamo interaktivitātes shēmu (3.2. apakšnodaļa, 3.3.att.);
- klasifikācijas sistēmas uzbūvi, tās moduļus un saites starp tiem (3.2.2. apakšnod. 3.4.att.);
- klasifikatora atjaunošanas ieteicamos variantus (3.4.1. apakšnod. 3.6. att.);
- vispārēju skaidrojumu par neskaidras klasifikācijas jēdzienu (3.3. apakšnod.), kuru jāspēj apstrādāt interaktīvajā sistēmā.



3.7. att. Interaktīvās klasifikācijas sistēmas pamatmodelis

3.7. attēlā shematiski parādītas visas *InClaS* modeli veidojošās komponentes, ar kuru palīdzību ieviest interaktivitāti automātiskā klasifikācijas sistēmā.

3.6. Nodaļas kopsavilkums

Klasifikācijas sistēmas papildināšana ar interaktivitāti ne tikai var uzlabot klasifikācijas rezultātus, bet arī vairot lietotāju uzticību, jo ļauj ieskatīties sistēmas darbībā. Šajā nodaļā izklāstīta interaktīvas uz induktīvo apmācību balstītas klasifikācijas sistēmas modeļa ***InClaS*** (ang. v. - ***In****teractive Inductive Learning Based* ***Cla****ssification* ***S****ystem*) izveide. Šis modelis piedāvā risinājumu to piemēru klasificēšanai, kurus klasifikators pats nespēj pārliecinoši klasificēt. Brīdī, kad klasifikators sastopas ar šādu piemēru, ir iespējams pavaicāt sistēmas

lietotājam – ekspertam kādu klasi viņš piešķirtu šim objektam. Pēc tam no šīs atbildes var izsecināt arī jaunu likumu, ko saglabāt likumu bāzē, tādējādi papildinot klasifikatora zināšanas.

Nodaļa apraksta un pamato interaktīvas klasifikācijas sistēmas projektēšanu un uzbūvi. Attiecībā uz interaktīvu klasifikācijas sistēmu:

- sniegta vispārīga **klasifikācijas procesa shēma** problēmsfērām ar konceptuāli sarežģītiem datiem;
- parādītas izmaiņas, kas jāveic klasifikācijas sistēmā, lai ieviestu interaktivitāti;
- definēta **sistēmas uzbūve**, moduļi un izskaidroti **projektēšanas soļi**.

Kā būtisks pamats tālākām darbībām, izskaidrots **neklasificēta, nepārliecinoši klasificēta un klasifikatoram neskaidra piemēra jēdziens**, nosakot, ka šī darba kontekstā neskaidra klasifikācija, kura būtu jānodod izvērtēšanai ekspertam, ir gan tādi piemēri, kuriem klasifikators nav spējis noteikt klases piederību, gan piemēri, kuriem klasifikators dod zemu pārliecību par veikto klasifikāciju. Pārliecības līmenis ir parametrs, kurš var tikt variēts un piemeklēts atbilstošākais katrā lietojumā; šī temata tālāks izklāsts sekos darba nākamajā nodaļā.

Kad neskaidri klasificētais piemērs ir nodots ekspertam, un ir saņemts eksperta lēmums par piemēra klasifikāciju, sistēmai nepieciešams apstrādāt iegūto informāciju. Klasifikatora atjaunošanai darba autore piedāvā divas alternatīvas, kuras nodrošina likumu bāzes saskanības ievērošanu:

Uz slieksni balstīta statiskās apmācības pieeja, kas, kā nosaukums liecina, izmanto statisku apmācības algoritmu un eksperta klasificētos piemērus pievieno kā pagaidu likumus, līdz tiek sasniegts iepriekš uzstādītais piemēru skaita slieksnis. Kad slieksnis ir sasniegts, piemēri tiek pievienoti sākotnējai apmācības piemēru kopai, un tiek veikta klasifikatora atkārtota apmācība, bet pagaidu likumi - dzēsti.

Dinamiskās apmācības pieeja izmanto dinamiskās apmācības algoritmu, eksperta klasificēto piemēru lietojot kā jaunu apmācības piemēru un darbojoties ar to atbilstoši savam algoritmam.

Uz *InClaS* pamatmodeļa pamata tālāk tiks attīstīts nākamais modeļa līmenis interaktīvai klasifikācijas sistēmai daudzkategoriju klasifikācijas uzdevumiem.

4. *INCLAS* MODELIS DAUDZKATEGORIJU KLASIFIKĀCIJAS UZDEVUMAM

Šajā nodaļā tiks aprakstīta 3. nodaļā sniegtā *InClaS* modeļa paplašinājuma izstrāde lietošanai daudzkategoriju klasifikācijas gadījumos, tas ir, klasifikācijas uzdevumos, kur katrs objekts jeb piemērs var piederēt vienlaicīgi vairākām klasēm. Vispirms tiks dziļāk analizēts neskaidras klasifikācijas jēdziens (4.1. apakšnodaļā) un sniegts algoritms neskaidru piemēru noteikšanai daudzkategoriju klasifikācijas gadījumā (4.2. apakšnodaļā). Metode piemērotākā pārliecības sliekšņa lieluma noteikšanai aprakstīta 4.3. apakšnodaļā. 4.4. apakšnodaļa seko ar klasifikācijas sistēmas detalizāciju studiju priekšmetu salīdzināšanas uzdevumam, nodaļu noslēdz 4.5. apakšnodaļa ar visu komponenšu apkopojumu, kas ietilpst *InClaS* modeļa paplašinājumā daudzkategoriju klasifikācijas gadījumā.

4.1. Neskaidras klasifikācijas jēdziens daudzkategoriju kontekstā

Vispārējie principi piemēra atzīšanai par neskaidru tika apspriesti 3.3. apakšnodaļā. Daudzkategoriju klasifikācijas uzdevumi ļauj plašāk palūkoties uz klasifikatora nespēju noteikt piemēra klases piederību. Iepriekš darba 2.1.1. sadaļā tika apskatīti vairāki daudzkategoriju uzdevuma risināšanas varianti, no kuriem plaši izmantota ir binārās saistības pieeja - daudzkategoriju problēmu sadalot vairākos vienas kategorijas uzdevumos. Tādējādi piemēra klasifikāciju nosaka kopējie rezultāti no *n* vienas kategorijas klasifikatoriem, kur katrs atsevišķais klasifikators lemj par piemēra piederību tikai vienai klasei. Ja netiek konstatēta piederība nevienai no atsevišķajām klasēm (pēc noklusējuma klasifikācijā tradicionāli lietoto pārliecības slieksni – 0.5 – nesasniedz neviena klase), tad piemērs var tikt uzskatīts par neklasificētu. 4.1. tabulā demonstrēts piemērs, kur objektam iespējamas vienlaicīgi 4 dažādas klases. Izmantojot binārās saistības pieeju šī daudzkategoriju klasifikācijas uzdevuma transformēšanai vienkategorijas uzdevumā, tiek iegūti 4 bināri klasifikatori, kur katrs nosaka piemēra piederību savai klasei (attiecīgi, klasei *A,* klasei *B,* utt.), un izsaka savu pārliecību par klases piešķiršanu. Rezultātā atsevišķo klasifikatoru sniegtās atbildes tiek apkopotas un, ievērojot pārliecības slieksni, klasificējamajam objektam piešķirtas klases. 4.1. tabulas piemēra gadījumā piemērs tiek atzīts par neklasificētu, jo pārliecība par nevienu no 4 atsevišķajām kasēm nesasniedz līmeni 0.5.

4.1. tabula

Neklasificēts piemērs pie noklusētā pārliecības sliekšņa 0.5

	Klases			
	A	B	C	D
Īstās klases	**1**	**0**	**1**	**0**
Klasifikatora pārliecība	**0.2**	**0.1**	**0.4**	**0.2**
Klasifikatora sniegtās atbildes	**0**	**0**	**0**	**0**

Dažādās problēmsfērās ir atšķirīga specifika attiecībā uz pārliecības līmeni. Vienā datu kopā līmenis 0.5 labi nošķir piederību vai nepiederību klasei, tikmēr citā var būt nepieciešams arī augstāks slieksnis, lai lēmums būtu pārliecinošs. Piemēram, 4.2. tabulā vidējais pārliecību līmenis par klasifikācijām visās klasēs ir augstāks kā iepriekšējā piemērā, un arī klase *D*, kas nav piemēra īstā klase, ir piešķirta ar pārliecību 0.5. Ja slieksnis, pie kura piešķirt klasifikāciju, tiktu pacelts līdz 0.6, tad klase *D* netiktu attiecināta uz šo piemēru.

4.2. tabula

Klasifikācija pie dažādiem pārliecības sliekšņa lielumiem

	Klases			
	A	B	C	D
Īstās klases	**1**	**0**	**1**	**0**
Klasifikatora pārliecība	**0.3**	**0.1**	**0.6**	**0.5**
Klasifikatora sniegtās atbildes (ja slieksnis ir 0.5)	**0**	**0**	**1**	**1**
Klasifikatora sniegtās atbildes (ja slieksnis ir 0.6)	**0**	**0**	**1**	**0**

Tātad, lai labāk pielāgotu klasifikatoru konkrētajai problēmsfērai, vēlams noteikt piemērotāko sliekšņa lielumu katrā konkrētā lietojumā, balstoties uz apmācības datu kopu. Darba autores izstrādātā metode piemērotākā sliekšņa lieluma noteikšanai tiks aprakstīta 4.3. apakšnodaļā.

Dažādas vienas kategorijas un daudzkategoriju klasifikācijas uzdevumos biežāk lietotās metrikas tika apskatītas darba 2.1.3. sadaļā. Papildus tām šajā darbā autore ir ieviesusi arī citas metrikas, kas konkrētāk saistās tieši ar darba mērķa sasniegšanas novērtēšanu – nepareizi klasificēto piemēru skaita samazināšanu un lietotāja iesaistīšanu klasifikācija procesā. Tās balstās uz vienkāršu parametru novērtēšanu:

Piemērs ir nepareizi klasificēts (N) – neviena no klasifikatora noteiktajām klasēm nav piemēra īstā klase (noteikto un īsto klašu kopām šķēlums ir tukšs) $Y_i \cap Z_i = \emptyset$,

kur Y_i – īstā klašu kopa *i*-tajam piemēram,

Z_i – klasifikatora prognozētā klašu kopa *i*-tajam piemēram.

Piemērs ir klasificēts vismaz daļēji pareizi (DP) – vismaz viena no prognozētajām klasēm ir piemēra īstā klase $Y_i \cap Z_i \neq \emptyset$.

Interaktīvās metodes novērtēšanai ir ieviesti papildu parametri, kas balstās uz klasifikatora pārliecību par piešķirtajām klasēm. Klasificējamais piemērs tiek atzīts par neskaidru un tiek nodots izvērtēšanai ekspertam, ja pie noteikta pārliecības līmeņa tam nav piešķirta neviena klase. Izvērtējot izvēlēto pārliecības līmeni, piemērs, kas ir nodots apstrādei ekspertam, var tikt pieskaitīts vienai no divām neskaidri klasificētu piemēru grupām:

Īsti neskaidra klasifikācija (ĪN) – piemērs būtu nepareizi klasificēts (N), ja tiktu uzticēts tikai klasifikatoram (tas ir, ja ar noklusēto pārliecības līmeni 0.5 netiktu pareizi prognozēta neviena no īstajām klasēm).

Nepamatoti neskaidra klasifikācija (NN) – piemērs būtu klasificēts kā vismaz daļēji pareizs (DP), ja būtu uzticēts klasifikatoram (tas ir, ja ar noklusēto pārliecības līmeni 0.5 vismaz viena prognozētā klase būtu piemēra īstā klase).

Protams, klasifikācijai ir jātiecas uz rezultātu, kurā pēc iespējas vairāk piemēru ir klasificēti pareizi vai vismaz daļēji pareizi (daudzkategoriju gadījumā), bet, ja tas nav iespējams klasifikatora nepilnību dēļ, tad klasifikācijas sistēmai būtu jāatpazīst nepareizi klasificēti piemēri un jāidentificē tie kā neskaidri. Neskaidru piemēru starpā, savukārt, ir nepieciešams maksimizēt īstos neskaidros piemērus un necelt ‘viltus trauksmi’, liekot ekspertam ieguldīt darbu un analizēt piemērus, kas būtu klasificēti vismaz daļēji pareizi.

Lai paskaidrotu ieviesto mēru lietošanu un to demonstrētu ar reālu klasifikācijas situāciju, 4.1. attēlā ir sniegts piemērs klasifikācijas rezultāta novērtēšanai ar aprakstītajiem mēriem piecos klasifikācijas gadījumos. Klasifikācijas rezultāts ir aprakstīts ar klašu vektoru. Pieņemot, ka objekts var piederēt 10 klasēm, klašu vektors satur 10 secīgus ierakstus, attiecīgi, ar vērtību "0", ja atbilstošā klase nav piešķirta un "1", ja ir.

Lai secinātu klasifikācijas pareizumu, tiek salīdzināti prognozētie un īstie zināmie klašu vektori. Nepareizas klasifikācijas gadījumā (ja klasifikators ar standarta slieksni nespēj noteikt nevienu klasi pareizi) jātiecas uz pārliecības sliekšņa paaugstināšanu, lai piemēru atzītu par neskaidru, un klasifikators pats lēmumu nepieņemtu, bet nodotu piemēru ekspertam. Savukārt daļēji pareizas klasifikācijas gadījumā piemēra nodošana ekspertam nav vēlama, jo klasifikators pats to būtu spējis klasificēt gana pareizi. Līdz ar to klasifikatora pārliecības sliekšņa noteikšana ir kompromiss, pie kura jānonāk katrā konkrētā datu kopā. Ja nepareizi klasificēts objekts interaktīvā sistēmā tiek atzīts par neskaidri klasificētu, tas tad patiešām ir neskaidrs un ir lietderīgi ticis piedāvāts klasifikācijai ekspertam. Pretējā gadījumā – ja piemērs ir daļēji pareizs, bet tiek atzīts par neskaidru, tad eksperta laika patēriņš, to klasificējot, nav lietderīgs.

	Objekta klašu vektori (klases tiek prognozētas ar standarta slieksni 0.5)		*Secinājums*	*Ja slieksnis ir augstāks un piemēru atzīst par neskaidru (prognozētais klašu vektors ir 0000000000) , tad piemērs ir..*
1. piemērs	0010010000	Īstās klases		
	1001000000	Prognozētās klases		
2. piemērs	0000000000	Īstās klases	**Nepareiza klasifikācija (N)**	➔ **Īsti neskaidra klasifikācija (ĪN)**
	0000000000	Prognozētās klases		
3. piemērs	0100111111	Īstās klases		
	0100000000	Prognozētās klases		
4. piemērs	0100000000	Īstās klases	**Daļēji pareiza klasifikācija (DP)**	➔ **Nepamatoti neskaidra klasifikācija (NN)**
	0100111111	Prognozētās klases		
5. piemērs	0100111000	Īstās klases		
	0001111100	Prognozētās klases		

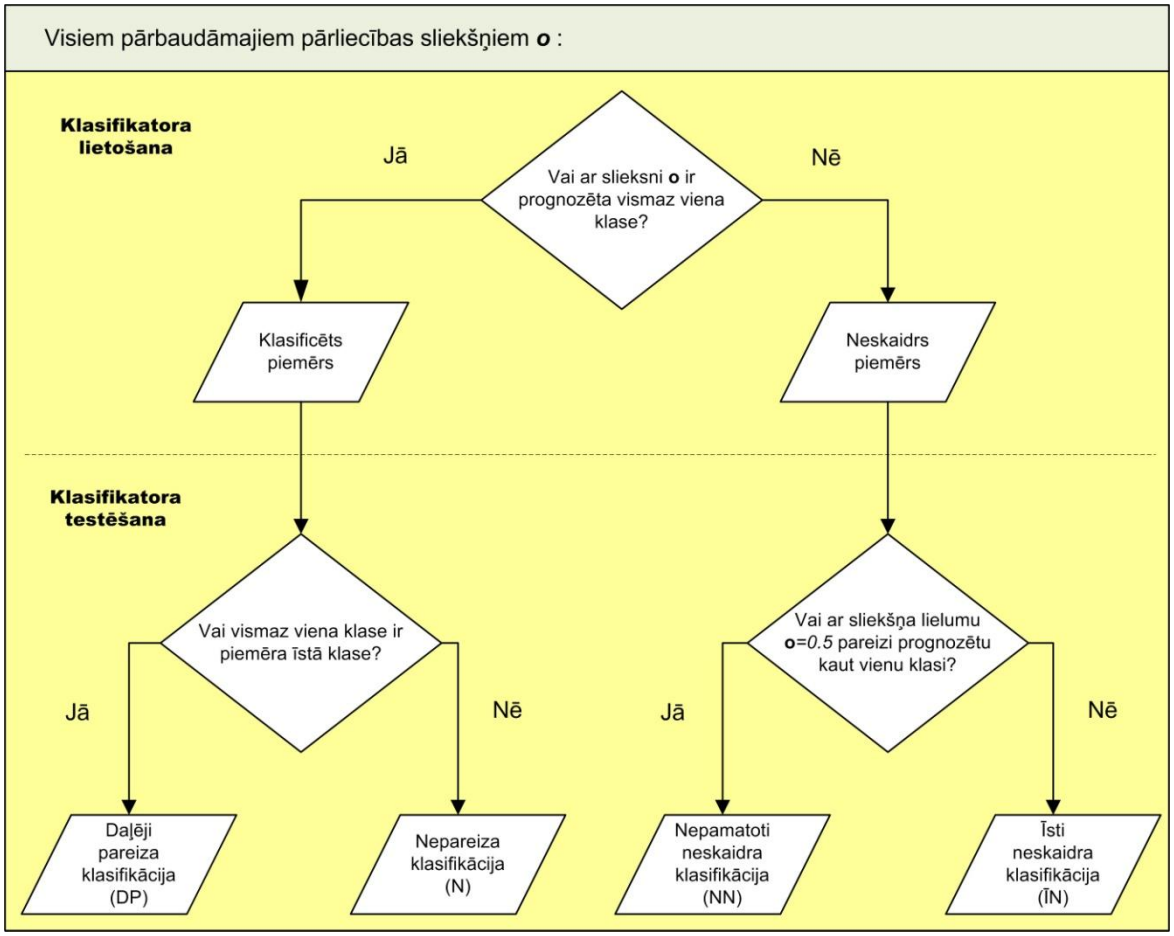
4.1. att. Piemērs novērtējumu mēru lietojumam

Daļēji pareizo klasifikāciju turpmākas analīzes rezultātā būtu nepieciešams diferencēt sīkāk, jo, kā redzams arī 4.1. attēlā, daļēji pareizas klasifikācijas atšķiras pēc klašu vektoru atbilstības. Var secināt, ka pareizi prognozētas 3 klases no 4 ir labāks un vēlamāks rezultāts par pareizi atpazītu tikai vienu klasi no 4 īstajām klasēm. Tomēr šī darba ietvaros par daļēji pareizu klasifikāciju tiks uzskatīta jebkura klasifikācija, kam vismaz viena īstā klase ir arī prognozēta.

4.2. Algoritms klasifikatoram neskaidru piemēru noteikšanai

Par neskaidru piemēru tiek atzīts tāds piemērs, kam pēc klasifikatora lietošanas neviena klase nav piešķirta ar pārliecību virs noteiktā sliekšņa lieluma. Pēc noklusējuma sliekšņa lielums pārliecībai ir 0.5. Vienkategorijas klasifikācijā nevienai no iespējamajām klasēm pārliecība nav sasniegusi slieksni; daudzkategoriju gadījumā ne par vienu no klasēm (binārās saistības

metodēm) pārliecība nav sasniegusi slieksni. Tātad daudzkategoriju klasifikācijā par neskaidru tiek atzīts tāds piemērs, kam visu klašu piederības pie lietotā pārliecības sliekšņa ir "0" vai *false*. Savukārt, klasifikatora testēšanas laikā nosakāmos parametrus, kas raksturo par neskaidriem atzīto piemēru īsto 'dabu' un sliekšņa lieluma adekvātumu, iegūst pēc 4.2. attēlā demonstrētā algoritma.



4.2. att. Algoritms klasifikatoram neskaidru piemēru noteikšanai daudzkategoriju klasifikācijā

Attēlā ir redzams, ka klasifikatora lietošanas laikā jauniem piemēriem tiek noteikts, vai par noteikto klases(-šu) piederību klasifikators ir pārliecināts, bet, ja tā ir klasifikatora testēšanas fāze, kur piemēram ir zināmas īstās klases, tad iespējams izvērtēt lēmuma pamatotību un novērtēt klasifikatora darba korektumu.

4.3. Piemērotākā pārliecības sliekšņa lieluma noteikšana

Iepriekš 3.3. apakšnodaļā tika secināts, ka, lai labāk pielāgotu klasifikatoru konkrētajai problēmsfērai un pieejamajai datu kopai, vēlams noteikt piemērotāko sliekšņa lielumu katrā konkrētā lietojumā. Pirmo ieskatu pārliecības sliekšņa lielumu vērtībās konkrētajā problēmsfērā spēj sniegt autores ieviestie mēri - **v**idējā klasifikatora **p**ārliecība par klasēm, kurām piemēri ir

piederīgi (VPP) un **v**idējā klasifikatora **p**ārliecība par klasēm, kurām piemēri ir **n**av piederīgi (VPN). VPP aprēķina dalot pārliecību summu par visām klasēm, kurām piemēri ir piederīgi ar piederīgo klašu skaitu, bet VPN - dalot pārliecību summu par visām klasēm, kurām piemēri nav piederīgi ar nepiederīgo klašu skaitu. Formāli to atspoguļo 4.1 un 4.2 formulas.

$$\mathrm{VPP} = \frac{\sum_{i=1}^{n} \sum_{j \in J_i} p_{ij}}{\sum_{i=1}^{n} |J_i|}, \tag{4.1}$$

$$\mathrm{VPN} = \frac{\sum_{i=1}^{n} \sum_{j \notin J_i} p_{ij}}{\sum_{i=1}^{n} |J \backslash J_i|}, \tag{4.2}$$

kur $X = \{X_i\},\ i = 1,2,\dots,n$ piemēru kopa,

$K = \{K_j\},\ j = 1,2,\dots,m$ klašu kopa,

$p_{ij} = p(X_i, K_j)$ pārliecība, ka $X_i \in K_j$,

$J = \{1,2,\dots,m\}$ visu klašu indeksu kopa,

J_i to klašu indeksu kopa, kurām pieder X_i,

$J \backslash J_i$ to klašu indeksu kopa, kurām nepieder X_i,

$|\ \ |$ kopas apjoms (elementu skaits tajā).

Kā mērķi izvirzot nepareizi klasificēto piemēru skaita samazināšanu, nepieciešams noteikt, kāds sliekšņa lielums ir vispiemērotākais. Vidējās pārliecības var palīdzēt izvēlēties apgabalu, kas ir pārmeklējams, un gūt priekšstatu, cik pārliecinoši atdalāmas ir klases, kurām piemēri pieder vai nepieder. 4.3. tabula un tai sekojošie aprēķini demonstrē piemēru, kā tiek noteikta VPP un VPN trīs klašu gadījumā datu kopā, kas sastāv no diviem piemēriem.

4.3. tabula

Klasifikācija pie dažādiem pārliecības sliekšņa lielumiem

Atribūti	Klases		
	A	B	C
Apmācības piemērs 1			
Īstās klases	**1**	**1**	**0**
Klasifikatora prognozētās pārliecības katrai klasei	**0.6**	**0.7**	**0.3**
Apmācības piemērs 2			
Īstās klases	**0**	**0**	**1**
Klasifikatora prognozētās pārliecības katrai klasei	**0.2**	**0.6**	**0.8**

$$\mathrm{VPP} = \frac{0.6 + 0.7 + 0.8}{3} = 0.7$$

$$\mathrm{VPN} = \frac{0.3 + 0.2 + 0.6}{3} = 0.367$$

No aprēķiniem var secināt, ka šajā datu kopā pārliecības slieksnis, kas nošķir klasei nepiederošos piemērus no klasei piederošiem, orientējoši ir meklējams starp 0.367 un 0.7. Tomēr īpaši pazemināt slieksni zem 0.5 nav lietderīgi, jo tādā gadījumā tiek ņemti vērā ļoti nepārliecinoši klasifikatora lēmumi.

Piemērotākā pārliecības sliekšņa noteikšanā jārēķinās arī ar papildu ierobežojumiem, kas saistīti ar eksperta iesaistīšanu neklasificēto piemēru apstrādē. Formāli to iespējams definēt kā darba apjomu, ko eksperts ir gatavs ieguldīt klasifikatora rezultātu uzlabošanā. Darba apjoma mērīšanai tiek piedāvāti divi mēri:

- cik piemēru eksperts klasificē (kopējais caurskatāmo piemēru skaits - visi klasifikatoram neskaidrie piemēri):

$$\mathrm{D}_{kopējais} = \bar{\mathrm{I}}\mathrm{N} + \mathrm{NN} \ . \tag{4.3}$$

  Šis mērs ir relatīvs pret klasificējamo piemēru skaitu, piemēram, $\mathrm{D}_{kopējais}$ = 40 uz 100 piemēriem.

- cik pareizi klasificētu piemēru eksperts caurskata, lai atrastu vienu nepareizi klasificētu piemēru (ieviešot mēru - eksperta veiktās klasifikācijas (darba) nelietderība, kuru izsaka attiecība starp nepamatoti neskaidrajām un īsti neskaidrajām klasifikācijām):

$$\mathrm{D}_{nelietderīgais} = \frac{\mathrm{NN}}{\bar{\mathrm{I}}\mathrm{N}} \ . \tag{4.4}$$

  Ja $\mathrm{D}_{nelietderīgais}$ vērtība ir 0, tad eksperta ieguldītais darbs klasifikācijā ir vislietderīgākais, jo uz visiem ĪN nav neviena NN piemēra.

Lai definētu metodi piemērotākā sliekšņa lieluma izvēlei, tiks veikti eksperimenti un uzskatāmi analizēti rezultāti. Eksperimentu plāns atspoguļots 4.4. tabulā.

4.4. tabula

Eksperimentu plāns sliekšņa lieluma ietekmes novērtēšanai

	1. variants	**2. variants**
Ieejas datu kopa	Medicīnas datu kopa [30]	
Klasifikācijas algoritms	Binārā saistība (ar *JRIP* algoritmu pamatā)	Binārā saistība (ar Naivo Beijesa klasifikatoru pamatā)
Novērtējamie parametri	DP, N, ĪN, NN, $\mathrm{D}_{kopējais}$, $\mathrm{D}_{nelietderīgais}$	
Pārbaudes apgabals	Sliekšņa lielumi [0.4; 0.8] ar soli 0.1	

Jāpiebilst, ka nav principiālas atšķirības, vai risināmais uzdevums pēc piešķiramo kategoriju skaita ir vienas kategorijas vai daudzkategoriju. Daudzkategoriju uzdevumā tiek lietots parametrs „daļēji pareizs" piemērs (DP), kamēr vienas kategorijas gadījumā būtu „pareizs" piemērs (P). Aprēķinos par pārliecības slieksni tas netiek iesaistīts. Tā kā darba

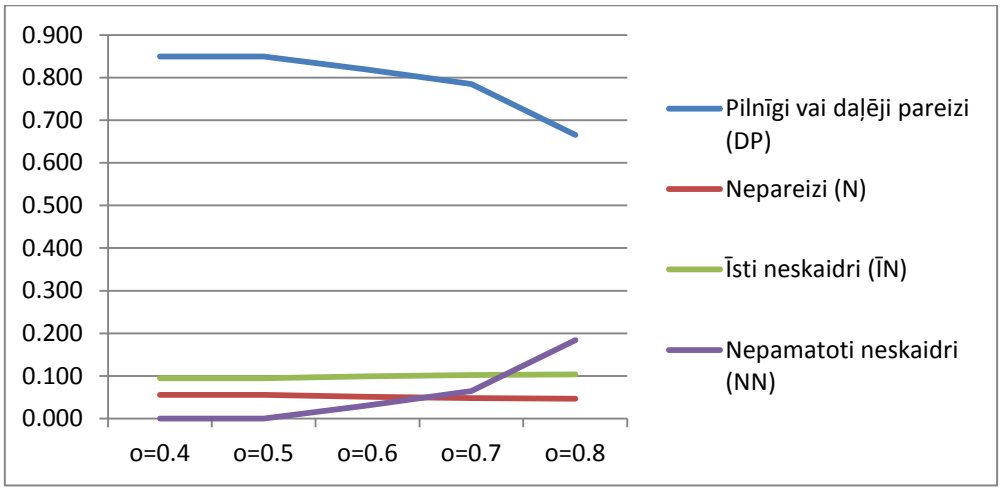
galvenā uzmanība tiek pievērsta daudzkategoriju klasifikācijai, tad eksperimenti arī tiek veikti ar daudzkategoriju datu kopām.

Vidējie sliekšņa lielumi starp pozitīvajiem un negatīvajiem klasifikatora spriedumiem apmācības kopā ir šādi (atbilstoši formulām 4.1 un 4.2):

$$VPP = 0.685$$

$$VPN = 0.019$$

Grafisks atspoguļojums klasifikācijas rezultātiem attiecībā uz DP, N, ĪN, NN, pie dažādiem sliekšņa lielumiem medicīnas datu kopai eksperimenta 1. variantā sniegts 4.3. attēlā. $D_{kopējais}$, $D_{nelietderīgais}$ attēlots 4.5. tabulā.



4.3.att. DP, N, ĪN, NN izmaiņas pie dažādiem pārliecības sliekšņa lielumiem eksperimenta 1. variantā

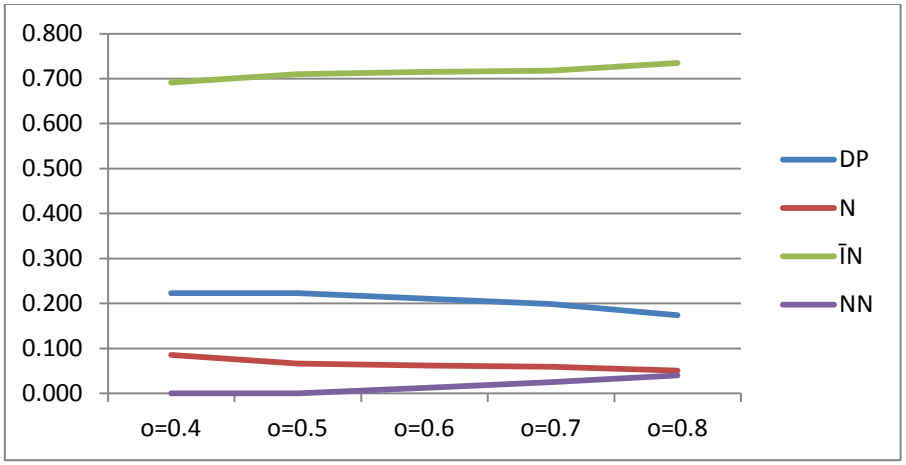
4.5. tabula

Eksperta ieguldāmais darbs eksperimenta 1. variantā

Slieksnis	**o=0.5**	**o=0.6**	**o=0.7**	**o=0.8**
$D_{kopējais}$	61	84	108	186
$D_{nelietderīgais}$	0	0.3125	0.636	1.776

4.5. tabulas dati ļauj secināt, ka eksperta darba lietderība, palielinot pārliecības slieksni, samazinās. Ja pie sliekšņa 0.6 ekspertam jāizskata vidēji 1 jau daļēji pareizi klasificēti piemērs uz trim patiesi neskaidrajiem, tad virs sliekšņa 0.8 šī attiecība ir 1,8 pret 1. Tāpat arī redzams, ka pie sākotnējā sliekšņa lieluma 0.5, eksperta darbs ir ar 100% lietderību, jo visiem piemēriem, kas pie šāda sliekšņa ir atzīti par neklasificētiem, standarta klasifikators bez interaktivitātes nespētu noteikt klases piederību. Palielinot slieksni, palielinās arī daļēji pareizo piemēru skaits starp tiem piemēriem, ko klasifikators atzīst par nepārliecinoši klasificētiem.

Apskatot rezultātus eksperimenta 2. variantā (4.4. attēlā un 4.6. tabulā) redzams, ka šeit sākotnējais DP īpatsvars ir ļoti mazs, bet ĪN - liels. Eksperta darba lietderība liecina, ka pat pie visaugstākā sliekšņa lieluma eksperta darbs ir ļoti vērtīgs ar nelielu NN īpatsvaru neklasificēto piemēru vidū.



4.4. att. DP, N, ĪN, NN izmaiņas pie dažādiem pārliecības sliekšņa lielumiem eksperimenta 2. variantā

4.6. tabula

Eksperta ieguldāmais darbs eksperimenta 2. variantā

Slieksnis	**o=0.5**	**o=0.6**	**o=0.7**	**o=0.8**
$D_{kopējais}$	458	469	479	521
$D_{nelietderīgais}$	0	0.017	0.035	0.055

Salīdzinot abu eksperimentu rezultātus, var secināt, ka parametrus nepieciešams skatīties kontekstā vienam ar otru. Ja otrajā eksperimentā $D_{nelietderīgais}$ it kā ir ekspertam draudzīgāks, tad, skatoties uz $D_{kopējo}$, redzams, ka eksperta reālā iesaistīšana ir vairākas reizes lielāka – pie sliekšņa 0,6 ekspertam ir jācaurskata 84 ieraksti pirmajā gadījumā, bet 469 – otrajā. Visticamāk – otrais rezultāts vispār netiktu apspriests kā praktiskajā dzīvē realizējams. Līdz ar to var secināt, ka svarīgi ir izvērtēt neklasificēto piemēru skaitu ar sākotnējo standarta slieksni 0.5 un atrast pieņemamāko klasifikācijas algoritmu no konkrētajā situācijā pieejamajiem variantiem. Palielinot slieksni, neklasificēto piemēru skaits tikai pieaugs, tāpēc, ja ar sākuma slieksni algoritms uzrāda vājus klasifikācijas rezultātus un pārāk daudz piemēriem nespēj piešķirt klasi, tad nepieciešams meklēt citu klasifikācijas algoritmu. 4.7. tabulā redzams klasifikācijas rezultātu apkopojums vienai un tai pašai datu kopai vienādos eksperimenta apstākļos ar dažādiem bāzes klasifikācijas algoritmiem, kas parāda, cik atšķirīgus rezultātus var sniegt klasifikācijas algoritmi.

4.7. tabula

Eksperimentu rezultāti piemērotākā sliekšņa lieluma atrašanai

Slieksnis	**o=0.5**	**o=0.6**	**o=0.7**	**o=0.8**
Eksperimenta 1. variants (*JRip* klasifikators)				
$D_{nelietderīgais}$ (NN / ĪN)	0	0.3125	0.636	1.776
DP	548	528	506	429
N	36	33	31	30
$D_{kopējais}$ (NN + ĪN)	61	84	108	186
Eksperimenta 2. variants (Naivais Beijesa klasifikators)				
$D_{nelietderīgais}$ (NN / ĪN)	0	0.017	0.035	0.055
DP	144	136	128	117
N	43	40	38	34
$D_{kopējais}$ (NN + ĪN)	458	469	479	521

Eksperimentu rezultātu analīze ļauj izvirzīt metodi atbilstošā pārliecības sliekšņa līmeņa noteikšanai konkrētiem gadījumiem. Metodes uzdevums: atrast atbilstošāko pārliecības slieksni, kur nepareizi klasificēto piemēru skaits N ir minimāls pie lietotāja izvirzītajiem ierobežojumiem (d1 un d2). Formalizējot: N→*min*, $D_{kopējais} \leq d1$; $D_{nelietderīgais} \leq d2$.

Metode manuālai piemērotākā algoritma un sliekšņa lieluma izvēlei

1. Izvēlēties klasifikācijas algoritmu, veikt klasifikatora apmācību un testēšanu, noteikt N, ĪN, NN, $D_{kopējais}$ un $D_{nelietderīgais}$ pie pārliecības sliekšņa 0.5. Ja neklasificēto piemēru skaits $D_{kopējais}$ prasa pārāk lielu eksperta darbu, izmēģināt citu klasifikācijas algoritmu.
2. Ja neklasificēto piemēru skaits $D_{kopējais}$ ir pieņemams praktiskam lietojumam, mainīt sliekšņa lielumus un noteikt N, ĪN, NN, $D_{kopējo}$ un $D_{nelietderīgo}$Intervāls, ko pārmeklēt, ir saistīts ar orientējošajiem vidējiem sliekšņa lielumiem VPP un VPN starpā. Standarta solis ir 0.1, bet iespējams lietot sīkāku soli.
3. Atspoguļot un izvērtēt rezultātus ar
   - $D_{kopējais}$, $D_{nelietderīgais}$, N.

Novērtējumam par pamatu tiek ņemts nepareizi klasificēto piemēru skaits. Izvērtēt sliekšņa lielumus, ņemot vērā šādus faktorus:

- ja pie konkrēta sliekšņa N nav lielāks kā iepriekšējā solī, ir vērts izskatīt sliekšņa palielināšanu;
- ja ir vienāds nepareizi klasificēto piemēru skaits vairākos stāvokļos, tad dot priekšroku stāvoklim ar mazāku $D_{kopējais}$ (un ĪN skaitu, kas ir tieši saistīti lielumi);
- ir iespējami vairāki savstarpēji ekvivalenti stāvokļi, kuros parametri nemainās.

Ir iespējams arī iegūt automātisku sliekšņa noteikšanu, neveicot manuālu datu analīzi.

Metode automātiskai piemērotākā algoritma un sliekšņa lieluma izvēlei

- Mērķis – minimizēt nepareizi klasificēto piemēru skaitu, ņemot vērā lietotāja izvirzītos ierobežojumus ieguldāmajam darbam.
- Ievadāmie parametri (viens vai abi)
  - slieksnis lietderīgajam darbam (piemēram, 5);
  - slieksnis kopējam klasificējamo piemēru skaitam (piemēram, 50).
- Izpildīt soļus 1 un 2 no manuālās metodes.
- Izvēlēties slieksni, kuram ir minimālais nepareizi klasificēto piemēru skaits un ir spēkā lietotāja sniegtie parametri. Liela datu apjoma gadījumā, kad eksperimentu veikšanas iespējas ir ierobežotas, turpināt palielināt slieksni, līdz pārsniegtas parametros norādītās vērtības.

Jāņem vērā, ka šādā veidā iegūtie aprēķini balstās uz sadalījumu apmācības kopā un var nebūt pilnībā patiesi datiem, ar kuriem klasifikācijas sistēma saskarsies nākotnē, klasificējot jaunus objektus. Metode darbosies viskorektāk, ja spēkā būs viens no tradicionāliem klasifikācijas uzdevuma pieņēmumiem, ka piemēri apmācības kopā atspoguļo objektu īpašības un sadalījumu reālajā problēmsfērā, ar kuru turpmāk saskarsies klasifikācijas sistēma savā darbībā.

4.4. Klasifikācijas sistēmas projektēšana studiju priekšmetu salīdzināšanai

Šī apakšnodaļa apraksta 3.2. apakšnodaļā sniegtās interaktīvās klasifikācijas sistēmas detalizāciju konkrētai daudzkategoriju klasifikācijas problēmsfērai - universitātes studiju priekšmetu salīdzināšanai. Daļa no lēmumiem, kas jāpieņem sistēmas projektēšanas laikā, ir tieši atkarīgi no problēmsfēras. Klasifikācijas sistēmas specifiku nosaka problēmsfērai raksturīgās īpašības, ierobežojumi un prasības. Kā secināts 2.3. apakšnodaļā, problēmas nostādne, sfēras specifisko faktoru analīze un lietojuma identificēšana ir sistēmas izstrādes pamatā. Visiem tālākajiem projektēšanas lēmumiem būtu jābalstās uz reālajām sfēras vajadzībām, ņemot vērā arī tehniskās iespējas.

Projektēšanas soļi interaktīvai klasifikācijas sistēmai studiju priekšmetu salīdzināšanai ir analizēti saskaņā ar iepriekš 2.3. nodaļā izvēlēto procedūru [127]. Tie ir aprakstīti 4.8 tabulā.

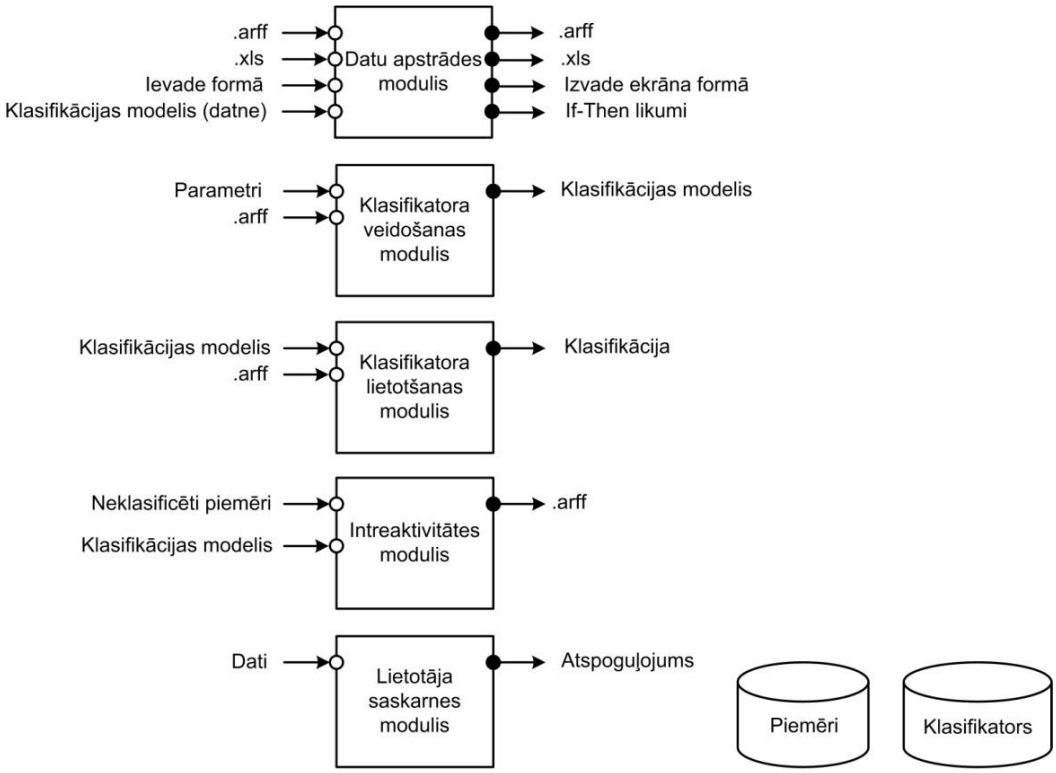
4.8 .tabula

Interaktīvas klasifikācijas sistēmas projektēšana studiju priekšmetu salīdzināšanai

1. Problēmas identificēšana
Globalizācija un studentu mobilitāte rada nepieciešamību pēc studiju programmu un studiju priekšmetu salīdzināšanas, lai būtu iespējams noteikt to savstarpējo atbilstību. Šī salīdzināšana ir nepieciešama tādēļ, ka mācībām citā institūcijā apmaiņas programmas ietvaros ir jāatbilst prasībām, ko nosaka studenta pamata mācību programma [9]. Cita nepieciešamība pēc priekšmetu salīdzināšanas ir jaunu studiju programmu izstrādē. Studiju programmas izstrādes gaitā ir jāveic salīdzinošā analīze ar līdzīgām programmām no citām universitātēm. Studiju programmu atbilstība ir balstīta uz atsevišķu priekšmetu salīdzināšanu, bet jāņem vērā, ka šāda salīdzināšana ir ļoti darbietilpīgs process, ja to veic tikai cilvēks. Tādēļ ir nepieciešama sistēma, kas atbalstītu ekspertus priekšmetu salīdzināšanas procesā. Galvenās problēmsfēras iezīmes, kam ir būtiska loma sistēmas projektēšanas lēmumos, ir šādas [130]. • **Iegūtie rezultāti ir jāsaprot cilvēkam - sistēmas lietotājam un ekspertam.** Šis nosacījums definē induktīvās apmācības - lēmumu koku vai likumus ģenerējošo algoritmu - izmantošanu, kuri spēj paskaidrot risinājuma iegūšanas ceļu. • **Maza sākotnējā apmācības kopa.** Šis apstāklis rada bažas par nepilnīga klasifikatora izveidi un, līdz ar to, nespēju klasificēt visus jaunos piemērus. Tādēļ būtu jānodrošina mehānisms klasifikatoram neskaidro jauno piemēru apstrādei. • **Daļēji strukturēti un nestrukturēti dati.** Studiju priekšmetu apraksti savā sākotnējā formā universitātēs ir dažādi un nav stingri formalizēti. Izgūstot nepieciešamo informāciju strukturētai priekšmetu salīdzināšanai, kā tas ir nepieciešams mašīnapmācības metožu lietošanas gadījumā, daļa informācijas var tikt pazaudēta vai atspoguļota neprecīzi. Arī šis apstāklis rada draudus nepilnīga vai neprecīza klasifikatora izveidei un, tāpat kā mazas sākotnējās apmācības kopas gadījumā, prasa nodrošināt papildu klasifikācijas iespējas jaunajiem piemēriem. Te varētu palīdzēt eksperts un interaktīva klasifikācijas sistēma. • **Daudzas klases, kuras sastopamas vienlīdz bieži.** Studiju programmas parasti satur desmit līdz piecdesmit mācību priekšmetu, un nav pamata uzskatīt, ka kāds no tiem parādītos biežāk vai būtu svarīgāks. Līdz ar to noklusētā likuma lietošana neklasificēto piemēru klases piederības noteikšanai šajā gadījumā nebūs piemērota. • **Piemērs var piederēt vienlaicīgi vairākām klasēm.** Studiju priekšmets vienā studiju programmā var atbilst vai pārklāties ar vairākiem studiju priekšmetiem citā studiju programmā, kas nosaka nepieciešamību nodrošināt iespēju piemēram piešķirt vairākas klases.
2. Zināšanu iegūšana un attēlošana
Lai studiju priekšmetus varētu salīdzināt, ir nepieciešams noteikt priekšmetus raksturojošās pazīmes, pēc kurām to darīt. Raksturīgo atribūtu izraudzīšanās ir ļoti būtisks solis. Studiju priekšmetu apraksti parasti skaidrā veidā nedefinē atribūtus, kas būtu nozīmīgi priekšmetu salīdzināšanai. Būtisks apstāklis atribūtu izvēlē ir arī tas, ka atribūtiem jābūt ne tikai sfēru labi raksturojošiem, bet arī reāli iegūstamiem. Turklāt satura ziņā vairāki priekšmeti var pārklāties, tas ir, viens priekšmets var saturiski iekļaut vairākus priekšmetus citā studiju programmā. Jāņem vērā arī fakts, ka izglītības sniedzējs ne vienmēr ļauj piekļūt pilnam priekšmeta satura aprakstam [9]. Tomēr sasniedzamie mācību rezultāti parasti ir aprakstīti, tādēļ tos var izvēlēties kā galvenos priekšmetu saturu raksturojošos atribūtus. Papildu šiem atribūtiem, var izmantot arī citus priekšmeta formālo pusi nosakošos lielumus, kas ir viegli pieejami – studiju līmenis un kredītpunktu skaits. Ja sasniedzamie mācību rezultāti ir aprakstīti brīvā valodā, pirms lietošanas priekšmetu salīdzināšanai, tos ir nepieciešams unificēt jeb atspoguļot vienotā formā. Par kopīgo formu var izvēlēties kompetenču ietvaru, kas spētu atspoguļot mācību sasniedzamajos rezultātos iegūstamās kompetences. Informācijas tehnoloģijas jomā par starpnieku kompetenču atspoguļošanā var izmantot Eiropas e-kompetenču ietvarstruktūru (ang. v. - *European e-Competence Framework, e-CF*) [35], kas ir Eiropas Komisijas atzīts kopīgais ietvars informācijas un komunikāciju tehnoloģiju kompetencēm. Cita iespēja formālu atribūtu iegūšanai ir priekšmetus biežāk aprakstošo vārdu

lietošana par raksturīgajiem atribūtiem - teksta klasifikācijas pieeja, kas iegūst vārdu vektorus.
3. Rīku izvēle
Klasifikācijas sistēmas izstrādes gadījumā ar rīkiem jāsaprot arī klasifikācijas algoritmi, ar kuru palīdzību iegūs klasifikatoru. Problēmas identificēšanas posmā iegūti vairāki nosacījumi, kas jāņem vērā, izvēloties apmācības algoritmus. Lai iegūtu caurskatāmu klasifikācijas modeli, jāizmanto apmācības metodes, kas ģenerē vispārinošo modeli lēmumu koka vai likumu veidā. Tā kā problēmsfērā iespējama dabiska objektu piederība vairākām klasēm (viens priekšmets var atbilst vairākiem priekšmetiem citā studiju programmā), tad izmantojamo metožu loks iekļauj tikai daudzkategoriju klasifikācijas metodes (kuras, attiecīgi, pēc tam var lietot arī vienkategorijas klasifikācijas algoritmus). Lai aiztaupītu laiku un pūles, implementējot apmācības algoritmus, vēlams izmantot jau gatavas pamata algoritmu realizācijas (bibliotēkas un rīkus). Daudzkategoriju klasifikācijas gadījumā izvēle nav pārāk plaša. *Mulan* [138] bibliotēka daudzkategoriju klasifikācijas realizēšanai ir izvēlēta šādu iemeslu dēļ: 1. tā ir balstīta uz rīka *Weka* [76] bāzes, kas satur daudz klasisko apmācības algoritmu realizāciju; 2. tā ir paplašināma, kas ir ļoti svarīgi interaktivitātes ieviešanai, jo esošie rīki nenodrošina sadarbību ar lietotāju klasifikācijas posmā. Tā kā gan *Mulan* bibliotēka, gan *Weka* rīks ir rakstīti programmēšanas valodā *Java*, tad kopējai interaktīvās klasifikācijas sistēmas prototipa izstrādei arī ir izvēlēta valoda *Java*. Galvenās izstrādājamās sastāvdaļas ir lietotājam draudzīga saskarne, datu transformācijas, klasifikācijas algoritmu darbināšana, sasaistes nodrošināšana starp gatavajiem klasifikācijas algoritmiem un to nepieciešamajiem paplašinājumiem interaktivitātes ieviešanai un saziņai ar ekspertu.
4. Prototipēšana un izstrāde
Piemērotākā klasifikācijas algoritma atrašana, parametru pielāgošana, izvēlēto problēmsfēru raksturojošo atribūtu pārbaude vislabāk var tikt veikta eksperimentējot. Prototipa izstrāde palīdz novērtēt nepieciešamos sistēmas parametrus un pieņemt pamatotus lēmumus par nepieciešamajām arhitektūras izmaiņām.
5. Testēšana un uzturēšana
Testēšanas posmā tiek novērtēti dažādi klasifikatora un citu klasifikācijas sistēmas daļu parametri. Klasifikatora veiktspējas novērtēšanai galvenais izraudzītais mērs ir nepareizi klasificēto piemēru īpatsvars, jo tas tiešā veidā nosaka, vai klasifikators ir praktiski izmantojams konkrētā jomā jaunu piemēru uzticamai klasificēšanai. Klasifikācijas sistēmai būtisks parametrs ir lietošanas ērtums, sevišķi, ja lietotājam ar sistēmu jāsadarbojas pastiprināti, kā tas ir interaktīvas sistēmas gadījumā. Kopumā šai sistēmai jābūt spējīgai klasificēt jaunus piemērus uz iepriekš sniegto apmācības piemēru pamata, komunicēt ar sistēmas lietotāju neklasificētu piemēru gadījumā un atjaunot klasifikatoru, izmantojot zināšanas, kas iegūtas, sadarbojoties ar ekspertu. Izstrādātā prototipa *lietojamību* raksturo tā spēja veikt visas paredzētās darbības klasifikācijas sistēmas funkciju nodrošināšanai un saziņai ar lietotāju.

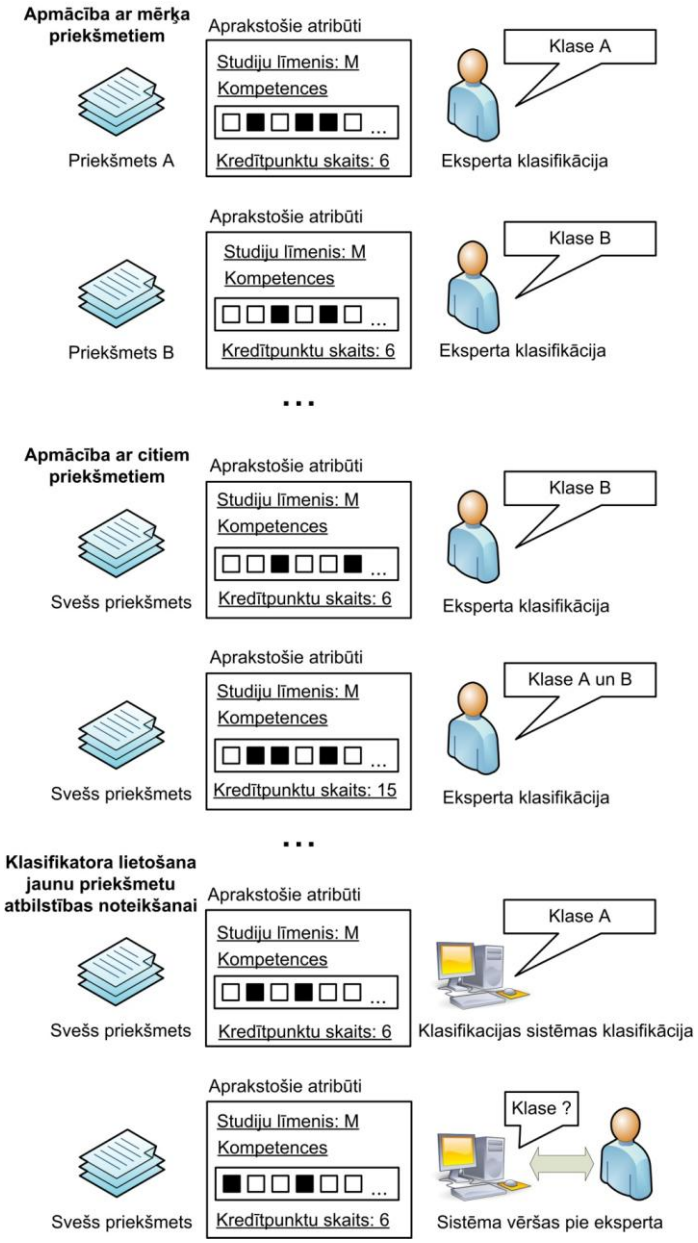
Specificējot sadaļā „Interaktīvas klasifikācijas sistēmas moduļi" sniegto vispārīgo klasifikācijas sistēmas arhitektūru, 4.5. attēlā ir grafiski atspoguļotas sistēmas moduļu galvenās ieejas un izejas. Moduļi integrē viena tipa uzdevumus un nodrošina sistēmas funkciju realizāciju.



4.5. att. Moduļu ieejas un izejas

Datnes tips *.arff* ir specifisks rīkam *Weka* un bibliotēkai *Mulan* un tiek lietots, lai aprakstītu atribūtus un apmācības vai klasificējamos piemērus. Parastam lietotājam vienkāršāk par *.arff* datnēm ir sagatavot datus *.xls* formātā, tomēr visvienkāršāk būtu ievadīt datus caur problēmsfērai pielāgotu formu. Modeļa datnes glabā klasifikatoru *Weka* un *Mulan* iekšējā formātā. Piemēru bāze glabā apmācības piemērus; praktiski tā ir datne *.arff* formātā. Klasifikators glabā klasifikācijas modeli, kas tiek izmantots jaunu piemēru klasificēšanai. No šī modeļa ir iespējams izgūt IF – THEN likumu sarakstu lietotājam lasāmā formā.

4.6. attēls shematiski atspoguļo klasifikatora apmācības un lietošanas procesu netiešās universitāšu studiju priekšmetu salīdzināšanas gadījumā. Par mērķa priekšmetiem tiek izvēlēti priekšmeti no studiju programmas, ar kuru plānots veikt salīdzināšanu. Mērķa priekšmetiem klases saskan ar priekšmeta nosaukumu. Pēc tam eksperts klasificē svešus studiju priekšmetus atbilstoši šīm klasēm, veidojot klasifikatora apmācības kopu.



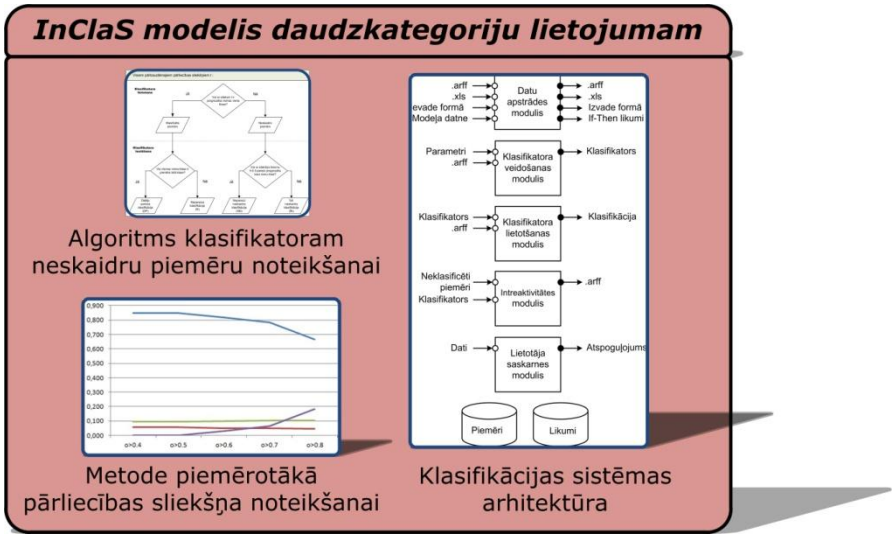
4.6. att. Klasifikatora apmācība un lietošana netiešās salīdzināšanas gadījumā

Jaunu studiju priekšmetu klasificēšanu uzticot klasifikācijas sistēmai, tā vai nu atrod priekšmeta klasifikāciju, vai atzīst savu nepārliecību par lēmumu un vēršas pie eksperta.

4.5. *InClaS* daudzkategoriju klasifikācijas modeļa komponentes

Šajā apakšnodaļā tiek apkopotas visas promocijas darbā izstrādātās un 4. nodaļā aprakstītās komponentes, kuras veido *InClaS* modeļa papildelementus, lai nodrošinātu interaktīvas klasifikācijas sistēmas realizāciju daudzkategoriju klasifikācijas uzdevumiem. Kā *InClaS* modeļa papildinājums daudzkategoriju klasifikācijas uzdevumiem paredzētas šādas komponentes (skat. 4.7. att.):

- algoritms neskaidri klasificētu piemēru noteikšanai (4.2. apakšnodaļa, 4.2. att.);
- metode piemērotākā pārliecības sliekšņa noteikšanai (4.3. apakšnod.);
- klasifikācijas sistēmas arhitektūra - projektēšanas procesa detalizācija un sistēmas uzbūve (4.4. apakšnod., 4.8. tabula, 4.5. att.).



4.7. att. *InClaS* modeļa papildinājums daudzkategoriju klasifikācijai

Šis modelis jāuzskata par papildinājumu pamata *InClaS* modelim un praktiskās realizācijas ieviešanā ir lietojams kopā ar to.

4.6. Nodaļas kopsavilkums

Šajā nodaļā ir precizētas un papildinātas interaktīvas klasifikācijas sistēmas sastāvdaļas daudzkategoriju klasifikācijas uzdevumiem. Nodaļā **ieviesti svarīgi tālākā darbā izmantoti jēdzieni**: Daļēji pareizi vai pilnīgi pareizi klasificēts piemērs (DP), Nepareizi klasificēts piemērs (N), Īsti neskaidra klasifikācija (ĪN), Nepatiesi neskaidra klasifikācija (NN), kā arī **mēri eksperta ieguldītā darba novērtēšanai**: $D_{nelietderīgais}$ - cik pareizi klasificētu piemēru ekspertam jācaurskata, lai klasificētu vienu nepareizi klasificētu piemēru; $D_{kopējais}$ - cik piemēru ekspertam pavisam jācaurskata.

Ir izstrādāts **algoritms neskaidri klasificētu piemēru noteikšanai daudzkategoriju klasifikācijas gadījumā**, kurš nosaka, ka piemērs tiek atzīts par neskaidru tad, ja ar noteiktu (noklusēto vai izvēlēto) pārliecības sliekšņa lielumu piemēram nav prognozēta neviena no iespējamajām klasēm.

Specifiska interaktīvai klasifikācijas sistēmai ir nepieciešamība noteikt sliekšņa lielumu, pie kura klasifikatora iegūtais rezultāts vairs netiek uzskatīts par uzticamu.

Piemērotākā sliekšņa noteikšanas metode palīdz atrast šo lielumu katrai datu kopai. Kā pārmeklējamā apgabala indikatori ieviesti mēri vidējā klasifikatora pārliecība par klasēm, kurām piemēri ir piederīgi (VPP) un vidējā klasifikatora pārliecība par klasēm, kurām piemēri nav piederīgi (VPN). Definējot metodi, secināts, ka sliekšņa lieluma noteikšana:

- jāveic individuāli katram uzdevumam;
- par sasniedzamo mērķi nosaka samazināt nepareizi klasificēto piemēru skaitu, ņemot vērā arī papildu ierobežojumus, kas limitē eksperta ieguldīto darbu:
  - kāds ir maksimālais piemēru skaits, ko eksperts apņemas klasificēt;
  - cik pareizu piemēru eksperts ir gatavs caurskatīt, lai sniegtu klasifikāciju nepareizi klasificētajiem vai neklasificētajiem piemēriem.

Sistēmas projektēšana studiju priekšmetu salīdzināšanai iekļauj pieņemto lēmumu aprakstu atbilstoši projektēšanas soļiem un klasifikācijas sistēmu veidojošo moduļu realizācijas detaļas.

Iegūtais *InClaS* pamatmodelis un tā papildinājums ar daudzkategoriju klasifikācijas nodrošināšanai nepieciešamajām komponentēm sniedz pietiekamu teorētisko un metodisko bāzi interaktīvas klasifikācijas sistēmas realizācijai programmatūras veidā. Darba nākamajā nodaļā tiks raksturota *InClaS* realizācija un prototipa izveide, kas vēlāk kalpos par pamatu eksperimentu veikšanai un paša modeļa pārbaudei.

5. *INCLAS* PROTOTIPS

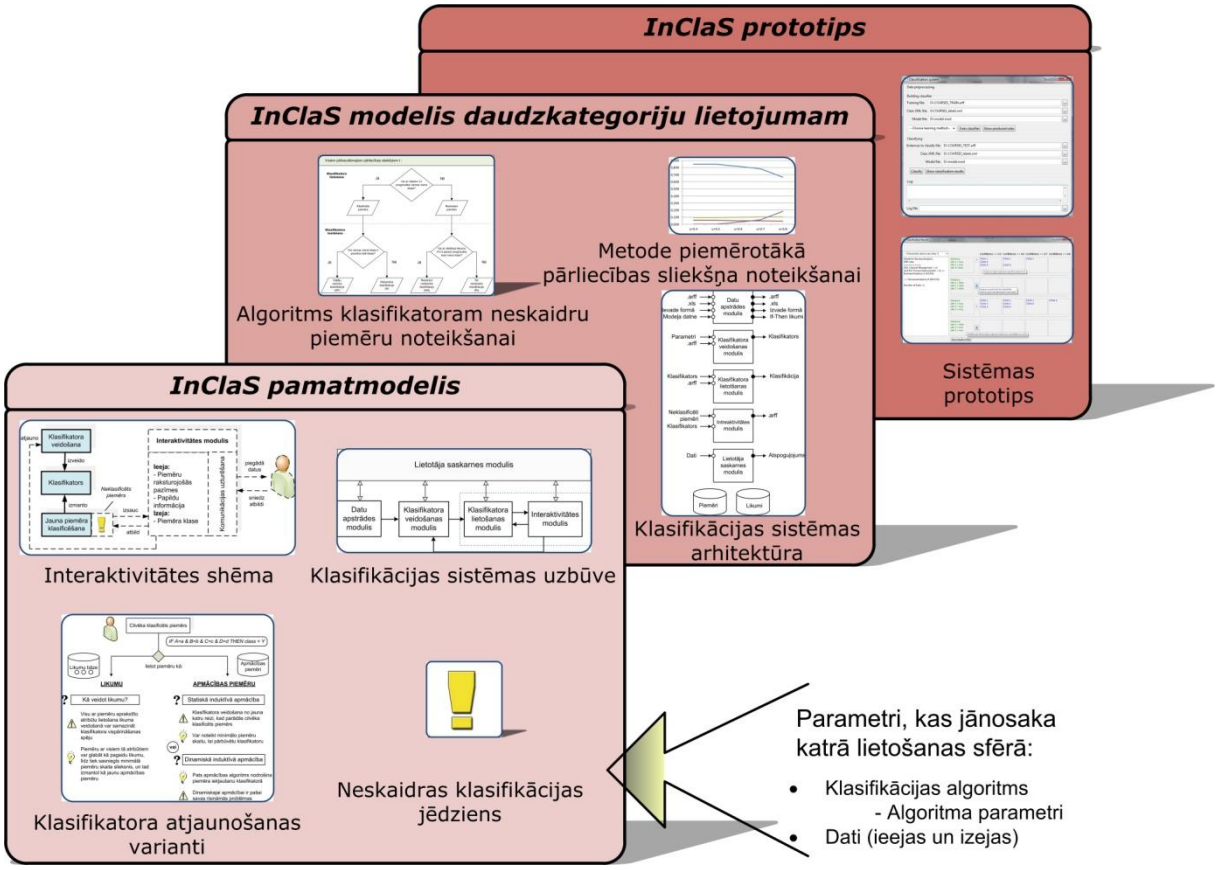
Šī darba nodaļa sniegs būtiskākās detaļas par *InClaS* modeļa praktisko realizāciju programmatūras prototipa veidā. Vispirms 5.1. apakšnodaļā tiks apkopotas līdz šim izstrādātās *InClaS* modeļa komponentes, organizējot tās trīs līmeņos, kur pirmais līmenis definē *InClaS* pamatmodeli, otrais līmenis to papildina ar daudzkategoriju klasifikācijas nodrošināšanai paredzētajiem elementiem, bet trešais līmenis, kurš tiks aprakstīts šajā nodaļā, atbild par koncepciju realizāciju interaktīvas klasifikācijas sistēmas prototipā. 5.2. apakšnodaļā sniegts prototipa funkcionalitātes apraksts, pievēršot uzmanību tam, kā atsevišķās *InClaS* komponentes ir realizētas programmatūrā, un 5.3. apakšnodaļā izskaidrots, kā izstrādātais sistēmas prototips atšķiras no klasifikācijas uzdevumiem plaši izmantotā rīka *Weka* un daudzkategoriju bibliotēkas *Mulan*. 5.4. apakšnodaļā sniegts ieskats prototipa lietošanas ekrāna formās, nodaļu noslēdzot ar kopsavilkumu.

5.1. *InClaS* trīs līmeņi

Interaktīvas klasifikācijas sistēmas modeļa *InClaS* galvenās komponentes un izstrādnes atspoguļotas 5.1. attēlā. *InClaS* pamatmodelis definē algoritmus, arhitektūras un vadlīnijas, kas ļauj izstrādāt interaktīvu klasifikācijas sistēmu nepareizi klasificēto objektu skaita samazināšanai jomās, kur klasifikācijas laikā pieejams eksperts. *InClaS* modelis konceptuālā līmenī (1) definē realizējamo interaktivitātes shēmu, (2) raksturo klasifikācijas sistēmas uzbūvi – funkcionālos moduļus un to sasaistes, (3) izskaidro iespējamos klasifikatora atjaunošanas (papildināšanas) variantus pēc eksperta veiktas klasifikācijas un (4) definē neskaidras klasifikācijas jēdzienu. Ir identificēti parametri, kas jānosaka katrā *InClaS* lietošanas sfērā. Klasifikācijas algoritma un tā iespējamo parametru izvēle ir jāveic ikvienā jomā, kur ir plānots izmantot klasifikāciju, un nav jaunums arī interaktīvas sistēmas kontekstā. Sistēmas izveide konkrētai problēmsfērai nevar tikt iepriekš pilnībā definēta, bet tā ir atbalstīta ar izskaidrotu piecu soļu metodi intelektuālu sistēmu projektēšanā [127], kas atvieglo darbu klasifikācijas sistēmas izstrādē (skat. 3.2.1. sadaļu).

Darba ietvaros tālāka pieejas specificēšana veikta daudzkategoriju klasifikācijas gadījumiem, (1) detalizējot sistēmas arhitektūru gan no projektēšanas, gan iegūtā projektējuma puses, (2) sniedzot algoritmu klasifikatoram neskaidru piemēru noteikšanai, ja objektam var būt vairākas klases, kā arī (3) aprakstot metodi piemērotākā pārliecības sliekšņa noteikšanai, pie kura jaunos piemērus atzīst par klasifikatoram neskaidriem un nodod eksperta pārziņā.

Izmantojot pirmajos divos līmeņos iegūtos teorētiskos rezultātus, nākamais līmenis realizē interaktīvas klasifikācijas sistēmas prototipu, izveidojot praktisko bāzi eksperimentālai modeļa pārbaudei konkrētās problēmsfērās.



5.1. att. *InClaS* modelis trīs detalizācijas līmeņos

Prototips ir izmantojams dažādām daudzkategoriju klasifikācijas problēmām, taču tas ir pielāgots ērtākai lietošanai universitātes studiju priekšmetu salīdzināšanas uzdevumam lietotājam sniedzamās informācijas atspoguļojuma ziņā.

5.2. *InClaS* modeļa realizācija prototipā

Tā kā *InClaS* modelis nesniedz tikai konkrētus ieviešamos algoritmus, bet arī ieteikumus, kurus praksē var realizēt dažādi, tad šajā apakšnodaļā tiks paskaidrots sīkāk, kā modelis ir ieviests prototipā.

- Prototipa ietvaros tiek izmantoti iepriekš realizēti klasifikācijas algoritmi; bāzes algoritmi tiek izsaukti no programmatūras *Weka* [76], bet tos izmantojošās daudzkategoriju klasifikācijas metodes aprakstītas bibliotēkā *Mulan* [138]. Sistēmā šobrīd ieviesti 11 statiskie klasifikācijas algoritmi no *Weka* un *Mulan* klāsta, izmantojot to noklusētos parametrus.

- Interaktivitātes shēma realizēta, jaunu piemēru klasifikācijas procedūru papildinot ar iespēju izsekot nepārliecinoši klasificētus piemērus - klasificēšanas brīdī pārbaudot, ar kādu pārliecību veikta klasifikācija, un demonstrējot rezultātus lietotājam. Atbilstoši, lietotājs var izvērtēt ar dažādām pārliecības pakāpēm piešķirtās klases un sniegt savu klasifikāciju gadījumos, kad neviena klase nav piešķirta ar pārliecību 0.5 vai vairāk.
- No klasifikatora atjaunošanas variantiem tiek izmantota *Uz slieksni balstītā statiskā pieeja* ar slieksni 1. Tas nozīmē, ka klasifikators tiek atjaunots pēc katra jauna eksperta klasificēta piemēra parādīšanās. Praktiski, ja vienā jaunu piemēru klasifikācijas reizē tiek klasifikācijai nodoti vairāki piemēri un arī eksperts klasificē vairāk par vienu piemēru, tad klasifikators tiks atjaunots, izmantojot reizē visus jaunos piemērus.
- Nodrošinātas datu ievades iespējas un sistēmas iegūto rezultātu izvade ekrāna formā.
- Algoritms klasifikatoram neskaidru piemēru noteikšanai ir pilnībā realizēts sistēmas prototipā klasifikatora lietošanas modulī. Jaunu piemēru klasifikācijas laikā tiek iegūtas klasifikatora pārliecības par visām piešķiramajām klasēm, kuras tālāk tiek atbilstoši izmantotas lēmumu pieņemšanai.
- Sistēma izgūst un saglabā teksta datnē lietotājam lasāmā formā klasifikatoru veidojošos likumus.
- Strādājot eksperimentu veikšanas režīmā, sistēmai nav pilnvērtīga lietotāja saskarne, bet ir pieejamas plašākas iespējas, tajā skaitā 20 algoritmu pārbaude (ko var vēl paplašināt ar citām metodēm un algoritmiem) un iespēja iegūt nepieciešamos sistēmas darbības rezultātus dažādiem novērtējumiem. Iegūstami rezultāti gan par populārākajiem daudzkategoriju klasifikācijas novērtējumu mēriem, gan aprēķini darba autores izvirzītajiem mēriem – DP, N, ĪN, NN, VPP un VPN.
- Piemērotākā pārliecības sliekšņa noteikšana ir veicama ar manuālo metodi.

5.3. Prototipa jaunieviesumi, salīdzinot ar *Weka* un *Mulan*

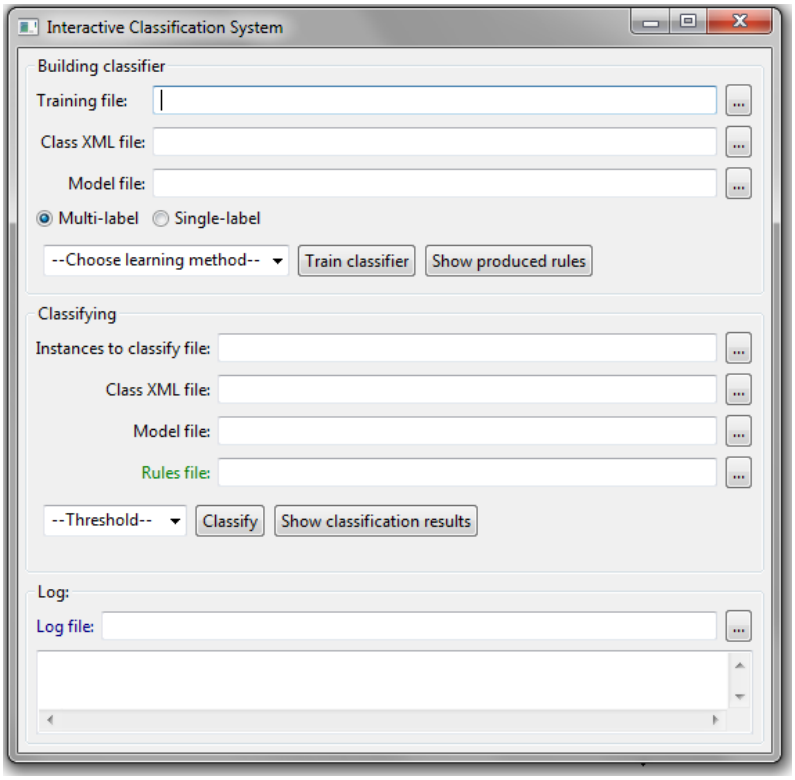
Lai piedāvāto sistēmas modeli pārbaudītu un interaktīvo klasifikācijas sistēmu ieviestu praktiskā lietošanā, ir izstrādāts *InClaS* prototips, kurš arī tiks izmantots eksperimentu veikšanai. Tas balstās uz esošo induktīvās apmācības algoritmu un dažādu novērtēšanas metriku izmantošanu no programmatūras *Weka,* tās paplašinājuma bibliotēku daudzkategoriju klasifikācijas veikšanai *Mulan* un autores izstrādātajām interaktivitātes nodrošināšanas metodēm un lietotāja saskarni. Lai saglabātu integritāti un atvieglotu esošo risinājumu izmantošanu, tāpat

kā *Weka* un *Mulan*, *InClaS* prototips ir izstrādāts programmēšanas valodā *Java*. Būtiskākie jaunieviesumi, kas papildina *Weka* un *Mulan* funkcionalitāti, ir šādi:

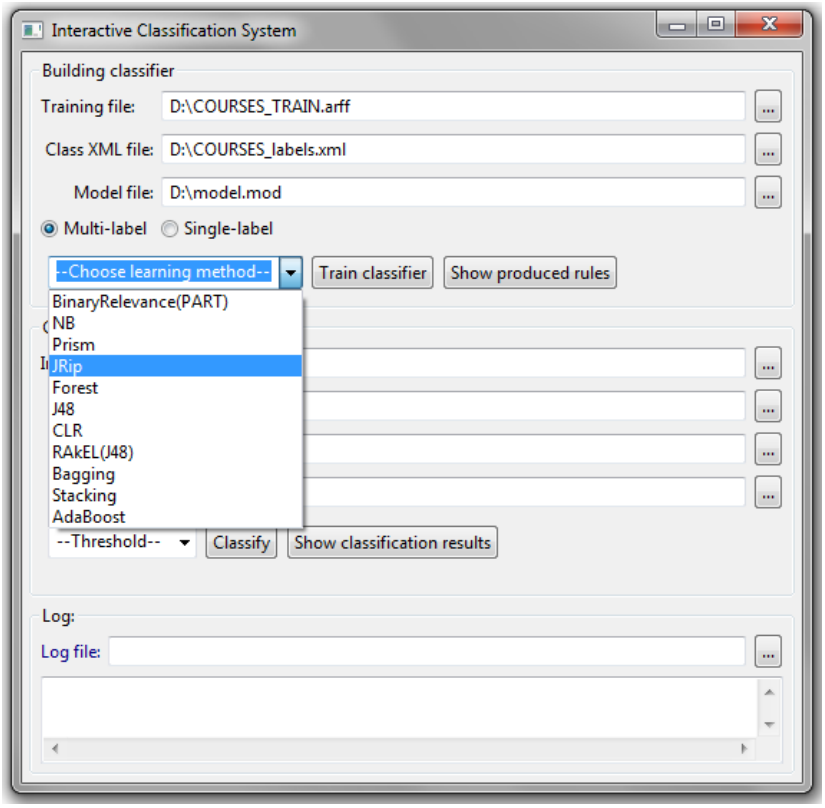
- lietotāja saskarne daudzkategoriju klasifikācijas bibliotēkas *Mulan* izmantošanai (*Mulan* izstrādātāji nepiedāvā savu grafisko lietotāja saskarni);
- likumu kopas (klasifikatora modeļa) uzskatāma attēlošana lietotājām (ja izvēlētais algoritms producē interpretējamu klasifikatoru);
- daļēji pareizo, nepareizo, īsti neskaidro un nepatiesi neskaidro klasifikāciju uzskaite klasifikatora testēšanas laikā;
- pārliecības līmeņa noteikšana un atspoguļošana katrai piemēram piešķirtajai klasei klasifikatora testēšanas un lietošanas laikā;
- eksperta sniegtās klasifikācijas apstrāde neskaidrajiem piemēriem, apmācības kopas un klasifikatora atjaunošana;
- grafiska lietotāja saskarne interaktivitātes nodrošināšanai (klasifikāciju pieņemšanai no eksperta un papildinformācijas sniegšanai par klasificējamo piemēru) ir izstrādes procesā un pagaidām nodrošina daļēju funkcionalitāti.

5.4. Prototipa demonstrācija

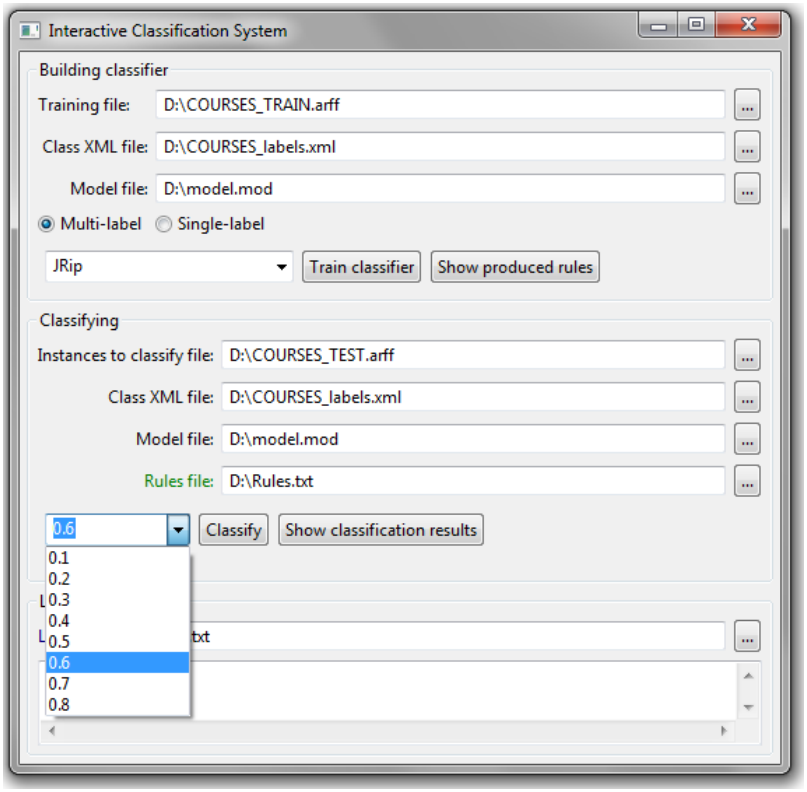
Turpmākajos attēlos tiks atspoguļots interaktīvās klasifikācijas sistēmas prototips: ekrānattēli un paskaidrojumi. Sākot darbu ar sistēmu, tiek atvērts galvenais logs (skat. 5.2. att.), kurā ir ievadāmi dažādi datņu nosaukumi un darbības parametri. 5.3. attēlā redzams, kā tiek ievadīts apmācības datus saturošās datnes nosaukums, klases aprakstošā *XML* datne un klasifikācijas modeļa saglabāšanas vieta. Lietotājam jāizvēlas viens no 11 klasifikācijas algoritmiem un jānospiež poga „*Train classifier*”. 5.4. attēlā redzams, kā tiek ievadīti parametri iegūtā klasifikatora lietošanai. Tiek norādīta datne, kur ievietot izgūtos likumus, un ievadīts pārliecības sliekšņa lielums, pie kura piemēru atzīt par neskaidru. 5.5. attēlā redzams dažādu klasifikācijas novērtēšanas parametru izvades logs. Šī informācija tiek izmantota eksperimentos, piemēram, piemērotākā pārliecības sliekšņa noteikšanai. Klasifikatora iegūto likumu piemērs redzams 5.6. attēlā.



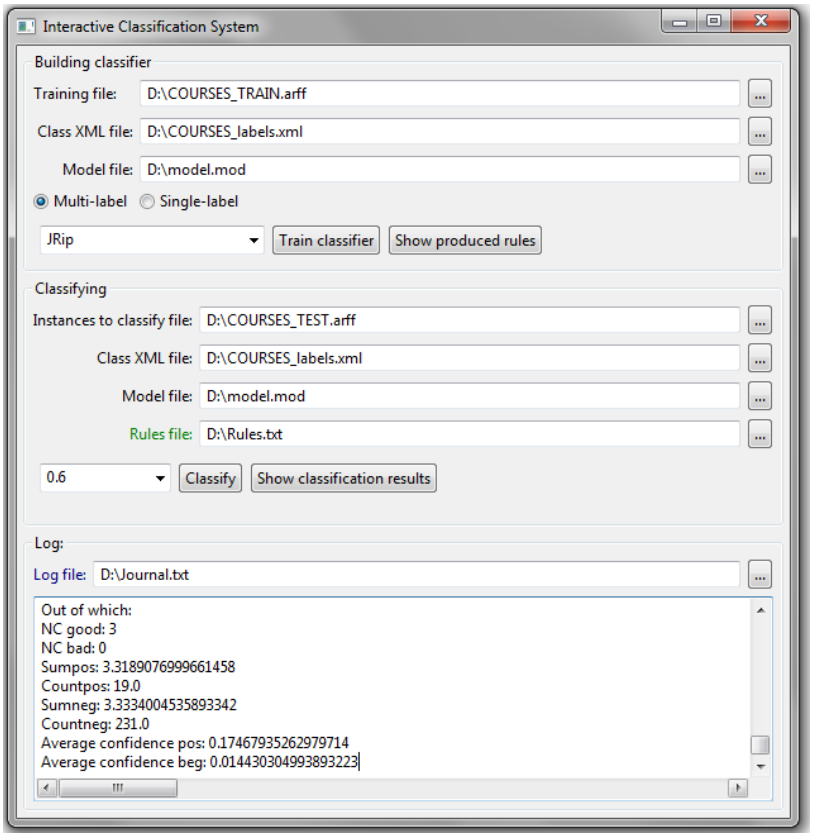
5.2. att. Sistēmas galvenais logs



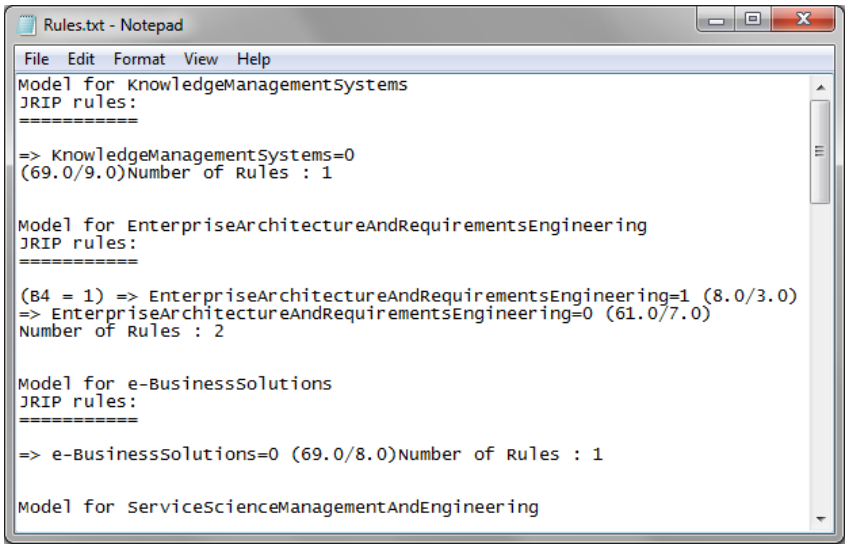
5.3. att. Klasifikatora apmācība



5.4. att. Klasifikatora testēšana

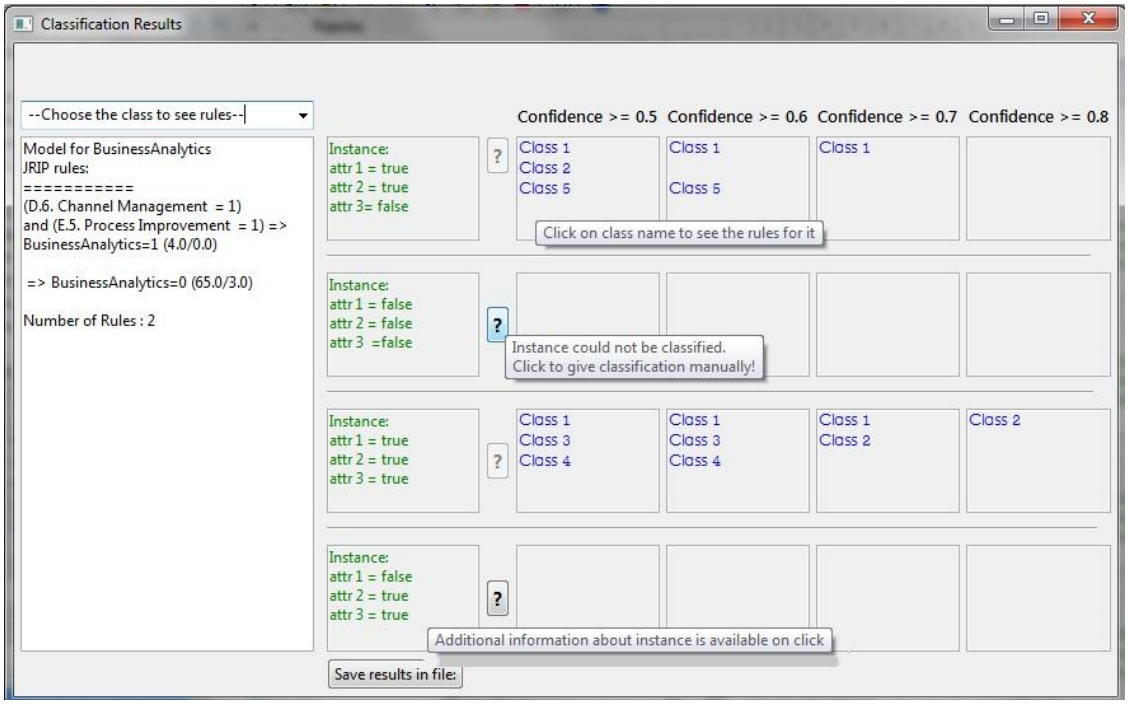


5.5. att. Klasifikatora testēšanas rezultātu izvade



5.6. att. Klasifikatora iegūtie likumi

5.7. attēlā redzamajā klasifikācijas rezultātu logā lietotājam ir iespēja ne tikai apskatīt iegūtos likumus katrai klasei un konkrētiem objektiem noteiktās klases (ar dažādu pārliecības pakāpi), bet arī sīkāk izprast klasifikatora darbību. Noklikšķinot uz kādas klases nosaukuma (attēlā ar zilas krāsas burtiem) rezultātu sarakstā, tiek parādīti tieši šo klasi aprakstošie likumi. Noklikšķinot uz objekta nozīmīgāko atribūtu apraksta (attēlā ar zaļiem burtiem), atveras papildu logs ar plašāku pieejamo informāciju par objektu. Studiju priekšmeta gadījumā tas ir dokuments ar priekšmeta aprakstu.



5.7. att. Klasifikācijas rezultātu logs

Ja objektam neviena klase nav noteikta ar 0.5 vai lielāku pārliecību, tad šo gadījumu ir tiek piedāvāts klasificēt lietotājam (aktivizējas „?“ zīme pie objekta apraksta). Lietotājs var apskatīt papildu informāciju par objektu un atzīmēt klases, kuras, viņaprāt, šis objekts pārstāv.

Interaktīvas klasifikācijas sistēmas prototips izstrādāts kā darbvirsmas lietotne.

5.5. Nodaļas kopsavilkums

Lai iepriekšējās darba nodaļās piedāvāto *InClaS* modeli pārbaudītu un interaktīvo klasifikācijas sistēmu ieviestu praktiskā lietošanā, ir izstrādāts *InClaS* prototips. Šajā darba nodaļā sniegts *InClaS* modeļa praktiskās realizācijas apraksts. Apkopotas ieviešamās ***InClaS* modeļa komponentes, organizējot tās trīs līmeņos**, kur pirmais līmenis definē *InClaS* pamatmodeli, otrais līmenis to papildina ar daudzkategoriju klasifikācijas nodrošināšanai paredzētajiem elementiem, bet trešais līmenis atbild par interaktīvas klasifikācijas sistēmas prototipu. Sniegts **prototipa funkcionalitātes apraksts**, pievēršot uzmanību tam, kā atsevišķās *InClaS* komponentes ir realizētas programmatūrā.

Uzsverot izstrādnes jauninājumus, paskaidrots, kā šis sistēmas prototips atšķiras un papildina klasifikācijas uzdevumiem plaši izmantoto rīku *Weka* un daudzkategoriju bibliotēku *Mulan*. Par galvenajām papildinošajām atšķirībām jāmin (1) izstrādā lietotāja saskarne daudzkategoriju klasifikācijas bibliotēkas *Mulan* izmantošanai (*Mulan* izstrādātāji nepiedāvā savu grafisko lietotāja saskarni), (2) iespēja lietotājam ērti apskatīt klasifikatora iegūto likumu kopu (ja izvēlētais algoritms producē interpretējamu klasifikatoru), (3) lietotāja saskarne interaktivitātes nodrošināšanai. Tas kopumā veido unikālu vidi daudzkategoriju klasifikācijas nodrošināšanai lietotājam ērtākā formā, kā bija iespējams ar pastāvošajiem rīkiem, kā arī līdz šim nebijušas interaktivitātes iespējas starp klasifikācijas sistēmu un lietotāju. Atsevišķa apakšnodaļa veltīta ieskata sniegšanai par prototipu no sistēmas lietotāja puses.

Izstrādātais prototips ir galvenā programmatūras bāze praktisko eksperimentu veikšanai, ar kuru palīdzību ir pārbaudīts *InClaS* modelis un piedāvātā interaktīvā pieeja. Eksperimenti tiks atspoguļoti darba nākamajā nodaļā.

6. INCLAS MODEĻA NOVĒRTĒJUMS

Šī nodaļa apraksta eksperimentu plānu un iegūtos rezultātus darbā ar interaktīvas klasifikācijas sistēmas prototipu daudzkategoriju klasifikācijas uzdevumos. Eksperimenti ir veikti ar mērķi pārbaudīt interaktīvās pieejas un *InClaS* modeļa lietderību un sistēmas prototipa lietojamību, kā arī apstiprināt izvirzītās tēzes (T1, T2 un T3). Atgādinot problēmapgabalu, kas tiek risināts promocijas darbā - no mašīnapmācības viedokļa tiek risināts uzdevums, kuram raksturīgas šādas īpašības: (1) iegūtie rezultāti ir jāsaprot klasifikatora lietotajam un ekspertam, (2) pieejamā apmācības kopa ir maza, (3) sākotnējie dati ir daļēji strukturēti vai nestrukturēti, (4) problēmsfērā ir raksturīgas daudzas klases, kuras sastopamas vienlīdz bieži, (5) objekts var piederēt vienlaicīgi vairākām klasēm.

Aprobācijai izmantotās problēmsfēras ir studiju priekšmetu salīdzināšana (6.1. apakšnodaļā) un diagnostika medicīnā (6.2. apakšnodaļā).

Ar eksperimentu palīdzību tiek pārbaudīti šādi aspekti *InClaS* lietderības novērtēšanai:

- *nepareizi klasificēto piemēru skaita salīdzināšana*, izmantojot klasisko neinteraktīvo klasifikācijas pieeju un piedāvāto interaktīvo pieeju (attiecas uz T1);
- *piemērotākā pārliecības sliekšņa lieluma noteikšanas metodes* pārbaude, atrodot atbilstošāko pārliecības slieksni, kur nepareizi klasificēto piemēru skaits *N* ir minimāls pie izvirzītajiem eksperta ieguldāmā darba ierobežojumiem (attiecas uz T2).

Attiecībā uz risinājuma lietderību izglītības sfērā:

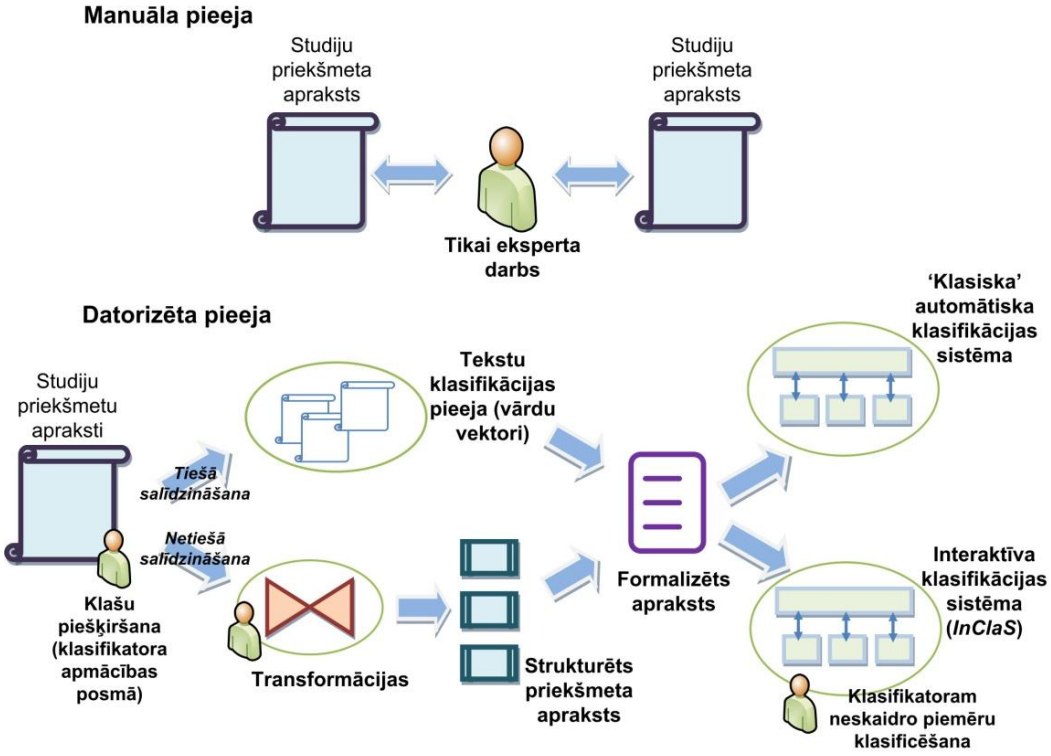
- pārbaude darbā izteiktajam apgalvojumam, ka šī problēmsfēra nav piemērota tradicionāliem mašīnapmācības risinājumiem, bet uz *induktīvo apmācību balstīta, interaktīva, daudzkategoriju klasifikācijas sistēma studiju priekšmetu salīdzināšanas atbalstam* var dot pieņemamu risinājumu (attiecas uz T3);
- studiju priekšmetu *tiešās un netiešās salīdzināšanas novērtēšana* – izmantojot priekšmetu aprakstus pilnībā vai veicot pastarpinātu salīdzināšanu caur *e-CF* ietvara kompetencēm.

Daļa no eksperimentu rezultātiem atspoguļoti autores publikācijā [139].

6.1. Eksperimenti izglītības jomā

6.1. attēlā ir shematiski parādīti vairāki veidi, kā studiju priekšmeti var tikt salīdzināti. Manuālā pieejā vienīgi cilvēks – eksperts lieto savas zināšanas problēmsfērā un spriež par priekšmetu atbilstību. Datorizētajās pieejās eksperta sākotnēji ieguldītais darbs tiek izmantots

automātisku vai interaktīvu klasifikācijas sistēmu izveidei. Attēlā atspoguļoti divi veidi formalizēta un klasifikācijas algoritmiem piemērota ieejas datu formāta iegūšanai – (1) caur daļēji strukturētu priekšmetu aprakstu tekstu tiešu izmantošanu un (2) strukturētu semantiski nozīmīgu daļu izgūšanu no pieejamajiem priekšmetu aprakstiem, izmantojot pastarpinātu unifikācijas ietvaru. Vienā vai otrā veidā iegūto formalizēto aprakstu iespējams apstrādāt ar 'klasisku' neinteraktīvu klasifikācijas sistēmu vai ar šajā darbā piedāvāto interaktīvo klasifikācijas sistēmu. Tādējādi datorizētajai pieejai realizējamas 4 kombinācijas: (1) teksta klasifikācija ar 'klasisko' klasifikāciju, (2) strukturētā aprakstu pieeja ar 'klasisko' klasifikāciju, (3) teksta klasifikācija ar *InClaS* un (4) strukturētā aprakstu pieeja ar *InClaS*.



6.1. att. Studiju priekšmetu manuālas un automatizētas klasifikācijas iespējas

Formalizētais studiju priekšmetu apraksts sniegts 6.1. tabulā. Atgādinot lietotos apzīmējumus:

$K = \{k_1, ..., k_j\}$: klašu kopa, j: klašu skaits,

$X = \{x_1, ..., x_i\}$: datu kopa, i: piemēru skaits,

$x_i = \{(a_1, v_{a1}), ..., (a_n, v_{an})\}$: datu kopas objekts (piemērs), atribūtu-vērtību pāru vektors,

n – atribūtu skaits,

v_a – atribūta a vērtība.

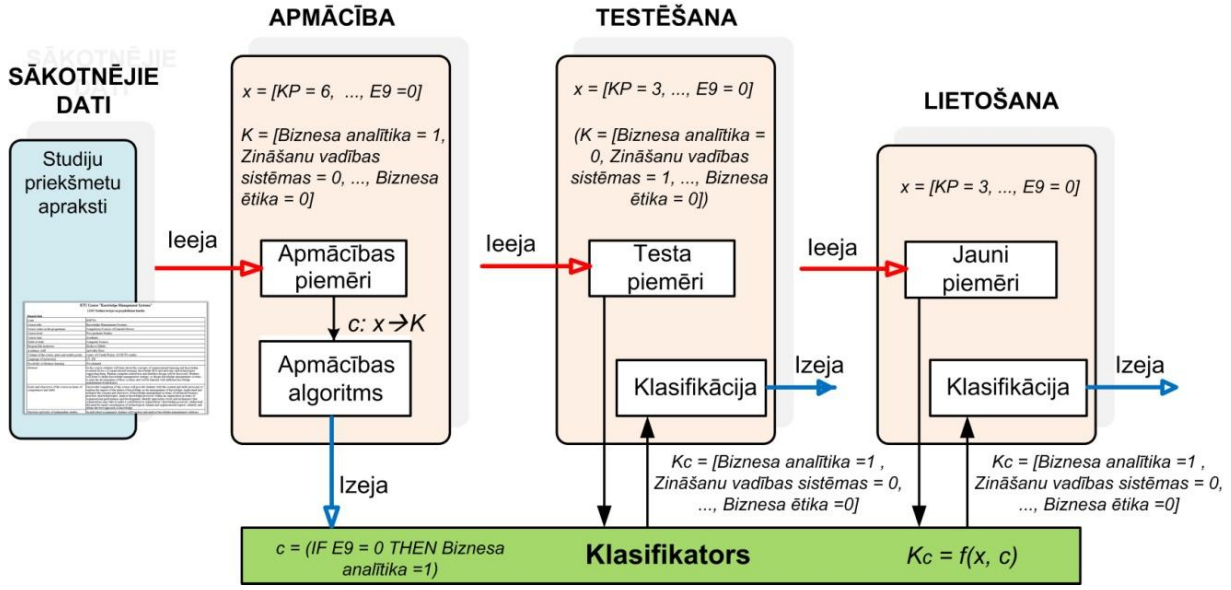
Par mērķa priekšmetiem, ar ko salīdzināt citus priekšmetus, jeb klasēm ir izmantoti 25 RTU maģistra studiju programmas *Biznesa informātika* priekšmeti.

6.1. tabula

Atribūti un klases tiešās un netiešās studiju priekšmetu salīdzināšanas gadījumā

Atribūti a	**Iespējamās vērtības** v_a	**Datu tips**	
Atribūti netiešās studiju priekšmetu salīdzināšanas gadījumā *(n = 38)*			
Kredītpunktu skaits (ECTS)	[3; 6; 9; 15]	Nominmāls	
Studiju līmenis	[bakalaura; maģistra]	Nominmāls	
Kompetence A1	[0; 1]	Nominmāls	36 atribūti
Kompetence A2	[0; 1]	Nominmāls	
..			
Kompetence E9	[0; 1]	Nominmāls	
Atribūti tiešās studiju priekšmetu salīdzināšanas gadījumā - vārdi priekšmetu aprakstos *(n = 1884)*			
attrib	[0; 1]	Nominmāls	1884 atribūti
compliance	[0; 1]	Nominmāls	
..			
model-diven	[0; 1]	Nominmāls	
Klases K *(j = 25)*			
Biznesa analītika	[0; 1]	Nominmāls	25 klases
Zināšanu vadības sistēmas	[0; 1]	Nominmāls	
..			

Klasifikatora izveides formāls atspoguļojums atbilstoši 2.3. attēlā sniegtajiem klasifikatora izveides posmiem un notācijai studiju priekšmetu netiešās salīdzināšanas gadījumam demonstrēts piemērā 6.2. attēlā.



6.2. att. Klasifikatora iegūšanas un lietošanas piemērs netiešajā priekšmetu salīdzināšanā

Studiju priekšmetu salīdzināšanas gadījumā izpildāmo eksperimentu vispārīgs plāns ir atspoguļots 6.2. tabulā. Eksperimentos izmantota viena un tā pati sākotnējā datu kopa, tas ir, vienu un to pašu priekšmetu apraksti. 1. un 3. variantā no studiju priekšmetu aprakstiem automātiski ir iegūti vārdu vektori, kuriem veikta vārdu celmu iegūšana (ang.v. – *stemming*), lai

izvairītos no vārda dažādu locījumu atsevišķas izmantošanas, un vārdu salikumu iegūšana. Priekšapstrādei izmantotas programmatūras *Weka* iebūvētās funkcijas (detalizētus eksperimenta parametrus skatīt darba 2. pielikumā). Jārēķinās, ka priekšmetu apraksti ir dažādas formas un satura un netiek veikta nekāda sadaļu diferencēšana. 2. un 4. variantā no studiju priekšmetu aprakstiem eksperts ir identificējis kompetences, kuras, atbilstoši Eiropas e-kompetenču ietvaram 2.0, šis priekšmets sniedz. Tā kā priekšmetu apraksti ir atšķirīgi savā detalizētībā un apraksta stilā un eksperts, kurš veic atbilstību noteikšanu, arī var būt tendenciozs, tad nevar garantēt absolūti korektu studiju priekšmetu novērtējumu pret kompetencēm. Šis apstāklis arī ir viens no problēmsfēras sarežģītības faktoriem. Identificētās kompetences, studiju priekšmeta apjoms kredītpunktos un studiju līmenis, kurā priekšmets tiek pasniegts, ir klasifikatora ieejas dati. Formu, kuru aizpildot, tiek iegūti klasifikatora ieejas dati 2. un 4. varianta eksperimentiem, var atrast darba 3. pielikumā.

6.2. tabula

Eksperimentu plāns studiju priekšmetu salīdzināšanai

	1. variants	**2. variants**	**3. variants**	**4. variants**
Ieejas datu kopa	Studiju priekšmetu apraksti pilnā apjomā (priekšapstrādē iegūstot vārdu vektorus)	Studiju priekšmetu kompetences (sagaidāmie mācību rezultāti), kredītpunktu skaits, studiju līmenis	Studiju priekšmetu apraksti pilnā apjomā (priekšapstrādē iegūstot vārdu vektorus)	Studiju priekšmetu kompetences (sagaidāmie mācību rezultāti), kredītpunktu skaits, studiju līmenis
Klasifikācijas pieeja	Automātiska klasifikācija	Automātiska klasifikācija	Interaktīva klasifikācija (*InClaS* modelis)	Interaktīva klasifikācija (*InClaS* modelis)
Izmantotās metodes	20 klasifikācijas metodes (no programmatūras Weka un bibliotēkas Mulan klāsta)	20 klasifikācijas metodes (no programmatūras Weka un bibliotēkas Mulan klāsta)	4 metodes, kas uzrāda labākos rezultātus no 1. varianta eksperimentiem	4 metodes, kas uzrāda labākos rezultātus no 2. varianta eksperimentiem
Novērtēšanas parametri	Haminga zaudējums (*Hamming loss*), Mikro-vidējā precizitāte (*Micro-average precision*), Mikro-vidējais atsaukums (*Micro-average recall*), Viena kļūda (*One-error*), Pārklāšana (*Coverage*)	Haminga zaudējums, Mikro-vidējā precizitāte, Mikro-vidējais atsaukums, Viena kļūda, Pārklāšana	Daļēji pareizo, nepareizo, īsti neskaidro, nepamatoti neskaidro klasifikāciju skaits	Daļēji pareizo, nepareizo, īsti neskaidro, nepamatoti neskaidro klasifikāciju skaits

6.1.1. Eksperimentu novērtēšanai izmantotās metrikas

Kā jau izklāstīts darba 2.1.4. sadaļā, daudzkategoriju klasifikācijas uzdevumos metožu novērtēšanas mēri atšķiras no vienas kategorijas uzdevumiem [65]. Klasifikācijas rezultātu novērtējumam ir izvēlētas piecas populāras metrikas no iepriekš aprakstītajām. Tās ir:

Haminga zaudējuma funkcija – uz piemēriem balstīts mērs, kurš izsaka procentuāli, cik klases ir noteiktas nepareizi.

Mikro-vidējā precizitāte un **mikro-vidējais atsaukums** – vienkategorijas klasifikācijas mēriem – precizitātei un atsaukumam - atbilstošas funkcijas daudzkategoriju gadījumā

Viena kļūda – uz ranžēšanu balstīts mērs, kurš raksturo, cik reizes visaugstāk vērtētā klase īstenībā nav starp īstajām piemēra klasēm

Pārklāšana – vidēji cik tālu ranžētā klašu sarakstā ir jāmeklē, lai pārklātu visas piemēra īstās klases.

Šīs metrikas tiks izmantotas, lai vispārēji novērtētu izmantotos algoritmus un izvēlētos tos, kuru veiktspēja konkrētās problēmas risināšanā ir labākā. Interaktīvās pieejas novērtēšanai tiek lietoti mēri, kas tika izklāstīti 4.1., 4.2. un 4.3. apakšnodaļā:

VPP (formula 4.1.);

VPN (formula 4.2.);

$\mathbf{D}_{kopējais}$ (formula 4.3.);

$\mathbf{D}_{nelietderīgais}$ (formula 4.4.);

daļēji pareizi vai pilnīgi pareizi klasificēto piemēru skaits (DP);

nepareizi klasificēto piemēru skaits (N);

īsti neskaidro klasifikāciju skaits (ĪN);

nepatiesi neskaidro klasifikāciju skaits (NN).

6.1.2. Eksperimentu parametri

Pilna izmantotā datu kopa, kas balstīta uz kompetencēm, sastāv no 79 piemēriem, kas raksturo dažādu Eiropas universitāšu studiju priekšmetu atbilstību RTU *Biznesa informātikas* priekšmetiem. Tātad 25 piemēri ir pašas RTU programmas priekšmetiem, 6 piemēri no Rostokas universitātes, 31 piemērs no Vīnes tehniskās universitātes un 17 no Vīnes universitātes. Katrs piemērs netiešās salīdzināšanas gadījumā ir aprakstīts ar 38 nomināliem atribūtiem, 36 no tiem apzīmē noteiktu kompetenču esamību vai neesamību priekšmeta sasniedzamajos rezultātos, viens definē kredītpunktu skaitu priekšmetam Eiropas kredītpunktu skalā un viens apraksta studiju līmeni. Samazinātajā datu kopā saglabāti piemēri tikai tām klasēm, kas aprakstītas ar

vismaz 4 piemēriem. Savukārt studiju priekšmetu netiešās salīdzināšanas datu kopa sastāv no 131 teksta dokumenta, kas apraksta tieši tos pašus priekšmetus, ko pilnā kompetenču datu kopa. Studiju priekšmetu apraksti nestrukturētu teksta dokumentu veidā (*.txt* datnes) tiek sakārtoti pa mapēm atbilstoši klasēm. Šeit arī rodas 131 piemērs iepriekš minēto 79 vietā, jo piemēru skaits ir mākslīgi jāpalielina, lai katrai klasei būtu sava dokumenta vienība. Līdz ar to apraksti atkārtojas, ja priekšmets attiecas uz vairākām klasēm. Šāda pieeja ir tehnisks risinājums atbilstošas datu kopas iegūšana ar vienkategorijas klasifikācijai paredzētiem līdzekļiem. No datnes ar iegūtajiem vārdu vektoriem, kas arī veido aprakstošos atribūtus, izmantojot speciāli izveidotu sintaktiski pārveidojošu utilītprogrammu, tiek iegūts dokuments, kas ir tālāk izmantojams daudzkategoriju klasifikācijā. Pārveidojošā programma ir raksturota darba 5. pielikumā. Studiju priekšmetu aprakstu transformēšanai klasifikācijas uzdevumam atbilstošā formā – vārdu vektoros – tiek izmantota programmatūra *Weka*. Procesa praktiskās detaļas un izmantotie parametri studiju priekšmetu aprakstu priekšapstrādei ir izklāstīti darba 2. pielikumā. Datu kopu parametri ir apkopoti 6.3. tabulā. Klašu blīvums norāda vidējo viena piemēra klašu skaitu pret kopējo klašu skaitu. Klašu kardinalitāte norāda vidējo vienam piemēram atbilstošo klašu skaitu. Savukārt, klašu kopu skaits norāda, cik daudz atšķirīgu klašu kombināciju ir datu kopā. Šie mēri ļauj spriest par datu kopas daudzkategoritātes iezīmēm. Gan pilnajai, gan samazinātajai datu kopai ir mazs vidējais klašu skaits katram piemēram – 1,6. Tātad vidēji katrs mācību priekšmets ir līdzīgs vienam vai diviem RTU *Biznesa informātikas* priekšmetiem.

6.3. tabula

Studiju priekšmetu datu kopa

	Atribūtu skaits	**Piemēru skaits**	**Klašu skaits**	**Klašu blīvums (*density*)**	**Klašu kardinalitāte (*cardinality*)**	**Klašu kopu skaits (*distinct labelsets*)**
Pilna datu kopa (kompetences)	38	79	25	0.0620	1.6203	52
Samazināta datu kopa (kompetences)	38	64	12	0.1341	1.6094	36
Pilna datu kopa (vārdu vektori)	1884	131 (79)	25	0.0620	1.6203	52

Izmantotās metodes 1. un 2. varianta eksperimentos

Sākotnējie eksperimenti ietver plaša metožu loka izmantošanu studiju priekšmetu salīdzināšanas uzdevumam, lai noteiktu perspektīvākās metodes tālākai lietošanai. Šajā posmā tiek izmantota klasiska automātiska pieeja. Pārbaudēm ir izraudzītas 20 metodes, kas plaši pārklāj daudzkategoriju klasifikācijas pieeju klāstu. Vairāk variāciju ir atvēlēts problēmu

transformācijas, nevis algoritmu transformācijas metodēm. Tāpat pārsvarā lietotas binārās saistības metodes, kas pārveido sākotnējo daudzkategoriju uzdevumu vairākos vienkategorijas uzdevumos un izmanto algoritmus un meta-algoritmus, kas plaši pieejami binārai klasifikācijai, jo šajā problēmsfērā ir pamats uzskatīt, ka tās varētu būt efektīvākās par klašu kopas veidojošām metodēm. Katras klašu kopas traktēšana par jaunu klasi nav perspektīva, jo klašu kopu skaits ir pārāk liels pret apmācības piemēru skaitu (pilnā datu kopā izveidotos 52 klases, kas raksturotas ar 79 piemēriem, tātad reti kurai klasei ir vairāk par vienu piemēru). Turklāt klašu kombinācijas nesniedz raksturīgas iezīmes studiju priekšmetu salīdzināšanā - ja viens priekšmets atbilst vairākiem citiem, netiek iegūta īpaši vērtīga informācija, kā klasifikācijas rezultātu iegūstot vairāku priekšmetu apvienojumu. Par izmantotajiem algoritmiem plašāka informācija ir sniegta darba 9. pielikumā. Ja nav minēts citādi, metodēm izmantoti noklusētie parametri, kas iestatīti *Mulan* bibliotēkā un *Weka* rīkā.

Izmantotās metodes 3. un 4. varianta eksperimentos

Sākotnējo atlasi izgājušās apmācības metodes, kas ieguvušas labākos rezultātus 1. un 2. eksperimentu variantā, tiek pārbaudītas darbībā ar interaktīvo pieeju. Uz kompetencēm balstīto strukturēto aprakstu gadījumā tiek izmantota pilnā un samazinātā datu kopa. Par galveno mēru visos gadījumos tiek lietots nepareizi klasificēto piemēru skaits un salīdzināts ar rezultātiem, ja interaktīvā pieeja netiktu lietota. Papildu aspekts, kas svarīgs šajā problēmsfērā, ir iegūto rezultātu interpretējamība, tāpēc uzmanība tiek pievērsta arī klasifikatora iegūtā modeļa atspoguļojumam.

6.1.3. Eksperimentu rezultāti

Šajā apakšnodaļā ir apkopti iepriekš aprakstīto eksperimentu rezultāti.

1. variants

6.4. tabula atspoguļo rezultātus 20 apmācības metodēm studiju priekšmetu datu kopai, kas iegūta no pilnajiem priekšmetu aprakstiem, veicot 10-kārtu šķērsvalidāciju un neizmantojot interaktivitāti. Labākais un -10% rezultāts no labākā konkrētajam mēram ir izcelts treknrakstā. Jāņem vērā, ka ne visus mērus varēja iegūt (NaN), jo atsevišķos gadījumos metriku aprēķinā iespējams saskarties ar dalījumu ar nulli, kā arī daži mēri nav piemērojami visām metodēm (-).

6.4. tabula

Rezultāti pilnai studiju priekšmetu datu kopai (vārdu vektori)

	Haminga zaudējums	**Mikro-vidējā precizitāte**	**Mikro-vidējais atsaukums**	**Viena kļūda**	**Pārklāšana**
Vēlamais virziens	*Min*	*Max*	*Max*	*Min*	*Min*
BR(NB)	0,0773	0,3203	0,1699	0,8625	14,5071
BR(KStar)	0,1081	0,1299	0,1168	0,8482	15,6625
BR(IBk)	0,1066	0,1325	0,1168	0,8482	15,4982
BR(Bagging)	**0,0655**	NaN	NaN	0,7089	**8,5518**
BR(Stacking)	**0,0645**	NaN	NaN	0,8875	10,6250
BR(AdaBoost)	0,0724	**0,4301**	0,1790	0,7107	**8,5875**
BR(PART)	0,0840	0,3184	**0,2523**	0,7625	13,9393
BR(PRISM)	0,1269	0,1934	**0,2601**	0,7714	13,8071
BR(JRIP)	0,0784	0,3393	0,1740	0,7250	9,5464
BR(REPTree)	**0,0685**	NaN	0,0221	0,8625	10,3911
BR(RF)	**0,0671**	NaN	0,0118	0,8357	12,4857
BR(J48)	0,0890	0,2823	**0,2467**	0,7214	13,5518
CLR	0,0742	0,2867	0,1000	0,8107	10,0196
MLkNN	**0,0655**	NaN	NaN	0,9125	10,5661
MC-Copy	-	-	-	0,7875	11,9018
MC-Ignore	-	-	-	0,8482	14,8304
IncludeLabels	**0,0645**	NaN	0,0000	0,9875	16,2446
RAkEL(J48)	0,0828	0,3333	**0,2507**	**0,6339**	11,7250
LP	0,1089	0,1628	0,1628	0,8482	14,2536
MLStacking	0,1453	0,0791	0,1274	0,9125	15,4446

Rezultāti rāda, ka neviena metode nedemonstrē absolūtu pārākumu pēc visiem parametriem, bet ir 3 metodes, kam ir labākie rezultāti divos parametros – binārās saistības meodes ar meta-algoritmiem *Bagging* (kurš pēc noklusējuma izmanto *REPTree algoritmu*) un *AdaBoost* (kurš pēc noklusējuma izmanto *DecisionStump* algoritmu) un klašu kopas veidojošais *RAkEL* ar *J48* pamatā.

Ja apskata katras metodes iegūtā klasifikatora saprotamību lietotājam, tad *Bagging* un *AdaBoost* nav intuitīvi un viegli uztverami, tomēr ir interpretējami, kamēr *RAkEL* ir klasifikatoru ansambļu metode, kas nesniedz lietotājam klasifikatora spriešanas ceļu.

2. variants

6.5. tabula atspoguļo rezultātus 20 apmācības metodēm pilnajai studiju priekšmetu datu kopai no kompetenču aprakstiem, veicot 10-kārtu šķērsvalidāciju un neizmantojot interaktivitāti. Labākais un -10% rezultāts no labākā konkrētajam mēram ir izcelts treknrakstā.

Arī šajā gadījumā neviena metode nav pārākā pēc visiem parametriem, tomēr ir četras metodes, kurām ir vairāk kā viens labākais rezultāts. Visos šajos gadījumos tie ir algoritmi, kas izmantoti apvienojumā ar binārās saistības metodi. Tālākos eksperimentos arī tiks izmantota

binārās saistības metode ar četriem algoritmiem – *Naivu Beijesu*, *JRip* un *Bagging* un *AdaBoost*. Ar likumu piemēriem šiem izvēlētajiem algoritmiem var iepazīties darba 10. pielikumā. Šie piemēri skaidri norāda uz *JRip* producēto likumu pārākumu to saprotamības un vispārināšanas ziņā, jo, atšķirībā no pārējiem rezultātiem, likumi ir īsi un sistēmas lietotājam viegli lasāmi.

6.5. tabula

Rezultāti pilnai studiju priekšmetu datu kopai (kompetences)

	Haminga zaudējums	**Mikro-vidējā precizitāte**	**Mikro-vidējais atsaukums**	**Viena kļūda**	**Pārklāšana**
Vēlamais virziens	*Min*	*Max*	*Max*	*Min*	*Min*
BR(NB)	0,0829	**0,3908**	**0,1945**	**0,7589**	**8,5214**
BR(KStar)	0,0909	0,2087	0,1117	0,8339	**8,9018**
BR(IBk)	0,1080	0,1873	0,1653	0,8589	13,7446
BR(Bagging)	**0,0650**	NaN	0,0375	**0,6964**	9,8893
BR(Stacking)	0,0649	NaN	0,0000	0,9125	10,4893
BR(AdaBoost)	0,0777	0,2785	0,1547	**0,7357**	**8,4696**
BR(PART)	0,0944	0,1777	0,1313	0,7964	12,3089
BR(PRISM)	0,1106	0,1978	0,1834	0,9357	16,0232
BR(JRIP)	**0,0696**	NaN	0,1696	**0,7089**	10,4196
BR(REPTree)	**0,0670**	NaN	0,0220	0,8232	10,5268
BR(RF)	0,0821	0,1983	0,1267	0,7839	10,2518
BR(J48)	0,0805	0,1283	0,0594	**0,7589**	11,5821
CLR	0,0734	**0,3965**	0,0780	0,8089	9,6643
MLkNN	**0,0654**	NaN	0,0148	0,8089	9,9911
MC-Copy	-	-	-	0,8196	13,8875
MC-Ignore	-	-	-	0,8357	14,3536
IncludeLabels	**0,0649**	NaN	0,0000	0,9875	16,2607
RAkEL(J48)	0,0800	0,1117	0,0523	0,8232	14,2982
LP	0,1046	0,1661	0,1544	0,9625	15,6089
MLStacking	0,1415	0,1486	**0,2081**	0,8804	15,4929

Meklējot iespēju uzlabot sākotnējos klasifikācijas rezultātus, apmācības datu kopā tiek atstātas tikai tās klases, kuras apraksta vismaz četri piemēri. Tādā veidā tiek zaudēti 15 apmācības piemēri, bet arī klašu skaits samazinās līdz 12, palielinot klašu blīvumu (skat. 6.3. tabulu). Rezultāti pēc 10-kārtu šķērsvalidācijas 6.6. tabulā uzrāda mainīgas tendences.

6.6. tabula

Rezultāti samazinātai studiju priekšmetu datu kopai

	Haminga zaudējums	**Mikro-vidējā precizitāte**	**Mikro-vidējais atsaukums**	**Viena kļūda**	**Pārklāšana**
Vēlamais virziens	*Min*	*Max*	*Max*	*Min*	*Min*
BR(NB)	0,1466	0,4183	0,2040	0,6571	**4,2786**
BR(Bagging)	**0,1248**	NaN	0,1227	**0,6024**	5,3929
BR(AdaBoost)	0,1306	NaN	0,2136	0,6690	4,5762
BR(JRIP)	0,1446	**0,5355**	0,2069	0,7476	5,2595

Haminga zaudējuma palielināšanās, salīdzinot ar pilno datu kopu, izskaidrojama ar to, ka skaitliski tāds pats nepareizi noteiktu klašu skaits procentuāli ir kļuvis lielāks, jo ir samazinājies kopējais klašu skaits. Pārklāšanas uzlabojams ir likumsakarīgs - ja jāatrod mazāk klašu, tad to var izdarīt ar mazāku ranžētā saraksta pārmeklēšanu. Līdz ar to nevar viennozīmīgi spriest, ka, izmantojot samazināto datu kopu, kur katru klasi apraksta vairāk piemēru, automātiskas klasifikācijas gadījumā tiktu iegūti labāki klasifikācijas rezultāti.

3. variants

Interaktīvās pieejas salīdzinājums ar tradicionālo klasifikāciju tiešākajā veidā tiks atspoguļots ar pareizi vai daļēji pareizi klasificēto (DP), nepareizi klasificēto (N) piemēru īpatsvaru, kā arī ar klasifikatoram īsti (ĪN) un nepamatoti neskaidrajiem (NN) piemēriem interaktivitātes lietošanas dēļ. Rezultāti ir iegūti, datu kopu 3 reizes sadalot apmācības un testa kopā (69 piemēri apmācībai, 10 piemēri testēšanai). Izmantots pārliecības sliekšņa lielums 0.5. Tabulā redzami rezultāti 4 algoritmiem tekstus izmantojošajai datu kopai. Izvēlēti 3 labākos rezultātus uzrādījušie algoritmi no 1. eksperimenta varianta (*RAkEL, AdaBoost* un *Bagging)*, kā arī *JRip*, jo tas sniedz skaidrus un lietotājam saprotamus likumus.

Starp mēriem pastāv šādas sakarības:

DP + Nepareizi (bez interaktivitātes)=1 (visi automātiskas klasifikācijas rezultāti).
DP+ĪN+NN+ Nepareizi (ar interaktivitāti)=1 (visi interaktīvas klasifikācijas rezultāti).
Nepareizi (bez interaktivitātes)= ĪN+NN+ Nepareizi (ar interaktivitāti).

6.7. tabula

Interaktīvas pieejas lietošana pilnai priekšmetu datu kopai (vārdu vektori)

	Daļēji pareizi (DP)	**Īsti neskaidrs (ĪN)**	**Nepamatoti neskaidrs (NN)**	**Nepareizi (ar interaktivitāti)**	**Nepareizi (bez interaktivitātes)**
RAkEL(J48)	0,267	0,333	0,000	0,400	0,733
BR(AdaBoost)	0,100	0,400	0,000	0,500	0,900
BR(Bagging)	0,067	0,600	0,000	0,333	0,933
BR(Jrip)	0,267	0,367	0,000	0,367	0,733

Rezultāti 6.7. tabulā ir interpretējami šādi. Izmantojot automātisku klasifikāciju, kur klasifikators pilnībā pieņem lēmumu par klašu piešķiršanu, daļēji pareizi tiktu klasificēti 27% piemēru (*RAkEL* gadījumā), bet nepareizi – visi piemēri tabulas zaļajā daļā, summāri veidojot tabulas sarkano daļu - 73%. Ja tiek izmantota interaktivitāte, tad daļēji pareizo klsifikāciju īpatsvars saglabājas tāds pats, bet nepareizi klasificēto piemēru skaits tiek samazināts, jo 33% piemēru tiek atzīti par klasifikatoram neskaidriem un nodoti ekspertam, atstājot nepareizi klasificētus 40%. Tabulas rezultāti liecina, ka, neizmantojot interaktivitāti, nepareizi klasificēto

piemēru skaits jebkuram algoritmam ir daudz lielāks nekā gadījumā, ja sistēmai tiek dota iespēja identificēt neskaidri klasificētos piemērus un atsijāt tos no nepareizo piemēru klāsta. Jāpiemin, ka eksperimentu rezultātu interpretācijā tiek izmantots pieņēmums, ka eksperta sniegtās klasifikācijas ir pareizas.

Kopumā var spriest, ka datu kopa nesniedz pētāmā koncepta pilnīgu aprakstu, kas arī tika pieņemts, uzsākot darbu šajā problēmsfērā.

4. variants

Rezultātu atspoguļojums ir dots tādā pašā veidā kā 3. eksperimenta variantā. Gan pilnajai (6.8. tabulā), gan samazinātajai datu kopai (6.9. tabulā) dati 3 reizes ir sadalīti apmācības un testa kopā, tāpat kā iepriekš, atstājot 10 piemērus testēšanai.

6.8. tabula

Interaktīvas pieejas lietošana pilnai priekšmetu datu kopai (kompetences)

	Daļēji pareizi	**Īsti neskaidrs**	**Nepamatoti neskaidrs**	**Nepareizi (ar interaktivitāti)**	**Nepareizi (bez interaktivitātes)**
BR(NB)	0,234	0,633	0,000	0,133	0,766
BR(Bagging)	0,167	0,733	0,000	0,100	0,833
BR(AdaBoost)	0,267	0,433	0,000	0,300	0,733
BR(JRIP)	0,267	0,367	0,000	0,366	0,733

6.9. tabula

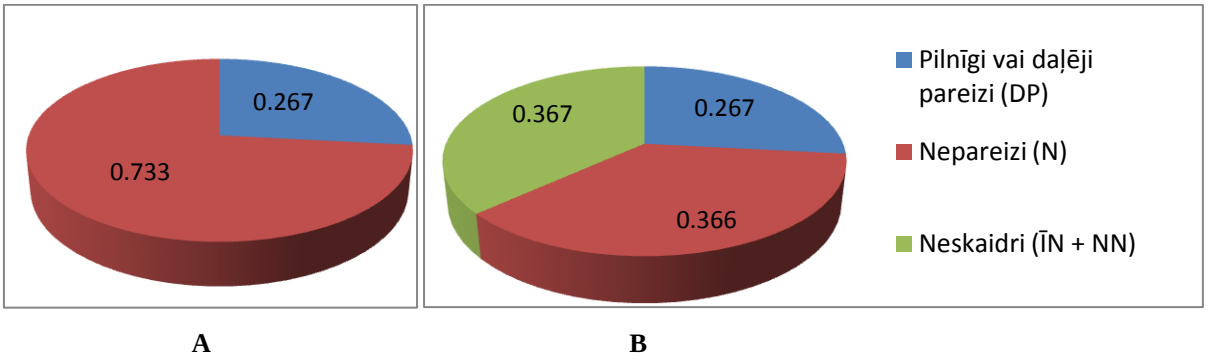
Interaktīvas pieejas lietošana samazinātai priekšmetu datu kopai (kompetences)

	Daļēji pareizi	**Īsti neskaidrs**	**Nepamatoti neskaidrs**	**Nepareizi (ar interaktivitāti)**	**Nepareizi (bez interaktivitātes)**
BR(NB)	0,467	0,467	0,000	0,067	0,467
BR(Bagging)	0,267	0,633	0,000	0,100	0,733
BR(AdaBoost)	0,633	0,167	0,000	0,200	0,367
BR(JRIP)	0,500	0,167	0,000	0,333	0,500

Līdzīgi kā 3. eksperimenta rezultātos, interaktivitāte visiem algoritmiem ļauj samazināt nepareizi klasificēto piemēru skaitu, atsijājot klasifikatoram neskaidros piemērus un nododot tos vērtēšanai ekspertam. Vēl uzskatāmāk tas redzams 6.3. attēlā, kur parādīts *JRip* algoritma rezultāts pilnajai priekšmetu datu kopai (saskaņā ar rezultātiem 6.8. tabulā).

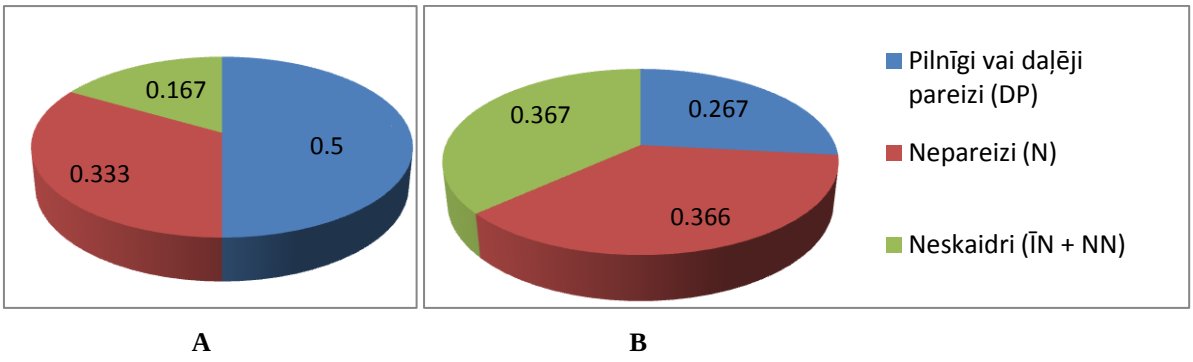
Neizmantojot interaktivitāti (6.3. attēla A daļa), nepareizi klasificēti būtu visi neskaidri klasificētie piemēri, iegūstot tikai 27% daļēji pareizu klasifikācijas rezultātu. Šādai klasifikācijas sistēmai tiešām nav vērts uzticēties. Savukārt, izmantojot interaktīvu pieeju (6.3. attēla B daļa) un nosakot neskaidro klasificētos piemērus, trešdaļu piemēru iespējams atzīt par klasifikatoram neskaidriem un pāradresēt izskatīšanai ekspertam. Tādā veidā nepareizi klasificēti tiek 37%

piemēru, kas, protams, arī nav precizitātes ziņā izcils rezultāts, tomēr ir ievērojami labāks par 73% nepareizi klasificētu piemēru. Kā redzams tabulās, iegūtie rezultāti un sadalījums pa pozīcijām ļoti variējas katram algoritmam.



6.3. att. *JRip* algoritma klasifikācijas rezultātu atspoguļojums priekšmetu salīdzināšanas uzdevumā ar automātisku (A) un interaktīvu (B) klasifikāciju

Salīdzinot pilnās un samazinātās datu kopas rezultātus (6.8. un 6.9. tabulā, 6.3. un 6.4. attēlā) ir redzams, ka jebkurš no izmantotajiem klasifikācijas algoritmiem sniedz labākus klasifikācijas rezultātus, palielinot daļēji pareizo un samazinot gan nepareizo, gan klasifikatoram neskaidro piemēru skaitu, ja tiek lietota samazinātā datu kopu. Abas datu kopas pārklāj vienus un tos pašus apmācības piemērus un saglabā nemainīgu atspoguļojumu (atribūtus), bet samazinātajā datu kopā ir atstātas tikai tās klases, kuras apraksta vismaz četri apmācības piemēri.



6.4. att. *JRip* algoritma interaktīvas klasifikācijas rezultātu atspoguļojums priekšmetu salīdzināšanas uzdevumā ar samazināto (A) un pilno (B) datu kopu

Šāds eksperimentu aspekts imitē situāciju, kad klasifikatora pieredze jomā ir pieaugusi. Tas ļauj secināt, ka interaktīva klasifikācijas sistēma, kas šajā problēmsfērā sākotnēji sniedz daļēji pilnvērtīgus klasifikācijas rezultātus, darbosies labāk un vērsīsies pie eksperta arvien retāk, pieaugot apmācības piemēru skaitam. Tātad ir lietderīgi ieguldīt lielāku eksperta darbu sistēmas izmantošanas sākumposmā, lai klasificētu klasifikatoram neskaidros piemērus un ar savām zināšanām papildinātu klasifikatoru, tādējādi uzlabojot klasifikācijas rezultātus nākotnē.

6.1.4. Klasifikatora iegūtie likumi

6.5. un 6.6. attēlā doti iegūto klasifikācijas likumu piemēri ar algoritmu *JRip* tiešās (vārdu vektori) un netiešās (kompetences) priekšmetu salīdzināšanas gadījumā. Šeit ir dota viena priekšmeta likumu kopa; kopējais klasifikators jebkurā no apraksta formām secīgi definē visus 25 priekšmetus raksturojošos likumus.

```
Model for KnowledgeManagementSystems JRIP rules:
==========
 (student will hav >= 1) => KnowledgeManagementSystems=1 (9.0/4.0)
 (of knowledge manag >= 1) => KnowledgeManagementSystems=1 (3.0/0.0)
 => KnowledgeManagementSystems=0 (57.0/1.0)
 Number of Rules : 3
```

6.5. att. Uz vārdu vektoriem balstīts klasifikators – likumu kopa – priekšmetam „Zināšanu vadības sistēmas"

```
Model for EnterpriseArchitectureAndRequirementsEngineering
JRIP rules:
===========

(B4 = 1) => EnterpriseArchitectureAndRequirementsEngineering=1 (8.0/3.0)
 => EnterpriseArchitectureAndRequirementsEngineering=0 (61.0/7.0)
Number of Rules : 2
```

6.6. att. Uz kompetencēm balstīts klasifikators – likumu kopa – priekšmetam „Uzņēmumarhitektūra un prasību inženierija"

6.5. attēlā redzamajā likumu kopā algoritma izraudzītie raksturīgie atribūti ir vārdu salikums „*student will hav*" (vārda sakne „*hav*" ir iegūta teksta priekšapstrādes procesā) un „*of knowledge manag"* (arī „*manag*" ir vārda sakne). Pirmais likums paredz, ka, ja jauna studiju priekšmeta aprakstā ir sastopams vārdu salikums „*student will hav*", tad ar 69% pārliecību (to apstiprina 9, bet noliedz 4 piemēri apmācības kopā) šis priekšmets atbilst RTU *Biznesa informātikas* priekšmetam „Zināšanu vadības sistēmas". Otrais likums nosaka, ka, ja priekšmeta aprakstā ir sastopams „*of knowledge manag*", tad tas arī ir atbilstošs „Zināšanu vadības sistēmām", savukārt noslēdzošais noklusētais likums paredz, ka priekšmets nav atbilstošs visos citos gadījumos.

6.6. attēlā priekšmeta „Uzņēmumarhitektūra un prasību inženierija" klasifikācijas modelis paredz, ka *e-CF* kompetences B4 *(Solution Deployment*) esamība nezināma studiju priekšmeta sasniedzamajos mācību rezultātos ar 73% pārliecību nosaka tā atbilstību šim priekšmetam. Citos gadījumos atbilstība netiek konstatēta (90% pārliecība).

6.1.5. Eksperimenti piemērotākā pārliecības sliekšņa noteikšanai

Pārliecības sliekšņa ietekme uz klasifikācijas rezultātiem tiks demonstrēta studiju priekšmetu salīdzināšanas datu kopai, kas balstīta uz kompetencēm, izmantojot autores piedāvāto piemērotākā pārliecības sliekšņa noteikšanas metodi. Piemērotākais sliekšņa līmenis (apzīmēts ar **o**) tiks meklēts gan pilnajai, gan samazinātajai studiju priekšmetu datu kopai. Pilnie eksperimentos iegūtie rezultāti atrodami darba 11. pielikumā.

Pilnajā datu kopā klasifikatora vidējā pārliecība par klasēm, kurām:

- piemēri ir piederīgi (VPP): 0.234;
- piemēri nav piederīgi (VPP): 0.016.

Redzams, ka pārliecība par pozitīvajām klasēm ir diezgan zema, līdz ar to pārliecības slieksni var sākt meklēt jau zemai pārliecībai – no 0.1. Zemā pārliecība par klasei piederošajiem piemēriem kopumā norāda, ka klasifikatora vispārināšanas spēja nav augsta, kas bija redzams arī no eksperimentu rezultātiem iepriekš.

Pieņemsim, ka ir uzstādīti vairāki ierobežojumi, kuri jāievēro piemērotākā sliekšņa lieluma izvēlē. *A* gadījumā noteikts kopējais piemēru skaits, ko eksperts ir ar mieru klasificēt, *B* gadījumā noteikts lietderīgais darbs, bet *C* gadījumā ierobežojumu pret eksperta ieguldīto darbu nav, bet jāizvēlas labākais slieksnis nepareizi klasificēto piemēru samazināšanai.

A) $D_{kopējais} \leq 5$ (uz 10 klasificējamiem piemēriem).

B) $D_{nelietderīgais} \leq 0.5$

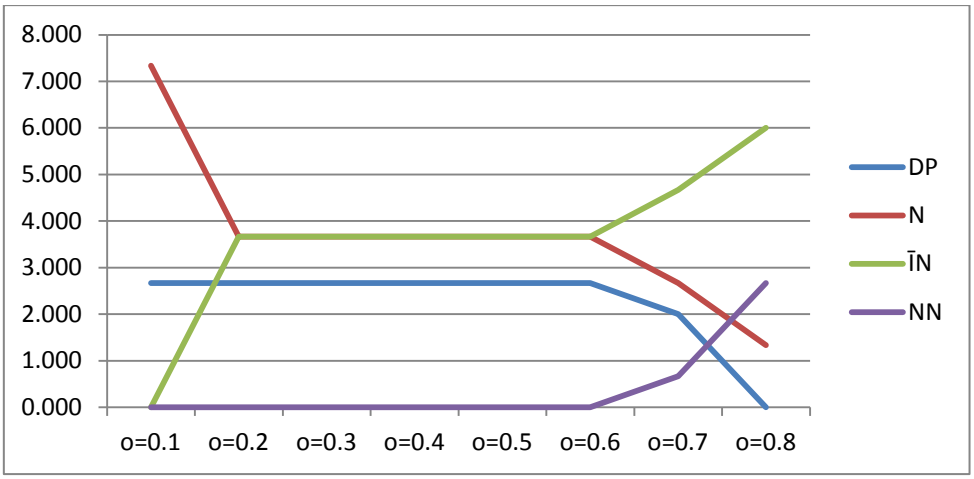
C) Labākais iespējamais (minimāls nepareizi klasificēto (N) piemēru skaits).

6.10. tabula un 6.7. attēls sniedz vidējos rezultātus pēc trīsreizējas datu kopas sadalīšanas pilnai priekšmetu datu kopai. DP, N, ĪN, un NN ir relatīvi pret testa kopas apjomu.

6.10. tabula

Novērtējamie parametri sliekšņa izvēlei pilnā datu kopā

	o=0.1	o=0.2	o=0.3	o=0.4	o=0.5	o=0.6	o=0.7	o=0.8
DP	0,267	0,267	0,267	0,267	0,267	0,267	0,200	0,000
N	0,733	0,366	0,366	0,366	0,366	0,366	0,266	0,133
ĪN	0,000	0,367	0,367	0,367	0,367	0,367	0,467	0,600
NN	0,000	0,000	0,000	0,000	0,000	0,000	0,067	0,267
$D_{nelietderīgais}$	-	0,000	0,000	0,000	0,000	0,000	0,250	0,526
$D_{kopējais}$	0,000	3,667	3,667	3,667	3,667	3,667	5,333	8,667



6.7. att. Grafisks parametru atspoguļojums pilnā datu kopā (X ass – klasifikatora pārliecība, Y ass – piemēru skaits)

No rezultātiem var secināt, ka labākie sliekšņa lielumi, atbilstoši kritērijiem, ir šādi:

A) $D_{kopējais} \leq 5$ (uz 10 klasificējamiem piemēriem): o=0.6. Var paplašināt meklēšanu, izvēršot sīkākus soļus starp sliekšņiem 0.6 un 0.7.

B) $D_{nelietderīgais} \leq 0.5$: o=0.7. Var paplašināt meklēšanu, izvēršot sīkākus soļus starp sliekšņiem 0.7 un 0.8. No grafika redzams, ka visi stāvokļi no o=0.2 līdz o=0.6 ir ekvivalenti.

C) Labākais iespējamais (minimāls nepareizi klasificēto piemēru (N) skaits): o=0.8

Samazinātajā datu kopā vidējā pārliecība par klasēm, kurām:

- piemēri ir piederīgi (VPP) : 0.339,
- piemēri nav piederīgi (VPN): 0.053.

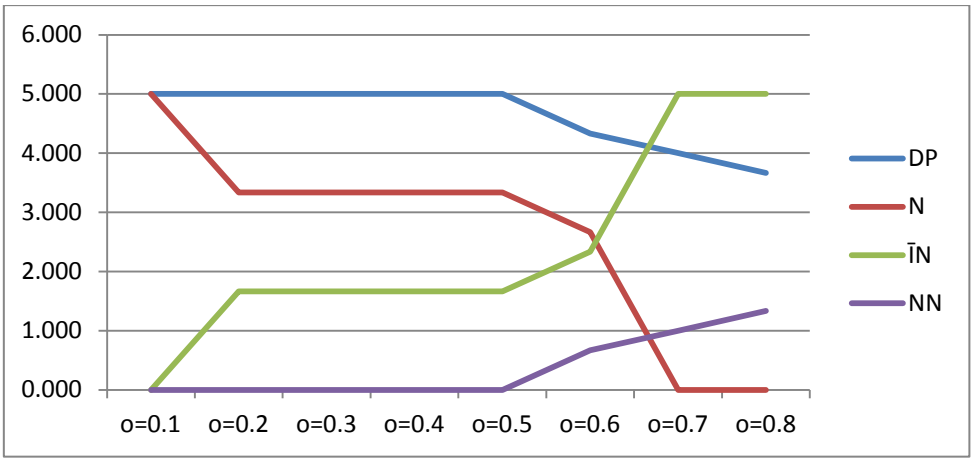
Pārliecība par pozitīvajām klasēm joprojām nav augsta, tomēr tā ir nedaudz izteiktāka kā pilnajā datu kopā.

6.11. tabula un 6.8. attēls sniedz vidējos rezultātus pēc trīsreizējas datu kopas dalīšanas samazinātajai datu kopai.

6.11. tabula

Novērtējamie parametri sliekšņa izvēlei samazinātā datu kopā

	o=0.1	o=0.2	o=0.3	o=0.4	o=0.5	o=0.6	o=0.7	o=0.8
DP	0,500	0,500	0,500	0,500	0,500	0,433	0,400	0,367
N	0,500	0,333	0,333	0,333	0,333	0,267	0,000	0,000
ĪN	0,000	0,167	0,167	0,167	0,167	0,233	0,500	0,500
NN	0,000	0,000	0,000	0,000	0,000	0,067	0,100	0,133
$D_{nelietderīgais}$	-	0,000	0,000	0,000	0,000	0,286	0,200	0,267
$D_{kopējais}$	0,000	1,667	1,667	1,667	1,667	3,000	6,000	6,333



6.8. att. Grafisks parametru atspoguļojums samazinātā datu kopā (X ass – klasifikatora pārliecība, Y ass – piemēru skaits)

No rezultātiem var secināt, ka labākie sliekšņa lielumi atbilstoši kritērijiem, ir šādi:

A) $D_{kopējais} \leq 5$ (uz 10 klasificējamiem piemēriem): o=0.6. Var paplašināt meklēšanu, izvēršot sīkākus soļus starp sliekšņiem 0.6 un 0.7.

B) $D_{nelietderīgais} \leq 0.5$: o=0.7. Lai arī kritērijam atbilst gan slieksnis 0.7, gan 0.8, slieksnim 0.7 ir dodama priekšroka pār 0.8, jo palielinot eksperta darbu (pie 0.8) netiek iegūts uzlabojums nepareizi klasificēto piemēru skaita ziņā.

C) Labākais iespējamais (minimāls nepareizi klasificēto piemēru (N) skaits): o=0.7 (balstoties uz B punktā minēto pamatojumu).

6.1.6. Secinājumi par eksperimentu rezultātiem studiju priekšmetu salīdzināšanā

Eksperimentāli tika pārbaudīta studiju priekšmetu automatizētas salīdzināšanas iespēja, izmantojot gan pilnus studiju priekšmetu aprakstus, neievērojot nekādu struktūru tajos, gan no aprakstiem izgūtas un atbilstoši *e-CF* atspoguļotas kompetences, ko studiju priekšmets sniedz. Darba 1. nodaļā veiktā līdzšinējo risinājumu analīze ļāva izdarīt pieņēmumu, ka strukturētu datu izgūšana no daļēji strukturētiem priekšmetu aprakstiem un to izmantošana priekšmetu salīdzināšanai varētu sniegt labākus rezultātus kā priekšmetu aprakstu tieša lietošana. Eksperimentos iegūtie rezultāti dažādu atbilstošas nozares priekšmetu salīdzināšanā pret RTU studiju programmu *Biznesa informātika* šo pieņēmumu neapstiprina. Izmantojot 5 daudzkategoriju klasifikācijas metrikas (Haminga zaudējums, Mikro-vidējā precizitāte, Mikro-vidējais atsaukums, Viena kļūda, Pārklāšana) un 20 dažādas daudzkategoriju klasifikācijas metodes, nedaudz labākus rezultātus uzrāda apmācības datu kopa, kur par atribūtiem izmantoti vārdi vai vārdu salikumi no priekšmetu aprakstiem, nevis kompetences, kredītpunktu skaits un studiju līmenis. Tomēr metodes, kuru klasifikatori ir lietotājam labi saprotamu likumu formā,

piemēram, *JRip*, uzrāda līdzvērtīgus rezultātus abiem datu atspoguļojuma veidiem. Izmantojot interaktīvo pieeju, abos gadījumos samērā vājie automātiskās klasifikācijas rezultāti lielā nepareizi klasificēto piemēru skaita ziņā ir ievērojami uzlaboti.

Līdz ar to var secināt, ka nav apstiprinājies pieņēmums par sasniedzamo mācību rezultātu izmantošanas lietderības pārākumu salīdzinājumā ar pilnu aprakstu lietošanu. Tas ir, kompetences ir izmantojamas kā priekšmetu raksturojoši atribūti, bet, ja tās nav viegli tiešā veidā iegūstamas, tad neatmaksājas ieguldīt lielo eksperta darbu, kas nepieciešams, lai tās no aprakstiem izgūtu. Ja mācību priekšmetu sasniedzamie rezultāti nākotnē tiks standartizēti, un šie standarti, piemēram, *e-CF*, biežāk tiks izmantoti reālajā dzīvē priekšmetu aprakstīšanai, tad tos var izmantot kā būtiskus priekšmeta parametrus. Tomēr kamēr tā nav, studiju priekšmetu pilni un nestrukturēti vai daļēji strukturēti apraksti ir pietiekami labi izmantojami klasifikatoru iegūšanai, turklāt to iegūšana, apstrāde un lietošana prasa mazāk eksperta darba. Kā negatīvs aspekts pilnu tekstu izmantošanā jāmin iegūto likumu kopas mazā semantiskā jēga problēmsfēras ekspertam. Šajā gadījumā netiek iegūtas sevišķi lietderīgas zināšanas, jo klasifikators vadās pēc dažādu vārdu biežuma priekšmetu aprakstos - lai kurā apraksta daļā tie arī atrastos (piemēram, nepieciešamo priekšzināšanu vai lasāmās literatūras sadaļās). Līdz ar to vairāk lietderīgu zināšanu par sakarībām datu kopā iespējams iegūt, izmantojot jēgpilnus ieejas datus, kas šajā gadījumā ir kompetences.

6.2. Eksperimenti medicīnas jomā

Papildus izglītības sfērai, interaktīvās pieejas pārbaudei tiks izmantota arī medicīnas datu kopa, kas apraksta pacientu stāvokli, piemēroto terapiju un diagnozi (vai vairākas diagnozes) *ICD-9-CM* kodu veidā. Katram ierakstam iespējami vairāki kodi, tādējādi veidojot daudzkategoriju klasifikācijas uzdevumu. Dati ir publiskoti *Computational Medicine Center's 2007 Medical Natural Language Processing Challenge* [30] ietvaros. Atšķirībā no uzdevuma izglītības jomā, šai datu kopai nav raksturīgs tik mazs apmācības piemēru skaits. Tomēr tas nemazina interaktīvās metodes lietošanas *iespējamību* (jo izpildās eksperta pieejamības un problēmas apraksta saprotamības nosacījums), tikai samazina tās *nepieciešamību*, jo arī automātiskas klasifikācijas metodes šeit darbojas pieņemami.

6.2.1. Eksperimentu parametri

Datu kopu raksturojošie parametri atspoguļoti 6.12. tabulā.

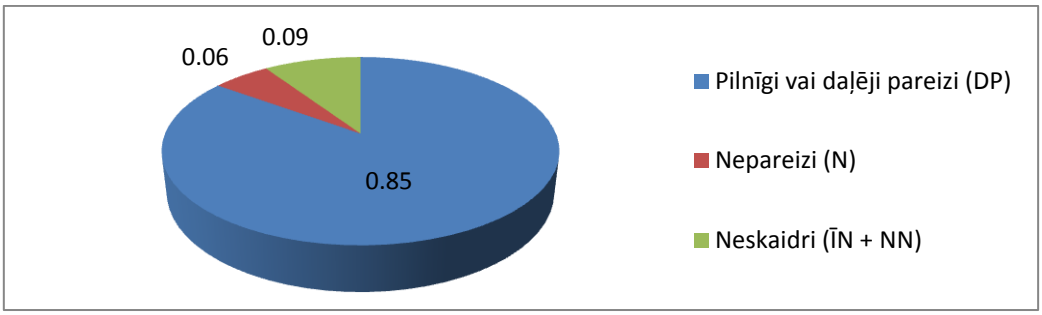
6.12. tabula

Medicīnas datu kopa

	Atribūtu skaits	Piemēru skaits	Klašu skaits	Klašu blīvums	Klašu kardinalitāte	Klašu kopu skaits
Datu kopa	1449	978	45	0.028	1.245	94

6.2.2. Eksperimentu rezultāti

Datu kopas sadalījums apmācības un testa kopā saglabāts atbilstoši oriģinālajiem datiem [30]. Izmantojot binārās saistības metodi ar *JRip* algoritmu, kurš uzrādīja labākus rezultātus par citiem izmantotajiem, medicīnas datu kopai iegūti 6.9. attēlā redzamie rezultāti. Lietots noklusētais pārliecības sliekšņa lielums - 0.5.



6.9. att. JRip algoritma klasifikācijas rezultātu atspoguļojums medicīnas uzdevumā

Kā liecina 6.9. attēls, šo datu gadījumā vismaz daļēji pareizu klasifikāciju skaits ir ievērojami lielāks kā studiju priekšmetu uzdevumā – 85%. Tas tā varētu būt, pateicoties daudz apjomīgākai apmācības kopai. Tomēr arī šeit interaktīva pieeja var palīdzēt uzlabot apmācības rezultātu, atklājot vēl 9% potenciāli nepareizi klasificētu piemēru. Jāpiemin, ka eksperimentu mērķis nav bijis atrast labāko klasifikatoru šai datu kopai (autore nepagalvo, ka citi klasifikācijas algoritmi nevarētu sasniegt labāku rezultātu kā 85% daļēji vai pilnīgi pareizu klasifikācijas spriedumu), bet gan demonstrēt, ka automātiskā ceļā iegūtos klasifikācijas rezultātus ir iespējams uzlabot, ja tiek izmantota interaktīvā pieeja.

6.3. Nodaļas kopsavilkums

Šī nodaļa tika veltīta interaktīvas klasifikācijas sistēmas *InClaS* modeļa un tā komponenšu eksperimentālai pārbaudei.

Interaktīvas klasifikācijas sistēmas novērtēšanā:

1. definēts eksperimentu plāns, atlasītas novērtējuma metrikas;
2. veikti salīdzinoši eksperimenti automātiskai un interaktīvai klasifikācijas pieejai studiju priekšmetu salīdzināšanas un medicīnas diagnostikas uzdevumā.

Eksperimentos tika pārbaudīts, vai

1. nepareizi klasificēto piemēru skaita ziņā interaktīvā pieeja ir labāka par klasisko automātisko klasifikāciju (rezultāti apstiprina T1);
2. metode piemērotākā pārliecības sliekšņa lieluma noteikšanai palīdz atrast labāko stāvokli nepareizi klasificēto piemēru skaita ziņā un eksperta ieguldītā darba apjomā (rezultāti apstiprina T2);
3. induktīvā daudzkategoriju klasifikācijas pieeja ir piemērota studiju priekšmetu salīdzināšanas uzdevumam (rezultāti apstiprina T3) un izvēlētie atribūti ir piemēroti, lai aprakstītu studiju priekšmetu sfēru;
4. *InClaS* modelis var būt lietderīgs arī citās problēmsfērās.

Ir aprakstītas rezultātu novērtēšanai izmantotās metrikas, eksperimentu plāns studiju priekšmetu salīdzināšanas uzdevumā, kā arī medicīnas uzdevumā, kam raksturīga daudzkategoriju klašu piederība.

Studiju priekšmetu salīdzināšanas uzdevumā veikti eksperimenti ar divos veidos atspoguļotām datu kopām un dažādiem parametriem. Izmantota priekšmetu tiešā salīdzināšana, netiešā salīdzināšana, kā arī datu kopa ar samazinātu klašu skaitu, kur katru klasi raksturo vismaz 4 piemēri, imitējot situāciju, ka klasifikators darbības laikā iegūst jaunu apmācības pieredzi no eksperta klasificētiem piemēriem, un analizējot, vai lielāks apmācības kopas apjoms atstāj pozitīvu iespaidu uz klasifikācijas rezultātiem. Iegūtie rezultāti ļauj secināt, ka interaktīva klasifikācijas sistēma darbosies labāk un vērsīsies pie eksperta arvien retāk, pieaugot apmācības piemēru skaitam. Tātad ir lietderīgi ieguldīt lielāku eksperta darbu sistēmas izmantošanas sākumposmā, lai klasificētu klasifikatoram neskaidros piemērus un ar savām zināšanām papildinātu klasifikatoru, tādējādi uzlabojot klasifikācijas rezultātus nākotnē.

Veiktie eksperimenti ar studiju priekšmetu salīdzināšanu pierāda piedāvātā interaktīvā klasifikācijas sistēmas modeļa lietderību un apstiprina eksperimentāli pārbaudāmos aspektus, kas atspoguļo arī darbā aizstāvamās tēzes. *InClaS* prototipa lietojamība, nodrošinot visu nepieciešamo funkciju izpildi, apstiprinājās eksperimentu veikšanas gaitā. Gan universitāšu priekšmetu salīdzināšanā, gan medicīnas diagnostikas uzdevumā nepareizi klasificēto piemēru skaitu iespējams samazināt, ja tiek izmantota piedāvātā interaktīvā pieeja. Studiju priekšmetu salīdzināšanas gadījumā nepareizi klasificēto piemēru skaita starpība (starp automātisku un interaktīvu pieeju) ir tik būtiska, ka interaktīvas pieejas lietošana paver iespēju izmantot mašīnapmācības metodes, kas bez interaktivitātes ieviešanas šajā jomā nesniedza apmierinošus rezultātus.

GALVENIE REZULTĀTI UN SECINĀJUMI

Promocijas darbā ir izstrādāts *InClaS* modelis, kas definē algoritmus, metodes un citas komponentes, kas ļauj izstrādāt interaktīvu klasifikācijas sistēmu nepareizi klasificēto objektu skaita samazināšanai jomās, kur pieejams cilvēks – eksperts. Modeļa pārbaudei ir izplānoti un veikti eksperimenti divās problēmsfēras – izglītībā un medicīnā –, kas pierāda, ka nepareizi klasificēto piemēru skaitu iespējams samazināt, ja klasifikatoram neskaidrie (t.i., neklasificētie un nepārliecinoši klasificētie) piemēri tiek atlasīti un nodoti eksperta izvērtēšanai. Sevišķi nozīmīgi ir ieguvumi, ja klasifikācijas rezultāti, izmantojot 'klasisku' neinteraktīvu klasifikatoru, ir nepieņemami vāji, kā tas ir studiju priekšmetu salīdzināšanas uzdevumā, kur bez interaktīvās pieejas klasifikators sniedz vairāk nepareizu klasifikācijas lēmumu nekā pareizu. Līdz ar to var secināt, ka ir sasniegts darba mērķis - *izstrādāt automatizētas klasifikācijas sistēmas modeli, kas pieļauj interaktivitāti ar ekspertu klasifikatora lietošanas laikā, ja klasifikators sastopas ar objektu, ko tas nespēj klasificēt vai nav pārliecināts par sava lēmuma pareizību* - un ir iespējams izteikt **rekomendācijas par *InClaS* lietošanu**.

Interaktīvas klasifikācijas sistēmas lietošana *ir iespējama* jomās, kur:

- cilvēks – eksperts ir pieejams un var sniegt klasifikāciju atsevišķiem piemēriem;
- problēmsfēras definēšanai tiek izmantoti ekspertam saprotami atribūti, kuru skaits nav pārāk liels, vai objekta aprakstu iespējams iegūt interpretējamā formā.

Interaktīvas klasifikācijas pieeja *ir piemērotāka* par klasisku automātisku klasifikāciju jomās, kur izpildās vismaz viens no apstākļiem:

- ir būtiski iegūt pareizu klasifikāciju pēc iespējas vairāk piemēriem, un tā sasniegšanai ir pieņemami ieguldīt eksperta darbu un laiku;
- ir grūti izgūt vai definēt raksturīgās iezīmes, kā rezultātā atribūti neapraksta pētāmo konceptu pilnīgi;
- ir pieejama tikai neliela sākotnējā apmācības kopa, un pastāv aizdomas, ka tā nav pietiekami reprezentabla.

Būtībā, izstrādātais *InClaS* modelis ir ieviešams tajās sfērās, kur ir pieejams vismaz neliels sākotnējais daudzums datu, ko izmantot klasifikatora apmācībai. Sistēmas pilnvērtīgai izmantošanai ir nepieciešams arī cilvēks - eksperts dotajā problēmsfērā, kurš var pareizi klasificēt tos piemērus, kurus klasifikators nespēj. Tomēr arī tad, ja eksperts sistēmas darba turpmākai uzlabošanai nav pieejams, izstrādātā sistēma sniedz paplašinātu informāciju, kas var

palīdzēt izmantot klasifikācijas sniegtos rezultātus. Tā, piemēram, sistēmas darbības novērtēšanai un izpratnes iegūšanai par klasifikatora darbu, ir iespējams apskatīt iegūtos likumus, uz kuru pamata notiek klasifikācija, kā arī interpretēt jaunu objektu klasifikācijas rezultātus, balstoties uz pārliecības līmeni, ar kādu objekts ir klasificēts.

Darba teorētiskie rezultāti

Darba izstrāde devusi teorētiskos rezultātus, kurus iespējams grupēt sekojoši:

- Izstrādāts interaktīvas klasifikācijas sistēmas *InClaS* modelis, kas apvieno interaktīvas klasifikācijas sistēmas radīšanai nepieciešamās komponentes:
    - Izstrādāta realizējamā interaktivitātes shēma, kas parāda atšķirības attiecībā pret 'klasisku' neinteraktīvu klasifikācijas pieeju.
    - Izstrādāta interaktīvas klasifikācijas sistēmas vispārīga struktūra - klasifikācijas sistēmas funkcionālie moduļi un to sasaistes, kā galvenos moduļus izdalot *Datu apstrādes, Lietotāja saskarnes, Klasifikatora veidošanas, Klasifikatora lietošanas* un *Interaktivitātes moduli*.
    - Izstrādātas divas klasifikatora atjaunošanas (papildināšanas) shēmas pēc eksperta veiktas klasifikācijas – *Uz slieksni balstītā statiskās apmācības pieeja* un *Dinamiskās apmācības pieeja*.
- Izstrādāts *InClaS* modeļa papildinājums, kas apvieno interaktīvas daudzkategoriju klasifikācijas sistēmas radīšanai nepieciešamās komponentes:
    - Izstrādāts algoritms klasifikatoram neskaidru piemēru noteikšanai daudzkategoriju klasifikācijas gadījumā.
    - Izstrādāta metode atbilstošākā pārliecības sliekšņa noteikšanai, pie kura algoritma klasificētos piemērus atzīt par klasifikatoram neskaidriem un nodot eksperta pārziņā.
    - Ieviesti un pamatoti vairāki mēri daudzkategoriju klasifikācijas novērtēšanai – *vidējā klasifikatora pārliecība par klasēm, kurām piemēri ir piederīgi* (VPP) un *nav piederīgi* (VPN), eksperta ieguldītā darba mēri: $D_{nelietderīgais}$ - cik pareizi klasificētu piemēru ekspertam jācaurskata, lai klasificētu vienu nepareizi klasificētu piemēru, $D_{kopējais}$ - cik piemēru ekspertam tiek lūgts klasificēt un jēdzieni *Daļēji pareizi vai pilnīgi pareizi klasificēts piemērs* (DP), *Nepareizi klasificēts piemērs* (N), *Īsti neskaidra klasifikācija* (ĪN), *Nepatiesi neskaidra klasifikācija* (NN).

- Ir adaptēta piecu soļu metode intelektuālu sistēmu projektēšanā [127], kas atvieglo analītisko darbu, ieviešot interaktīvo klasifikācijas sistēmu konkrētā problēmsfērā.
- Veikts interaktīvas daudzkategoriju klasifikācijas sistēmas projektējums sistēmu veidojošo moduļu, to ieeju un izeju apraksta veidā.

- Oriģinālapkopojumi līdzšinējo darbu analīzes rezultātā:
  - Izglītības dokumentu datorizētas salīdzināšanas risinājumu apskats.
  - Induktīvās apmācības algoritmu klasifikācija pēc dažādiem parametriem.
  - Esošo interaktīvo klasifikācijas pieeju sistematizācija un salīdzinājums.
  - Klasifikācijas sistēmu arhitektūru apkopojums un salīdzinājums.

Darba praktiskie rezultāti

Darba izstrāde ļāvusi sasniegt šādus praktiskos rezultātus:

- Izstrādāts interaktīvas klasifikācijas sistēmas prototips daudzkategoriju klasifikācijas uzdevumam, kurš pielāgots studiju priekšmetu salīdzināšanai.
- Radīta utilītprogramma datņu sintaktiskai pārveidošanai no vienkategorijas apraksta formas uz daudzkategoriju (*.arff* formāta datnēm), kas ir praktiski izmantojama arī citos uzdevumos.
- Noteikts priekšmetu atbilstības atspoguļojums starp Rīgas Tehniskās universitātes maģistra studiju programmas *Biznesa informātika* priekšmetiem un vairāku Eiropas universitāšu atbilstošās nozares priekšmetiem.

Turpmākajos pētījumos risināmās problēmas

Promocijas darbā aplūkotajai tēmai ir plašs tālāko pētījumu potenciāls gan no izstrādātā klasifikācijas modeļa pilnveidošanas, gan galvenās risinātās problēmsfēras puses.

- Atbilstības definēšanai studiju priekšmetu salīdzināšanas uzdevumā būtu ieteicams izmantot detalizētāku un precīzāku aprēķinu par šobrīd izmantoto kategorisko iedalījumu *atbilst* vai *neatbilst*. Pastāv vairākas iespējas:
  - izmantot vairākus diskrētus stāvokļus, piemēram, balstītus uz identiskumu, līdzību u.c. definētiem attiecību stāvokļiem;
  - mainīt priekšmetu svaru atbilstības noteikšanas laikā. Atšķirībā no medicīnas diagnostikas uzdevumiem, kur vienam pacientam var būt teju neierobežots slimību skaits, priekšmeta apjomu ierobežo kredītpunktu daudzums. No tā izriet,

ka, ja viens priekšmets ir atbilstošs vairākiem citiem, šo priekšmetu temati var tikt ietverti nepilnā apmērā.

- Turpmākā studiju priekšmetu salīdzināšanas risinājuma pilnveidošanā būtu lietderīgi izskatīt kopīgo apmācību (ang.v. - *co-training*) attiecībā uz studiju priekšmetu tekstuālo aprakstu un strukturēto kompetenču izmantošanu, kā arī transduktīvo vai daļēji pārraudzīto apmācību, kas izmanto gan klasificētus, gan neklasificētus piemērus apmācības uzlabošanai. Papildus apskatāms ir gadījumos sakņotas spriešanas (ang. v. - *case-based reasoning*) izmantošanas potenciāls.
- Ir ieteicams izstrādāt lietotājam pieņemamu risinājumu liela atribūtu skaita gadījumā. Pieeja atribūtu loģiskai grupēšanai un sakārtošanai varētu atvieglot informācijas uztveramību ekspertam, ja klasificējamiem piemēriem ir liels skaits atribūtu.
- Papildinājumu var sniegt sīkāk definēts daļēji pareizas klasifikācijas mērs daudzkategoriju uzdevumu novērtējumā.
- Tālākā darbā ir iespējams paplašināt pētījumus un praktiskos eksperimentus klasifikatora veiktspējas izmaiņu novērtēšanai laika gaitā, kad klasifikators tiek papildināts jauniem ar eksperta klasificētiem piemēriem.
- No praktiskās realizācijas puses ir ieteicams pilnveidot izstrādāto sistēmas prototipu, piemēram, paplašinot lietotāja grafisko saskarņu iespējas.

LITERATŪRA

1. Cios K.J., Kurgan L.A., Hybrid Inductive Machine Learning: An Overview of CLIP Algorithms, in New Learning Paradigms in Soft Computing. 2002, Physica-Verlag GmbH: Heidelberg, Germany. pp. 276-321.
2. Witten I.H., Frank E. Data Mining: Practical Machine Learning Tools and Techniques. 3rd ed. 2011: Morgan Kaufmann. 629 p.
3. Machine Learning Interaction Research in the Computer Science Department at Carnegie Mellon 2012. Available from: http://www.csd.cs.cmu.edu/research/areas /machinelearning/.
4. Mitchell T. Machine Learning. 1997: McGraw Hill. 414 p.
5. Coletta R., Castanier E., Valduriez P., Bellahsene Z., DuyHoa N. WebSmatch: a platform for data and metadata integration, DataRing Project meeting June 20 2011, Montpellier. Available from: http://www-sop.inria.fr/members/Reza.Akbarinia/pmwiki/dataring/ uploads//Meetings/Websmatch.pdf.
6. Buntine W., Stirling D., Interactive Induction, in Machine Intelligence: Towards An Automated Logic of Human Thought, J.E. Hayes, D. Michie, and Ė. Tyugu, (Eds.). 1991, Clarendon Press: New York pp. 121-137.
7. Whitehorn M. How to manage semi-structured data. TechTarget, 2011. Available from: http://searchdatamanagement.techtarget.co.uk/feature/How-to-manage-semi-structured-data.
8. Biletskiy Y., Brown J.A., Ranganathan G. Information extraction from syllabi for academic e-Advising. Expert Systems with Applications, 2009. Vol. 36(3, Part 1), pp. 4508-4516.
9. Birzniece I., Kirikova M. Interactive Inductive Learning Service for Indirect Analysis of Study Subject Compatibility. In BeneLearn 2010. Belgium, Leuven: Katholieke Universiteit Leuven pp. 1-6.
10. Biletska O., Biletskiy Y., Li H., Vovk R. A semantic approach to expert system for e-Assessment of credentials and competencies. Expert Systems with Applications, 2010. Vol. 37(10), pp. 7003-7014.

11. Alves H., Figueira Á. A Educational Library based on Clusters of Semantic Proximity. In IADIS European Conference on Data Mining. 2011. pp. 226-228.

12. Ranganthan G.R., Biletskiy Y., MacIsaac D. Machine Learning for Classifying Learning Objects. IEEE CCECE/CCGEI, 2006, pp. 280-283.

13. Rudzājs P., Kirikova M. Towards Monitoring Correspondence Between Education Demand and Offer. In 21st International Conference on Information Systems Development (ISD2011). 2012: Italy, Prato. pp. 1-12.

14. Rudzājs P., Kirikova M. IT Knowledge Requirements Identification in Organizational Networks: Cooperation between Industrial Organizations and Universities. In Proceedings of the 18th International Conference on Information Systems Development (ISD 2009). 2009: China, Nanchang. pp. 187-199.

15. Anohina-Naumeca A., Graudiņa V., Grundspeņķis J. Curricula Comparison Using Concept Maps and Ontologies. In Proceedings of the 5th International Scientific Conference on Applied Information and Communication Technology. 2012: Latvia, Jelgava. pp. 177-183.

16. Jirapanthong W. Classification Model for Selecting Undergraduate Programs. In Eighth International Symposium on Natural Language Processing. 2009. pp. 89-95.

17. Kirikova M. Project report "Services for Curricula Comparison", 2012.

18. Kennedy D., Hyland Á., Ryan N. Writing and Using Learning Outcomes: a Practical Guide, 2009. Available from: http://sss.dcu.ie/afi/docs/bologna/writing_and_ using_learning_outcomes.pdf

19. Rīgas Tehniskā universitāte. Available from: http://www.rtu.lv/ (accessed 2012).

20. Ziemeļkentuki universitāte. Available from: http://www.nku.edu/ (accessed 2012).

21. Rāvensburgas-Vaingartenas lietišķo zinātņu augstskola. Available from: http://www.hs-weingarten.de/ (accessed 2012).

22. Sebastiani F. Machine Learning in Automated Text Categorization. ACM Computing Surveys, 2002. Vol. 34, pp. 1-47.

23. Jianwu Yang, Chen X. A semi-structured document model for text mining. Journal of Computer Science and Technology, 2002. Vol. 17(5), pp. 603-610.

24. Saleem K., Bellahsene Z., Hunt E. PORSCHE: Performance ORiented SCHEma mediation. Information Systems, 2008. Vol. 33(7-8), pp. 637-657.

25. Goth G. Digging Deeper into Text Mining. Computing Now, 2012. Vol. Jan/Feb pp. 7-9.

26. Daud M.N.R., Corne D.W. Human Readable Rule Induction In Medical Data Mining: A Survey Of Existing Algorithms. In WSEAS European Computing Conference. 2007: Athens, Greece.

27. Rak R., Kurgan L., Reformat M. Multi-label associative classification of medical documents from MEDLINE. In Fourth International Conference on Machine Learning and Applications. 2005. pp. 8.

28. Qu G., Zhang H., Hartrick C.T. Multi-label classification with Bayes' theorem. In 4th International Conference on Biomedical Engineering and Informatics (BMEI). 2011. pp. 2281-2285.

29. Yan Y., Fung G., Dy J.G., Rosales R. Medical Coding Classification by Leveraging Inter-Code Relationships. In The Sixteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10). 2010. pp. 193-201.

30. Computational Medicine Center's 2007 Medical Natural Language Processing Challenge. 2007. Available from: http://computationalmedicine.org/challenge/ previous (accessed 2012).

31. Suominen H. Machine Learning and Clinical Text: Supporting Health Information Flow. 2009, Doctoral dissertation, University of Turku, Finland.

32. Larkey L.S., Croft W.B. Combining classifiers in text categorization. In Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. 1996, ACM: Zurich, Switzerland. pp. 289-297.

33. Cios K.J., Moore W. Uniqueness of Medical Data Mining. Artificial Intelligence in Medicine, 2002. Vol. 26, pp. 1-24.

34. Google Translate. Available from: http://translate.google.lv/ 2007. (accessed 2013).

35. European e-Competence Framework 2012. Available from: http://www.ecompetences.eu/ (accessed 2012).

36. Michalski R.S., Kaufman K., Pietrzykowski J. Natural Induction and Conceptual Clustering: A Review of Applications, USA, 2006.

37. Ašmanis A., Bērziņa E., Buiķe M., u.c., eds. Svešvārdu vārdnīca. ed. D. Guļevska. 1996, Norden: Rīga. 800 p.

38. Grundspeņķis J. Ievads intelektuālajās sistēmās. Lekciju konspekts. 1993, Rīga: Rīgas Tehniskā universitāte. 159 p.

39. Michalski R.S. Pattern Recognition as Knowledge-Guided Computer Induction, Department of Computer Science, University of Illinois, Urbana, 1978.

40. Wu X., Kumar V., Quinlan J.R., Ghosh J., Yang Q., Motoda H., McLachlan G.J., Ng A., Liu B., Yu P.S., Zhou Z.-H., Steinbach M., Hand D.J., and Steinberg D. Top 10 algorithms in data mining. Knowledge and Information Systems, 2007. Vol. 14(1), pp. 1-37.

41. Rexer Analytics, 4th Annual Data Miner Survey – 2010 Survey Summary Report. 2011.

42. Duda R.O., Hart P.E., Stork D.G. Pattern Classification. 2nd ed. 2001: Wiley - Interscience. 654 p.

43. Aksoy M.S. Applications of RULES-3 Induction System. In Proceedings of the Innovative Production Machines and Systems. 2008.

44. Michalski R.S. Pattern Recognition as Rule-Guided Inductive Inference. IEEE Transactions on Pattern Recognition and machine Intelligence, 1980. Vol. July(4), pp. 349-361.

45. Thabtah F.A., Cowling P., Peng Y. Multiple labels associative classification. Knowledge and Information Systems, 2005. Vol. 9(1), pp. 109-129.

46. Dumais S., Platt J., Heckerman D., Sahami M. Inductive Learning Algorithms and Representations for Text Categorization. In 7th International Conference on Information and Knowledge Management. 1998. pp. 148-152.

47. Kusiak A. Data mining, lecture materials. The University of Iowa, Intelligent Systems Laboratory, USA, 2006.

48. Wei C.-P., Hu P.J.-H., Sheng O.R.L., Lee Y.-H. Inductive Learning Approach to Intelligent Patient Image Pre-fetching: Extension and Evaluation of CN2 Algorithm, In Proceedings of the 33rd Hawaii International Conference on System Sciences. 2000, IEEE. pp. 10.

49. Bhavsar V.C., Ghorbani A.A., Goldfarb L. Inductive Learning Inability of Artificial Neural Networks. In Canadian Conference on Electrical and Computer Engineering. 2000. pp. 712-716.

50. Mohamed A.H., Haris M., Jahabar S.B., Selangor A. Implementation and Comparison of Inductive Learning Algorithms on Timetabling. International Journal of Information Technology, 2006. Vol. 12(7), pp. 97-113.

51. Shaw M.J., Gentry J.A. Inductive Learning for Risk Classification. IEEE Expert: Intelligent Systems and Their Applications, 1990. Vol. 5(1), pp. 47-53

52. Michalski R.S., A Theory and Methodology of Inductive Learning, in Machine Learning: An Artifical Intelligence Approach, R.S. Michalski, J. Carbonell, and T. Mitchell, (Eds.). 1983, TIOGA Publishing Co.: California.

53. Akgobek O., Aydin Y.S., Oztemel E., Aksoy M.S. A new algorithm for automatic knowledge acquisition in inductive learning. Knowledge-Based Systems, 2006. Vol. 19, pp. 388-395.

54. Ching J.Y., Wong A.K.C., Chan K.C. Class-Dependent Discretization for Inductive Learning from Continuous and Mixed-Mode Data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995. Vol. 17(7).

55. Kuusik R., Lind G. New Solution for Extracting Inductive Learning Rules and their Post-Analysis. In The First International Conference on Advances in Information Mining and Management (IMMM 2011). 2011, IARIA. pp. 121-126.

56. Haunter D. Near Knowledge: Inductive Learning Systems in Law. Virginia Journal of Law and Tchnologies, 2000. Vol. 9.

57. Theodoris S., Koutrumbas K. Pattern Recognition. 3rd ed. 2006: Elsevier. 837 p.

58. Morris D.T., Kalles D. Decision trees and domain knowledge in pattern recognition In Pattern Recognition in Practice IV Conference. 1994.

59. Russell S.J., Norvig P. Artificial Intelligence. A Modern Approach. 1995, USA: Prentice-Hall.

60. Valeskalne I. Induktīvās spriešanas metodes un to pielietojums. 2007, Bakalaura darbs, RTU.

61. Birzniece I. Induktīvo apmācības metožu pielietojums tēlu pazīšanā. 2009, Maģistra darbs, RTU.

62. Cohen W.W. Fast Effective Rule Induction, in Twelfth International Conference on Machine Learning. 1995. pp. 115-123.

63. J.M.Franczak. Fast Effective Rule Induction Overview, Illinois Institute of Technology, 2000.

64. Sergejevs R. Lietotāja iesaistīšana induktīvās apmācības procesā. 2012, Bakalaura darbs, RTU.

65. Tsoumakas G., Katakis I., Vlahavas I., Mining Multi-label Data, in Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, (Eds.). 2010, Springer.

66. Zhang M.-L., Zhou Z.-H. ML-KNN: A lazy learning approach to multi-label learning. Pattern Recogn., 2007. Vol. 40(7), pp. 2038-2048.

67. Tsoumakas G., Katakis I. Multi-Label Classification: An Overview. International Journal of Data Warehousing and Mining, 2007. Vol. 3, pp. 1-13.

68. Vens C., Struyf J., Schietgat L., Džeroski S., Blockeel H. Decision trees for hierarchical multi-label classification. Machine Learning, 2008. Vol. 73(2), pp. 185-214.

69. Ramakrishnan N. The Pervasivness of Data Mining and Machine Learning. Computer, 2009. Vol. Aug pp. 28-29.

70. Zadeh L.A. Fuzzy sets. Information and Control, 1965. Vol. 8, pp. 338–353.

71. Boutell M.R., J. Luo X.S., Brown C.M. Learning multi-label scene classification. Pattern Recognition, 2004. Vol. 37(9), pp. 1757-1771.

72. Sorower M.S. A Literature Survey on Algorithms for Multi-label Learning, Oregon State University, Corvallis, 2010.

73. Zaffalon M. The Naive Credal Classifier. Journal of Statistical Planning and Inference, 2000. Vol. 105(1), pp. 5-21.

74. Data Mining Server: Rule Induction Methods Rudjer Boskovic Institute, 2001. Available from: http://dms.irb.hr/tutorial/tut_rinduct_meth.php (accessed December 19 2011).

75. Sukovs A., Aleksejeva L., Makejeva K., Borisovs A. Datu ieguve: Pamati. 2007, Rīga: RTU.

76. Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I.H. The WEKA Data Mining Software: An Update. SIGKDD Explorations, 2009. Vol. 11, pp. 10-18.

77. Fan R.-E., Lin C.-J. A Study on Threshold Selection for Multi-label Classification, Technical report, National Taiwan University, 2007, 23 p.

78. Li T., Zhu C.Z.S. Empirical Studies on Multi-label Classification. In Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence ICTAI '06. 2006. pp. 86-92.

79. Gao W., Zhou Z.-H. On the consistency of multi-label learning. In Proceedings of the 24th Annual Conference on Learning Theory (COLT'11). 2011: Budapest, Hungary. pp. 341-358.

80. Cherkassky V., Mulier F. Learning from Data: Conceptrs, Theory, and Methods. 2nd ed. 2007: John Wiley & Sons. 538 p.

81. Pyle D. Data Preparation for Data Mining. 1999, San Francisko: Morgan Kaufman. 540 p.

82. Utgoff P.E. Incremental induction of decision trees. Machine Learning Journal, 1989(4), pp. 161-186.

83. Clark P., Niblett T. The CN2 Induction Algorithm. Machine Learning Journal, 1989(3), pp. 261-283.

84. FuturICT: Participatory Computing for Our Complex World, 2012. Available from: http://www.futurict.eu/ (accessed December 7 2012).

85. Krishna A. Why Big Data? Why Now?, in IBM Research - Almaden Colloquium: September 20, 2011. 2011.

86. Rooney B. Big Data's Big Problem: Little Talent. The Wall Street Journal. April 29, 2012, Dow Jones & Company: New York City.

87. Chang F.M. Characteristics Analysis for Small Data Set Learning and the Comparison of Classification Methods. In 7th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'08). 2008. pp. 122-127.

88. Forman G., Cohen I. Learning from Little: Comparison of Classifiers Given Little Training. In 8th European Conference on Principles and Practice of Knowledge Discovery in Databases. 2004. pp. 161-172.

89. Hämäläinen W., Vinni M. Comparison of machine learning methods for intelligent tutoring systems. In International Conference in Intelligent Tutoring Systems. 2006. pp. 525-534.

90. Jang J.-S.R. ANFIS: Adaptive-Network-based Fuzzy Inference Systems. IEEE Transactions on System, Man, and Cybernetics, 1993. Vol. 23(3), pp. 665-685.

91. Birzniece I. Interactive Inductive Learning System: The Proposal. In Proceedings of the Ninth International Baltic Conference Baltic DB&IS 2010. 2010. Latvia, Riga: University of Latvia Press, pp. 245-260.

92. Kwedlo W., Kretowski M. An Evolutionary Algorithn for Cost-Sensitive Decision Rule Learning. In LNCS, Proceedings of the 12th European Conference on Machine Learning. 2001. Springer-Verlag, pp. 288-299.

93. Torgo L. Rule Combination in Inductive Learning. In LNCS, Proceedings of the European Conference on Machine Learning. 1993. Springer-Verlag, pp. 384-389.

94. Quinlan J. C4.5: Programs for Machine Learning. 1992: Morgan Kaufmann.

95. Baig A.R., Shahzad W., Khan S., Altaf F. ACO Based Discovery of Comprehensible and Accurate Rules from Medical Datasets. International Journal of Innovative Computing, Information and Control, November 2011. Vol. 7(11), pp. 6147-6159.

96. Kamruzzaman S.M. Extracting Symbolic Rules for Medical Diagnosis Problem. In Proc. 8th International Conference on Computer and Information Technology (ICCIT 2005). 2005: Dhaka, Bangladesh. pp. 602-607.

97. Zhang J., Nloedorn E., Rosen L., Venese D. Learning Rules from Highly Unbalanced Data Sets. In Fourth IEEE International Conference on Data Mining. 2004. pp. 571-574.

98. Taha I., Ghosh J. Characterization of the Wisconsin Breast cancer Database Using a Hybrid Symbolic-Connectionist System, Center for Vision and Image Sciences, University of Texas, Austin, 1997.

99. Brian R.G., Compton P. Induction of ripple-down rules applied to modelling large databases. Journal of Intelligent Information Systems, 1995. Vol. 5(4), pp. 211-228.

100. Ferrer-Troyano F., Aguilar-Ruiz J.S., Riquelme J.C. Incremental Rule Learning based on Example Nearness from Numerical Data Streams. In Proceedings of the 2005 ACM Symposium on Applied Computing. 2005. ACM, pp. 568-572.

101. Baxter J. A model of inductive bias learning. Journal of Artificial Intelligence Research, 2000(12), pp. 149-198.

102. Birzniece I. From Inductive Learning towards Interactive Inductive Learning. Scientific Journal of Riga Technical University. Computer Sciences. - Applied Computer Systems 2010. Vol. 41, pp. 106-112.

103. Okabe M., Yamada S., Interactive Web Page Retrieval, in Active Mining: New Directions of Data Mining. 2002, OS Press: Amsterdam pp. 31-40.

104. Tanumara R.C., Xie M., Au C.K. Learning Human-Like Color Categorization through Interaction. International Journal of Computational Intelligence, 2007. Vol. 4, pp. 338-345.

105. Hadjimichael M., Wasilevska A. Interactive Inductive Learning. International Journal of Man-Machine Studies, 1993(2), pp. 147-167.

106. Wong M.L., Laung K.S. Data Mining Using Grammar-Based Genetic Programming and Applications. 2000, USA: Kluwer Academic Publishers. 228 p.

107. Li X., Feng L., Zhou L., Shi Y. Learning in an Ambient Intelligent World: Enabling Technologies and Practices. IEEE Transactions on Knowledge and Data Engineering, 2009. Vol. 21(6), pp. 910-924.

108. Muggleton S. Inductive logic programming. New Generation Computing, 1991. Vol. 4(8), pp. 295-318.

109. Minka T.P., Picard R.W. Interactive Learning with a „Society of Models". In Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition. 1996. IEEE, pp. 447-452.

110. Settles B. Curious Machines: Active Learning with Structured Instances. 2008, Doctoral dissertation, University of Wisconsin-Madison.

111. Baum E.B., Lang K. Query learning can work poorly when a human oracle is used. In Proceedings of the IEEE International Joint Conference on Neural Networks. 1992. IEEE Press, pp. 335-340.

112. Settles B. Active Learning Literature Survey, University of Wisconsin–Madison, 2010.

113. Compton P., Jansen R. Knowledge in Context: a strategy for expert system maintenance. In Second Australian Joint Artificial Intelligence Conference. 1988. pp. 292-306.

114. Ho V., Wobcke W., Compton P. EMMA: An E-mail Management Assistant. In IEEE/WIC International Conference on Intelligent Agent Technology. 2003, IEEE: Los Alamitos. pp. 67-74.

115. Compton P., Peters L., G.Edwards, T.G.Lavers. Experience with Ripple-Down Rules. Knowledge-Based System Journal, 2006. Vol. 5(19), pp. 356-362.

116. Compton P., Richards D. Extending Ripple-Down Rules. In 12th International Conference on Knowledge Engineering and Knowledge Managemetnrs (EKAW'2000). 1999.

117. Suryanto H., Richards D., Compton P. The automatic compression of multiple classification ripple down rule knowledge based systems: preliminary experiments In Third International Conference on Knowledge-Based Intelligent Information Engineering Systems. 1999. pp. 203-206.

118. Birzniece I. Architecture of an Interactive Classification System. In The Fifth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC 2012). 2012, IARIA: Lisbon, Portugal. pp. 91-100.

119. Verdenius F., Someren M.W.v. Applications of inductive learning techniques: a survey in the Netherlands. AI Communications. 1997, IOS Press. pp. 3 - 20.

120. Fritz W. Intelligent Systems and Their Societies. 2010. Available from: http://intelligent-systems.com.ar/intsyst/glossary.htm (accessed September 12 2011).

121. Brown D.C. Intelligent Computer-Aided Design, in Encyclopedia of Computer Science and Technology, J.G.Williams and K.Sochats (Eds.) 1998

122. Gero J.S. Design Prototypes: A Knowledge Representation Schema for Design. AI Magazine 1990. Vol. 11(4), pp. 26-36.

123. Bahrami A. Routine design with information content and fuzzy quality function deployment. Journal of Intelligent Manufacturing, 1994. Vol. 5 (4), pp. 203-210.

124. Rosenman M.A., Gero J.S. Creativity in design using a design prototype approach, in Modeling Creativity and Knowledge-Based Creative Design. 1993, Hillsdale: Erlbaum. pp. 119-148.

125. Dowdy S.M., Wearden S. Statistics for research. 2nd ed. ed. 1991, New York: Wiley 629 p.

126. Bradley C.E., Smyth P. The process of applying machine learning algorithms. In Applying Machine Learning in Practice IMLC-95. 1998. Tahoe city, CA.

127. Bielawski L., Lewand R. Intellignet Systems Design: Integrating Expert Systems, Hypermedia, and Database Technologies. 1991: John Wiley & Sons. 302 p.

128. DTI. Neural Computing. "Department of Trade and Industry" ed. 1994: Learning Solutions p.

129. Han J., Kamber M. Data Mining: Concepts and Techniques. 2nd ed. The Morgan Kaufmann Series in Data Management Systems. 2005: Elsevier. 743 p.

130. Birzniece I. Interactive Inductive Learning System. In Selected papers from the DB&IS 2010. 2010. Latvia, Riga: IOS Press, pp. 380-393.

131. Huysmans J., B.Baesebs, J.Vanthienen. A New Approach for Measuring Rule Set Consistency. Data & Knowledge Engineering, 2007. Vol. 63(1), pp. 167-182.

132. Maloof M.A., Michalski R.S. Incremental Learning with Partial Instance Memory. Artificial Intelligence, 2004. Vol. 154(1-2), pp. 95-126.

133. Utgoff P.E. An Improved Algorithm for Incremental Induction of Decision Trees. In Proceedings of the Eleventh International Conference on Machine Learning. 1994. pp. 318-325.

134. Giraud-Carrier C., Martinez T. An Incremental Learning Model for Commonsense Reasoning. In Proceedings of the Seventh International Symposium on Artificial Intelligence. 1994. pp. 134-141.

135. Bry F., Decker H., Manthey R. A uniform approach to constraint satisfaction and constraint satisfiability in deductive databases. In Proceedings of 1st Extending Data Base Technology. 1988. Venice: Springer-Verlag, pp. 488-505.

136. Fathi M., Alimohammadi A. Inconsistency Detection between Spatial Rules in Land Use Planning Application, September 17 2009. Available from: http://www.geospatialworld.net/index.php?option=com_content&view=article&id=16684:inconsistency-detection-between-spatial-rules-in-urban-planning-application&catid=161:urban-planning-emerging-technologies (accessed December 12 2011).

137. Aksoy M.S. Dynamic System Modelling Using Rules3 Induction Algorithm. Mathematical and Computatuional Applications, 2005. Vol. 10(1), pp. 121-132.

138. Tsoumakas G., Spyromitros-Xioufis E., Vilcek J., Vlahavas I. Mulan: A Java Library for Multi-Label Learning. Journal of Machine Learning Research, 2011. Vol. 12, pp. 2411-2414.

139. Birzniece I. Machine Learning Approach for Study Course Comparison. In International Conference on Machine Learning and Data Mining (MLDM 2012). 2012. Berlin, Germany: IBai, pp. 1-13.

PIELIKUMI

1. pielikums. Termini

Termins, tā sinonīmi	Skaidrojums promocijas darba kontekstā
Apmācības algoritms, klasifikācijas algoritms	Algoritms, kas nodrošina klasifikatora jeb klasifikācijas modeļa iegūšanu no piemēru datu kopas
Automātisks (*automatic*)	Tāds (mehānisms, ierīce), kas darbojas bez tiešas cilvēka līdzdalības. Darbību realizē vadības sistēma pēc noteiktas programmas bez cilvēka līdzdalības [1].
Automatizēts, daļēji automātisks (*automated*)	Process, kas izmanto automātiskus mehānismus, bet saglabā daļēju cilvēka līdzdalību procesa kontrolē vai izpildē
Apmācības dati, datu kopa	Klasifikācijā - problēmsfēru raksturojošs datu paraugs, kur katram piemēram ir zināma klase
Daļēji strukturēti dati	Dati, kuru formātā ir kāda struktūra, kura tiek ievērota – nepastāv strikti formatēšanas likumi, bet ir sastopamas regularitātes, kuras to atšķir no pilnībā nestrukturētiem datiem. Šāda tipa dati ir, piemēram, *XML* formāts vai interneta lapu skripti.
Datizrace, zināšanu izgūšana no datiem (*data mining*)	Process, kurā no datu bāzēm tiek automātiski izgūta noderīga informācija vai zināšanu iegūšana no lieliem datu apjomiem [2]
Deskriptors, selektors, tests, nosacījums	Atribūtu-vērtību konjunkcija, kas veido klasifikācijas likumu
Dinamiskās apmācības algoritmi (*incremental learning, step-by-step*)	Dinamiskās apmācības algoritmi, atšķirībā no statiskajiem apmācības algoritmiem, ļauj caurskatīt un labot klasifikācijas modeli, neveidojot klasifikatoru pilnībā no jauna, pienākot jauniem apmācības piemēriem
Klasifikācija (*classification*)	Klases piederības noteikšana piemēram, balstoties uz iepriekš dotajiem apmācības piemēriem vai klasifikācijas modeli
Klasifikators (*classifier*), klasifikācijas modelis, likumu kopa, likumu bāze	Klasifikators ir modelis (piemēram, likumu kopa), kas kalpo klases (klašu) piederības noteikšanai jaunam piemēram konkrētā problēmsfērā
Klasifikācijas sistēma (*classification system*)	Klasifikācijas sistēma – programmatūra, kas ietver (apvieno) klasifikatoru, lietotāja saskarni un citas saistītās komponentes
Konceptuāli līdzīgas problēmsfēras	Problēmsfēras, kurām ir līdzīgas prasības pret mašīnapmācības risinājumu, vai to atšķirīgās prasības neietekmē risinājuma izvēli
Neklasificēts (*unclassified*) piemērs	Tāds piemērs, kam pēc klasifikatora lietošanas nav izdevies noteikt klases piederību, balstoties uz klasifikācijas modeli
Nepārliecinoši (*low confidence*) klasificēts piemērs	Tāds piemērs, kam klasifikatora iegūtā klases pārliecība ir pārāk zema, lai piešķirtu klasi
Klasifikatoram neskaidrs piemērs, neskaidra klasifikācija (*uncertain classification*)	Jauns klasificējams piemērs, kuram klasifikators nav spējis noteikt klases piederību - *neklasificēts* vai *nepārliecinoši klasificēts* piemērs
Nepilnīgi apmācības dati	Neprecīzi, nepietiekama apjoma dati, kas neapraksta būtisku pētāmās problēmsfēras daļu

Nestrukturēti dati	Datu formāts, kas satur informāciju, kura nav sadalīta diskrētās vienībās, piemēram, teksta dokuments bez sadaļām
Nomināli dati, kategoriski dati	Dati, kam ir galīgs vērtību skaits, tiem iespējams noteikt vienādību vai atšķirību (piemēram, temperatūras kategorijas „silts", „auksts")
Piemērs, objekts, instance, gadījums, novērojums, ieraksts	Atsevišķa datu vienība, kas tiek raksturota ar noteiktām pazīmēm [2]. Piemēram, viena ieraksta rindiņa datu bāzē, viens aplūkojamais objekts ar savām pazīmēm. Darba ietvaros pārsvarā tiek lietots vārds *objekts*, lai atsauktos uz reālās pasaules artefaktu, bet vārds *piemērs* tiek lietots saistībā ar datu kopas, kas satur objektu aprakstus, apstrādi.
Raksturīgā iezīme, atribūts, pazīme	Datu objekta atsevišķs skalārais komponents
Sistēmas arhitektūra	„Sarežģītas sistēmas elementu savstarpējās saistības koncepcija. Projektējot datu apstrādes sistēmu un tai atbilstošo aparatūru, ar šo terminu parasti saprot galveno šīs sistēmas funkcionālo bloku darbības principu, konfigurācijas un savstarpējo savienojumu aprakstu loģiskajā, programmu un fizikālajā līmenī." [3]
Skaitliski dati, nepārtraukti dati	Dati, kam ir bezgalīgs vērtību skaits, tos iespējams salīdzināt un sakārtot (piemēram, temperatūras vērtības 37,001; 37,002; ...)
Strukturēti dati	Dati, kas ir sadalīti diskrētās, identificējamās vienībās, parasti tie ir ērti glabājami datu bāzēs un izgūstami no tām

LITERATŪRA

1. Latviešu literārās valodas vārdnīca. 1972.–1996., Zinātne: Rīga.
2. Sukovs A., Aleksejeva L., Makejeva K., Borisovs A. Datu ieguve: Pamati. 2007, Rīga: RTU.
3. LZA. Akadēmiskā terminu datubāze AkadTerm. 2012. Available from: http://termini.lza.lv/term.php.

2. pielikums. Studiju priekšmetu aprakstu priekšapstrāde teksta klasifikācijai

Teksta priekšapstrādei ir izmantota programmatūra *Weka* [1].

Realizētais priekšapstrādes algoritms teksta klasifikācijai ar *Weka* programmatūru izmantojot vārdu vektorus

1. Datu sagatavošana

Organizēt klasificējamos dokumentus mapēs *.txt* datņu veidā atbilstoši klasēm (mapēm piešķirot vēlamos klašu nosaukumus).

2. Vārdu virknes iegūšana

Atverot programmu *Weka ->Explorer ->Preprocess ->Open file*, logā *Choose* izvēlēties *TextDirectoryLoader* un norādīt kodējumu *UTF-8*. Tiek iegūti divi atritibūti – *text* (dokumenta atspoguļojums *String* virknes veidā) un *@@class@@*.

3. Vārdu vektora iegūšana

Izvēlēties filtru *StringToWordVector* un norādīt vēlamos papildu parametrus, piemēram, vārdu sakņu izgūšanu (*stemmer*), saistītājvārdu izslēgšanu (*stopwords*), vārdu salikumu izmantošanu (*tokenizer*), vārdu skaitu, ko paturēt vektorā (*wordsToKeep*), un citus.

4. Manuāla vārdu vektora rediģēšana

Izņemt kļūdainos vai neatbilstošos atribūtus no vārdu vektora, izmantojot *Remove.*

5. Nesaspiesta datu formāta iegūšana

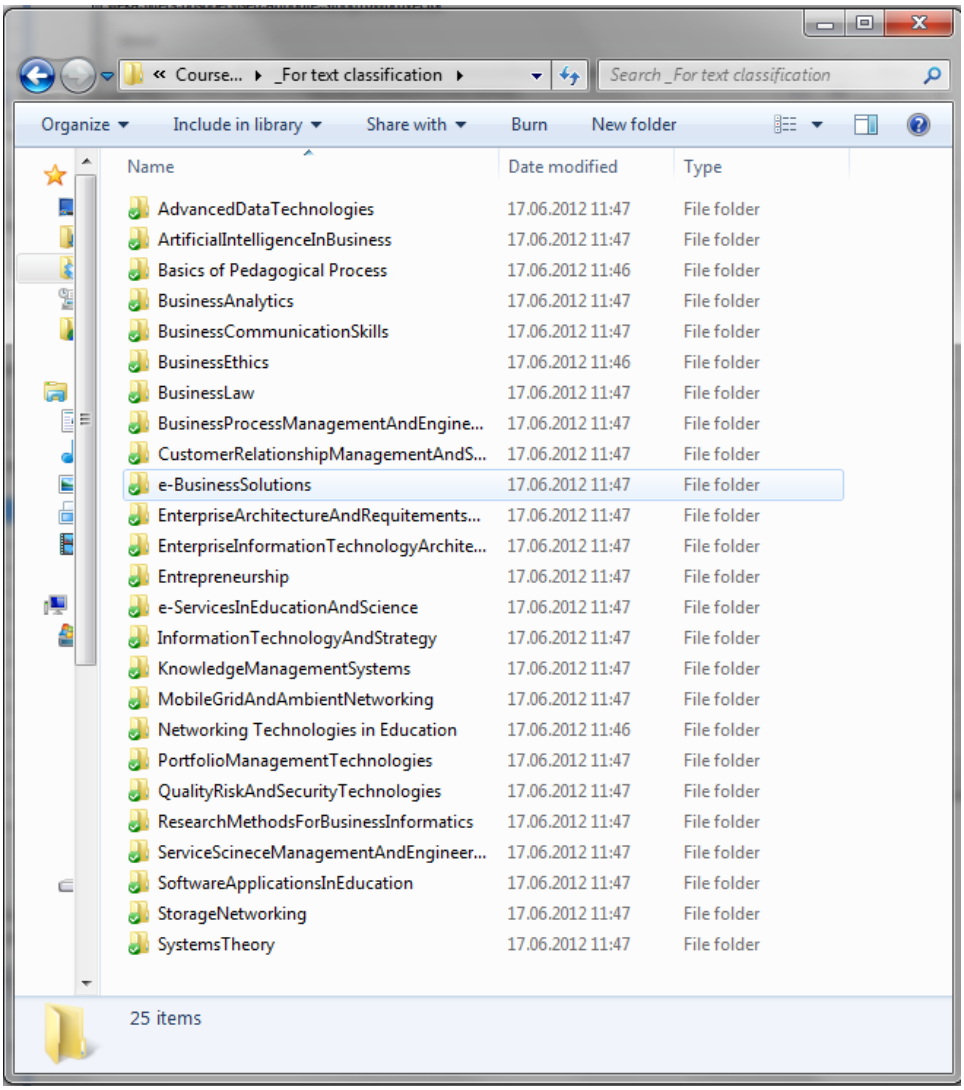
Lietot filtru *SparseToNonSparse*. Saglabāt iegūto apmācības piemēru datni, izmantojot *Save*.

6. Datu formāta pārveidošana

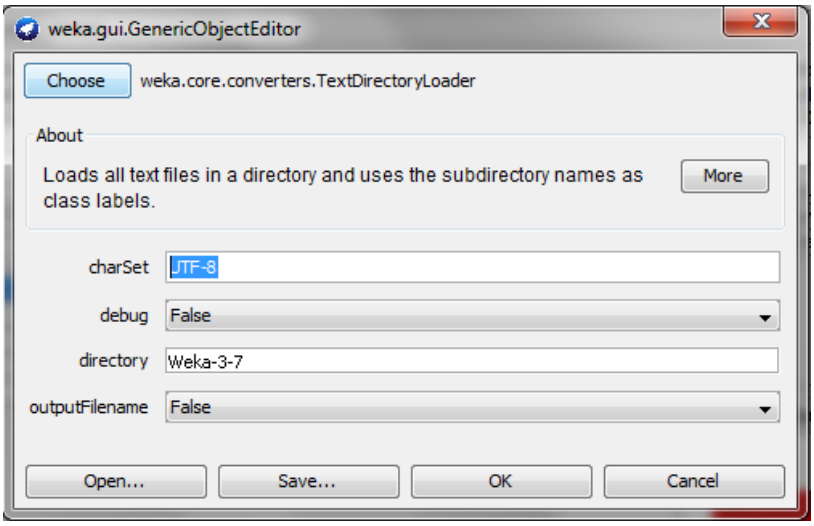
Lai iegūtu daudzkategoriju klasifikācijas aprakstam atbilstošu un ar bibliotēku *Mulan* izmantojamu datu apraksta formātu, izmantot utilītprogrammu, kas aprakstīta 5. pielikumā.

Studiju priekšmetu datu kopas priekšapstrāde vārdu vektoru iegūšanai

1. Vispirms visi dokumenti teksta datņu veidā tiek saglabāti mapēs, kuru nosaukumi atbilst to klasēm. Ja viens dokuments pieder vairākām klasēm, tas tiek ievietos attiecīgi vairākās mapēs.

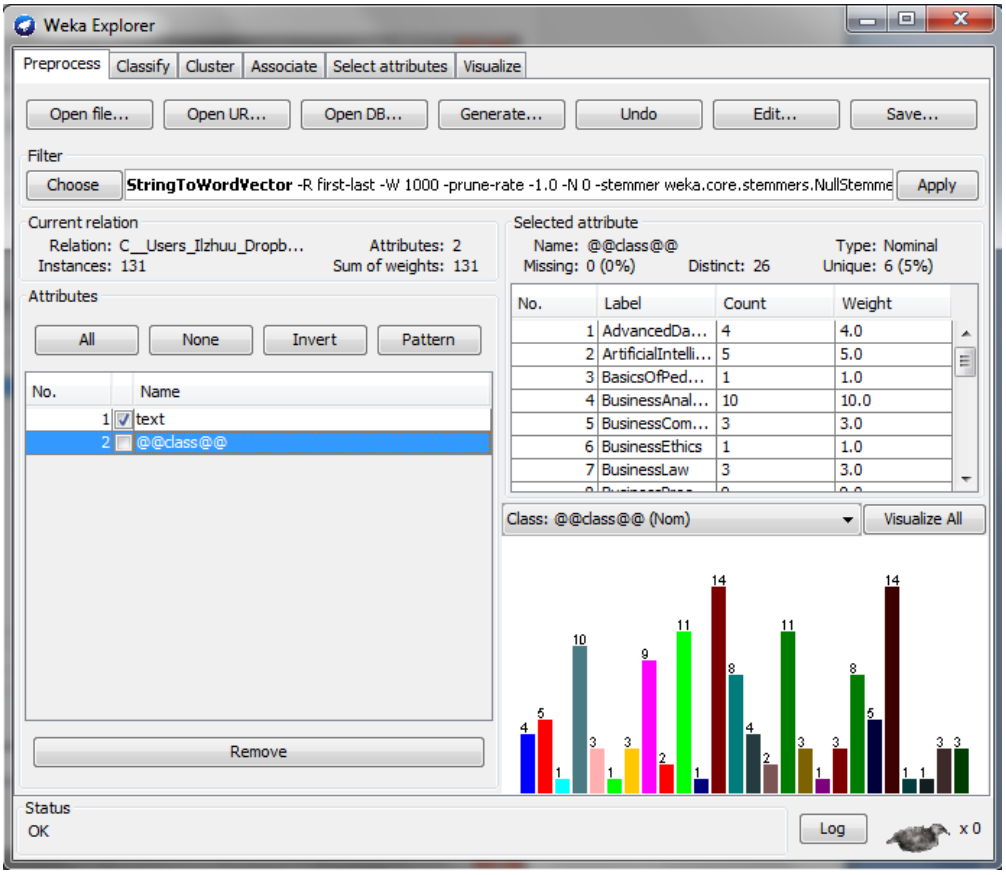


2. Atverot programmu *Weka ->Explorer ->Preprocess ->Open file* un norādot mapi, kura satur visas iepriekš izveidotās klašu mapes ar dokumentiem, sākotnēji tiek izdots paziņojums par neizdevušos datņu ielādi un piedāvāts precizēt ievades datus. Logā *Choose* izvēloties *TextDirectoryLoader* un norādot kodējumu *UTF-8*, tiks nodrošināta pareiza ielāde.

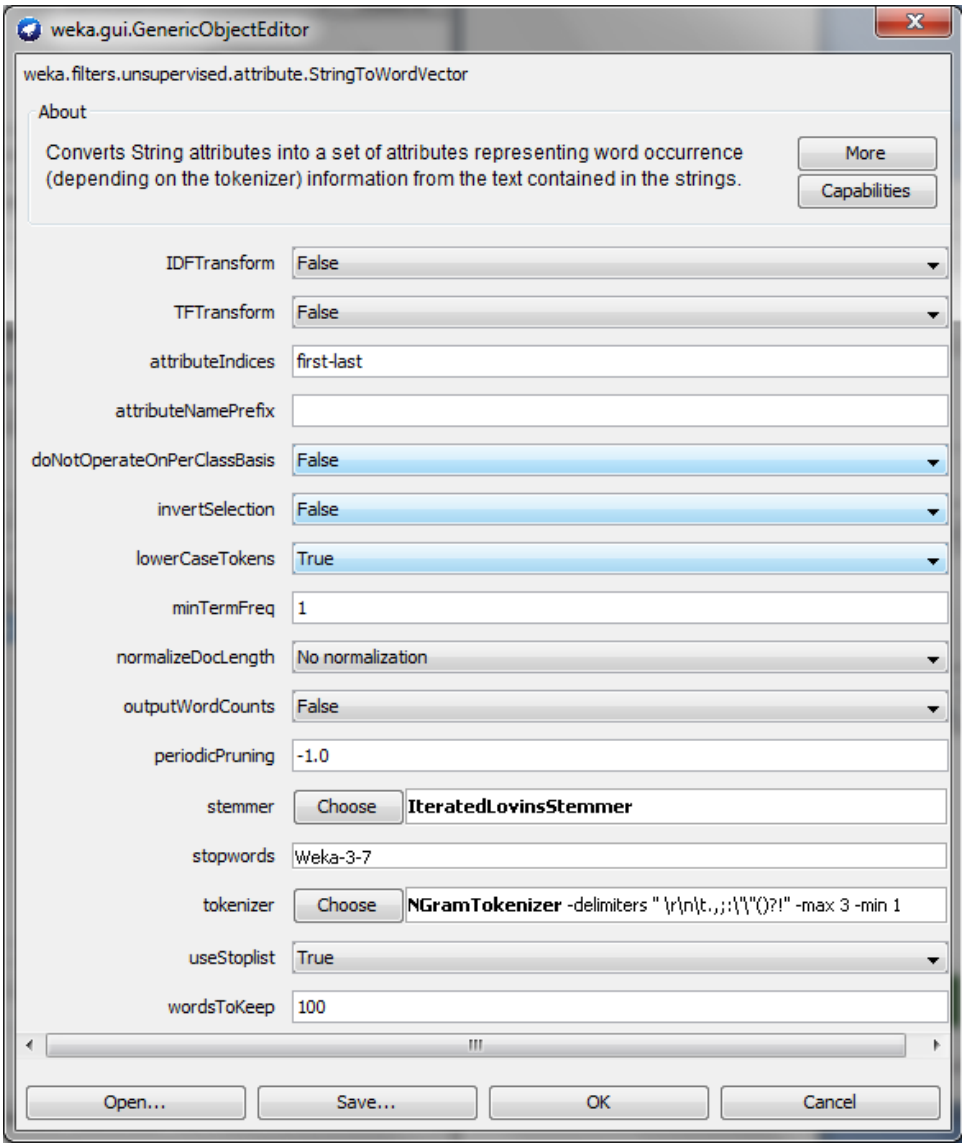


Ielādētie dati sastāv no 2 atribūtiem - katra dokumenta teksts *String* datu veidā un atbilstošā klase.

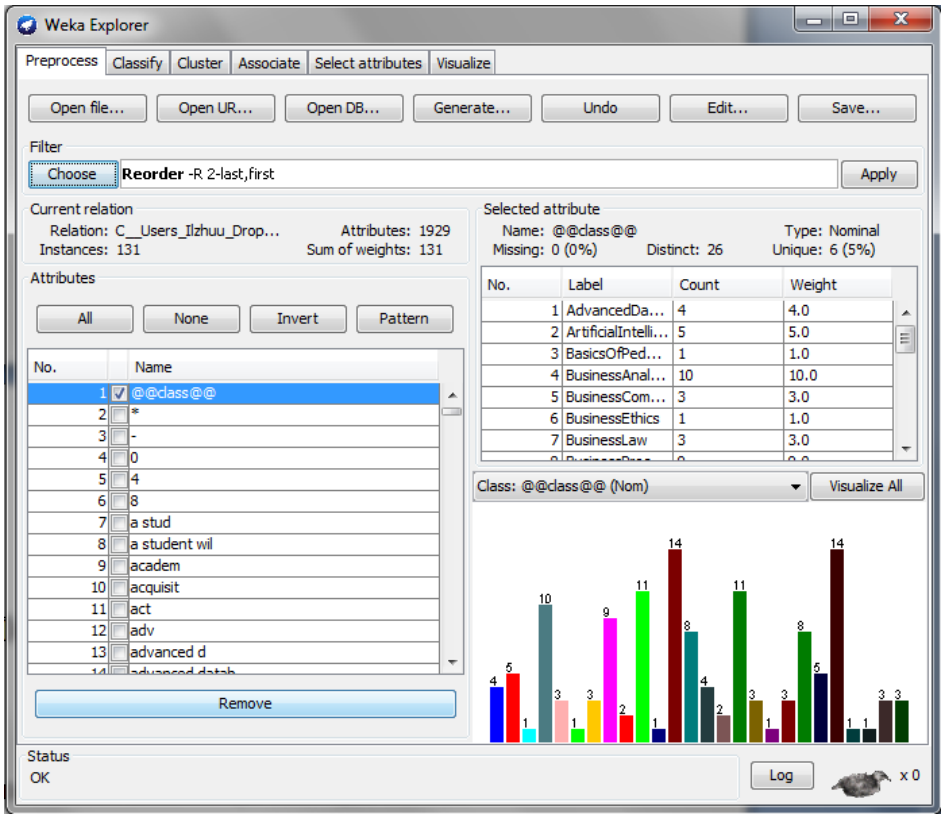
3. Lai iegūtu tālāk apstrādājamus vārdu vektorus, tiek izvēlēts filtrs *StringToWordVector*.



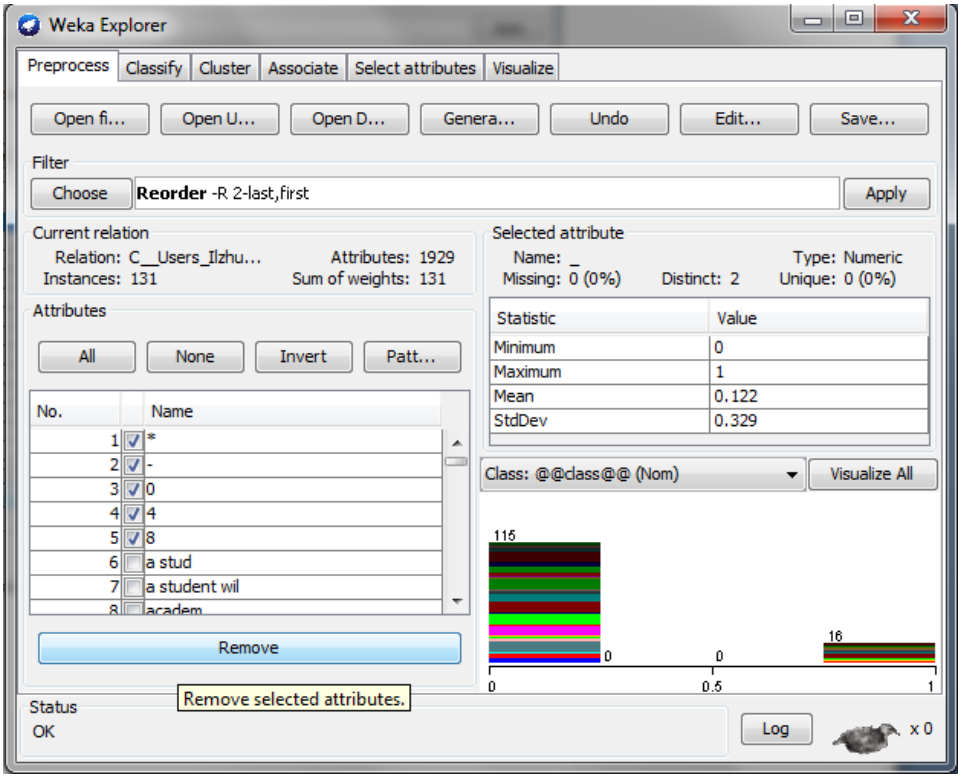
Vārdu vektoru iegūšanai tiek norādīti papildu parametri, būtiskākie no kuriem nosaka vārdu sakņu izgūšanu (*stemmer*), saistītājvārdu (*stopwords*) izslēgšanu un vārdu salikumu izmantošanu (garumā līdz 3 vārdiem), kā arī 100 biežāko, nevis visu, vārdu paturēšanu vārdu vektorā.



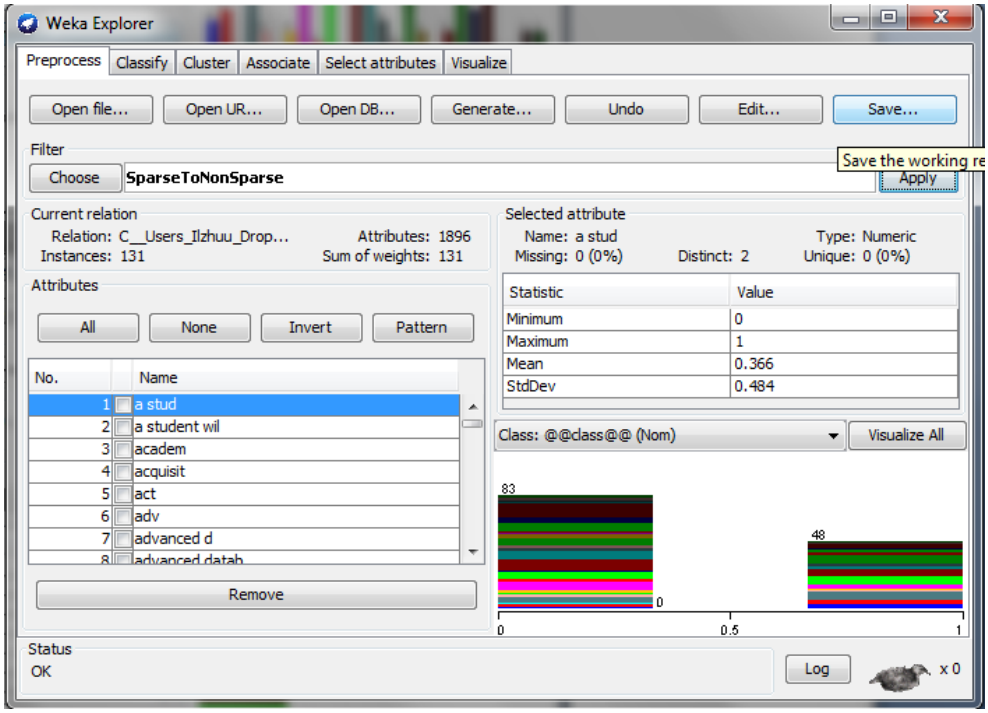
Ar *Reorder* filtra palīdzību, klases atribūts tiek pārvietots uz saraksta beigām.



4. Simboli, cipari un citi kļūdaini sarakstā nonākuši atribūti tiek izņemti manuāli ar *Remove* palīdzību.



5. Filtrs *SparseToNonSparse* nodrošina tālākai izmantošanai izdevīgāka "nesaspiesta" datu apraksta formāta iegūšanu.



6. Iegūtā datne ar konvertora (skat. sīkāk 5. pielikumā) palīdzību tiek iegūta daudzkategoriju klasifikācijas apraksta formātam atbilstošā datnē, un tālākiem klasifikācijas soļiem tiek izmantota bibliotēka *Mulan*.

LITERATŪRA

1. Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I.H. The WEKA Data Mining Software: An Update. SIGKDD Explorations, 2009. Vol. 11, pp. 10 - 18.

3. pielikums. Datu ieguves forma

Course name: Information Economics and Social Simulation

Target courses:

Belongs to	Class name
0	KnowledgeManagementSystems
0	EnterpriseArchitectureAndRequirementsEngineering
0	e-BusinessSolutions
0	ServiceScienceManagementAndEngineering
1	ArtificialIntelligenceInBusiness
0	EnterpriseInformationTechnologyArchitectureApplicationsAndIntegration
0	BusinessProcessManagementAndEngineering
0	AdvancedDataTechnologies
0	BusinessAnalytics
0	ResearchMethodsForBusinessInfomatics
0	Entrepreneurship
0	QualityRiskAndSecurityTechnologies
0	SystemsTheory
0	PortfolioManagementTechnologies
0	BusinessLaw
1	BusinessCommunicationSkills
0	StorageNetworking
0	e-ServicesInEducationAndScience
0	InfomationTechnologyAndStrategy
0	CustomerRelationshipManagementAndSocialNetworkTechnologies
0	MobileGridAndAmbientNetworking
0	SoftwareApplicationsInEducation
0	NetworkingTechnologiesInEducation
0	BasicsOfPedagogicalProcess
0	BusinessEthics

NCP	**6**
Level	**2**

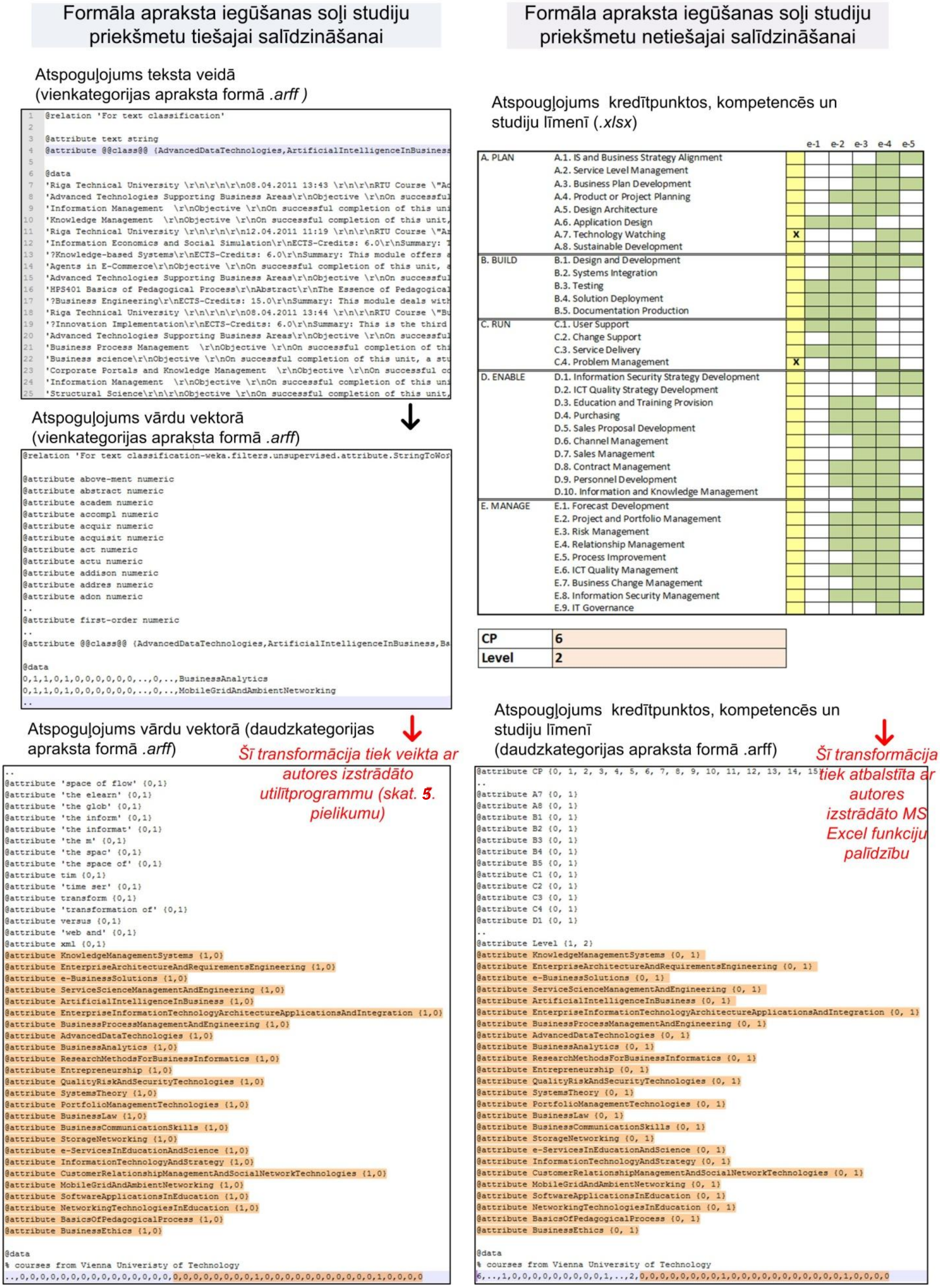
Dimension 1	Dimension 2	Dimension 3	any level	e1	e2	e3	e4	e5
A. PLAN	A.1. IS and Business Strategy Align.							
	A.2. Service Level Management							
	A.3. Business Plan Development							
	A.4. Product or Project Planning							
	A.5. Design Architecture							
	A.6. Application Design							
	A.7. Technology Watching							
e-CF 2.0	A.8. Sustainable Development		x					
B. BUILD	B.1. Design and Development							
	B.2. Systems Integration							
	B.3. Testing							
	B.4. Solution Deployment							
	B.5. Documentation Production							
C. RUN	C.1. User Support							
	C.2. Change Support							
	C.3. Service Delivery							
	C.4. Problem Management							
D. ENABLE	D.1. Inform. Secur. Strategy Devel.							
	D.2. ICT Quality Strategy Develop.							
	D.3. Education and Training Provis.							
	D.4. Purchasing							
	D.5. Sales Proposal Development							
	D.6. Channel Management							
	D.7. Sales Management							
	D.8. Contract Management							
e-CF 2.0	D.9. Personnel Development							
e-CF 2.0	D.10. Inform. and Know. Manag.							
E. MANAGE	E.1. Forecast Development		x			x		
	E.2. Project and Portfolio Manag.							
	E.3. Risk Management							
	E.4. Relationship Management							
	E.5. Process Improvement							
	E.6. ICT Quality Management							
	E.7. Business Change Manag.							
	E.8. Information Security Manag.							
e-CF 2.0	E.9. IT Governance							

Formā atspoguļots piemērs studiju priekšmeta *Informācijas ekonomika un sociālā simulācija* novērtēšanai caur netiešo priekšmetu salīdzināšanu. Balstoties uz šī kursa aprakstu, eksperts nosaka, kādas kompetences tas sniedz un atzīmē *e-CF* [1] ietvarā otrajā (*any level*) vai trešajā dimensijā (konkrēts līmenis). Klasifikācijas sistēma šobrīd strādā ar otrā līmeņa kompetencēm. Izceltās rindiņas kompetenču sarakstā norāda jaunieviesumus, kas iekļauti, sākot no *e-CF* 2.0. Tāpat eksperts arī nolemj, vai un kuram(-iem) no mērķa priekšmetiem (šajā piemērā - 25 RTU *Biznesa informātikas* studiju priekšmeti) šis priekšmets ir līdzīgs (atzīmējot "1"), kā arī norāda kredītpunktu skaitu un studiju līmeni ("1" – bakalaura, "2" – maģistra). Aizpildot priekšmetu novērtējumus šādā sagatavotā *.xls* dokumentā, iespējams automātiski iegūt *.arff* formātā noformētu datu vienību, ko tiešā veidā izmantot klasifikatora ieejā.

LITERATŪRA

1. European e-Competence Framework 2012. Available from: http://www.ecompetences.eu/.

4. pielikums. Studiju priekšmetu formalizētu aprakstu iegūšana



5. pielikums. Utilītprogamma datņu pārveidošanai

Izstrādāta programma, kas nodrošina *.arff* datņu pārveidošanu no vienkategorijas apraksta formāta (kādu iegūst, piemēram, no direktorijās izvietotiem teksta dokumentiem ar programmatūras *Weka* [1] palīdzību) daudzkategoriju formātā (kādu spēj izmantot daudzkategoriju klasifikācijas bibliotēka *Mulan* [2]). Līdz šim šāda veida pārveidojumiem nebija automātiska rīka. Abas datnes pēc būtības apraksta vienu un to pašu konceptu, bet dažādas klasifikācijas programmas izmanto katra savu atšķirīgo formātu. Attēli parāda piemēru, kāds izskatās sākotnējais pārveidojamais datu formāts, pārveidotais datu formāts un *.xml* datne, kas, papildus datiem, nepieciešama daudzkategoriju klasifikācijas bibliotēkai *Mulan*.

```
@relation SL1_test
@attribute a numeric
@attribute b numeric
@attribute class (c, d, e)

@data
1,1,c
1,1,e
1,0,d
0,1,c
```

Vienkategorijas apraksta formāts

```
@relation ML_test
@attribute a numeric
@attribute b numeric
@attribute c {1, 0}
@attribute d {1, 0}
@attribute e {1, 0}

@data
1,1,1,0,1
1,0,0,1,0
0,1,1,0,0
```

Daudzkategoriju apraksta formāts

```
<labels xmlns="http://mulan.sourceforge.net/labels">
<label name="attrib_c"></label>
<label name="attrib_e"></label>
<label name="attrib_d"></label>
</labels>
```

Klašu apraksts XML formā

LITERATŪRA

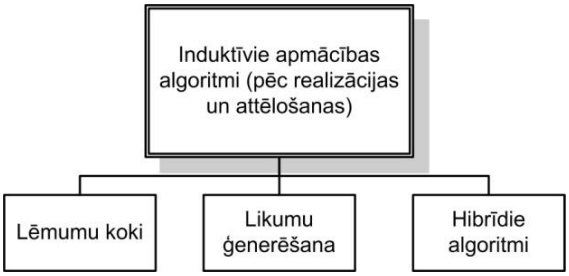
1. Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I.H. The WEKA Data Mining Software: An Update. SIGKDD Explorations, 2009. Vol. 11, pp. 10 - 18.
2. Tsoumakas G., Spyromitros-Xioufis E., Vilcek J., Vlahavas I. Mulan: A Java Library for Multi-Label Learning. Journal of Machine Learning Research, 2011. Vol. 12, pp. 2411-2414.

6. pielikums. Induktīvās apmācības algoritmu iedalījums

Induktīvās apmācības algoritmus var iedalīt pēc dažādām pazīmēm; šajā darbā tiks apskatīti vairāki iedalījumi. Veiktie apkopojumi pirmo reizi ir publicēti autores bakalaura [1] un maģistra darbā [2].

Pēc realizācijas un attēlošanas formas

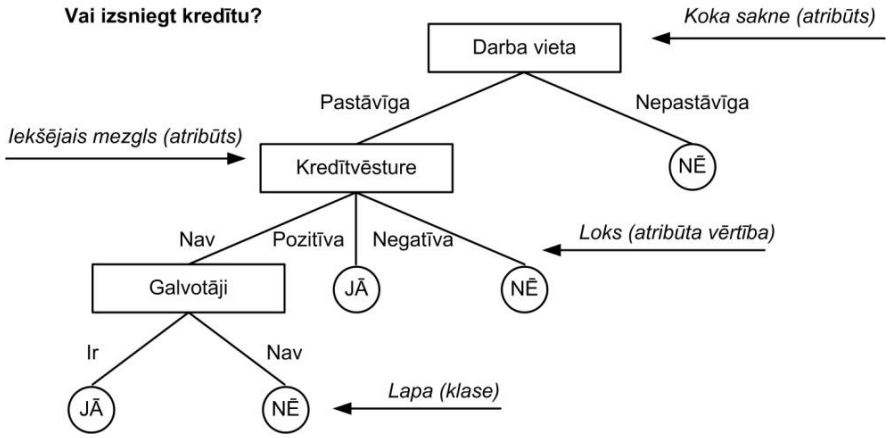
Pēc realizācijas un attēlošanas formas algoritmus iedala lēmumu kokus veidojošos, likumus ģenerējošos un hibrīdos (skat. 1. att.). Hibrīdie algoritmi bieži vien netiek nošķirti no likumus ģenerējošajiem algoritmiem, tāpēc tiks apskatīti kopā ar tiem.



1. att. Algoritmu iedalījums pēc realizācijas un attēlošanas veida

Lēmumu koks

Lēmumu koks ir klasifikators, kas grafiski atspoguļo lēmumu pieņemšanas procesa alternatīvas. Tā struktūra sastāv no koka saknes, lokiem, iekšējiem mezgliem un lapām. Lēmumu koka piemērs, kurā tiek risināta problēma, vai personai izsniegt kredītu, redzams 2. attēlā. Iespējamās atbildes (klases) ir "jā" vai "nē" (attiecīgi – izsniegt kredītu vai neizsniegt).



2. att. Lēmumu koka piemērs

Lēmumu koku konstruēšanas algoritmi ir balstīti uz "skaldi un valdi" pieeju klasifikācijas problēmai [3]. Tie darbojas ar lejupejošo pieeju, katrā posmā meklējot jaunu

dalīšanas atribūtu, kurš vislabāk atdalītu klases. Šis process tiek rekursīvi atkārtots visām apakškopām.

Lēmumu koki ir pierādījuši sevi kā vērtīgus rīkus datu aprakstīšanai, klasificēšanai un vispārināšanai [4]. Lēmumu koki datu apstrādei tiek izmantoti daudzās sfērās. Automātiskās indukcijas lēmumu koku sākotnējais darbs nāk no diviem avotiem [5]. Viens no tiem ir Kvinlana 1986. gadā publicētais *ID3* ar tā pēcteci *C4.5* 1993. gadā. Šī sistēma vēl joprojām mākslīgajā intelektā ir ļoti populāra. Kā rāda statistika, *CART* sistēma, ko 1984. gadā laidis klajā Breimans, arī ir izplatīta. Vēl minami citi populāri lēmumu koku algoritmi – *CN2* [6], *RULES* algoritmu saime [7], *ID5R*, kas ir dinamiskās apmācības algoritms, *OC-SEP* un *OC1*, kas pieder pie slīpo plakņu algoritmiem (tiks aprakstīti turpinājumā).

Lēmumu koku algoritmu priekšrocības[8]:

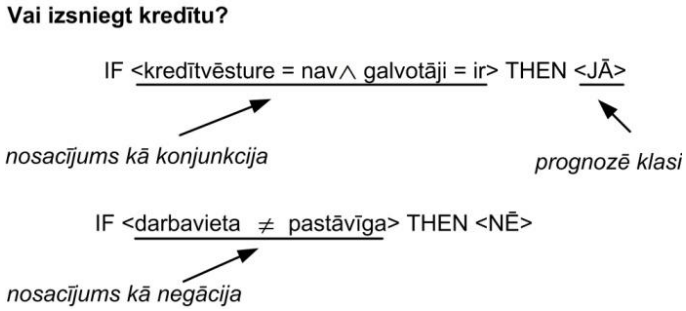
- lēmumu koki atklāj attiecības starp likumiem, kurus var iegūt no šī koka;
- tie atklāj likumus, kas vislabāk raksturo apmācības kopas klases;
- no skaitļošanas viedokļa tie ir vienkārši.

Lēmumu koku trūkumi [8, 9]:

- lēmumu koki var ģenerēt ļoti sarežģītus un garus likumus, kurus ir grūti saīsināt;
- tie ģenerē pārmērīgi daudz likumu, ja netiek lietota kāda koka atzarošanas (ang. v. - *pruning*) tehnika, lai padarītu likumus vispārīgākus;
- liela lēmumu koka veidošana prasa daudz skaitļošanas resursu;
- nepieciešams pieņemt lēmumu, kuru atribūtu lietot par mezglu punktu kokā;
- dinamiskās apmācības ieviešana nav vienkārša.

Likumu ģenerēšana

Likumu ģenerēšana jeb likumu indukcija ir alternatīva pieeja lēmumu kokiem [3]. Likumus var iegūt arī no lēmumu kokiem, bet efektīvāka likumu ģenerēšanas pieeja ir izmanot algoritmus, kas uzreiz rada likumus. Atkarībā no algoritma, likumu apraksta ar atribūtu pozitīvām vērtībām vai to negācijām. 3. attēlā parādīti likumu piemēri kredīta izsniegšanas lēmumam.



3. att. Induktīvo likumu piemēri

- Likumu aprakstā ir lietojams loģiskais AND (UN) un matemātiskās loģikas $\wedge$.
- Disjunkciju var apzīmēt ar OR (VAI), kā arī $\vee$.
- Negāciju jeb noliegumu likumu kontekstā raksturo gan $\neg$, gan $\neq$, gan NE.

Atšķirībā no lēmumu kokiem, kas strādā ar piemēru kopas sadalīšanu, likumu ģenerēšana tiek saukta par pārklājošo pieeju. Tas nozīmē, ka atsevišķi tiek ņemta katra klase, mēģinot pārklāt visus šai klasei piederošos piemērus, tajā pašā laikā izslēdzot klasei nepiederošos. Ja lēmumu koka algoritmi izvēlas atribūtu, kas palielinātu klašu atšķirtību, tad pārklājošie algoritmi lieto atribūta-vērtības pāri, lai maksimizētu vēlamās klasifikācijas iespējamību. Katrs likums pārklāj kādu kopas apakškopu [10]. Vienlaicīgi tiek apskatīti vienas klases piemēri, kurus tajā brīdī sauc par pozitīvajiem piemēriem (pretēji negatīvajiem, kuri ir visām citām klasēm piederošie piemēri). Likumus attēlo disjunktīvajā normālformā. Pie likumu (un hibrīdajiem) algoritmiem pieskaitāmas šādas izplatītas metodes – *AQ* [11], *CN2* [6] (kurš ir radies, izmantojot *ID3* koku veidojošo algoritmu un *AQ* likumu indukciju), *Ripper* [12], *CLIP* [8] un *RULES* [7] algoritmu saime.

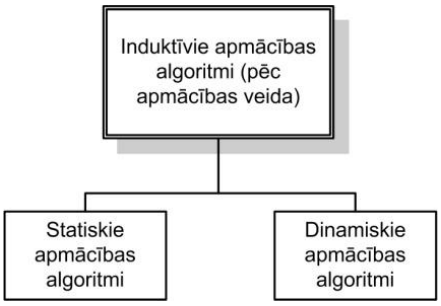
Likumu indukcijai ir šādas priekšrocības:

- viegla likumu uztveramība;
- maz skaitļošanas – netiek prasīti lieli atmiņas resursi.

Kā galvenais trūkums ir minams samērā ilgs apmācības laiks, jo tiek apskatītas visas vai gandrīz visas alternatīvas.

Pēc apmācības veida

Apmācības algoritmus var iedalīt pēc tā, vai tie darbojas statiski vai dinamiski. 4. attēlā parādīts šis iedalījums.



4. att. Algoritmu iedalījums pēc apmācības veida

Statiskie algoritmi ir piemēroti apmācības uzdevumiem, kur ir zināma nemainīga apmācības kopa, dinamiskie – ja apmācības kopa ar laiku papildinās [13]. Populārākie statiskās apmācības algoritmi ir *ID3, C4.5, CART, AQ,* un *CN2* [8, 14].

Dinamiskajā apmācībā (ang. v. - *incremental learning* vai *step-by-step*), pienākot jaunam piemēram, algoritms pārskata un izlabo pašreizējo koncepta definējumu, ja tas ir nepieciešams. Statiskajā gadījumā pēc katra jauna piemēra ienākšanas apmācības kopā būtu jāveido lēmumu koks pilnībā no jauna, un izmaksas laika un resursu ziņā būtu augstas. Tādēļ ir vērts lietot atšķirīgu algoritmu uzdevumiem, kas iekļauj nepieciešamību pēc dinamiskas apmācības. Turklāt šāds algoritms arī apmācības piemērus izvēlas rūpīgāk, tādēļ veidojot mazākus lēmumu kokus [13].

Dinamiskie apmācības algoritmi kā induktīvās apmācības realizācijas veids ir jāizskata arī gadījumos, kad problēmsfērā jāsastopas ar slēpto kontekstu (ang.v. - *hidden context*) un koncepta mainību (ang.v. - *concept drift*) [15]. Slēptais konteksts nozīmē, ka mērķa koncepts var būt atkarīgs no atribūtiem, kas apmācības laikā nav doti. Ar koncepta mainību jāsaskaras tad, ja izmaiņas slēptajā kontekstā izraisa izmaiņas mērķa konceptā. Šīs problēmas detalizētāk ir apskatītas [15, 16]. Dinamiskās apmācības algoritmiem piemīt spēja adaptēties izmaiņām mērķa konceptā [15]. Šādas iespējas ir apgrūtinātas vai neiespējamas statiskās apmācības metodēm. Dinamiskās apmācības metodes, kas realizētas likumu veidā, ir labāk piemērotas koncepta maiņai, jo lēmumu koka veidā atspoguļotu modeļu atjaunošana ir sarežģītāka [15]. Dinamiskās apmācības algoritmus var iedalīt trīs grupās, atkarībā no piemēru atmiņas, ko tie glabā.

1. Pilna piemēru atmiņa. Klasifikators glabā visus apmācības piemērus, kas nodrošina efektīvu klasifikatora atjaunošanu, pienākot jauniem apmācības piemēriem, kā arī parasti sniedz labu prognozēšanas precizitāti, toties prasa glabātuvi visiem apmācības piemēriem, kas ne vienmēr ir iespējams. Pilnas piemēru atmiņas algoritmi ir *ID5, ID5R, ITI,* un citi [13, 16].

2. Bez piemēru atmiņas. Klasifikators neglabā apmācības piemērus, tikai statistiku. Tādējādi tiek ietaupīts uz glabātuves rēķina, bet samazinās jaunu piemēru klasificēšanas precizitāte. Šai grupai pieder tādas apmācības metodes kā *ID4, STAGGER, AQ11* [16].
3. Daļēja piemēru atmiņa. Šajā gadījumā klasifikators glabā tikai izvēlētos apmācības piemērus, kas nodrošina kompromisu starp glabāšanai nepieciešamo atmiņas apjomu un klasifikatora precizitāti. Populārākās daļējas piemēru atmiņas metodes ir *HILLARY, FLORA, AQ-PM* [15, 16].

Viens no iemesliem daļējas piemēru atmiņas algoritmu pētījumiem un lietošanai ir atziņa, ka „cilvēki glabā ne tikai vispārinātus konceptus, bet arī atceras specifiskus gadījumus" [16]. Atšķirībā no cilvēka atmiņas, induktīvās apmācības sistēmai šie specifiskie gadījumi ir jāizvēlas. Ir trīs galvenās pieejas glabājamo piemēru izvēlē daļējas piemēru atmiņas gadījumā.

1. Izvēlēties reprezentatīvus piemērus. Šajā gadījumā ir jālieto kāds kritērijs, kas noteiktu piemēra nozīmīgumu.
2. Atcerēties secīgus piemērus par kādu laika periodu. Laika periods jeb laika logs var būt fiksēta vai mainīga garuma. Loga izmērs arī ir jāizvēlas.
3. Saglabāt robežpiemērus, kas atrodas tuvu pašreizējā koncepta malām. Laika gaitā šie piemēri ir jāatjauno.

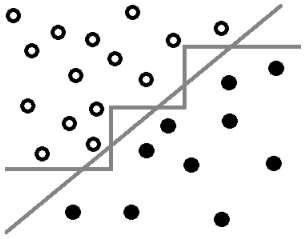
Kā jau minēts pieeju paskaidrojumos, katrai no metodēm ir savi jautājumi, kas jārisina, lai nodrošinātu klasifikatora veiksmīgu darbību. Statiskās apmācības metožu gadījumā šādu problēmu nav, tādēļ ir rūpīgi jāizsver, vai ir nepieciešams lietot dinamisku apmācības pieeju. To, visticamāk, nav vērts darīt, ja apmācības kopa ir konstanta vai mainās lēni. Turklāt, jebkurš statiskās apmācības algoritms var tikt lietots dinamiskā veidā, glabājot visus apmācības piemērus, pievienojot jaunus (kad tādi parādās), un atkārtoti apmācot klasifikatoru [16]. Trūkums statisko metožu lietošanai dinamiskā veidā ir nepieciešamība glabāt visus apmācības piemērus. No otras puses, šādas sistēmas ir mazāk jutīgas uz piemēru sakārtojumu (ang. v. - *ordering effects*) kā dinamiskie algoritmi, jo tiek izmantoti visi pieejamie dati klasifikatora veidošanai. Piemēru sakārtojuma ietekme ir novērojama tad, ja dažādā secībā sakārtoti piemēri apmācības kopā noved pie atšķirīgiem klasifikatoriem [17].

Pēc šķeļošo plakņu novietojuma

Induktīvie algoritmi var tikt iedalīti arī pēc tā, vai tie šķeļ atribūtu kopu ar koordinātu asīm paralēlām plaknēm vai slīpām plaknēm. Klasiskie lēmumu koku algoritmi konstruē kokus, balstoties uz entropiju vai kādu citu informatīvuma mēru. Lēmumu kokus veido, lietojot vienu vai vairākas atribūtus dalošas virsmas vai plaknes (ang. v. - *hyperplane*) [9]. Katrs mezglu

punkts attēlo kādu plakni. Lielākā daļa populāro induktīvo algoritmu ir ierobežoti savā darbībā, jo var sadalīt atribūtu telpu tikai ar koordinātu asīm paralēlām plaknēm. Tas ierobežo iegūto likumu vispārināšanu un rada lielākus kokus.

Atribūtu telpas šķelšanai lietojot slīpas (ang. v. - *oblique*) plaknes, iespējams iegūt labākus rezultātus [18]. Šādi algoritmi, kas realizē slīpo plakņu lēmumu kokus, ir, piemēram, *OC1* [18] un *OC-SEP* [5].



5. att. Datu kopas dalījums ar slīpajām un asīm paralēlajām plaknēm

Bieži vien ir nepieciešams daudz mazāk slīpo plakņu nekā asīm paralēlo plakņu, lai atdalītu klases (skat. 5. attēlā demonstrēto piemēru). Tas nozīmē, ka kokā, kas konstruēts ar slīpām plaknēm, ir mazāk lapu, līdz ar to ir lielāka iespēja, ka tas spēs labāk klasificēt jaunus piemērus [19]. Tomēr šai pieejai ir arī savi trūkumi. Ja datu kopā ir vairāk nekā 2 klases, aprēķini kļūst sarežģīti un klasifikācija nav intuitīvi tik viegli saprotama kā klasiskajos lēmumu kokos. Slīpo plakņu algoritmiem vajadzīgs arī samērā ilgs skaitļošanas laiks.

LITERATŪRA

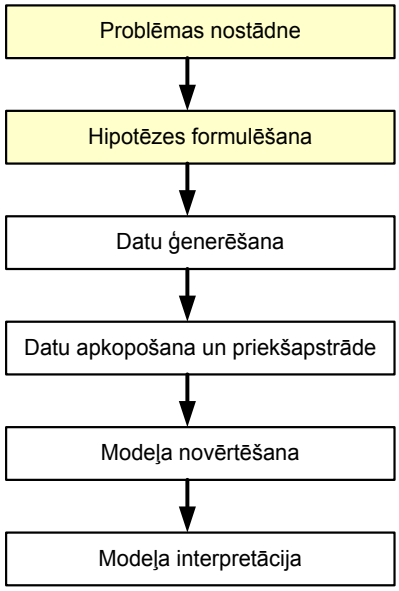
1. Valeskalne I. Induktīvās spriešanas metodes un to pielietojums. 2007, Bakalaura darbs, RTU.
2. Birzniece I. Induktīvo apmācības metožu pielietojums tēlu pazīšanā. 2009, Maģistra darbs, RTU.
3. Data Mining Server: Rule Induction Methods Rudjer Boskovic Institute, 2001. Available from: http://dms.irb.hr/tutorial/tut_rinduct_meth.php (accessed December 19 2011).
4. Murthy S.K. Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. Data Mining and Knowledge Discovery, 1998. Vol. 2, pp. 345-389.
5. Street N.W. Oblique Multicategory Decision Trees Using Nonlinear Programming. Informs Journal on Computing, 2005(1), pp. 25-31.
6. Clark P., Niblett T. The CN2 Induction Algorithm. Machine Learning Journal, 1989(3), pp. 261 - 283.
7. Pham D.T., Aksoy M.S. RULES: A simple rule extraction system. Expert Systems with Applications, 1995. Vol. 8, pp. 59-65.
8. Cios K.J., Kurgan L.A., Hybrid Inductive Machine Learning: An Overview of CLIP Algorithms, in New Learning Paradigms in Soft Computing. 2002, Physica-Verlag GmbH: Heidelberg, Germany. pp. 276-321.
9. Sarigul E. Interactive Machine Learning for Refinement and Analysis of Segmented CT/MRI Images. 2004, Doctoral Thesis, Virginia Polytechnic Institute.

10. Mitchell T. Machine Learning. 1997: McGraw Hill. 414 p.
11. Michalski R.S., Mozetic I., Hong J., Lavrac N. The multipurpose incremental learning system AQ15 and its testing application to three medical domains, in 5th National Conference on Artificial Intelligence. 1986, Morgan-Kaufmann: San Francisco. pp. 1041–1045.
12. Cohen W.W. Fast Effective Rule Induction, in Twelfth International Conference on Machine Learning. 1995. pp. 115-123.
13. Utgoff P.E. Incremental induction of decision trees. Machine Learning Journal, 1989(4), pp. 161-186.
14. Pham D.T., Afify A.A. Applications of machine learning in manufacturing. In Intelligent Production Machines and Systems. 1st I*PROMS Virtual International Conference. 2005. Elsevier, pp. 225 - 230.
15. Ferrer-Troyano F., Aguilar-Ruiz J.S., Riquelme J.C. Incremental Rule Learning based on Example Nearness from Numerical Data Streams. In Proceedings of the 2005 ACM Symposium on Applied Computing. 2005. ACM, pp. 568-572.
16. Maloof M.A., Michalski R.S. Incremental Learning with Partial Instance Memory. Artificial Intelligence, 2004. Vol. 154(1-2), pp. 95-126.
17. Cornuéjols A. Getting Order Independence in Incremental Learning. In Proceedings of the European Conference on Machine Learning, Lecture Notes in Artificial Intelligence. 1993. Springer-Verlag, pp. 196-212.
18. Murthy S.K., Kasif S., Salzberg S. OC1: A randomized algorithm for building ablique decision trees, in Proceedings of the National Conference on Artificial Intelligence (AAAI-93). 1993, The MIT Press. pp. 322-327.
19. Cantú-Paz E., Kamath C. Inducing Oblique Decision Trees with Evolutionary Algorithms. IEEE Transactions on Evolutionary Computing, 2003. Vol. 7(1), pp. 54-68.

7. pielikums. Klasifikācijas sistēmu projektēšana

Šis pielikums apkopo klasifikācijas sistēmu projektēšanas pieejas, kas aprakstītas Čerkaskija un Muliera [1], Dovdija un Verdena [2], Mičela [3], Dudas un Hārta [4], Teodora un Kotrumbas [5], Vardeniusa un Somerena [6], Brodleja un Smita [7], Bielavska un Levanda [8] un *Lielbritānijas Tirdzniecības un industrijas departamenta* [9] darbos. Projektēšanas pieejas ir atspoguļotas vienotā formātā, saglabājot gan sākotnējo autora ideju, gan radot pamatu kopīgu secinājumu izdarīšanai. Ar dzeltenu krāsu attēlos atspoguļoti soļi, kas saistīti ar problēmsfēras izpēti un problēmas formulēšanu, bet ar zaļu - labākā risinājuma meklēšanas soļi – analītiski, eksperimentējot vai radot sistēmas prototipu. Izklāstītos apkopojuma rezultātus darba autore pirmo reizi ir publicējusi [10].

❖ Čerkaskijs [1] apgalvo, ka laba izpratne par visu klasifikācijas procesu ir jebkuras veiksmīgas sistēmas pamatā. Adaptējot pieeju, ko aprakstījuši Dovdijs un Verdens [2], viņš piedāvā vispārēju eksperimentālu procedūru klasifikācijas sistēmu izstrādei (skat. 1. att.).



1. att. Vispārēja eksperimentāla procedūra klasifikācijas sistēmu izstrādei [1]

- **Problēmas nostādne**

Lai iegūtu jēgpilnu problēmas nostādni, parasti ir nepieciešamas zināšanas un pieredze problēmsfērā. Šajā stadijā ir svarīgi nefokusēties uz tālāk izmantojamajām apmācības metodēm, bet gan uz skaidru problēmas definējumu.

- **Hipotēzes formulēšana**

Hipotēze formulē iepriekš nezināmu atkarību, kura jānovērtē eksperimenta datu kopā. Katrai problēmai var tikt definētas arī vairākas hipotēzes. Šajā solī parasti tiek specificētas ieejas un izejas mainīgo kopas.

- **Datu ģenerēšana un eksperimenta plānošana**

Šajā solī veicamās darbības ir atkarīgas no datu ieguves veida. Plānoti eksperimenti, kas pazīstami statistikā, ir iespējami tad, ja datu ģenerēšanas procesu var kontrolēt. Pretējā gadījumā dati tiek iegūti novērojumu ceļā, nekontrolējot datu ģenerēšanas procesu. Ir svarīgi pārliecināties, ka vēsturiski iegūtie dati, kas tiek izmantoti apmācībai, un nākotnes dati, kurus būs nepieciešams klasificēt, nāk no viena un tā paša sadalījuma. Ja tā nav, tad, visticamāk, iegūtais modelis nespēs patstāvīgi klasificēt nākotnes datus.

- **Datu apkopošana un priekšapstrāde**

Novērojumu ceļā iegūtie dati parasti tiek izgūti no datu bāzes. Datu priekšapstrāde parasti ietver vismaz dažus no šiem uzdevumiem – nepiederošo datu izķeršanu, datu kodēšanu, raksturīgo iezīmju izvēli.

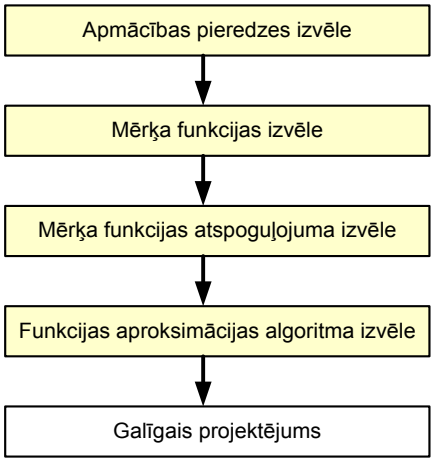
- **Modeļa novērtēšana**

Katra hipotēze attiecas uz kādu nezināmu atkarību starp ieejas un izejas lielumiem. Galvenais mērķis ir konstruēt modeli pēc iespējas precīzākai izejas lielumu prognozēšanai, ja zināmi ieejas lielumi.

- **Modeļa interpretēšana un secinājumu izdarīšana**

Daudzos gadījumos klasifikācijas modeļi ir nepieciešami cilvēkam tālākai lēmumu pieņemšanai. Šādās situācijās modelim ir jābūt saprotamam un interpretējamam, jo ir maz ticams, ka cilvēks balstīs savus lēmumus uz necaurredzamiem ‘melnās kastes’ modeļiem, kas nevar paskaidrot risinājuma iegūšanas ceļu. Jāņem vērā, ka modeļa prognozēšanas precizitāte un interpretējamība ir dažādi un atšķirīgi mērķi, jo cilvēkam saprotamam modelim ir jābūt samērā vienkāršam, savukārt precīzākie modeļi var būt diezgan sarežģīti. Modernās pieejas dod priekšroku metodēm, kas nodrošina augstu prognozēšanas precizitāti, modeļa interpretēšanu apskatot kā atsevišķu uzdevumu.

❖ Dažādas savstarpēji saistītas projektēšanas komponentes ir sniedzis Mičels savā nozīmīgajā grāmatā „Mašīnapmācība” [3]. Projektēšanas soļi parādīti 2. attēlā.



2. att. Sistēmas projektēšanas soļi [3]

- **Apmācības pieredzes izvēle**

Pirmais lēmums, kas jāpieņem, projektējot sistēmu, ir pieredzes izvēle, no kuras sistēma mācīsies. Tas ir atbildīgs uzdevums, jo atbilstoši vai neatbilstoši izvēlēta apmācības pieredze ļoti būtiski ietekmē sistēmas spēju veiksmīgi darboties. Apmācības pieredze var būt tieša vai netieša, ar skolotāja sniegtiem vai sistēmas pašas ģenerētiem mācību piemēriem, kā arī tā precīzāk vai mazāk precīzi atspoguļo reālo piemēru sadalījumu.

- **Mērķa funkcijas izvēle**

Arī šī ir grūti izdarāma izvēle. Ir situācijas, kad mērķa funkcija ir pārāk sarežģīta, lai to pilnībā apmācītos; šajā gadījumā var izvēlēties un lietot šīs funkcijas aproksimāciju.

- **Mērķa funkcijas atspoguļojuma izvēle**

Atspoguļojuma izvēlē ir būtiski sabalansēt izteiksmīgumu un uzskatāmību.

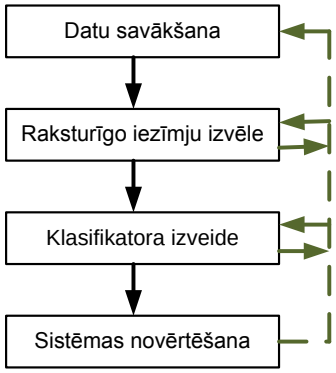
- **Funkcijas aproksimācijas algoritma izvēle**

Lai klasifikators apgūtu mērķa funkciju, ir nepieciešama apmācības datu kopa. Apmācības datu kopa tiek iegūta no iepriekš izvēlētās apmācības pieredzes. Izvēlētais apmācības algoritms pielāgo svarus, lai funkcija vislabāk atbilstu dotajiem apmācības piemēriem.

- **Galīgais projektējums**

Galīgajā projektējuma fāzē tiek izstrādāts sistēmas modelis un apmācības sistēma tiek aprakstīta ar četriem atsevišķiem moduļiem, kas atspoguļo galvenās komponentes daudzās apmācību sistēmās. Mičela piedāvātais sistēmas modelis ir sniegts 8. pielikumā klasifikācijas sistēmu uzbūves kontekstā.

❖ Klasifikācijas sistēmu projektēšanas process ir daudzpusīgi aprakstīts tēlu pazīšanas sistēmu kontekstā. Dažādās tēlu pazīšanas sistēmās galvenie projektēšanas posmi ir vienādi; variācijas galvenokārt ievieš atšķirīgs fokuss un dažādi tēlu veidi, kas dod savu specifiku. 3. attēlā parādīts tēlu pazīšanas sistēmas projektēšanas process, kas iegūts, adaptējot Dudas un Hārta [4] un Teodora un Kotrumbas [5] dotos aprakstus.



3. att. Tēlu pazīšanas sistēmas projektēšanas process

Klasifikatora projektēšanas cikls (izveides posmi) ietver datu savākšanu, raksturīgo iezīmju noteikšanu un iegūšanu, klasifikatora iegūšanu (t.sk. klasifikatora apmācību) un novērtēšanu [4]. Kā ir redzams 3. attēlā, starp posmiem ir atgriezeniskās saites, kas definē iespēju atgriezties agrākos izstrādes posmos un veikt izmaiņas.

- **Datu savākšana**

Šis posms aizņem daudz laika un prasa ievērojamas pūles kopējā sistēmas izstrādes procesā.

- **Raksturīgo iezīmju izvēle**

Šis posms ietver ne tikai iezīmju izvēli un izgūšanu, bet arī iezīmju ģenerēšanu, ja tas ir nepieciešams. Šis posms sasaucas ar „Apmācības pieredzes izvēli" no Mičela modeļa 2. attēlā. Vislielākā nozīme atribūtu izvēlē ir projektētāja zināšanām par problēmsfēru. Vislabākajā gadījumā raksturīgās iezīmes ir viegli iegūstamas, noturīgas pret nebūtiskām transformācijām, noturīgas pret troksni un spēj labi nošķirt klases.

- **Klasifikatora izveide**

Klasifikatora izveide nozīmē arī klasifikatora apmācību. Tā ietver dažādus projektēšanas lēmumus, tajā skaitā lēmumu par izmantojamo metožu klasi, konkrēta algoritma izvēli, parametru pielāgošanu utt. Lai šos jautājumus atrisinātu, nepieciešams veikt analītisku un/vai eksperimentālu darbu.

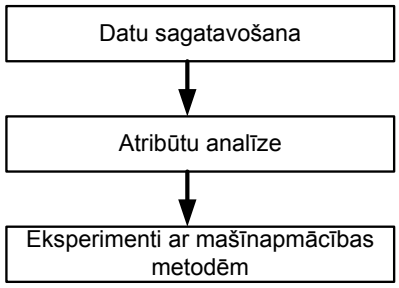
- **Sistēmas novērtēšana**

Iegūtās sistēmas novērtēšana ir nepieciešama gan tādēļ, lai uzzinātu sistēmas veiktspēju (piemēram, klasifikatora prognozēšanas precizitāti), gan tādēļ, lai noskaidrotu nepieciešamos uzlabojumus sistēmas komponentēs. Atgriežoties iepriekšējos projektēšanas posmos, iespējams veikt izmaiņas un atkārtoti novērtēt rezultātus.

❖ Vardeniusa un Somerena pārskatā par induktīvās apmācības tehniku (IAT) lietojumu Nīderlandē [6] pausts viedoklis, ka skatījumam jābūt plašākam par vienkāršu IAT pielietošanu datu kopai. Autori apgalvo, ka IAT ieviešana ir projekts, kura rezultāts ir vai nu (1) sistēma, kas palīdz cilvēkam problēmas risināšanā, vai (2) iegūtās zināšanas, kas ļauj cilvēkam pašam atrisināt viņa problēmu. Projekta pieeja tiek aprakstīta procesa modeļa formā.

Izstrādātajām sistēmām var būt dažādi mērķi, pamatā tie iedalās datu analīzes un klasifikācijas uzdevumos. Datu analīzes gadījumā mērķis ir iegūt datus izskaidrojošu modeli, savukārt klasifikācijas mērķis ir noteikt klases piederību jauniem un iepriekš neredzētiem piemēriem. Klasifikācijas uzdevums ir saukts arī par induktīvo programmēšanu [6], paskaidrojot to kā tādas datorsistēmas izstrādi, kas var automātiski pielietot zināšanas, kas iegūtas ar IAT palīdzību. Vēl citi apmācības sistēmu mērķi ir konceptu pierādīšana, kas fokusējas uz tehniskām aktivitātēm, lai pierādītu IAT pielietošanas iespējamību un tehniku salīdzināšana, lai noteiktu piemērotāko tehniku konkrētai lietojuma sfērai.

❖ Viens no procesu modeļiem darbam ar mašīnapmācības tehnikām ir definēts *Weka* ietvaram (skat. 4. att.).



4. att. Programmatūras *Weka* klasifikācijas process [6]

Weka procesu modelī ir izdalītas trīs fāzes [6].

- **Datu sagatavošana**

Datu sagatavošana iekļauj datu izgūšanu no strukturētas datu bāzes, datu transformācijas un citas priekšapstrādes darbības.

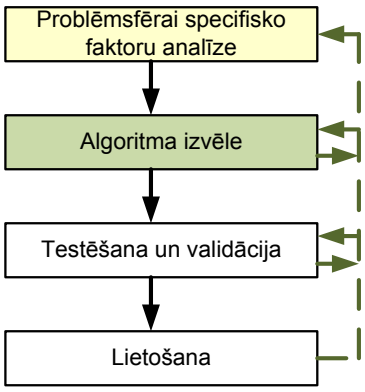
- **Atribūtu analīze**

  Šī posma mērķis ir samazināt atribūtu kopu un iekļaut tikai nepieciešamos atribūtus.

- **Eksperimenti ar mašīnapmācības metodēm**

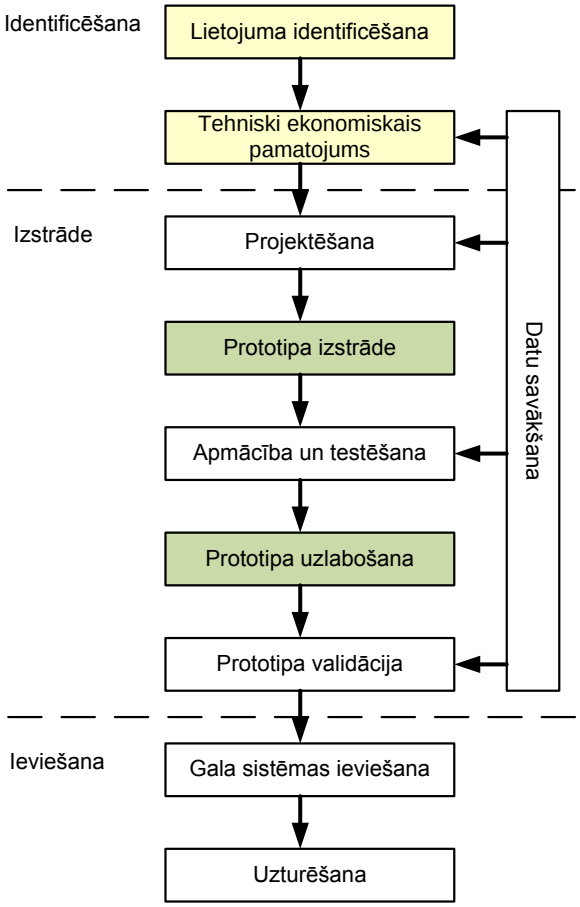
Weka rīkā ir iekļautas dažādas metodes, ar kurām var eksperimentēt un izvēlēties atbilstošāko konkrētajai datu kopai. Tāpat iespējams veikt rezultātu novērtēšanu un salīdzināšanu, kā arī veikt pēcapstrādi. *Weka* fokusējas uz datu analīzes uzdevumu un sniedz iespējas eksperimentēt ar dažādām mašīnapmācības metodēm. Lietotāja prasības un kopējais sistēmas ieviešanas process *Weka* modelī netiek apskatīts, līdz ar to šī pieeja neatbilst Vardeniusa un Somerena ieskatiem par klasifikācijas sistēmas izstrādi kā projektu.

❖ Brodlejs un Smits [7] piedāvā savu modeli konkrētā problēmsfērā lietojamu klasifikācijas sistēmu izstrādei (skat. 5. att.). Šī pieeja domāta pastāvīgi lietojamas klasifikācijas sistēmas iegūšanai, un projektēšanas posmi ir līdzīgi tēlu pazīšanas sistēmas (3. att.) un Čerkaskija piedāvātajai vispārējai klasifikācijas sistēmas (1. att.) soļiem.



5. att. Klasifikācijas sistēmas izstrādes process [7]

❖ Klasifikācijas sistēmas izstrādes modelis, kurš iekļauj arī visus būtiskākos projekta dzīves cikla posmus, ir aprakstīts *Lielbritānijas Tirdzniecības un industrijas departamenta* pārskatā [9]. Līdz ar to viņu aprakstītā pieeja iekļauj tādus etapus, kas nebija minēti iepriekš apskatītajos procesos (skat. 6. att.). Tādi ir, piemēram, tehniski ekonomiskais pamatojums un sistēmas uzturēšana.



6. att. Klasifikācijas sistēmas izstrāde projekta dzīves ciklā [9]

Atkārtotās darbības, tādas kā prototipa izstrāde un uzlabošana, citos modeļos tiek panāktas ar atgriezenisko saišu palīdzību. Formālie procesa aspekti šajā modelī nav sīki aprakstīti, galveno uzsvaru liekot uz izstrādes procesa monitoringu un kontroles iespējām projekta izstrādes gaitā.

❖ Vardeniuss un Somerens [6] secina, ka nepastāv vienots skats uz induktīvās apmācības (plašāk skatoties – klasifikācijas) sistēmu izstrādi. Tomēr autori ir sastādījuši tipisku klasifikācijas sistēmu izstrādes procesa modeli, kurš sastāv no trim līmeņiem un kontroles elementa.

1. Lietojuma līmenis

Šajā līmenī tiek analizēta problēmsfēra, ieskaitot pieejamo resursu (datu, ekspertu) identificēšanu, problēmas dekompozīciju, konceptuālā modeļa konstruēšanu, risinājuma mēroga definēšanu. Mašīnapmācības metodes var lietot, lai risinātu visu problēmu vai tikai kādu daļu no tās. Šī posma izpildes rezultātā tiek radīts plāns tālākai problēmas risināšanai, kā arī prasības pret nepieciešamajiem resursiem un plānoto gala risinājumu.

2. Analīzes līmenis

Šajā posmā notiek datu apkopošana, raksturīgo iezīmju iegūšana, priekšapstrāde utt. Būtisks uzdevums ir arī vienas vai vairāku atbilstošu apmācības metožu izvēle.

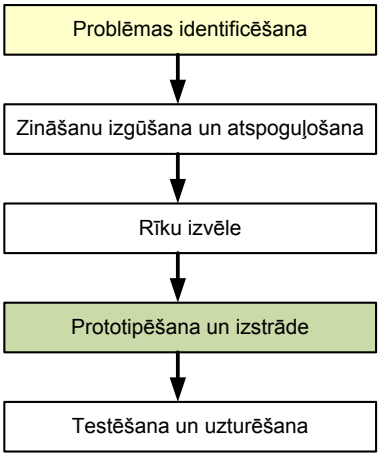
3. Tehniskais līmenis

Apmācības tehnikām parasti ir dažādi parametri, kuru noteikšana prasa arī praktiskus eksperimentus.

Kontroles elements – projekta vadība

Visu līmeņu izpildes un dažādu lēmumu pieņemšanas gaitā ir jāseko līdzi, vai projektējamā sistēma joprojām atbilst lietotāja izvirzītajām prasībām.

❖ Klasifikācijas sistēmas ietilpst arī intelektuālu sistēmu kategorijā. Bielavskis un Levands grāmatā "Intelektuālu sistēmu projektēšana" [8] piedāvā piecu soļu procedūru intelektuālu sistēmu projektēšanai (skat. 7. att.).



7. att. 5 soļu procedūra intelektuālu sistēmu projektēšanai [8]

Par būtiskākajiem soļiem ir atzīti pirmie divi. Tas arī saskan arī ar iepriekš apskatītajiem citu autoru modeļiem, jo saprātīga problēmas nostādne un ieejas dati veido pamatu visai tālākajai sistēmas izstrādei. Ja nopietnas kļūdas būs šajos posmos, viss tālākais darbs nesniegs vajadzīgos rezultātus.

LITERATŪRA

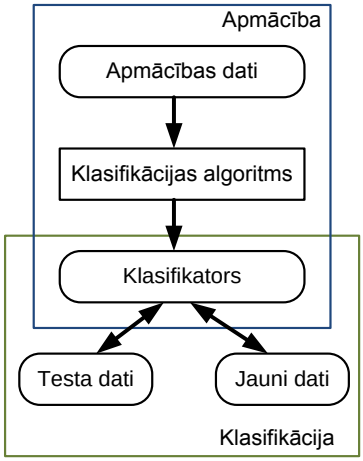
1. Cherkassky V., Mulier F. Learning from Data: Conceptrs, Theory, and Methods. 2nd ed. 2007: John Wiley & Sons. 538 p.
2. Dowdy S.M., Wearden S. Statistics for research. 2nd ed. ed. 1991, New York: Wiley 629 p.
3. Mitchell T. Machine Learning. 1997: McGraw Hill. 414 p.

4. Duda R.O., Hart P.E., Stork D.G. Pattern Classification. 2nd ed. 2001: Wiley - Interscience. 654 p.
5. Theodoris S., Koutrumbas K. Pattern Recognition. 3rd ed. 2006: Elsevier. 837 p.
6. Verdenius F., Someren M.W.v. Applications of inductive learning techniques: a survey in the Netherlands, in AI Communications. 1997, IOS Press. pp. 3 - 20.
7. Bradley C.E., Smyth P. The process of applying machine learning algorithms. In Applying Machine Learning in Practice IMLC-95. 1998. Tahoe city, CA.
8. Bielawski L., Lewand R. Intellignet Systems Design: Integrating Expert Systems, Hypermedia, and Database Technologies. 1991: John Wiley & Sons. 302 p.
9. DTI. Neural Computing. "Department of Trade and Industry", Learning Solutions report. 1994.
10. Birzniece I. Architecture of an Interactive Classification System in The Fifth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC 2012). 2012, IARIA: Lisbon, Portugal. pp. 91-100.

8. pielikums. Klasifikācijas sistēmu uzbūve

Ja projektēšana apraksta, kā izveidot klasifikācijas sistēmu un kādi lēmumi jāpieņem šī procesa gaitā, tad klasifikācijas sistēmas uzbūve raksturo, kā sistēma funkcionē un no kādām komponentēm sastāv. Klasifikācijas sistēmu uzbūves apkopojumu darba autore pirmo reizi ir publicējusi [1].

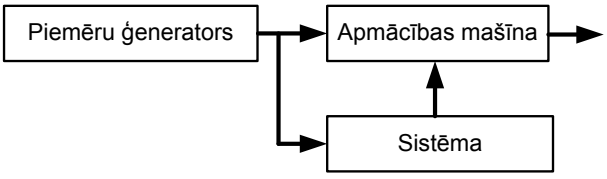
❖ Vienkāršu, bet uzskatāmu klasifikatora funkcionēšanas shēmu ir snieguši Hans un Kembers [2] (skat.1. att.). Līdzīgi šo procesu raksturo arī citi autori.



1. att. Klasifikatora funkcionēšana [2]

Klasifikatora veidošanas process sastāv no divām fāzēm. Apmācības laikā ar kāda klasifikācijas algoritma palīdzību no apmācības datiem tiek iegūts klasifikators jeb modelis, kurš var tikt atspoguļots dažādās formās, piemēram, likumu veidā. Klasifikācijas fāzē klasifikators vispirms tiek pārbaudīts ar testa datiem, kuriem klasifikācija ir iepriekš zināma. Ja klasifikatora precizitāte ir apmierinoša, to var izmanot jaunu un iepriekš neredzētu datu klasificēšanai. Šis modelis atspoguļo tikai klasifikatora funkcionēšanu, bet nav vesela klasifikācijas sistēma, lai arī klasifikators ir būtiskākā tās daļa.

❖ Čerkaskijs un Muliers [3] ir aprakstījuši vispārīgu apmācības scenāriju, kurā ietilpst trīs komponentes (skat. 2. att.).



2. att. Vispārīgs apmācības scenārijs [3]

- **Piemēru ģenerators**

  Šī komponente ģenerē ieejas datus sistēmai un apmācības mašīnai.

- **Sistēma**

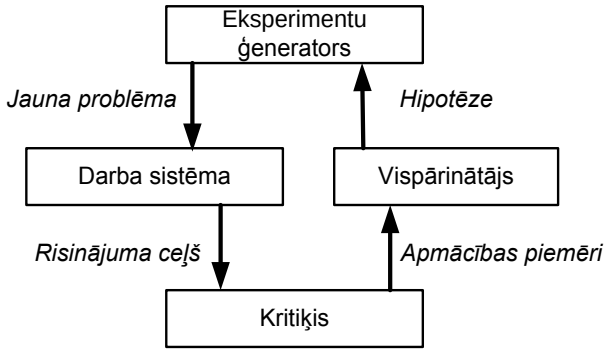
  Katram ieejā saņemtajam vektoram sistēma rada izejas vektoru, tādā veidā imitējot nejaušus novērojumus un radot varbūtīgu sadalījumu.

- **Apmācības mašīna**

  Vispārīgākajā gadījumā apmācības mašīna realizē iepriekš izvēlētu funkciju kopu. Tās uzdevums ir novērtēt iepriekš nezināmu attēlojumu starp ieejas vektoru un no *Sistēmas* saņemto izejas vektoru.

  Dotais sistēmas formulējums ir loti vispārīgs un var aprakstīt daudzas praktiskas apmācības problēmas inženierijā un statistikā, ieskaitot klasifikāciju.

❖ Nedaudz specifiskāku apmācības uzdevumu ir aprakstījis Mičels [4]. Viņa sniegtais sistēmas apmācības cikls uzlabo savu darbību ar atkārtojumu palīdzību (skat. 3. att.).



3. att. Sistēmas apmācības cikls [4]

- **Darba sistēma**

  Šis modulis risina uzdoto problēmu, izmantojot apgūto apmācības funkciju vai funkcijas. Ieejā saņemot jaunu piemēru, izejā tiek sniegts risinājuma iegūšanas ceļš.

- **Kritiķis**

  Ieejā saņemot risinājuma sasniegšanas ceļu, kritiķis rada piemērus mērķa funkcijas apmācībai.

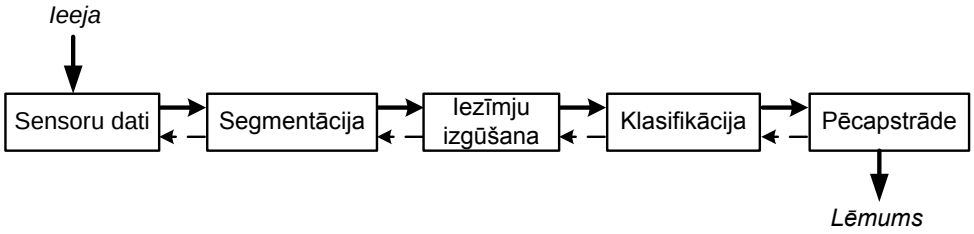
- **Vispārinātājs**

  Saņemtajiem apmācības piemēriem vispārinātājs rada hipotēzi kā savu mērķa funkcijas novērtējumu. Specifiskiem apmācības piemēriem tiek iegūts vispārinājums.

- **Eksperimentu ģenerators**

Šī moduļa loma ir tādu nākamo problēmu izvēle, kas palielinātu sistēmas kopējo apmācības līmeni. Ieejā saņemot pašreizējo hipotēzi jeb konkrētajā brīdī apgūto funkciju, darba sistēmai tiek dota jauna problēma, ko izpētīt.

❖ Tipiskas tēlu pazīšanas sistēmas funkcionēšanas modelis ir redzams 4. attēlā. [5].



4. att. Tēlu pazīšanas sistēmas funkcionēšana [5]

Sistēma ieejas tiek pārvērstas apstrādājamos datos. Segmentācijas posmā tālāk apstrādājamie objekti ir jānošķir no fona un citiem objektiem, kam seko objekta raksturīgo iezīmju iegūšana. Balstoties uz saņemtajām pazīmēm, objektam tiek noteikta klases piederība. Pēcapstrādē tiek ņemti vērā dažādi apsvērumi, lai pieņemtu lēmumu par tālāko rīcību. Kā norāda abu virzienu bultas starp elementiem, procesā iespējama arī atgriezeniska darbība.

LITERATŪRA

1. Birzniece I. Architecture of an Interactive Classification System in The Fifth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC 2012). 2012, IARIA: Lisbon, Portugal. pp. 91-100.
2. Han J., Kamber M. Data Mining: Concepts and Techniques. 2nd ed. The Morgan Kaufmann Series in Data Management Systems. 2005: Elsevier. 743 p.
3. Cherkassky V., Mulier F. Learning from Data: Conceptrs, Theory, and Methods. 2nd ed. 2007: John Wiley & Sons. 538 p.
4. Mitchell T. Machine Learning. 1997: McGraw Hill. 414 p.
5. Duda R.O., Hart P.E., Stork D.G. Pattern Classification. 2nd ed. 2001: Wiley - Interscience. 654 p.

9. pielikums. Eksperimentos izmantoto metožu saīsinājumu skaidrojums

Eksperimentos izmantotas bibliotēkas *Mulan* [1] piedāvātās daudzkategoriju klasifikācijas metodes. Visas problēmu transformācijas metodes balstās uz vienas kategorijas klasifikācijas algoritmu vai to kombināciju implementācijām programmatūrā *Weka* [2]; saglabāti atbilstošie nosaukumi. Vietās, kur nav papildu komentāru, izmantoti noklusētie *Mulan* un *Weka* parametri konkrētajām metodēm.

	Pilns nosaukums	**Paskaidrojums**
Problēmas transformācijas metodes		
BR(NB)	*BinaryRelevance (Naive Bayes)*	
BR(KStar)	*BinaryRelevance (KStar)*	
BR(IBk)	*BinaryRelevance (K Nearest Neighbours)*	
BR(Bagging)	*BinaryRelevance (Bagging)*	
BR(Stacking)	*BinaryRelevance (Stacking)*	
BR(AdaBoost)	*BinaryRelevance (AdaBoostM1)*	
BR(PART)	*BinaryRelevance (PART)*	
BR(PRISM)	*BinaryRelevance (PRISM)*	
BR(JRIP)	*BinaryRelevance (JRip)*	*JRip* – *Weka* implementācija algoritmam *Ripper*
BR(REPTree)	*BinaryRelevance (REPTRee)*	
BR(RF)	*BinaryRelevance (RandomForest (REPTree))*	
BR(J48)	*BinaryRelevance (J48)*	*J48* – *Weka* implementācija algoritmam *C4.5*
CLR	*Calibrated label ranking (J48)*	
LP	*Label Powerset (J48)*	
MLStacking	*Multi label stacking* (bāzes klasifikators – K Nearest Neighbours, metaklasifikators - *Logistic*)	Kaimiņu skaits *K* =10
MC-Copy	*Copy (J48)*	
MC-Ignore	*Ignore (J48)*	
RAkEL(J48)	*RAndom k-labELsets (Label Powerset (J48))*	
IncludeLabels	*IncludeLabelsClassifier (J48)*	
Algoritmu transformācijas metodes		
MLkNN	*Multi-Label k Nearest Neighbours*	Kaimiņu skaits *K*=10

LITERATŪRA

1. Tsoumakas G., Spyromitros-Xioufis E., Vilcek J., Vlahavas I. Mulan: A Java Library for Multi-Label Learning. Journal of Machine Learning Research, 2011. Vol. 12, pp. 2411-2414.
2. Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I.H. The WEKA Data Mining Software: An Update. SIGKDD Explorations, 2009. Vol. 11, pp. 10 - 18.

10. pielikums. Klasifikācijas rezultātu atspoguļojums dažādiem algoritmiem

Visos piemēros sniegti klasifikācijas modeļi vienām un tām pašām 2 klasēm (studiju priekšmeti *Knowledge Management Systems* un *Enterprise Architecture and Requirements Engineering* RTU studiju programmā *Biznesa informātika*), kas iegūti ar dažādiem klasifikācijas algoritmiem. Atribūti D10, A7 utt. apzīmē atbilstošo kompetenci Eiropas e-kompetenču ietvarā [1].

```
Model for KnowledgeManagementSystems
All the base classifiers:

REPTree
============

D10 = 0 : 0 (36/0) [18/1]
D10 = 1
|   A7 = 0 : 0 (8/4) [4/1]
|   A7 = 1 : 1 (2/0) [1/0]

Size of the tree : 5

REPTree
============

D10 = 0 : 0 (36/1) [19/0]
D10 = 1
|   A7 = 0 : 0 (8/1) [2/0]
|   A7 = 1 : 1 (2/0) [2/0]

Size of the tree : 5

REPTree
============
 : 0 (46/6) [23/2]

Size of the tree : 1

REPTree
============
 : 0 (46/7) [23/3]

Size of the tree : 1

REPTree
============
 : 0 (46/8) [23/3]

Size of the tree : 1

REPTree
============
 : 0 (46/6) [23/3]

Size of the tree : 1

REPTree
============
 : 0 (46/8) [23/3]

Size of the tree : 1

REPTree
============

D10 = 0 : 0 (38/2) [14/0]
D10 = 1
|   CP = 0 : 1 (0/0) [0/0]
|   CP = 1 : 1 (0/0) [0/0]
|   CP = 2 : 1 (0/0) [0/0]
|   CP = 3 : 0 (1/0) [1/0]
|   CP = 4 : 1 (0/0) [0/0]
|   CP = 5 : 1 (0/0) [0/0]
|   CP = 6
|   |   D3 = 0 : 0 (2/0) [6/2]
|   |   D3 = 1 : 1 (3/0) [2/0]
|   CP = 7 : 1 (0/0) [0/0]
|   CP = 8 : 1 (0/0) [0/0]
|   CP = 9 : 1 (0/0) [0/0]
|   CP = 10 : 1 (0/0) [0/0]
|   CP = 11 : 1 (0/0) [0/0]
|   CP = 12 : 1 (0/0) [0/0]
|   CP = 13 : 1 (0/0) [0/0]
|   CP = 14 : 1 (0/0) [0/0]
|   CP = 15 : 1 (2/0) [0/0]

Size of the tree : 21

REPTree
============

D10 = 0 : 0 (32/1) [19/1]
D10 = 1
|   A6 = 0 : 1 (10/3) [4/1]
|   A6 = 1 : 0 (4/0) [0/0]

Size of the tree : 5

REPTree
============
D10 = 0
|   A2 = 0 : 0 (30/0) [17/0]
|   A2 = 1
|   |   A3 = 0 : 0 (2/0) [0/0]
|   |   A3 = 1 : 1 (2/0) [1/0]
D10 = 1
|   A6 = 0 : 1 (8/2) [5/2]
|   A6 = 1 : 0 (4/0) [0/0]

Size of the tree : 9

Model for
EnterpriseArchitectureAndRequirementsEngineer
ing
All the base classifiers:

REPTree
============
 : 0 (46/7) [23/4]

Size of the tree : 1

REPTree
============
B2 = 0
|   A4 = 0 : 0 (34/2) [15/1]
|   A4 = 1
|   |   A7 = 0 : 1 (3/1) [2/0]
|   |   A7 = 1 : 0 (2/0) [2/0]
```

```
B2 = 1 : 1 (7/3) [4/2]

Size of the tree : 7

REPTree
============
C1 = 0 : 0 (42/4) [22/2]
C1 = 1 : 1 (4/1) [1/0]

Size of the tree : 3

REPTree
============
 : 0 (46/8) [23/5]

Size of the tree : 1

REPTree
============
B4 = 0 : 0 (39/4) [20/3]
B4 = 1 : 1 (7/2) [3/1]

Size of the tree : 3

REPTree
============
C1 = 0 : 0 (42/4) [22/2]
C1 = 1 : 1 (4/1) [1/0]
```

```
Size of the tree : 3

REPTree
============
C1 = 0 : 0 (42/4) [22/3]
C1 = 1 : 1 (4/0) [1/0]

Size of the tree : 3

REPTree
============
 : 0 (46/5) [23/2]

Size of the tree : 1

REPTree
============
B4 = 0 : 0 (43/7) [21/2]
B4 = 1 : 1 (3/0) [2/0]

Size of the tree : 3

REPTree
============
 : 0 (46/7) [23/3]

Size of the tree : 1
```

Bagging likumu piemērs uz kompetencēm balstītai datu kopai

Model for KnowledgeManagementSystems

```
AdaBoostM1: Base classifiers and their
weights:

Decision Stump

Classifications

D10 = 1 : 0
D10 != 1 : 0
D10 is missing : 0

Class distributions

D10 = 1
0       1
0.5333333333333333     0.4666666666666667
D10 != 1
0       1
0.9629629629629629     0.037037037037037035
D10 is missing
0       1
0.8695652173913043     0.13043478260869565

Weight: 1.9

Decision Stump

Classifications

D10 = 1 : 1
D10 != 1 : 0
D10 is missing : 1

Class distributions

D10 = 1
0       1
0.14634146341463417    0.8536585365853658
D10 != 1
0       1
0.7959183673469387     0.20408163265306142
D10 is missing
0       1
0.4999999999999998     0.5000000000000002
```

```
Weight: 1.53

Decision Stump

Classifications

A7 = 0 : 0
A7 != 0 : 1
A7 is missing : 1

Class distributions

A7 = 0
0       1
0.5768169273229069     0.4231830726770932
A7 != 0
0       1
0.17288135593220336    0.8271186440677967
A7 is missing
0       1
0.45101351351351326    0.5489864864864867
Weight: 0.64

Decision Stump

Classifications

A3 = 0 : 0
A3 != 0 : 1
A3 is missing : 1

Class distributions

A3 = 0
0       1
0.5170656790856332     0.48293432091436694
A3 != 0
0       1
0.12017120992671693    0.8798287900732831
A3 is missing
0       1
0.3812709129046859     0.618729087095314
Weight: 0.58
```

```
Decision Stump

Classifications

B2 = 0 : 1
B2 != 0 : 0
B2 is missing : 1

Class distributions

B2 = 0
0       1
0.2595207720895551      0.740479227910445
B2 != 0
0       1
0.820837870600563       0.17916212939943693
B2 is missing
0       1
0.3225494985237524      0.6774505014762476
Weight: 1.1

Decision Stump

Classifications

E2 = 0 : 1
E2 != 0 : 0
E2 is missing : 0

Class distributions

E2 = 0
0       1
0.46408332375680666     0.5359166762431933
E2 != 0
0       1
1.0     0.0
E2 is missing
0       1
0.5213318650557042      0.4786681349442957
Weight: 0.35

Decision Stump

Classifications

D8 = 0 : 0
D8 != 0 : 0
D8 is missing : 0

Class distributions

D8 = 0
0       1
0.5425769908956591      0.45742300910434075
D8 != 0
0       1
1.0     0.0
D8 is missing
0       1
0.5912255309515903      0.4087744690484097
Weight: 0.37

Decision Stump

Classifications

D8 = 0 : 1
D8 != 0 : 0
D8 is missing : 1

Class distributions

D8 = 0
0       1
0.4505837069486309      0.5494162930513691
D8 != 0
0       1
1.0     0.0
D8 is missing
0       1
0.4999999999999995      0.5000000000000006
Weight: 0.36

Decision Stump

Classifications

E3 = 0 : 0
E3 != 0 : 0
E3 is missing : 0

Class distributions

E3 = 0
0       1
0.5373973175596841      0.46260268244031594
E3 != 0
0       1
1.0     0.0
E3 is missing
0       1
0.5762304334653868      0.4237695665346133
Weight: 0.31

Decision Stump

Classifications

E3 = 1 : 0
E3 != 1 : 1
E3 is missing : 0

Class distributions

E3 = 1
0       1
1.0     0.0
E3 != 1
0       1
0.4607189522940937      0.5392810477059063
E3 is missing
0       1
0.5     0.5

Weight: 0.29

Number of performed Iterations: 10

Model for
EnterpriseArchitectureAndRequirementsEngineer
ing
AdaBoostM1: Base classifiers and their
weights:

Decision Stump

Classifications

B4 = 1 : 1
B4 != 1 : 0
B4 is missing : 0

Class distributions

B4 = 1
0       1
0.375   0.625
B4 != 1
0       1
0.8852459016393442      0.11475409836065574
```

```
B4 is missing
0       1
0.8260869565217391      0.17391304347826086
Weight: 1.77

Decision Stump

Classifications

B2 = 0 : 0
B2 != 0 : 1
B2 is missing : 0

Class distributions

B2 = 0
0       1
0.7086527929901423      0.29134720700985767
B2 != 0
0       1
0.26217228464419484     0.7378277153558052
B2 is missing
0       1
0.6076271186440677      0.39237288135593223
Weight: 0.92

Decision Stump

Classifications

D10 = 1 : 0
D10 != 1 : 1
D10 is missing : 1

Class distributions

D10 = 1
0       1
0.961460818498807       0.0385391815011929
D10 != 1
0       1
0.4013633787792928      0.5986366212207072
D10 is missing
0       1
0.48746050552922604     0.512539494470774
Weight: 0.64
Decision Stump

Classifications

A8 = 0 : 0
A8 != 0 : 0
A8 is missing : 0

Class distributions

A8 = 0
0       1
0.5600229804522867      0.4399770195477133
A8 != 0
0       1
1.0     0.0
A8 is missing
0       1
0.6043505723085O3       0.395649427691497

Weight: 0.42

Decision Stump

Classifications

A8 = 0 : 1
A8 != 0 : 0
A8 is missing : 0
```

```
Class distributions

A8 = 0
0       1
0.4545332976851198      0.5454667023148801
A8 != 0
0       1
1.0     0.0
A8 is missing
0       1
0.5     0.4999999999999999

Weight: 0.34

Decision Stump

Classifications

D3 = 0 : 0
D3 != 0 : 0
D3 is missing : 0

Class distributions

D3 = 0
0       1
0.5357933990441206      0.4642066009558794
D3 != 0
0       1
1.0     0.0
D3 is missing
0       1
0.5714435805694095      0.4285564194305906
Weight: 0.29

Decision Stump

Classifications

D3 = 1 : 0
D3 != 1 : 1
D3 is missing : 0

Class distributions

D3 = 1
0       1
1.0     0.0
D3 != 1
0       1
0.46398139008018596     0.5360186099198141
D3 is missing
0       1
0.5000000000000004      0.49999999999999967
Weight: 0.27

Decision Stump

Classifications

A3 = 1 : 1
A3 != 1 : 0
A3 is missing : 0

Class distributions

A3 = 1
0       1
0.26083219845124334     0.7391678015487566
A3 != 1
0       1
0.6348662460941764      0.36513375390582353
A3 is missing
0       1
0.5592356954032732      0.44076430459672683
Weight: 0.65
```

```
Decision Stump

Classifications

E3 = 0 : 1
E3 != 0 : 0
E3 is missing : 1

Class distributions

E3 = 0
0       1
0.4267055803971172     0.5732944196028829
E3 != 0
0       1
1.0     0.0
E3 is missing
0       1
0.4627221251081073     0.5372778748918926

Weight: 0.41
```

```
Decision Stump

Classifications

A8 = 0 : 0
A8 != 0 : 0
A8 is missing : 0

Class distributions

A8 = 0
0       1
0.5150334053765945     0.4849665946234055
A8 != 0
0       1
1.0     0.0
A8 is missing
0       1
0.552344310490242      0.447655689509758
Weight: 0.21

Number of performed Iterations: 10
```

AdaBoost likumu piemērs uz kompetencēm balstītai datu kopai

```
Model for KnowledgeManagementSystems
Naive Bayes Classifier
                Class
Attribute           0      1
               (0.86) (0.14)
==============================
CP
  0                1.0    1.0
  1                1.0    1.0
  2                1.0    1.0
  3                5.0    1.0
  4                1.0    1.0
  5                1.0    1.0
  6               56.0    9.0
  7                1.0    1.0
  8                1.0    1.0
  9                2.0    1.0
  10               1.0    1.0
  11               1.0    1.0
  12               1.0    1.0
  13               1.0    1.0
  14               1.0    1.0
  15               1.0    2.0
  [total]         76.0   25.0

A1
  0               39.0    5.0
  1               23.0    6.0
  [total]         62.0   11.0
A2
  0               54.0    9.0
  1                8.0    2.0
  [total]         62.0   11.0

A3
  0               54.0    7.0
  1                8.0    4.0
  [total]         62.0   11.0

A4
  0               53.0    6.0
  1                9.0    5.0
  [total]         62.0   11.0

A5
  0               47.0    9.0
  1               15.0    2.0
  [total]         62.0   11.0
```

```
A6
  0               43.0    9.0
  1               19.0    2.0
  [total]         62.0   11.0

A7
  0               54.0    6.0
  1                8.0    5.0
  [total]         62.0   11.0

A8
  0               55.0    9.0
  1                7.0    2.0
  [total]         62.0   11.0

B1
  0               39.0    6.0
  1               23.0    5.0
  [total]         62.0   11.0

B2
  0               50.0    9.0
  1               12.0    2.0
  [total]         62.0   11.0

B3
  0               57.0   10.0
  1                5.0    1.0
  [total]         62.0   11.0

B4
  0               55.0    8.0
  1                7.0    3.0
  [total]         62.0   11.0

B5
  0               56.0    9.0
  1                6.0    2.0
  [total]         62.0   11.0

C1
  0               58.0    9.0
  1                4.0    2.0
  [total]         62.0   11.0

C2
  0               55.0    9.0
```

```
  1                7.0    2.0
  [total]         62.0   11.0

C3
  0               56.0    9.0
  1                6.0    2.0
  [total]         62.0   11.0

C4
  0               58.0   10.0
  1                4.0    1.0
  [total]         62.0   11.0

D1
  0               53.0    9.0
  1                9.0    2.0
  [total]         62.0   11.0

D2
  0               58.0   10.0
  1                4.0    1.0
  [total]         62.0   11.0

D3
  0               56.0    6.0
  1                6.0    5.0
  [total]         62.0   11.0

D4
  0               57.0   10.0
  1                5.0    1.0
  [total]         62.0   11.0

D5
  0               55.0   10.0
  1                7.0    1.0
  [total]         62.0   11.0

D6
  0               53.0    9.0
  1                9.0    2.0
  [total]         62.0   11.0

D7
  0               56.0   10.0
  1                6.0    1.0
  [total]         62.0   11.0

D8
  0               52.0   10.0
  1               10.0    1.0
  [total]         62.0   11.0

D9
  0               55.0    9.0
  1                7.0    2.0
  [total]         62.0   11.0

D10
  0               53.0    3.0
  1                9.0    8.0
  [total]         62.0   11.0

E1
  0               49.0    9.0
  1               13.0    2.0
  [total]         62.0   11.0

E2
  0               53.0   10.0
  1                9.0    1.0
  [total]         62.0   11.0

E3
  0               53.0   10.0
  1                9.0    1.0
  [total]         62.0   11.0

E4
  0               51.0    9.0
  1               11.0    2.0
  [total]         62.0   11.0

E5
  0               48.0    7.0
  1               14.0    4.0
  [total]         62.0   11.0

E6
  0               55.0    9.0
  1                7.0    2.0
  [total]         62.0   11.0

E7
  0               44.0    8.0
  1               18.0    3.0
  [total]         62.0   11.0

E8
  0               55.0   10.0
  1                7.0    1.0
  [total]         62.0   11.0

E9
  0               57.0    9.0
  1                5.0    2.0
  [total]         62.0   11.0

Level
  1                1.0    1.0
  2               61.0   10.0
  [total]         62.0   11.0

Model for
EnterpriseArchitectureAndRequirementsEngineer
ing
Naive Bayes Classifier

                Class
Attribute           0      1
               (0.82) (0.18)
==============================
CP
  0                1.0    1.0
  1                1.0    1.0
  2                1.0    1.0
  3                5.0    1.0
  4                1.0    1.0
  5                1.0    1.0
  6               52.0   13.0
  7                1.0    1.0
  8                1.0    1.0
  9                2.0    1.0
  10               1.0    1.0
  11               1.0    1.0
  12               1.0    1.0
  13               1.0    1.0
  14               1.0    1.0
  15               2.0    1.0
  [total]         73.0   28.0

A1
  0               38.0    6.0
  1               21.0    8.0
  [total]         59.0   14.0

A2
  0               52.0   11.0
  1                7.0    3.0
  [total]         59.0   14.0

A3
```

```
  0                50.0   11.0
  1                 9.0    3.0
  [total]          59.0   14.0

A4
  0                49.0   10.0
  1                10.0    4.0
  [total]          59.0   14.0

A5
  0                47.0    9.0
  1                12.0    5.0
  [total]          59.0   14.0

A6
  0                44.0    8.0
  1                15.0    6.0
  [total]          59.0   14.0

A7
  0                49.0   11.0
  1                10.0    3.0
  [total]          59.0   14.0

A8
  0                51.0   13.0
  1                 8.0    1.0
  [total]          59.0   14.0

B1
  0                38.0    7.0
  1                21.0    7.0
  [total]          59.0   14.0

B2
  0                51.0    8.0
  1                 8.0    6.0
  [total]          59.0   14.0

B3
  0                56.0   11.0
  1                 3.0    3.0
  [total]          59.0   14.0

B4
  0                55.0    8.0
  1                 4.0    6.0
  [total]          59.0   14.0

B5
  0                54.0   11.0
  1                 5.0    3.0
  [total]          59.0   14.0

C1
  0                57.0   10.0
  1                 2.0    4.0
  [total]          59.0   14.0

C2
  0                54.0   10.0
  1                 5.0    4.0
  [total]          59.0   14.0

C3
  0                55.0   10.0
  1                 4.0    4.0
  [total]          59.0   14.0

C4
  0                56.0   12.0
  1                 3.0    2.0
  [total]          59.0   14.0

D1
  0                50.0   12.0
  1                 9.0    2.0
  [total]          59.0   14.0

D2
  0                55.0   13.0
  1                 4.0    1.0
  [total]          59.0   14.0

D3
  0                49.0   13.0
  1                10.0    1.0
  [total]          59.0   14.0

D4
  0                55.0   12.0
  1                 4.0    2.0
  [total]          59.0   14.0

D5
  0                53.0   12.0
  1                 6.0    2.0
  [total]          59.0   14.0

D6
  0                51.0   11.0
  1                 8.0    3.0
  [total]          59.0   14.0

D7
  0                54.0   12.0
  1                 5.0    2.0
  [total]          59.0   14.0

D8
  0                50.0   12.0
  1                 9.0    2.0
  [total]          59.0   14.0

D9
  0                51.0   13.0
  1                 8.0    1.0
  [total]          59.0   14.0

D10
  0                44.0   12.0
  1                15.0    2.0
  [total]          59.0   14.0

E1
  0                46.0   12.0
  1                13.0    2.0
  [total]          59.0   14.0

E2
  0                51.0   12.0
  1                 8.0    2.0
  [total]          59.0   14.0

E3
  0                50.0   13.0
  1                 9.0    1.0
  [total]          59.0   14.0

E4
  0                48.0   12.0
  1                11.0    2.0
  [total]          59.0   14.0

E5
  0                46.0    9.0
  1                13.0    5.0
  [total]          59.0   14.0

E6
  0                52.0   12.0
  1                 7.0    2.0
```

```
  [total]          59.0   14.0

E7
  0                44.0    8.0
  1                15.0    6.0
  [total]          59.0   14.0

E8
  0                53.0   12.0
  1                 6.0    2.0
  [total]          59.0   14.0
```

```
E9
  0                54.0   12.0
  1                 5.0    2.0
  [total]          59.0   14.0

Level
  1                 1.0    1.0
  2                58.0   13.0
  [total]          59.0   14.0
```

Naivā Beijesa klasifikācijas modeļa piemērs uz kompetencēm balstītai datu kopai

```
Model for KnowledgeManagementSystems JRIP rules:
===========
 => KnowledgeManagementSystems=0 (69.0/9.0)

Number of Rules : 1


Model for EnterpriseArchitectureAndRequirementsEngineering
JRIP rules:
===========

(B4 = 1) => EnterpriseArchitectureAndRequirementsEngineering=1 (8.0/3.0)
 => EnterpriseArchitectureAndRequirementsEngineering=0 (61.0/7.0)

Number of Rules : 2
```

JRip likumu piemērs uz kompetencēm balstītai datu kopai

```
Model for KnowledgeManagementSystems JRIP rules:
===========
 (student will hav >= 1) => KnowledgeManagementSystems=1 (9.0/4.0)
 (of knowledge manag >= 1) => KnowledgeManagementSystems=1 (3.0/0.0)
 => KnowledgeManagementSystems=0 (57.0/1.0)
 Number of Rules : 3


Model for EnterpriseArchitectureAndRequirementsEngineering JRIP rules:
===========
 (found >= 1) => EnterpriseArchitectureAndRequirementsEngineering=1 (11.0/4.0)
 => EnterpriseArchitectureAndRequirementsEngineering=0 (58.0/5.0)
 Number of Rules : 2
```

JRip likumu piemērs uz vārdu vektoriem balstītai datu kopai

LITERATŪRA

1. *European e-Competence Framework* 2012; Available from: http://www.ecompetences.eu/.

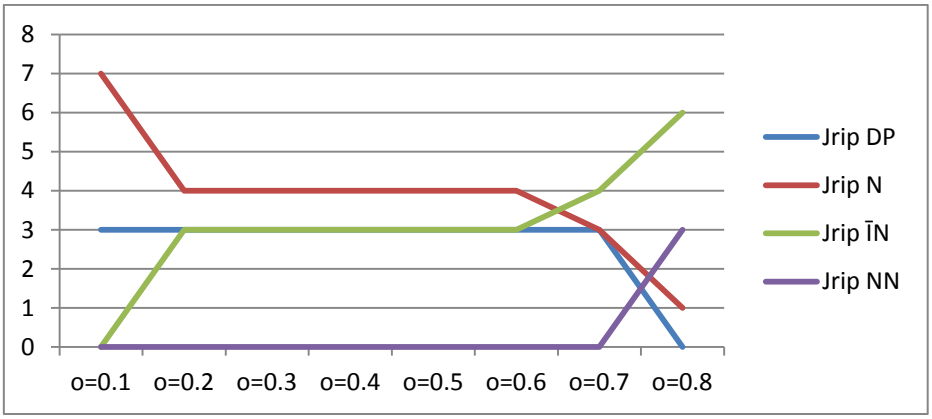
11. pielikums. Piemērotākā sliekšņa lieluma atrašana studiju priekšmetu datu kopai

Pilna datu kopa

```
Average confidence Pos: 0.234
Average confidence Neg: 0.016
```

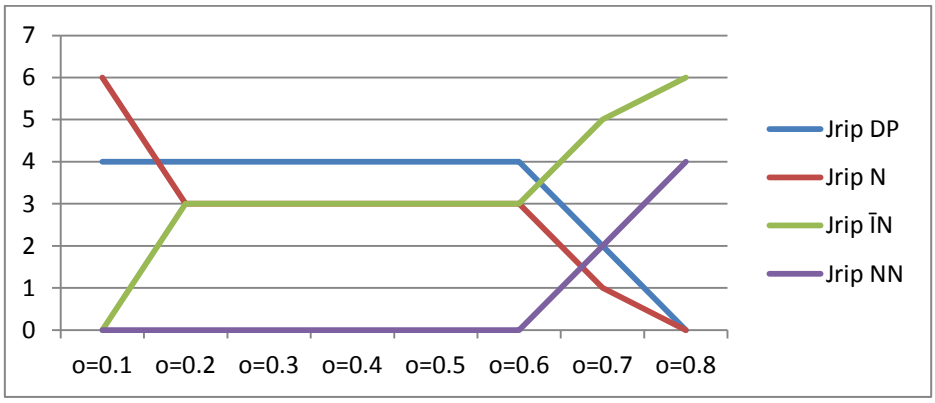
Pirmais dalījums

Courses		o=0.1	o=0.2	o=0.3	o=0.4	o=0.5	o=0.6	o=0.7	o=0.8
Jrip	DP	3	3	3	3	3	3	3	0
	N	7	4	4	4	4	4	3	1
	ĪN	0	3	3	3	3	3	4	6
	NN	0	0	0	0	0	0	0	3
	$D_{nelietderīgais}$	-	0	0	0	0	0	0	0,500
	$D_{kopējais}$	0	3	3	3	3	3	4	9



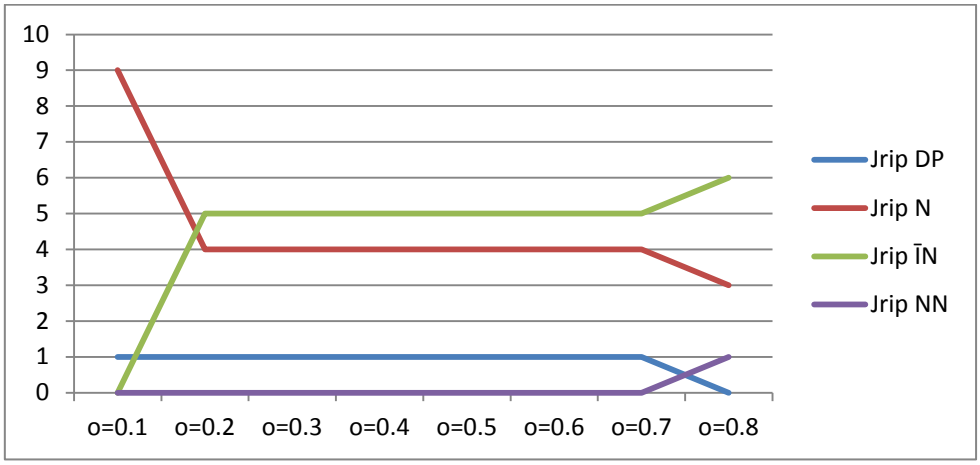
Otrais dalījums

Courses		o=0.1	o=0.2	o=0.3	o=0.4	o=0.5	o=0.6	o=0.7	o=0.8
Jrip	DP	4	4	4	4	4	4	2	0
	N	6	3	3	3	3	3	1	0
	ĪN	0	3	3	3	3	3	5	6
	NN	0	0	0	0	0	0	2	4
	$D_{nelietderīgais}$	-	0	0	0	0	0	0,400	0,667
	$D_{kopējais}$	0	3	3	3	3	3	7	10



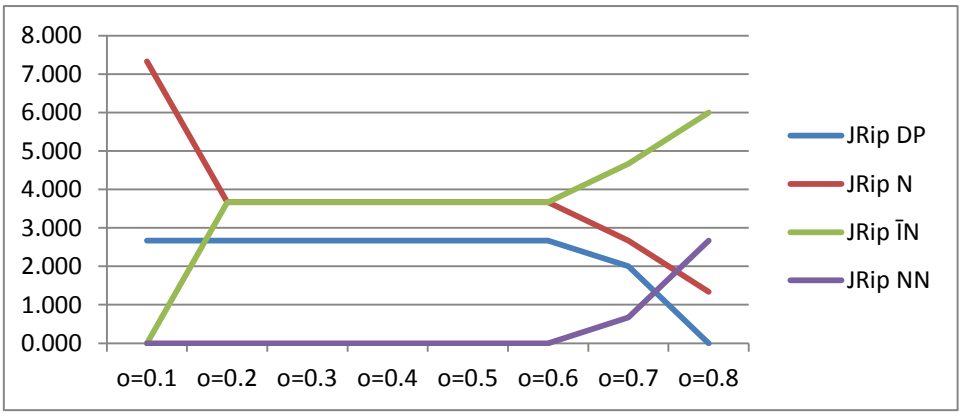
Trešais dalījums

Courses		o=0.1	o=0.2	o=0.3	o=0.4	o=0.5	o=0.6	o=0.7	o=0.8
Jrip	DP	1	1	1	1	1	1	1	0
	N	9	4	4	4	4	4	4	3
	ĪN	0	5	5	5	5	5	5	6
	NN	0	0	0	0	0	0	0	1
	$D_{nelietderīgais}$	-	0	0	0	0	0	0	0,167
	$D_{kopējais}$	0	5	5	5	5	5	5	7



Vidējais pēc 3 dalījumiem

Courses		o=0.1	o=0.2	o=0.3	o=0.4	o=0.5	o=0.6	o=0.7	o=0.8
Jrip	DP	2,667	2,667	2,667	2,667	2,667	2,667	2,000	0,000
	N	7,333	3,667	3,667	3,667	3,667	3,667	2,667	1,333
	ĪN	0,000	3,667	3,667	3,667	3,667	3,667	4,667	6,000
	NN	0,000	0,000	0,000	0,000	0,000	0,000	0,667	2,667
	$D_{nelietderīgais}$	-	0,000	0,000	0,000	0,000	0,000	0,143	0,444
	$D_{kopējais}$	0,000	3,667	3,667	3,667	3,667	3,667	5,333	8,667

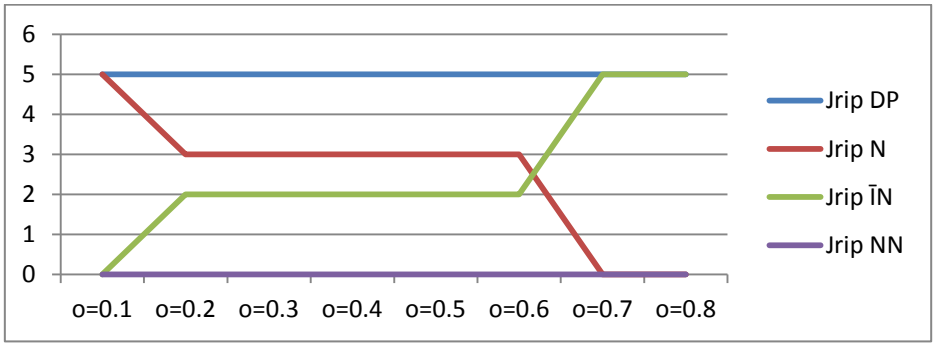


Samazināta datu kopa

```
Average confidence Pos: 0.339
Average confidence Neg: 0.053
```

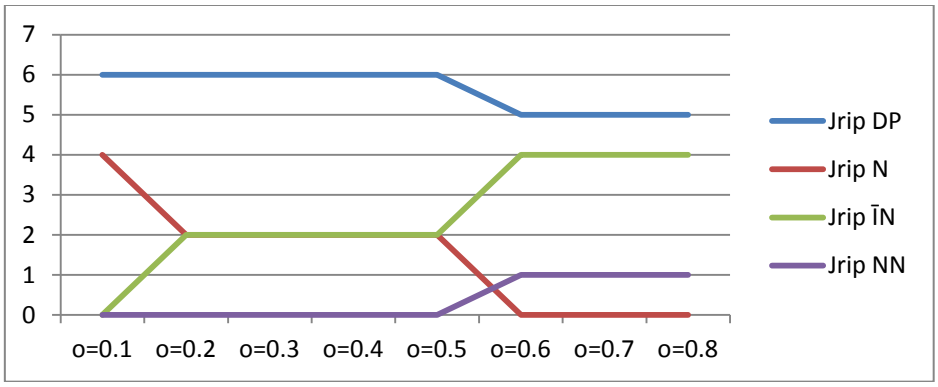
Pirmais dalījums

Courses		o=0.1	o=0.2	o=0.3	o=0.4	o=0.5	o=0.6	o=0.7	o=0.8
Jrip	DP	5	5	5	5	5	5	5	5
	N	5	3	3	3	3	3	0	0
	ĪN	0	2	2	2	2	2	5	5
	NN	0	0	0	0	0	0	0	0
	$D_{nelietderīgais}$	-	0	0	0	0	0	0	0
	$D_{kopējais}$	0	2	2	2	2	2	5	5



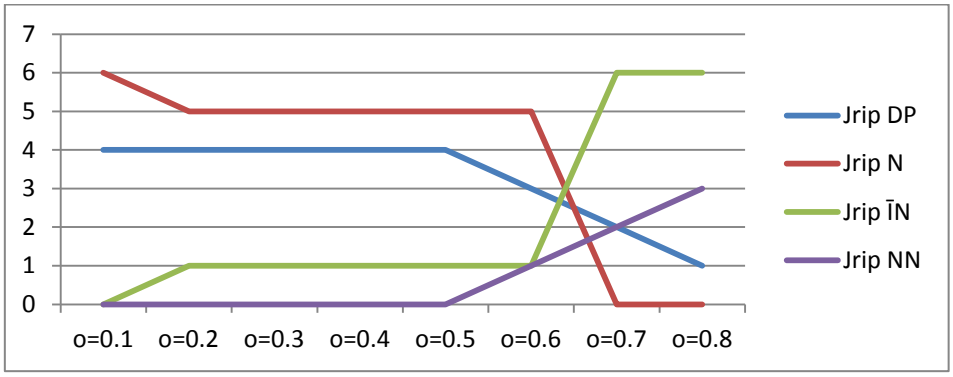
Otrais dalījums

Courses		o=0.1	o=0.2	o=0.3	o=0.4	o=0.5	o=0.6	o=0.7	o=0.8
Jrip	DP	6	6	6	6	6	5	5	5
	N	4	2	2	2	2	0	0	0
	ĪN	0	2	2	2	2	4	4	4
	NN	0	0	0	0	0	1	1	1
	$D_{nelietderīgais}$	-	0	0	0	0	0,250	0,250	0,250
	$Dk_{opējais}$	0	2	2	2	2	5	5	5



Trešais dalījums

Courses		o=0.1	o=0.2	o=0.3	o=0.4	o=0.5	o=0.6	o=0.7	o=0.8
Jrip	DP	4	4	4	4	4	3	2	1
	N	6	5	5	5	5	5	0	0
	ĪN	0	1	1	1	1	1	6	6
	NN	0	0	0	0	0	1	2	3
	$D_{nelietderīgais}$	-	0	0	0	0	1	0,333	0,500
	$D_{kopējais}$	0	1	1	1	1	2	8	9



Vidējais pēc 3 dalījumiem

Courses		o=0.1	o=0.2	o=0.3	o=0.4	o=0.5	o=0.6	o=0.7	o=0.8
Jrip	DP	5,000	5,000	5,000	5,000	5,000	4,333	4,000	3,667
	N	5,000	3,333	3,333	3,333	3,333	2,667	0,000	0,000
	ĪN	0,000	1,667	1,667	1,667	1,667	2,333	5,000	5,000
	NN	0,000	0,000	0,000	0,000	0,000	0,667	1,000	1,333
	$D_{nelietderīgais}$	-	0,000	0,000	0,000	0,000	0,286	0,200	0,267
	$D_{kopējais}$	0,000	1,667	1,667	1,667	1,667	3,000	6,000	6,333