RĪGAS TEHNISKĀ UNIVERSITĀTE
Datorzinātnes un informācijas tehnoloģijas fakultāte
Informācijas tehnoloģijas institūts

Arnis KIRŠNERS
Doktora studiju programmas „Informācijas tehnoloģija" doktorants

ĪSU LAIKA RINDU UN TO RAKSTUROJOŠO PARAMETRU APSTRĀDES SISTĒMA PROGNOZĒŠANAS UZDEVUMIEM

Promocijas darbs

Zinātniskais vadītājs
Dr. habil. sc. comp., profesors
Arkādijs BORISOVS

Rīga 2015

PATEICĪBAS

Paldies profesoram Dr.habil.sc.comp. Arkādijam Borisovam par atbalstu un padziļinātu zināšanu sniegšanu īsu laika rindu un to raksturojošo parametru apstrādes sistēmu izstrādāšanas procesā.

Paldies SIA "BRĀZMA T.V." vadītājai Tatjanai Jegorovai par ieinteresētību un atbalstu datu sagatavošanā un apkopošanā, kas bija nepieciešami eksperimentu veikšanai pārdošanas apjomu prognozēšanas sistēmas izstrādāšanā, kā arī izstrādātās sistēmas aprobācijā viņas vadītajā uzņēmumā.

Paldies Latvijas Universitātes Medicīnas fakultātes profesoram, *Dr.* Mārcim Lejam un viņa vadītajai Rīgas Austrumu klīniskās universitātes slimnīcas Zinātnes daļai par atbalstu datu iegūšanā un sagatavošanā eksperimentu veikšanai, izstrādājot un aprobējot baktēriju proliferācijas sindroma noteikšanas sistēmu.

Paldies Latvijas Organiskās sintēzes institūta un Biomedicīnas centra darbiniekiem, farmaceitiskās farmakoloģijas laboratorijas vadošajam pētniekam, *Dr. phrm.* Edgaram Liepiņam un pētniekam, *Dr. phrm.* Jānim Kūkam par izrādīto ieinteresētību un sagatavotajiem pētniecības datiem par sirds kontrakcijas spēku un to raksturojošajiem parametriem, kuri tika izmantoti eksperimentiem, izstrādājot un aprobējot sirds nekrozes riska prognozēšanas sistēmu.

Paldies kolēģiem asoc.prof. *Dr. sc. ing.* Ludmilai Aleksejevai, pētniecei *Dr. sc. ing.* Inesei Poļakai, docentam *Dr. sc. ing.* Sergejam Paršutinam par atbalstu promocijas darba tapšanā.

Rīgā, 2015. gada decembrī

Arnis Kiršners

ESF
EIROPAS SOCIĀLAIS FONDS
EIROPAS SAVIENĪBA

Šis darbs izstrādāts ar Eiropas Sociālā fonda atbalstu projektā «Atbalsts RTU doktora studiju īstenošanai».

ĪSU LAIKA RINDU UN TO RAKSTUROJOŠO PARAMETRU APSTRĀDES SISTĒMA PROGNOZĒŠANAS UZDEVUMIEM

ANOTĀCIJA

Promocijas darbs veltīts īsu laika rindu un to raksturojošo parametru apstrādes pētīšanai ar datu iegūšanas metodēm un algoritmiem, nosakot pētāmā objekta raksturlieluma iespējamo vērtību nākotnē. Promocijas darba mērķis ir izstrādāt uz datu iegūšanu balstītu īsu laika rindu un to raksturojošo parametru apstrādes pieeju kopumu prognozēšanas sistēmai, kas būtu pielietojams dažādās problēmvidēs un prognozes veikšanai izmantotu tikai analizējamā objekta raksturojošos parametrus.

Īsu laika rindu un to raksturojošo parametru apstrādes process ir grūti formalizējams, jo atrast tajā likumsakarības ir sarežģīti. Pielietotajiem algoritmiem un pieejām ir jāspēj apstrādāt atšķirīgas datu struktūras. Īsu laika rindu apstrādes process darbā formalizēts kā klasterizācijas uzdevums, bet raksturojošo parametru apstrādes process formalizēts kā klasifikācijas uzdevums. Īsu laika rindu apstrādei izstrādāta *k-vidējo sadalošā* algoritma modifikācija, kas nodrošina piemērotākā klasteru skaita noteikšanu, vadoties pēc vidējās absolūtās klasterizācijas kļūdas. Piedāvātas jaunas pieejas klasterizācijas rezultātu un raksturojošo parametru apvienošanai. Izstrādāto prognozēšanas sistēmu ietvaros realizētas jaunas pieejas iegūto klasterizācijas un klasifikācijas rezultātu analīzei.

Promocijas darbā piedāvātas trīs dažādas īsu laika rindu un to raksturojošo parametru apstrādes sistēmas, kas pielāgotas dažādām problēmvidēm: tirdzniecība, farmakoloģija un medicīna. Pirmā sistēma veic preču pārdošanas apjoma prognozēšanu nākamajam periodam, balstoties uz sistēmā ievadītajiem analizējamās preces raksturojošajiem parametriem. Otrā realizē sirds nekrozes riska prognozēšanu, pamatojoties uz indivīda raksturojošajiem parametriem. Bet trešā veic baktēriju proliferācijas sindroma noteikšanu tievajā zarnā, balstoties uz respondenta pašsajūtu raksturojošajiem parametriem. Izstrādāto sistēmu precizitāte un darbība ir pārbaudīta reālās problēmvidēs ar reāliem datiem. Sistēmu izveidē pielietoto algoritmu un pieeju izmantošana ir eksperimentāli pamatota.

Promocijas darba gaitā izstrādāto sistēmu un pieeju kopuma analīze dažādām problēmvidēm ļauj sniegt vadlīnijas līdzīgu sistēmu izveidē, kurās tiek apstrādātas īsas laika rindas un to raksturojošie parametri.

Promocijas darbs uzrakstīts latviešu valodā, kopējais darba apjoms ir 144 lappuses, tās satur ievadu, 4. nodaļas, secinājumus, 25 tabulas, 53 attēlus un 2 pielikumus. Darbā ir izmantotas atsauces uz 77 literatūras avotiem.

A SYSTEM FOR PROCESSING SHORT TIME SERIES AND THEIR CHARACTERISTIC PARAMETERS IN FORECASTING TASKS

ANNOTATION

This Thesis is dedicated to the research of the problem of processing short time-series and their characteristic parameter data by applying data mining methods and algorithms to determine the prospective future value of the analyzed object. The goal of the Thesis is to develop a data mining based set of approaches for the processing of short time-series and their characteristic parameters in a forecasting system that would be applicable in various fields and would only use the characteristic data of a analyzed object in order to determine the forecast.

The process of analysis of short time-series and their parameters is hard to formalize because the process of finding patterns in these data is very complex. The algorithms and approaches applied in such task would have to perform well with different data structures. The process of short time-series analysis is formalized in the Thesis as a clustering task; whereas the process of descriptive parameter analysis is formalized as classification task. Processing of short time-series is carried out using an adapted *k-means* algorithm, which allows determining the most eligible number of clusters based on the mean absolute clustering error. New approaches are proposed for merging of clustering results and characteristic parameters. The developed forecasting systems also include implementations of new approaches to the analysis of clustering and classification results.

The Thesis offers three different systems for short time-series and their characteristic parameter processing that are adapted to different fields of application: sales, pharmacology and healthcare. The first system makes forecasts of product sales volumes for a following period of time based on the characteristic parameters of a new product entered into the system. The second system is used to forecast the risk of heart necrosis based on the characteristic parameters of a analyzed individual. The third one carries out the forecasting of bacterial proliferation syndrome in the small intestine based on the characteristic parameters of health self-assessment of a respondent. The performance and functioning of the system is tested in practical fields with real data. The use of algorithms and approaches employed in the system implementation is experimentally substantiated.

The analysis of the system and approach collection developed and proposed in the Thesis for different fields offers guidelines for development of similar systems that can be used to process data of short time-series and their characteristic parameters.

The thesis is written in Latvian. It contains an introduction, 4 chapters, conclusions, the list of references, 2 appendixes, 53 pictures, 25 tables, 144 pages. The list of references contains 77 records.

СИСТЕМА ОБРАБОТКИ КОРОТКИХ ВРЕМЕННЫХ РЯДОВ И ИХ ХАРАКТЕРИЗУЮЩИХ ПАРАМЕТРОВ В ЗАДАЧАХ ПРОГНОЗИРОВАНИЯ

АННОТАЦИЯ

Диссертация посвящена изучению коротких временных рядов и их характеризующих параметров с применением методов и алгоритмов добычи данных, определяя возможное значение анализируемого объекта в будущем. Задача диссертационной работы — создать на основе добычи данных комплекс подходов для системы прогнозирования коротких временных рядов и их характеризующих параметров для применения в разных отраслях с целью прогнозирования на основе характеризующих параметров анализируемого объекта.

Процесс обработки коротких временных рядов и их характеризующих параметров трудно формализуем, так как найти в нем закономерности очень сложно. Использованным алгоритмам и подходам надо уметь обрабатывать различные структуры данных. Процесс обработки коротких временных рядов в диссертации формализован как задача кластеризации, а процесс обработки характеризующих параметров — задача классификации. Для обработки коротких временных рядов разработана модификация алгоритма к-средних, которая определяет наиболее подходящее количество кластеров, используя оценку средней абсолютной ошибки кластеризации. Предложены новые подходы для объединения характеризующих параметров и результатов кластеризации. В разработанных системах прогнозирования предложены новые подходы анализа результатов кластеризации и классификации.

Диссертация предлагает три разные системы прогнозирования коротких временных рядов и их характеризующих параметров в разных отраслях: торговля, фармакология и медицина. Первая система предлагает прогнозирование объёма продаж товара, основываясь на данных характеризующих параметров нового товара. Вторая — предлагает прогнозирование риска некроза сердца, на основе данных характеризующих параметров анализируемого индивида. А третья — предлагает определить синдром пролиферации бактерий в тонкой кишке, основываясь на данных характеризующих параметров о самочувствии респондента. Точность и работоспособность разработанных систем проверены в разных отраслях с реальными данными. В процессе разработки систем экспериментально подтверждена обоснованность выбранных алгоритмов и подходов.

В диссертации по результатам разработки систем прогнозирования и совокупности подходов для разных отраслей предлагаются ключевые моменты для создания похожих систем с обработкой коротких временных рядов и их характеризующих параметров.

Диссертация написана на латышском языке, содержит введение, 4 главы, выводы, список литературы, 2 приложения, 53 рисунков и 25 таблиц, общий объём работы 144 страницы. Список литературы содержит 77 позиций.

SATURS

Ievads 9
Pamatjēdzieni un terminoloģija 9
Problēmas nostādne 11
Pētījuma objekts un priekšmets 12
Pētījuma metodes 12
Pētījuma ierobežojumi 13
Darba mērķis un uzdevumi 13
Izvirzītās hipotēzes 13
Darba aktualitāte un zinātniskā novitāte 14
Darba praktiskais nozīmīgums 15
Darba aprobācija 15
1. Īsu laika rindu un to raksturojošo parametru apstrādes pamatprincipi 19
1.1. Datu iegūšana 19
1.2. Īsas laika rindas 23
1.3. Klasterizācijas algoritmu izvēle īsu laika rindu analīzei 25
1.4. Īsu laika rindu raksturojošie parametri 26
1.5. Klasifikācijas algoritmu izvēle raksturojošo parametru analīzei 28
1.6. Uzdevuma definējums 29
1.7. Uzdevuma formālā nostādne 30
1.8. Prognozēšanas sistēmas teorētiskais modelis 30
1.9. Kopsavilkums un secinājumi 32
2. Prognozēšanas sistēmas izstrādāšanā lietoto metožu un to modifikāciju apskats 33
2.1. Datu pirmapstrāde 33
2.1.1. Datu attīrīšana 33
2.1.1.1. Trūkstošu datu apstrāde 34
2.1.1.2. Trokšņainu datu apstrāde 34
2.1.2. Datu transformācija 35
2.1.2.1. Īsu laika rindu normalizācija 35
2.1.2.2. Īsu laika rindu raksturojošo atribūtu normalizācija 37
2.1.3. Īsu laika rindu raksturojošo atribūtu atlase 37
2.2. Datu klasterizācija 38
2.3. Klasterizācijas algoritmi un to modifikācijas 39

2.3.1. K-vidējo sadalošais algoritms 40
2.3.2. Aglomeratīvais hierarhiskais algoritms 42
2.3.3. Maksimālās līdzības algoritms 45
2.3.4. Modificēts k-vidējo sadalošais algoritms 46
2.3.4.1. Klasterizācijas kļūdas aprēķināšana apmācības datu kopai 47
2.3.4.2. Paraugmodeļi klasteros 48
2.3.4.3. Testēšanas datu kopas klasterizācija 49
2.3.4.4. Klasterizācijas kļūdas aprēķins testēšanas datu kopai 50
2.4. Datu klasifikācija 52
2.5. Klasifikācijas algoritmi 55
2.5.1. C4.5 algoritms 55
2.5.2. K-tuvāko kaimiņu algoritms 58
2.5.3. CN2 algoritms 59
2.5.4. Naivais Baijesa algoritms 59
2.5.5. Klasifikācijas precizitātes novērtēšana 60
2.5.6. Jauna objekta prognozēšana 64
2.6. Kopsavilkums un secinājumi 64
3. Apstrādes sistēmas prognozēšanas uzdevumiem 66
3.1. Pieprasījuma prognozēšanas sistēma 68
3.1.1. PPS uzdevuma nostādne 69
3.1.2. PPS uzbūve 70
3.1.3. PPS eksperimentu norise un iegūtie rezultāti 75
3.1.4. Kopsavilkums un secinājumi par PPS 87
3.2. Sirds nekrozes riska prognozēšanas sistēma 88
3.2.1. SNRPS uzdevuma nostādne 90
3.2.2. SNRPS uzbūve 90
3.2.3. SNRPS eksperimenti un iegūtie rezultāti 95
3.2.4. Kopsavilkums un secinājumi par SNRPS 108
3.3. Baktēriju proliferācijas sindroma noteikšanas sistēma 109
3.3.1. BPS noteikšanas sistēmas uzdevuma nostādne 111
3.3.2. BPS noteikšanas sistēmas darbības princips 112
3.3.3. Eksperimenti ar BPS datu kopu un iegūto rezultātu analīze 117
3.3.4. Kopsavilkums un secinājumi par BPS noteikšanas sistēmu 122
4. Prognozēšanas sistēmas izstrādāšanas vadlīnijas 123

4.1. Prognozēšanas sistēmas struktūra 123
4.2. Prognozēšanas sistēmas scenārija izvēle 125
4.3. Prognozēšanas sistēmas izstrādāšana un testēšana 128
4.4. Kopsavilkums un secinājumi 129
5. Rezultātu analīze un secinājumi 130
Izmantotās literatūras saraksts 134
Pielikumi 140
1. pielikums PPS klastera piederības noteikšana testēšanas kopas datiem 141
2. pielikums SNRPS klasifikatora iegūtie rezultāti dažādām apakškopām 144

IEVADS

Informācijas tehnoloģijām attīstoties un ienākot mūsu dzīvē, palielinās pieprasījums pēc to pielietojuma dažādās problēmvidēs. Informācijas tehnoloģijas arvien biežāk tiek izmantotas sarežģītu uzdevumu risināšanai, kurus agrāk varēja atrisināt tikai ar cilvēka — eksperta klātbūtni. Informācijas tehnoloģiju pielietojums ļauj radīt jaunu informāciju vai arī veikt vēsturiskas informācijas analīzi, kuras apstrādes rezultātā tiek izgūtas jaunas zināšanas, kas varētu kalpot par pamatu sarežģītu uzdevumu risinājuma iegūšanai. Pieaugot šo tehnoloģiju izmantošanas intensitātei, palielinās pieprasījums pēc ātra un precīza lēmuma — prognozes saņemšanas, pēc iespējas īsākā laika posmā. Pastāv vairākas problēmvides, piemēram, tirdzniecība, medicīna vai farmakoloģija, kurās kā datu avots kalpo laika rindas ar vēsturisku informāciju par īsu laika posmu, kas tiek iegūtas, iedarbojoties uz pētāmo objektu, kā arī aprakstoša veida informācija, kas raksturo šo objektu. Lai varētu apstrādāt šādas atšķirīgas datu struktūras, ir nepieciešams izstrādāt sistēmu, kas spētu prognozēt notikumus nākotnē, balstoties uz datu iegūšanas metodēm un algoritmiem.

Pamatjēdzieni un terminoloģija

Šajā apakšnodaļā ir paskaidroti jēdzieni un terminoloģija, kuri izmantoti promocijas darbā, lai precizētu to lietojumu.

Centroīds *(Centroid)* — punkts, kas norāda uz klastera jeb objektu grupas centru, apvienojot ap sevi klastera objektus.

Datu iegūšana *(Data Mining)* — līdz šim nezināmu, apslēptu, netriviālu, praktiski pielietojamu zināšanu atklāšana vai iegūšana ar „mašīnas" darbības (algoritmi, mākslīgais intelekts) automātisku vai pusautomātisku palīdzību, kas pieejamas cilvēka interpretācijai [63, 76].

Dimensiju lāsts (*Curse of dimensionality*) — problēma liela datu apjomu apstrādē, kas palielina izmaksas un patērēto laiku.

Disbakterioze *(Disbacteriosis)* — vides reakcija jeb pH līmeņa izmaiņa.

Entropija *(Entrophy)* — raksturo datu sakārtotību datu kopā, jo lielāka entropija, jo vienmērīgāks klašu sadalījums datu kopā. Ja entropija ir 0, tad objekti datu kopā pieder vienai klasei.

Išēmija *(Ischemia)* — nepietiekama oksigenētu asiņu perfūzija miokardā. Išēmija apzīmē ne tikai nepietiekamu miokarda apgādi ar skābekli, bet arī barības vielu nepietiekamību un nepietiekamu metabolisma gala produktu aizvadīšanu.

Īsas laika rindas *(Short time series)* — objekta mērījuma vērtības laika periodos no *T1*

līdz *Tn*, kuru periodu skaits *n* nav lielāks par 40 [21, 29, 36, 53, 72].

Kibernētika *(Cybernetics)* — zinātne par dabas un sabiedrības vispārīgiem vadības un sakaru likumiem. Par kibernētikas pamatlicēju uzskata N. Vīneru (ASV), kurš 1948. gadā monogrāfijā [71] formulēja tās pamatnostādnes. Parasti izšķir teorētisko jeb matemātisko kibernētiku un lietišķo jeb tehnisko kibernētiku. Pirmās uzdevums ir izstrādāt matemātisko aparātu un metodes dažādu vadības sistēmu pētīšanai. Otra ir zinātne par vadības procesiem tehniskās sistēmās, un ar to saistīta ražošanas procesu automatizācija un vadības sistēmu izstrāde [4].

Klasifikators (*classifier*) — modelis, kas ietver mašīnapmācības algoritma rezultātu ar informāciju par atribūtu vektora vērtību un saistību ar iepriekš definētām objektu grupām jeb klasēm.

Klasifikācija (*classification*) — objektu piederības noteikšana iepriekš definētām objektu grupām.

Klasteris (*cluster*) — līdzīgu objektu grupa ar noteiktu iezīmju kopu, objektu sabiezējums.

Klasterizācija (*clustering*) — klasteru analīzes process, kurā objekti tiek apvienoti grupās — klasteros, vadoties pēc līdzības mēriem.

Lēmumu koki (*decision trees*) — klasifikatori, kas veidoti orientēta grafa veidā, kur saknes virsotne un iekšējās virsotnes ir objektu atribūti, loki — atribūtu vērtības, bet lapas — rezultējošās klases.

Mašīnapmācība (*machine learning*) — datu apmācības process, kurš tiek nodrošināts ar datora (mašīnas) palīdzību, bet tā kā dators nevar atbildēt uz jautājumu vai tas ir kaut ko jaunu iemācījies, tad par mācīšanos tiek uzskatīta datora uzvedības maiņa, kas radīs darbības uzlabojumus nākotnē; aplūkojamā gadījumā mācīšanās nav datora, bet tā lietotāja — cilvēka vēlme, kurš padara šo procesu par apmācību [70].

Metabolisms (*metabolism*) — vielmaiņa jeb ķīmisko reakciju kopums, kas norisinās katrā organismā dzīvības uzturēšanai.

Objekta raksturojošā informācija (*information characterizing the object*) — atribūti, kas raksturo pētāmo objektu.

Objekts (*object*) — ieraksts pētāmajā datu kopā.

Oklūzija (*occlusion*) — pilnīga asinsvadu noslēgšana orgānā — asins plūsmas apturēšana.

Pārapmācība (*overfitting*) — klasifikatora apmācība, izmantojot nenozīmīgus atribūtus, kas samazina tā precizitāti, pārmērīgi pielāgojot to konkrētiem datiem.

Problēmvide (*problem domain*) — kāda no tautsaimniecības nozarēm, piemēram, tirdzniecība, farmakoloģija vai medicīna, kurā tiek risināts prognozēšanas uzdevums.

Proliferācija (*proliferation*) — pārmērīga baktēriju daudzuma veidošanās.

Raksturojošie parametri (*characteristic parameters*) — darbā tiek lietots kā jēdziens, kas apraksta atribūtu kopu. Šajā kopā ir iespējami atribūti, kas sastāv no diskrētām (piemēram, atribūts *Preces tips* = {1, 2, …, 11}, atribūts *Izmērs* ={S, M, L, XL, XXL}, u. c.) un nepārtrauktām (piemēram, atribūts *Preces cena*={0,99; 1,00; …; 49,99; …; 99,99}) vērtībām.

Reperfūzija (*reperfusion*) — asins plūsmas atjaunošana orgānā.

Sirds kontrakcija (*Cardiac contraction*) — sirds muskuļu saraušanās un atslābšana.

Stenokardija (*stenocardia*) — lēkmjveida sāpes sirds apvidū, kuras rada skābekļa trūkums sirds muskulatūrā.

Taksonomija (*taxonomy*) — zinātnes novirziens, kas pēta augsti organizētu hierarhisku īstenības sistēmu elementu (piemēram, dabas vai valodas objektu) klasifikācijas un nomenklatūras principus un metodes; arī — konkrētas sistēmas klasificēto elementu zinātnisko apzīmējumu kopums, piemēram, augu latīniskie nosaukumi, ķīmisko elementu nosaukumi [4].

Problēmas nostādne

Pastāv problēmvides, kurās par datu avotu izmanto īsas laika rindas un to raksturojošos parametrus. Īsas laika rindas raksturo objekta funkcionālās izmaiņas laika periodā, bet raksturojošie parametri — šī objekta īpašības. Piemēram, medicīnā pacientam tiek iedotas asinsspiedienu pazeminošas zāles un katru minūti tiek mērīts asinsspiediens 20 minūšu garumā. Tad asinsspiediena mērījumi laika intervālā būs īsas laika rindas, bet pacienta augums, svars, dzimums, vecums u. c. — raksturojošie parametri. Šāda un līdzīga veida problēmvidēs ir jāatrisina prognozēšanas uzdevums (piemēram, vai asinsspiedienu pazeminošās zāles palīdzēs pacientam), izmantojot tikai analizējamā objekta (piemēram, pacienta) raksturojošos parametrus, lai noteiktu iespējamo prognozējamo vērtību. Šāda dažādu veidu datu struktūru atšķirība pirms darba uzsākšanas pieprasa pret izstrādājamo prognozēšanas sistēmu izvirzīt šādas prasības:

- Tai jāspēj apstrādāt diskrēti un nepārtraukti dati. Datiem jābūt normalizētiem;
- Klasterizācijas procesā, analizējot īsas laika rindas, jānosaka piemērotākais klasteru skaits datu kopas apstrādei, vadoties pēc klasterizācijas kļūdas aprēķina, un jānosaka analizējamā objekta piederība kādam no klasteriem;

- Iegūtie klasterizācijas rezultāti jāapvieno ar īsu laika rindu raksturojošajiem parametriem, saglabājot datu integritāti, lai tos tālāk varētu izmantot datu klasifikācijā;
- Klasifikācijas procesā jānosaka objekta saikne starp klasterizācijā iegūto klasi un raksturojošajiem parametriem, maksimizējot klasifikācijas precizitāti, jutīgumu un specifiskumu;
- Jāinterpretē iegūtie klasifikācijas rezultāti atbilstoši problēmsfērā izvirzītajam uzdevumam;
- Uz izveidotās sistēmas bāzes jāveic jauna objekta klasifikācija, nosakot šī objekta prognozējamo vērtību.

Pētījuma objekts un priekšmets

Pētījuma objekts ir prognozēšanas sistēma. **Pētījuma priekšmets** ir datu iegūšanas un mašīnapmācības metodes un algoritmi, uz kuru bāzes tiek izstrādāta prognozēšanas sistēma, kas apstrādā īsas laika rindas un apvieno līdzīgus objektus grupās. Iegūtās objektu grupas tiek apvienotas ar šo objektu raksturojošajiem parametriem, lietojot klasifikācijas modeli tiek apmācīti, un attēlo prognozēšanas sistēmu pirms prognozes iegūšanas.

Pētījuma metodes

Promocijas darbā tiek izmantotas datu analīzes un datu iegūšanas metodes. Datu pirmapstrādes procesā tiek pielietotas *z-novērtējuma normalizācija ar standarta novirzi*, *pieprasījuma normalizācija ar dzīves līkni*. Atribūtu informatīvuma noteikšana tiek lietota *CfsSubsetEval* metode, bet atribūtu pārmeklēšanai *BestFirst* metode . Klasteru analīzē tiek izmantoti *k-vidējo sadalošais*, darba autora izstrādāts *modificēts k-vidējo sadalošais*, *maksimālās līdzības* un *aglomeratīvais hierarhiskais* algoritmi. Klasifikācijai tiek lietoti *ZeroR, OneR, k-tuvāko kaimiņu, CN2, C4.5, Naivais Baijesa* un *JRip* algoritmi. Klasifikācijas precizitātes novērtēšanai izmantota *10-kārtu šķērsvalidācija*, *izlaist vienu* un *datu sadalīšana apmācības un testēšanas kopās* attiecībā 70:30. No klasifikatoru iegūtajiem rezultātiem tiek veidoti *nosacījumu likumi*, apvienojot tos ar problēmvides veikto pētījumu rezultātiem. Klasterizācijas rezultātu klašu struktūru transformācijai uz problēmvidē izmantoto klašu struktūru tiek pielietota darba autora izstrādāta pieeja. No klasifikācijas rezultātiem tiek veidoti nosacījumu likumi, kuri raksturo pētāmā objekta grupu, klasifikācijas procesā noteikto klasi un problēmvides pētījumos noteikto vērtību. Analizējamā objekta prognozes vērtības aprēķinam tiek lietots matemātiskās cerības aprēķins un darba autora piedāvātā pieeja, kas balstās uz attālumu metriku.

Pētījuma ierobežojumi

Promocijas darba pētījuma ierobežojumi ir saistīti ar īsu laika rindu analīzi, kas izstrādājamajai prognozēšanas sistēmai uzliek papildus nosacījumus datu iegūšanas metožu un algoritmu izvēlē. Prognozēšanas sistēmai ir jārealizē dažādu datu struktūru apvienošana tā, lai šīs datu struktūras būtu iespējams analizēt ar datu iegūšanas metodēm un algoritmiem.

Darba mērķis un uzdevumi

Promocijas darba mērķis ir izstrādāt īsu laika rindu un to raksturojošo parametru apstrādes sistēmu prognozēšanas uzdevumiem, kas būtu lietojama dažādās problēmvidēs un balstītos uz datu iegūšanas metodēm un algoritmiem. Izvirzītā mērķa sasniegšanai nepieciešams realizēt šādus uzdevumus:

1. Izanalizēt īsu laika rindu un to raksturojošo parametru apstrādes pamatprincipus;
2. Izpildīt īsu laika rindu un to raksturojošo parametru datu pirmapstrādes pieeju analīzi;
3. Modificēt atbilstoši problēmvidei klasterizācijas algoritmu, lai tas apstrādātu īsas laika rindas, un salīdzināt to ar citiem klasterizācijas algoritmiem;
4. Izstrādāt iegūto klasterizācijas rezultātu un īsu laika rindu raksturojošo parametru datu apvienošanas pieeju;
5. Izstrādāt dažādām problēmvidēm prognozēšanas sistēmu, kas apstrādā īsas laika rindas un to raksturojošos parametrus un izdara prognozi, pamatojoties tikai uz sistēmā ievadītajiem analizējamā objekta raksturojošajiem parametriem;
6. Novērtēt izstrādātās prognozēšanas sistēmas precizitāti dažādās problēmvidēs;
7. Izstrādāt nosacījumu likumu veidošanas un pielietošanas pieejas dažādām problēmvidēm;
8. Pamatojoties uz izveidoto prognozēšanas sistēmu dažādām problēmvidēm, izstrādāt vadlīnijas līdzīgu sistēmu izstrādāšanai.

Izvirzītās hipotēzes

Promocijas darba izstrādāšanas gaitā tiek izvirzītas vairākas hipotēzes:

1. Īsu laika rindu un to raksturojošo parametru apstrādes sistēmas izstrādāšana nodrošina grūti formalizējama uzdevuma atrisināšanu ar datu iegūšanas metodēm un algoritmiem.
2. *Modificēts k-vidējo sadalošais* algoritms uzlabo piemērotākā klasteru skaita noteikšanu datu klasterizācijas procesā, analizējot īsas laika rindas.

3. Izstrādātā datu apstrādes sistēma realizē prognozēšanas uzdevumu izpildi dažādās problēmvidēs.

Pirmā hipotēze norāda uz to, ka pastāv datu struktūras, kurās par datu avotu izmanto īsas laika rindas un to raksturojošos parametrus. Hipotēze tiks uzskatīta par apstiprinātu, ja būs radīta sistēma, kas spēj apstrādāt šīs atšķirīgās datu struktūras un rezultātā tiks iegūta risināmā uzdevuma prognoze.

Otro hipotēzi raksturo tas, ka tiks pārbaudīti un salīdzināti vairāki klasterizācijas algoritmi dažādās problēmvidēs, kā tie spēj apstrādāt datus ar īsām laika rindām. Ja modificētais klasterizācijas algoritms uzrādīs labākus rezultātus, tad hipotēze būs uzskatāma par apstiprinātu.

Trešā hipotēze pamatojas idejā, ka izstrādātā sistēma ir pielietojama vai adaptējama dažādām problēmvidēm. Hipotēze tiks uzskatīta par apstiprinātu, ja izstrādātā sistēma būs izmantota vairāku prognozēšanas uzdevumu atrisinājumam dažādās problēmvidēs.

Darba aktualitāte un zinātniskā novitāte

Promocijas darba **aktualitāte** ir saistīta ar atšķirīga veida (īsas laika rindas un to raksturojošie parametri) datu struktūru analīzi. Nav zināma metode vai algoritms, kas varētu veikt īsu laika rindu un to raksturojošo parametru analīzi. Tāpēc ir svarīgi noteikt pieeju kopumu, kas lieto datu iegūšanas metodes uz algoritmus, lai varētu apstrādāt īsas laika rindas un to raksturojošos parametrus. Īsas laika rindas tiek piedāvāts apstrādāt, izmantojot klasterizāciju, lai noteiktu līdzīgas objektu grupas. Raksturojošie parametri tiek apstrādāti, lietojot klasifikāciju, lai atrastu sakarības starp šiem parametriem un klasterizācijā iegūtajiem rezultātiem. Pētāmā objekta raksturlieluma iespējamā vērtība tiek noteikta, klasificējot šī objekta raksturojošos parametrus uz izveidotā klasifikatora bāzes.

Promocijas darba **zinātniskā novitāte** ir izstrādātā prognozēšanas sistēma dažādām problēmvidēm, kas realizē īsu laika rindu un to raksturojošo parametru apstrādi. Izstrādātā sistēma spēj analizēt sarežģītas datu struktūras. Sistēmas ietvaros:

1. Izstrādāts m*odificēts k-vidējo sadalošais* algoritms, kas nodrošina īsu laika rindu apstrādi dažādās problēmvidēs, nosakot piemērotāko klasteru skaitu pēc klasterizācijas vidējās absolūtās kļūdas novērtējuma.
2. Izstrādāta divu atšķirīgu datu struktūru apvienošanas pieeja dažādās problēmvidēs.
3. Izstrādāts sistēmas iegūto klasifikācijas rezultātu atspoguļojums dažādām problēmvidēm un risināmajiem uzdevumiem.
4. Izstrādātas nosacījumu likumu veidošanas un pielietošanas pieejas dažādām problēmvidēm.

Darba praktiskais nozīmīgums

Promocijas darba praktiskais nozīmīgums ir izstrādātā pieprasījuma prognozēšanas sistēma, kas realizē jaunas preces iespējamo pieprasījumu nākamajam periodam, sistēmā ievadot tikai preces raksturojošos parametrus. Šo sistēmu iespējams pielietot uzņēmumos, kuros ir nepieciešamība prognozēt iespējamo preces pieprasījumu nākamajiem periodiem.

Izstrādāta sirds nekrozes riska prognozēšanas sistēma, kas nosaka sirds nekrozes risku laboratorijas dzīvniekam, sistēmā ievadot tikai šī dzīvnieka raksturojošo informāciju. Izstrādāto sistēmu var pielietot pētniecības iestādēs, kurās tiek izmantoti laboratorijas dzīvnieki, lai prognozētu pētāmās vielas ietekmi.

Izstrādāta baktēriju proliferācijas sindroma noteikšanas sistēma, kas nosaka vai indivīdam ir nepieciešamība veikt laktulozes testu, sistēmā ievadot tikai indivīda pašsajūtas novērtējuma parametrus. Izveidotā sistēma lietojama medicīnā — gastroenteroloģijā, nosakot baktēriju proliferācijas sindromu tievajā zarnā, prognozējot laktozes testa nepieciešamību.

Izpildīts izstrādātās sistēmas precizitātes novērtējums dažādās problēmvidēs.

Izstrādātas prognozēšanas sistēmas vadlīnijas, kas sniedz ieteikumus izstrādātājam līdzīgu sistēmu izveidošanai dažādās problēmvidēs.

Darba aprobācija

Promocijas darba izstrādāšanas procesa gaitā tika sagatavotas un publicētas 13 publikācijas:

1. Parshutin S., Kirshners A. Research on Clinical Decision Support Systems Development for Atrophic Gastritis Screening// Expert Systems with Applications. - 2013. – Vol. 40, Iss. 15, pp. 6041-6046. Citēts: ScienceDirect, SCOPUS, Thomson Reuters ISI Web of Science.
2. Kirshners A., Parshutin S. Application of Data Mining Methods in Detecting of Bacteria Proliferation Syndrome in the Small Intestine// In: European Conference on Data Analysis 2013: Book of Abstracts: European Conference on Data Analysis 2013. – 2013. - pp. 139–139.
3. Kirshners A., Parshutin S., Leja M. Research in application of data mining methods to diagnosing gastric cancer// LNAI 7377. Proceedings of the 12th Industrial Conference on Data Mining ICDM'2012. – 2012. – pp. 24-37. Citēts: SpringerLink, SCOPUS.
4. Kirshners A., Liepinsh E., Parshutin S., Kuka J., Borisov A. Risk Prediction System

for Pharmacological Problems// Automatic Control and Computer Sciences. – 2012. –Vol. 46, No.2. – pp. 57–65. Citēts: SpringerLink, SCOPUS.

5. Kirshners A., Borisov A. A Comparative Analysis of Short Time Series Processing Methods// Scientific Journal of Riga Technical University, Information Technology and Management Science, – 2012. – Vol.15. – pp. 65-69. Citēts: VINITI, EBSCO, CSA/ProQuest.
6. Kirshners A., Borisov A., Parshutin S. Robust Cluster Analysis in Forecasting Task// Proceedings of the 5th International Conference on Applied Information and Communication Technologies (AICT2012). – 2012. – pp. 77–81.
7. Parshutin S., Kirshners A. Intelligent Agent Technology in Modern Production and Trade Management// Efficient Decision Support Systems: Practice and Challenges – From Current to Future/ Book Chapter. INTECH. – 2011. – pp. 21–42. Citēts: NetLibrary; Scirus; IntechOpen; WorldCat.
8. Kirshners A. Clustering-based Behavioural Analysis of Biological Objects// Environment. Technology. Resources: Proceedings of the 8th International Scientific and Practical Conference. – 2011. – Vol.2. – pp. 24–32. Citēts: SCOPUS.
9. Kirshners A., Borisov A. Multilevel Classifier Use in a Prediction Task// Proceedings of the 17th International Conference on Soft Computing. – 2011. – pp. 403–410. Citēts: Thomson Reuters ISI Web of Science.
10. Kirshners A., Borisov A. Processing short time series with data mining methods// Scientific Proceedings of Riga Technical University, Information Technology and Management Science. – 2011. – Iss.5, Vol.49. – pp. 91–96. Citēts: VINITI, EBSCO, CSA/ProQuest.
11. Kirshners A., Parshutin S., Borisov A. Combining clustering and a decision tree classifier in a forecasting task// Automatic Control and Computer Science. – 2010. – Vol.44, No.3. – pp. 124–132. Citēts: SpringerLink, SCOPUS.
12. Kirshners A., Borisov A. Analysis of short time series in gene expression tasks// Scientific Proceedings of Riga Technical University, Information Technology and Management Science. – 2010. – Iss.5, Vol.44. – pp. 144–149. Citēts: VINITI, EBSCO, CSA/ProQuest.
13. Kirshners A., Kuleshova G., Borisov A. Demand forecasting based on the set of short time series// Scientific Proceedings of Riga Technical University, Information Technology and Management Science. – 2010. – Iss.5, Vol.44. – pp. 130–137. Citēts: VINITI, EBSCO, CSA/ProQuest.

Promocijas darba gaitā iegūtie pētījumu un sasniegumu rezultāti ar mutiskiem ziņojumiem prezentēti 9 konferencēs:

1. *European Conference on Data Analysis 2013,* Luksemburga, Luksemburga, 2013. gada 10.–12. jūlijs.
2. *53rd International Scientific Conference of Riga Technical University,* RTU, Rīga, Latvija, 2012. gada 10.–12. oktobris.
3. *12th Industrial Conference on Data Mining ICDM'2012,* Berlīne, Vācija, 2012. gada 13.–20. jūlijs.
4. *5th International Conference on Applied Information and Communication Technologies AICT2012,* LLU, Jelgava, Latvija, 2012. gada 26.–27. aprīlis.
5. *52nd International Scientific Conference of Riga Technical University,* RTU, Rīga, Latvija, 2011. gada 12.–25. oktobris.
6. *17th International Conference on Soft Computing,* Brno, Čehija, 2011. gada 15.–17, jūnijs.
7. *Informācijas tehnoloģija: Zinības un prakse, Latvijas universitāte,* Rīga, Latvija, 2010. gada 7. decembris.
8. *51st International Scientific Conference of Riga Technical University,* RTU, Rīga, Latvija, 2010. gada 11.–15. oktobris.
9. *8th International Scientific and Practical Conference,* Rēzekne, Latvija, 2010. gada 20.–22. jūnijs.

Promocijas darba rezultāti un sasniegumi izmantoti šādu projektu ietvaros:

1. No 2012. gada jūnija līdz 2013. gada martam – RTU ZP projekts «Jaunajiem zinātniekiem 2012/2013». «Lēmuma atbalsta sistēmas kuņģa vēža diagnosticēšanas uzdevuma pētījums». Vadītājs: S. Paršutins.
2. No 2010. gada janvāra līdz 2012. gada decembrim – LU projekts 2009/0220/1DP/1.1.1.2.0/09/APIA/VIAA/016 "Agrīnas audzēju diagnostikas un novēršanas starpdisciplināra izpētes grupa". Vadītājs profesors M. Leja.
3. No 2010. gada jūnija līdz 2011. gada decembrim – Latvijas-Baltkrievijas sadarbības programma zinātnē un tehnikā, līgums Nr. L7631, „Medicīnisko un bioloģisko datu intelektuālo metožu un apstrādes algoritmu kompleksa izstrāde onkoloģisko slimību diagnostikas pilnveidošanai". Vadītājs: profesors A. Borisovs.

Promocijas darbs sastāv no ievada, četrām nodaļām, rezultātu analīzes un secinājumiem, literatūras avotu saraksta un pielikumiem.

Pirmajā nodaļā sniegti īsu laika rindu un to raksturojošo parametru apstrādes pamatprincipi. Dots ieskats datu iegūšanā. Apskatīta teorija par īsām laika rindām un to raksturojošajiem parametriem. Pamatota algoritmu izvēle promocijas darbā risināmo prognozēšanas uzdevumu risinājumu realizācijai. Izvirzīta formalizētā uzdevuma nostādne.

Otrajā nodaļā aprakstītas pielietotās metodes un algoritmi, kas izmantoti prognozēšanas sistēmu izstrādāšanā. Nodaļas ietvaros aprakstīta darba autora izstrādātā *k-vidējo sadalošā algoritma modifikācija*, kas paredzēta īsu laika rindu klasterizācijai ar dažādu objektu skaitu.

Trešajā nodaļā piedāvāta prognozēšanas sistēma dažādām problēmvidēm, uz kuras pamata izstrādātas: pieprasījuma prognozēšanas sistēma, sirds nekrozes riska prognozēšanas sistēma un baktēriju proliferācijas sindroma noteikšanas sistēma. Apskatīti šo sistēmu uzbūves un darbības pamatprincipi. Aprakstīti veiktie eksperimenti un iegūtie rezultāti. Veikts izstrādāto sistēmu precizitātes novērtējums. Izdarīti secinājumi par izstrādātajām prognozēšanas sistēmām.

Ceturtajā nodaļā sniegts izklāsts par prognozēšanas sistēmas izstrādāšanas vadlīnijām, kuras palīdz izstrādātājam izvēlēties piemērotāko sistēmas izveidošanas procesu īsu laika rindu un raksturojošo parametru apstrādāšanai. Sistēmas izveidošanas process balstīts uz pieredzi, kas iegūta preču pieprasījuma, sirds nekrozes riska un baktēriju proliferācijas sindroma prognozēšanas sistēmu izstrādāšanā dažādās problēmvidēs.

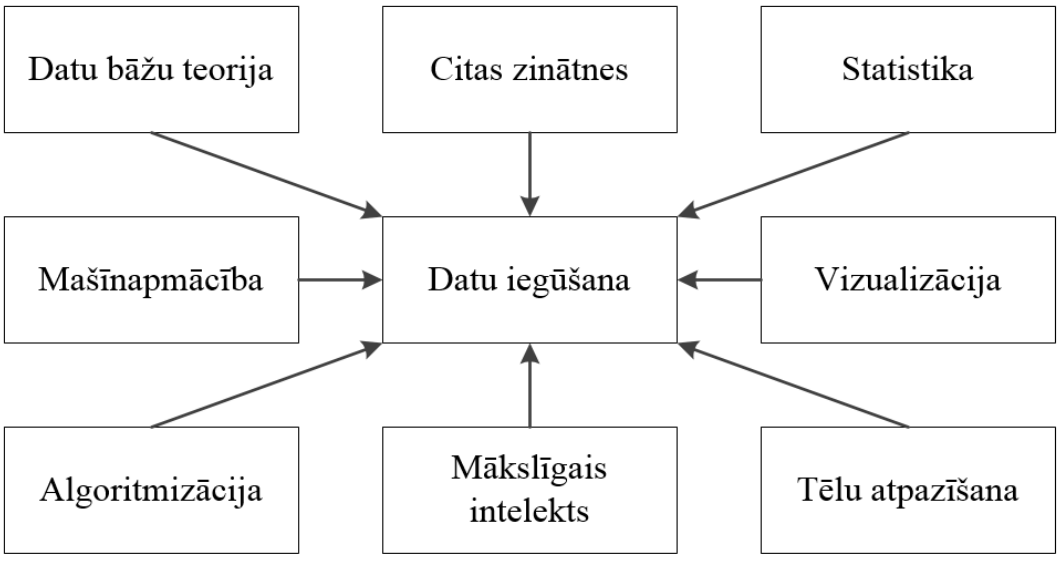
Pēdējā nodaļā veikta sasniegto rezultātu analīze un izdarīti secinājumi par izstrādātajām sistēmām un vadlīnijām.

1. ĪSU LAIKA RINDU UN TO RAKSTUROJOŠO PARAMETRU APSTRĀDES PAMATPRINCIPI

Nodaļā tiks iepazīstināts ar datu iegūšanu. Aprakstīti datu iegūšanā risināmie uzdevumi un jaunu zināšanu iegūšanas process. Izklāstīta īsu laika rindu un to raksturojošo parametru apstrādāšanas iespējas ar datu iegūšanas metodēm un algoritmiem. Veikta šo algoritmu salīdzinošā analīze, lai pamatotu to izvēli un izmantošanu prognozēšanas sistēmas izstrādāšanā. Dots uzdevuma definējums un izvirzīta uzdevuma formālā nostādne. Aprakstīts prognozēšanas sistēmas koncepts.

1.1. Datu iegūšana

Datu iegūšana [4] ir radusies vairāku zinātņu nozaru krustpunktos (skatīt 1.1. attēlu). Datu iegūšanu bieži dēvē arī par datizraci, datu ieguvi, datu intelektuālo analīzi, likumsakarību meklēšanu, šablonu (veidņu) analīzi u. c. Turpmāk darbā tiks lietots termins datu iegūšana. Jēdziens *Datu iegūšana* radās 1978. gadā, bet plašāku popularitāti tas ieguva 1990. gadā. Liela daļa datu iegūšanas metožu ir izstrādātas, balstoties uz mākslīgā intelekta teorijas bāzes.



1.1. att. Datu iegūšanas vieta citu zinātņu nozaru vidū

Datu iegūšanas pamata iezīme ir tā, ka tiek apvienotas klasiskās statistiskās analīzes un jaunas kibernētikas metodes, kas izstrādātas informācijas tehnoloģijas nozarē. Statistiskās metodes tiek iedalītas šāda veida grupās:

1. Aprakstošās statistikas analīze (sadalījuma funkcijas novērtējums, vidējā rādītāja

noteikšana, variēšanas jeb izkliedes rādītāja noteikšana, reprezentācijas jeb standartkļūdas noteikšana u. c.) [2];

2. Sakarību analīze (korelācijas un regresijas analīze, faktoru analīze, dispersijas analīze) [5];
3. Daudzdimensiju statistiskā analīze (komponentu analīze, diskriminantu analīze, daudzdimensiju regresijas analīze, kanoniskā korelācija u. c.) [2];
4. Laika rindu analīze (dinamiskie modeļi un prognozēšana) [52].

Savukārt, kibernētikas metodes [59] tiek iedalītas šādi:

1. Mākslīgie neironu tīkli (tēlu atpazīšana, klasterizācija, prognozēšana) [74, 75];
2. Ģenētiskā programmēšana (optimizācija, klasifikācija, klasterizācija) [45];
3. Asociatīvā atmiņa (analogu vai prototipu atrašana) [62];
4. Nestriktā loģika (klasifikācija, klasterizācija) [3];
5. Modeļu indukcija (klasifikācija) [58].

Datu iegūšanas uzdevumus iedala divos tipos: aprakstošie (*descriptive*) un pareģojošie jeb paredzošie (*predictive*). Aprakstoši uzdevumi veltī vairāk uzmanības analizējamo datu izpratnes uzlabošanai. Šādu modeļu galvenā priekšrocība ir iegūto rezultātu vienkāršums un caurspīdīgums. Iespējams, ka iegūtās likumsakarības var izrādīties ļoti specifiskas un to izmantošana iespējama tikai konkrētā uzdevuma ietvaros. Pie aprakstošiem uzdevumiem tiek pieskaitīta: klasteru analīze, asociatīvo likumu meklēšana un virkņu atklāšana [76].

Klasteru analīze ir analizējamās datu kopas objektu (piemēram īsas laika rindas) grupēšana pēc līdzības pazīmēm. Iegūtās grupas sauc par klasteriem. Vārds klasteris nozīmē objektu sabiezējums, kopums vai puduris [4]. Zinātniskajā literatūrā lieto arī sinonīmus — klase jeb objektu sabiezējums. Klasteru analīzi sauc arī par klasterizāciju. Klasterizācijas uzdevumos ir jānosaka objekta piederība klasei.

Asociatīvo likumu meklēšana ir atribūtu vērtību atklāšana datos un šo asociāciju formulēšana likumu kopā. Mērķis ir iegūt informāciju par dažādu vērtību parādīšanos vairākos objektos datu kopā. (Piemēram, pērkot veikalā bērnu autiņus, 60 % pircēju nopirks arī alu).

Virkņu atklāšana — informācijas iegūšana starp notikumiem, kas saistīti laikā (piemēram, pēc jauna mājokļa iegādes, divu nedēļu laikā 60 % gadījumos saimnieki iegādājas ledusskapi, bet divu mēnešu laikā 50 % saimnieku iegādājas televizoru).

Savukārt, pareģojošie uzdevumi tiek iedalīti divās daļās. Pirmajā daļā, pamatojoties uz analizējamo datu kopu ar zināmiem rezultātiem, tiek veidots modelis. Otrajā — izveidotais

modelis tiek izmantots prognozes veidošanā ar jaunu datu kopu. Pie tam, izveidotajam modelim ir jāstrādā ar augstu precizitāti (klasifikācijas precizitāte, jutīgums, specifiskums). Precizitāti katrā modelī individuāli nosaka eksperts vai empīriski noteikts precizitātes līmenis, atkarībā no risināmā uzdevuma. Pareģojošos uzdevumus iedala: klasifikācijas, regresijas un prognozēšanas uzdevumos [76].

Klasifikācijas uzdevumos tiek noteikti likumi vai likumu kopas, no kuriem nosaka jaunā analizējamā objekta piederību kādai no klasēm. Likumi tiek veidoti no informācijas par esošajiem objektiem (piemēram, īsu laika rindu raksturojošie parametri), sadalot tos klasēs. Klasifikācijas uzdevumos mērķa atribūtam (klasēm) ir diskrētas vērtības. Klasifikācijas uzdevumos objekta klases piederība ir sākotnēji definēta.

Regresijas uzdevumi ir līdzīgi klasifikācijas uzdevumiem, ar to atšķirību, ka klasi raksturo reālu skaitļu kopa [76].

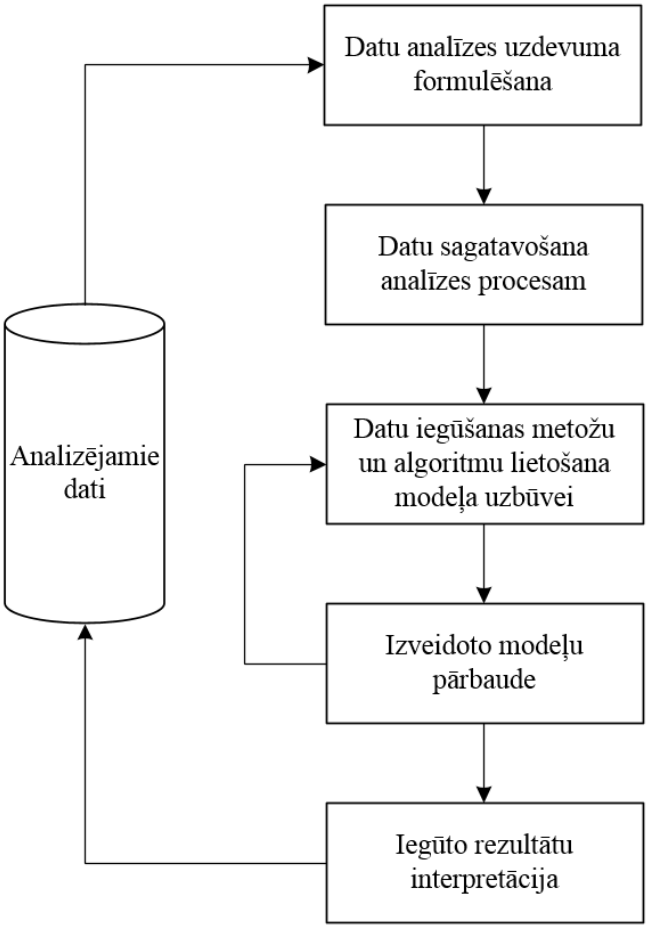
Prognozēšana ir mērķa atribūta vērtības pareģošana nākotnē [1].

Šai grupai varētu pievienot arī asociatīvo likumu meklēšanu, bet ar vienu nosacījumu, ja iegūtie rezultāti varētu tikt izmantoti dažu notikumu prognozēšanai.

Pēc risināšanas pieejām uzdevumus iedala: apmācība ar skolotāju (*supervised learning*) un apmācība bez skolotāja (*unsupervised learning*) [76]. Šāda veida nosaukums radies no termina — mašīnapmācība. Apmācībā ar skolotāju datu analīzes uzdevums tiek risināts vairākos posmos. Sākumā, lietojot kādu no datu iegūšanas algoritmiem tiek izveidots modelis — klasifikators. Tālāk notiek šī modeļa apmācība jeb tā darba kvalitātes pārbaude, ja tā izrādās neapmierinoša, tad tiek veikta atkārtota modeļa apmācība, kamēr tiks sasniegts nepieciešamais precizitātes līmenis, vai arī kļūs skaidrs, ka ar doto algoritmu, pie esošās datu kopas, nav iespējams atrisināt šo uzdevumu. Uz šo uzdevumu tipu ir attiecināmi klasifikācijas un regresijas uzdevumi.

Apmācība bez skolotāja apvieno uzdevumus, kuri raksturo aprakstošos modeļus kā, piemēram, likumsakarību atrašana preču iegādē lielveikalā, ko izdara klients. Ja tiek atrastas šāda veida likumsakarības, tas nozīmē, ka modelim tās ir jāattēlo un šeit nav runa par modeļa apmācību. Šo modeļu galvenā priekšrocība ir iespēja rast risinājumu bez iepriekšējām zināšanām par analizējamiem datiem. Pie šiem uzdevumiem pieskaitāmi klasterizācija un asociatīvo likumu meklēšana [70, 76].

Datu iegūšanas procesā iegūtajām zināšanām ir jābūt: jaunām, iepriekš nezināmām; netriviālām; praktiski pielietojamām un cilvēkam viegli interpretējamām — saprotamām. Lai datos atklātu zināšanas, nepietiek tikai ar datu iegūšanas metožu pielietošanu. Zināšanu atklāšanas process ietver vairākus datu iegūšanas procesus, kuri parādīti 1.2. attēlā.



1.2. att. Zināšanu atklāšanas process

Sākotnēji, tiek apzināts izvirzītais uzdevums un precizēts mērķis, kas tiks sasniegts, izmantojot datu iegūšanu. Svarīgi ir pareizi izvēlēties metodes un algoritmus, ar kuriem tiks sasniegts izvirzītais mērķis. Otrajā posmā dati tiek sagatavoti atbilstoši pielietojamām datu iegūšanas metodēm un algoritmiem. Trešajā posmā notiek datu iegūšanas metožu un algoritmu pielietošana. Ceturtajā posmā notiek iegūto rezultātu pārbaude un validācija. Pēdējā posmā tiek veikta iegūto rezultātu interpretācija lietotājam saprotamā formā un veidā.

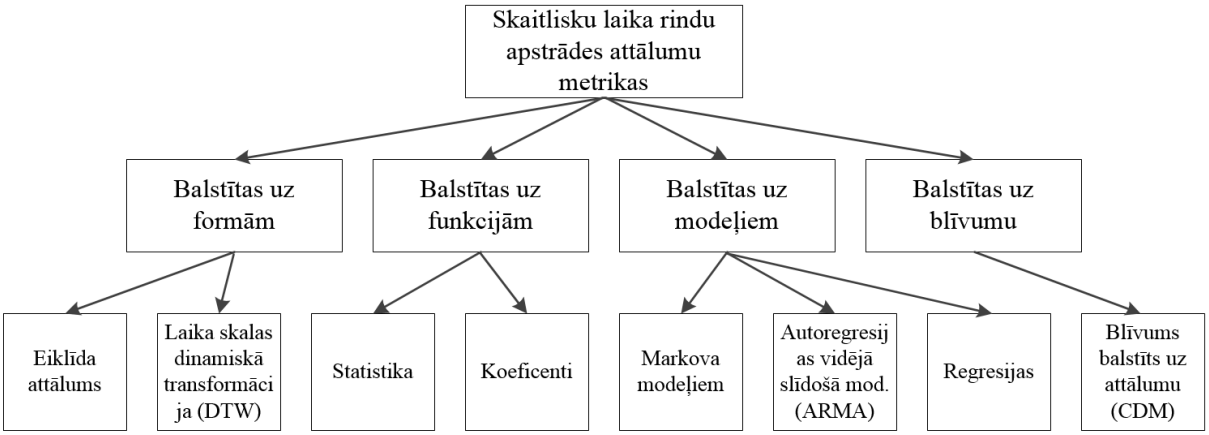
Datu iegūšanas metodes un algoritmi palīdz atrisināt dažāda veida uzdevumus, ar kuriem jāsaskaras analītiķim, piemēram, laika rindu analīze, klasterizācija vai klasifikācija. Laika rindu analīze meklē līdzības starp objektiem, analizējot vēsturiskas vērtības. Klasterizācija atrod līdzīgas objektu grupas, kuras apvieno klasteros jeb klasēs, pamatojoties uz kādu no līdzības mēriem. Klasifikācija atrod sakarības likumu veidā starp objekta raksturojošiem parametriem un to vērtībām ar iepriekš definētu objekta klasi. Iegūtie klasifikācijas rezultāti, kas attēloti likumu veidā, tiek izmantoti par pamatu jauna objekta prognozējamās vērtības vai klases (iegūstamā prognoze tiek definēta uzdevuma ietvaros) noteikšanai.

1.2. Īsas laika rindas

Statistikā laika rindu analīzes process balstās uz likumsakarību meklēšanu ilgākā laika posmā, analizējot vērtību izmaiņas laika momentos, kā rezultātā tiek noteiktas analizējamā objekta iespējamās vērtības nākotnē. Šādi procesi var būt saistīti ar ekonomisko indeksu izmaiņām, piemēram, valūtas kursu svārstībām [22]. Līdzīgi procesi tiek analizēti, piemēram, nosakot tūristu pieplūdumu nākamajai atpūtas sezonai [7] u. c. Visi šie analīzes procesi ir līdzīgi ar to, ka, veicot problēmvides funkcionēšanas izmaiņas, mainās sistēmas parametri. Tiek postulēts, ka laika rinda ir stacionāra, un jo garāki ir šīs laika rindas novērojumi, jo pastāv lielāka varbūtība efektīvāk atklāt likumsakarības laika rindās [9]. Laika rindas var būt skaitliskas vai simboliskas. Laika rindās pastāv līdzības problēma, kā noteikt vai izskaitļot līdzību *Sim(X,Y)* starp divām laika rindām $X=\{x_1, x_2, \ldots, x_n\}$ un $Y=\{y_1, y_2, \ldots, y_n\}$. Līdzības noteikšanai izmanto uz dažādām metodēm bāzētas aprēķinu metrikas, kuras ir parādītas 1.3. attēlā. Garu laika rindu salīdzināšanai piemērotākās ir metrikas, kas balstās uz funkcijām un modeļiem. Savukārt, īsāku laika rindu apstrādei izmanto uz formām balstītas attālumu metrikas, kas pielieto *Eiklīda* attāluma mēru vai laika dinamisko transformāciju [11].

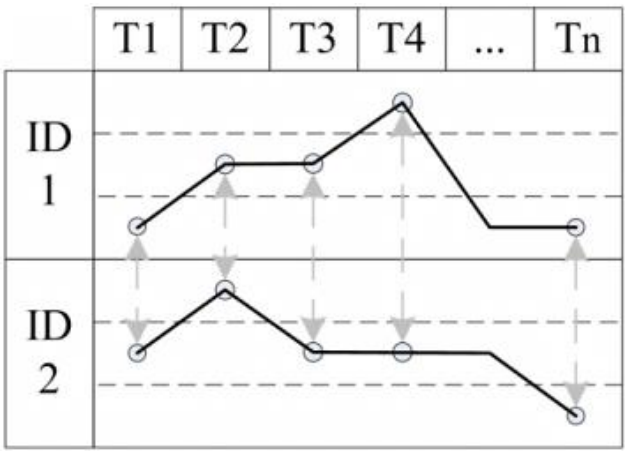
Taču reālajā dzīvē pastāv uzdevumi, kuros laika rindas garums ir samērā īss — līdz 40 periodiem, un noteikt šajās laika rindās likumsakarības ar statistiskās analīzes metodēm ir praktiski neiespējami. Pie šādiem uzdevumiem pieskaitāmi: preces dzīves cikla analīze [42, 53, 54], e-pakalpojumu analīze to ieviešanas sākuma stadijā [36], tekstila preču pārdošanas analīze pie liela produktu klāsta un īsa dzīves ilguma [33, 64], bioinformātikā gēnu ekspresijas analīze [21, 32], medicīnā kuņģa vēža rašanās riska mazināšana, analizējot biomarķierus [55] u. c. Analizējot zinātnisko literatūru, piemēram bioinformātikā, risinot gēnu ekspresijas uzdevumus, īsu laika rindu garums tiek postulēts [21] no 3 līdz 8 periodiem (laika novērojumiem). Citā zinātniskajā rakstā [72] autori norāda, ka īsa laika rinda ir diskrēti laika punkti, nekonkretizējot laika periodu skaitu. Rakstā par ekoloģisko datu analīzi [29] raksta autori analizē datus ar 30 laika periodiem, definējot tos kā īsas laika rindas. Tāpēc svarīgi ir izveidot piemērotu pieeju, kura risinātu īsu laika rindu apstrādes problēmu pie nosacīti neliela novērojumu skaita.

Ir zināmas daudz un dažādas laika rindu apstrādes metodes. Starp šīm metodēm ir liela statisko modeļu daudzveidība, kā lineārā diskriminantu analīze vai loģiskā regresija, šo modeļu izmantošana kļūst ierobežota, kad analizējamie dati ir sarežģīti un nelineāri [52], bet tieši šādas situācijas veidojas reālās dzīves sfērās. Tāpēc šādu situāciju analīzei par vairāk piemērotām tiek uzskatītas mašīnapmācības metodes.



1.3. att. Skaitlisku laika rindu apstrādes attālumu metrikas

Īsas laika rindas raksturo secīgu notikumu kopumu, kas sakārtoti pēc noteikta laika un novērojumiem. Īsu laika rindu periodu skaits ir robežas no 3 līdz 40 un tas tiek raksturots ar izteiksmi $T=\{T_1,T_2,...,T_n\}$. Īsu laika rindu analīze tiek veikta ar mērķi noteikt rindas struktūru un prognozēt nākotnes vērtības. Analīzes procesā tiek noteiktas laika rindu struktūras un likumsakarības, kas ietver trokšņainas, pīķveida, sezonālas vai cikliskas vērtības. Lai varētu veikt datu analīzi īsās laika rindās, nepieciešams nodrošināt pētāmo objektu integritāti. Atkarībā no uzdevuma specifikas ir jāizvēlas piemērotākās datu pirmapstrādes pieejas, kuras nodrošina analizējamo datu sagatavošanu datu analīzes procesam. Kā pierādīja pētījumi [31, 53], īsu laika rindu datu analīzei piemērotākā ir klasterizācija, jo tā spēj veikt divu īsu laika rindu salīdzināšanu katrā laika periodā (skatīt 1.4. attēlu) un noteikt laika rindas piederību kādai no grupām — klasteriem (objektu grupa vai grupas, kurās apvienoti tuvākie objekti, kas noteikti, piemēram, lietojot attāluma mēru).



1.4. att. Laika rindu salīdzinājums katrā laika periodā

Tāpēc, ja savstarpēji netiek salīdzinātas un analizētas laika rindas katrā laika periodā, tiek zaudēta virkne svarīgas informācijas (izmaiņas laika periodā), kas novedīs pie tā, ka nebūs iespējams analizējamos objektus apvienot grupās, kuras pēc tam tiktu ņemtas vērā, analizējot raksturojošo parametrus.

Pirms pieņemt lēmumu, kādu klasterizācijas algoritmu izmantot īsu laika rindu apstrādāšanai, nepieciešams izvērtēt virkni problēmu, kuras var rasties klasteru analīzes procesā. Uzmanība ir jāpievērš šādiem procesiem:

1) Raksturlielumu izvēles sarežģītība, uz kuru bāzes tiek veikta klasterizācija.
2) Klasterizācijas algoritmu izvēles sarežģītība. Lai izdarītu pareizo izvēli ir nepieciešams zināt algoritmu uzbūvi un to darbības specifiku.
3) Klasteru skaita izvēles problēma. Ja nav zināms iespējamais klasteru skaits, tad nepieciešams veikt virkni eksperimentu pārmeklējot dažādu klasteru skaitu un izvēloties piemērotāko, balstoties uz kādu no klasterizācijas precizitātes novērtējumiem.
4) Klasterizācijas rezultātu interpretācijas problēma. Klasteru forma, vairums gadījumos, tiek noteikta ar apvienošanas metodes tehnikas izvēli. Taču jāatceras, ka konkrētas klasterizācijas algoritms tiecas izveidot noteiktas klasteru formas, kas, piemēram, pie lielas objektu piesātinātības nelielā apgabalā un ar klasteru pārklāšanos nav pielietojamas, jo var radīt nepareizu priekšstatu par iegūtajiem klasterizācijas rezultātiem.

1.3. Klasterizācijas algoritmu izvēle īsu laika rindu analīzei

Lai izvēlētos piemērotākos klasterizācijas algoritmus īsu laika rindu analīzei tika veikta vairāku algoritmu salīdzinošā analīze, izvēloties dažādus kritērijus: vai algoritms ietilpst 10 populārāko algoritmu sarakstā [73]; algoritma iegūto rezultātu interpretējamība, šeit pozitīvu vērtējumu iegūst tie algoritmi, kuru iegūtie rezultāti ir interpretējami bez eksperta klātbūtnes un to realizācija ir pieejama eksperimentos izmantotajā programmatūra. Salīdzinošie kritēriji tiek vērtēti ar divām vērtībām: pozitīvs vērtējums (+) vai negatīvs vērtējums (–). Klasterizācijas algoritmu salīdzinošās analīzes novērtējuma rezultāti ir parādīti 1.1. tabulā. Pozitīvo vērtējumu skaita summa ir norādīta kolonā *Novērtējums*. No salīdzinošās analīzes izriet, ka piemērotākie klasterizācijas algoritmi īsu laika rindu analīzei būtu *k-vidējo sadalošais* un *maksimālās līdzības* algoritmi. Dažos eksperimentos būtu lietderīgi arī pārbaudīt *aglomeratīvā hierarhiskā* algoritma pielietošanas iespējamību, tā kā šis algoritms nākamais sarakstā pēc diviem jau izvēlētajiem algoritmiem.

1.1. tabula

Klasterizācijas algoritmu salīdzinošā analīze

	Salīdzināšanas kritēriji, iespējamās vērtības (+ vai -)					
Klasterizācijas algoritmi	Vai ir *Top10* algoritms (vieta topā)	Rezultātu interpretējamība	Realizācijai lietojamā programmatūra			Novērtējums
			Weka	*Orange Canvas*	*Statistica*	
K-vidējo sadalošais	+(3)	+	+	+	+	**5**
Maksimālās līdzības (EM)	+(5)	+	+	–	+	**4**
Pašorganizējošie neironu tīkli (SOM)	–	–	–	+	–	1
Aglomeratīvais hierarhiskais	–	+	+	+	–	3
C-vidējo sadalošais	–	+	–	–	–	1

1.4. Īsu laika rindu raksturojošie parametri

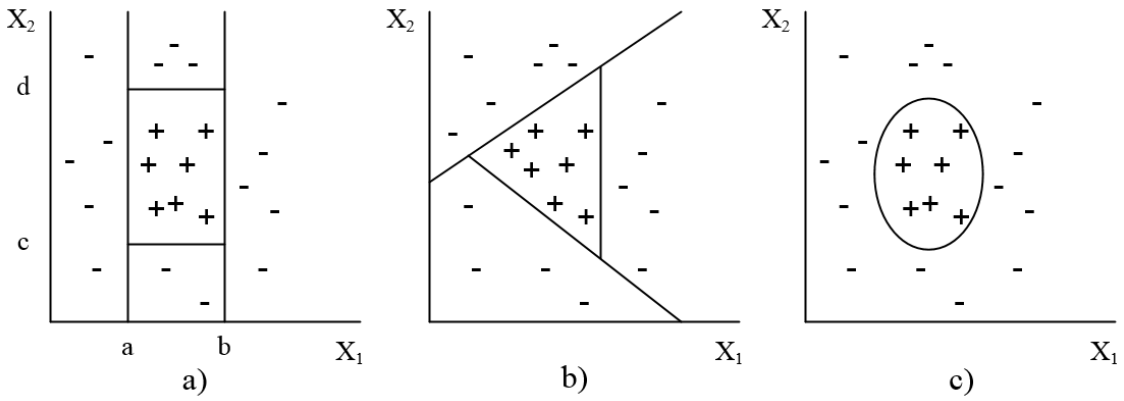
Ir situācijas, kad īsas laika rindas tiek reprezentētas kopā ar šo rindu raksturojošo informāciju, piemēram, tirdzniecībā tekstila preču pārdošanas apjomu daudzumu noteiktā laika intervālā, raksturo preces cena, daudzums, tips, krāsa, izmērs u. c. Medicīnā vielas koncentrāciju organismā pēc medikamenta lietošanas noteiktā laika intervālā raksturo indivīda (laboratorijas dzīvnieka) svars, augums, vecums, dzimums, holesterīna līmenis asinīs, kalcija daudzums u. c. Šāda veida informāciju, kas raksturo īsas laika rindas, sauc par raksturojošiem parametriem. Īsas laika rindas un to raksturojošos parametrus ir nepieciešams apstrādāt ar dažādām metodēm un pieejām, jo šīs datu struktūras ir kardināli atšķirīgas. Īsās laika rindās nepieciešams analizēt objektus un salīdzināt tos savā starpā noteiktos laika periodos. Raksturojošie parametri tiek analizēti, meklējot likumsakarības starp objektu atribūtiem, to vērtībām un raksturojošām klasēm. Datu iegūšanas uzdevumus, kas saistīti ar likumsakarību noteikšanu, vadoties pēc objektu atribūtiem, to vērtībām un raksturojošajās klases vērtībām, sauc par klasifikāciju. Klasifikācijā nosakāmo parametru bieži dēvē par atkarīgo mainīgo, bet parametrus, kuri piedalās tā noteikšanā par neatkarīgajiem mainīgajiem. Jāpievērš uzmanība aplūkojamo parametru vērtībām, ja neatkarīgie un atkarīgie mainīgie satur skaitliskas vērtības,

tad šāda veida uzdevumu sauc par regresijas uzdevumu.

Klasifikācijas un regresijas uzdevumi tiek risināti vairākos etapos. Pirmajā, tiek noteikta apmācības datu kopa, kurā ietilpst objekti, kas sastāv gan no neatkarīgiem, gan atkarīgiem mainīgajiem. Pamatojoties uz apmācības datu kopu tiek veidots modelis, kas nosaka atkarīgā mainīgā vērtību. To dēvē par klasifikācijas vai regresijas funkciju. Lai šī funkcija būtu pēc iespējas precīzāka, tai jāatbilst šādiem kritērijiem [76]:

1. Iekļauto objektu skaitam datu kopā ir jābūt pietiekoši lielam. Jo vairāk objektu būs iekļauts, jo precīzāk tikts uzbūvēta klasifikācijas vai regresijas funkcija.
2. Datu kopā jābūt iekļautiem objektiem, kas satur visas iespējamās klasifikācijas klases vai *min* un *max* vērtību apgabalu regresijas uzdevumā.
3. Katrai klasei klasifikācijas uzdevumā un katram vērtību intervālam regresijas uzdevumā izvēlētajai datu kopai jāsatur pietiekami liels objektu skaits.

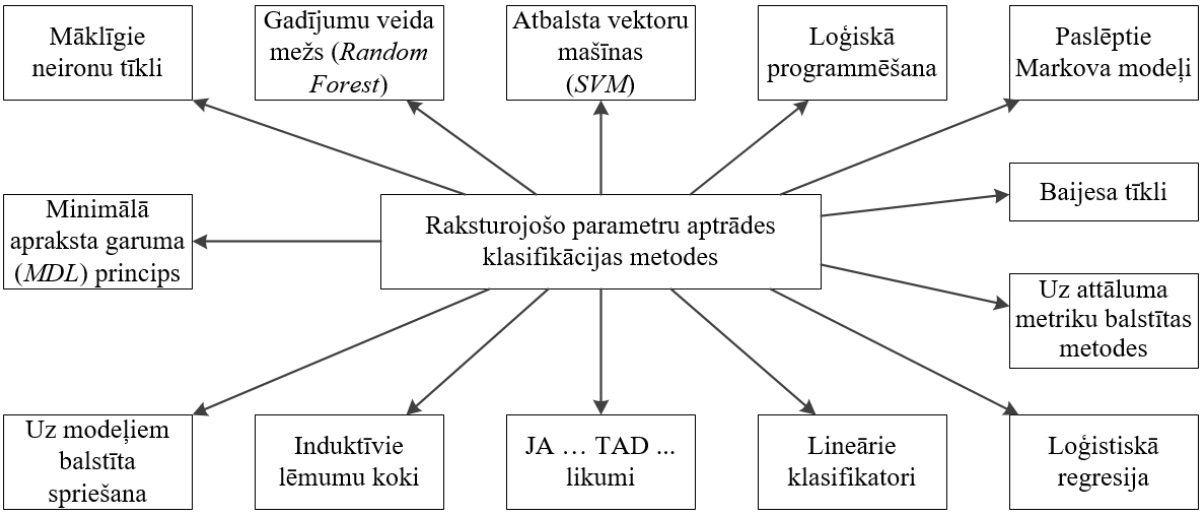
Otrajā etapā izveidoto modeli pārbauda ar testēšanas datu kopu, kurā arī ietilpst objekti, kas sastāv gan no neatkarīgiem, gan atkarīgiem mainīgiem. Modeļa iegūtie rezultāti tiek vērtēti, vadoties pēc klasifikācijas precizitātes (klasifikācijas uzdevumos), kas sīkāk izklāstīta 2.5.5. apakšpunktā, vai klasifikācijas kļūdas (regresijas uzdevumos). Ja modeļa iegūtā klasifikācijas precizitāte ir pietiekama, ko nosaka izvirzītais uzdevums, tad šo modeli iespējams lietot jauna objekta analīzei. Trešajā etapā izveidoto modeli lieto jaunu objektu analīzei, kuriem nav zināma atkarīgā mainīgā vērtība. Klasifikācijas un regresijas uzdevumiem piemīt ģeometriska interpretācija, kas parādīta 1.5. attēlā, kur divi neatkarīgi mainīgie attēloti divdimensiju telpā.



1.5. att. Klasifikācija divdimensiju telpā

Katram objektam tiek attēlots punkts uz virsmas, kur simboli «+» un «–» norāda piederību kādai no divām klasēm. No 1.5. attēla redzams, ka datiem ir spilgti izteikta struktūra, jo visi klases «+» objekti ir izvietoti centrālajā daļā. Klasifikācijas funkciju izveide

notiek uz virsmas, kas veido centrālo daļu, iekļaujot tajā ar dažādu ģeometrisko figūru palīdzību klases «+» objektus. Funkciju veids ir atkarīgs no pielietojamā klasifikācijas algoritma. 1.5. attēlā a) parādītas lineāras funkcijas, kur *y* vai *x* ir konstante, attēlā b) arī parādīta lineāru funkciju izmantošana, kur *y=F(x)*, bet attēlā c) attāluma funkcijas izmantošana. Populārākās mašīnapmācības klasifikācijas metodes ir parādītas 1.6. attēlā.



1.6. att. Mašīnapmācības klasifikācijas metodes

Galvenās problēmas, kas rodas klasifikācijas un regresijas uzdevumu risināšanā ir neapmierinoša ieejas datu kvalitāte, kuros ir gan kļūdaini dati, gan izlaisti dati vai dažādi datu tipi (skaitliski, piemēram, svars, vecums un kategoriski, piemēram, pasta indekss, dzīves vietas adrese, studenta numurs) viena atribūta ietvaros. Problēmas būtība ir, ka veidojot klasifikācijas funkciju, tajā var būt jau pieminētās kļūdas vai anomālas vērtības, kas var būt struktūrai piederoši dati. Saprotami, ka šādu modeli turpmāk nevar izmantot, jo tā iegūtie rezultāti satur pārāk lielu klasifikācijas kļūdu, tāpēc datu struktūru normalizācijai lieto datu pirmapstrādi [57, 63, 70, 76].

1.5. Klasifikācijas algoritmu izvēle raksturojošo parametru analīzei

Izvēloties klasifikācijas algoritmu, ir jāņem vērā, kā tiks interpretēti iegūtie rezultāti, vai ir nepieciešams caurskatāms un viegli saprotams iegūto rezultātu skaidrojums. Kas izmantos iegūtos rezultātus — eksperts vai ierindas sistēmas lietotājs? Cik ātri ir iegūstams algoritma izpildes rezultāts? Vai algoritms ir piemērots analizējamajai datu struktūrai? Atbildot uz šiem jautājumiem, ir iespējams izvēlēties vairākus klasifikatorus un eksperimentāli noteikt piemērotāko problēmvides uzdevuma risinājumu. Klasifikācijas algoritmu izvēle tika veikta pēc salīdzinošās analīzes, kas parādīta 1.2. tabulā.

1.2. tabula

Klasifikācijas algoritmu salīdzinošā analīze

	Salīdzināšanas kritēriji, iespējamās vērtības (+ vai -)				
Klasifikācijas algoritmi	Vai ir *Top10* algoritms (vieta topā)	Rezultātu interpretējamība (lietotā metode)	Realizācijai lietojamā programmatūra		Novērtējums
			Weka	*Orange Canvas*	
C4.5	+(1)	+ (induktīvie lēmuma koki)	+	+	**4**
k-tuvāko kaimiņu (kNN)	+(8)	+ (attāluma metrika)	+	+	**4**
Naivais Baijesa	+(9)	+ (varbūtības)	+	+	**4**
CN2	–	+ (nosacījumu likumi)	+	+	**3**
OneR	–	+ (nosacījumu likumi)	+	–	2
ZeroR	–	+ (nosacījumu likumi)	+	–	2

1.6. Uzdevuma definējums

Ir doti problēmvides dati, kuri apraksta informāciju par vēsturiskiem notikumiem un to raksturojošajiem parametriem. Ar piedāvātām datu iegūšanas metodēm un algoritmiem tiek risināts prognozēšanas uzdevums, kuram jānosaka analizējamā objekta vērtība nākotnē, balstoties tikai uz šī objekta raksturojošajiem parametriem.

Pielietojot datu iegūšanas metodes un algoritmus, ir jānosaka piemērotākie, ar kuriem ir iespējams atrisināt problēmvidē izvirzītos uzdevumus. Datu iegūšanas pieeju kopums, kas pielietojams konkrētu uzdevumu izpildei ir šāds:

1) Vēsturisku notikumu (īsas laika rindas) un to raksturojošo parametru (atribūtu) pirmapstrāde.
2) Likumsakarību noteikšana vēsturiskos datos, veidojot klasterizācijas modeli, ar kura palīdzību tiek noteiktas līdzīgas objektu grupas — klasteri. Atkarībā no problēmvides izvirzītā uzdevuma, iespējams veidot iegūto līdzīgo objektu grupu attēlojumu ar paraugmodeļu palīdzību.

3) Klasterizācijas rezultātā iegūto līdzīgo objektu grupu skaita transformācija uz raksturojošo parametru atkarīgā mainīgā (klašu) skaitu.
4) Likumsakarību noteikšana ar klasifikācijas palīdzību starp raksturojošajiem parametriem un klasterizācijas rezultātiem.
5) Klasifikācijas rezultātu interpretācija ar nosacījumu likumiem.
6) Klasterizācijas un klasifikācijas precizitātes pārbaude.
7) Pētāmā objekta raksturlieluma noteikšana, lietojot tikai šī objekta raksturojošos parametrus.

1.7. Uzdevuma formālā nostādne

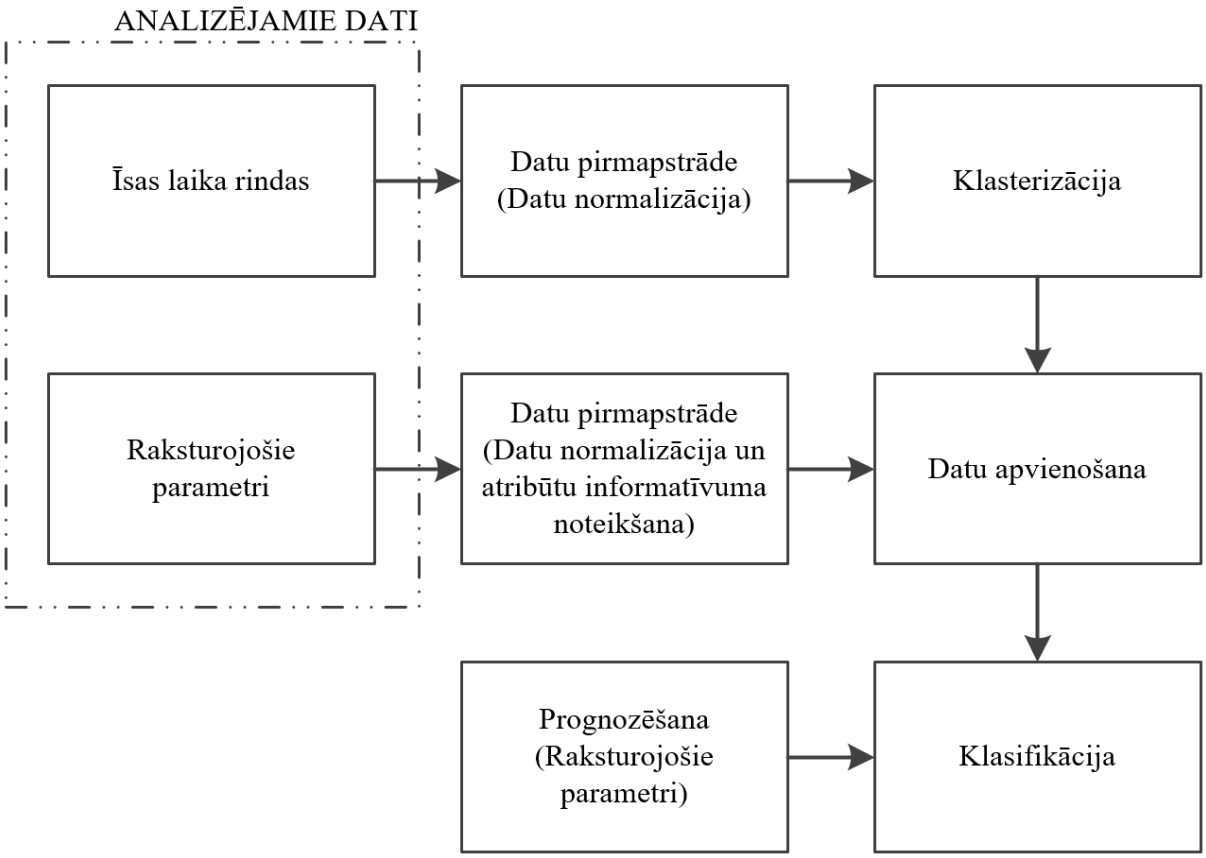
Īsu laika rindu apstrādes process darbā formalizēts kā klasterizācijas uzdevums, kura mērķis ir noteikt līdzīgas objektu grupas, pēc kurām būtu iespējams grupēt objektus analizējamajā datu kopā. Lai pārliecinātos par izvēlēto metožu piemērotību, nepieciešams izpildīt īsu laika rindu klasterizācijas algoritmu salīdzinošo analīzi un izvēlēties piemērotāko. Ir klasterizācijas uzdevumi, kad izvēlētās metodes tikai daļēji nodrošina šo uzdevumu atrisināšanu. Lai pilnībā atrisinātu šos uzdevumus, nepieciešams izpētīt izvēlētos algoritmus un izstrādāt to modifikācijas īsu laika rindu klasterizācijai. Klasterizācijai tiek izmantota apmācība bez skolotāja.

Īsu laika rindu raksturojošo parametru apstrādes process darbā formalizēts kā klasifikācijas uzdevums, kura mērķis ir apmācības procesā ar skolotāju, atrast likumsakarības starp raksturojošajiem parametriem, to vērtībām un mērķa atribūtiem — klasēm, testēšanas procesā noteikt izmantotā klasifikācijas algoritma precizitāti, un, lietojot izveidoto klasifikācijas modeli jeb klasifikatoru, prognozēt jaunā analizējamā objekta mērķa atribūtu uz apmācības procesā izveidotā modeļa bāzes.

1.8. Prognozēšanas sistēmas teorētiskais modelis

Pamatojoties uz uzdevuma formālo nostādni, analizējamo datu struktūru un literatūras analīzi, tiek piedāvāts prognozēšanas sistēmas teorētiskais modelis, kas balstīts uz datu iegūšanas metodēm un algoritmiem (skatīt 1.7. attēlu). Analizējamajiem datiem, kas sastāv no diviem datu veidiem, īsām laika rindām un to raksturojošajiem parametriem, tiek piedāvāts veikt datu pirmapstrādi. Datu pirmapstrādes procesā notiek to objektu izslēgšana no analizējamās datu kopas, kuriem ir trūkstošas vērtības. Datu normalizāciju lieto, lai izvairītos no atribūtu dominances datu kopā, kas rodas no dažādiem atribūtu vērtību (piemēram, atribūts *Vecums* un atribūts *Ienākumi*) intervālu diapazoniem [1]. Raksturojošajiem parametriem lieto

atribūtu informatīvuma noteikšanu, kas palīdz izslēgt no datu kopas neinformatīvus atribūtus, kuri negatīvi var ietekmēt klasifikācijas precizitāti [27]. Klasterizācijas procesā objekti ar īsām laika rindām tiek apvienoti grupās — klasteros pēc līdzības pazīmēm, lietojot klasterizācijas algoritmu [64]. Iegūtos klasterizācijas rezultātus apvieno ar raksturojošajiem parametriem, tad ar klasifikācijas palīdzību nosaka sakarības starp raksturojošajiem parametriem (atribūtiem) un klasterizācijas procesā noteikto klasi, lietojot kādu no klasifikācijas algoritmiem [27, 64].



1.7. att. Prognozēšanas sistēmas teorētiskais modelis

Jauna analizējamā objekta mērķa atribūta jeb klases (prognozēšanas uzdevums) noteikšana tiek veikta pamatojoties tikai uz šī objekta raksturojošajiem parametriem, lietojot izveidoto klasifikācijas modeli [27, 58, 64, 70].

1.9. Kopsavilkums un secinājumi

Nodaļas ietvaros sniegts ieskats datu iegūšanā. Izklāstīti datu iegūšanā risināmie uzdevumi. Iezīmētas promocijas darbā risināmas problēmas, kas saistītas ar īsu laika rindu un to raksturojošo parametru apstrādāšanu. Apskatīti risināmie uzdevumi un definēta uzdevumu formāla nostādne.

Apzinātas problēmas, kas saistītas ar īsu laika rindu un to raksturojošo parametru apstrādāšanu. Iezīmētas pieejas, metodes un algoritmi, kas būtu pielietojami īsu laika rindu un to raksturojošo parametru apstrādāšanai, izstrādājot prognozēšanas sistēmas dažādām problēmvidēm. Veikta klasterizācijas un klasifikācijas algoritmu salīdzinošā analīze, nosakot piemērotākos algoritmus, kuri būtu izmantojami prognozēšanas sistēmas izstrādāšanā.

Pamatojoties uz izvirzīto uzdevumu definējumu un formālo nostādni, izvēlētas datu iegūšanas metodes un algoritmi īsu laika rindu un to raksturojošo parametru apstrādāšanai.

Aprakstīts prognozēšanas sistēmas teorētiskais modelis, uz kura bāzes tika izstrādāta reāla īsu laika rindu un to raksturojošo parametru apstrādes sistēma prognozēšanas uzdevumiem dažādās problēmvidēs.

2. PROGNOZĒŠANAS SISTĒMAS IZSTRĀDĀŠANĀ LIETOTO METOŽU UN TO MODIFIKĀCIJU APSKATS

Šajā nodaļā ir apskatītas prognozēšanas sistēmu izstrādāšanā izmantotās datu iegūšanas metodes, algoritmi, algoritmu modifikācijas un aplūkoti to darbības principi. Pirmajā apakšnodaļā aprakstītas datu pirmapstrādes pieejas. Otrajā apakšnodaļā sniegta informācija par datu klasterizāciju. Trešajā apakšnodaļā izklāstīti klasterizācijas algoritmi un darba autora piedāvātās modifikācijas. Ceturtajā apakšnodaļā dots ieskats datu klasifikācijā. Piektajā apakšnodaļā aprakstīti datu klasifikācijas algoritmi. Sestajā apakšnodaļā sniegts kopsavilkums par aprakstīto metožu un algoritmu pielietojumu prognozēšanas sistēmu realizācijā.

2.1. Datu pirmapstrāde

Dažādās problēmvidēs ar datu iegūšanu risināmie uzdevumi tiek saistīti ar datu kopām, kas var ietvert trokšņainus, pretrunīgus vai trūkstošus datus. Šādu datu rašanās iemesli var būt dažādi, piemēram, visbiežāk tas ir cilvēka faktors — datu ievades operatora kļūdas vai nepilnīga informācija, kā arī tehnoloģiju nesavietojamība (piemēram, valodu īpatnības). Datu iegūšanas tehnoloģiju pielietošana šādiem neapstrādātiem, nesagatavotiem datiem, var novest pie kļūdainu rezultātu iegūšanas, tāpēc sākumā datu kopā ir jāveic datu pirmapstrāde. Tā ietver vairākas datu apstrādes tehnoloģijas.

Datu pirmapstrāde ir viens no darba ietilpīgākajiem procesiem datu iegūšanā. Datu pirmapstrāde parasti aizņem aptuveni 80 % laika no visas datu analīzes [57]. Datu attīrīšanas un transformācijas tehnoloģijas ir visbiežāk pielietotās datu pirmapstrādē [1].

Promocijas darba izstrādāšanā tika izmantotas šādas datu pirmapstrādes tehnoloģijas: datu attīrīšana, datu transformācija un atribūtu atlase.

2.1.1. Datu attīrīšana

Parasti analizējamie dati tiek attēloti taisnstūra tabulas veidā, kur kolonas ir atribūti, bet rindas — objekti jeb ieraksti. Šāda veida dati sākotnēji ir jāapstrādā, jo tie var ietvert atribūtus, kuru vērtības, piemēram, visā kolonā ir vienādas vai arī pilnīgi atšķirīgas viena no otras. Tad šāda veida atribūts vai atribūti ir jāizslēdz no analizējamās datu kopas, jo to vērtības neindividualizē pētāmos objektus [76].

Datu attīrīšana tiek pielietota, ja datu kopa satur trokšņainus, pretrunīgus vai trūkstošus datus. Šāda veida datus dēvē arī par nepilnīgiem datiem.

2.1.1.1. Trūkstošu datu apstrāde

Pastāv vairākas pieejas trūkstošu datu apstrādāšanai. Ir pieejas, kas neietekmē kopējo datu struktūru un nemaina informācijas būtību, piemēram:

a) ieraksta ignorēšana — pieeja tiek lietota, ja ierakstam nav informācijas par klasi. Tā nav īpaši efektīva, ja ieraksts satur vairākus atribūtus, kuriem nav vērtības. Pieeja ir neefektīva, ja trūkstošo vērtību skaits katram atribūtam ievērojami mainās;
b) manuāla vērtību ievadīšana — pieeja tiek lietota, ja trūkstošo datu apjoms ir minimāls un tos ievada problēmvides eksperts.

Pārējās trūkstošo datu apstrādes pieejas ietekmē kopējo datu struktūru un izmaina informācijas būtību. Datu iegūšanas procesā iegūtie kopējie rezultāti var būt atkarīgi no šīm jaunajām vērtībām, tāpēc šīs pieejas netika izmantotas promocijas darba izstrādāšanas procesā.

2.1.1.2. Trokšņainu datu apstrāde

Troksnis ir gadījuma kļūda vai neatbilstība datos (analizējamā atribūta vērtība atrodas ārpus šī atribūta definētā vērtību apgabala). Troksnis var veidoties, piemēram, devēja vai uztvērēja pieslēgšanas (nolasītās vērtības atrodas ārpus definētā vērtību diapazona) brīdī. Pastāv dažas pieejas trokšņa nogludināšanai [1]:

1) Sadalīšana grupās. Šī pieeja nogludina analizējamo vērtību, aplūkojot tuvumā esošās grupas vērtības un nepieciešamības gadījumā aizstāj to ar grupas vidējo vērtību, mediānas vērtību vai grupas augstāko vai zemāko robežu atkarībā no tā, kurai analizējamā vērtība ir tuvāk.
2) Regresija. Dati var tikt nogludināti, mēģinot tos piemērot kādai no funkcijām. Lineāra regresija ir tādas taisnes noteikšana, kurai var piemērot divus mainīgos (atribūtus), lai vienu no tiem izmantotu otra prognozēšanai. Daudzkārtēja lineāra regresija ir lineāras regresijas paplašinājums, kurā ir iesaistīti vairāk nekā divi mainīgie un dati tiek interpretēti daudzdimensiju telpā. Dati, kas neiekļaujas modelī tiek uzskatīti par troksni.
3) Klasterizācija. Nepiederošus datus var noteikt arī, izmantojot klasterizāciju, kurā vērtības pēc līdzības pazīmēm tiek apvienotas grupās vai klasteros. Vērtības, kuras neiekļaujas klasteros, var tikt uzskatītas par troksni. Vairāk par klasterizāciju lasīt 2.2. apakšpunktā.

2.1.2. Datu transformācija

Datu transformācija nepieciešama, lai uzlabotu datu ieguves procesa precizitāti un efektivitāti, piemēram, novēršot kādas datu daļas dominanci vai izveidojot jaunus atribūtus (piemēram, apvienojot atribūtus: cilvēka ķermeņa masa un augums, un izveidojot jaunu atribūtu — cilvēka ķermeņa masas indekss). Datu transformācija ietver šādas pieejas:

a) nogludināšana — pieeja tiek lietota, lai novērstu troksni;
b) agregācija — pieeja tiek lietota datu kubu (3-dimensiju objekts, kur uz dimensiju asīm izvietojas dati, kas raksturo analizējamos faktus, piemēram pārdošanas apjomi, izteikti ar gabalu skaitu vai naudas vienībām, preces atlikums noliktavā, izdevumi u.tml.) konstruēšanas laikā, pielietojot saskaitīšanas un apvienošanas operācijas;
c) vispārināšana — pieeja tiek lietota, ja zemāka līmeņa atribūti tiek aizstāti ar augstāka līmeņa koncepcijām, izmantojot taksonomiju, piemēram, atribūts "Iela" tiek vispārināts par atribūtu "Pilsēta" vai "Novads";
d) normalizācija — pieeja tiek lietota, lai atribūta vērtības ar lielu izkliedi tiktu transformētas kādā intervālā, piemēram no "-1" līdz "1";
e) atribūtu konstruēšana — pieeja tiek lietota, ja, izveidojot jaunu atribūtu (piemēram no atribūta "Garums" un "Platums", izveido jaunu atribūtu "Laukums"), iespējams, tiks uzlabots datu ieguves process.

Visbiežāk no datu transformācijas pieejām datu iegūšanas procesā tiek izmantota normalizācija, jo analizējamo datu atribūtu vērtību diapazoni ir atšķirīgi [1, 27, 63].

2.1.2.1. Īsu laika rindu normalizācija

Īsu laika rindu datu normalizācijai darbā tiek izmantotas divas metodes: *z-novērtējuma normalizācija ar standarta novirzi* [1] un *pieprasījuma normalizācija ar dzīves līkni* [64]. *Z-novērtējuma normalizācija ar standarta novirzi* ir izvēlēta, jo tiek uzskatīta par vienu no vairāk lietotajām datu normalizācijas metodēm datu iegūšanā [63], bet *pieprasījuma normalizācija ar dzīves līkni* tiek izvēlēta, jo nozares speciālisti iesaka to lietot īsu laika rindu normalizēšanai [64].

Z-novērtējuma normalizācija ar standarta novirzi balstās uz laika rindas vidējās aritmētiskās vērtības aprēķinu pēc formulas (2.1), saskaitot laika rindas visu periodu vērtības (piemēram, ja rinda sastāv no 4 laika periodiem ar vērtībām {1, 2, 3, 4}, tad summa būs 10) un dalot iegūto summu ar periodu skaitu laika rindā.

$$\overline{T_i} = \frac{1}{n}\sum_{i=1}^{n} T_i, \qquad (2.1)$$

kur T_i — laika rindas T vērtība laika periodā i;

n — periodu skaits laika rindā;

$\overline{T_i}$ — laika rindas vidējais aritmētiskais.

Standarta novirze jeb vidējā kvadrātiskā novirze raksturo varianšu [4] izkliedi ap vidējo aritmētisko. Standarta novirze ir skaitlis, kam ir tāda pati mērvienība kā attiecīgajām variantēm. Jo lielāka ir varianšu izkliede ap vidējo aritmētisko, jo lielāka ir standarta novirze [6]. Standarta novirzes σ_{T_i} aprēķinam izmanto formulu (2.2), kurā tiek summētas visas rindas vērtību un vidējā aritmētiskā starpību kvadrātiskās vērtības, kas dalītas ar periodu skaitu *n-1* un no iegūtās vērtības izvilkta kvadrātsakne.

$$\sigma_{T_i} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(T_i - \overline{T_i})^2}, \qquad (2.2)$$

kur σ_{T_i} — standarta novirze laika rindai T_i.

No formulas (2.2) izriet, ka *n-1* ir brīvības pakāpju skaits. Šāds nosaukums radies, jo visām izlases variantēm, izņemot vienu, ir iespēja variēt (piemēram, ja 4 skaitļu summa ir 10, tad pirmās trīs variantes būs brīvi izvēlētas 1, 2, 4, bet ceturtā stingri noteikta 10-(1+2+4)=3). Iespējami arī citi ierobežojumi, pie tam to skaits var būt dažāds. Aprēķinot brīvības pakāpju skaitu, izlases (periodu) apjoms jāsamazina par tik daudz vienībām, cik ir ierobežojumu. Viendimensijas empīriskajiem izlasēm parasti ir viens šāds ierobežojums — izlases vidējais aritmētiskais. Tāpēc brīvības pakāpju skaita aprēķināšanai šādu izlašu apjomu samazina par vienu. Ir pierādīts, ka, izlases apjomu aizstājot ar brīvības pakāpju skaitu, izlases standarta novirze labāk reprezentē ģenerālkopas standarta novirzi. Ja $n > 30$, atšķirība, kas rodas šos abu rādītāju savstarpējās aizvietošanas rezultātā, nav liela, un tāpēc formulā (2.2) *n-1* vietā var lietot *n* [6].

Savukārt, normalizēto vērtību katram periodam T_i' iegūst pēc formulas (2.3), kur no perioda vērtības atņem laika rindas vidējo aritmētisko un dala ar laika rindas standarta novirzi:

$$T_i^{'} = \frac{T_i - \overline{T_i}}{\sigma_{T_i}}, \qquad (2.3)$$

kur T_i' — normalizētā vērtība.

Izmantojot *z-novērtējuma normalizāciju ar standarta novirzi*, iegūto normalizēto vērtību diapazons svārstās starp negatīvām un pozitīvām vērtībām.

Pieprasījumu normalizāciju ar dzīves līkni aprēķina, ņemot objekta (laika rindas) perioda vērtību un dalot to ar šī objekta periodu vērtību summu laika rindā pēc formulas (2.4):

$$y_i = \frac{x_i}{\sum_{j=1}^{n} x_j}, \tag{2.4}$$

kur y_i — normalizētā vērtība laika periodā i;

x_i — laika rindas x vērtība laika periodā i;

x_j — laika rindas periodu vērtību summa;

n — periodu skaits laika rindā.

Šī pieeja ir īpaši piemērota normalizācijai laika rindām ar vēsturisku notikumu pieprasījumu (piemēram, preču pārdošanas apjoms pa mēnešiem gada laikā). Iegūtās vērtības tiek normalizētas tikai pozitīvajā vērtību diapazonā [64].

2.1.2.2. Īsu laika rindu raksturojošo atribūtu normalizācija

Tāpat kā īsām laika rindām arī to raksturojošajiem atribūtiem, ja šo atribūtu vērtību diapazoni atšķiras (piemēram, atribūta "Preces veids" vērtību diapazons ir no 0 līdz 11, bet atribūta "Cena" vērtību diapazons no 0,99 līdz 600), nepieciešams veikt datu normalizāciju. Raksturojošo atribūtu normalizācijai darbā tika izmantota *z-novērtējuma normalizācija ar standarta novirzi*, kuras galvenā atšķirība no īsu laika rindu normalizācijas pieejas ir tā, ka periodu vietā (laika rindas), aprēķiniem izmanto atribūtus (kolonnas) un periodu skaita n vietā lieto objektu (ierakstu) skaitu datu kopā, bet citādi formulu (2.1) un (2.3) būtība nemainās, bet formulas (2.2) vietā lieto formulu (2.5), ja n>30:

$$\sigma_{T_i} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(T_i - \overline{T})^2}. \tag{2.5}$$

2.1.3. Īsu laika rindu raksturojošo atribūtu atlase

Datu pirmapstrādes rezultātā sagatavotajā īsu laika rindu raksturojošo atribūtu datu kopā, parasti, vēl joprojām pastāv liels atribūtu daudzums, kas var radīt tā saucamo dimensiju lāstu vai pārapmācību. Lai izvairītos no šīm parādībām un varētu noteikt nozīmīgākos atribūtus, tiek pielietotas atribūtu novērtēšanas vai apakškopas atlases metodes.

Lai uzlabotu klasifikācijas algoritmu iegūto rezultātu precizitāti, tiek lietota atribūtu

informatīvuma pārbaude, kuras rezultātā no sākotnējās datu kopas tiek izslēgti neinformatīvi atribūti [28]. Promocijas darbā informatīvāko atribūtu noteikšanai tiek lietota *CfsSubsetEval* metode, kas tiek uzskatīta par vienu no populārākajām atribūtu informatīvuma noteikšanas metodēm, kuru izmanto datu iegūšanā [70]. Metode *Cfs* (Correlation-Based Feature Selection) ir uz korelāciju balstīta atribūtu atlases metode. Tās galvenā ideja ir tā, ka atribūtu atlases rezultātā tiek iegūta datu kopa, kuras atribūtiem ir augsta korelācija ar nosakāmo klasi, taču tajā pašā laikā ir pēc iespējas zemāka korelācija atribūtu starpā. Atribūtu kopas novērtējums tiek noteikts pēc formulas (2.6), kas ir Pīrsona korelācijas formula ar standartizētām vērtībām [27]:

$$r_{zc} = \frac{k\overline{r_{zi}}}{\sqrt{k + k(k-1)\overline{r_{ii}}}}, \qquad (2.6)$$

kur r_{zc} — korelācijas koeficients (novērtējums);

k — atribūtu skaits;

$\overline{r_{zi}}$ — vidējā atribūta — klases korelācija;

$\overline{r_{ii}}$ — vidējā atribūta — atribūta korelācija.

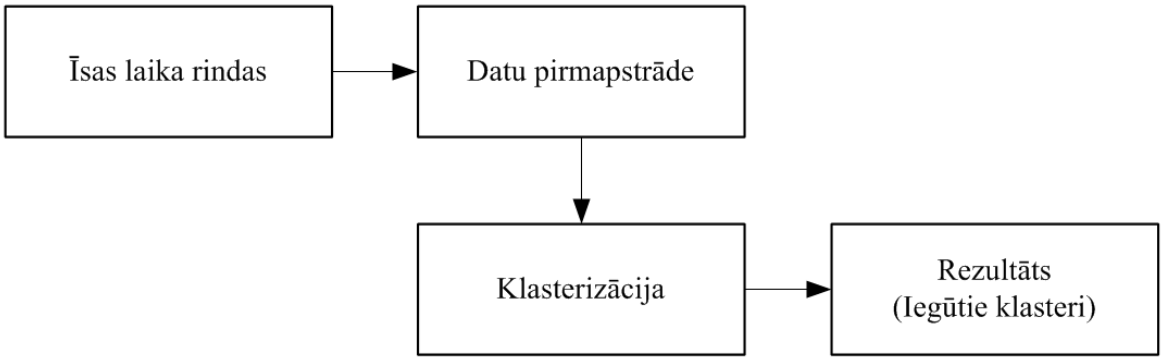
Tāpat kā lielākā daļa atribūtu atlases metožu, arī *CfsSubsetEval* pārmeklēšanai lieto *BestFirst* metodi. *BestFirst* pārmeklēšanas metode uzsāk darbu ar tukšu atribūtu sarakstu, tad tiek ģenerētas visas iespējamās atribūtu kombinācijas no viena atribūta. Tiek izvēlēta kombinācija, kurai ir lielākais novērtējums. Tālāk šī kombinācija tiek papildināta ar vienu no atribūtiem visās iespējamās kombinācijās. Ja papildinot atribūtus, netiek iegūts novērtējuma uzlabojums, tad pārmeklēšana atkāpjas par soli atpakaļ un turpina pārmeklēt nākamo labāko neapskatīto atribūtu kombināciju. Ja pārmeklēšanai tiek atvēlēts pietiekami daudz laika, tad metode aplūko visu atribūtu sarakstu, taču parasti pārmeklēšanas laiks tiek ierobežots, norādot pārmeklējamo kombināciju skaitu, pie kura vairs netiek sasniegts novērtējuma uzlabojums. Šī metode strādā līdzīgi kā *kalnā kāpšanas* algoritms [27].

Atribūtu informatīvuma pārbaudes precizitātes vērtēšanai tiek lietota *k-kārtu šķērsvalidācija* [70].

2.2. Datu klasterizācija

Datu klasterizācija ir klasteru analīze jeb objektu grupēšana klasteros datu kopā, tā lai līdzība starp objektiem vienā klasterī būtu lielāka nekā līdzība starp dažādu klasteru objektiem. Objektiem klasterī ir jāsatur līdzības pazīmes, pēc kurām notiek to apvienošana.

Salīdzinot divus objektus $x=\{x_1, x_2, ..., x_n\}$ un $y=\{y_1, y_2, ..., y_n\}$, līdzība tiek noteikta, pamatojoties uz kādu no līdzības metrikām vai attāluma mēriem. Promocijas darbā tiek analizēti gadījumi, kad laika rindu garums ir vienāds. Klasterizācijas algoritmi izmanto dažādas pieejas klasteru veidošanai. Populārākie no tiem izmanto uz attāluma vai blīvuma balstītas funkcijas [73]. Klasterizācijas rezultātā iegūst informāciju par sagrupētajiem objektiem klasterī. Īsu laika rindu apstrādāšanas process ir parādīts 2.1. attēlā.



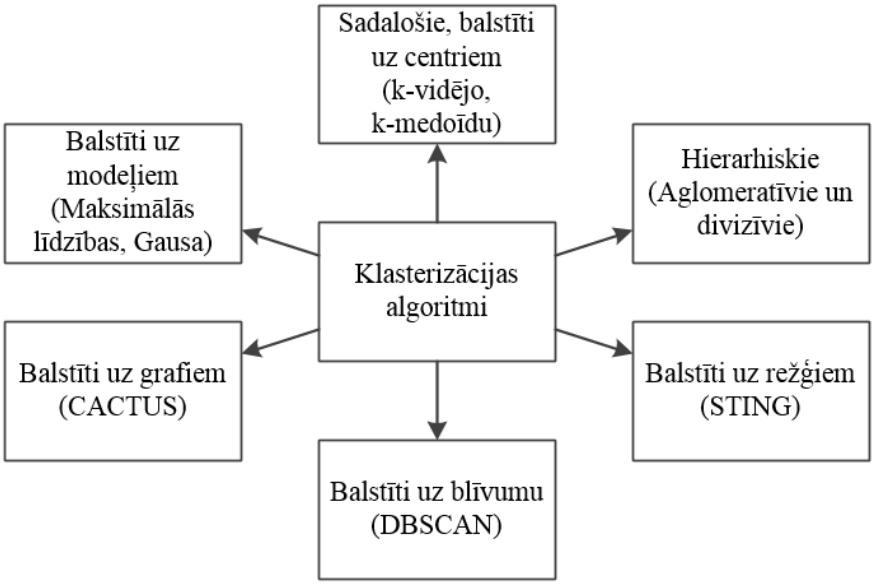
2.1. att. Īsu laika rindu apstrādāšanas process

Pastāv vairākas klasterizācijas pieejas, kā veidot klasterus. Viena no pieejām veido klasteru hierarhijas, kas apvieno vai sadala klasterus un ierakstus tajos, šādus algoritmus sauc par hierarhiskajiem klasterizācijas algoritmiem. Cita pieeja meklē klasteru centrus un iteratīvi ap šiem centriem atrod tuvākos objektus, veidojot klasterus. Šādus algoritmus dēvē par sadalošajiem klasterizācijas algoritmiem. Cita pieeja lieto gadījuma veida sadalījuma kopu, kurā ar blīvuma funkcijas palīdzību tiek meklēti klasteri, šāda pieeja balstās uz modeļiem. Retāk tiek lietoti klasterizācijas algoritmi, kuru darbība balstās uz grafiem, blīvumu vai režģiem, kuri sadala atribūtu telpu. Biežāk lietotās klasterizācijas pieejas un to populārākie algoritmi ir parādīti 2.2. attēlā.

Prognozēšanas sistēmas izstrādāšanā tiks lietoti tie klasterizācijas algoritmi, kuri tika atlasīti klasterizācijas algoritmu salīdzinošās analīzes rezultātā (skatīt 1.1. tabulu). Izvēlētie klasterizācijas algoritmi tiks sīkāk aplūkoti nākošas apakšnodaļas ietvaros.

2.3. Klasterizācijas algoritmi un to modifikācijas

Īsu laika rindu analizējamajās datu kopās nav informācijas par klasēm un to saistību ar aplūkojamiem datiem, tāpēc rodas nepieciešamība veikt datu analīzi, citiem vārdiem sakot - atrast sakarības datos, kas palīdzētu noteikt analizējamajos datos līdzīgas objektu grupas — klasterus.



2.2. att. Klasterizācijas pieeju un to populārāko algoritmu iedalījums

Īsu laika rindu salīdzināšanai par piemērotākajiem tiek uzskatīti tie klasterizācijas algoritmi, kas balstās uz līdzības mēriem, jo tie veic divu objektu (laika rindu) savstarpēju salīdzināšanu katrā laika punktā, nosakot attālumu starp tiem.

2.3.1. K-vidējo sadalošais algoritms

Šis algoritms pieder pie sadalošās klasterizācijas algoritmiem. Algoritms uzsāk darbu, saņemot *m* vērtību, kas raksturo klasteru skaitu analizējamajā *x* objektu datu kopā. Katram klasterim tiek piešķirts centroīds jeb smaguma punkts m_i, kur $1 < i \leq m$. Sākotnējie centroīdi tiek noteikti gadījuma veidā. Tālāk visi datu kopas objekti tiek piešķirti klasteriem, pamatojoties uz objektu attālumu līdz centroīdiem. Objekts tiek piešķirts tam klasterim, līdz kuram attālums ir minimālākais. Par attāluma mēru tiek lietots *Eiklīda* attālums, kas nosaka divu objektu līdzību. Pastāv dažādi attālumu mēri, piemēram *Eiklīda*, *Manhetenas*, *Minkovksi* u. c. Prognozēšanas sistēmas izstrādāšanā tiek lietots *Eiklīda* attāluma mērs, jo kā pierādījuši pētījumi [25], tad analizējot dažādas datu kopas *k-vidējo sadalošais algoritms* uzrāda gandrīz līdzīgus rezultātus ar dažādiem attālumu mēriem. *Eiklīda* attālums tiek aprēķināts pēc formulas (2.7):

$$d(x_i, x_j) = \sqrt{\sum_{t=1}^{m} (x_{it} - x_{jt})^2}, \qquad (2.7)$$

kur x_i, x_j — divi punkti *m*-dimensiju telpā;

m — dimensiju (atribūtu) skaits;

d(·) — attāluma funkcija.

Iterācijas beigās, kad visi objekti ir ieguvuši piederību kādam no klasteriem, tiek pārrēķināti katra klastera centroīdi atkarībā no tam piederošajiem objektiem, nosakot vidējo aritmētisko vērtību starp viena klastera objektiem pēc formulas (2.8):

$$m_i = \frac{1}{g_i} \sum x_i, \tag{2.8}$$

kur m_i — i-tā klastera centroīds;

x_i — i-tā klastera objekti m-dimensiju telpā;

g_i — elementu skaits i-jā klasterī.

Dotais process turpinās iteratīvi, līdz objekti pēc kārtējās iterācijas vairs nemaina savu piederību centroīdiem.

K-vidējo sadalošais algoritma galvenais trūkums ir tā atkarība no sākotnējās sadales, jo centroīdi tiek izvēlēti nejauši, tāpēc klasteru sadale var būt neprecīza. Tāpēc lai izvairītos no neprecizitātēm, algoritms tiek pielietots vairākas reizes ar atšķirīgiem sākotnējiem centroīdiem, iegūtie rezultāti tiek salīdzināti savā starpā, lai izvēlētos labāko. Rezultātu novērtēšanai izmanto *summēto kvadrātisko kļūdu*, kuru aprēķina pēc formulas (2.9):

$$E = \sum_{i=1}^{k} \sum_{x \in c_i} d^2 \left(x, m_i \right), \tag{2.9}$$

kur x — dotais daudz dimensiju objekts klasterī c_i;

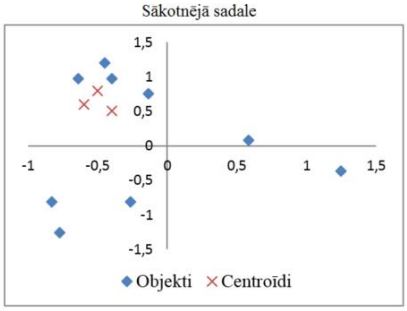
m_i — klastera c_i daudz dimensiju centroīds;

$d(\cdot)$ — attāluma funkcija;

k — klasteru skaits.

K-vidējo sadalošā algoritma realizācijas piemērs ir parādīts 2.3. attēlā ar datu kopu kurā ietilpst 9 objekti, kas sastāv no diviem atribūtiem {T1, T2} ar normalizētam vērtībām, kā arī parādīta objektu sākotnējā sadale divdimensiju telpā un brīvi izvēlētie centroīdi. Parādīti aprēķinātie objektu attālumi (no I līdz IV iterācijai) līdz katram centroīdam un ar treknrakstu izcelts tas objekta attālums, kas atrodas vistuvāk kādam no centroīdiem. Norādīti arī centroīdu vērtību pārrēķini pēc katras iterācijas. Tā kā pēc IV iterācijas objekti vairs nemaina savu piederību centroīdiem, salīdzinot ar III iterāciju, tad algoritms beidz darbu. Šī attēla lejas daļā ir parādīta kopējās summētās kvadrātiskās kļūdas E aprēķins un šīs kļūdas aprēķins katrā no klasteriem.

Objekta numurs	Atribūtu normalizētās vērtības T1	T2
1	-0,136	0,752
2	1,247	-0,365
3	0,584	0,082
4	-0,828	-0,812
5	-0,262	-0,812
6	-0,4	0,976
7	-0,775	-1,259
8	-0,641	0,976
9	-0,451	1,199



Centroīda numurs	Sākotnējo centroīdu vērtības T1	T2
C1	-0,6	0,6
C2	-0,5	0,8
C3	-0,4	0,5

Centroīda numurs	Centroīdu vērtības pēc I iterācijas T1	T2
C1	0,000	0,000
C2	-0,497	1,050
C3	-0,028	-0,402

Centroīda numurs	Centroīdu vērtības pēc II iterācijas T1	T2
C1	0,584	0,082
C2	-0,407	0,976
C3	-0,155	-0,812

Centroīda numurs	Centroīdu vērtības pēc III iterācijas T1	T2
C1	0,916	-0,142
C2	-0,407	0,976
C3	-0,622	-0,961

Objekta numurs	Attālumi līdz centroīdiem pēc I iterācijas C1	C2	C3
1	0,488	0,367	**0,365**
2	2,084	2,100	**1,860**
3	1,292	1,300	**1,069**
4	1,430	1,645	**1,380**
5	1,452	1,629	**1,319**
6	0,426	**0,202**	0,476
7	1,867	2,077	**1,799**
8	0,378	**0,226**	0,534
9	0,617	**0,402**	0,701

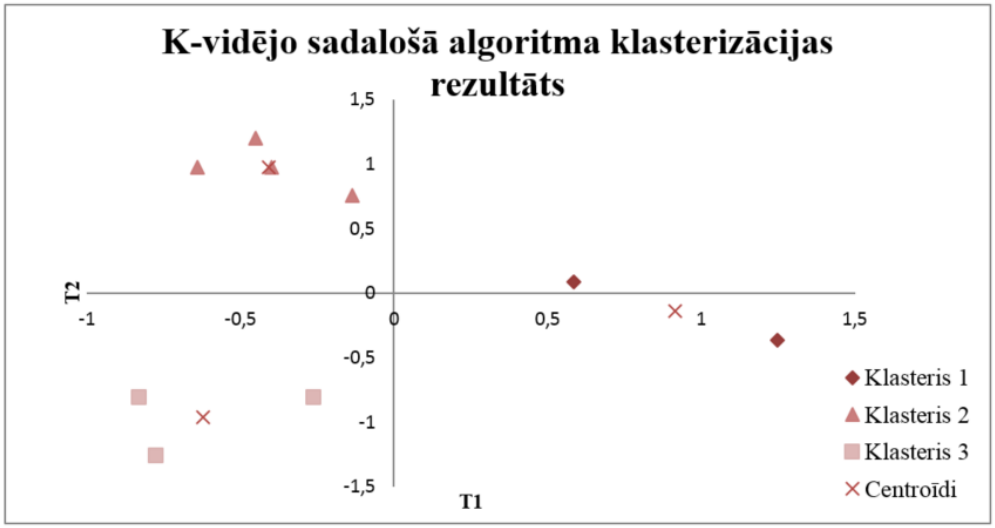
Objekta numurs	Attālumi līdz centroīdiem pēc II iterācijas C1	C2	C3
1	0,764	**0,469**	1,159
2	1,299	2,246	**1,276**
3	**0,590**	1,452	0,781
4	1,160	1,891	**0,898**
5	0,853	1,877	**0,472**
6	1,055	**0,122**	1,428
7	1,478	2,326	**1,136**
8	1,168	**0,162**	1,508
9	1,281	**0,156**	1,656

Objekta numurs	Attālumi līdz centroīdiem pēc III iterācijas C1	C2	C3
1	0,984	**0,351**	1,564
2	**0,800**	2,129	1,471
3	**0,000**	1,334	1,160
4	1,671	1,837	**0,674**
5	1,231	1,794	**0,108**
6	1,329	**0,007**	1,805
7	1,909	2,265	**0,765**
8	1,517	**0,234**	1,853
9	1,523	**0,228**	2,033

Objekta numurs	Attālumi līdz centroīdiem pēc IV iterācijas C1	C2	C3
1	1,380	**0,351**	1,781
2	**0,400**	2,129	1,961
3	**0,400**	1,334	1,594
4	1,868	1,837	**0,255**
5	1,355	1,794	**0,389**
6	1,726	**0,007**	1,950
7	2,026	2,265	**0,335**
8	1,916	**0,234**	1,937
9	1,914	**0,228**	2,167
E= 0,88	0,32	0,23	0,33

2.3. att. *K-vidējo sadalošā* klasterizācijas algoritma darbības piemērs

Attēlā 2.4. parādīts *k-vidējo sadalošā* algoritma klasterizācijas rezultāts ar piemēra datu kopu, kas attēlots divdimensiju telpā.



2.4. att. *K-vidējo sadalošā* klasterizācijas algoritma noteiktā objektu piederība klasteriem

2.3.2. Aglomeratīvais hierarhiskais algoritms

Aglomeratīvais klasterizācijas algoritms pieder pie hierarhiskiem klasterizācijas algoritmiem. Algoritma galvenais darbības princips ir pakāpeniska objektu apvienošana grupās. Sākumā katrs objekts tiek pieņemts kā atsevišķs klasteris. Vispirms apvieno pašus tuvākos divus objektus vai objektu grupas, vadoties pēc minimālā attāluma starp tiem, tad nākamos tuvākos objektus vai objektu grupas. Process turpinās, kamēr visi objekti tiek

apvienoti vienā grupā jeb klasterī, kas tad arī norāda uz algoritma apstāšanās kritēriju. Lai varētu noteikt attālumu jeb tuvumu starp diviem klasteriem un izveidot attālumu matricu, pielieto „tuvākā kaimiņa" principu attāluma noteikšanai starp objektiem pēc formulas (2.10):

$$d_{\min}\left(S_l, S_m\right) = \min_{\substack{x_i \in S_l \\ x_j \in S_m}} d\left(x_i, x_j\right), \tag{2.10}$$

kur $d(\cdot)$ — attāluma funkcija;

S_l, S_m — divi dažādi klasteri;

x_i, x_j — divi dažādu klasteru objekti.

Algoritma formālie izpildes soļi ir šādi:

1) Aprēķināt attālumu matricu.
2) Izmantojot attālumu matricu, atrast divus tuvākos klasterus.
3) Pārrēķināt attālumu matricu, izmantojot „tuvākā kaimiņa" principu.
4) Pārbaudīt nosacījumu: vai ir viens kopīgs klasteris:
   a) ja ir — pāriet uz 5. soli;
   b) ja nav — pāriet uz 2. soli.
5) Beigt algoritma darbu.
6) Ja nepieciešams, uzzīmēt dendrogrammu.

Klasterizācijas rezultātā iegūtā dendrogramma ar objektu apvienojumiem noteiktā attālumā, uzskatāmi parāda objektu apvienošanas secību katrā algoritma iterācijas solī [63, 70]. Par piemērotāko klasteru skaitu tiek uzskatīts lielākais apvienošanas attālums starp objektiem vai objektu grupām (klasteriem), kas dendrogrammā parādīta kā vertikāla līnija (skatīt 3.23. attēlu, kur starp attālumiem 4 un 4,5 novilkta vertikāla līnija).

Aglomeratīvā hierarhiskā algoritma darbības princips ar piemēra palīdzību ir parādīts 2.5. attēlā, bet 2.1. tabulā ir doti datu kopas objektu (klasteru) savstarpējie attālumi. Tabulā ar treknrakstu iezīmēts minimālais attālums (0,799), kuru objekti {X2, X3} pēc I iterācijas tiks apvienoti vienā klasterī. Pēc I iterācijas attālumu matrica tiek samazināta par vienu dimensiju, iegūtos rezultātus (skatīt 2.2. tabulā). Tā kā ir palikuši tikai divi klasteri un ir zināms to apvienošanas attālums (0,983), tad algoritms beidz savu darbu.

2.1. tabula

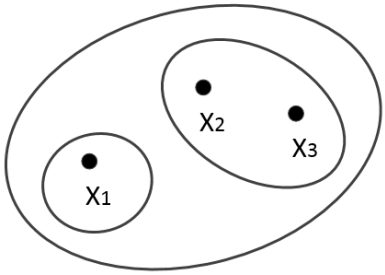
Attālumu matrica

Objektu numuri	**X1**	**X2**	**X3**
X1	X	0,983	1,242
X2		X	**0,799**
X3			X

2.2. tabula

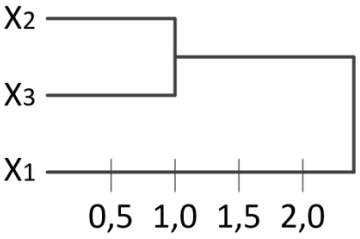
Attālumu matrica pēc I iterācijas

Objektu numuri	**X1**	**X2,3**
X1	X	0,983
X2,3		X



2.5. att. Hierarhiskā datu klasterizācija

Iegūtos klasterizācijas rezultātus var attēlot arī hierarhiskā apvienojumu diagrammā, ko sauc par dendrogrammu, kas parādīta 2.6 attēlā, kur uz *y* ass parādīti klasteru numuri, bet uz *x* ass sadalījuma metrikas (attāluma mēra) mērvienības, pēc kurām tika apvienoti klasteri.



2.6. att. Hierarhiskās datu klasterizācijas dendrogramma

2.3.3. Maksimālās līdzības algoritms

Maksimālās līdzības (*EM*) algoritms [17, 51] pieder uz modeļiem balstītu algoritmu saimei. Šis algoritms izmanto varbūtības, attāluma mēra vietā un analizē sadalījuma līknes katrai no dimensijām, kur objekti atbilst konkrētam klasterim ar noteiktu varbūtību. Šo pieeju dēvē par *nestingro apvienošanu klasteros* (*soft clustering*), jo klasteri var pārklāties un tiem nav izteiktu robežu. Algoritmu var pielietot, kad klasteru robežas nav stingri izteiktas.

Maksimālās līdzības algoritms sastāv no diviem soļiem (*E* un *M*), kuros tiek mēģināts pielāgot funkciju $p(x,z)$ kādai objektu kopai. Iespējamība šai funkcijai tiek aprēķināta pret parametru θ pēc formulas (2.11):

$$l(\theta) = \sum_{i=1}^{m} \log \sum_{z} p(x, z; \theta), \tag{2.11}$$

kur x — datu kopas X objekts;

z — latenti gadījuma veida mainīgie;

m — objektu skaits datu kopā.

Analizējamajā datu kopā $X=\{x_i\}$ parametru $\theta=\{\theta_k\}$ maksimālās līdzības novērtējums tiek noteikts, maksimizējot maksimālas līdzības logaritmisko vērtību, kas pirmo reizi tika reprezentēta A. Dempstera u. c. autoru darbā [18].

E solī notiek apakšējās robežas konstruēšana, bet *M* solī — šīs robežas optimizēšana. *E* solī katram *i* tiek uzstādīts Q_i, pēc formulas (2.12), lietojot Baijesa varbūtības:

$$Q_i(z_i) := p(z_i \mid x_i; \theta), \tag{2.12}$$

kur Q_i — z vērtību sadalījums.

M solī tiek maksimizēta funkcijas *p* robeža, nosakot labākos parametrus pēc formulas (2.13):

$$\theta := \arg\max_{\theta} \sum_{i} \sum_{z_i} Q_i(z_i) \log \frac{p(x_i, z_i; \theta)}{Q_i(z_i)}, \tag{2.13}$$

Neskatoties uz to, ka šo algoritmu iespējams lietot uzdevumos ar pārklājošiem klasteriem, tomēr pastāv negatīvi faktori, kas var ietekmēt algoritma lietošanu: pārāk liels parametru daudzums, no kuru inicializācijas ir stipri atkarīgs iegūstamais rezultāts; nav izmantojams, ja objektu skaits klasteros ir neliels, jo nevar novērtēt kovariācijas matricu.

2.3.4. Modificēts k-vidējo sadalošais algoritms

Modificētais k-vidējo sadalošais algoritms tika izstrādāts, balstoties uz eksperimentu rezultātiem ar īsām laika rindām [30, 35] dažādās problēmvidēs, kas pierādīja, ka, piemēram, klasiskais *k-vidējo sadalošais* algoritms nespēj noteikt piemērotāko klasteru skaitu, analizējot īsas laika rindas [34, 37]. *K-vidējo sadalošā* algoritma iegūtie rezultāti norādīja uz vienotu tendenci — palielinoties klasteru skaitam, samazinās kopējā klasterizācijas kļūda, bet, piemēram, *maksimālās līdzības* algoritms tieši pretēji mazāko kļūdu noteica pie minimālā klasteru skaita. Tāpēc tika izstrādāta *k-vidējo sadalošā algoritma* modifikācija, kas spēj novērst augstāk minētās nepilnības.

Sākotnēji, tiek ieviests jēdziens — maksimālais klasteru skaits, ar kādu tiks klasterizēta analizējamā datu kopa. Šāda pieeja nodrošina algoritma ātrdarbības palielināšanos atšķirībā no klasiskā *k-vidējo sadalošā* algoritma, kur manuāli tiek noteikts klasteru skaits līdz kuram algoritms veic objektu sadalīšanu [73]. Veicot klasterizāciju ar kādu no algoritmiem, jānosaka meklējamo klasteru diapazons (no minimālā līdz maksimālajam skaitam, kur minimālais skaits parasti ir 2), lai klasterizācijas process būtu efektīvs un neaizņemtu daudz laika resursu. Jāatrod piemērotākais klasteru skaits diapazonā no 2 līdz maksimālajam klasteru skaitam. Šim maksimumam jābūt pietiekami lielam, lai precīzi (iespējams noteikt pēc klasterizācijas vidējās absolūtās kļūdas vai kvadrātiskās kļūdas) veiktu klasterizāciju datu kopā, bet tas nevar būt arī pārlieku liels. Šādā gadījumā pastāv iespēja nepareizai rezultātu interpretācijai, jo, piemēram, izveidotu klasteri algoritms cenšas sadalīt vēl mazākos klasteros. Tāpēc maksimālā klasteru skaita C_{max} [64, 70] aprēķinam izmanto teorētisko pieņēmumu, kas tiek aprēķināts pēc formulas (2.14):

$$C_{\max} = \sqrt{n}, \tag{2.14}$$

kur n — objektu skaits analizējamajā datu kopā.

Turpinājumā *modificētais k-vidējo sadalošais* algoritms lieto 2.3.1. punktā aprakstīto metodiku, līdz algoritms sasniedz summētās kvadrātiskās kļūdas aprēķinu, tad notiek algoritma modifikācija. Katram klasterim tiek iegūta attālumu matrica, kura raksturo objektu (laika rindas {T1, T2, …, T12}) c_n attālumus d_n (iegūts, izmantojot *Eiklīda* attāluma mēru) līdz tuvākajam centroīdam (skatīt 2.3. tabulu). No iegūtās attālumu matricas rezultātiem, tiek aprēķināta klasterizācijas kļūda apmācības datu kopai, izmantojot vidējās absolūtās novirzes aprēķinu katrā klasterī un kopējās klasterizācijas kļūdas aprēķinu. Vadoties pēc minimālās

aprēķinātās kopējās klasterizācijas kļūdas vērtības katrā no analizētajiem klasteriem, tiek noteikts piemērotākais klasteru skaits, kas nepieciešams analizējamās datu kopas klasterizācijai.

Ja analizējamās datu kopas objektu skaits nav lielāks par 200, tad klasterizācijas precizitātes novērtējumam jāizmanto *10-kārtu šķērsvalidācija* [43] un klasterizācijas kļūdas aprēķināšana ir jāveic saskaņā ar 2.3.4.1. apakšpunktu, jo, sadalot datu kopu apmācības un testēšanas kopās, objektu skaits tajās būs neliels, kas var iespaidot klasterizācijas rezultātus. Savukārt, ja ierakstu skaits ir lielāks par 200, tad precizitātes novērtējumam var izmantot kopas sadalīšanu apmācības un testēšanas datu kopās ar proporcijas attiecību 70 % pret 30 % [70]. Šajā gadījumā arī ir jāveic klasterizācijas kļūdas aprēķināšana saskaņā ar 2.3.4.1. apakšpunktu un jāizveido paraugmodeļi katram noteiktajam klasterim pēc 2.3.4.2. apakšpunkta. Tad jāizpilda testēšanas datu kopas klasterizācija atbilstoši 2.3.4.3. apakšpunktam, bet klasterizācijas kļūdas aprēķins testēšanas datu kopai tiek veikts saskaņā ar 2.3.4.4. apakšpunktu.

2.3.4.1. Klasterizācijas kļūdas aprēķināšana apmācības datu kopai

Vidējā absolūtā kļūda tiek noteikta summējot katrā klasterī objektu attālumu d_n līdz centroīdam (skatīt 2.3. tabulu) un izdalot iegūto attālumu summu ar objektu skaitu klasterī c_n pēc formulas (2.15), iegūstot vidējās absolūtās novirzes AD_i vērtību:

$$AD_i = \frac{d_1 + d_2 + \ldots + d_n}{c_n}, \tag{2.15}$$

kur d_1, d_2, …, d_n — attiecīgā objekta attālums līdz centroīdam;
c_n — objektu skaits klasterī;
AD_i — vidējā absolūtā novirze i-tajā klasterī.

Pēc tam summē katrā i-tajā klasterī iegūtās vidējās absolūtās novirzes vērtības AD_i un izdala ar noteikto klasteru skaitu C_i analizējamajā datu kopā, iegūstot klasterizācijas vidējo absolūto kļūdu *MeanAE* [52] pēc formulas (2.16):

$$MeanAE = \frac{AD_1 + AD_2 + \ldots + AD_i}{C_i}, \tag{2.16}$$

kur C_i — kopējais noteiktais klasteru skaits datu kopā;
MeanAE — klasterizācijas vidējā absolūtā kļūda.

2.3. tabula

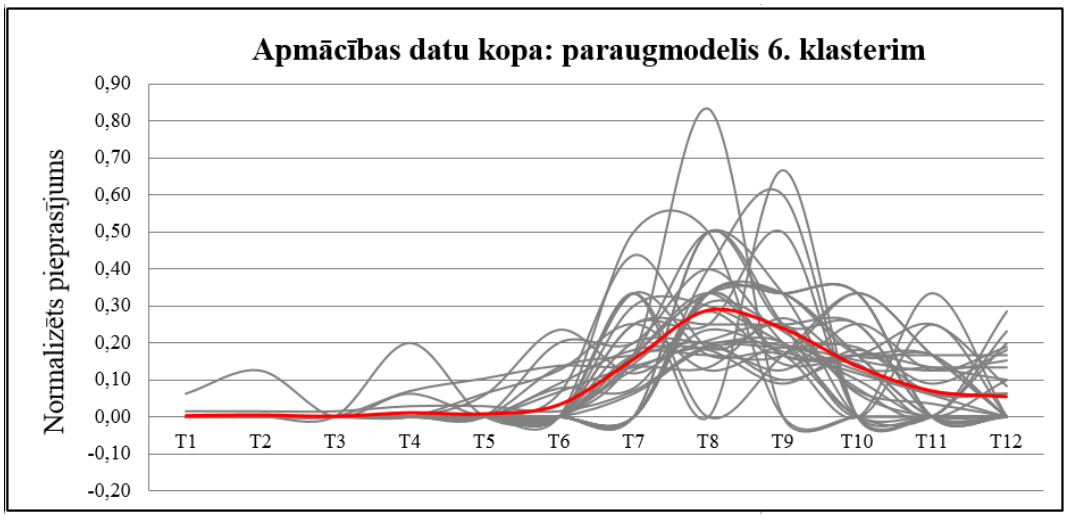
Objektu attālumi līdz centroīdiem klasterizācijas vidējās absolūtās kļūdas aprēķinam

Objekta numurs	Periodi				Attāluma mērs
c	**T1**	**T2**	**…**	**T12**	**d**
c_1					d_1
c_2					d_2
…					…
c_n					d_n

Šāda veida pieeja realizē visu datu kopā ietilpstošo objektu attālumu līdz centroīdiem analīzi katrā no klasteriem, tādejādi nodrošinot klasterizācijas vidējās absolūtās kļūdas aprēķinu, ar kuras palīdzību, pēc mazākās iegūtās *MeanAE* vērtības starp klasteriem C_i nosaka to klasteru skaitu, kas būtu piemērotākais analizējamās datu kopas klasterizācijai.

2.3.4.2. Paraugmodeļi klasteros

Paraugmodelis jeb prototips tiek izveidots katram klasterim, kas noteikts klasterizācijas rezultātā. Paraugmodeļu skaits ir atkarīgs no noteiktā piemērotākā klasteru skaita. Paraugmodeļa iegūtā līkne, kas izveidota no 6. klastera objektu vidējām vērtībām katrā laika periodā, parādīta 2.7. attēlā ar sarkanu līniju, bet ar pelēkām līnijām parādīti šī klastera objekti. Uz *x* ass norādīti periodu numuri, bet *y* ass — īsu laika rindu normalizētās vērtības. Iegūtie paraugmodeļi raksturo klastera objektu uzvedību noteiktā laika periodā.



2.7. att. Izveidotais paraugmodelis (attēlots ar sarkanu līniju) 6. Klasterim

2.3.4.3. Testēšanas datu kopas klasterizācija

Testēšanas datu kopas klasterizācija balstīta uz apmācības kopas klasterizācijā iegūtajām paraugmodeļa vidējām vērtībām un testēšanas kopas (īsas laika rindas) datiem. Objekta attālumu līdz attiecīgā klastera vidējai vērtībai nosaka, aprēķinot katra testēšanas kopas objekta vērtības (skatīt 2.4. tabulu) starpību līdz attiecīgā paraugmodeļa klastera vidējai vērtībai (skatīt 2.5. tabulu), to kāpinot kvadrātā un no kvadrātu summas izvelkot kvadrātsakni pēc formulas (2.17). Iegūtie rezultāti tiek fiksēti attālumu matricā, kura parādīta 2.6. tabulā.

2.4. tabula

Testēšanas datu kopas objektu vērtības pa periodiem

	Periodi			
Objekta numurs	**T1**	**T2**	**…**	**T12**
1	P_1	P_2	…	P_{12}
2	P_{13}	P_{14}	…	P_{24}
…	…	…	…	…
m	P_n	P_{n+1}	…	P_{n+11}

Testēšanas datu kopas klasterizācijā katram analizējamajam objektam tiek noteikta piederība klasterim pēc minimālā attāluma [70]. Tiek ņemts katrs testēšanas kopas objekts pa periodiem un izskaitļots tā attālums līdz katra klastera centroīdam (apmācības kopas paraugmodelim) pēc formulas (2.17):

$$d_{i,j} = \sqrt{\left(P_1 - C_1\right)^2 + \left(P_2 - C_2\right)^2 + \ldots + \left(Pm_n - Cz_n\right)^2}, \qquad (2.17)$$

kur $d_{i,j}$ — testēšanas datu kopas objekta *i* attālums līdz paraugmodelim *j*;

P_m — objekta *m* vērtība testēšanas datu kopā periodā *n*;

C_z — no apmācības datiem iegūtā *z* paraugmodeļa vidējā vērtība periodā *n*.

Kad veikti testēšanas datu kopas objektu attālumu mēru aprēķini, kuru rezultāti atspoguļoti 2.6. tabulā, tad jānosaka katra objekta minimālais attālums katrā rindā $d_{i.}$ un iegūtā vērtība jāreģistrē tabulas laukā *Minimālais attālums*.

2.5. tabula

Apmācības datu kopas klasterizācijas rezultātā
iegūto paraugmodeļu vidējās vērtības katrā laika periodā

Apmācības kopas paraugmodeļa jeb klastera numurs	Periodi			
Nr.	**T1**	**T2**	**…**	**T12**
C1	$C1_1$	$C1_2$	…	$C1_{12}$
C2	$C2_1$	$C2_2$	…	$C2_{12}$
…	…	…	…	…
Cz	Cn_1	Cn_2	…	Cn_{12}

2.6. tabula

Testēšanas datu kopas attālumu matrica klasteru piederības noteikšanai

Objekta numurs	Objekta attālums līdz attiecīgā klastera centram (paraugmodelim)				Minimālais attālums	Piešķirtā klase pēc minimālā attāluma
Nr.	**C1**	**C2**	**…**	**Cj**	**d_i(min)**	**C(j)**
d_1	$d_{1,1}$	$d_{1,2}$	…	$d_{1,j}$		
d_2	$d_{2,1}$	$d_{2,2}$	…	$d_{2,j}$		
…	…	…	…	…		
d_i	$d_{i,1}$	$d_{i,2}$	…	$d_{i,j}$		

Pēc iegūtā minimālā attāluma, kas noteikts katram testēšanas datu kopas objektam, ir jāpiešķir klastera numurs *C(j)*, kuram no *j* klasteriem analizējamais objekts atrodas vistuvāk.

2.3.4.4. Klasterizācijas kļūdas aprēķins testēšanas datu kopai

Lai varētu spriest par testēšanas kopas klasterizācijas rezultātiem, nepieciešams veikt novērtējumu. To var izdarīt, aprēķinot vidējo absolūto novirzi *MAD* (*mean absolute deviation*) un vidējo absolūto kļūdu *MAE* (*mean absolute error*) katrā klasterī. Pēc tam summējot katrā klasterī aprēķinātās vidējās absolūtās kļūdas vērtības. Tādā veidā tiek panākts

novērtējums starp neatkarīgām datu kopām.

MAD [52, 76] aprēķina, summējot pēc moduļa starpību starp apmācības datu kopas klasterizācijas rezultātā iegūtā paraugmodeļa vidējo vērtību laika rindā klasterī *n* un testēšanas datu kopas objekta vērtību *m* un izdalot ar objektu skaitu *N* attiecīgajā klasterī pēc formulas (2.18):

$$MAD = \frac{1}{N}\sum_{i=1}^{N}\left|\left(Cvid_{n_i} - P_{m_i}\right)\right|, \tag{2.18}$$

kur $Cvid_n$ — iegūtā apmācības kopas paraugmodeļa vidējā vērtība klasterī *n* periodā *i*;

P_m — reāla testēšanas datu kopas objekta *m* vērtība periodā *i*;

N — periodu skaits laika rindā.

MAE [52, 76] klasterī *i*, kas norāda uz kopējo kļūdu klasterī attiecībā pret apmācības datu kopu, aprēķina pēc formulas (2.19):

$$MAE_i = \frac{MAD}{k}, \tag{2.19}$$

kur *k* — objektu skaits testēšanas datu kopā klasterī *i*.

Savukārt, kopējo absolūto kļūdu *TAE* (*total absolute error*) [52, 76] testēšanas datu kopai aprēķina, summējot iegūtās vidējās absolūtās kļūdas vērtības klasterī *i* un izdalot apmācības datu kopas klasterizācijas rezultātā noteikto piemērotāko klasteru skaitu *n* pēc formulas (2.20):

$$TAE = \frac{\sum_{i=1}^{n} MAE_i}{n}, \tag{2.20}$$

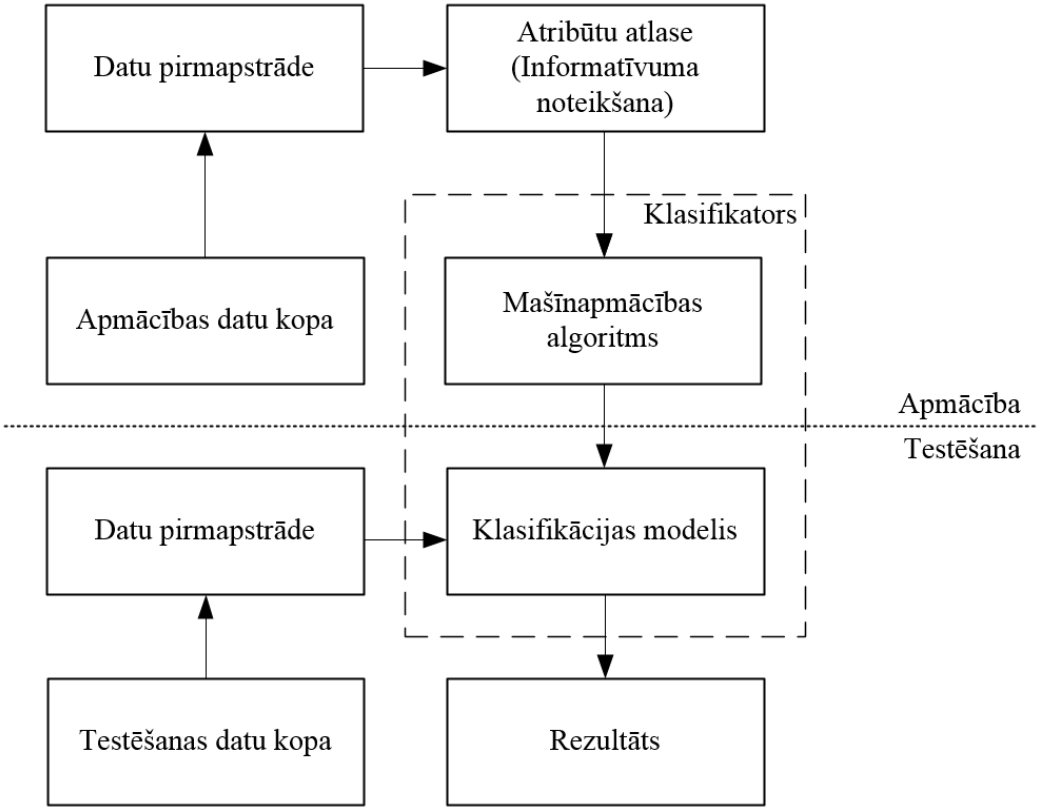
kur MAE_i — vidējā absolūtā kļūda *i-tajā* klasterī;

Iegūstot kopējās absolūtās kļūdas novērtējumu, klasterizējot testēšanas datu kopu un validējot iegūtos rezultātus pret paraugmodeļiem, kuri iegūti klasterizējot apmācības datu kopu, var spriest par klasterizācijas rezultātu precizitāti. Klasterizācijas procesu testēšanas datu kopai, kā arī vidējās absolūtās novirzes, vidējās absolūtās kļūdas un kopējās absolūtās kļūdas aprēķinu piemērus, kas detalizēti aprakstīti, skatīt 1. pielikumā.

2.4. Datu klasifikācija

Datu klasifikācija ir process, kurā tiek izveidots modelis — klasifikators, kas nosaka jaunu novērojumu piederību grupai jeb klasei, modeļa apmācībai izmantojot objektu kopu ar iepriekš zināmu grupu jeb klašu piederību. Klasifikācijas process, kas parādīts 2.8. attēlā, ietver divus posmus:

a) apmācība, kur, lietojot apmācības kopas datus, tiek apmācīts klasifikators;
b) testēšana, kur lietojot testēšanas kopas datus, tiek pārbaudīta klasifikatora precizitāte. Ja klasifikatora precizitāte ir bijusi pietiekama (nosaka sistēmas izstrādātājs vai eksperts), tad šo klasifikatoru iespējams tālāk izmantot jaunu objektu klases prognozēšanai.



2.8. att. Datu klasifikācijas process

Parasti apmācības un testēšanas datu kopās pirms datu analīzes tiek veikta datu pirmapstrāde, piemēram, lai izvairītos no atribūtu vērtību diapazonu atšķirībām. Apmācības datu kopā iespējams veikt arī atribūtu informatīvuma noteikšanu, kas palīdz izvairīties no neinformatīvu atribūtu iekļaušanas datu kopā, kā arī samazina dimensiju skaitu, datu apstrādes laiku un izmaksas.

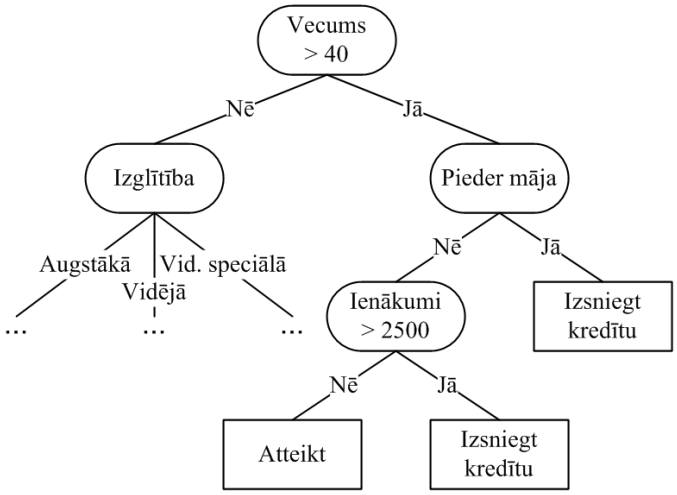
Klasifikācijas uzdevumos, lai noteiktu funkcionālo atkarību starp atribūtiem, var tikt

pielietota viena no pieejām, piemēram:

1) Klasifikācijas likumi — sastāv no divām daļām: nosacījuma un slēdziena, ko apraksta ar JA…TAD … (*IF…THEN …*) likumu. Klasifikācijas likumus sauc arī par nosacījumu likumiem. Nosacījums var sastāvēt no viena vai vairākiem atribūtiem. Vairāku atribūtu iekļaušanu nosacījumā var veikt ar operāciju UN(*AND*), VAI(*OR*) un NE(*NOT*) palīdzību. Slēdziens ir atkarīgā atribūta vērtība vai klašu varbūtības sadalījums. Galvenā šo likumu priekšrocība ir to uztveres vieglums un pieraksts saprotamā valodā. Likumu kopai var viegli pievienot jaunu likumu, bez nepieciešamības koriģēt eksistējošos. Likumu neatkarības relativitāte saistīta ar iespēju, būt pretrunā vienam ar otru.
2) Lēmumu koks — grafs ar vairākām virsotnēm, kas savā starpā ir saistītas ar lokiem. Pastāv trīs veida virsotnes:
   a) sakne — virsotne, kurai nav ieejošu loku, bet viens vai vairāki izejoši loki;
   b) iekšējais mezgls — virsotne, kurai ir viens ieejošs loks un vismaz divi izejoši loki;
   c) koka lapa — virsotne, kurai ir viens ieejošs loks un nav izejošu loku.

   Piemēram, dota datu kopa $D=\{d_1, d_2, \ldots, d_n\}$, kur n ir objektu skaits datu kopā. Datu kopa D satur klašu kopu $C=\{c_1, c_2, \ldots, c_m\}$, kur m ir klašu skaits, kā arī atribūtu kopu $A=\{a_1, a_2, \ldots, a_s\}$, kur s ir atribūtu skaits. Katrs ieraksts datu kopā sastāv no s atribūtu vērtībām un vienas klases mainīgā vērtības $d_i=\{ d_i^1, d_i^2, \ldots, d_i^5,..., c_j\}$, kur i ir objekta numurs un j — klases vērtība. Lēmumu koks nodrošina likumu attēlojumu sakārtotā hierarhiskā struktūrā. Likumi ir skaidrojami ar loģisko «JA … TAD …» konstrukciju. Lēmuma koka piemērs, kredīta izsniegšanas problēmai bankā, ir parādīts 2.9. attēlā. No 2.9. attēlā parādītā lēmuma koka daži iegūtie nosacījumu likumi:
      a) JA "Vecums >40"="Nē" TAD "Izglītība";
      b) JA "Vecums >40"="Jā" TAD "Pieder māja";
      c) JA "Pieder māja"="Jā" TAD "Izsniegt kredītu".

   Lēmumu kokus ir vienkārši lietot, jo tie veido nosacījumu likumus, kas viegli izprotami, kā arī jauna objekta klasifikācijas process ir vizuāli uztverams un saprotams. Jauni objekti tiek klasificēti, virzoties hierarhiski lejup pa koka struktūru. Šī veida process aizņem noteiktu laiku vienību, jo dziļāka un sarežģītāka koka struktūra, jo ilgāks laiks tiek patērēts jaunā objekta klasificēšanai.



2.9. att. Lēmuma koks kredīta izsniegšanas uzdevumam bankā

Izveidotajam lēmumu kokam, kas iegūts klasifikatoru apmācot ar apmācības datu kopu un pēc tam pārbaudīts ar testēšanas datu kopu, piemīt klasifikācijas kļūda, kas rodas no nepareizi klasificētiem un neklasificētiem objektiem. Labiem klasifikācijas rezultātiem ir jābūt ar mazu klasifikācijas kļūdu. Tas ir ļoti svarīgi, jo ja modeļa izmērs ir pārāk liels, tas rada likumu daudzveidību, kuri pārklāj mazāk objektu skaitu un veicina pārapmācību. Pārapmācības iemesli var būt dažādi [1, 3, 58]:

a) nepareiza apjoma apmācības kopas izvēle;
b) trokšņainu datu lietošana;
c) apmācības piemēru trūkums.

3) Matemātiskas funkcijas — metodes, kas vairāk piemērotas nepārtrauktu datu apstrādei, izmanto pieeju funkcijas izvēlei ar mazāko kļūdas līmeni, kuru raksturo izteiksme (2.21):

$$\min_{f \in F} R(f) = \frac{1}{m} \sum_{i=1}^{m} c(y_i, f(x_i)), \qquad (2.21)$$

kur F — visu iespējamo funkciju kopa;
$c(y_i, f(x_i))$ — zaudējumu funkcija (*loss function*);
$f(x_i)$ — atkarīgā atribūta vērtība, kas iegūta ar funkciju f vektoram $x_i \in F$;
y_i — zināmā (precīzā) atribūta vērtība.

Zaudējuma funkcija, parasti, ir pozitīva, jo nav iespējams iegūt "balvu" par ļoti labu prognozi. Gadījumā, ja tomēr zaudējumu funkcijas vērtība ir negatīva, tad to var

labot, ieviešot pozitīvu nobīdi, kas atkarīga no *x*. Pielietojot šo nobīdi iespējams iegūt nulles zaudējumus, pie absolūti precīzas prognozes, kad $f(x)=y$. Šādu zaudējumu funkcijas ierobežojumu priekšrocība ir vienmēr zināms minimums un tas, ka šis minimums vienmēr ir sasniedzams, vismaz pārim *x* un *y* [76].

2.5. Klasifikācijas algoritmi

Pamatojoties uz sākotnēji izvirzītajiem uzdevumiem, kuri nosaka, ka sistēmas iegūtajiem rezultātiem ir jābūt viegli interpretējamiem un saprotamiem, kā arī veikto klasifikatoru salīdzinošo analīzi, tad datu klasifikācijai labāk ir izvēlieties algoritmus, kuri balstīti uz lēmumu kokiem [58], induktīvo secināšanu [14] vai attāluma mēru metriku [12]. Šo algoritmu iegūtie rezultāti ir viegli uztverami un interpretējami, salīdzinot, piemēram, ar mākslīgiem neironu tīkliem [65].

2.5.1. C4.5 algoritms

Induktīvās apmācības algoritms *C4.5* ir veidots uz algoritma *ID3* bāzes ar iespēju apstrādāt diskrētus, nepārtrauktus un trūkstošus datus [58].

Par atribūtu izvēles kritēriju algoritmā *C4.5* tiek lietots *Information Gain* un *GainRatio*. *Information Gain* ļauj aprēķināt entropijas izmaiņas, kas iestātos, ja informācijas stāvoklis izmainītos un kļūtu zināma daļa no šīs informācijas (piemēram, kāda atribūta vērtība). Entropiju *E* datu kopai *D* aprēķina pēc formulas (2.23):

$$E(D) = -\sum_{i=1}^{\gamma} p_i \log_2(p_i), \qquad (2.22)$$

kur γ — klašu skaits;

p_i — varbūtība, ka ieraksts pieder *i*-tajai klasei.

Gadījumā, ja $p_i = 0$, tad pieņem, ka $log_2(p_i)=0$. *Information Gain* dod priekšroku tiem atribūtiem, kuriem ir lielāks vērtību skaits. Lai tas nenotiktu, tiek lietots *GainRatio*, kas atribūtiem ar lielāku skaitu neļauj "dominēt", tos sadalot *Information Gain* ar paša entropiju. *Information Gain* aprēķina pēc formulas (2.23)

$$InformationGain(D,S) = E(D) - \sum_{i=1}^{s} P(D_i)E(D_i), \qquad (2.23)$$

kur *S* — potenciālais sadales atribūts;

s — atribūta vērtību (stāvokļu) skaits;

D_i — datu kopas D daļa, kur atribūtam S ir vērtība i;

$P(Di)$ — varbūtība, ar kādu atribūts D pieder vērtībai i.

GainRatio atrisina iepriekš minēto problēmu, normalizējot entropijas samazinājumu ar datu kopas sadalījuma metriku *SplitInfo* [27, 58], kuru aprēķina pēc formulas (2.24):

$$SplitInfo(S) = -\sum_{i=1}^{n} \frac{|A_i|}{|A|} \log_2 (\frac{|A_i|}{|A|}), \tag{2.24}$$

kur A — datu kopa, kas sadalīta A_i apakškopās pēc atribūta S;

n — atribūtu skaits.

Atribūtiem ar lielu vērtību skaitu *SplitInfo* rādītājs būs lielāks, nekā citiem iespējami informatīvākiem atribūtiem ar mazāku vērtību skaitu, šo nevienlīdzību iespējams kompensēt aprēķinot *GainRatio* [27, 58] pēc formulas (2.25):

$$GainRatio(X) = \frac{InformationGain(S)}{SplitInfo(S)}. \tag{2.25}$$

Algoritms *C4.5* strādā iteratīvi, būvējot lēmuma koku pa līmeņiem. Lai uzsāktu lēmuma koka konstruēšanu, ir jānosaka sadalošais atribūts, aprēķinot ieguvumu *Information Gain,* kas kļūs par sakni. Pirms atribūtu sadales tiek aprēķināta entropija E pilnai apmācības datu kopai D. Pēc tam katram atribūtam atsevišķi, visām iespējamām šī atribūta vērtībām, tiek aprēķināta entropija pēc formulas (2.22), vadoties pēc klases atribūta vērtības relatīvā parādīšanās biežuma apmācības datu kopā. Tad katram atribūtam tiek aprēķināts informācijas ieguvums *Information Gain* pēc formulas (2.23). Par sakni kļūst tas atribūts, kuram *Information Gain* rādītājs ir bijis lielākais. No saknes, iziet tieši tik zaru, cik vērtību ir šim atribūtam, ja tiek lietoti atribūti ar diskrētām vērtībām. Lai izvēlētos vērtību skaitliskam vai nepārtrauktam atribūtam, tiek aplūkoti visi blakus esošie vērtību pāri un aprēķināts *Information Gain*, pēc kura izvēlēts pāris ar augstāko vērtību. Atribūts, kas izvēlēts par sakni, vairs netiks izmantots nākamo lēmuma koka līmeņu konstruēšanā.

Entropijas maksimālā $E(D)_{max}$ vērtība pie divām klasēm ir 1, bet, pieaugot klašu skaitam n, to aprēķina pēc formulas (2.26):

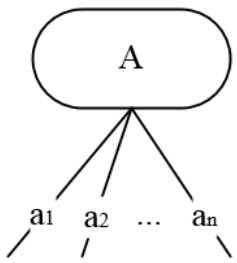
$$E(D)_{\max} = \log_2(n). \tag{2.26}$$

Tālāk atkal apmācības datu kopas ieraksti tiek sadalīti pēc atribūta vērtībām un klasēm.

Ja kādas atribūta vērtības visi ieraksti pieder vienai klasei, tad lēmuma koka zara galā tiek izveidota lapa ar šo klasi, bet ja ieraksti pieder dažādām klasēm, tad turpinās iteratīvs process, meklējot informatīvāko atribūtu nākamajam iekšējam mezglam. Gadījumā, ja lēmuma koka izveidē ir atlicis viens neizmantots atribūts, bet pastāv vēl citas apakškopas, kas satur ierakstus no dažādām klasēm, tad *Gain* vairs nav vērts aprēķināt. Algoritms beidz darbu un zara galā izveido lapu ar dominējošo klasi. Tas nozīmē, ka klase tiek piešķita pēc zarā esošo vērtību raksturojošas klases vairākuma.

Algoritms darbības laikā no sākuma veido maksimālo koku, kas var izraisīt pārapmācību (klasifikators pielāgojas apmācības kopas datiem un uzrāda lielu kļūdas līmeni testēšanas datu kopā darbā ar vēl nezināmiem klasificējamiem objektiem). Lai izvairītos no pārapmācības, parasti tiek lietota atzarošana vai koka veidošanas ierobežojums, kuru izmanto pēc pilna lēmuma koka iegūšanas.

Atzarošanas laikā tiek aplūkoti visi apakškoki, kas ietilpst pilnā lēmuma kokā, un tie tiek aizstāti ar lapu vai apakškoka zaru. Algoritms, savā klasiskajā formā, lieto samazinātās kļūdas atzarošanu ar atsevišķu testēšanas kopu [44]. Šī procesa ietvaros tiek aplūkots katrs apakškoks, vienreiz hierarhiski pārvietojoties no koka augšas uz leju, un novērtējot, vai apakškoka aizvietošana ar lapu dos klasifikācijas kļūdas samazinājumu. Pieņemsim, ka ir dots apakškoks K (skatīt 2.10. attēlu), kas izveidots no datu kopas D un saknē dala atribūtu A pēc tā vērtībām no a_1 līdz a_n. Uzskatīsim, ka K_i jau ir atzarots un, ka a_2 ir biežāk sastopamā atribūta A vērtība, bet L ir lapa ar datu kopas D biežāk sastopamo klasi. Nepareizi klasificēto ierakstu skaits kokam K, apakškokam K_2 un lapai L tiek apzīmēts attiecīgi ar E_K, E_{K2} un E_L. Tad algoritms aplūko visu koku, ņemot vērā šādus kļūdu novērtējumus: $U_{CF}(E_K,|D|)$, $U_{CF}(E_{K2},|D|)$, $U_{CF}(E_L,|D|)$, un kurš no tiem ir mazāks, atstāj apakškoku K bez izmaiņām, aizstāj apakškoku ar tā zaru K_2 vai aizstāj apakškoku ar lapu.



2.10. att. Apakškoka K struktūra

Atzarošanas priekšrocība ir tā, ka katrs apakškoks tiek aplūkots vienreiz un šī pieeja ir ātrāka par citām, bet mīnuss — nepieciešama atsevišķa datu kopa koka atzarošanai [58].

2.5.2. K-tuvāko kaimiņu algoritms

K-tuvāko kaimiņu (*kNN*) algoritms [63] jauna objekta (testēšanas kopa) klasifikācijai izmanto attāluma mēru, lai noteiktu "tuvību" (attālumu) līdz katram objektam apmācības datu kopā, izveidojot attālumu matricu *x*y*, kur *y* ir apmācības kopas objektu skaits, bet *x* — testēšanas kopas objektu skaits. Klasifikācijas rezultātā katram jaunajam objektam no testēšanas kopas tiek noteikti *n* tuvākie kaimiņi, lietojot *Eiklīda* attāluma mēru, kuru aprēķina pēc formulas (2.7). Tuvāko *k* objektu (kaimiņu) skaits tiek aprēķināts pēc formulas $k \le \sqrt{n}$, kur *n* ir objektu skaits apmācības datu kopā. Objekti *k* ar mazāko attālumu veido tuvāko kaimiņu apakškopu L_k. Klases noteikšana tiek veikta pēc vairākuma principa (dominējošā klase starp tuvākiem kaimiņiem), izmantojot formulu (2.27):

$$c' = \arg\max_{\gamma} \sum_{(x_i, c_j \in L_k)} I(\gamma = c_j), \tag{2.27}$$

kur γ — klases iezīme;

L_k — tuvāko kaimiņu apakškopa;

c_j — viena no tuvāko kaimiņu klases iezīmēm;

$I(\cdot)$ — bināra funkcija, ja $\gamma=c$ tad tās arguments ir 1, pretējā gadījumā 0.

Lai mainīgā *k* izvēle neietekmētu klasifikācijas rezultātus, tiek pielietoti svari, ar kuru palīdzību regulē apmācības kopas objektu ietekmi uz klasificējamo objektu. Pielietojot svarus, klases aprēķina formula (2.27) tiek modificēta šādi:

$$c' = \arg\max_{\gamma} \sum_{(x_i, c_j \in L_k)} \frac{1}{d(x'_j, x_i)^2} \cdot I(\gamma = c_j), \tag{2.28}$$

kur $d(x'_j, x_i)$ — attāluma mēra vērtība no klasificējamā ieraksta x'_j līdz kaimiņam x_i.

Pastāv divas heiristikas, kuras var piemērot jauna objekta klasifikācijai, neaprēķinot svarus. Pirmā heiristika nosaka — ja klasifikācijas rezultātā visiem tuvākajiem kaimiņiem ir iegūta viena un tā pati klase, tad klasificējamais objekts iegūst šo tuvāko kaimiņu klasi. Savukārt, otrā heiristika nosaka — ja, atmetot tālāko tuvāko kaimiņu (piemēram, ja ir 3 tuvākie kaimiņi, tad atmet to kaimiņu, kura attālums līdz klasificējamajam objektam ir pats lielākais), pārējie ieraksti pieder vienai un tai pašai klasei, tad klasificējamais objekts iegūst šo atlikušo (piemēra gadījumā — divu) tuvāko kaimiņu klasi.

2.5.3. CN2 algoritms

Sākotnēji *CN2* algoritms [14] katrā iterācijā meklē nosacījumu, kas pārklātu lielu objektu skaitu no vienas klases C_i un dažus objektus no citām klasēm C_j, kur $C_i \neq C_j$. Kā nosacījums kalpo konjukcija, kas veidota no atribūtu vērtībām, piemēram, *Dzimums*=1 UN *Slikta dūša*=1 UN *Atraugas*=0 UN *Meteorisms* =0 UN *Svars*=73. Kad nosacījums ir atrasts, algoritms no datu kopas izslēdz objektus, kuri pārklāj atrasto nosacījumu un pievieno likumu: JA <nosacījums> TAD <iegūtā klase> saraksta beigās. Process turpinās, līdz nav iespējams atrast nevienu likumam atbilstošu nosacījumu. Tad algoritms izvērtē atrasto nosacījumu, lietojot novērtējuma funkciju. Viena no novērtējuma funkcijām — entropija *E* — jo mazāka entropijas vērtība, jo piemērotāks būs nosacījums. *Entropija* tiek aprēķināta pēc formulas (2.22). Otra novērtējuma funkcija *Ratio* pārbauda atrastā nosacījuma nozīmīgumu, kas norāda, ka atrastajam likumam piemīt atkārtojamība un tas nav ticis atrasts nejauši. Šī funkcija norāda uz korelāciju starp atribūtu vērtībām un klasi. Lai novērtētu *Ratio*, algoritms *CN2* salīdzina aplūkoto sadalījumu starp klašu objektiem, kas atbilst nosacījumam ar gaidāmo sadalījumu, kas varētu tikt iegūts, ja nosacījums atlasītu objektus nejaušā veidā. Nozīmīgumu *Ratio* aprēķina, izmantojot piemērotības koeficienta [14] statistiku pēc formulas (2.29):

$$Ratio = 2\sum_{i=1}^{v} f_i \log_2(\frac{f_i}{g_i}), \qquad (2.29)$$

kur f — atpazīto objektu klašu biežuma sadalījums, kas atbilst atrastajiem likumiem; (ar likumu pārklāto objektu skaits starp klasēm);

v — klašu skaits;

g — gaidāmais sadalījums atpazīto objektu kopā, kas atlasīti nejaušā veidā no kopējā objektu skaita.

2.5.4. Naivais Baijesa algoritms

Naivais Baijesa (*Naive Bayes*) algoritms [63] lieto Baijesa teorēmu un veic klasifikāciju, izmantojot aprioras un nosacītas (aposterioras) varbūtības, kuras tiek aprēķinātas, analizējot apmācības datu kopas objektus. Tiek pieņemts, ka klasēm ir nenoteikta saistība ar atribūtiem.

Katram objektam ir dota atribūtu kopa $A=\{a_1, \ldots, a_i\}$ un objekti iedalāmi $C=\{c_1, \ldots, c_j\}$ klasēs. Apmācības laikā klasifikators katrai atribūta a_i vērtībai aprēķina aprioro varbūtību, kas norāda uz to, ka objekts ar šādu dotā atribūta vērtību pieder klasei c_j, kas tiek aprēķināta pēc formulas (2.30):

$$P\left(a_i \mid c_j\right) = \frac{n_c}{n}, \tag{2.30}$$

Kur n — objektu skaits apmācības datu kopā;

n_c — objektu skaits apmācības datu kopā, kas pieder klasei c_j.

Savukārt, ja datu kopā pastāv atribūti ar trūkstošām vērtībām un kādam atribūtam klases nosacītā varbūtība ir nulle, tad kopējā $P(C|A)$ varbūtība klasei arī būs nulle. Lietojot vienkāršo dalīšanu šādai klases nosacītās varbūtības noteikšanas metodei, rezultāts var būt neprecīzs, sevišķi, ja ir daži apmācības piemēri un liels skaits atribūtu. Vissliktākais gadījums ir tad, ja apmācības piemēri neietver kādu atribūta vērtību. Šādā gadījumā nav iespējams klasificēt jauno objektu, jo vairākām varbūtībām $P(C|A)$ vērtība var būt nulle.

Trūkstošo datu apstrādes problēmu iespējams atrisināt, aprēķinot nosacītās varbūtības katra atribūta a_i iespējamai vērtībai ar klasi c_j, lietojot m vērtēšanas metodi pēc formulas (2.31):

$$P\left(a_i \mid c_j\right) = \frac{n_j + mp}{n_c + m}, \tag{2.31}$$

kur p — apriora varbūtība, ar kādu atribūta vērtība pieder atribūta a_i;

n_j — objektu skaits ar atribūta a_i un klasi c_j;

m — konstante.

Lai apstrādātu nepārtrauktus datus, to vērtības ir jādiskretizē, vai ir jāpielieto kādu no zināmiem varbūtības sadalījumiem.

Naivais Baijesa algoritms izvēlas visiespējamāko objekta klasifikāciju C_{NB}, pie definētiem atribūtiem a_i, nosakot varbūtīgāko klasi pēc formulas (2.32):

$$C_{NB} = \arg\max_{c_j \in C} P\left(c_j\right) \prod_{i=1}^{m} P\left(a_i \mid c_j\right). \tag{2.32}$$

2.5.5. Klasifikācijas precizitātes novērtēšana

Iegūtos klasifikatoru rezultātus nepieciešams izvērtēt, lai varētu spriest par pielietotā algoritma lietderību uzdevuma risināšanā. Lai salīdzinātu algoritmu rezultātus, ir nepieciešama skaitliska metrika, ar kuras palīdzību varētu novērtēt klasifikācijas precizitāti. Parasti, klasifikācijas novērtēšanai izmanto precizitāti. Lai noteiktu precizitāti divu klašu uzdevumam (piemēram, medicīnā klase "C1" raksturo slimos pacientus, bet klase "C0" — veselos) izveido neskaidrības matricu, kā parādīts 2.7. tabulā, kur zaļā krāsā ir iekrāsoti

pareizi klasificētie objekti, bet sarkanā — kļūdaini klasificētie objekti. ĪP klases "C1" objekti, kas tika klasificēti kā "C1" klases objekti. ĪN klases "C0" objekti, kas tika klasificēti kā "C0" klases objekti. KP klases "C1" objekti, kas tika klasificēti kā "C0" klases objekti. KN klases "C0" objekti, kas tika klasificēti kā "C1" klases objekti.

2.7. tabula

Neskaidrības matrica

Prognozētā klase	Īstā klase	
	Pozitīvie	**Negatīvie**
Pozitīvie	ĪP (Īstie pozitīvie)	KP (Kļūdaini pozitīvie)
Negatīvie	KN (Kļūdaini negatīvie)	ĪN (Īstie negatīvie)

Ja datu kopā ir vairāk par divām klasēm, tad precizitātes mērīšanai izmanto informāciju no neskaidrības matricas, kuras izmērus S x S, nosaka pēc klašu S skaita. Izmantojot neskaidrības matricu, var aprēķināt kopējo klasifikatora precizitāti (*accuracy*) *KKP*, to var izteikt kā attiecību starp pareizi klasificētajiem visu klašu objektiem un kopējo objektu skaitu, lietojot 2.7. tabulā izmantotos apzīmējumus, pēc formulas (2.33):

$$KKP = \frac{\bar{I}P + \bar{I}N}{\bar{I}P + KP + \bar{I}N + KN}. \qquad (2.33)$$

Kopējo klasifikatora kļūdu *KKK* aprēķina, kļūdaini klasificēto objektu skaitu dalot ar kopējo objektu skaitu pēc formulas (2.34):

$$KKK = \frac{KP + KN}{\bar{I}P + KP + \bar{I}N + KN} = 1 - KKP. \qquad (2.34)$$

Tomēr, šādai metrikai piemīt viena īpašība, kas visiem objektiem datu kopā piešķir vienādus "svarus", un kura pie nevienmērīga klašu sadalījuma var iespaidot klasifikatora rezultātus. Īpaši svarīgi tas ir medicīnas vai farmakoloģijas uzdevumu risināšanā, lai sabalansētu klasifikatora spēju atpazīt gan pozitīvo, gan negatīvo klasi (piemēram, medicīnā divu klašu uzdevumā, kur viena klase ir veselie, otra — slimie), tāpēc tiek lietota paplašinātā neskaidrības matrica, skatīt 2.8. tabulu, kurā tiek doti papildus skaidrojumi par kolonnu un rindu summu raksturojošiem lielumiem [74].

2.8. tabula

Paplašinātā neskaidrības matrica

Diagnostikas testa rezultāts	Nosacījums (piemēram, slimības diagnozes rezultāts)		
	Pozitīvie	**Negatīvie**	**Rinda kopā**
Pozitīvie	ĪP	KP	ĪP+KP (Kopējais objektu skaits ar pozitīvu testu)
Negatīvie	KN	ĪN	KN+ĪN (Kopējais objektu skaits ar negatīvu testu)
Kolonna kopā	ĪP+KN (Kopējais objektu skaits, kas atbilst nosacījumam)	KP+ĪN (Kopējais objektu skaits, kas neatbilst nosacījumam)	K=ĪP+ĪN+KP+KN (Kopējais objektu skaits)

No neskaidrības matricas raksturlielumiem, iespējams noteikt to kā klasifikators ir atpazinis pozitīvo klasi, kuru raksturo jutīgums (*sensitivity*), kas izteikts ar attiecību starp precīzi klasificētajiem pozitīvās klases objektiem un kopējo pozitīvās klases objektu skaitu pēc formulas (2.35):

$$Jutīgums = \frac{\bar{I}P}{\bar{I}P + KN}. \qquad (2.35)$$

Kā klasifikators ir atpazinis negatīvo klasi, kuru raksturo specifiskums (*specificity*), kas izteikts ar attiecību starp precīzi klasificētiem negatīvās klases objektiem un kopējo negatīvās klases objektu skaitu pēc formulas (2.36):

$$Specifiskums = \frac{\bar{I}N}{\bar{I}N + KP}. \qquad (2.36)$$

Vienlaicīgi ar precizitātes metrikas izvēli, jāpielieto kāda no datu sadalīšanas pieejām:

a) datu kopas sadalīšana apmācības un testēšanas kopās, kur 70 % ierakstu tiek atvēlēti *apmācības* kopai, bet — 30 % *testēšanas* kopai [70]. Apmācības kopu izmanto klasifikatora izveidošanai, bet testēšana kopu — izveidotā klasifikatora pārbaudei, lai novērtētu klasifikatora precizitāti. Pieeja ir efektīva, ja objektu skaits ir pietiekami liels un testēšanas kopā ir pārstāvētas visas klases, atribūti un to iespējamās vērtības. Ja šie nosacījumi neizpildās, tad būs gadījumi, kad klasifikators nevarēs prognozēt iespējamo klasi;

b) datu kopas sadalīšana, lietojot šķērsvalidāciju, kur tiek izmantota *u kārtu šķērsvalidācija* (*u* fold cross-validation) vai *Izlaist vienu* (leave-one-out) pieeja. Šķērsvalidācijā sākotnējā datu kopa tiek sadalīta *u* apakškopās un klasifikators tiek apmācīts un testēts *u* reizes. Šķērsvalidācija garantē, ka objekta piederība kādai no apakškopām tiek noteikta nejaušā veidā. Katrs ieraksts *u–1* reizes tiks iekļauts apmācības datu kopā un tieši vienu reizi testēšanas datu kopā. Klasifikatora katras iterācijas testēšanas rezultāti ir jāreģistrē atsevišķā neskaidrības matricā, pēc tam šie rezultāti tiek summēti [43]. „Izlaist vienu" pieeja nodrošina, ka testēšanas kopā katrā kārtā tiek izmantots tikai viens datu kopas ieraksts, pārējie — apmācības datu kopā. Process turpinās, kamēr katrs datu kopas ieraksts ir pabijis testēšanas kopā tieši vienu reizi. Pieeja ir efektīva pie neliela ierakstu skaita, jo palielina apmācības datu kopas apjomu. Pastāv arī trūkumi: katrā testēšanas datu kopā pārstāvēta tikai viena klase; lēna darbība, jo tiek veiktas *u* iterācijas [1].

Abas pieejas garantē novērtējuma drošumu.

Klasifikācijas precizitātes salīdzināšanai bieži izmanto arī *vidējās absolūtās kļūdas* un *vidējās kvadrātiskās kļūdas* aprēķinu, jo dažādas kļūdu aprēķinu tehnikas nodrošina precīzāku iegūto rezultātu skaidrojumu un novērtējumu [1, 52].

Vidējo kvadrātisko kļūdu *RMSE* [10], aprēķina starp klasifikatora klasificēto un faktisko klasi pēc formulas (2.37), kura reducēta vairāku klašu gadījumam:

$$RMSE = \sqrt{\frac{1}{B}\sum_{j=1}^{B}\left(\frac{1}{C}\sum_{i=1}^{C}\left(N_j(C_i) - N_j^{'}(C_i)\right)^2\right)}, \qquad (2.37)$$

kur $N_j(C_i)$ — klasificētā klase;

$N_j^{'}(C_i)$ — faktiskā klase;

B — objektu skaits;

C — klašu skaits.

Šo kļūdas novērtēšanas pieeju parasti izmanto tās vienkāršuma dēļ, bet dažos rakstos tā tiek kritizēta [8, 10]. Vidējās kvadrātiskās kļūdas aprēķinu pielieto, nosakot ticamības intervālus [52].

Vidējo absolūto kļūdu *MAE* procentos [52, 70] aprēķina starp klasifikatora klasificētām un faktiskām klasēm pēc formulas (2.38):

$$MAE = \frac{1}{B}\sum_{j=1}^{B}\left(\frac{1}{C}\sum_{i=1}^{C}\left|N_j(C_i) - N_j^{'}(C_i)\right|\right). \qquad (2.38)$$

Vidējā absolūtā kļūda ir mazāk jutīga pret lielām kļūdām, salīdzinot ar vidējo kvadrātisko kļūdu [64].

2.5.6. Jauna objekta prognozēšana

Jauna objekta prognozēšana tiek veikta, pamatojoties uz klasifikācijas rezultātiem, kas iegūti apmācot izmantoto klasifikatoru. Prognozēšana vai arī pareģošana (nav īsti piemērots termins procesa norisei) ir jaunu piemēru pārbaude izveidotajā klasifikatorā. Klasifikatorā tiek padoti jaunā (prognozējamā) objekta raksturojošie parametri (atribūti) ar noteiktām vērtībām, kā rezultātā klasifikators izdara prognozi, nosakot objekta iespējamo klasi. Lai pēc iespējas vieglāk būtu interpretējami un uzskatāmi iegūtie rezultāti, ieteicams izmantot klasifikatorus, kuri nodrošina saprotamu iegūto rezultātu interpretāciju, kā piemēram, induktīvie lēmumu koki, klasifikatori, kas izmanto varbūtības vai attālumu mērus.

2.6. Kopsavilkums un secinājumi

Nodaļā dots ieskats par sistēmas izstrādāšanā lietotajām datu iegūšanas pieejām un algoritmiem. Izklāstīta darba autora izstrādāta *k-vidējo sadaloš*ā algoritma modifikācija, kas piedāvā veikt īsu laika rindu klasterizāciju, nosakot piemērotāko klasteru skaitu.

Pirms datu analīzes procesa ar datu iegūšanas metodēm un algoritmiem ir jāveic datu pirmapstrāde, kurā notiek analizējamo datu kopu sagatavošana. Datu pirmapstrāde nodrošina analizējamo datu kopu sagatavošanu kvalitatīvākam datu analīzes procesam. Kvalitātes uzlabojumus var izmērīt, nosakot klasifikatora precizitāti, piemēram, lietojot neapstrādātu un pirmapstrādātu datu kopu. Īsu laika rindu sagatavošanas procesā ir jāveic periodu izlīdzināšana, lai varētu veikt salīdzināšanu un trūkstošu vērtību atmešanu. Datu pirmapstrādes procesā laika rindās jāveic dominējošu vērtību nolīdzināšana, lietojot *z-novērtējuma normalizāciju ar standarta novirzi* un *normalizāciju ar dzīves līkni*. Raksturojošo parametru informatīvuma noteikšanai tika lietota *CfsSubsetEval* metode, bet atribūtu pārmeklēšanai *BestFirst* metode. Datu normalizācijai tika lietota *z-novērtējuma normalizācija ar standarta novirzi*. Šīs pieejas izraudzītas, jo tās ir populārākās datu pirmapstrādē.

Klasterizācijas uzdevuma risināšanai tika izmantoti vairāki algoritmi, tādi kā *k-vidējo sadalošais*, *maksimālās līdzības*, *aglomeratīvais hierarhiskais* un *modificēts k-vidējo sadalošais* algoritmi, lai eksperimentāli noteiktu piemērotāko klasterizācijas algoritmu izmantošanai prognozēšanas sistēmu uzbūvē. Klasterizācijas precizitātes novērtēšanai tika ieteikts noteikt piemērotāko klasteru skaitu, vadoties pēc vidējās absolūtās un klasterizācijas vidējās absolūtās kļūdas. Izmantoto klasterizācijas algoritmu izvēle tika noteikta pēc klasterizācijas algoritmu salīdzinošās analīzes rezultātiem.

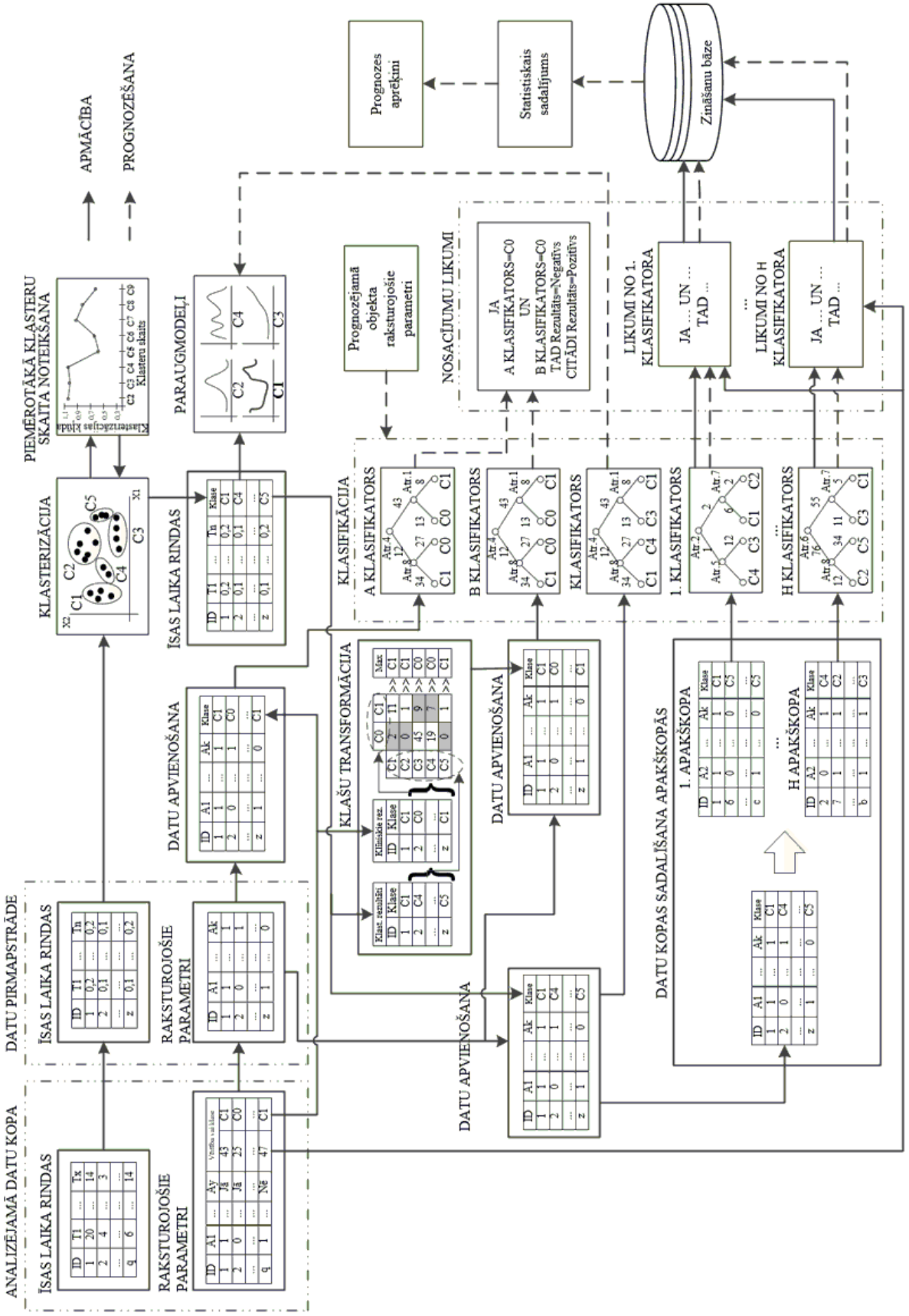
Klasifikācijas uzdevumu atrisināšanai tika izvēlēti vairāki algoritmi, tādi kā *CN2*, *C4.5*, *Naivais Bajesa* un *k-tuvāko kaimiņu* algoritmi, lai eksperimentāli noteiktu piemērotāko klasifikācijas algoritmu izmantošanai prognozēšanas sistēmu uzbūvē. Klasifikācijas precizitātes noteikšanai tika lietoti neskaidrības matricas raksturlielumi: precizitāte, jutīgums un specifiskums. Iegūto rezultātu validācijai tika lietota *10-kartu šķērsvalidācija*, *atmest vienu* vai *datu kopas sadalīšana apmācības un testēšanas datu kopās ar attiecību 70:30* pieejas. Pielietoto klasifikācijas algoritmu izvēle notiek pēc klasifikācijas algoritmu salīdzinošās analīzes rezultātiem.

"Jaunā" objekta klases prognozēšana tiek balstīta uz noteiktā piemērotākā klasifikatora bāzes, kas apmācīts ar apmācības datu kopu un pārbaudīts ar testēšanas datu kopu.

3. APSTRĀDES SISTĒMAS PROGNOZĒŠANAS UZDEVUMIEM

Pēc definētās problēmas, kuru apraksta teorētiskais modelis, kas parādīts 1.7. attēlā, sistēma risinās īsu laika rindu un to raksturojošo parametru apstrādi dažādās problēmvidēs ar datu iegūšanas metodēm un algoritmiem. Piedāvātā prognozēšanas sistēma dažādām problēmvidēm ir parādīta 3.1. attēlā. Analizējamā datu kopa, kas sastāv no īsām laka rindām un to raksturojošajiem parametriem, tiek pakļauta datu pirmapstrādes procesam. Īsās laika rindās tiek noteiktas objektu grupas — klasteri, lietojot darba autora piedāvātu klasterizācijas algoritma modifikāciju. No iegūtajiem klasteriem tiek veidoti paraugmodeļi, kas raksturo klastera objektu vidējās vērtības katrā laika periodā. Iegūtos klasterizācijas rezultātus (objektu klasteru numurus) apvieno ar pirmapstrādātajiem raksturojošajiem parametriem datu apvienošanas blokos. Ja nepieciešams (pēc uzdevuma nosacījumiem), apstrādāt klasterizācijas rezultātus ar dažādu klašu skaitu, tad lieto darba autora piedāvātu klašu transformāciju. Ja nepieciešams apvienoto datu kopu sadalīt apakškopās, tad lieto datu kopas sadalīšanas bloku, kur sadalīšana tiek veikta pēc sadalošā atribūta (nosaka eksperts) vērtību skaita, cik šim atribūtam ir iespējamo vērtību tik apakškopas H tiks izveidotas. Objektu sadalīšana apakškopās tiek veikta pēc sadalošā atribūta vērtības. Izveidotajās datu kopās ar klasifikācijas palīdzību, lietojot klasifikācijas algoritmu, nosaka sakarības starp raksturojošajiem parametriem un klasi. Nosacījumu likumi kalpo divu klasifikatoru rezultātu salīdzināšanai vai zināšanu bāzes izveidošanai, kurā tiek glabāti likumi par katra objekta klasi, datu kopas sadalošā atribūta vērtību un klīnisko pētījumu rezultātu (izteikts kā skaitliska vērtība vai klase).

Prognozēšanas specifika ir atkarīga no izvirzītā uzdevuma. Prognozēšanā ar paraugmodeļi ir jānosaka tā numurs (klase). To nosaka ar apmācīto klasifikatoru, klasificējot analizējamā objekta raksturojošos parametrus. Noteiktais numurs norāda uz paraugmodeli, kas raksturo analizējamā objekta pieprasījumu. Cita prognozēšanas pieeja klasificē analizējamā objekta raksturojošos parametrus ar divu klasifikatoru palīdzību un iegūtos rezultātus salīdzina, lietojot nosacījumu likumus. Nākamā pieeja piedāvā, izveidot pagaidu nosacījuma likumu, kas iegūts, klasificējot analizējamā objekta raksturojošos parametrus nosakot klasi, kā otrs parametrs nosacījumam kalpo sadalošā atribūta vērtība. Pēc tam no zināšanu bāzes atlasa tos likumus, kuri atbilst izveidotajam pagaidu nosacītajam likumam. No atlasītajiem likumiem izveidojas statistiskais sadalījums, no kura ar darba autora piedāvātu pieeju aprēķina prognozes vērtību analizējamajam objektam.



3.1. att. Prognozēšanas sistēma dažādām problēmvidēm

Piedāvātā prognozēšanas sistēma dažādām problēmvidēm kalpo kā teorētiskais modelis, uz kura pamata tika izstrādātas preču pieprasījuma, sirds nekrozes riska un baktēriju proliferācijas sindroma prognozēšanas sistēmas. Šo sistēmu uzbūve sīkāk ir aprakstīta nākošajos apakšpunktos.

3.1. Pieprasījuma prognozēšanas sistēma

Nelielam uzņēmumam, kas nodarbojas ar plaši pazīstamu zīmolu apģērbu mazumtirdzniecību, radās nepieciešamība izveidot pieprasījumu prognozēšanas sistēmu (PPS), ar kuras palīdzību būtu iespējams noteikt jaunā pasūtamā produkta iespējamo pieprasījuma prognozi nākamajam periodam. Citiem vārdiem sakot, veicot jaunas kolekcijas pasūtījumus uz izveidotās sistēmas bāzes, varētu prognozēt vajadzīgo pasūtījuma daudzumu. Šāda prognoze nepieciešama, lai neveidotos lieki uzkrājumi noliktavās un uzņēmums ieekonomētu un optimizētu savus apgrozāmos līdzekļus [13]. Uzņēmuma vēlmes pret izveidojamo sistēmu bija šādas: prognozes veidošanai var izmantot tikai preces raksturojošos parametrus (piemēram, cena, veids, krāsa, izmērs u. c.), iespējas izvēlēties laika periodu (gadu) pēc kura veikt sistēmas apmācību, sistēmas izmantošanas vienkāršība un iegūto rezultātu saprotams atspoguļojums. Balstoties uz uzņēmuma vēlmēm, tiek izvirzītas prasības izveidojamajai sistēmai:

1. Jāapstrādā vēsturiski preču pieprasījumi ar dažādu darbības laika periodu un to raksturojošie parametri.
2. Jāveic raksturojošo parametru analīze un atlase.
3. Jāveic prognozēšana ar vēsturiskiem pieprasījuma datiem un to raksturojošiem parametriem.
4. Jāizveido pēc iespējas vienkāršāka iegūto rezultātu atspoguļošana.

Tradicionālās prognozēšanas sistēmas pamatojas uz likumsakarību meklēšanu laika rindās, piemēram, piegāžu ķēžu analīze [25], tehnisko sistēmu parametru izmaiņa [60] vai ekonomisko indeksu izmaiņa valūtas kursu svārstībās [22] u. c. Tiek pieņemts, ka laika rinda ir stacionāra un, jo garāks ir šo novērojumus skaits, jo lielāka ir varbūtība precīzāk noteikt likumsakarības [8].

Taču reālajā dzīvē pastāv uzdevumi, kuros laika rindas garums ir samērā īss līdz 40 periodiem un noteikt tajā procesa funkcionēšanas likumsakarības praktiski nav iespējams. Par šādiem uzdevumiem tiek uzskatīti: preces dzīves cikla analīze [42], tekstila preču pārdošanas apjomu analīze pie liela piedāvājumu klāsta un īsa dzīves cikla [64, 66, 67] u. c. Šeit izveidojas prognozēšanas uzdevums, kuram jārisina laika rindu analīze ar liela apjoma

salīdzinoši īsām laika rindām, kur katrā aplūkojamā laika periodā izmainās vēsturiskie pieprasījuma dati. Jaunā produkta prognoze ir jāveic, lietojot tikai preces raksturojošos parametrus (cena, tips, krāsa u.tml.). Tāpēc lietderīgi ir lietot klasterizāciju, kas vēsturiskus preču pieprasījuma datus pēc līdzības pazīmēm [68] apvieno grupās vai klasteros [18, 20, 29], uz kuru bāzes izveido pieprasījumu paraugmodeļus. Ar izveidotiem pieprasījuma paraugmodeļiem tiek noteikta saite starp vēsturiskiem preču pieprasījuma datiem un šo preču raksturojošajiem parametriem.

Klasterizācijas rezultātā iegūtie paraugmodeļi, aizvieto reālo objektu pieprasījuma novērtējumu klasterī un tiek apvienoti ar preces raksturojošajiem parametriem. Analizējot iegūtās pieprasījuma paraugmodeļu kopas ar induktīvajiem lēmuma kokiem [58], tiek iegūtas zināšanas nosacījumu likumu veidā, kur likuma nosacījuma daļa raksturo preces pieprasījumu, bet slēdziena daļa — produkta raksturojošos parametrus [70, 76]. Jaunās preces iespējamais pieprasījums (prognoze) tiek noteikts pēc klasterizācijas rezultātā iegūtā paraugmodeļa. Paraugmodeļa numurs (klastera numurs) tiek noteikts klasifikācijas procesā, izmantojot induktīvos lēmuma kokus, projicējot uz tiem jaunās preces raksturojošos parametrus. Izstrādājamajai prognozēšanas sistēmai ir jānodrošina augsta precizitāte, tāpēc tās izveides procesā ir jācenšas samazināt gan klasterizācijas, gan klasifikācijas kļūdas.

3.1.1. PPS uzdevuma nostādne

Pieprasījuma prognozēšanas sistēma uzņēmuma pasūtījumu optimizācijai pamatojas uz iepriekšējo gadu preču pieprasījuma pieredzi un jaunā pasūtamā produkta (preces) raksturojošajiem datiem ar šādiem parametriem: preces cena, tips, sezonalitāte, kolekcijas ilgums (mēnešos), krāsa, izmērs u. c. Lai izveidotu PPS, ir jārealizē šādi uzdevumi:

1. Vēsturisku pieprasījumu datu apkopošana.
2. Atlasīto datu sagatavošana datu iegūšanas procesiem.
3. Sagatavoto datu pirmapstrāde.
4. Piemērotākā klasteru skaita noteikšana vēsturiskās pieprasījumu laika rindās.
5. Iegūto klasterizācijas rezultātu analīze.
6. Klasterizācijas rezultātu apvienošana ar produkta aprakstošajiem datiem, piemēram, preces cena, tips, sezonalitāte, krāsa, izmērs u. c.
7. Klasifikācijas procesu un rezultātu analīze.
8. Izveidotās sistēmas precizitātes pārbaude.
9. Jauna produkta pieprasījuma prognozēšana.
10. Iegūto rezultātu atspoguļošana.

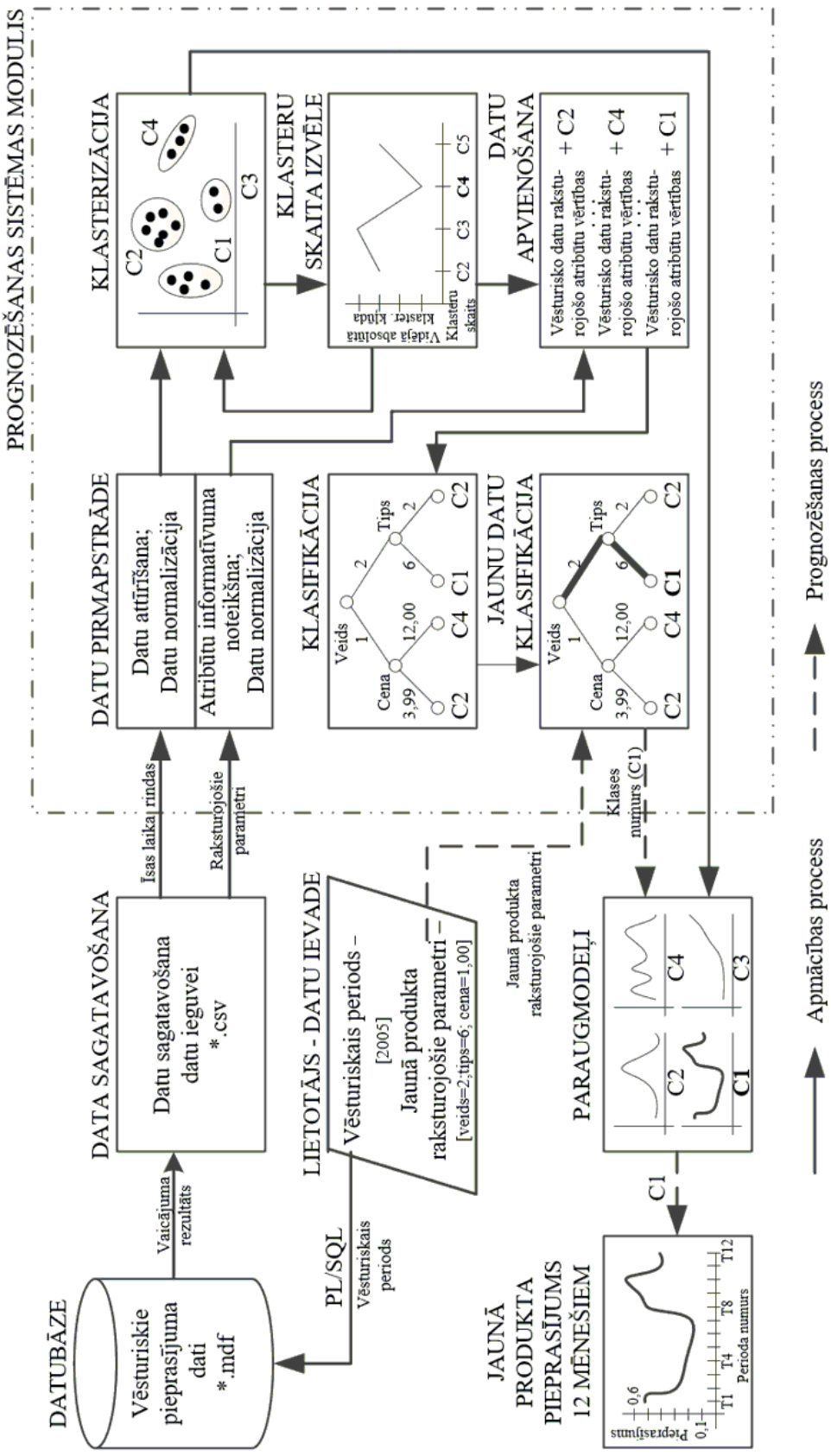
3.1.2. PPS uzbūve

PPS uzbūve ir parādīta 3.2. attēlā. Uzsākot darbu (apmācības process), sistēmai ir jāsaņem vēsturiskais pieprasījumu periods (gads={2006}), kuru ievada lietotājs — eksperts, balstoties uz savu pieredzi. Pēc perioda saņemšanas sistēma izveido vaicājumu (piemēram, SELECT PRECE_2006.ID, PAR_PR_2006.Janvārs, …, PAR_PR_2006_Decembris, PRECE_2006.Veids, PRECE_2006.Tips, PRECE_2006_Cena FROM PAR_PR_2006 LEFT JOIN PRECE_2006 ON PAR_PR_2006.[ID]=PRECE_2006.[ID]), kuru nodod datu bāzei ar valodas *PL/SQL* palīdzību. Datu bāze glabā informāciju par preču pieprasījumu dažādos laika periodos, preču veidiem, izmēriem, krāsām, cenām, pavadzīmēm, piegādātājiem, klientiem, svītrkodiem u. c. Visi vaicājumi sastāv no teikumiem, un šos teikumus ievada operators *SELECT*. Operators *FROM* norāda, no kādas tabulas tiks izgūti dati. Operators *LEFT JOIN* norāda, ar kādu tabulu notiks datu apvienošana, bet operators *ON* — pēc kādiem laukiem notiks tabulu apvienošana. Vaicājuma izpildes rezultātā tiek atlasīta datu kopa, kas tālāk nodota datu sagatavošanas blokam, kurā informācija sadalās divās datu plūsmās. Pirmo veido īsas laika rindas (preces identifikators un šīs preces pieprasījums gada griezumā pa mēnešiem), bet otro — raksturojošie parametri (preces identifikators, veids, tips un cena), kas tālāk nonāk prognozēšanas sistēmas modulī.

Prognozēšanas sistēmas modulī īsas laika rindas tiek pakļautas datu pirmapstrādei, kuras rezultātā dati tiek attīrīti no trokšņainām vērtībām (objekti ar šāda veida vērtībām tiek izslēgti no datu kopas) un normalizēti (izlīdzināti vērtību diapazoni). Ar klasterizācijas algoritmu nosaka sakarības starp īsām laika rindām, kuras pēc līdzības pazīmēm apvieno grupās jeb klasteros. Piemērotāko klasteru skaitu nosaka klasterizācijas algoritms, pēc minimālās klasterizācijas absolūtās kļūdas. Tad katram noteiktajam klasterim tiek izveidots paraugmodelis, kas raksturo klastera objektu vidējās vērtības katrā laika periodā.

Arī otra datu plūsma ir pakļauta datu pirmapstrādei, kuras rezultātā no raksturojošo parametru datu kopas tiek izslēgti objekti ar trūkstošām vērtībām un normalizētas atribūtu vērtības. Pēc tam noteikti informatīvākie atribūti (neinformatīvākie atribūti tiek izslēgti no datu kopas), lietojot *CfsSubsetEval* metodi, bet atribūtu pārmeklēšanai *BestFirst* metodi.

Tālāk vienā datu kopā apvieno pirmapstrādātos raksturojošos parametrus un klastera numuru, kas objektam noteikts klasterizācijas procesā. Apvienošana tiek veikta pēc objektu identifikācijas numuriem. Ar apvienoto datu kopu tiek veikta klasifikācija, lietojot induktīvos lēmuma kokus, rezultātā nosakot sakarības starp paraugmodeli (klastera numuru) un raksturojošajiem parametriem.



3.2. att. Pieprasījuma prognozēšanas sistēma

Pēc apmācības procesa pabeigšanas ir iespējams veikt jaunā produkta pieprasījuma prognozi. Sistēmā tiek ievadīti jaunā produkta raksturojošie parametri (veids={2}, tips={6}, cena={1,00}). Uz klasifikācijas procesā "uzbūvētā" induktīvā lēmuma koka, projicējot tajā (prognozēšanas process) jaunā prognozējamā produkta raksturojošos parametrus, nosaka šī produkta klasi (piemēram, "C1"). Datu projicēšana notiek, virzoties no lēmuma koka saknes uz leju pa koka līmeņiem, līdz tiek sasniegta apakšējā līmeņa koka lapa ar klases numuru. Iegūtās klases numurs (piemēram, "C1") norāda uz attiecīgo paraugmodeli, kas noteikts klasterizācijas procesa rezultātā. Paraugmodelis (C1) norāda kāds būs prognozētā produkta, kura raksturojošie parametri tika ievadīti sistēmā iepriekš, iespējamais pieprasījums 12 mēnešiem (periodiem no T1 līdz T12) nākotnē [40].

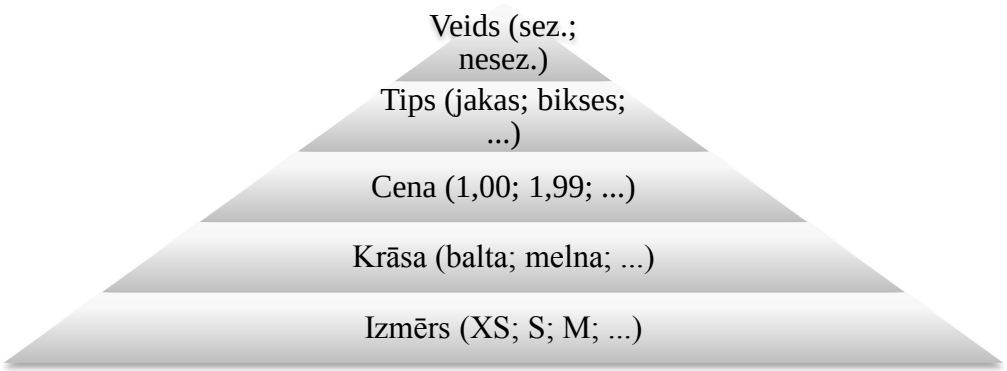
Vēsturisku pieprasījuma un to raksturojošo parametru datu pirmapstrāde

Vēsturisko pieprasījuma datu (īsas laika rindas) pirmapstrāde ir svarīgs datu apstrādes posms, jo no tā būs atkarīga turpmāk iegūto rezultātu precizitāte sistēmā. Vēsturiskos pieprasījuma datus, ja tas ir nepieciešams, attīra no trokšņainām un nepiederošām vērtībām. Piemēram, ja objektam 12 mēnešu periodā ir tikai viens pārdošanas fakts, tad šāds objekts tiek izslēgts no analizējamās datu kopas, jo tas nesatur informāciju, no kuras varētu tikt iegūtas jaunas zināšanas. Lai izvairītos no dominējošām vērtībām īsās laika rindās, tiek pielietota datu normalizācija.

Īsu laika rindu datu normalizācijai izmanto *z-novērtējuma normalizāciju ar standarta novirzi*, kuras darbība aprakstīta 2.1.2.1. apakšpunktā, un *pieprasījumu normalizāciju ar dzīves līkni*, kuru aprēķina pēc formulas (2.4), eksperimentāli nosakot piemērotāko, kuru turpmāk varētu izmantot pieprasījumu prognozēšanas sistēmā.

Pirms uzsākt datu pirmapstrādi raksturojošajos parametros, ir jāveic šo parametru agregācijas stāvokļu noteikšana uzņēmumā kā parādīts 3.3. attēlā, kas tiek saskaņota ar sistēmas pasūtītāju. Protams, pastāv arī alternatīvi risinājumi, piemēram, uzņēmuma pasūtījumu eksperts var strikti nodefinēt parametru agregācijas stāvokļus, pēc kuriem ir jāveido prognozēšanas sistēma, bet šāda veida risinājumi var izrādīties neprecīzi, jo pastāv tā saucamais cilvēka faktors, kas var radīt neviennozīmīgu faktu interpretāciju. Tāpēc tiek veikta raksturojošo parametru informatīvuma noteikšana. Datu pirmapstrādes procesā no raksturojošo parametru datu kopas, tiek izslēgti tie objekti, kuriem ir nepilnīga informācija — iztrūkst kāda atribūta vērtība. Ja nepieciešama dominējošu atribūtu vērtību izlīdzināšana, tiek pielietota datu normalizācija, izmantojot *Z-novērtējuma normalizāciju ar standarta novirzi* (skatīt 2.1.2.1. apakšpunktu). Turpinājumā tiek veikta atribūtu informatīvuma noteikšana [64].

Raksturojošo parametru savstarpējā mijiedarbība ietekmē klasifikatora precizitātes rezultātus jo precīzāk tiek noteikta raksturojošo parametru agregācija vai informatīvums, jo precīzāki ir iegūtie klasifikācijas rezultāti [64]. Tiek uzskatīts, ka piegādes sistēmas efektivitāte uzņēmumā ir tieši atkarīga no komercprognozes rezultāta [23]. Agregācijas stāvokļus uzņēmumā definē eksperts.



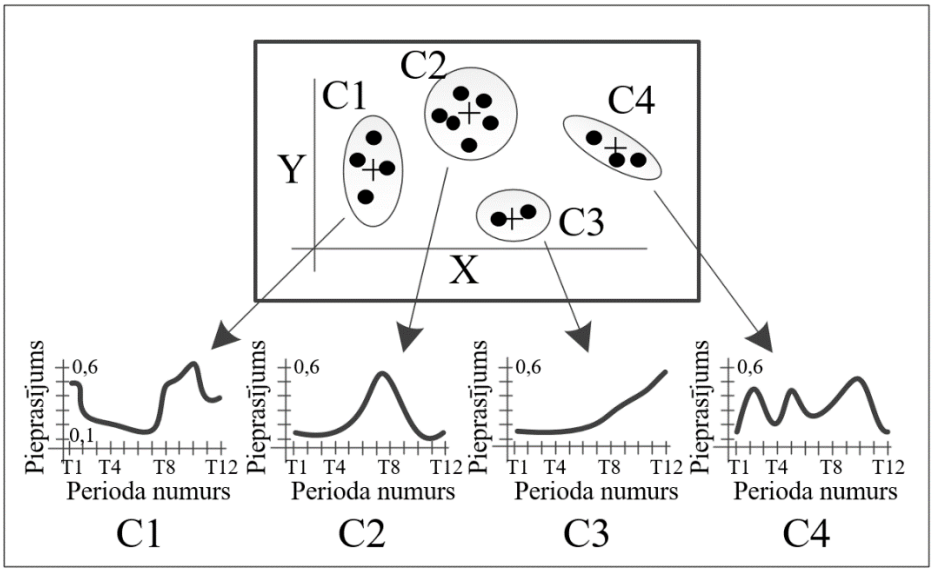
3.3. att. Atribūtu agregācijas stāvokļu noteikšana uzņēmumā

Vēsturisku pieprasījumu klasterizācija un klasteru paraugmodeļu izveide

Pieprasījumu apvienošana klasteros jeb klasterizācija tiek realizēta tā, lai grupā apvienotu tos vēsturiskos pieprasījumus, kuriem ir līdzīga struktūra, lietojot *Eiklīda* attāluma mēru [70]. Šādu apvienojuma tehniku nodrošina *k–vidējo sadalošais* algoritms [63], *maksimālās līdzības* algoritms [51] un darba autora izstrādātais *modificētais k-vidējo sadalošais* algoritms, kurš aprakstīts 2.3.2. apakšpunktā. Ar klasterizācijas palīdzību tiek noteiktas, kuras no preču pieprasījuma datu pirmapstrādes pieejām ir piemērotākās īsu laika rindu analīzei.

Klasterizācijas algoritmu iegūtie rezultāti pie dažādu klasteru skaita ir savstarpēji jāsalīdzina un jāizvēlas piemērotākais algoritms pieprasījuma prognozēšanas sistēmas izstrādāšanā, kā arī jānosaka piemērotākais klasteru skaits, kas nepieciešams datu kopas klasterizācijai. Iespējams, ka lietotā pieeja, piemērotāka klasteru skaita noteikšanai, nav vienīgais risinājums, bet ar izvēlētajiem klasterizācijas algoritmiem tas ir vienīgais ceļš, kā noteikt piemērotāko klasteru skaitu. Kad piemērotākais klasterizācijas algoritms un klasteru skaits ir noteikts, tad katram noteiktajam klasterim tiek izveidots paraugmodelis, kura piemērs parādīts 3.4. attēlā, kas attēlo klastera objektu vidējās pieprasījumu vērtības katrā laika punktā.

Iegūtie paraugmodeļi turpmāk tiek pielietoti jaunā produkta pieprasījuma prognozēšanā. Šie paraugmodeļi raksturos jaunā prognozējamā produkta iespējamo pieprasījumu nākotnē.



3.4. att. Klasterizācijas rezultātā iegūto klasteru paraugmodeļi

Dažādu datu plūsmu apvienošana

Dažādu datu plūsmu apvienošanas procedūra tiek veikta ar objektu raksturojošajiem parametriem (preces tips, cena un veids), pievienojot objekta klasterizācijas rezultātā iegūto klasi pēc to identifikācijas numuriem. Izveidoto datu kopu turpmāk izmanto klasifikatora apmācībai, meklējot sakarības starp klasēm un raksturojošajiem parametriem.

Dažādu datu plūsmu apvienošanas procedūra tiek veikta, apvienojot objekta raksturojošos parametrus (preces tips, cena un veids) un klastera numuru, kas objektam noteikts klasterizācijas procesā. Apvienošana tiek veikta pēc objektu identifikācijas numuriem.

PPS izveidē izmantotie klasifikatori

Klasifikācijas procesā ar klasifikatora palīdzību, izmantojot apmācības kopas datus, tiek noteiktas sakarības starp preces raksturojošajiem parametriem un klasterizācijas procesā iegūto klasi. Lai analizētu izveidotā klasifikatora veiktspēju, nepieciešams klasifikatoru pārbaudīt ar testēšanas kopas datiem un veikt iegūto datu analīzi. Klasifikācijas precizitātes salīdzināšanai izmantoti vairāki klasifikatori, lai eksperimentāli noteiktu piemērotāko PPS:

- *ZeroR* algoritms atrod datu kopā biežāk izmantoto klasi [76];
- *OneR* algoritms veido vienkāršus likumus, veidojot asociācijas, no kurām izskaitļo mazāko klasifikācijas kļūdu [63];

- *JRip* algoritms veido secīgus pārklājuma likumus, kas tiek iegūti secīgi (pa vienam), kur katrs likums dotajai klasei pārklās vairākus šīs klases objektus. Secīgie pārklājuma algoritmi ir visplašāk pielietotie disjunktīvo likuma kopu klasifikācijai [63];
- *Naivais Baijesa* algoritms lieto klases aprioro un nosacīto varbūtību aprēķinu [63];
- *IBk* (k-tuvāko kaimiņu) algoritms jauna ieraksta klasifikācijai izmanto attāluma funkciju [15];
- *J48* (algoritma nosaukums datu analīzes rīkā *Weka*) induktīvais lēmumu koku algoritms (*C4.5*) [58].

Precizitātes novērtēšanai klasifikatora apmācībai tiek lietota apmācības datu kopa, bet klasifikatora pārbaudei testēšanas datu kopa. Izmantoto datu kopu apraksts ir dots zemāk.

3.1.3. PPS eksperimentu norise un iegūtie rezultāti

PPS izstrādāšanas procesā tika veikta virkne eksperimentu, lai noteiktu sistēmā izmantojamo algoritmu vai pieeju izvēles pamatotību. Vēsturisku pieprasījumu normalizācijai izmanto *z-novērtējuma normalizāciju ar standarta novirzi* un *normalizāciju ar dzīves līkni*, eksperimentāli nosakot piemērotāko. Raksturojošo parametru informatīvuma noteikšanai tika lietots programmnodrošinājumā *Weka* izveidots modulis. Klasterizācijai izmantoti *k-vidējo sadalošais*, *maksimālās līdzības* un *modificēts k-vidējo sadalošai algoritmi*, ar kuru palīdzību noteikts piemērotākais algoritms un klasteru skaits īsu laika rindu klasterizācijai. Klasterizācijas rezultāti tika apvienoti ar raksturojošajiem parametriem un ar dažādu klasifikatoru palīdzību noteikts piemērotākais, apmācot klasifikatoru ar apmācības datu kopu un pārbaudot klasifikatora precizitāti ar testēšanas datu kopu. Uz izveidotā klasifikatora apmācības procesā iegūto rezultātu bāzes tika veikta "jaunā" produkta prognoze, izmantojot tikai preces raksturojošos parametrus. Eksperimentu veikšanai tika izmantots programmnodrošinājums *OrangeCanvas*, *Statistica* un *MS Office Excel*.

PPS izmantoto datu kopu apraksts

Eksperimentos tika izmantoti reāli apģērbu mazumtirdzniecības uzņēmuma vēsturiski pieprasījuma dati par 2005. gadu, kuri saturēja 423 objektus pēc datu attīrīšanas procesa, kas tika izmantoti modeļa apmācībai un 149 objekti par 2006. gadu — modeļa testēšanai. Lai tos varētu izmantot salīdzināšanai ar izveidotajiem paraugmodeļiem, to dzīves ilgums reducēts līdz 12 periodiem (mēnešiem). Apmācības un testēšanas datu kopas saturēja arī produktu raksturojošos parametrus (preces veids, tips un cena) katram objektam. Preces veids raksturo sezonālus(1) un nesezonālus(2) produktus. Savukārt, preces tipi bija šādi: jakas(1); vestes(2);

bikses(3); krekli(4); džemperi(5); somas(6); jostas(7); šorti(8); T-krekli(9), apavi(10) un dažādi(11).

Vēsturisku pieprasījuma datu normalizācija

Sākotnēji, vēsturisku pieprasījumu apmācības datu kopas normalizācijai tika izmantota *z-novērtējuma normalizācija ar standarta novirzi*, kas parasti tiek pielietota datu iegūšanā dominējošu atribūtu vērtību nolīdzināšanai, ja nav zināmas maksimālās vērtību robežas. Kā otra pieeja tika izmantota *pārdošanas apjomu normalizācija ar dzīves likni*, kas tika izmantota citu autoru darbos [64] līdzīgu datu struktūru normalizācijai. Eksperimentāli veicot klasterizāciju, lietojot *k-vidējo sadalošo* algoritmu ar desmit klasteriem (eksperimentāli noteikts kā piemērotākais), dažādām normalizācijas pieejām un izmantojot *10-kārtu šķērsvalidāciju*, un nosakot apmācības kļūdas, tika secināts, ka labākus rezultātus uzrāda vēsturisku pieprasījuma datu *normalizācija ar dzīves līkni*, kuras iegūto rezultātu novērtējums klasifikācijā ir parādīts 3.5. attēlā.

	Summary for k-means clustering (B_Dati_A) Number of clusters: 10 Total number of training cases: 423
Algorithm	k-Means
Distance method	Euclidean distances
Initial centers	Maximize initial distance
MD casewise deletion	Yes
Cross-validation	10 folds
Testing sample	0
Training cases	423
Training error	0,283546
Number of clusters	10

3.5. att. Pārdošanas apjomu datu kopas *normalizācija ar dzīves līkni*

Savukārt, pieeja, kas izmantoja *z-novērtējuma normalizāciju ar standarta novirzi*, kuras iegūtie rezultāti ir parādīti 3.6. attēlā, uzrāda lielāku kļūdas līmeni (3,28), nekā pieejas *normalizācija ar dzīves līkni* noteiktā kļūda (0,28).

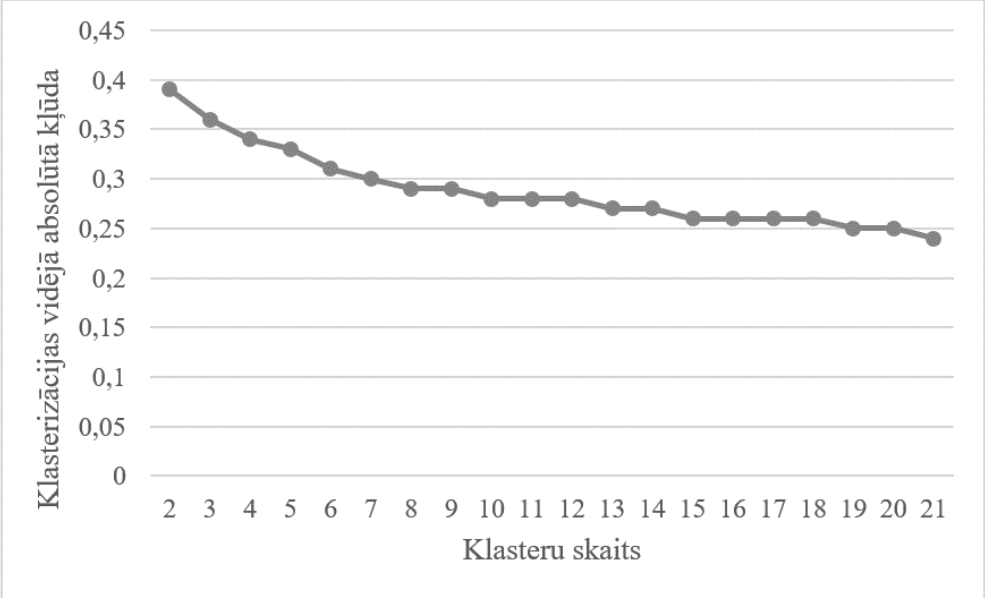
	Summary for k-means clustering (Laika_rindas_Z-novert-norm) Number of clusters: 10 Total number of training cases: 423
Algorithm	k-Means
Distance method	Euclidean distances
Initial centers	Maximize initial distance
MD casewise deletion	Yes
Cross-validation	10 folds
Testing sample	0
Training cases	423
Training error	3,281848
Number of clusters	10

3.6. att. Pārdošanas apjomu datu kopas *z-novērtējuma normalizācija ar standarta novirzi*

Vadoties pēc vēsturisku pieprasījuma datu ar dažādu normalizācijas pieeju veiktajiem eksperimentiem, par piemērotāko PPS izstrādē tiek uzskatīta *normalizācijas pieeja ar dzīves līkni*.

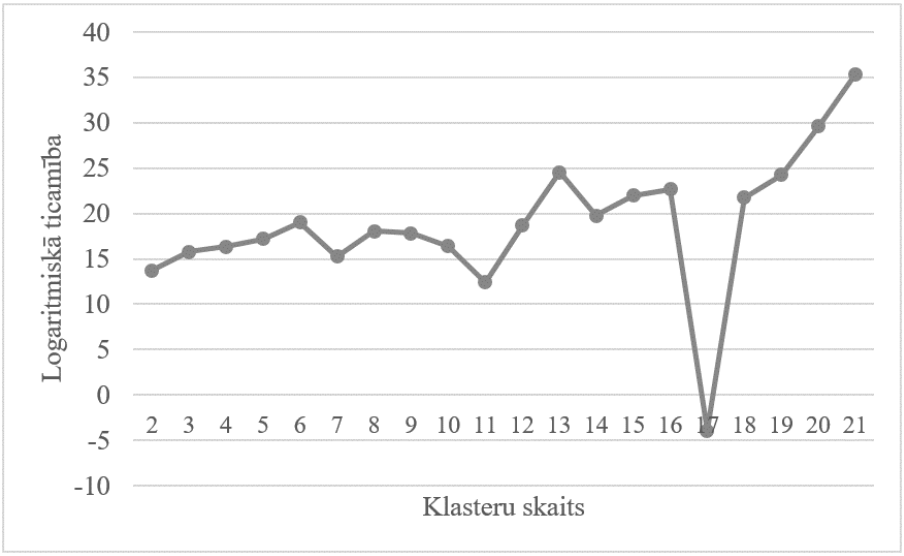
Vēsturisku pieprasījuma datu klasterizācija

Klasterizācijas process ar apmācības datu kopu tika veikts, pielietojot *k-vidējo sadalošo* un *maksimālās līdzības* algoritmus. Iegūtie ar *k-vidējo sadalošo* algoritmu rezultāti parādīti 3.7. attēlā un ar *maksimālās līdzības* algoritmu — 3.8. attēlā. Logaritmiskās ticamības samazinājums pie 17 klastera (skatīt 3.8. attēlu), iespējams, rodas tāpēc ka pie šī klasteru skaita, salīdzinot ar citiem klasterim, ir neliels objektu skaits, kuriem Baijesa varbūtība bija 1.



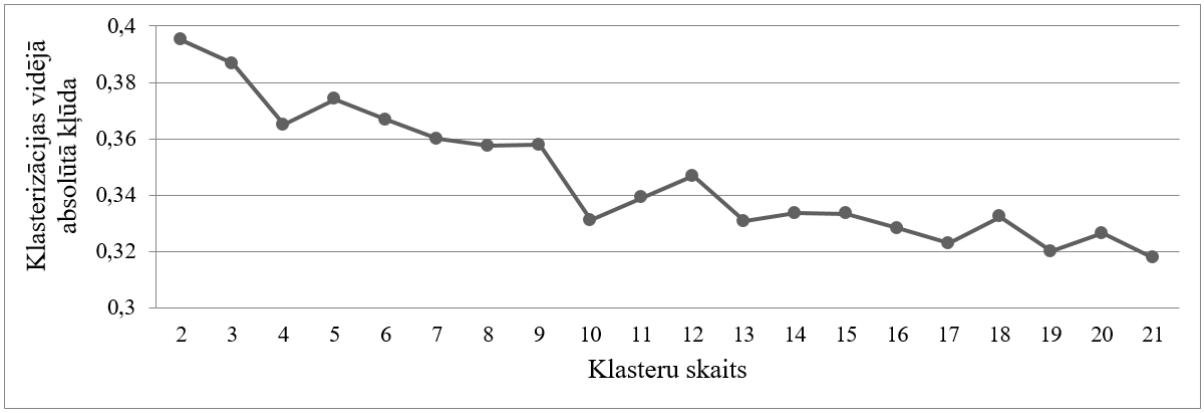
3.7. att. *K-vidējo sadalošā* algoritma klasterizācijas kļūdas novērtējums pie dažādu klasteru skaita

Iegūtie rezultāti parāda, ka noteiktais piemērotākais klasteru skaits abos gadījumos ir maksimālais 21, jo vidējā absolūtā klasterizācijas kļūda ir jāņem pēc minimālās vērtības, bet logaritmiskā ticamība — pēc maksimālās vērtības. Abi klasterizācijas algoritmi nespēj veikt klasteru analīzi, apstrādājot vēsturiskas pieprasījuma vērtības — īsas laika rindas. Gadījumā ar *k-vidējo sadalošo* algoritmu, pieaugot klasteru skaitam, kļūda samazinās, bet ar *maksimālās līdzības* algoritmu, pieaugot klasteru skaitam, logaritmiskā ticamība palielinās. Tas norāda, ka šie klasterizācijas algoritmi nav robusti, un nozīmē, ka tie nespēj veikt precīzu datu klasterizāciju. Tāpēc tika izmantots *modificēts k-vidējo sadalošais* algoritms, lai noteiktu piemērotāko klasteru skaitu, kas būtu nepieciešams apmācības datu kopas klasterizācijai.



3.8. att. *Maksimālās līdzības* algoritma logaritmiskās ticamības novērtējums pie dažādu klasteru skaita

Piemērotākais klasteru skaits ar *modificētu k-vidējo sadalošo* algoritmu tiek noteikts, vadoties pēc 2.3.4. un 2.3.4.1. apakšpunktos aprakstītās metodikas, nosakot klasterizācijas vidējo absolūto kļūdu, kuras rezultāti parādīti 3.9. attēlā.



3.9. att. Klasterizācijas kļūdas aprēķins pie dažādu klasteru skaitu

Pēc klasterizācijas vidējās absolūtās kļūdas aprēķinu rezultātu analīzes, redzams, ka pie 10 klasteriem ir sasniegts pirmais vērā ņemamais minimums, un tā kā turpmāk poligona svārstības ir niecīgas, tad par piemērotāko klasteru skaitu apmācības datu kopas klasterizācijai tiek pieņemts 10. Apmācības datu kopas klasterizācijas rezultāti, ar noteikto piemērotāko klasteru skaitu, ir redzami 3.10. attēlā, kur ar sarkanu krāsu ir parādīti paraugmodeļi, bet ar pelēku — pārējie attiecīgā klastera objekti.

Kļūdas aprēķins starp paraugmodeli un testēšanas datu kopu

Tā kā ir svarīgi noteikt klasterizācijas rezultātā iegūto paraugmodeļu precizitāti, tad jāveic to novērtējums ar testēšanas kopas datiem. Šāda pieeja tiek lietota, jo nav iespējams PPS klasterizācijas modeļa iegūtos rezultātus salīdzināt ar klasiskajiem laika rindu prognozēšanas modeļu rezultātiem [64]. Klasiskie prognozēšanas modeļi lieto laika rindu analīzi, meklējot sakarības laika rindās ilgākā laika posmā, bet PPS analizē laika rindas, kuru garums ir īss un unikāls (neatkārtojas). Testēšanas datu kopas klasterizācijā tika noteikta katra tās objekta piederība kādam no paraugmodeļiem (iegūti apmācības datu kopas klasterizācijas rezultātā), aprēķinot *Eiklīda* attālumu. Testēšanas datu kopas objekts savu piederību paraugmodelim iegūs pēc mazākā attāluma līdz kādam no paraugmodeļiem. Paraugmodeļu attēlojums ar testēšanas kopas datiem ir parādīts 3.11. attēlā, kur ar sarkanu krāsu ir attēloti paraugmodeļi, bet ar pelēku — testēšanas kopas objekti. Pilns kļūdu aprēķinu metodikas pielietojums ar piemēru pa soļiem ir parādīts 1. pielikumā.

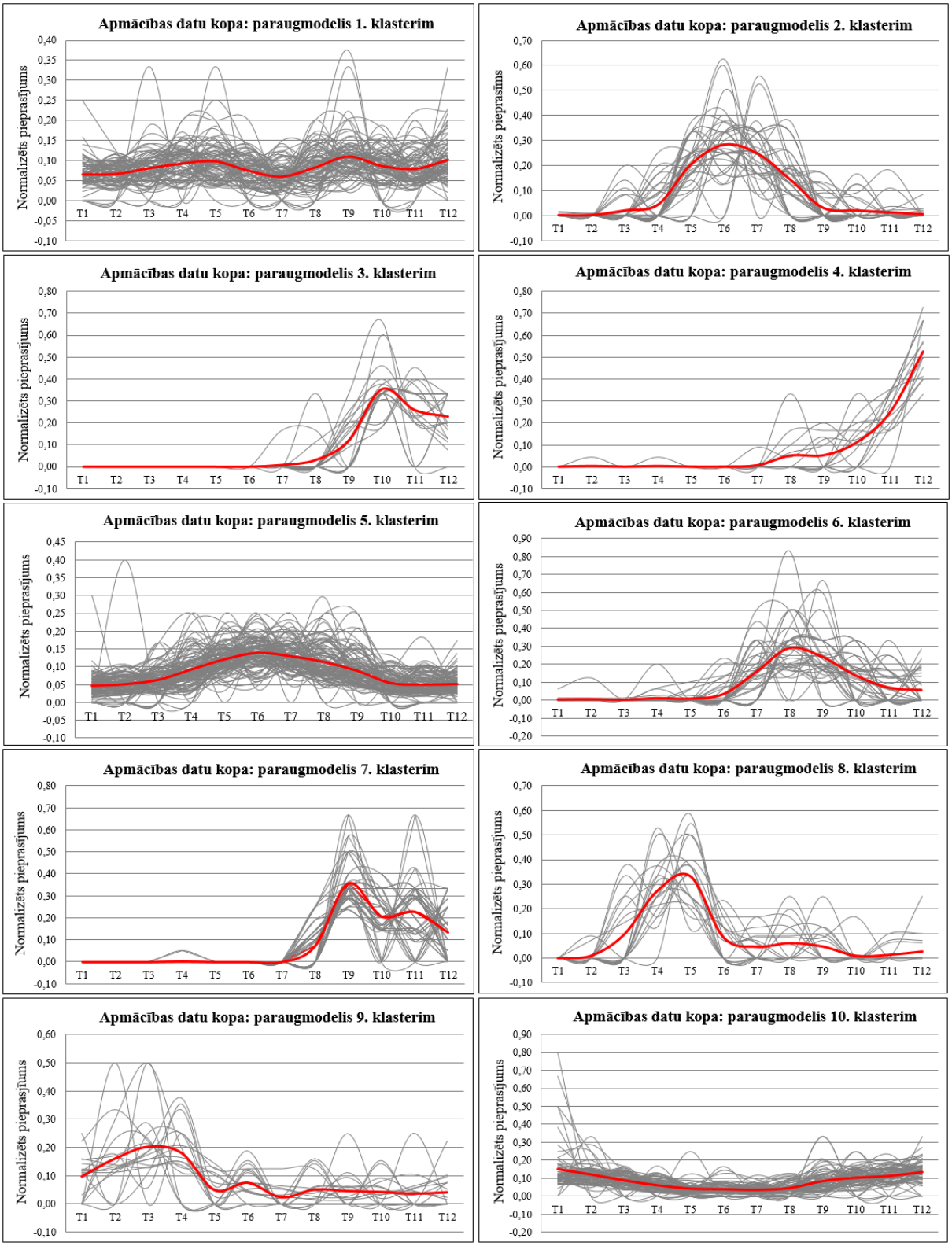
Pēc veiktajiem aprēķiniem, nosakot vidējo absolūto kļūdu (0,0546), testēšanas datu kopas objektus salīdzinot ar apmācības procesā iegūtajiem paraugmodeļiem, tika iegūts modeļa efektivitātes novērtējums. Noteiktā kopējā absolūtā kļūda norāda, ka testēšanas kopas objekti ar pietiekami augstu precizitāti ir "atraduši" piederību apmācības procesā noteiktajiem paraugmodeļiem, arī vizuāli aplūkojot 3.11. attēlu, ir novērojama šāda tendence. Iegūtais

novērtējums pamato, klasterizācijas procesā noteikto piemērotākā klasteru skaita izvēles piemērotību.

PPS piemērotākā klasifikatora noteikšana un precizitātes novērtējums

Lai izvēlētos piemērotāko klasifikatoru PPS uzbūvei, ir nepieciešams veikt virkni eksperimentu. Klasifikatora precizitātes novērtēšanai tiek izmantoti algoritmi, kas atlasīti salīdzinošās analīzes rezultātā (skatīt 1.5. apakšpunktu), lai pēc precizitātes testēšanas rezultātiem noteiktu atbilstošāko klasifikācijas algoritmu, kuru varētu izmantot PPS.

Klasifikatoru precizitātes salīdzināšanai tika lietoti vidējās absolūtās kļūdas un vidējās kvadrātiskās kļūdas aprēķini, jo dažādas kļūdu aprēķinu tehnikas nodrošina precīzāku iegūto rezultātu interpretāciju [64]. Precizitātes novērtēšanai klasifikatoru apmāca, lietojot apmācības datu kopu, bet pārbaudei — testēšanas datu kopu. Programmnodrošinājumā *OrangeCanvas* tika no vairākiem klasifikācijas algoritmiem izveidots modelis, ar kura palīdzību veikti eksperimenti piemērotākā klasifikatora noteikšanai. Iegūtie rezultāti ir parādīti 3.1. tabulā, no kuras redzams, ka zemāko kļūdas vērtību ar abām kļūdas novērtējuma metodēm, uzrādīja *C4.5* algoritms.



3.10. att. Apmācības datu kopas klasterizācijas rezultātā iegūtie klasteru paraugmodeļi

3.1. tabula

Dažādu klasifikatoru uzrādītās kļūdas ar testēšanas datu kopu

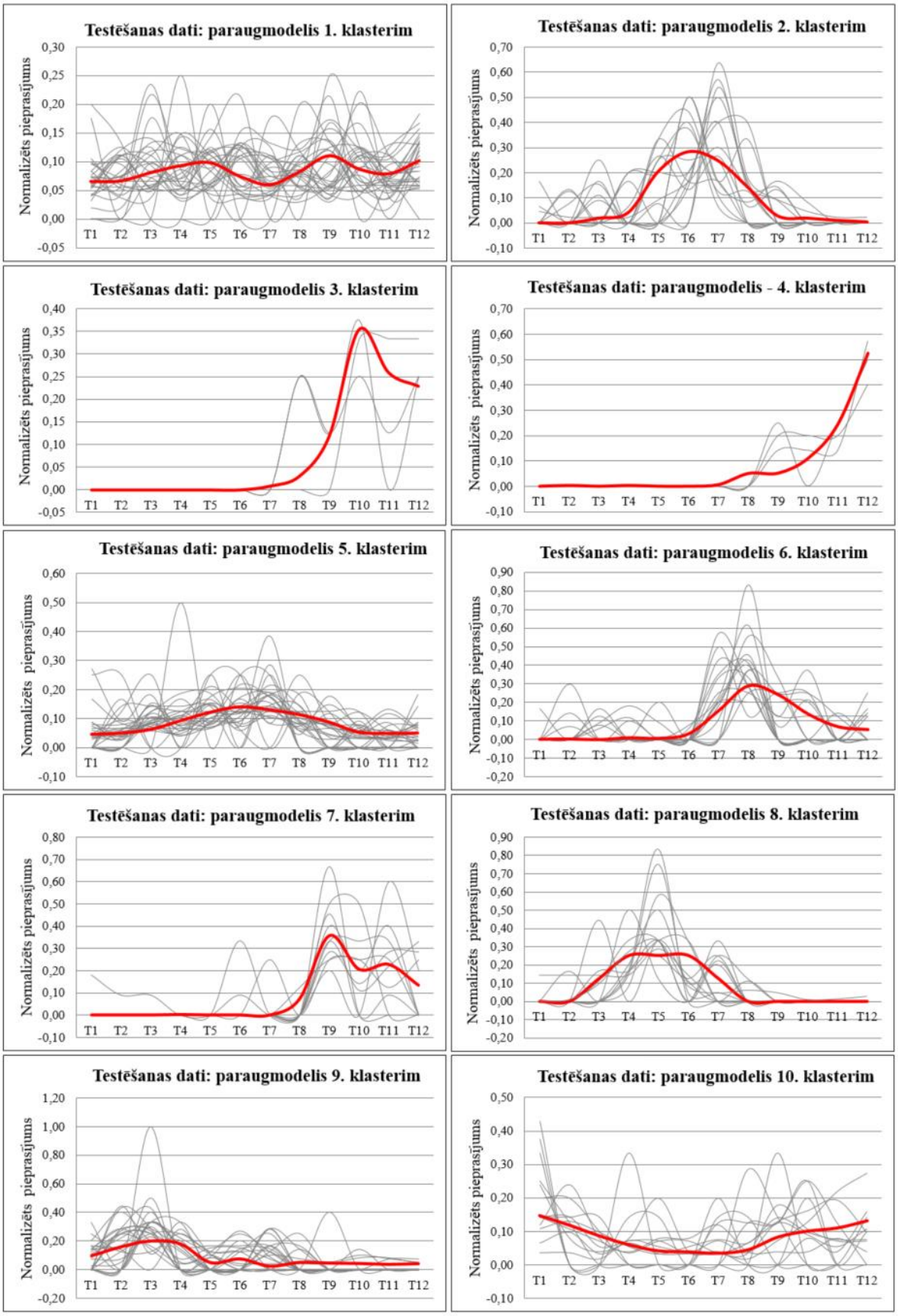
Kļūdas novērtējuma veids	Klasifikators					
	ZeroR	*OneR*	*JRip*	*Naivais Baijesa*	*k-tuvāko kaimiņu*	*C4.5*
Vidējā kvadrātiskā kļūda (RMSE)	0,295	0,375	0,284	0,282	0,298	**0,264**
Vidējā absolūtā kļūda (MAE)	17,4	14,1	16,1	15,6	16,0	**12,9**

Iegūtie induktīvā lēmuma koka raksturojošie parametri ir parādīti 3.2. tabulā, bet izveidotais induktīvais lēmumu koks redzams 3.12. attēlā. Pēc klasifikācijas precizitātes rezultātiem PPS-ā tiek izmantots *C4.5* algoritms.

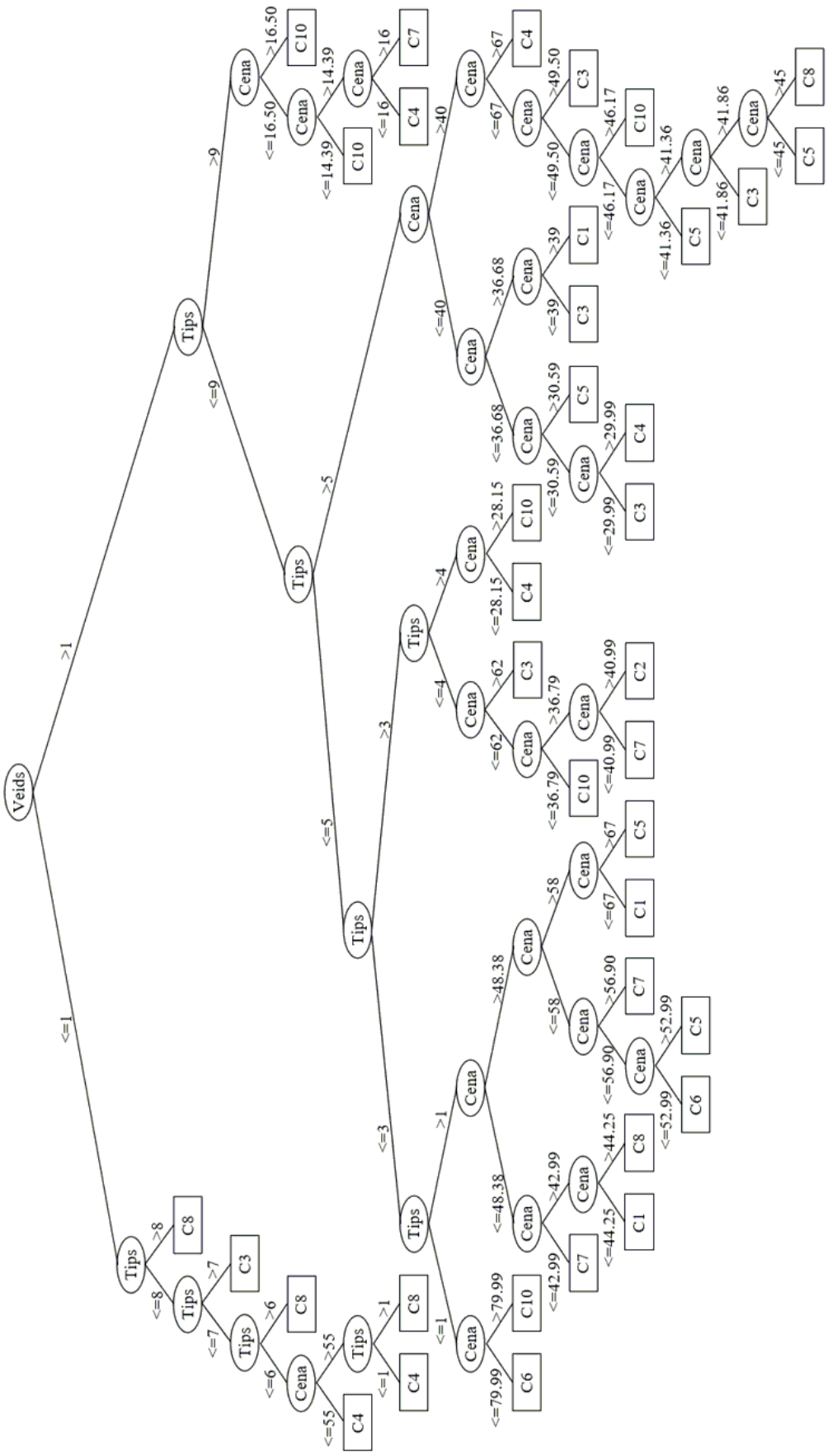
3.2. tabula

Izveidotā induktīvā lēmumu koka raksturojošie parametri

Klašu skaits	Objektu skaits	Atribūtu skaits	Lapu skaits	Iekšējo mezglu skaits
10	423	4	38	75



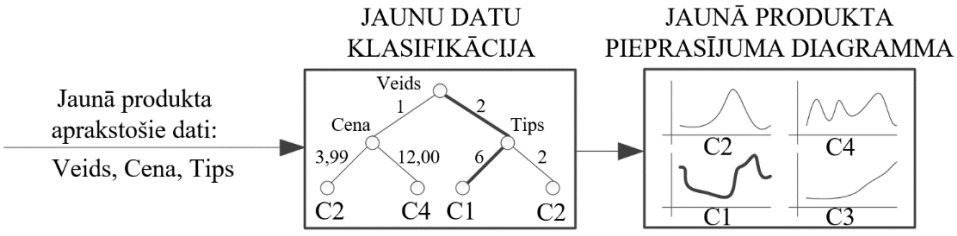
3.11. att. Testēšanas datu kopas objektu sadalījums pēc noteiktās piederības pa paraugmodeļim, kas noteikti apmācības kopas klasterizācijas rezultātā



3.12. att. Klasifikācijas procesā iegūtais PPS induktīvais lēmumu koks

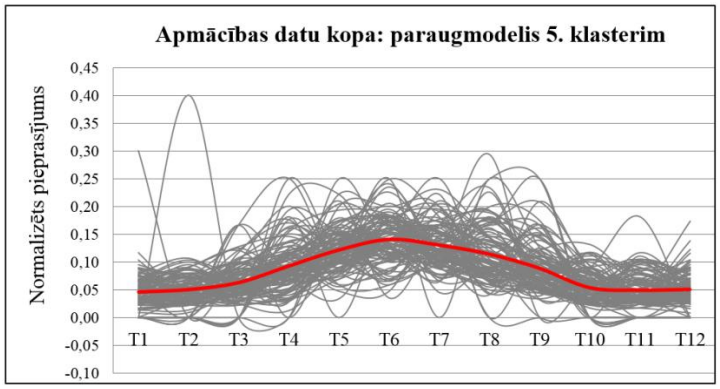
Jauna produkta pieprasījuma prognozēšana

Prognozēšanas procesa gaita shematiski ir parādīta 3.13. attēlā. Jauna produkta pieprasījums tiek prognozēts uz klasifikācijas procesā iegūtā klasifikācijas koka, kurš parādīts 3.12. attēlā, projicējot uz tā jaunā produkta raksturojošos parametrus (preces tips, cena un veids). Pieņemsim, ka jaunā produkta raksturojošie parametri ir šādi: *Produkta veids=2* (sezonāls), *Preces tips=3* (bikses) un *Cena=56,69*. Raksturojošie parametri tiek projicēti uz induktīvo lēmumu koku, virzoties pa to no saknes uz leju, zara virzienu izvēloties pēc parametra vērtības. Process turpinās līdz tiek sasniegta kāda no induktīvā lēmuma koka lapām.



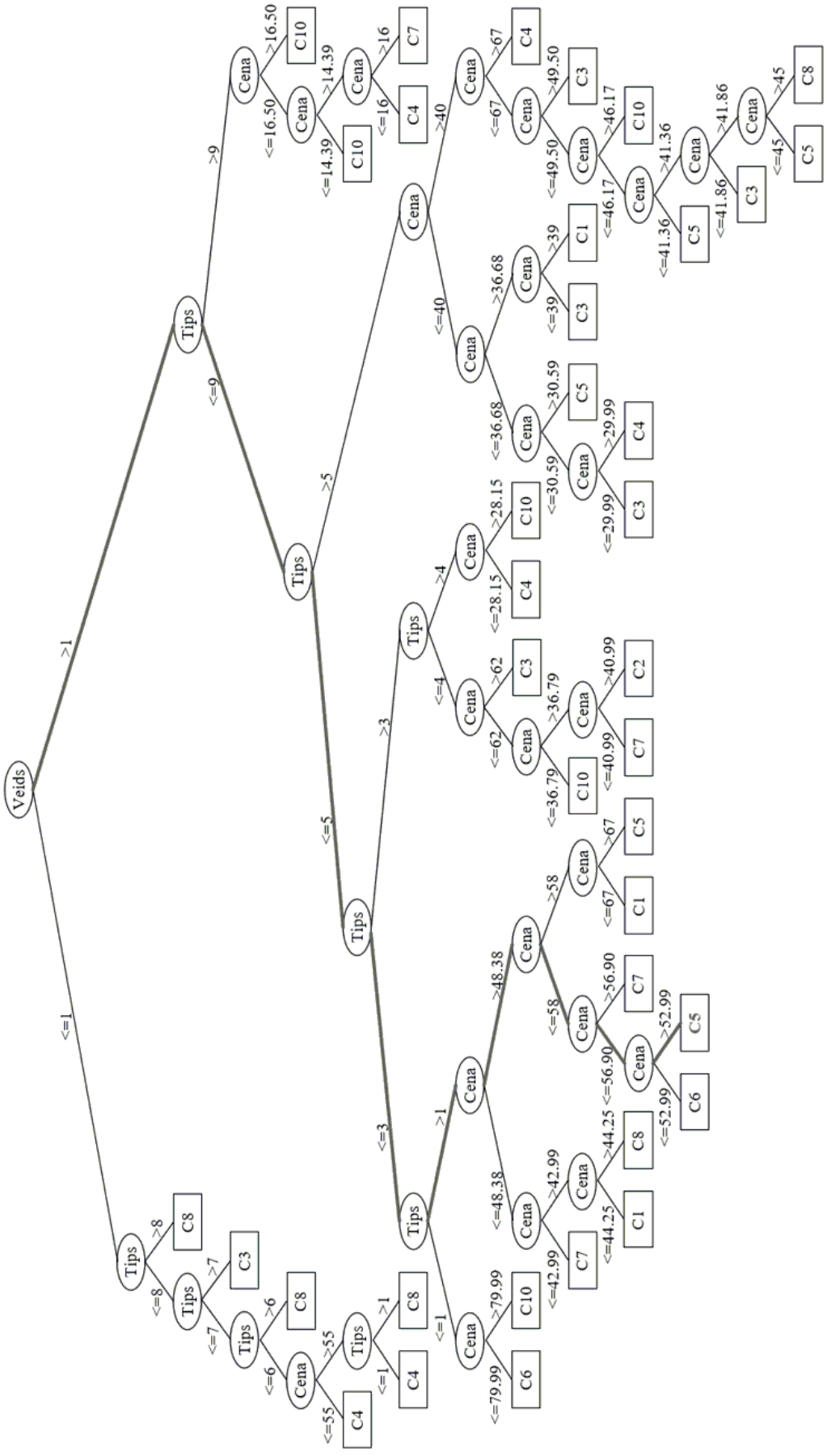
3.13. att. Jauna produkta pieprasījuma prognoze

Iegūtā lapa norāda uz paraugmodeli (C5), kas tika noteikts klasterizācijas procesā un ir redzams 3.14. attēlā ar sarkanu līniju, kāds būs prognozētā jaunā produkta iespējamais pieprasījums 12 mēnešiem. Pilnu raksturojošo parametru projicēšanu uz induktīvo lēmumu koku ir redzams 3.15. attēlā.



3.14. att. Jaunā produkta iespējamais pieprasījums — prognoze

Uz šīs prognozes pamata lēmējpersona var pieņem lēmumu par jaunā produkta pasūtījuma apjoma daudzumu, kas būs nepieciešams turpmāk tirdzniecībai uzņēmumā. No iegūtās līknes lēmējpersona var izdarīt secinājumus, ka jaunā produkta iespējamais pieprasījums būs aktuāls visu gadu ar augšupejošu tendenci pavasara un vasaras mēnešos.



3.15. att. Jaunā produkta raksturojošo parametru projicēšna uz induktīvā lēmumu koka

3.1.4. Kopsavilkums un secinājumi par PPS

Izveidota PPS [29], ar kuru iespējams noteikt jaunā pasūtamā produkta pieprasījuma prognozi nākamajiem 12 mēnešiem. Iegūtais prognozes rezultāts palīdz lēmējpersonai pieņemt lēmumu, veicot jaunu produktu pasūtījumus. Eksperimenti parāda, ka izveidotā PPS spēj apstrādāt atšķirīgas datu struktūras — īsas laka rindas un to raksturojošos parametrus. Iegūtie rezultāti, kas attēloti paraugmodeļu veidā, ir viegli interpretējami sistēmas lietotājam.

No veiktajiem eksperimentiem un iegūtajiem rezultātiem var secināt, ka:

a) normalizējot tirdzniecības pieprasījuma datus, labākus rezultātus uzrāda *normalizācija ar dzīves līkni* pieeja, kas skaidrojams ar to, ka *z-novērtējuma normalizācija ar standarta novirzi* normalizē laika rindas vērtības negatīvā un pozitīvā vērtību diapazonā, bet *normalizācija ar dzīves līkni* — tikai pozitīvā vērtību diapazonā;

b) pētot klasterizācijas algoritmus, eksperimentāli tika noteikts, ka *k-vidējo sadalošā* un *maksimālās līdzības* algoritmu iegūtos rezultātus tiešā veidā pielietot PPS-ā nav iespējams, tāpēc tika izstrādāta *k-vidējo sadalošā* algoritma modifikācija, kas ļāva noteikt piemērotāko klasteru skaitu analizējamajai datu kopai;

c) eksperimentāli tika noteikts testēšanas datu kopas klasterizācijas precizitātes novērtējums attiecība pret iegūtajiem apmācības datu kopas paraugmodeļiem;

d) eksperimentāli tika noteiktas dažādu klasifikatoru radītās kļūdas. Lai iegūtie rezultāti būtu precīzāk novērtējami, tika izmantotas vidējās absolūtās kļūdas un vidējās kvadrātiskās kļūdas aprēķinu pieejas. Iegūtie rezultāti parādīja, ka PPS-ai jāizmanto *C4.5* algoritms, kas balstās uz induktīvajiem lēmuma kokiem, jo šī algoritma radītās klasifikācijas kļūdas bija mazākās;

e) iegūtie klasifikācijas rezultāti, kas tiek attēloti induktīvo lēmumu koku veidā, ļoti uzskatāmi ataino klasifikācijas procesa būtību un ir viegli saprotami un uztverami arī lietotājam, kas nav datu analīzes speciālists;

f) izveidotās PPS rezultātu interpretācija, kas iegūto prognozi attēlo paraugmodeļa (līknes) veidā, ir viegli uztverama un saprotama sistēmas lietotājam, jo skaidri parāda prognozējamā produkta pieprasījumu nākamajiem 12 mēnešiem.

3.2. Sirds nekrozes riska prognozēšanas sistēma

Sirds nekrozes riska prognozēšanas sistēmas (SNRPS) ietvaros tiek piedāvāts risinājums bioinformātikas nozares speciālistiem, kā efektīvāk apstrādāt īsas laika rindas ar datu iegūšanas metodēm un algoritmiem. Tradicionāli, farmakoloģiskajos pētījumos iegūto rezultātu novērtējums tiek veikts ar vispārpieņemtu nozares metodoloģiju [6]. Iegūto rezultātu analīze un novērtējums, parasti, tiek pamatots ar matemātiskās statistikas pieejām. Bioinformātikas nozarē — farmakoloģijā, risinot uzdevumus, kas saistīti ar liela apjoma un laikietilpīgiem eksperimentiem, tiek iegūti jauni dati, kuri tiek uzkrāti. Balstoties uz šiem uzkrātajiem datiem, pielietojot datu iegūšanas tehnoloģijas, iespējams iegūt jaunas zināšanas, uz kuru bāzes farmakologi, interpretējot iegūtos rezultātus, varētu uzlabot pētījumu rezultātu precizitāti, kā arī samazināt eksperimentu apjomu, kuros ir iesaistīti dzīvnieki (laboratorijas žurkas).

Sistēmas izstrādāšanas procesā tiek analizēti farmakoloģisko pētījumu laboratorisko izmeklējumu dati, kas iegūti, izmantojot izolētās sirds išēmijas-reperfūzijas modeli [48,49]. Eksperimenti veikti, izmantojot *Wistar* līnijas žurkas, kam noteiktu laika periodu (astoņu nedēļu garumā) ievadītas pētāmās sirds darbību stimulējošās vielas, kuras pievienotas pārtikai. Katra dzīvnieku grupa tiek barota noteiktu laika periodu ar vienu no pētāmās vielas veidiem. Farmakoloģisko pētījumu mērķis ir noteikt pētāmo vielu efektivitāti sirds šūnu aizsardzībai pret išēmijas-reperfūzijas bojājumu, nosakot sirds nekrozes (atmirušo audu) daudzumu. Pētījumos izmantotais Mildronāts® ir Latvijā izstrādāts antiišēmisks līdzeklis, kas, optimizējot sirds enerģijas metabolismu [16], aizsargā šūnas no išēmiskiem bojājumiem slimniekiem ar stenokardiju, hronisku sirds mazspēju, smadzeņu asinsrites traucējumiem, un uzlabo fiziskās slodzes toleranci un intelektuālās funkcijas išēmijas gadījumā [16, 61]. Eksperimenti ar dzīvnieku sirdīm tika veikti atbilstoši Eiropas Savienības dzīvnieku aizsardzības ētikas noteikumiem [49]. Šo eksperimentu, vai kā to sauc farmakoloģijas nozares speciālisti, ”Izolētā sirds”, rezultātā fiksēti mērījumi par sirds kontrakcijas spēku noteiktā laika posmā.

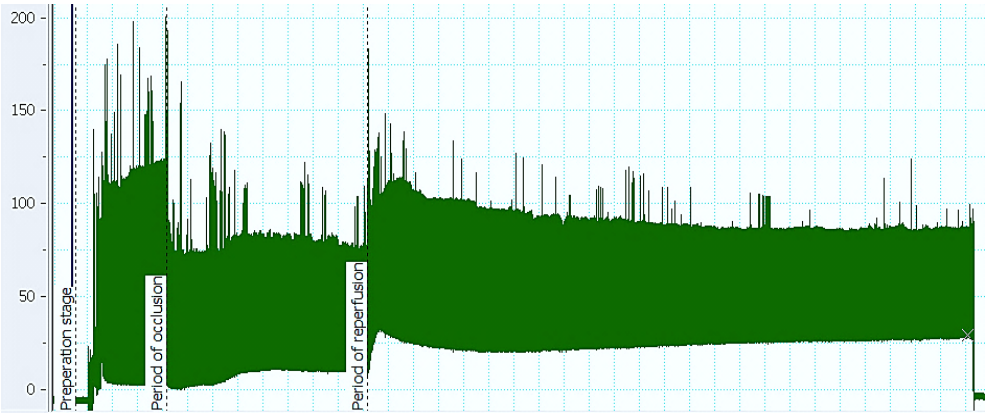
“Izolētās sirds” eksperiments tika veikts pēc tehnoloģijas, kas aprakstīta zinātniskajā rakstā [47]. Farmakoloģiskā eksperimenta mērķis ir noteikt atmirušo sirds audu (nekrozes) procentuālo attiecību, kas atkarīga no pētāmās vielas veidiem, ar kuriem tika baroti dzīvnieki.

Izolētās sirds eksperimentus iespējams sadalīt trīs posmos, kuri ir parādīti 3.16. attēlā:

a) sirds adaptācijas periods *(Preperation stage)*, kas ilgst 20 min., kad izolētajā sirdī ievieto zondi sirds kontrakcijas spēka un sirds ritma nolasīšanai;
b) oklūzija *(Period of occlusion)* — lejupejošās koronārās artērijas nosiešana, kas ilgst

40 min.;

c) reperfūzija (*Period of reperfusion*) — asins vadu atbrīvošana, kas ilgst 120 min.



3.16. att. Izolētās sirds eksperimentu norises posmi

Sirds kontrakcijas spēka un sirds ritma reģistrēšanai oklūzijas laikā tiek izmantota firmas *ADInstruments* aparatūra un programmnodrošinājums, nolasot datus ar intervālu 60 sekundes, iegūstot datu kopu, kas sastāv no 40 laika nolasījumiem oklūzijas periodā (katru minūti tiek reģistrēts sirds kontrakcijas spēks) ar sirds kontrakcijas spēka izmaiņām, un veido pirmo datu grupu. Iegūtās datu kopas vērtības ir laika rindas, bet tā kā to novērojumu ilgums ir pārāk īss un tas neatkārtojas, tad šīs vērtības ir uzskatāmas par īsām laika rindām. Analizējot īsas laika rindas ir praktiski neiespējami noteikt tajās funkcionēšanas likumsakarības, tāpēc šāda veida uzdevumi ir uzskatāmi par grūti formalizējamiem. Pie šāda veida uzdevumiem pieskaitāmi, piemēram, gēnu un proteīnu ekspresijas analīze [21, 24] u. c. Otra datu grupa ir laboratorijas dzīvniekus raksturojošie parametri un farmakoloģisko eksperimentu laikā iegūtais sirds nekrozes riska novērtējums. Raksturojošie parametri apraksta laboratorijas dzīvnieka, piemēram, svaru, pētāmo vielu, asins plazmas analīzes parametrus un sirds nekrozes risku. Raksturojošos parametros ir jāatrod likumsakarības. Šeit izveidojas problēma, starp īsām laika rindām un to raksturojošajiem parametriem, kura ir jāatrisina ar datu iegūšanas metodēm un algoritmiem. Pastāv vairākas pieejas, kas izmanto dažādu datu iegūšanas metožu kombinēšanu, kuras nodrošina nepārtrauktu un diskrētu datu apstrādi ar mērķi iegūt jaunas zināšanas [40, 64].

Ar klasterizācijas palīdzību, analizējot īsas laika rindas, iespējams noteikt līdzīgas objektu grupas. Šīs atrastās grupas pievieno raksturojošo parametru datu kopai, kurā ar klasifikācijas palīdzību meklē sakarības nosacījumu likumu formā [27]. Iegūtajiem nosacījumu likumiem pievieno farmakoloģiskajos eksperimentos noteikto sirds nekrozes riska novērtējumu, veidojot nosacījumu likumu bāzi (objekta klasifikācijas rezultātā tiek veidots viens likums). Dzīvnieka sirds nekrozes riska prognoze tiek veikta, pamatojoties uz

raksturojošajiem parametriem (tiek ievadīti sistēmā), izveidoto nosacījumu likumu bāzi, no kuras tiek atlasīts statistiskais sadalījums, pēc kuras aprēķina sirds nekrozes risku. Šādai prognozēšanas sistēmai ir nepieciešams darboties ar augstu precizitāti, tāpēc tās izstrādes laikā ir jāminimizē kā klasterizācijas tā arī klasifikācijas kļūdas.

3.2.1. SNRPS uzdevuma nostādne

Farmakoloģisko pētījumu institūts, kas nodarbojas ar medikamentu izstrādi sirds darbības funkciju uzlabošanai un veic laboratoriskus eksperimentus sirds nekrozes riska noteikšanai, izvirzīja uzdevumu izstrādāt sirds nekrozes riska prognozēšanas sistēmu, kura varētu noteikt iespējamo sirds nekrozes risku, tikai ar laboratorijas dzīvnieka raksturojošajiem parametriem. Šādas sistēmas izveide palīdzētu farmakoloģijas nozares speciālistiem ietaupīt laiku pētījumu rezultātu iegūšanai un eksperimentos iesaistīt mazāku laboratorijas dzīvnieku skaitu. Pēc izvirzītajiem kritērijiem, sirds nekrozes riska prognozēšanas sistēmai ir jārealizē šādi uzdevumi:

1. Jāapstrādā sirds kontrakcijas spēka dati, izveidojot īsas laika rindas.
2. Jāveic īsu laika rindu pirmapstrāde un to sagatavošana datu analīzes procesam.
3. Laika rindās ir jānosaka līdzības pazīmes ar klasteru analīzes palīdzību.
4. Jānosaka informatīvākie raksturojošie parametri, lietojot atribūtu informatīvuma noteikšanu.
5. Jāapvieno klasterizācijas rezultāti ar laboratorijas dzīvnieka atlasītajiem raksturojošajiem parametru datiem, nosakot sakarības starp šīm grupām ar klasifikācijas palīdzību.
6. Jāizveido nosacījumu likumi no klasifikācijas rezultātiem un farmakoloģisko eksperimentu rezultātā iegūtā sirds nekrozes riska novērtējuma.
7. Jāpārbauda izveidotās sistēmas precizitāte.
8. Jārealizē "jauna" laboratorijas dzīvnieka sirds nekrozes riska prognozēšana.
9. Jāatspoguļo iegūtie rezultāti.

Lai realizētu izvirzītos uzdevumus, jāveic virkne pētījumu, kas saistīti ar datu klasterizāciju un klasifikāciju.

3.2.2. SNRPS uzbūve

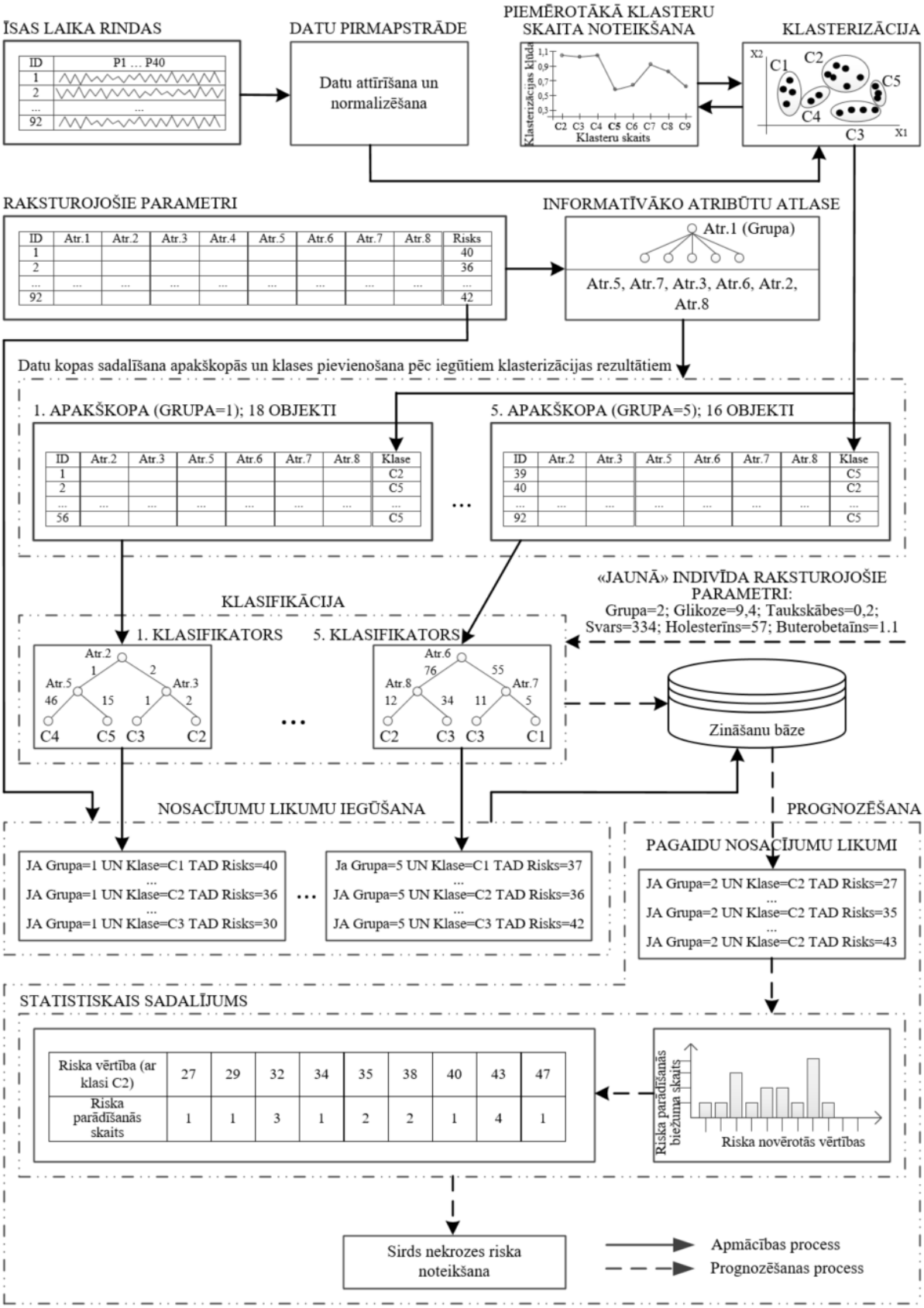
Sākotnēji, tiek veikta īsu laika rindu pirmapstrāde, kurā tiek atmesti pirmais un pēdējais mērījums, lai izvairītos no iespējamā trokšņa (kļūdaini rādījumi). Tad tiek izveidota laika rinda ar 38 periodiem katram datu kopas objektam, atlasot datus par sirds kontrakcijas spēku. Datu normalizācija tiek veikta, izmantojot *normalizāciju ar dzīves līkni* [40, 64] un *z-*

novērtējuma normalizāciju ar standarta novirzi [1] pieejas, kuras tika izmantotas arī PPS izstrādāšanā. Sirds nekrozes riska prognozēšanas sistēmas darbības princips parādīts 3.17. attēlā. Tās uzbūve balstās uz vairāku datu iegūšanas metožu pielietošanu. Matemātiskās statistiskas metožu pieejas ir ierobežotas, risinot uzdevumus, kur jāanalizē sakarības starp īsām laika rindām. Tāpēc šādu uzdevumu atrisināšanai tiek pielietotas datu iegūšanas pieejas, kuras tiek uzskatītas par piemērotākām [76]. Īsas laika rindas tiek klasterizētas ar *modificētu k-vidējo* algoritmu, nosakot piemērotāko klasteru skaitu, kas nepieciešams līdzīgu objektu apvienošanai klasteros. Indivīda raksturojošajos parametros tiek meklēta atribūtu savstarpējā korelācija, nosakot to mijiedarbības pakāpi un atlasot informatīvākos atribūtus tālākajai datu analīzei. Raksturojošo atribūtu datu kopa tiek sadalīta piecās vienādās apakškopās, tādēļ, ka barībai pievieno piecu veidu pētāmās vielas. Sadalītajām apakškopām pievieno klasterizācijas rezultātā iegūto klasi, vadoties pēc objekta identifikatora. Katrai no izveidotajām apakškopām ar klasifikācijas palīdzību tiek noteikta saikne starp klasterizācijā iegūto objekta klasi un šī objekta raksturojošajiem parametriem, izmantojot induktīvo lēmumu koku algoritmu *C4.5* [58]. Objekta klasifikācijas procesā, no uzbūvētā lēmuma koka, tiek iegūts nosacījuma likums „JA …UN … TAD …" veidā, kas glabā informāciju par klasterizācijā iegūto klasi, pētāmās vielas grupu un farmakoloģiskajos eksperimentos iegūto sirds nekrozes riska novērtējumu. Iegūtie nosacījumu likumi tiek saglabāti zināšanu bāzē.

Sirds nekrozes riska prognozēšana tiek veikta, ievadot sistēmā "jaunā" indivīda raksturojošos parametrus. Vadoties pēc pētāmās vielas grupas, raksturojošie parametri tiek projicēti uz attiecīgo klasifikatoru, iegūstot klases vērtību un izveidojot nosacījumu "JA … UN …" (piemēram, JA *Grupa*=2 UN *Klase*=C2), kas tiek nodots zināšanu bāzei. Tālāk no zināšanu bāzes tiek atlasīti visi tie gadījumi, kas atbilst izveidotajam nosacījumam (JA *Grupa*=2 UN *Klase*=C2). Katrs nosacījums tiek papildināts ar atrasto informāciju zināšanu bāzē, veidojot pagaidu nosacījumu likumu kopu "JA … UN … TAD", piemēram:

- JA *Grupa*=2 UN *Klase*=C2 TAD *Risks*=27;
- JA *Grupa*=2 UN *Klase*=C2 TAD *Risks*=32;
- JA *Grupa*=2 UN *Klase*=C2 TAD *Risks*=54;
- JA *Grupa*=2 UN *Klase*=C2 TAD *Risks*=32;
- …;
- JA *Grupa*=2 UN *Klase*=C2 TAD *Risks*= 40

No atlasītās pagaidu nosacījumu likumu kopas izveidojas sirds nekrozes riska parādīšanās biežuma statistika (statistiskais sadalījums), kas veido riska sadalījuma funkciju, no kuras iegūstam, cik reizes attiecīgā riska vērtība parādās pagaidu likumu kopā.

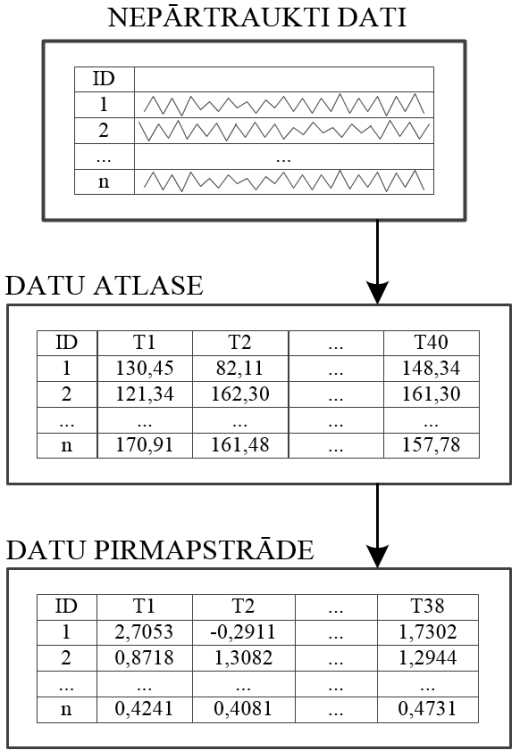


3.17. att. Sirds nekrozes riska prognozēšanas sistēmas uzbūve

Sirds nekrozes riska noteikšanai tiek lietots matemātiskās cerības aprēķins un darba autora piedāvāta pieeja, kas balstīta uz attāluma aprēķinu metrikas. Matemātiskās cerības aprēķinam tiek izmantots sirds nekrozes riska statistiskais sadalījums, kur katrai biežuma statistikas vērtībai piešķir varbūtību pēc riska biežuma skaita, jo biežāk šī vērtība ir sastopama pagaidu nosacījumu likumu kopā, jo varbūtība būs lielāka. Tālāk, saskaitot katras riska aprēķinātās matemātiskās cerības vērtības, tiek iegūta sirds nekrozes riska prognoze. Darba autora izstrādātā pieeja lieto attālumu aprēķinu starp riska parādīšanās biežuma skaitu un riska vērtībām. Šī pieeja ir detalizēti aprakstīta apakšpunktā "Sirds nekrozes riska prognozēšana" un parādīta 3.10. tabulā.

Sirds kontrakcijas spēka pirmapstrādes process

Datu pirmapstrādes procesā veic laika rindu atlasi un datu normalizāciju, kā parādīts 3.18. attēlā. Datu atlases procesā nepārtraukti dati (iegūti ar speciālas programmatūras palīdzību, kas reģistrē sirds kontrakcijas spēku katras 0,25 sekundes 40 minūšu garumā, no šiem datiem tiek atlasīti 40 mērījumi ar 60 sekunžu intensitāti) tiek sadalīti periodos, atmetot pirmā un pēdējā perioda atlasītās laika rindu vērtības, jo tās var saturēt kļūdainus rādījums, kas saistīti ar koronāro asinsvadu nosiešanu vai atbrīvošanu. Atlasītajām laika rindām tiek pielietota datu normalizācija.



3.18. att. Nepārtrauktu datu pirmapstrādes process

Lai noteiktu piemērotāko datu normalizācijas pieeju, sistēmas ietvaros tiek piedāvāts aprobēt *z-novērtējuma normalizāciju ar standarta novirzi* un *pieprasījumu normalizāciju ar dzīves līkni* (skatīt 2.1.2.1. apakšpunktu).

Sirds kontrakcijas spēka klasterizācija

Sirds kontrakcijas spēka izmaiņas laika periodos tiek definētas kā īsas laika rindas. Īsu laika rindu apvienošana grupās ir realizēta tādā veidā, lai vienā grupā atrastos tie objekti, kas ir līdzīgi pēc līknes struktūras un formas [68]. Līdzība starp diviem objektiem tiek salīdzināta katrā laika punktā un noteikta ar attāluma mēra palīdzību. Sistēmas izveidošanas procesā tiek eksperimentāli testēti dažādi klasterizācijas algoritmi: *k-vidējais sadalošais algoritms* (skatīt 2.3.1. apakšpunktu); *maksimālās līdzības algoritms* (skatīt 2.3.3. apakšpunktu); *aglomeratīvais hierarhiskais* (skatīt 2.3.2. apakšpunktu) un darba autora izstrādātais *modificēts k-vidējo sadalošais algoritms* (skatīt 2.3.4. apakšpunktu). Ar klasterizācijas algoritmu ir jānosaka piemērotākais klasteru skaits, kas būtu izmantojams analizējamās datu kopas klasterizācijai. Klasterizācijas rezultātā līdzīgās laika rindas tiek apvienotas klasteros. Svarīgi, lai pielietotais algoritms nodrošinātu klasterizācijas noturību jeb robustumu [35].

Nepārtrauktu un diskrētu datu apvienošana

Diskrēti dati apraksta indivīda raksturojošos parametrus (*Svars*, *Grupa* — barības tips ar kādu barots dzīvnieks, asins plazmas parametri: *Taukskābes, Holesterīns, Glikoze, Triglicerīdi, Butirobetaīns* un sirds nekrozes *Risks*). Nepārtraukti dati raksturo īsas laika rindas, kas attēlo sirds kontrakcijas spēka izmaiņas noteiktā laika periodā. Klasterizācijas rezultātā iegūto klasi, kas raksturo indivīda piederību noteiktai objektu grupai, pievieno šī indivīda diskrētajiem datiem, vadoties pēc identifikācijas numura, ar šādu pieeju tiek atrisināta īsu laika rindu un to raksturojošo parametru apstrāde. Pēc informatīvāko atribūtu atlases un analīzes rezultātiem tiek noteikts, ka pēc atribūta *Grupa* notiek datu kopas sadalīšana apakškopās. Apakškopu skaits tiek noteikts pēc izvēlētā atribūta vērtību skaita (datu kopā sastopami 5 dažādi barības tipu veidi — pētāmā viela). Vadoties pēc atribūta *Grupa* vērtībām objekti tiek iedalīti apakškopās. Jauniegūtās datu kopas tiek izmantotas klasifikācijas procesā, lai noteiktu sakarības starp raksturojošajiem parametriem un klasterizācijā iegūto klasi. Parametrs *Risks*, kas raksturo sirds nekrozes riska novērtējumu, kurš iegūts farmakoloģisko pētījumu rezultātā, izveidojamās datu kopās netiek iekļauts. Šis parametrs tiks izmantots nosacījumu likumu veidošanai, to nevar izmantot sistēmas apmācības procesā, jo tas raksturo sirds nekrozes riska novērtējumu.

Sirds nekrozes riska klasifikācija

Sirds nekrozes riska klasifikācijas procesam ir jānosaka likumsakarības starp indivīda raksturojošajiem parametriem un klasterizācijas procesā iegūtajiem rezultātiem. Šādas likumsakarības iespējams noteikt, pielietojot klasifikācijas algoritmus, kuri atlasīti salīdzinošās analīzes rezultātā, kas sīkāk aprakstīta 1.5. apakšpunktā. Pastāv nosacījums, ka algoritma iegūtajiem rezultātiem ir jābūt viegli interpretējamiem. Tāpēc par pamatu tiek izvēlēti vairāki algoritmi, tādi kā *Naivais Baijess*, kas lieto aprioras un nosacītas varbūtības; *kNN* — izmanto attālumus; *C4.5* — balstīts uz induktīvajiem lēmumu kokiem, no kuriem eksperimentāli tika noteikts piemērotākais sirds nekrozes riska klasifikācijai, vadoties pēc klasifikatoru precizitātes novērtējuma.

Riska biežuma parādīšanās statistika

No pagaidu nosacījumu likumu kopas (piemēram, {JA *Grupa*=2 UN *Klase*=C2 TAD *Risks*=27; JA *Grupa*=2 UN *Klase*=C2 TAD *Risks*=32; JA *Grupa*=2 UN *Klase*=C2 TAD *Risks*=54; JA *Grupa*=2 UN *Klase*=C2 TAD *Risks*=32; …}), kas raksturo klasterizācijas rezultātā iegūto indivīda klasi C_n, pētāmo vielu — *Grupa* un sirds nekrozes risku — *Risks*, izveido riska statistisko sadalījumu, kur riska vērtība 27 ir bijusi sastopama pagaidu nosacījumu likumu kopā vienu reizi. Visiem pagaidu nosacījumu likumiem tika izveidotas līdzīgas izteiksmes, kuras ar piemēru parādītas 3.3. tabulā.

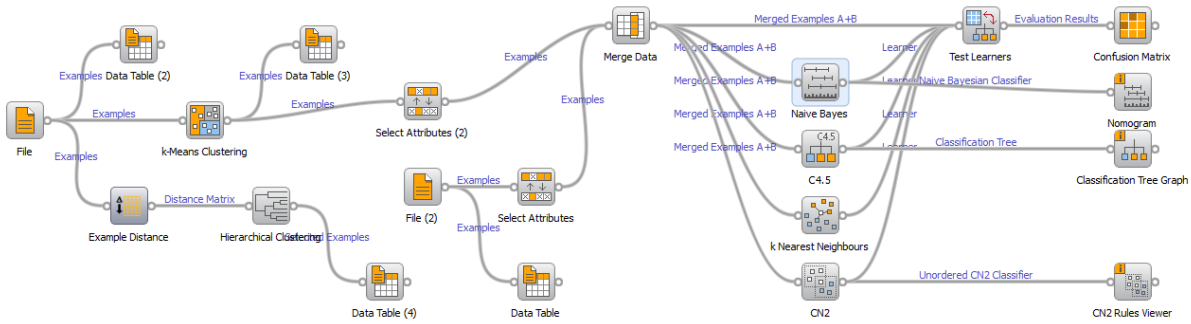
3.3. tabula

Riska parādīšanās biežuma statistika

Riska parādīšanās biežuma skaits (RPBS)	1	2	1	…
Riska vērtība (Risks)	27	32	54	…

3.2.3. SNRPS eksperimenti un iegūtie rezultāti

Eksperimentu izpildīšanai tika izmantota šāda programmatūra: *ADInstruments LabChart 7 View, Microsoft Excel 2010, Weka 3.6.3, Statistica 8.0.* un *Orange Canvas 2.0*, kurā tika izveidots modelis, kas parādīts 3.19. attēlā, ar kura palīdzību izpildīti eksperimenti klasifikācijas un klasterizācijas uzdevumu veikšanai.



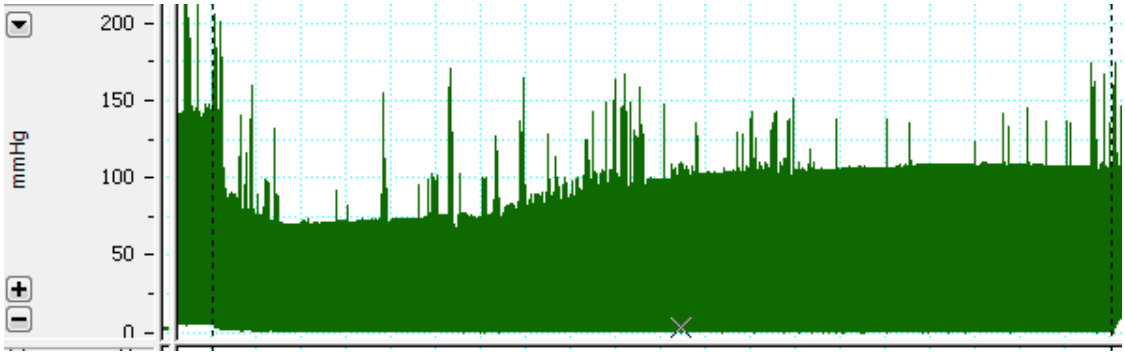
3.19. att. Izveidotais modelis *Orange Canvas* vidē eksperimentu veikšanai

SNRPS eksperimentu norise un izmantotie dati

Eksperimentiem izmantota datu kopa, kas sastāvēja no 92 objektiem, kas raksturoja sirds kontrakcijas spēka vērtības išēmijas stadijā (īsas laika rindas) un šo objektu raksturojošajiem parametriem. Īsu laika rindu garums pēc datu attīrīšanas bija 38 periodi, kas tika nolasīti ar specializētas aparatūras palīdzību, izmantojot programmatūru *LabChart 7 View*. Šī aparatūra ļauj fiksēt sirds kontrakcijas spēka izmaiņas ar intensitāti 0,25 sekundes. Nolasījumi reģistrēti katras 60 sekundes 40 minūšu garumā, atmetot pirmo un pēdējo nolasījumu. Sirds kontrakcijas spēka vērtības katrā laika periodā tiek izteiktas ar dzīvsudraba staba augstumu (mmHg). Iegūtās īsās laika rindas tika normalizētas, pielietojot *z-novērtējuma normalizāciju ar standarta novirzi* un *normalizāciju ar dzīves līkni* pieejas, iegūstot katrai no tām divas dažādas datu kopas, ar kurām eksperimentējot tika noteikts, ka piemērotākā bija *normalizācija ar dzīves līkni*. Izveidotās datu kopas tika klasterizētas ar *k-vidējo sadalošo*, *maksimālās līdzības, aglomeratīvo hierarhisko* un *modificētu k-vidējo sadalošo* algoritmiem. Iegūtie rezultāti tika analizēti un meklēta piemērotākā klasterizācijas pieeja, un eksperimentāli tika noskaidrots, ka tas ir *modificēts k-vidējo sadalošais* algoritmiem. Tāpat katrs objekts tika raksturots ar raksturojošiem parametriem (atribūtiem), tādiem kā: *Grupa(Atr.1.)* — pētāmās vielas tips, ar kuru tika baroti indivīdi; *Svars(Atr.2.*); asins plazmas parametri: *Karnitīns(Atr.3.*), *Triglicerīdi(Atr.4.*), *Taukskābes(Atr.5.*), *Glikoze(Atr.6.*), *Holesterīns(Atr.7.*), *Butirobetaīns(Atr.8.*) un *Risks(Klase)* — sirds nekrozes riska novērtējums, kas iegūts farmakoloģisko pētījumu rezultātā. Raksturojošajos parametros tika meklēta informatīvāko atribūtu kopa, kas būtu pielietojama klasifikatora uzbūvē. Piemērotākais klasifikators tika izvēlēts pēc klasifikācijas precizitātes rezultātiem. Eksperimentāli starp *Naivo Baijesa*, *k-tuvāko kaimiņu*, *C4.5* un *CN2* algoritmiem noteikts, ka piemērotākais ir *C4.5* algoritms. No iegūtajiem klasifikācijas rezultātiem tika veidoti nosacījumu likumi, kas tika uzglabāti datu bāzē. Prognozēšanas procesā tika noteikts sirds nekrozes risks "jaunam" indivīdam.

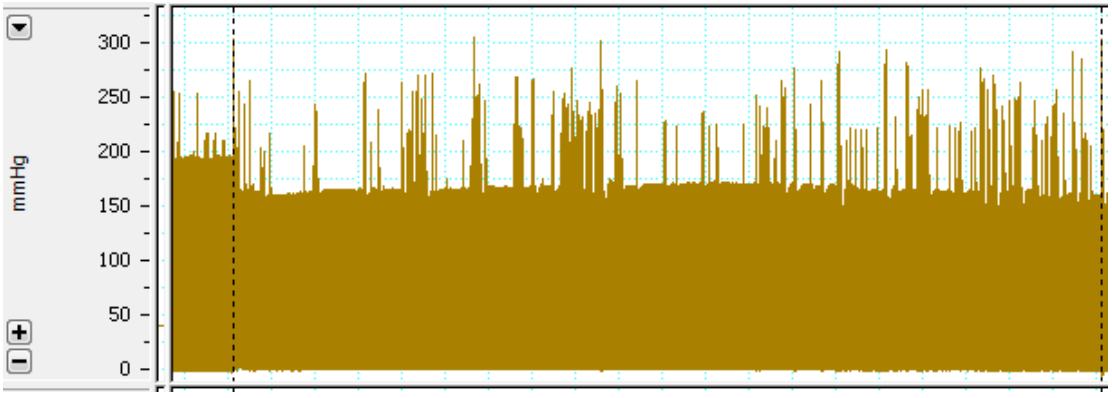
Sirds kontrakcijas spēka datu pirmapstrāde

Pielietojot *LabChart 7 View* programmatūru, katram indivīdam (laboratorijas žurkai) tika atlasīti sirds kontrakcijas spēka dati 40 minūšu garumā no oklūzijas perioda. Sirds kontrakcijas spēka mērījumu grafiki, dažādiem indivīdiem ir parādīti 3.20. un 3.21. attēlos, kur ar raustītu vertikālu līniju ir norādītas mērījumu sākuma un beigu fāzes, skatoties no kreisās puses uz labo. Sākuma fāze norāda uz oklūziju, bet beigu fāze — reperfūziju. Kā redzams no šiem attēliem, tad parādīto indivīdu siržu kontrakcijas spēku izmaiņas ir izteikti atšķirīgas gan sirds kontrakcijas spēka amplitūdas līmeņa ziņā, gan sirds kontrakcijas spēka līknes uzvedības ziņā laika griezumā. Novērojams ir arī dažāds trokšņu (pīķveida) līmenis abiem grafikiem.



3.20. att. Sirds kontrakcijas spēka mērījumu grafiks (84. indivīds)

Pielietotā programmatūra nodrošināja datu nolasīšanu noteiktā laika periodā un trokšņainu (pīķveida) vērtību atmešanu, izmantojot iebūvēto matemātisko analizatoru.



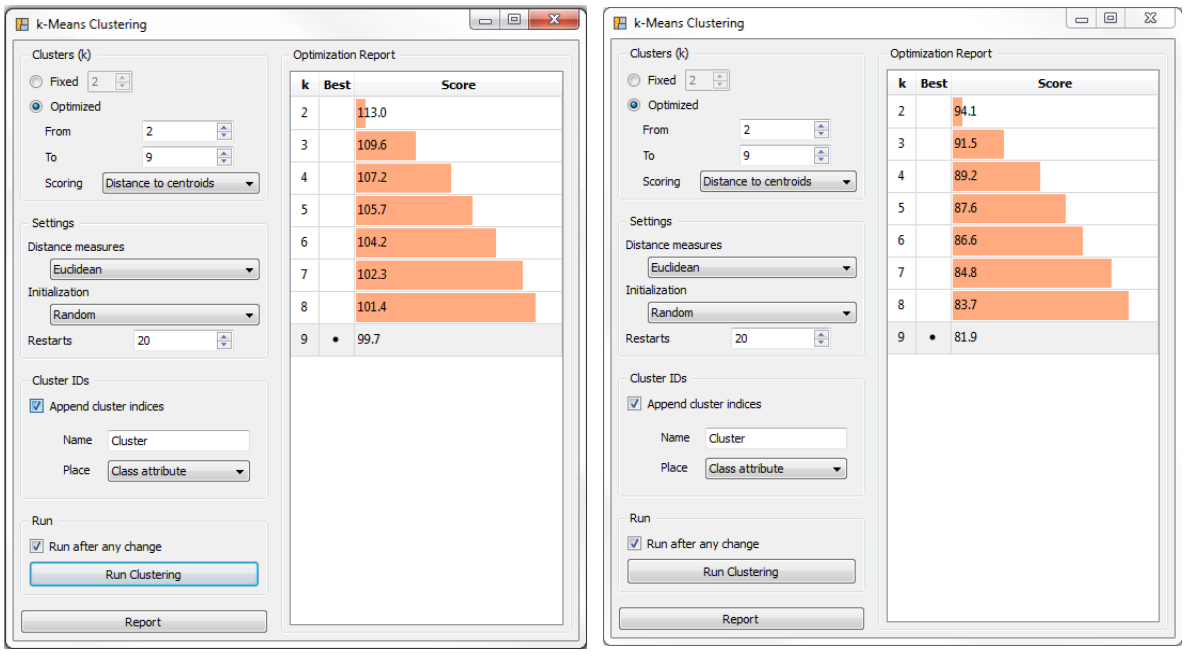
3.21. att. Sirds kontrakcijas spēka mērījumu grafiks (92. indivīds)

Sagatavotie dati tika normalizēti, pielietojot *z-normalizāciju ar standarta novirzi* un *normalizāciju ar dzīves līkni*, iegūstot divas datu kopas ar sirds kontrakcijas spēka mērījumiem, kas tiek raksturoti kā īsas laika rindas. Analizējot abas izveidotās datu kopas, tika noteikta piemērotākā īsu laika rindu normalizācijas pieeja (skatīt nākošo apakšnodaļu).

Sirds kontrakcijas spēka datu klasterizācija

Datu klasterizācijai tika izmantoti *k-vidējo sadalošais*, *maksimālās līdzības*, *aglomeratīvais hierarhiskais* un *modificēts k-vidējo sadalošais* algoritmi, lai noteiktu piemērotāko, kas būtu izmantojams par pamatu sirds nekrozes riska prognozēšanai.

Klasterizējot datu kopas ar *k-vidējo sadalošo* algoritmu, tika izmantots pieņemts klasteru skaita diapazons, kur minimālais klasteru skaits ir 2, bet maksimālais — aprēķināts pēc formulas (2.14) un bija 9. Par objekta attāluma mēru līdz tuvākajam centroīdam, tika lietots *Eiklīda* attālums, kuru aprēķina pēc formulas (2.7). Eksperimentiem tika izmantota programmatūra *Orange Canvas*, kuras iegūtie klasterizācijas rezultāti ir parādīti 3.22. attēlā, pa kreisi datu kopa, kas tika normalizēta, lietojot *z-normalizāciju ar standarta novirzi*, pa labi — normalizāciju ar *dzīves līkni*. Abām datu kopām bija identiski uzstādījumi eksperimentu veikšanai: klasteru skaits no 2 līdz 9; novērtējums (*scoring*) — attālumu līdz centroīdiem (jo mazāka vērtība, jo labāks novērtējums); distances mērs (*distance measures*) — *Eiklīda* attālums; inicializācija — nejaušā veidā; algoritma atkārtojumi (*restarts*) — 20 reizes; klasterizācijas atskaite (*optimization report*)– izvadītas novērtējumu vērtības (*score*) pie klastera numura *k*, labākais rezultāts (*best)* iezīmēts ar punktu. Eksperimenti ar katru no datu kopām tika atkārtoti 20 reizes katram klasteru skaitam, ik reizi centroīdus izvēloties pēc nejaušības principa.



3.22. att. Klasterizācijas rezultāti ar *k-vidējo sadalošo* algoritmu programmā *OrangeCanvas*

Līdzīgi eksperimenti ar *k-vidējo sadalošo* algoritmu tika veikti programmā *Weka*, kas klasterizācijas rezultātu salīdzināšanai lieto summētās kvadrātiskās kļūdas aprēķinu. Iegūtie klasterizācijas rezultāti tika novērtēti pēc summētās kvadrātiskās kļūdas, kas parādīta 3.4. tabulā, no kuras redzams, ka precīzākos rezultātus (85,79) uzrādījusi pieeja, normalizējot datus ar *dzīves līkni*. Abām pieejām mazākā kļūda sasniegta pie 9 klasteriem, kas norāda uz klasterizācijas algoritma nepiemērotību īsu laika rindu klasterizācijai, jo palielinoties klasteru skaitam, samazinās kļūda.

3.4. tabula

K-vidējo sadalošā algoritma summētā kvadrātiskā kļūda

	Klasteru skaits							
Normalizācijas pieeja	2	3	4	5	6	7	8	9
Z-novērtējuma normalizācija ar standarta novirzi	141,67	138,42	128,36	125,89	122,64	120,56	116,34	**113,92**
Normalizācija ar dzīves līkni	115,73	112,99	101,81	96,36	92,54	89,44	87,39	**85,79**

Eksperimenti tika turpināti programmā *Statistica 8* ar *Maksimālās līdzības* algoritmu, kas klasterizācijas rezultātu salīdzināšanai lieto logaritmisko ticamību. Klasterizācijas procesā pirmajai datu kopai, pielietojot *z-novērtējuma normalizāciju ar standarta novirzi* noteica piemērotāko klasteru skatu — 9, bet otrajai datu kopai, pielietojot *normalizāciju ar dzīves līkni* noteica kā piemērotākos — 8 klasterus pēc logaritmiskās ticamības, rezultāti parādīti 3.5. tabulā.

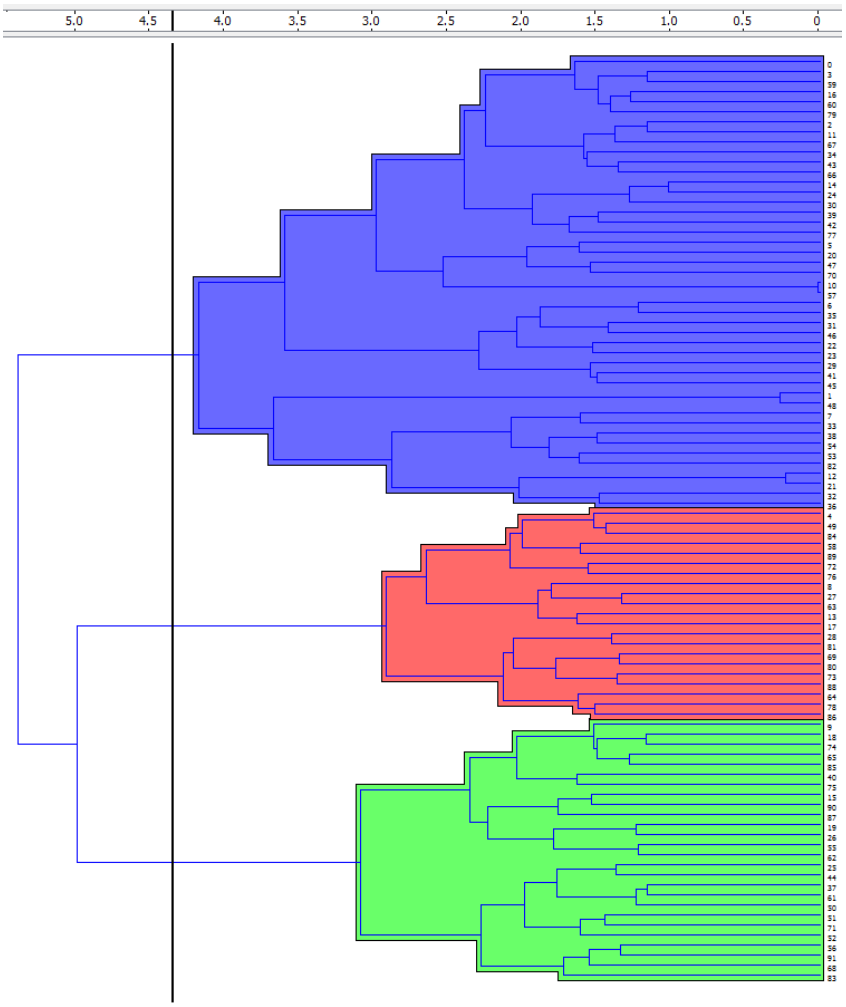
3.5. tabula

Maksimālās līdzības algoritma logaritmiskā ticamība

	Klasteru skaits							
Normalizācijas pieeja	2	3	4	5	6	7	8	9
Z-novērtējuma normalizācija ar standarta novirzi	-49,64	-48,64	-47,93	-47,39	-46,36	-45,47	-45,25	**-44,13**
Normalizācija ar dzīves līkni	144,49	145,40	148,42	149,06	150,06	150,32	**151,13**	150,87

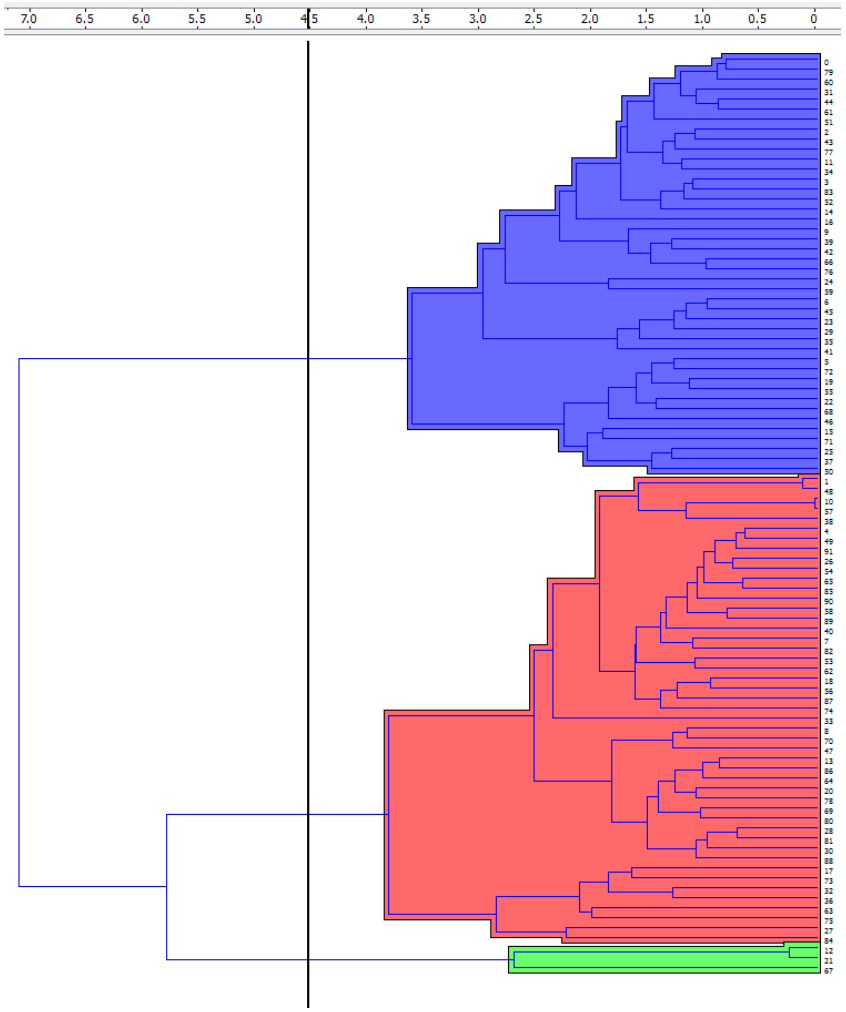
Kā parāda rezultāti, tad arī šis klasterizācijas algoritms nespēj noteikt piemērotāko klasteru skaitu, jo, pieaugot to skaitam palielinās logaritmiskā ticamība, kas rada šaubas par algoritma robustumu.

Aglomeratīvais hierarhiskais algoritms ar abām normalizācijas pieejām kā piemērotākos noteica 3 klasterus, izmantojot klasteru apvienošanas metodi *Ward's linkage* [69]. *Varda* metode lieto objektu attālumu kvadrātu summas pieaugumu starp klasteriem un to centroīdiem, kuru iegūst klasteru apvienojuma rezultātā. Izmantojot *z-novērtējuma normalizāciju ar standarta novirzi*, kur algoritma apstāšanās kritērijs ir attālums no 5 līdz 4,2, iezīmēts ar melnu vertikālu nepārtrauktu līniju, parādīts 3.23. attēlā. Apstāšanās kritēriju nosaka lielākais attālums datu kopā starp objektiem vai to grupām. Kā redzams no šī paša attēla, tad ir noteikti 3 klasteri, kas ir iezīmēti dažādās krāsās. Grafikā *x* ass norāda attāluma vērtību, bet *y* ass — objektu numurus. Zilā līnija grafikā norāda, kuri objekti vai objektu grupas tiek apvienotas.



3.23. att. Datu kopas klasterizācija ar *aglomeratīvo hierarhisko* algoritmu, izmantojot *z-novērtējuma normalizāciju ar standarta novirzi*

Savukārt, otra pieeja, kas izmanto *normalizāciju ar dzīves līkni*, kur algoritma apstāšanās kritērijs ir attālums no 5,8 līdz 3,8, iezīmēts ar melnu vertikālu nepārtrauktu līniju, parādīts 3.24. attēlā. Pie šī sadalījuma arī ir noteikti 3 klasteri, bet trešajā klasterī ir apvienoti tikai 3 objekti. Un arī šī algoritma iegūtie rezultāti ir pretrunīgi vērtējami. Rezultāti nedod priekšroku nevienai no normalizācijas pieejām un arī objektu piederība klasteriem ir pretrunīga.



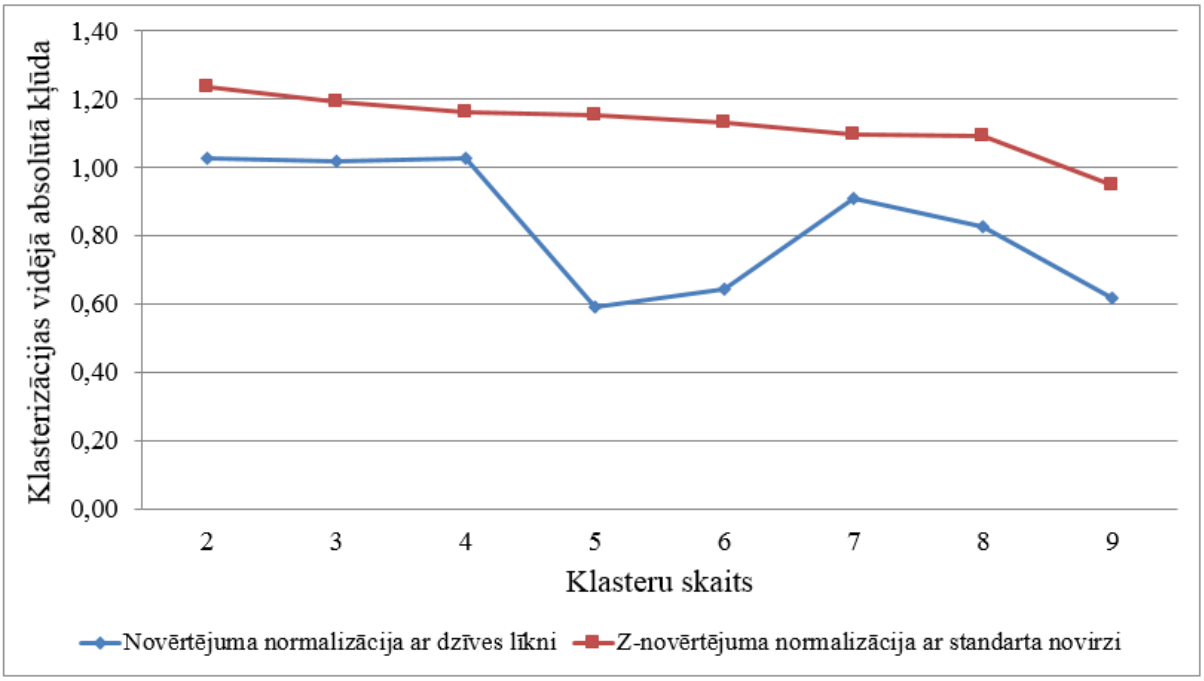
3.24. att. Datu kopas klasterizācija ar *aglomeratīvo hierarhisko* algoritmu, izmantojot *normalizāciju ar dzīves līkni*

Tāpēc tiek pielietots vēl viens klasterizācijas algoritms — *modificēts k-vidējo sadalošais* algoritms. Ar šo algoritmu katram klasterim aprēķina absolūto kļūdu pēc formulas (2.15) un vidējo absolūto kļūdu pēc formulas (2.16), iegūtie rezultāti atspoguļoti 3.6. tabulā. Balstoties uz vidējās absolūtās kļūdas novērtējumu, tiek noteikta piemērotākā normalizācijas pieeja – *normalizācija ar dzīves līkni* un piemērotākais klasteru skaits — 5, kā parādīts 3.25. attēlā.

3.6. tabula

Modificēta k-vidējo sadaloša algoritma vidējā absolūtā kļūda

	Klasteru skaits							
Normalizācijas pieeja	2	3	4	5	6	7	8	9
Z-novērtējuma normalizācija ar standarta novirzi	1,233	1,192	1,158	1,150	1,128	1,095	1,093	0,946
Normalizācija ar dzīves līkni	1,026	1,017	1,026	**0,59**	0,644	0,91	0,824	0,615



3.25. att. Klasterizācijā ar *modificētu k-vidējo* algoritmu noteiktā vidējā absolūtā kļūda

Indivīda raksturojošo parametru informatīvuma noteikšana

Datu klasifikācija ir atkarīga no atribūtu informatīvuma. Ja starp atribūtiem pastāv korelācija, tad uzskata, ka atribūti savstarpēji ir saistīti [70]. Savukārt, ja atribūti nekorelē, tas nozīmē, ka starp šiem atribūtiem ir vāja saite un tiem nav nozīmes pielietot klasifikāciju. Tāpēc eksperimentāli pirms klasifikācijas procesa uzsākšanas, ir jāveic atribūtu informatīvuma noteikšana vai to sadalīšana pēc atribūtu svarīguma (agregācija). Tā kā programmatūra *Weka* piedāvā plašas atribūtu atlases iespējas, tad pētījumos tika izmantota *CfsSubsetEval* atribūtu novērtējuma un *BestFirst* pārmeklēšanas metodes, kas tiek bieži lietotas datu iegūšanā atribūtu informatīvuma noteikšanai [70]. Tika iegūti šādi rezultāti:

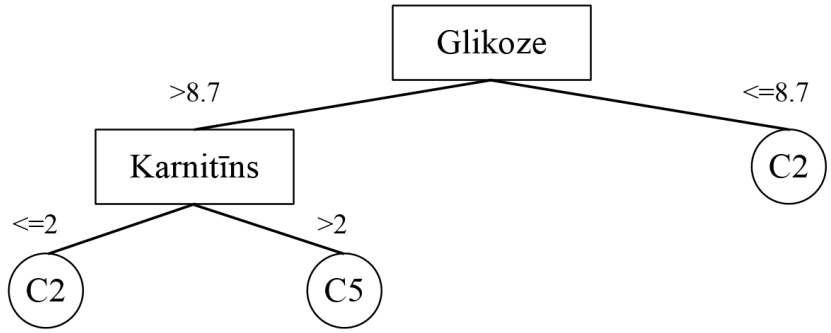
Taukskābes, *Holesterīns, Karnitīns* uzrādīja 100 %, *Glikoze* 90 %, *Svars* 80 %, un *Butirobetaīns* 20 % korelāciju, bet *Triglicerīdi* un *Grupa* 0 %. Pēdējos divus parametrus pēc loģikas vajadzēja atmest, bet parametrs *Grupa* satur ļoti svarīgu informāciju par pētāmās vielas veidiem, kas ir šī pētījuma pamatā, tad šis parametrs tika saglabāts. Tad tika pieņemts lēmums sadalīt pilno datu kopu apakškopās, balstoties pēc parametra *Grupa* vērtībām, iegūstot piecas apakškopas. Šāda datu kopas sadalīšana, ļauj atbrīvoties no neinformatīva atribūta izmantošanas klasifikatora apmācības procesā un tajā pat laika patur svarīgu informāciju par pētāmo vielu, kas tālāk tiks izmantota nosacījumu likumu veidošanai. Rezultātā seši atlasītie laboratorijas dzīvnieka raksturojošie atribūti tālāk tika izmantoti klasifikācijas uzdevumā.

SNRPS piemērotākā klasifikatora noteikšana

Piemērotākā klasifikatora izvēlei tika veikts eksperimentu kopums, izmantojot pirmapstrādāto datu kopu ar informatīvākajiem atribūtiem, pievienojot katram objektam datu kopā klasterizācijā noteikto klasi. Tika izveidots modelis, kurā iespējams mainīt klasteru skaitu, veicot klasterizāciju ar *k-vidējo sadalošo* un *modificētu k-vidējo sadalošo* algoritmiem. Klasifikācijas procesā tika lietoti *Naivais Baijesa, k-tuvāko kaimiņu, C4.5, CN2* algoritmi, kas izvēlēti salīdzinošās analīzes rezultātā (skatīt 1.5. apakšpunktu) un 10-kārtu šķērsvalidācija [43]. Pielietojot izveidoto modeli, tika noteikts piemērotākais klasifikācijas algoritms sistēmas izveidei, pamatojoties uz klasifikācijas precizitātes novērtējumu visā klasteru diapazonā. Kā rāda klasifikācijas precizitātes novērtējums, kas parādīts 3.7. tabulā, labākie rezultāti ir sasniegti ar induktīvo lēmumu koku algoritmu *C4.5*, tāpēc šis algoritms tika izmantots turpmāko eksperimentu gaitā un kļuva par pamatu SNRPS izveidē. Tāpat arī jāatzīmē, ka induktīvie lēmumu koki ir viegli interpretējami iegūto rezultātu atspoguļošanai, jo uz tiem var uzskatāmi veikt "jaunā" indivīda raksturojošo parametru klasifikāciju. Klasifikācijā iegūtais rezultāts 1. apakškopai ir parādīts 3.26. attēlā, kur, lietojot *C4.5* algoritmu, uzbūvēts induktīvais lēmuma koks no 1. apakškopas 18 objektiem. Pārējos klasifikatora izveidotos induktīvos lēmuma kokus citām apakškopām skatīt 2. pielikumā.

SNRPS precizitātes novērtējums

Izveidotās SNRPS precizitāte tiek pārbaudīta, izmantojot *Naivo Baijesa, k-tuvāko kaimiņu, C4.5, CN2* algoritmus, ar katru no izveidotajām apakškopām, pielietojot *10-kārtu šķērsvalidāciju*, kuras pielietošana palīdz izvairīties no klasifikatora pielāgošanās apmācības kopas datiem.



3.26. att. Klasifikācijas rezultātā iegūtais induktīvais lēmuma koks 1. apakškopai

Iegūtie klasifikatora precizitātes novērtējuma rezultāti, parādīti 3.7. tabulā, no kuras redzams, ka piedāvātā SNRPS sasniedz visaugstāko precizitāti, izmantojot *C4.5* algoritmu, vadoties pēc aprēķinātās vidējās vērtības starp visām apakškopām. Lai pārliecinātos par apakškopu izveides precizitāti, tika veikti eksperimenti ar piemērotāko klasifikatoru nesadalītai datu kopai. Kā redzams no 3.7. tabulas, tad nesadalītas datu kopas precizitāte ir zemāka kā vidējā vērtība sadalītām apakškopām, kas norāda, ka izvēlētā datu kopas sadalīšanas pieeja ļauj palielināt klasifikatora precizitāti un ir pielietojama SNRP sistēmā. Eksperimenti parādīja, ka datu kopas sadalīšana apakškopās palīdz uzlabot klasifikācijas precizitāti par 0,1. Klasifikācijas precizitātes starpība starp *C4.5* un *CN2* algoritmiem, protams, ir niecīga, tomēr precizitātes amplitūda starp apakškopām *C4.5* algoritmam ir mazāka, tāpēc par piemērotāko SNRPS izstrādāšanā tika izvēlēts *C4.5* algoritms.

3.7. tabula

Ar klasifikācijas algoritmiem, lietojot *modificētā k-vidējo sadalošā* algoritma atrasto piemērotāko klašu skaitu, noteiktā klasifikatoru precizitāte

	Klasifikators			
	Naivais Baijesa	*C4.5*	*k-tuvāko kaimiņu*	*CN2*
1. apakškopa	0,45	0,7	0,65	0,6
2. apakškopa	0,25	0,3	0,2	0,15
3. apakškopa	0,25	0,6	0,4	0,5
4. apakškopa	0,45	0,55	0,3	0,75
5. apakškopa	0,2	0,25	0,5	0,3
Vidējā vērtība	0,32	**0,48**	0,41	0,46
Nesadalīta datu kopa		0,38		

Iegūto rezultātu salīdzināšanai tika veikti papildus eksperimenti, aizstājot *modificētu k-vidējo sadalošo* algoritmu ar *k-vidējo sadalošo* algoritmu, kā redzams no rezultātiem, kas parādīti 3.8. tabulā, tad iegūtā klasifikatoru precizitāte ir samazinājusies par 0,21, salīdzinot ar iepriekšējiem rezultātiem 3.7. tabulā, kur klasterizācijai tika izmantots *modificēts k-vidējo sadalošais* algoritms. Nesadalītas datu kopas precizitāte šajā gadījumā ir augstāka kā vidējā vērtība sadalītām apakškopām. No iegūtajiem rezultātiem secināms, ka īsu laika rindu klasterizācijai piemērotākais ir *modificēts k-vidējo sadalošais* algoritms, bet klasifikācijai — *C4.5* algoritms.

3.8. tabula

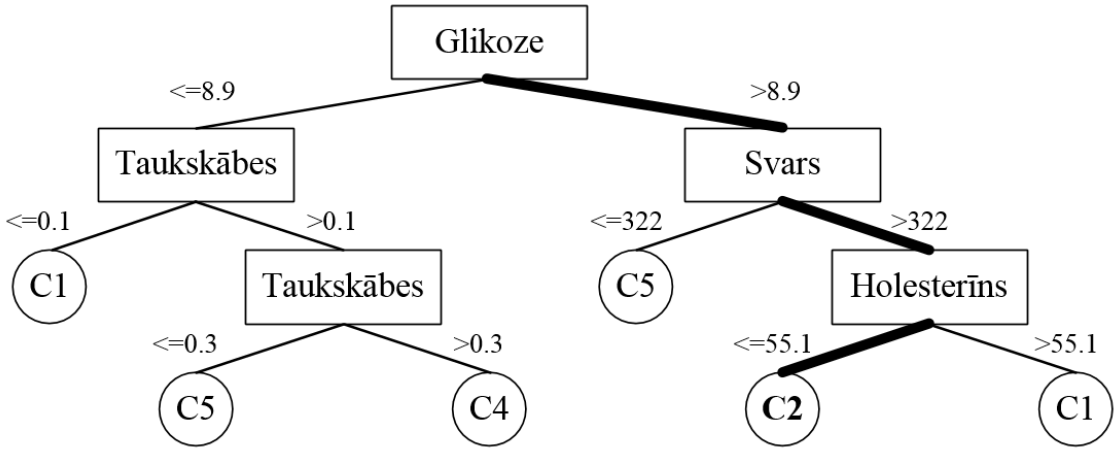
Ar klasifikācijas algoritmiem, lietojot *k-vidējo sadalošo* algoritmu ar iepriekš atrasto piemērotāko klašu skaitu, noteiktā klasifikatoru precizitāte

	Klasifikators			
	Naivais Baijesa	*C4.5*	*k-tuvāko kaimiņu*	*CN2*
1. apakškopa	0,35	0,45	0,4	0,1
2. apakškopa	0	0,1	0,05	0,05
3. apakškopa	0,45	0,35	0,4	0,2
4. apakškopa	0,05	0,05	0,2	0,05
5. apakškopa	0,27	0,4	0,25	0,25
Vidējā vērtība	0,224	0,27	0,26	0,13
Nesadalīta datu kopa		0,3		

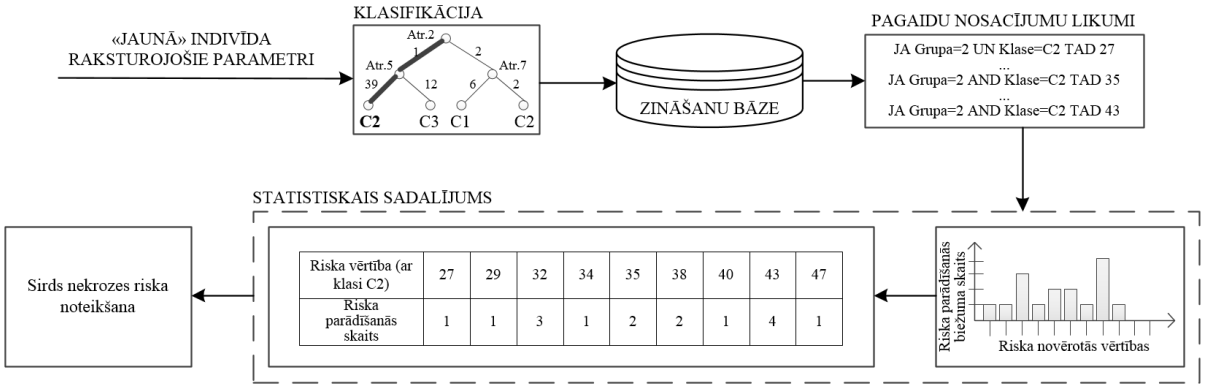
Sirds nekrozes riska prognozēšana

Sirds nekrozes riska prognoze tiek veikta, pamatojoties uz klasifikācijas procesā iegūtā lēmuma koka un "jaunā" indivīda raksturojošajiem parametriem. Lai prognozētu iespējamo sirds nekrozes risku "jaunajam" indivīdam, kura raksturojošie parametri ir šādi: *Grupa=2*; *Glikoze=9,4*; *Taukskābes=0,2*; *Svars=334*; *Holesterīns=53*; *Butirobetaīns=1,1*, tiek izvēlēts klasifikators pēc atribūta *Grupa* vērtības, kura ir 2. Uz otrā klasifikatora, kas noteikts apmācības procesā, tiek projicēti pārējie raksturojošie parametri (izņemot atribūtu *Grupa*), "uzbūvētais" lēmuma koks parādīts 3.27. attēlā. Virzoties secīgi pa līmeņiem no koka saknes uz leju, tiek sasniegta lēmuma koka lapa, kas norāda uz iegūto rezultātu — prognozes klasi. Koka zara vērtības tiek salīdzinātas ar analizējamā indivīda atribūta vērtību, nosakot virzības ceļu, kas 3.27. attēlā iezīmēts ar treknrakstu, līdz tiek sasniegta viena no induktīvā lēmuma

koka lapām, kas norāda uz klasi (C2). Tad pēc "jaunajā" indivīda atribūta vērtības (*Grupa*=2) un klasifikācijā noteiktās klases (C2), no zināšanu bāzes izgūst pagaidu nosacījumu likumus ar *Riska* vērtībām (JA *Grupa*=2 un *Klase*=C2 TAD *Risks*=27), kuri pārklāj minētos nosacījumus. No izgūtajiem pagaidu nosacījumu likumiem izveido riska statistisko sadalījumu. No iegūtā riska statistiskā sadalījuma tiek noteikts iespējamais sirds nekrozes risks "jaunam" indivīdam. Pilns "jaunā" indivīda riska prognozēšanas process ir parādīts 3.28. attēlā.



3.27. att. "Jaunā" indivīda klasifikācijas process (raksturojošo parametru projekcija) uz apmācības procesā izveidotā induktīvā lēmuma koka 2. apakškopai



3.28. att. Sirds nekrozes riska prognozēšanas process "jaunam" indivīdam

Iespējamo risku var noteikt vairākos veidos: viens — ar matemātisko cerību, nosakot visiespējamāko risku pēc tā statistiskā sadalījuma, otrais — nosakot riska vērtību pēc parādīšanās biežuma attāluma novērtējuma un starpības aprēķina starp analizējamo un minimālo vērtību.

Nosakot visiespējamāko risku, matemātiskā cerība [77] tiek aprēķināta šādi: ņemot iegūto riska statistisko sadalījumu, pēc kura katra riska vērtībai atkarībā no tās paradīšanās

biežuma (atlasītajos pagaidu nosacījumu likumos) tiek piešķita varbūtība, kas parādīta 3.9. tabulā, kura tiek sareizināta ar riska vērtību. Sirds nekrozes riska vērtība "jaunajam" indivīdam tiek noteikta, summējot visas iegūtās matemātiskākas cerības vērtības, rezultātu noapaļojot līdz veselai vērtībai. Iegūtais rezultāts norāda uz visiespējamāko riska vērtību 37.

3.9. tabula

Matemātiskās cerības aprēķins

	Aprēķini									Summa (noapaļota)
Riska parādīšanās biežuma skaits	1	1	3	1	2	2	1	4	1	
Riska vērtības	27	29	32	34	35	38	40	43	47	
Varbūtība	0,06	0,06	0,19	0,06	0,13	0,13	0,06	0,25	0,06	
Matemātiskā cerība	1,62	1,74	6,08	2,04	4,55	4,94	2,40	10,8	2,82	**37**

Otra pieeja piedāvā iespējamās vērtības noteikt pēc riska parādīšanās biežuma, lietojot darba autora piedāvātu attālumu metriku, sākotnēji aprēķinot attālumu pēc moduļa starp riska vērtībām (|27-29|=2; |29-32|=3; …, |43-47|=4), iegūtie rezultāti tiek reģistrēti laukā *Attālums starp riska vērtībām*, kas parādīti 3.10. tabulā.

Tālāk tiek aprēķināta starpība starp riska parādīšanās biežuma skaita (RPBS) minimālo vērtību 1 un riska parādīšanās biežuma skaitu, rezultātus reģistrējot laukā *Starpība starp RPBS min un RPBS vērtībām*. Tālāk tiek summētas lauku *Attālums starp riska vērtībām* un *Starpība starp RPBS min un RPBS vērtībām* vērtības un izvēlas tos laukus, kuriem summētā vērtība bijusi vismazākā (piemēra gadījumā summa ir 0, kas iekrāsota ar treknrakstu 3.10. tabulā). Iegūtais rezultāts — noteikta sirds nekrozes riska vērtība vai vērtības (35 un/vai 43) šim jaunajam indivīdam — laboratorijas dzīvniekam. Lietojot pieeju ar matemātiskās cerības aprēķinu, tiek iegūta konkrēta riska prognozes vērtība, bet, lietojot uz attālumu mēru balstītu pieeju, var iegūt novērtējuma vērtību intervālu (35; 43). Intervāls tiek veidots uz zināšanu bāzē esošo vērtību pamata. Rezultātā iegūstot intervālu (35; 43), tas tiek interpretēts, aizstājot to ar vidējo vērtību 39. Piemērotākā ir uz attālumiem izveidotā pieeja, jo pie lielas statistiskā sadalījuma izkliedes, šī pieeja uzrādīs precīzākus rezultātus.

3.10. tabula

Iespējamā sirds nekrozes riska aprēķins pēc riska parādīšanās biežuma attāluma novērtējuma

	Aprēķini								
Riska parādīšanās biežuma skaits (RPBS)	1	1	3	1	2	2	1	4	1
Riska vērtības	27	29	32	34	35	38	40	43	47
Attālums starp riska vērtībām	-	2	3	2	1	3	2	3	4
Starpība starp RPBS min un RPBS vērtībām	0	0	-2	0	-1	-1	0	-3	0
Summa: **Attālums** + **starpība**		2	1	2	**0**	2	2	**0**	4
Novērtējums					35			43	
Novērtējuma vidējā vērtība	**39**								

3.2.4. Kopsavilkums un secinājumi par SNRPS

Rezultātā izveidota SNRPS [34], ar kuras palīdzību iespējams prognozēt "jaunā" indivīda sirds nekrozes riska vērtību. Iegūtais prognozes rezultāts norāda uz sirds nekrozes risku indivīdam, kas barots ar noteiktu pētāmās vielas veidu, neveicot ķirurģisku iejaukšanos. Eksperimenti, lietojot klasterizāciju un klasifikāciju parāda, ka izveidotā SNRPS spēj apstrādāt atšķirīgas datu struktūras — īsas laka rindas un indivīda raksturojošos parametrus. Iegūtie rezultāti ir viegli interpretējami sistēmas lietotājam.

Eksperimentāli tika noskaidrots, ka, normalizējot sirds kontrakcijas spēka datus, labākus rezultātus parādīja pieeja — *normalizācija ar dzīves līkni*, kas skaidrojams ar to, ka *z-novērtējuma normalizācija ar standarta novirzi* vērtības normalizē negatīvā un pozitīvā vērtību diapazonā, bet *normalizācija ar dzīves līkni* — tikai pozitīvā vērtību diapazonā, uz ko norādīja šo pieeju vidējās absolūtās kļūdas novērtējums, klasterizējot datu kopu ar *modificētu k-vidējo* algoritmu.

Klasterizācijas algoritmi uzrādīja atšķirīgus rezultātus. *Maksimālās līdzības* algoritms, klasterizējot abas datu kopas, parādīja vienotu tendenci — palielinoties klasteru skaitam, pieaug logaritmiskā ticamība. Iegūtie rezultāti norāda, ka *maksimālās līdzības* algoritms nav piemērots īsu laika rindu apstrādei.

Klasiskais *k-vidējo sadalošais* algoritms arī uzrādīja vienotu tendenci kļūdu novērtēšanas rezultātos — pieaugot klasteru skaitam, samazinās klasterizācijas kopējā kļūda. Tika izmantotas dažādas attāluma mēra metrikas, bet uzlabojumu sasniegto rezultātu ziņā nebija. Visas izmantotās metrikas uzrādīja vienotu tendenci. No iegūtajiem rezultātiem secināms, ka arī *k-vidējo sadalošais* algoritms nav īsti piemērots īsu laika rindu apstrādei.

Aglomeratīvais hierarhiskais algoritms, izmantojot dažādas normalizācijas pieejas, uzrādīja līdzīgas tendences, nosakot vienādu klasteru skaitu. Vizuāli aplūkojot iegūtās dendrogrammas, tika secināts, ka klasterizācijas rezultātiem piemīt tikai informatīvs raksturs un iegūtie rezultāti nav pielietojami izstrādājamajā sistēmā.

Modificētais k-vidējo sadalošais algoritms pēc iegūto rezultātu un vidējās absolūtās kļūdas aprēķina ir piemērotākais starp analizētajiem klasterizācijas algoritmiem. Šis algoritms, izmantojot *pieprasījuma normalizāciju ar dzīves līkni*, uzrādīja labākos rezultātus, vadoties pēc vidējās absolūtās klasterizācijas kļūdas novērtējuma. Tāpēc šis algoritms tika izvēlēts par pamatu sirds nekrozes riska prognozēšanas sistēmas izveidei.

No iegūtajiem eksperimentu rezultātiem var apgalvot, ka izveidotā SNRPS, kas saistīta ar sirds mazspējas diagnosticēšanu, ir efektīva, jo palīdz ātrāk un efektīvāk iegūt farmakoloģisko pētījumu rezultātus, ekonomējot eksperimentu laiku un izmantoto laboratorijas dzīvnieku skaitu.

Eksperimenti parādīja, ka izveidotā sistēma nodrošina sirds nekrozes riska prognozēšanu, tikai uz sistēmā ievadītajiem "jaunā" laboratorijas dzīvnieka raksturojošiem parametriem. Starp dzīvnieka raksturojošajiem parametriem, kurus raksturo astoņi atribūti, tika veikta informatīvuma noteikšana. Rezultātā tika atlasīti seši atribūti, kas tika izmantoti turpmākajos eksperimentos. Bet viens atribūts *Grupa*, kas tika izslēgts no datu kopas, kļuva par apakškopu sadalīšanas kritēriju (pētāmās vielas veids), jo atribūts satur svarīgu pētījuma informāciju.

Eksperimentāli tika noteikts piemērotākais klasifikators apakškopu klasifikācijai, kas bija *C4.5* algoritms, kura iegūtie rezultāti ir viegli interpretējami lietotājiem, jo lieto induktīvos lēmumu kokus. Izveidotās SNRPS precizitāte tika noteikta, pielietojot 10-kārtu šķērsvalidāciju, jo pie neliela objektu skaita nav lietderīgi datu kopu sadalīt apmācības un testēšanas kopās, jo tas vēl vairāk samazinās apmācības kopas izmēru.

Darbā netika izpildīta iegūto rezultātu salīdzinošā analīze ar līdzīgām sistēmām, jo tādas nebija atrodamas analizētajā zinātniskajā literatūrā.

3.3. Baktēriju proliferācijas sindroma noteikšanas sistēma

Cilvēka gremošanas sistēmā ir ideāli apstākļi dažādu baktēriju eksistencei. Cilvēks ir simbiotiska būtne, kas savu eksistenci var nodrošināt tikai sadarbojoties ar šīm baktērijām. Gremošanas traktā mītošo baktēriju sugu dažādība ir ļoti liela. Šo mazo mikroorganismu kopējā masa cilvēka ķermenī ir ievērojama — 2 līdz 4 kilogrami. Gremošanas traktā mītošo baktēriju aptuvenais kopējais skaits ir ap 10^{14}. Šis baktēriju daudzums apmēram 10 reizes

pārsniedz cilvēka organisma šūnu skaitu.

Baktēriju proliferācija tievajā zarnā ir šīs zarnas kolonizācija ar resnās zarnas mikroorganismiem, kas var radīt plašu klīnisko spektru, sākot no viegliem un nekonkrētiem simptomiem, līdz smagiem gremošanas traucējumiem. Par baktēriju proliferāciju tievajā zarnā tās sākumstadijā, kad baktērijas pārvietojas no resnās zarnas uz tievās zarnas sākumdaļu, sūdzības parasti ir tikai indivīdiem ar hroniskām gremošanas trakta slimībām. Simptomi, ko rada baktēriju proliferācija tievajā zarnā, ir ļoti netipiski. Šie slimnieki sūdzas par zarnu diskomfortu, pastiprinātu gāzu veidošanos (meteorisms), nekārtīgu vēdera izeju. Medicīnā tiek lietota virkne testu, lai noteiktu pārmērīgu baktēriju augšanas sindromu tievajās zarnās (*Small intestine bacterial overgrowth* (*SIBO*)). Šis sindroms ir sastopams biežāk nekā tiek diagnosticēts. Sindroma attīstību var izraisīt kuņģa skābes barjeras traucējumi, kā arī tievo zarnu disbakteriozes attīstība pēc antibiotiku lietošanas. Bez tam zinātniskās publikācijas, kas norāda uz šī sindroma klātesamību indivīdiem, kuriem tika diagnosticēts kairinātu zarnu sindroms [56], kā arī ir pētījums, kas norāda uz tiešu *SIBO* ietekmi karinātu zarnu sindromā [50].

Baktēriju proliferācijas sindroma (BPS) noteikšanai diagnostikas testi tiek iedalīti invazīvajos un neinvazīvajos testos. Invazīvajos testos tiek veikta pacienta intubācija un zarnu sulas aspirācija no tievās zarnas, bet neinvazīvie testi nosaka baktēriju vielmaiņas produkta koncentrāciju plazmā, urīnā vai izelpā. Tukšās zarnas aspirāta uzsējums ir uzskatāms par "zelta standarta" diagnostikas testu. To iegūst, izmantojot pacienta intubāciju un zarnas aspirāciju dažādās zarnu vietās. Šķidruma daudzums, savākšanas vieta un mikrobioloģisko testu tehniskās detaļas, kā arī izslēgšanas vērtības *BPS* noteikšanai vēl nav standartizētas [20], tāpēc šobrīd klīniskajā praksē tiek izmantoti glikozes (*GET*) un laktulozes (*LET*) elpas testi. Klīniski *BPS* diagnostikas algoritms nosaka, ka indivīdam sākotnēji tiek veikts *GET* tests. Ja tas ir izrādījies pozitīvs, tad tiek veikts *LET* tests. Tikai abu pozitīvu rezultātu gadījumā indivīdam diagnosticē *BPS*. Elpas testos izmanto noteiktu ūdeņraža koncentrāciju izelpas gaisā. Ūdeņradi un metānu veido tievo zarnu piesārņojošās baktērijas, fermentējot intraluminālus substrātus (ogļhidrātus). Visbiežāk izmantojamie H_2-ET substrāti *BPS* diagnozē ir glikoze un laktuloze. Pirmais ir monosaharīds, kas pilnīgi uzsūcas proksimālajā tievajā zarnā, otrais ir slikti absorbēts disaharīds, kas sasniedz aklo zarnu. Abus substrātus fermentē piesārņojošo baktēriju flora tievajā zarnā ar ūdeņraža izstrādi. Indivīdiem, kas cieš no *BPS*, glikozes elpas tests (GET) parasti uzrāda vienu „agro" maksimālo ūdeņraža izdalīšanos. Vismaz 6 stundas pirms testa veikšanas indivīdam ir jāatturas no ēdienreizes. Vismaz 30 minūtes pirms substrāta uzņemšanas indivīdam ir jāatturas no smēķēšanas un

fiziskas slodzes. Alveolārais gaiss, t.i., pēdējā daļa no vienas izelpas (aptuveni 150 ml), tiek lietota mērījumam. *GET* procedūra: 75 g glikozes, kas ir izšķīdināta 400 ml ūdens, ievada orāli [46]. Daudzumu pagatavo atbilstoši attiecībai 1 g glikozes/1 kg ķermeņa svara. Izelpas paraugi tiek ievākti nekavējoties pirms procedūras, kā arī katras 20 minūtes pēc substrāta ieņemšanas 3 stundu laikā. Ja H_2 koncentrācija palielinās vairāk kā $\Delta = 20$ Pm, tad pieņem, ka tests ir pozitīvs un ir apstiprināts *BPS* tievajā zarnā.

Mērījumu novērtēšanā lieto H_2 koncentrāciju atkarībā no laika (salīdzinājumā ar H_2 koncentrāciju ieelpotajā gaisā). Parasti tā sauktais nogrieztais lielums Δ, t.i. starpība starp maksimālo H_2 koncentrāciju elpā noteiktā laikā pēc substrāta ievadīšanas un H_2 koncentrāciju tieši pirms substrāta ievadīšanas, tiek izmantota kā diagnozes kritērijs. Ja šī starpība pārsniedz kritisko vērtību Δ^*, ir jāpieņem noteiktas metaboliskas anomālijas iespējamība. Tā kā kuņģī jebkura substrāta atrašanās laiks izplešas ļoti plašā intervālā pat normāliem cilvēkiem, tad parasti H_2 veidošanās nesākas brīdī, kad substrāts jau ir izgājis cauri kuņģim un divpadsmitpirkstu zarnai. Elpas testu mērījumi nedrīkst tikt izbeigti pirms H_2 koncentrācija elpā nav sasniegusi maksimumu.

Tā kā *GET* un *LET* testi ir laikietilpīgi, tad tie prasa ilgu pirmssagatavošanās procesu, kas parasti indivīdiem izraisa negatīvu attieksmi pret šo procedūru. Tāpēc mērķis ir piedāvāt alternatīvas pieejas *BPS* noteikšanai tievajā zarnā, izmantojot datu iegūšanas metodes un algoritmus. No izmeklējumu rezultātiem, veicot *GET* testus un atbildot uz anketēšanas jautājumiem par indivīda pašsajūtas novērtējumu, tiek piedāvāta sistēma, kas nosaka *BPS* tievajā zarnā. *GET* testu rezultāti ir uzskatāmi par īsām laika rindām [21], jo nolasījumu skaits vienai laika rindai sastāv no 10 periodiem (laika intervāliem), kas iegūti no izelpas paraugiem (ievākti nekavējoties pirms procedūras, kā arī katras 20 minūtes pēc substrāta ieņemšanas 3 stundu laikā). Indivīda pašsajūtas novērtējums tiek aprakstīts ar virkni jautājumu, uz kuriem atbildes sniedz indivīds *GET* testa izpildes laikā. Jautājumus uzdod kvalificēts medicīnas darbinieks un atbildes reģistrē speciālā protokolā.

3.3.1. BPS noteikšanas sistēmas uzdevuma nostādne

Lai nodrošinātu kvalitatīvāku medicīnisko aprūpi un neradītu indivīdiem negatīvu attieksmi pret veicamo testu, nepieciešams izstrādāt pieeju, kura nodrošina *BPS* noteikšanu tievajā zarnā, pielietojot datu iegūšanu. Izmantojot datu iegūšanu, izstrādāt sistēmu *BPS* noteikšanai tievajā zarnā. Izstrādātajai sistēmai jānodrošina šādi nosacījumi:

1. Noteikt *BPS* tievajā zarnā pēc vēsturiskiem *GET* testu rezultātiem — īsas laika rindas un indivīda pašsajūtas novērtējumu raksturojošos parametros;

2. Jaunam indivīdam *BPS* noteikšana tievajā zarnā ir jābalsta uz esošo pašsajūtas novērtējumu;
3. Iegūtajiem rezultātiem ir jābūt viegli interpretējamiem.

Lai realizētu izvirzītos uzdevumus, nepieciešams izveidot sistēmu, kas nodrošina un izpilda:

1. Īsu laika rindu apstrādi (*GET* testu rezultāti) ar klasterizācijas algoritmiem, un izvērtē iegūtos rezultātus;
2. Indivīda aprakstošo un pašsajūtas novērtējuma informatīvāko atribūtu atlasi;
3. Klasterizācijas rezultātu (klašu) transformāciju, atbilstoši klīnisko rezultātu klašu struktūrai;
4. Noteikto informatīvāko atribūtu apstrādi, pielietojot klasifikācijas algoritmus;
5. *BPS* noteikšanu tievajā zarnā "jaunam" indivīdam tikai ar pašsajūtas novērtējuma atribūtiem.

3.3.2. BPS noteikšanas sistēmas darbības princips

No izvirzītajiem uzdevumiem, piedāvātajai sistēmai, kas parādīta 3.29. attēlā, ir jāapstrādā divu veidu datu plūsmas. Viena ir *GET* testu vēsturiskie rezultāti, kas attēloti kā īsas laika rindas ar desmit laika intervāliem, otra — indivīda pašsajūtas novērtējums, kuru raksturo atribūti: *dzimums*; *slikta dūša*; *atraugas*; *meteorisms*; *pilnuma sajūta* un klīnisko izmeklējumu rezultāti, kuru raksturo atribūts *klase*. Sistēmai jāveic: informatīvāko atribūtu atlasi, datu klasterizāciju, piemērotākā klasteru skaita noteikšanu, klašu transformāciju, jaunas datu kopas izveidi, apvienojot dažādas datu struktūras, divu veidu datu plūsmu klasifikāciju un *BPS* noteikšanu.

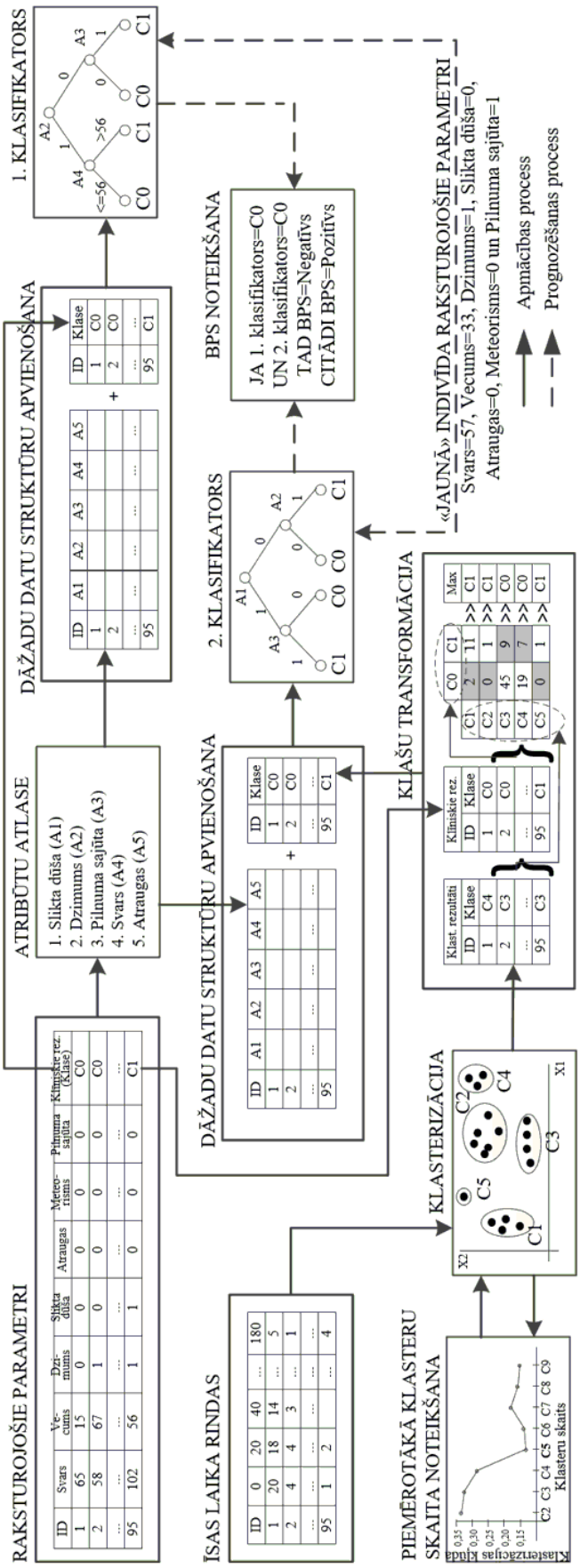
Atribūtu informatīvuma noteikšanai, tiek lietotas *CfsSubsetEval* atribūtu novērtējuma un *BestFirst* pārmeklēšanas metodes, kas bieži tiek izmantotas datu iegūšanā [70]. Pēc indivīda pašsajūtas novērtējuma un raksturojošo informatīvāko atribūtu atlases datu kopai, kura darba ietvaros tiks apzīmēta ar *DS1*, pievieno klīnisko izmeklējumu rezultātus (atribūts *klase* — klīniski apstiprināts baktēriju proliferācijas sindroms). Izveidotā datu kopa *DS1* tiek nodota 1. klasifikatoram, lietojot *k-tuvāko kaimiņu*, *C4.5* un *CN2* klasifikācijas algoritmus, nosaka sakarības starp indivīda raksturojošajiem parametriem un klasēm. Īsas laika rindas tiek apstrādātas klasterizācijas modelī, lietojot *modificētu k-vidējo sadalošo* algoritmu, tiek noteikts piemērotākais klasteru skaits datu kopas klasterizācijai. Klasterizācijas rezultātā iegūtās klases transformācijas modelī tiek pārveidotas uz klīnisko izmeklējumu klašu struktūru. Iegūtā klašu struktūra, vadoties pēc indivīda identifikācijas numura, tiek pievienota atlasītajai informatīvāko atribūtu kopai, kas darba ietvaros tiks apzīmēta ar *DS2*, kas tālāk tiek

apstrādāta 2. klasifikatorā, lietojot *k-tuvāko kaimiņu*, *C4.5* un *CN2* klasifikācijas algoritmus, nosakot sakarības starp indivīda raksturojošajiem parametriem un klasterizācijas rezultātiem. No 1. klasifikatora, kas apstrādā *DS1* datu kopu, tiek iegūti rezultāti, kas raksturo saikni starp indivīda pašsajūtas novērtējuma raksturojošiem parametriem un klīniskajiem rezultātiem. No 2. klasifikatora, kas apstrādā *DS2* datu kopu, tiek iegūti rezultāti, kuri raksturo saikni starp elpas testu mērījumiem un klasterizācijā iegūto klasi, kura tiek reducēta uz klīnisko rezultātu klašu struktūru.

Divu dažādu datu plūsmu apstrādes modelis nodrošina to, ka iegūtie rezultāti tiek analizēti no dažādām pusēm. Tiek iegūti divi neatkarīgi novērtējumi, kuru rezultātus salīdzina *BPS* noteikšanas modelī, kas izvada paziņojumu vai indivīdam ir nepieciešama tālāka izmeklēšana. Šīs piedāvātais divu plūsmu apstrādes modelis garantē, ja indivīds negodprātīgi atbildējis uz pašsajūtas novērtējuma jautājumiem, pastāv iespēja, ka otrs klasifikators atklās nepatiesi sniegtās ziņas.

Indivīda pašsajūtas novērtējuma informatīvāko atribūtu atlase

Indivīda raksturojošajos un pašsajūtas novērtējuma atribūtos tiek meklēti tie atribūti, kuri veido informatīvu atribūtu kopu. Datu klasifikācijā, lai klasifikators uz rādītu augstāku precizitāti, nepieciešams izmantot datu kopu, kurā atribūti ir informatīvi. Tāpēc pirms klasifikācijas procesa uzsākšanas ir eksperimentāli jāveic atribūtu informatīvuma noteikšana vai to sadalīšana pēc atribūtu svarīguma. Atribūtu informatīvuma noteikšanai pētījumos tika izmantota *CfsSubsetEval* atribūtu novērtējuma un *BestFirst* pārmeklēšanas metodes [70]. Pēc atribūtu atlases informatīvākajiem atribūtiem tiek pievienota klīnisko pētījumu rezultātā iegūtā klase, vadoties pēc indivīda identifikācijas numura. Izveidotā *DS1* datu kopa tiek nodota 1. klasifikatoram.



3.29. att. Baktēriju proliferācijas sindroma noteikšanas sistēma

Izelpas ūdeņraža koncentrācijas datu klasterizācija

Izelpas ūdeņraža koncentrāciju, kas formalizēta kā īsas laika rindas, analizē ar klasterizācijas palīdzību, nosakot līdzību starp laika rindām, savstarpēji salīdzinot tās katrā laika punktā. Kā pierādīja pētījumi ar pieprasījuma prognozēšanas sistēmu, kas aprakstīta 3.1.3. apakšpunktā, un sirds nekrozes prognozēšanas sistēmu, kas aprakstīts 3.2.3. apakšpunktā, tad piemērotākais īsu laika rindu klasterizācijas algoritms ir *modificēts k-vidējo sadalošais* algoritms, kura darbības princips ir izklāstīts 2.3.4. apakšpunktā.

Klasterizācijas algoritmam ir jānosaka piemērotākais klasteru skaits, kas būtu izmantojams analizējamās datu kopas klasterizācijai. Svarīgi, lai pielietotais algoritms nodrošinātu klasterizācijas noturību jeb robustumu [35].

Klašu transformācija

Klašu transformācijas modelī, kas parādīts 3.29. attēlā, tiek realizēta klasterizācijā iegūto klašu pārveidošana atbilstoši klīnisko rezultātu klašu struktūrai. Piemēram, ja klasterizācijas rezultātā ir iegūtas piecas klases (klasterizācijā iegūtie klasteri no "C1" līdz "C5"), bet klīnisko rezultātu struktūra sastāv no divām klasēm ("C1" un "C2"), tad tiek veidota jauna struktūra (transformētā klase, sastāv no "C0" un "C1"). Jaunā struktūra, kuras izveidošana parādīta 3.11. tabulā, nosaka, ka katram klasterizācijas rezultātā iegūtajam klasterim tiek aprēķināts (saskatīti objekti katrā no klasteriem) objektu skaits, kas atbilst klīnisko pētījumu klašu struktūrai. Transformētā klase tiek piešķirta pēc lielākā aprēķinātā objektu skaita, kas 3.11. tabulā iezīmēts ar treknrakstu, klīnisko pētījumu klašu struktūrā.

3.11. tabula

Klašu transformācija

Klasterizācijā iegūtie klasteri	Klīnisko rezultātu klašu struktūra		Transformētā klase
	Objektu skaits ar klasi "C0"	Objektu skaits ar klasi "C1"	Max(C0;C1)
C1	2	**11**	C1
C2	0	**1**	C1
C3	**45**	9	C0
C4	**19**	7	C0
C5	0	**1**	C1

Datu savienošanas modelis

Datu savienošanas modelī informatīvākajiem atribūtiem, kas iegūti no indivīda raksturojošajiem un pašsajūtas novērtējuma atribūtiem, tiek pievienota klase, kura noteikta transformācijas modelī, pārveidojot klasterizācijas rezultātus atbilstoši klīnisko rezultātu klašu struktūrai. Datu savienošana notiek pēc indivīda identifikācijas numura. Izveidotā datu kopa tiek apzīmēta ar *DS2* un nodota 2. klasifikatoram (skatīt 3.29. attēlu.).

BPS datu klasifikācija

Pēc sākotnēji izvirzītajiem uzdevumiem, kuri nosaka, ka sistēmas iegūtajiem rezultātiem ir jābūt viegli interpretējamiem un saprotamiem, tāpēc datu klasifikācijai labāk ir izvēlieties tādus algoritmus kā *C4.5*, lieto induktīvos lēmumu kokus [58], *CN2*, lieto induktīvo secināšanu [14] un *k-tuvāko kaimiņu* (*kNN*), lieto attāluma mēru metriku [12]. Šo algoritmu iegūtie rezultāti ir viegli uztverami un interpretējami, salīdzinot, piemēram, ar neironu tīkliem. Klasifikācija kalpo par pamatu, lai noteiktu sakarības starp indivīda aprakstošajiem pašsajūtas novērtējuma atribūtiem un klasterizācijā iegūtajiem rezultātiem, apstrādājot datu kopu *DS2* ar 2. klasifikatoru. Savukārt, apstrādājot datu kopu *DS1* ar 1. klasifikatoru, tiek noteiktas sakarības starp indivīda raksturojošajiem pašsajūtas novērtējuma atribūtiem un klīnisko diagnozi (*GET* elpas testa rezultāti). Katra izveidotā datu kopa tiek analizēta atsevišķā klasifikatorā, un šo klasifikatoru iegūtie rezultāti tiek nodoti *BPS* noteikšanas modelim. Klasifikatoros eksperimentāli tiek lietoti *C4.5*, *CN2* un *k-tuvāko kaimiņu* algoritmi, lai pēc to precizitātes rezultātiem izvēlētos piemērotāko klasifikatoru problēmsfēras uzdevumu realizācijai.

BPS noteikšana

BPS noteikšanas modelī no klasifikatoru (1. klasifikators un 2. klasifikators) saņemtajiem rezultātiem ar nosacījuma likumu tiek noteikts BPS. Ar nosacījuma likumu (piemēram, JA *1.klasifikators*=”C0” UN *2.klasifikators*=”C0” TAD *BPS*=”Negatīvs”) tiek salīdzināti divi klasifikatoru rezultāti, ja *1. klasifikatora* klasifikācijas rezultāts ir klase *C0* un *2. klasifikatora* klasifikācijas rezultāts ir “*C0*”, tad sistēma izvada paziņojumu, ka indivīda *BPS* testa rezultāts ir negatīvs un nav nepieciešana tālāka izmeklēšana. Ja vienā vai abos klasifikatoros rezultāts ir klase “*C1*”, tad sistēma izvada paziņojumu, ka indivīdam ir nepieciešams veikt tālāku izmeklēšanu, veicot glikozes testu.

3.3.3. Eksperimenti ar BPS datu kopu un iegūto rezultātu analīze

Pētījumiem tika izmantoti brīvi pieejamie datu iegūšanas rīki *Weka* un *Orange Canvas*, kurus izmantojot veikti eksperimenti, nosakot:

a) informatīvākos indivīda raksturojošos un pašsajūtas novērtējuma atribūtus;
b) ar klasterizāciju piemērotāko klasteru skaitu H_2 koncentrācijas datos izelpā;
c) piemērotāko klasifikatoru;
d) "jaunā" indivīda *BPS*.

Eksperimentu veikšanai klasterizējot īsu laika rindas, tika izmantota darba autora izstrādāta programmatūra, kas realizē datu apstrādi *modificētam k-vidējo sadalošajam* algoritmam.

BPS noteikšanas sistēmas datu kopas apraksts

Eksperimentiem izmantoti dati no retrospektīva pētījuma, kurā iekļauti abu dzimumu indivīdi bez vecuma ierobežojuma. Pētījumā piedalījās 95 indivīdi, kuriem tika veikti glikozes elpas testi. Tika izstrādāts protokols, kurā ietverts: glikozes elpas testa ūdeņraža nolasījums bez CO_2 korekcijas laika periodā no 0 līdz 180 minūtēm ar intervālu 20 minūtes, kā arī indivīda svars un vecums iekļaušanas brīdī pētījumā; dzimums (Vīrietis-0; Sieviete-1); glikozes testa slēdziens jeb klīniskā pētījuma rezultāts — klase (Negatīvs-0; Pozitīvs-1); indivīda pašsajūtas novērtējuma parametri: Slikta dūša (*Feeling sick*) (Ir-1; Nav-0); Atraugas (*Eructation*) (Ir-1; Nav-0); Meteorisms (*Flatulence*) — gāzu uzkrāšanās (Ir-1; Nav-0) un Pilnuma sajūta (*Fullness*) (Ir-1; Nav-0). Negatīvs glikozes testa rezultāts norāda, ka tālāka pārbaude nav nepieciešama, bet pozitīvs rezultāts — indivīdam ir nepieciešama papildus izmeklēšana.

Informatīvāko atribūtu noteikšana

Izmantojot datu apstrādes rīku *Weka* tika noteikti informatīvākie atribūti, lietojot *CfsSubsetEval* atribūtu novērtējuma un *BestFirst* pārmeklēšanas metodes. No sākotnējās datu kopas, kurā bija septiņi atribūti — *Dzimums*, *Svars*, *Vecums*, *Slikta dūša*, *Atraugas*, *Meteorisms*, *Pilnuma sajūta*, tika atlasīti pieci atribūti: *Slikta dūša* (100 %), *Dzimums* (70 %), *Pilnuma sajūta* (50 %), *Svars* (40 %) un *Atraugas* (40 %). Iegūtie rezultāti iekavās norāda, cik procentos gadījumu atribūts ir ticis iekļauts atribūtu kombinācijās ar lielāko novērtējumu. Atlasītie atribūti tika izmantoti turpmākajos eksperimentos.

Izelpas ūdeņraža koncentrācijas klasterizācija

Eksperimentu gaitā tika noteikts piemērotākais klasteru skaits, kas jāizvēlas datu kopas klasterizācijai, vadoties pēc vidējās klasterizācijas kļūdas pie dažādu klasteru skaita. Kā redzams no 3.12. tabulas, tad piemērotākais klasteru skaits ir pieci, jo pie šī klasteru skaita tiek sasniegta mazākā vidējā absolūtā kļūda (*MAE*).

3.12. tabula

Klasterizācijas rezultāti ar *modificētu k-vidējo sadalošo* algoritmu

	Klasteru skaits							
	2	3	4	5	6	7	8	9
Vidējā absolūtā kļūda	0,335	0,329	0,283	**0,125**	0,146	0,183	0,167	0,154

Rezultāti norāda arī uz klasterizācijas robustumu, pielietojot šo algoritmu īsu laika rindu apstrādei, kas nav iegūstams pielietojot, piemēram, klasisko *k-vidējo sadalošo* algoritmu, kura iegūtie rezultāti parādīti 3.13. tabulā.

3.13. tabula

Klasterizācijas rezultāti ar *k-vidējo sadalošo* algoritmu

	Klasteru skaits							
	2	3	4	5	6	7	8	9
Summēta kvadrātiskā kļūda	20,56	19,15	17,42	15,93	15,23	14,74	13,76	**12,59**

Klasifikācijas algoritmu precizitātes salīdzinošā analīze

Lai noteiktu piemērotāko klasifikācijas algoritmu, kuru varētu izmantot par pamatu realizējamās sistēmas izveidē, tika veikta virkne eksperimentu ar *C4.5*, *CN2* un *k-tuvāko kaimiņu* klasifikācijas algoritmiem. Iegūtie klasifikācijas eksperimentu rezultāti ir parādīti

3.14. tabulā, lietojot 1. klasifikatoru ar *DS1* datu kopu un dažādām precizitātes novērtēšanas pieejām.

3.14. tabula

Klasifikācijas rezultāti ar *DS1* datu kopu

Klasifikācijas algoritmi	10-kārtas šķērsvalidācija			Izlaist vienu		
	Klasifikācijas precizitāte	Jutīgums	Specifiskums	Klasifikācijas precizitāte	Jutīgums	Specifiskums
C4.5	0,65	0,87	0,08	0,64	0,87	0,04
kNN	0,63	0,86	0,04	0,68	0,87	0,12
CN2	0,67	0,91	0,04	0,63	0,86	0,04

Novērojams, ka izmantotie klasifikatori labāk ir atpazinuši negatīvo klasi (“C0”) — indivīdus, kuriem nav nepieciešam tālāka izmeklēšana. Ja raugās no izvirzītā darba uzdevuma, tad sasniegts pozitīvs rezultāts, bet, ja skatās — cik labi klasifikators ir atpazinis pozitīvo klasi (“C1”), tad rezultāts nav iepriecinošs, jo specifiskums, piemēram *C4.5* algoritmam ar 10-kārtu šķērsvalidāciju ir tikai 0,08.

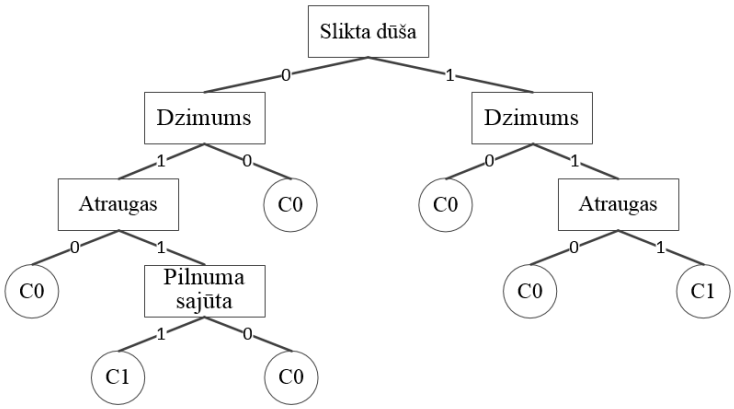
Nākošajā eksperimentā, lietojot 2. klasifikatoru ar *DS2* datu kopu, tika iegūti rezultāti, kuri parādīti **Error! Reference source not found.**. tabulā. Rezultāti parāda, ka *kNN* algoritms vienlīdz labi ir atpazinis gan pozitīvo, gan negatīvo klasi. *CN2* algoritms uzrāda gandrīz 100 % negatīvās klases atpazīstamību, taču pozitīvās klases atpazīstamību varētu vēlēties augstāku. Savukārt, *C4.5* algoritms ar abām precizitātes novērtējuma pieejām uzrādīja līdzīgus rezultātus, atpazīstot negatīvo klasi, jutīgums bija attiecīgi 0,65 un 0,69, bet — pozitīvo klasi, specifiskums bija 0,39 abos gadījumos.

3.15. tabula

Klasifikācijas rezultāti ar *DS2* datu kopu

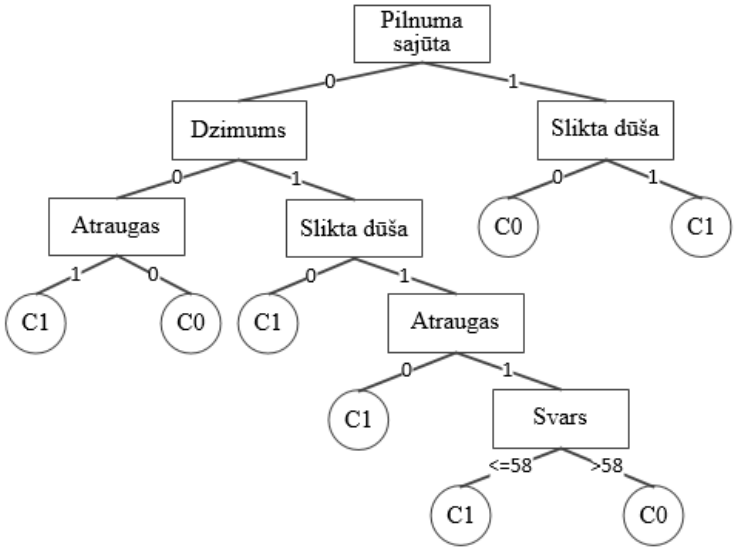
Klasifikācijas algoritmi	10-kārtas šķērsvalidācija			Izlaist vienu		
	Klasifikācijas precizitāte	Jutīgums	Specifiskums	Klasifikācijas precizitāte	Jutīgums	Specifiskums
C4.5	0,53	0,65	0,39	0,55	0,69	0,39
kNN	0,46	0,51	0,39	0,42	0,43	0,41
CN2	0,6	0,98	0,16	0,63	0,98	0,23

Tā kā *C4.5* algoritms uzrādīja salīdzinoši vienādus rezultātus ar abām precizitātes novērtējuma pieejām, klasificējot *DS1* un *DS2* datu kopas attiecīgi ar 1. klasifikatoru un 2. klasifikatoru, par pamatu datu kopu klasifikācijai BPS noteikšanas sistēmā tika izvēlēts *C4.5* algoritms, kura iegūtais klasifikācijas koks ar 1. klasifikatoru ir parādīts 3.30. attēlā.



3.30. att. "Uzbūvētais" klasifikācijas koks 1. klasifikatoram ar *DS1* datu kopu

Iegūtais klasifikācijas koks ar 2. klasifikatoru ir parādīts 3.31 attēlā.



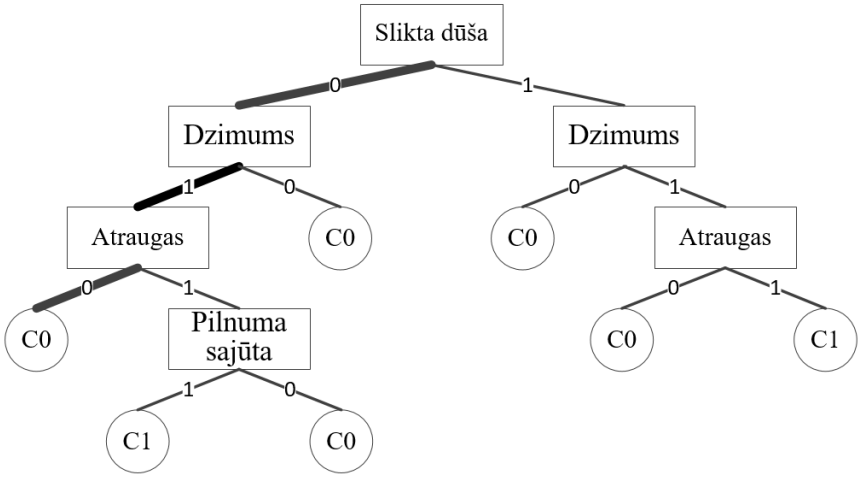
3.31. att. "Uzbūvētais" klasifikācijas koks 2. klasifikatoram ar *DS2* datu kopu

Pētījumi parādīja, ka, eksperimentējot ar *C4.5*, *CN2* un *k-tuvāko kaimiņu* klasifikācijas algoritmiem un precizitātes novērtēšanas pieejām, ir iespējams noteikt piemērotāko klasifikācijas algoritmu, kas nepieciešams BPS noteikšanas sistēmas izstrādāšanai.

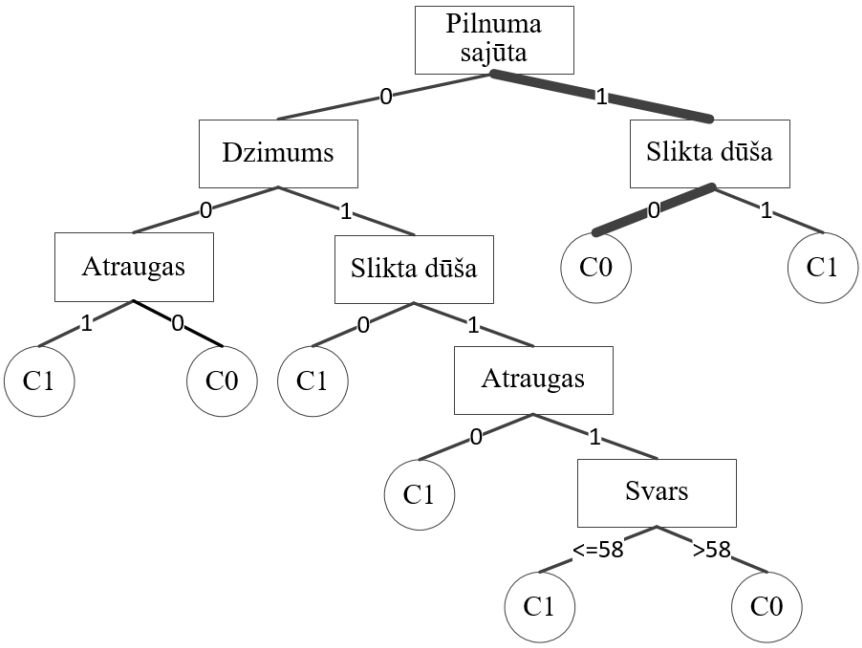
BPS noteikšana "jaunam" indivīdam

BPS noteikšana "jaunam" indivīdam tiek veikta uz klasifikatora apmācības rezultātā iegūtiem klasifikācijas kokiem, uz kuriem tiek projicēti "jaunā" indivīda raksturojošo un pašsajūtas novērtējuma atribūtu vērtības, sasniedzot kādu no klasifikācijas koka lapām, kas raksturo iegūto rezultātu klases norāda uz rezultātu (klasi). Pieņemsim, ka "jaunā" indivīda raksturojošie un pašsajūtas novērtējuma atribūti ir šādi: *Svars=57*, *Vecums=33*, *Dzimums=1*, *Slikta dūša=0*, *Atraugas=0*, *Meteorisms=0* un *Pilnuma sajūta=1*. Apmācības rezultātā iegūtie

klasifikācijas koki ir parādīti 3.32. un 3.33. attēlos pa kreisi, bet "jaunā" indivīda raksturojošie un pašsajūtas novērtējuma atribūti, kas ir projicēti uz iegūto klasifikācijas koku (skatīt 3.32. un 3.33.) — attēlos pa labi. Ar izceltu līniju tiek norādīts ceļš un nākamo koka līmeni, vadoties pēc attiecīgā atribūta vērtības. Piemēra gadījumā tika iegūta klase „C0", kas norāda uz negatīvu rezultātu — indivīdam nav nepieciešama tālāka izmeklēšana. Ja projekcijas rezultātā netiek noteikta klase, citiem vārdiem sakot, nav sasniegta neviena no klasifikācijas koka lapām, tad iegūtais rezultāts tiek pielīdzināts klasei „C1" — indivīdam ir nepieciešams veikt tālāku izmeklēšanu.



3.32. att. Jaunā indivīda raksturojošo atribūtu vērtību projicēšana uz iegūto lēmumu koku, kas iegūts klasifikācijas rezultātā ar 1. klasifikatoru un *DS1* datu kopu



3.33. att. Jaunā indivīda raksturojošo atribūtu vērtību projicēšana uz iegūto lēmumu koku, kas iegūts klasifikācijas rezultātā ar 2. klasifikatoru un *DS2* datu kopu

3.3.4. Kopsavilkums un secinājumi par BPS noteikšanas sistēmu

Rezultātā izveidota *BPS* noteikšanas sistēma [39], ar kuru iespējams noteikt vai indivīdam ir nepieciešams veikt laktulozes testu. Eksperimenti parādīja, ka izveidotā BPS noteikšanas sistēma ir spējīga apstrādāt dažādas datu struktūras — īsas laika rindas un indivīda raksturojošos parametrus. Iegūtie rezultāti ir viegli uztverami (BPS pozitīvs vai negatīvs) sistēmas lietotājam.

Pētījumi parādīja, ka, lietojot datu iegūšanas metodes un algoritmus, iespējams izveidot BPS noteikšanas sistēmu, nepakļaujot indivīdu garai un nepatīkamai medicīniskai procedūrai.

Klasterizējot īsas laika rindas ar *modificētu k-vidējo sadalošo* algoritmu tika noteikts piemērotākais klasteru skaits pēc vidējās absolūtās kļūdas aprēķina.

Darba autora piedāvātā pieeja nodrošina klasterizācijā iegūto klašu struktūru transformāciju vienotā klīnisko izmeklējumu klašu struktūrā.

Lietojot *C4.5, CN2* un *k-tuvāko kaimiņu* algoritmus, eksperimentāli tika noteikts piemērotākais klasifikators *DS1* un *DS2* datu kopu klasifikācijai, kas bija *C4.5* algoritms. Izveidotās BPS noteikšanas sistēmas precizitāte tika noteikta, pielietojot 10-kārtu šķērsvalidāciju, jo pie neliela objektu skaita nav lietderīgi datu kopu sadalīt apmācības un testēšanas kopās, tā kā tas vēl vairāk samazinās apmācības kopas izmēru.

Divu klasifikatoru izmantošana rezultātā var ar lielāku ticamību noteikt *BPS* tievajā zarnā, jo tiek ņemti vērā indivīda pašsajūtas novērtējuma faktori un vēsturiski elpas testu rezultāti. Ja indivīds ir negodprātīgi atbildējis uz pašsajūtas novērtējuma jautājumiem, tad pastāv iespēja, ka klasifikators, kurš apmācīts ar datiem par H_2 koncentrāciju izelpā, klasificēs paaugstinātu H_2 līmeni izelpā, kas norādīs uz pozitīvu BPS.

Eksperimentos ar klasifikatoriem tika noskaidrots, ka izveidotā sistēma labāk atpazīt negatīvo klasi — indivīdus, kuriem nav nepieciešama tālāka izmeklēšana, kas būtībā ir izveidotās sistēmas mērķis — atlasīt indivīdus, kuriem netiek lieki tērēti veselības aprūpes resursi un finansējums, kā arī indivīdi netiek pakļauti garam un nepatīkamam izmeklējuma procesam. Protams, lai piedāvāto sistēmu integrētu veselības aprūpes vai gastroenteroloģijas skrīninga sistēmās tā vēl ir jāpilnveido, iegūstot lielāku indivīdu populācijas kopu, kas iespējams uzlabotu negatīvās klases atpazīstamību.

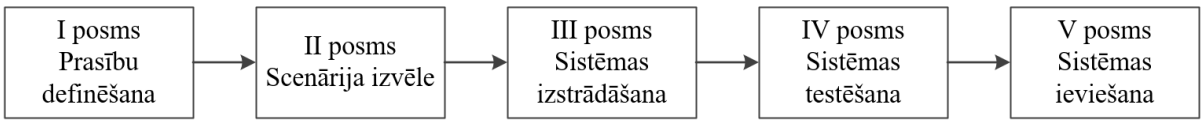
Pēc indivīdu populācijas grupas palielināšanas un eksperimentiem ar izveidoto modeli tiks izstrādāts sistēmas prototips, ar domu to integrēt veselības aprūpes iestādē tālākai testēšanai.

4. PROGNOZĒŠANAS SISTĒMAS IZSTRĀDĀŠANAS VADLĪNIJAS

Šīs nodaļas ietvaros tiek aprakstītas prognozēšanas sistēmas izstrādāšanas vadlīnijas (PSIV), kas radītas, balstoties uz iegūto pieredzi līdzīgu sistēmu izveidē, par kurām tiek stāstīts iepriekšējā nodaļā. PSIV sniedz paskaidrojumus, kā realizēt datu apstrādes sistēmas, kur par datu avotu kalpo īsas laika rindas un to raksturojošie parametri, balstoties uz iegūto pieredzi līdzīgu sistēmu izveidošanā. Vadlīnijas veidotas, balstoties uz iegūto pieredzi preču pieprasījumu, sirds nekrozes riska un baktēriju proliferācijas sindroma prognozēšanas sistēmu izstrādāšanā. Vadlīnijas norāda izpildītājam, kādu modeļa struktūru ieteicams veidot, balstoties uz pasūtītāja iesniegto datu aprakstu. Pasūtītājs ir persona vai organizācija, kas definē prasības un iesniedz analizējamo datu aprakstu. Izpildītājs ir persona vai organizācija, kas novērtē iesniegtās prasības, izanalizē tās un sniedz pasūtītājam atbildi. Ja puses vienojas par nākamo soli, tad izpildītājs uz iesniegto prasību pamata izstrādā piemērotāko sistēmas risinājumu, balstoties uz vadlīnijām. Pēc prognozēšanas sistēmas risinājuma izvēles, izstrādātājs realizē sistēmu un veic tās darbības pārbaudi, vadoties pēc klasifikācijas precizitātes novērtējuma. Ja novērtējums ir pietiekošs, izpildītājs veic sistēmas ieviešanu un integrāciju pasūtītāja informācijas sistēmās.

4.1. Prognozēšanas sistēmas struktūra

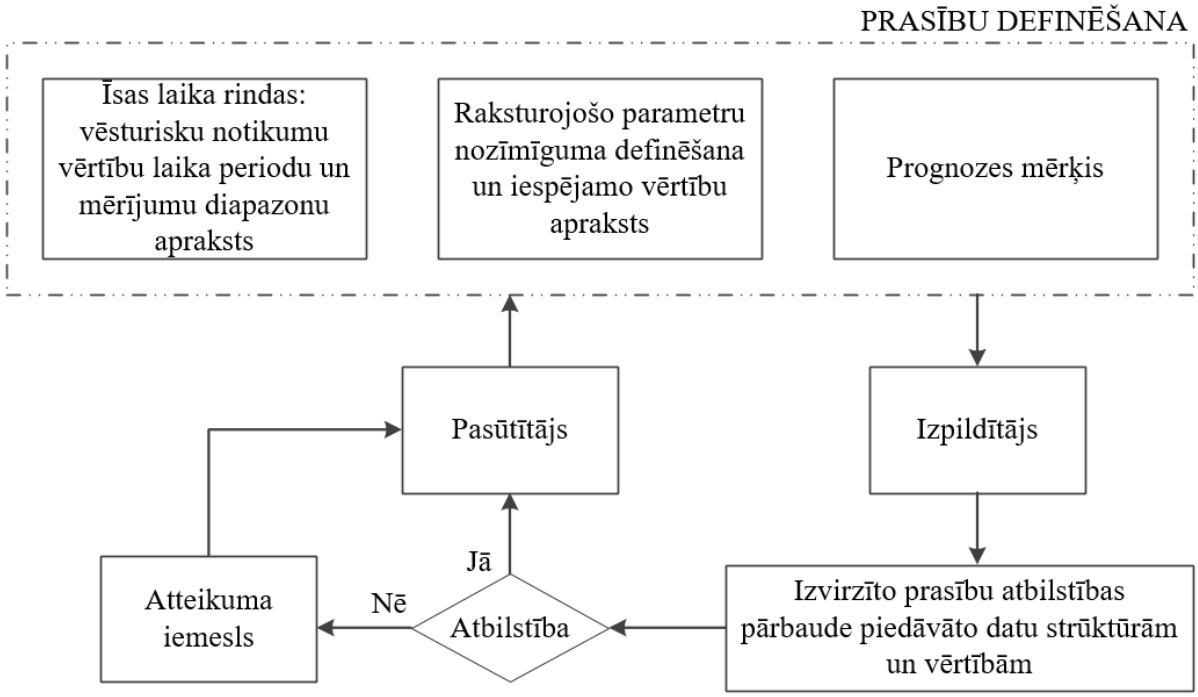
Prognozēšanas sistēmas struktūra ir parādīta 4.1. attēlā, tā sastāv no pieciem izstrādāšanas posmiem. Pirmajā posmā tiek veikta prasību definēšana, kas ietver pasūtītāja un izpildītāja dialogu, analizējot pasūtītāja piestādīto datu atbilstību izstrādājamajai sistēmai un izvirzītajam uzdevumam — prognozes mērķim. Otrajā posmā notiek piemērotākā sistēmas izstrādāšanas scenārija izvēle. Trešajā posmā tiek izstrādāts sistēmas koncepts. Ceturtajā posmā notiek izstrādātā sistēmas koncepta testēšana, ja šī izveidotā koncepta precizitāte ir pietiekoša, tad koncepts tiek pārveidots sistēmā. Piektajā posmā tiek veikta izveidotās sistēmas integrācija pasūtītāja informācijas sistēmā.



4.1. att. Prognozēšanas sistēmas izstrādāšanas posmi

Prasību definēšanas posms izvērstā veidā parādīts 4.2. attēlā, kur pasūtītājs iesniedz izpildītājam analizējamo datu kopumu un definē prognozes mērķi. Lai izveidotu

prognozēšanas sistēmu, analizējamiem datiem ir jāsatur vēsturisku notikumu vērtības un to raksturojošie parametri. Vēsturiskās notikumu vērtības tiek definētas kā īsas laika rindas, to novērojumu periodiem ir jābūt nelieliem un unikāliem. Vēsturisko notikumu vērtību diapazona atbilstību analizējamiem procesiem nodrošina ar datu pirmapstrādes palīdzību. Raksturojošie parametri apraksta vēsturisku notikumu vērtību īpašības. Raksturojošo parametru (atribūtu) vērtību diapazons var būt gan nepārtraukts, gan diskrēts, piemēram, preces cena un preces veids. Raksturojošo parametru atbilstību analizējamiem procesiem nodrošina ar datu pirmapstrādes palīdzību. Pasūtītājs var arī strikti definēt (sakārtot pēc nozīmīguma) raksturojošos parametrus, tad izstrādātājam ir jāveic attiecīgi soļi sistēmas izstrādāšanas vadlīniju izvēlē.



4.2. att. Prognozēšanas sistēmas prasību definēšana (I posms)

Prognozes mērķim ir jābūt skaidri definētam, tā sasniegšanai jālieto tikai "jaunā" analizējamā objekta raksturojošie parametri. Gadījumā, ja sistēmas apmācības procesā, ir jālieto noteikts vēsturisku notikumu periods (īsas laika rindas par noteiktu laika periodu, piemēram, preču pārdošanas apjoms par 2012. gadu), tad pasūtītājam jāpieprasa šīs iespējas integrēšana sistēmā. Pēc pasūtītāja iesniegto prasību definēšanas, izpildītājs novērtē datu struktūru atbilstību izvirzītajam mērķim, ja ir atbilstība, tad var pāriet pie nākamā posma izpildes, ja nē, tad izpildītājs izveido atskaiti par atteikuma iemeslu un veic neatbilstības novēršanu sadarbībā ar pasūtītāju.

4.2. Prognozēšanas sistēmas scenārija izvēle

Prognozēšanas sistēmas scenārija izvēle sākas no lietotāja datu bāzes (skatīt 4.3. attēlu), no kuras ar vaicājuma palīdzību tiek atlasīta analīzei nepieciešamā informācija (vēsturiskas īsas laika rindas un to raksturojošie parametri), kas nonāk datu sadalīšanas blokā. Datu sadalīšanas blokā tiek izdalītas divas datu plūsmas: vēsturiskas laika rindas un to raksturojošie parametri. Vēsturiskām laika rindām ir iespēja pielietot datu sagatavošanu un/vai datu pirmapstrādi. Datu sagatavošanas procesā tiek piedāvāts veikt periodu izlīdzināšanu, iztrūkstošu vērtību apstrādāšanu. Datu normalizācijas procesā tiek piedāvāts veikt dominējošu vērtību *normalizāciju ar dzīves līkni*. Līdzīgu objektu grupu noteikšana tiek veikta ar klasteru analīzi, pielietojot *modificētu k-vidējo sadalošo* algoritmu, nosakot piemērotāko klasteru skaitu datu kopas klasterizācijai. Vadoties pēc uzdevuma specifikas, ir iespējams veikt klasterizācijas procesā iegūto klašu skaita transformāciju uz raksturojošo parametru klašu skaitu. Iespējams arī klasterizācijas rezultātus pārveidot paraugmodeļos, kas attēlo klasteru vidējās vērtības katrā laika periodā. Raksturojošo parametru datu sagatavošanas procesā tiek piedāvāts veikt iztrūkstošu vērtību apstrādāšanu, datu normalizāciju, pielietojot *z-novērtējuma normalizāciju ar standarta novirzi*, un datu diskretizāciju. Raksturojošo parametru klasifikācijas procesam tiek piedāvātas vairākas iespējas:

a) klasifikācijai izmantot datu kopu, kas sastāv no raksturojošajiem parametriem un šo parametru raksturojošās klases (noteikta klīnisko, farmakoloģisko vai citu pētījumu rezultātā un tiek uzskatīta par "zelta standartu");
b) ja nepieciešams, veikt raksturojošo parametru datu kopas sadalīšanu apakškopās, vadoties pēc atribūtu nozīmīguma sadalījuma, kas noteikts eksperimentāli, vai arī to ir definējis pasūtītājs;
c) klasifikācijai izmantot datu kopu, kas sastāv no raksturojošajiem parametriem un klasterizācijā iegūtās klases.

Dažādu datu tipu apvienošana tiek realizēta, ja izvēlēta pieeja, kas aprakstīta apakšpunktos "b" un "c". Sakarību noteikšanai starp dažādu struktūru datiem jeb klasifikācijai, tiek piedāvāts izmantot *C4.5* algoritmu, atkarībā no uzdevuma specifikas klasifikatoru skaits var mainīties.

Nosacījumu likumu veidošana balstās uz vairāku procesu darbības rezultātu apkopošanu JA … UN … TAD … likumu veidā, kur:

a) pirmais nosacījums ir izveidotās apakškopas sadalošā parametra vērtība (piemēram, analizējamās vielas veids — atribūts *Grupa*, kas pievienots barībai; vērtību

diapazons 1-5);

b) otrais nosacījums ir analizējamā objekta klase (iegūta klasterizācijas rezultātā);

c) slēdziens — atribūts *Risks* ir objekta "zelta standarta" vērtība.

Katra objekta klasifikācijas rezultātā tiek izveidots viens nosacījumu likums, kas tiek saglabāts likumu datu bāzē, piemēram JA Grupa=1 UN Klase=C2 TAD Risks=35.

Promocijas darbā dažādām problēmvidēm izstrādāto sistēmu scenāriju attēlojumi ir parādīti 4.3. attēlā ar sarkano krāsu — pieprasījumu prognozēšanas sistēma (PPS), ar zilu krāsu — sirds nekrozes riska prognozēšanas sistēma (SNRPS) un ar zaļu krāsu — baktēriju proliferācijas sindroma noteikšanas sistēma (BPSNS).

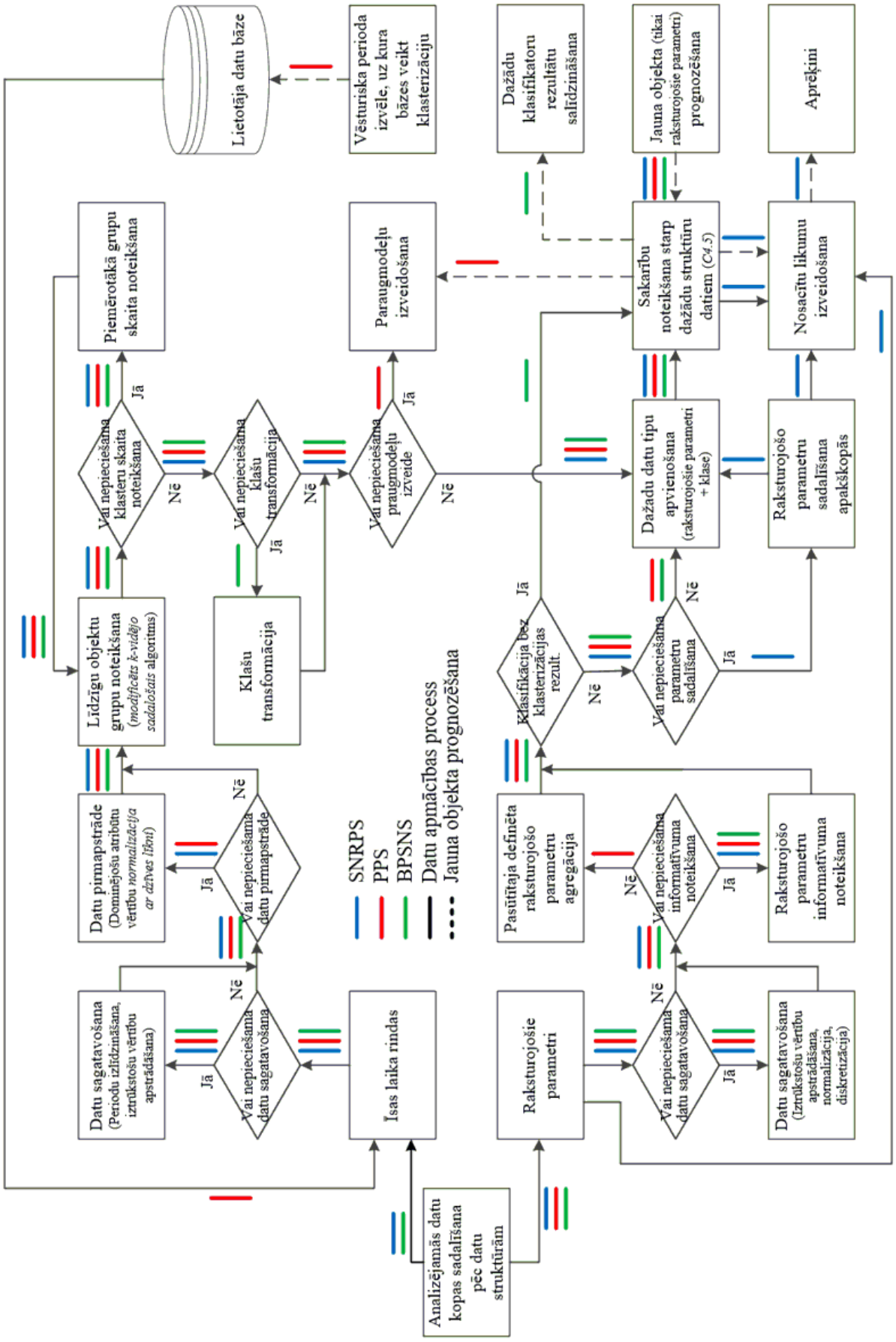
Jauna objekta prognozēšana tiek veikta, ievadot sistēmā objekta raksturojošos parametrus, kur ar klasifikatora palīdzību nosaka objekta piederību kādai no klasēm un tālāk atkarībā no uzdevuma specifikas tiek realizēta iegūto klasifikācijas rezultātu interpretācija:

a) ja prognozēšana tiek veikta, izmantojot paraugmodeļus, tad klasifikatora iegūtā klase norāda uz attiecīgā paraugmodeļa numuru, kas raksturo objekta pieprasījumu nākotnē noteiktā laika posmā. Lietotājam pastāv iespēja, ka pirms prognozes veikšanas sistēma tiek apmācīta uz konkrēta izvēlēta vēsturiska pieprasījuma perioda;

b) ja prognozēšana tiek veikta izmantojot nosacītus likumus, tad klasifikatora iegūtā klase un sadalošā parametra vērtība norāda kādus nosacītus likums, ir jāatlasa no nosacītu likumu datu bāzes. Balstoties uz atlasītajiem likumiem, kas raksturo nosacīto likumu parādīšanās biežumu, tiek veikti aprēķini izskaitļojot prognozēto vērtību;

c) ja prognozēšana tiek veikta, izmantojot datu kopas ar un bez klasterizācijas rezultātu izmantošanas, tad prognozes noteikšana balstās uz dažādu klasifikatoru rezultātu salīdzināšanu. Ja abi klasifikatori nosaka klasi "C0", tad prognoze nozīmē, ka nav jāveic turpmākas darbības. Bet ja rezultāts ir klase "C1" vai rezultāts netika noteikts, tad prognoze nozīmē, ka jāveic turpmākas darbības.

Scenārija izvēles patiesumu var noskaidrot tikai eksperimentāli, izveidojot sistēmas konceptu un veicot testēšanu ar pasūtītāja reālo datu kopu.



4.3. att. Prognozēšanas sistēmas izveidošanas scenārija izvēle (II posms)

4.3. Prognozēšanas sistēmas izstrādāšana un testēšana

Pēc scenārija izvēles tiek izstrādāts prognozēšanas sistēmas koncepts, uz kura bāzes tiek veikti eksperimenti ar pasūtītāja reālu datu kopu, rezultātā nosakot klasifikatora precizitāti, jutīgumu un specifiskumu. Atsevišķos gadījumos, it īpaši medicīnā, var tikt lietotas arī papildus precizitātes novērtējuma vērtības, kuras var aprēķināt no paplašinātās neskaidrības matricas, kas parādīta 2.8. tabulā. Ja analizējamā datu kopa sastāv no 200 un vairāk objektiem, tad izveidojamās sistēmas klasifikators tiek apmācīts ar apmācības kopas datiem un pēc tam pārbaudīts ar testēšanas kopas datiem. Bet, ja analizējamās datu kopas objektu skaits ir mazāks par 200, tad tiek lietota 10-kārtu šķērsvalidācija, kura garantē novērtējuma drošumu pie neliela objektu skaita. Skaitlis 200 ir aptuvenā robeža. Ja iegūtais izstrādātās sistēmas klasifikācijas rezultātu novērtējums ir pietiekošs, kuru nosaka eksperts — sistēmas izstrādātājs, tad var no šī koncepta izveidot lietojumu, kurš tālāk tiktu integrēts pasūtītāja informācijas sistēmā.

Gadījumā, ja izstrādātās sistēmas iegūto klasifikācijas rezultātu novērtējums ir izrādījies nepietiekošs, ko nosaka eksperts — sistēmas izstrādātājs, tad, balstoties uz vadlīnijām, iespējams izvēlēties citus risinājumus un veikt atkārtotu izveidotās sistēmas precizitātes novērtēšanu. Svarīgi ir pareizi izvēlēties piemērotākās datu sagatavošanas un datu pirmapstrādes pieejas, jo lielākās neprecizitātes datu ieguves procesā veido tieši nepiemērotu datu pirmapstrādes pieeju pielietošana vai nekvalitatīvi pirmapstrādāta analizējamā datu kopa.

Piedāvātās vadlīnijas īsu laika rindu un raksturojošo parametru apstrādāšanai varētu būt arī nepilnīgas jaunu sistēmu izveidošanai dažādās problēmvidēs, jo tās veidos balstoties uz iegūto pieredzi preču pieprasījuma prognozēšanā tirdzniecībā, sirds nekrozes riska prognozēšanā farmakoloģijā un baktēriju proliferācijas sindroma noteikšanā medicīnā. Tāpēc, veidojot jaunas sistēmas citās problēmvidēs, vadlīnijas noteikti varētu papildināt ar jauniem blokiem, iespējamiem notikumiem un citām datu plūsmām.

4.4. Kopsavilkums un secinājumi

Nodaļā aprakstītas prognozēšanas sistēmas izstrādāšanas vadlīnijas, kas veidotas, balstoties uz līdzīgu sistēmu izveidi dažādās problēmvidēs. Sistēmu līdzību pamato fakts, ka tās izstrādātas uz vienota prognozēšanas sistēmas teorētiskā modeļa bāzes.

Izveidota prognozēšanas sistēmas struktūra, kas sastāv no pieciem posmiem: prasību definēšanas, scenārija izveides, sistēmas koncepta izstrādāšanas un testēšanas, lietotnes izstrādāšanas un ieviešanas.

Izveidots prasību definēšanas process, kurā notiek pasūtītāja un izstrādātāja dialogs par analizējamajām datu struktūrām un sasniedzamo mērķi — prognozi.

Izveidots prognozēšanas sistēmas izstrādāšanas scenārijs, kas balstīts uz līdzīgām sistēmām, uz kura pamata var izveidot jaunu prognozēšanas sistēmu, kas apstrādā īsas laika rindas un to raksturojošos parametrus.

Vadlīniju pielietošana jaunu prognozēšanas sistēmu izstrādāšanā nodrošina piemērotākā scenārija izvēli, vizuāli uzskatāmu sistēmas izveides principu un pielietoto pieeju izvēli.

Vadlīnijas iespējams ērti un vienkārši papildināt, integrējot tajās jaunas izstrādātās sistēmas komponentes vai nosacījumus.

5. REZULTĀTU ANALĪZE UN SECINĀJUMI

Promocijas darbā piedāvāta uz datu iegūšanu balstīta īsu laika rindu un to raksturojošo parametru apstrādes sistēma prognozēšanas uzdevumiem, kas izstrādāta dažādām problēmvidēm. Darba izstrādāšanas gaitā tika atrisināti šādi uzdevumi:

1. Izanalizēti īsu laika rindu un to raksturojošo parametru apstrādes pamatprincipi, nosakot metodes, kas izmantojamas sistēmas realizācijai.
2. Analīzes rezultātā noteiktas piemērotākas īsu laika rindu un to raksturojošo parametru datu pirmapstrādes metodes.
3. Izstrādāts atbilstoši problēmvidei *modificēts k-vidējo sadalošais* algoritms, kas realizē īsu laika rindu klasterizāciju, nosakot piemērotāko klasteru skaitu. Izstrādātā algoritma modifikācija salīdzināta ar citu klasterizācijas algoritmu darbību.
4. Izstrādāta klasterizācijas rezultātu un īsu laika rindu raksturojošo parametru datu apvienošanas tehnika, piedāvājot klašu transformācijas pieejas, lai izlīdzinātu klašu skaitu analizējamajās datu struktūrās.
5. Izstrādātas tirdzniecības, farmakoloģijas un medicīnas nozaru prognozēšanas sistēmas, kas veic jauna objekta prognozi, balsoties tikai uz sistēmā ievadītajiem objekta raksturojošajiem parametriem.
6. Veikts sistēmā pielietoto klasterizācijas un klasifikācijas modeļu precizitātes novērtējums.
7. Izstrādātas nosacījumu likumu veidošanas un pielietošanas pieejas dažādām problēmvidēm.
8. Izstrādātas vadlīnijas, uz kuru bāzes iespējams izveidot jaunas prognozēšanas sistēmas, kur kā datu avots kalpo īsas laika rindas un to raksturojošie parametri.

Preču pieprasījumu prognozēšanas sistēmā realizēta klasterizācijas rezultātu attēlošana ar paraugmodeļiem, kuri pēc tam tika izmantoti arī prognozēšanā. Realizēta vairāku klasifikatoru izmantošana sirds nekrozes riska prognozēšanas sistēmā un baktēriju proliferācijas sindroma noteikšanas sistēmā. Izstrādāta klasifikācijas apmācības procesā iegūto rezultātu un farmakoloģisko pētījumu rezultātu apvienošanas pieeja, kas realizēta nosacījumu likumu veidā.

Izstrādātā klasterizācijas algoritma modifikācija un visas trīs izveidotās prognozēšanas sistēmas tika eksperimentāli testētas, lai pārbaudītu izvirzītās hipotēzes un to rezultāti ir šādi:

1. Izveidotās sistēmas dažādām problēmvidēm apstiprina pirmo izvirzīto hipotēzi, ka īsu laika rindu un to raksturojošo parametru apstrādes sistēmas izstrādāšana nodrošina grūti formalizējama uzdevuma atrisināšanu ar datu iegūšanas metodēm un algoritmiem.
2. Otro hipotēzi apstiprina fakts, ka ir izstrādāts *modificēts k-vidējo sadalošais* algoritms, kas uzlabo piemērotākā klasteru skaita noteikšanu datu klasterizācijas procesā, analizējot īsas laika rindas. Izstrādātā algoritma modifikācija eksperimentāli pārbaudīta dažādās problēmvidēs un salīdzināta ar citu klasterizācijas algoritmu darbību.
3. Trešo hipotēzi apstiprina fakts, ka ir izstrādātas preču pieprasījuma prognozēšanas, sirds nekrozes riska prognozēšanas un baktēriju proliferācijas sindroma noteikšanas sistēmas, kuras tiek lietotas dažādās problēmvidēs: tirdzniecībā, farmakoloģijā un medicīnā, apstrādājot datu kopas, kas sastāv no īsam laika rindām un to raksturojošajiem parametriem.

Darba izstrādāšanas gaitā eksperimentāli tika noskaidrots, ka:

a) piemērotākā īsu laika rindu normalizācijas metode ir *normalizācija ar dzīves līkni*, kas uzrādīja labāko novērtējumu visās izstrādātajās sistēmās;
b) piemērotākais īsu laika rindu klasterizācijai ir *modificēts k-vidējo sadalošais* algoritms, pie tam tas saglabāja robustumu visā analizējamajā klasteru diapazonā un uzrādīja labākos novērtējumus visās izstrādātajās sistēmās;
c) piemērotākais klasifikators ir *C4.5* algoritms, jo tas uzrādīja labākos rādītājus klasifikācijas precizitātes novērtējumos;
d) datu kopām ar ierakstu skaitu līdz 200, klasifikācijas precizitātes novērtējumam, ieteicams izmantot 10-kārtu šķērsvalidāciju, bet datu kopām virs 200 ierakstiem — izmantot datu sadalīšanu apmācības un testēšanas kopās attiecībā 70:30;
e) no statistiskā sadalījuma aprēķinot iespējamo sirds nekrozes risku, ieteicams lietot darba autora piedāvātu pieeju, kas balstās uz attālumu mēriem.

Promocijas darba sasniegtie zinātniskie rezultāti:

1. Izstrādāta klasterizācijas algoritma modifikācija īsu laika rindu klasteru analīzei.
2. Izstrādāta pieeja dažāda skaita klašu struktūru transformācijai vienotā klašu struktūrā.
3. Izstrādātas vadlīnijas, kuras norāda kā veidojamas līdzīgas prognozēšanas sistēmas.

Promocijas darba sasniegtie praktiskie rezultāti:

1. Izstrādāta preču pieprasījuma prognozēšanas sistēma, kas nosaka analizējamā objekta iespējamo pieprasījumu, un kura tika pārbaudīta ar reāliem preču pieprasījuma datiem.
2. Izstrādāta sirds nekrozes riska prognozēšanas sistēma, kas nosaka analizējamā objekta sirds nekrozes riska iespējamo vērtību. Sistēma tika pārbaudīta ar reāliem farmakoloģisko pētījumu datiem.
3. Izstrādāta baktēriju proliferācijas sindroma noteikšanas sistēma tievajā zarnā, kas nosaka vai analizējamajam indivīdam ir nepieciešams veikt laktozes testu. Sistēma tika pārbaudīta ar reāliem medicīnas datiem.

Promocijas darba izstrādāšanas gaitā ir iegūti šādi secinājumi par izveidoto īsu laika rindu un to raksturojošo parametru apstrādes sistēmu prognozēšanas uzdevumiem:

1. Izstrādājot PPS un SNRPS, eksperimentāli tika pierādīts, ka datu normalizēšanai abās sistēmās labākus rezultātus uzrādīja pieeja — *normalizācija ar dzīves līkni*.
2. Raksturojošo parametru informatīvuma noteikšana un atlasīšana uzlabo klasifikācijas rezultātus, samazina analizējamās datu kopas apjomu un palielina algoritma izpildes ātrumu.
3. Izstrādātā klasterizācijas algoritma modifikācija ir pielietojama arī citu klasteru analīzes uzdevumu risināšanā, kur par datu avotu kalpo īsas laika rindas.
4. Izstrādātā klasterizācijas algoritma modifikācija pieļauj dažādu novērtēšanas pieeju izmantošanu: 10-kārtu šķērsvalidāciju vai datu kopas sadalīšanu apmācības un testēšanas kopās.
5. Izstrādātā klašu transformācijas pieeja nodrošina dažāda klašu skaita datu struktūru salīdzināšanu.
6. Izstrādājot līdzīgas prognozēšanas sistēmas medicīnā, klasifikācijas precizitātes novērtējumam paralēli jāsalīdzina arī jutīgums un specifiskums, jo šie parametri būtiski ietekmē klasifikatora izvēli.
7. Visās trijās izstrādātajās prognozēšanas sistēmās par piemērotāko klasifikācijas algoritmu eksperimentāli tika noteikts *C4.5* algoritms.

8. Vairāku klasifikatoru pielietošana sistēmu izstrādāšanā uzlabo klasifikatora kopējo precizitāti, kas eksperimentāli tika pierādīts ar sadalītu un nesadalītu datu kopām.
9. Izstrādātās īsu laika rindu un to raksturojošo parametru prognozēšanas sistēmas pamato un nodrošina grūti formalizējama uzdevuma atrisinājumu, lietojot klasterizācijas, to modifikācijas un klasifikācijas algoritmu apvienošanu.
10. Izstrādātās īsu laika rindu un to raksturojošo parametru prognozēšanas sistēmas, realizē "jauna" objekta prognozēšanu, pamatojoties tikai uz šī objekta raksturojošajiem parametriem.
11. Izstrādātās prognozēšanas sistēmas vadlīnijas palīdz nodrošināt jaunu prognozēšanas sistēmu izveidošanu, kā arī pieļauj iespēju papildināt esošas vadlīnijas.

Turpmākie pētījumi ir saistīti ar medicīnu — kuņģa vēža riska mazināšanas skrīninga sistēmas izstrādāšanu [41], kur promocijas darba gaitā izstrādātās prognozēšanas sistēmas vadlīnijas varētu izmantot kuņģa vēža skrīninga sistēmas izveidošanai, sasaistot neinvazīvu izmeklējumu rezultātus — īsas laika rindas ar respondenta raksturojošajiem parametriem. Šā veida skrīninga sistēmas varētu integrēt veselības aprūpes centros, kas palīdzētu nozares speciālistiem lēmumu pieņemšanā, nosakot diagnozes vai nozīmējot pacientam izmeklējumu pie speciālista.

IZMANTOTĀS LITERATŪRAS SARAKSTS

1. Datu ieguve: Pamati/ A. Sukovs, L. Aleksejeva, K. Makejeva u. c. – Rīga: Rīgas Tehniskā universitāte, SIA „Drukātava", 2006. – 130 lpp.
2. Dravnieks J. Matemātiskās statistikas metodes sporta zinātnē. – Rīga, 2004. – 76 lpp.
3. Klasifikācija un klasterizācija izplūdušajā vidē/ L. Aleksejeva, O. Užga-Rebrovs, A. Borisovs – Rīga: Rīgas Tehniskā universitāte, 2012. – 248 lpp.
4. Latvijas Zinātņu Akadēmijas TK ITTEA terminu datubāze/ Internets. – http://termini.lza.lv - Resurss apskatīts 2015. gada 22. janvārī.
5. Smotrovs J. Varbūtības teorija un matemātiskā statistika. – Rīga: Apgāds Zvaigzne ABC, 2004. – 264 lpp.
6. Varbūtību teorijas un matemātiskās statistikas elementi medicīnas studentiem/ U. Teibe, U. Berķis – Rīga: AML/RSU, 2001. – 88 lpp.
7. Alba E., Mendoza M. Bayesian forecasting methods for short time series// Foresight: The International Journal of Applied Forecasting, International Institute of Forecasters. – 2007. – Issue 8. – pp. 41–44.
8. Armstrong J.S., Collopy F. Error measures for generalizing about forecasting methods: Empirical comparisons// International Journal of Forecasting 8. – 1992. – pp. 69–80.
9. Armstrong J. S., Collopy F., Yokum J. T. Decomposition by causal forces: A procedure for forecasting complex time series// International Journal of Forecasting 21. – 2005. – pp. 25–36.
10. Armstrong J.S., Fildes R. Correspondence on the selection of error measures for comparisons among forecasting methods// International Journal of Forecasting 14. – 1995. – pp. 67–71.
11. Berndt D. J., Clifford J. Using dynamic time warping to find patterns in time series// Association for the Advancement of Artificial Intelligence, Workshop on Knowledge Discovery in Databases (AAAI), – 1994. – pp. 229–248.
12. Berry M.W., Browne M. Lecture notes in data mining. – World Scientific Publishing Co. Pte. Ltd., 2006. – 222 p.
13. Boyer K. K., Verma R. Operations and Supply Chain Management for the 21st Century. – USA: South–Western Cengage Learning, 2010. – 560 p
14. Clark P., Niblett T. The CN2 induction algorithm. Machine Learning, 3(4), – 1989. – pp. 261–283.

15. Cost S., Salzberg S. A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features // Machine Learning. – 1993. – Vol.10. – pp. 57–78.
16. Dambrova M., Liepinsh E., Kalvinsh I., Mildronate. Cardioprotective Action through Carnitine-Lowering Effect// Trends Cardiovasc. Med. – 2002. – Vol.12. – pp. 275–279.
17. Dellaert F. The Expectation Maximization Algorithm. College of Computing, Georgia Institute of Technology, Technical Report number GIT-GVU-02-20, – 2002.
18. Dempster A.P., Laird N.M., Rubin D.B. Maximum likelihood from incomplete data via the EM algorithm// Journal of the Royal Statistical Society. Series B (Methodological). – 1977. – Vol.39 (1). – pp. 1–38.
19. Devisscher M., De Baets B., Nopens I. Pattern discovery in intensive care data through sequence alignment of qualitative trends: proof of concept on a diuresis dataset// Appearing in the Proceedings of the ICML/UAI/COLT 2008 Workshop on Machine Learning for Health-Care Applications, Helsinki, Finland. – 2008.
20. Donald I.P., Kitchingmam G., Donald F., Kupfer, R. M. The diagnosis of small bowel bacterial overgrowth in elderly patients// J. Am. Geriatr. Soc. – 1992. – Vol.40(7). – pp. 692–696.
21. Ernst J., Nau G. J., Bar-Joseph Z. Clustering short time series gene expression data// Bioinformatics. – 2005. – Vol.21. – pp. 159–168.
22. Flores J. J., Loaeza R. Financial time series forecasting using a hybrid neural-evolutive approach// Proceedings of the XV SIGEF International Conference, Lugo, Spain. – 2009. – pp. 547–555.
23. Gardner E. S. Jr., Exponential Smoothing: The State of Art// Journal of Forecasting. – 1985. – Vol.4. – pp. 1–28.
24. Golub T. R., Slonim D. K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J. P., Coller H., Loh M. L., Downing J. R., Caligiuri M. A., Bloomfield C. D., Lander E. S. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring// Science. – 1999. – Vol. 286 (5439). – pp. 531–537.
25. Grabusts P. The choice of metrics for clustering algorithms// Environment. Technology. Resources: Proceedings of the 8th International Scientific and Practical Conference. – 2011. – Vol.2. – pp. 70–76.
26. Graves S.C., Kletter D.B., Hetzel W.B. A dynamic model for requirements planning with application to supply chain optimization// Operation Research. – 1998. – Vol.46(3). – pp. 35–49.

27. Hall M. A. Correlation-based feature selection for machine learning/ Doctoral Thesis, Hamilton: University of Waikato. – 1999. – 178 p.
28. Han J., Kamber M. Data Mining: Concepts and Techniques. Second Edition. – Morgan Kaufmann, Elsevier Inc., 2006. – 800 p.
29. Hsieh C. H., Anderson C., Sugihara G. Extending nonlinear analysis to short ecological time series// Am Nat. – 2008. – Vol.171(1). – pp. 71-80.
30. Kirshners A. Clustering-based behavioural analysis of biological objects// Environment. Technology. Resources: Proceedings of the 8th International Scientific and Practical Conference. – 2011. – Vol.2. – pp. 24–32.
31. Kirshners A., Borisov A. A Comparative analysis of short time series processing methods// Scientific Journal of Riga Technical University, Information Technology and Management Science. – 2012. – Vol.15. – pp. 65–69.
32. Kirshners A., Borisov A. Analysis of short time series in gene expression tasks// Scientific Proceedings of Riga Technical University, Information Technology and Management Science. – 2010. – Iss.5, Vol.44. – pp. 144–149.
33. Kirshners A., Borisov A. Multilevel classifier use in a prediction task// Proceedings of the 17th International Conference on Soft Computing. – 2011. – pp. 403–410.
34. Kirshners A., Borisov A. Processing short time series with data mining methods// Scientific Proceedings of Riga Technical University, Information Technology and Management Science. – 2011. – Iss.5, Vol.49. – pp. 91–96.
35. Kirshners A., Borisov A., Parshutin S. Robust cluster analysis in forecasting task// Proceedings of the 5th International Conference on Applied Information and Communication Technologies (AICT2012). – 2012. – pp. 77–81.
36. Kirshners A., Kornienko Y. Time-series data mining for e-service application analysis// Scientific Proceedings of Riga Technical University, Information Technology and Management Science. – 2009. – Iss.5, Vol.40. – pp. 94–100.
37. Kirshners A., Kuleshova G., Borisov A. Demand forecasting based on the set of short time series// Scientific Proceedings of Riga Technical University, Information Technology and Management Science. – 2010. – Iss.5, Vol.44. – pp. 130–137.
38. Kirshners A., Liepinsh E., Parshutin S., Kuka J., Borisov A. Risk prediction system for pharmacological problems// Automatic Control and Computer Sciences. – 2012. – Vol.46, No.2. – pp. 57–65.

39. Kirshners A., Parshutin S. Application of data mining methods in detecting of bacteria proliferation syndrome in the small intestine// In: European Conference on Data Analysis 2013: Book of Abstracts: European Conference on Data Analysis 2013. – 2013. – pp. 139–139.
40. Kirshners A., Parshutin S., Borisov A. Combining clustering and a decision tree classifier in a forecasting task// Automatic Control and Computer Science. – 2010. – Vol.44, No.3. – pp. 124–132.
41. Kirshners A., Parshutin S., Leja M. Research in application of data mining methods to diagnosing gastric cancer// LNAI 7377. Proceedings of the 12th Industrial Conference on Data Mining ICDM'2012. – 2012. – pp. 24–37.
42. Kirshners A., Sukov A. Rule induction for forecasting transition points in product life cycle data// Scientific Proceedings of Riga Technical University, Information Technology and Management Sciences. – 2008. – Iss.5, Vol.36 – pp. 170–177.
43. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection// Proceedings of the 14th International Conference on Artificial Intelligence (IJCAI-95). – San Mateo, CA: Morgan Kaufman. – 1995. – pp. 1137–1143.
44. Kohavi R., Quinlan J. R. Decision-tree discovery// Handbook of Data Mining and Knowledge Discovery. – Klosgen W., Zytkow J.M., Eds. – Oxford: Oxford University Press. – 2002. – pp. 267–276.
45. Koza J. R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. – Cambridge, MA: The MIT Press, 1992. – 840 p.
46. Lembcke B. Atemtests bei Darmkrankheiten und in der gastroenterologischen Funktionsdiagnostik. Schweiz// Rundschau Medizin (Praxis). – 1997. – Vol.86. – pp. 1060–1067.
47. Liepinsh E., Vilskersts R., Loca D., Kirjanova O., Pugovichs O., Kalvinsh I., Dambrova M. Mildronate, an inhibitor of carnitine biosynthesis, induces an increase in gamma-butyrobetaine contents and cardioprotection in isolated rat heart infarction// J Cardiovasc Pharmacol. – 2006. – Vol.48(6). – pp. 314–319.
48. Liepinsh E., Vilskersts R., Skapare E., Svalbe B., Kuka J., Cirule H. et al. Mildronate decreases carnitine availability and up-regulates glucose uptake and related gene expression in the mouse heart// Life Sci. – 2008. – Vol.83. – pp. 613–619.
49. Liepinsh E., Vilskersts R., Zvejniece L., Svalbe B., Skapare E., Kuka J. et al. Protective effects of mildronate in an experimental model of type 2 diabetes in Goto-Kakizaki rats// British Journal of Pharmacology. – 2009. – Vol.157. – pp. 1549–1556.

50. Lupascu A., Gabrielli M., Lauritano E. C., Scarpellini E., Scantoliquido A., Cammarota G., Flore R., Tondi P., Pola P., Gasbarrini G.,Gasbarrini A. Hydrogen glucose breath test to detect small intestinal bacterial overgrowth: a prevalence case-control study in irritable bowel syndrome// Aliment Pharmacol Ther. – 2005. – 22(11–12). – pp. 1157–1160.
51. McLachlan G., Krishnan T. The EM algorithm and extensions, 2nd edition. Wiley series in probability and statistics. – John Wiley & Sons, 2008. – 400 p.
52. Montgomery D. C., Jennings C. L., Kulachi M. Introduction to Time Series Analysis and Forecasting. – Wiley-interscience, 2008. – 445 p.
53. Parshutin S., Aleksejeva L., Borisov A. Forecasting product life cycle phase transition points with modular neural networks based system// Proceedings of 9th Industrial Conference on Data Mining ICDM`2009, Springer-Verlag. – 2009. – LNAI 5633. – pp. 88–102.
54. Parshutin S., Kirshners A. Intelligent agent technology in modern production and trade management// Efficient Decision Support Systems: Practice and Challenges – From Current to Future/ Book Chapter. INTECH. – 2011. – pp. 21–42.
55. Parshutin S., Kirshners A. Research on clinical decision support systems development for atrophic gastritis screening// Expert Systems with Applications. – 2013. – Vol.40, Iss.15. – pp. 6041–6046.
56. Posserud I., Stotzer P. O., Bjornsson E. S., Abrahamsson H., Simren M. Small intestinal bacterial overgrowth in patients with irritable bowel syndrome// Gut. – 2007. – Vol.56(6). – pp. 802–808.
57. Pyle D. Data Preparation for Data Mining. – San Francisco etc.: Morgan Kaufmann, 1999. - 540 p.
58. Quinlan J. R. C4.5: Programs for Machine Learning. – San Mateo: Morgan Kaufmann Pub., 1993. – 302 p.
59. Russell S. J., Norvig P. Artificial Intelligence: A Modern Approach. – Prentice-Hall, Inc., 1995. – 932 p.
60. Salam A. Najim, Zakaria A. M. Al-Omari, Samir M. Said. On the application of artificial neural network in analyzing and studying daily loads of Jordan power system plant// Computer Science and Information Systems. – 2008, – Vol.5, Iss.1. – pp. 127–136.
61. Sjakste N., Gutcaits A., Kalvinsh I. Mildronate: An antiischemic drug for neurological indications// CNS Drug Reviews. – 2005. – Vol.11(2). – pp. 151–168.
62. Starzyk, J. A., Haibo H., Yue L. A Hierarchical Self-organizing Associative Memory for Machine Learning// Advances in Neural Networks. – 2007. – pp. 413–423.

63. Tan P. N., Steinbach M., Kumar V. Introduction to Data Mining. – Boston: Pearson Addison-Wesley, 2006. – 769 p.
64. Thomassey S., Fiordaliso A. A hybrid sales forecasting system based on clustering and decision trees// Decision Support Systems. – 2006. – Vol.42, Iss.1. – pp. 408–421.
65. Thomassey S., Happiette M. A neural clustering and classification system for sales forecasting of new apparel items// Applied Soft Comp. – 2007. – Vol.7. – pp. 1177–1187.
66. Thomassey S., Happiette M., Castelain J. A global forecasting support system adapted to textile distribution// International Journal of Production Economics. – 2005. – Vol.96. – pp. 81–95.
67. Thomassey S., Happiette M., Castelain J. A short and mean - term automatic forecasting system – application to textile logistics// European Journal of Operations Research. – 2005. – Vol.161. – pp. 275–284.
68. Toshniwal D., Joshi R. C. Similarity search in time series data using time weighted slopes// Informatica, An International Journal of Computing and Informatics. – 2005. – Vol.29, No.1. – pp. 79–88.
69. Ward J. H., Jr. Hierarchical Grouping to Optimize an Objective Function// Journal of the American Statistical Association. – 1963. – Vol.58. – pp. 236–244.
70. Witten I. H., Frank E. Data mining: Practical machine learning tools and techniques (Second edition). – San Francisco, CA: Morgan Kaufmann, 2005. – 560 p.
71. Wiener N. Cybernetics: Or Control and Communication in the Animal and the Machine. – Boston, MA: Technology Press, 1948. – 219 p.
72. Wang X., Wu M., Li Z., Chan C. Short time-series microarray analysis: Methods and challenges// BMC Systems Biology – 2008. – 2:58. – pp. 1–6.
73. Wu X., Kumar V., Quinlan J.R., et al. Top 10 algorithms in data mining// Knowl. Inf. Syst. – 2007. – 14. – pp. 1–37.
74. Zhu W., Zeng. N., Wang N. Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementation// NESUG Proceedings: Health Care and Life Sciences, Baltimore, Maryland. – 2010. – pp. 1–9.
75. Zurada J. M. Introduction to Artificial Neural Systems. – West: St. Paul, MN, 1992. – 679 p.
76. Барсегян А., Куприянов М., Степаненко В. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP – Санкт-Петербург: БХВ – Петербург, 2007. – 384 с.
77. Гринглаз Л., Копытов Е. Математическая статистика с примерами решения на компьютере: Учеб. Пособие. – 2-е изд. – Рига: ВШЭК, 2002. – 326 с.

PIELIKUMI

1. pielikums

PPS klastera piederības noteikšana testēšanas kopas datiem

Klasterizācija testēšanas datu kopai ar *modificētu k-vidējo sadalošo* algoritmu tika veikta ar normalizētiem testēšanas kopas pieprasījuma datiem (skatīt P1.1. attēlu), kur parādīts sešu objektu pieprasījums un apmācības kopas klasterizācijas rezultātā iegūto desmit klasteru (C1–C10) vidējās vērtības katrā laika periodā. Ar šī parauga palīdzību, tiks parādīts aprēķinu process, lai noteiktu klasteru piederību testēšanas kopas datiem.

No	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
1	0,0364	0,0364	0,0182	0,0727	0,0545	0,0364	0,0364	0,0182	0,0727	0,1273	0,2182	0,2727
2	0,0000	0,0000	0,0000	0,0000	0,0000	0,3333	0,0000	0,0000	0,6667	0,0000	0,0000	0,0000
3	0,0000	0,1250	0,2500	0,2500	0,1250	0,1250	0,1250	0,0000	0,0000	0,0000	0,0000	0,0000
4	0,0526	0,1053	0,0526	0,0526	0,0526	0,1053	0,1053	0,0526	0,0526	0,1053	0,1053	0,1579
5	0,0000	0,0000	0,0000	0,1667	0,8333	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
6	0,3333	0,0000	0,0000	0,3333	0,0000	0,0000	0,0000	0,0000	0,3333	0,0000	0,0000	0,0000
C1	0,0660	0,0668	0,0809	0,0928	0,0980	0,0746	0,0602	0,0829	0,1100	0,0869	0,0793	0,1016
C2	0,0008	0,0005	0,0188	0,0419	0,2044	0,2826	0,2475	0,1414	0,0282	0,0191	0,0106	0,0042
C3	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0083	0,0306	0,1188	0,3541	0,2590	0,2291
C4	0,0000	0,0033	0,0000	0,0033	0,0000	0,0000	0,0065	0,0509	0,0520	0,1112	0,2470	0,5260
C5	0,0461	0,0505	0,0630	0,0927	0,1214	0,1402	0,1300	0,1138	0,0885	0,0538	0,0494	0,0508
C6	0,0021	0,0039	0,0004	0,0100	0,0070	0,0317	0,1556	0,2889	0,2401	0,1370	0,0692	0,0541
C7	0,0000	0,0000	0,0000	0,0022	0,0000	0,0000	0,0000	0,0722	0,3571	0,2064	0,2280	0,1342
C8	0,0000	0,0090	0,0954	0,2740	0,3344	0,0841	0,0459	0,0611	0,0470	0,0093	0,0130	0,0269
C9	0,0970	0,1602	0,2020	0,1803	0,0486	0,0747	0,0227	0,0501	0,0451	0,0422	0,0362	0,0410
C10	0,1485	0,1193	0,0869	0,0597	0,0411	0,0381	0,0340	0,0445	0,0832	0,1012	0,1109	0,1326

P1.1. att. Testēšanas datu kopas paraugs ar 6 objektu normalizētām vērtībām un apmācības kopas klasterizācijas procesā noteiktās 10 klasteru vidējas vērtības (C1-C10)

Sākotnēji, pēc formulas (2.17) un apakšpunktā 2.3.4.3. aprakstītās metodoloģijas tika aprēķināts testēšanas datu kopas katra objekta attālums līdz apmācības kopas attiecīgā paraugmodeļa vidējai vērtībai (skatīt P1.1. attēlu vērtības ar pelēko fonu). Attāluma aprēķinu process testēšanas kopas pirmā objekta attālums $d_{1,C1}$ līdz pirmajam paraugmodelim (iegūts apmācības procesā) un pirmā objekta attālums $d_{1,C2}$ līdz otrajam paraugmodelim (iegūts apmācības procesā) bija šāds:

$$d_{1,C1} = \sqrt{\begin{array}{l}(0,0364-0,066)^2+(0,0364-0,0668)^2+(0,0182-0,0809)^2+\\+(0,0727-0,0928)^2+(0,0545-0,098)^2+(0,0364-0,0746)^2+\\+(0,0364-0,0602)^2+(0,0182-0,0829)^2+(0,0727-0,11)^2+\\+(0,1273-0,0869)^2+(0,2182-0,0793)^2+(0,2727-0,1016)^2\end{array}} = 0,2566;$$

$$d_{1,C2} = \sqrt{\begin{matrix}(0,0364-0,0008)^2+(0,0364-0,0005)^2+(0,0182-0,0188)^2+\\+(0,0727-0,0419)^2+(0,0545-0,2044)^2+(0,0364-0,2826)^2+\\+(0,0364-0,2475)^2+(0,0182-0,1414)^2+(0,0727-0,0282)^2+\\+(0,1273-0,0191)^2+(0,2182-0,0106)^2+(0,2727-0,0042)^2\end{matrix}} = 0,5246.$$

Līdzīgā veidā tika aprēķināti visi pārējie objektu attālumi, līdz attiecīgā klastera paraugmodelim. Iegūtos aprēķinu rezultātus (skatīt. P1.2. attēlā), kur ar sarkano krāsu ir izcelta katra objekta minimālā attāluma vērtība līdz attiecīgā klastera vidējai vērtībai (paraugmodelim) apmācības datu kopā. Šīs izceltās vērtības, zem attiecīgā klastera numura, arī kļūst par šī objekta klasterizācijas rezultātā iegūto klasi.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	MIN()	Class
d1	0,2566	0,5246	0,2658	0,2819	0,3453	0,4408	0,3506	0,4982	0,4024	0,2391	0,2391	C10
d2	0,6671	0,7321	0,8107	0,9174	0,6643	0,6381	0,5698	0,8056	0,7562	0,7211	0,5698	C7
d3	0,3341	0,4239	0,6675	0,7340	0,3104	0,5707	0,6571	0,3108	0,2144	0,3960	0,2144	C9
d4	0,1323	0,3717	0,3717	0,4445	0,1832	0,3625	0,4038	0,4257	0,2717	0,1484	0,1323	C1
d5	0,7834	0,7575	0,9911	1,0375	0,7657	0,9488	0,9833	0,5345	0,8410	0,8570	0,5345	C8
d6	0,4899	0,7017	0,7171	0,8074	0,5282	0,5985	0,5823	0,5758	0,4946	0,4897	0,4897	C10

P1.2. att. Testēšanas datu kopas aprēķinātie objektu attālumi līdz apmācības datu kopas paraugmodeļu (C1-C10) vērtībām

Pēc klases piešķiršanas testēšanas kopas objektiem tika veikta objektu šķirošana pēc noteiktās klases vērtības. Katram klases objektam tika izskaitļota vidējā absolūtā novirze *MAD* (skatīt P1.3. attēlu), kur ar dzeltenu krāsu parādīti apmācības datu kopas klasterizācijā iegūtais paraugmodelis ar desmit klasteriem, kuru skaits tika noteiks eksperimentāli. Vidējo absolūto novirzi aprēķina pēc formulas (2.18), kuras aprēķina piemērs pirmajam objektam, ir parādīts zemāk:

CLASS	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	MAD	MAE
C10	0,0364	0,0364	0,0182	0,0727	0,0545	0,0364	0,0364	0,0182	0,0727	0,1273	0,2182	0,2727	0,0504	
C10	0,3333	0,0000	0,0000	0,3333	0,0000	0,0000	0,0000	0,0000	0,3333	0,0000	0,0000	0,0000	0,1181	
C10	0,1200	0,2400	0,0800	0,0400	0,0800	0,0800	0,1200	0,0400	0,0400	0,0400	0,0800	0,0400	0,0479	
C10	0,2000	0,2000	0,1000	0,0000	0,0000	0,0000	0,2000	0,0000	0,0000	0,2000	0,0000	0,1000	0,0684	
C10	0,2400	0,0800	0,0400	0,0800	0,2000	0,0400	0,0400	0,0000	0,0000	0,0800	0,0400	0,1600	0,0510	
C10	0,1075	0,1290	0,0968	0,0538	0,0323	0,0538	0,0753	0,0645	0,1398	0,0968	0,0753	0,0753	0,0255	
C10	0,1429	0,1429	0,1429	0,0000	0,0000	0,0000	0,0000	0,2857	0,1429	0,0000	0,0000	0,1429	0,0651	
C10	0,3750	0,0000	0,0000	0,0000	0,0000	0,0000	0,1250	0,1250	0,1250	0,2500	0,0000	0,0000	0,0981	
C10	0,4286	0,0000	0,0000	0,1429	0,1429	0,0000	0,1429	0,1429	0,0000	0,0000	0,0000	0,0000	0,1120	
C10	0,0645	0,0968	0,0323	0,0645	0,0323	0,0323	0,0323	0,0645	0,1290	0,1613	0,1935	0,0968	0,0356	
C10	0,2500	0,1250	0,0000	0,0000	0,0000	0,0000	0,0000	0,1250	0,1250	0,2500	0,1250	0,0000	0,0654	
C10	0,1429	0,1429	0,0536	0,0714	0,0714	0,0357	0,0714	0,0357	0,1250	0,0893	0,0536	0,1071	0,0241	
C10	0,1429	0,1071	0,0714	0,0357	0,0714	0,0714	0,0357	0,0357	0,1786	0,1071	0,0714	0,0714	0,0278	
	0,1485	0,1193	0,0869	0,0597	0,0411	0,0381	0,0340	0,0445	0,0832	0,1012	0,1109	0,1326	0,7894	0,061

P1.3. att. Testa datu kopai klasterizācijas rezultātā piešķirtais klastera numurs

$$MAD = \frac{\begin{array}{c}|0,1485-0,0364|+|0,1193-0,0364|+|0,0869-0,0182|+\\+|0,0597-0,0727|+|0,0411-0,0545|+|0,0381-0,0364|+\\+|0,034-0,0364|+|0,0445-0,0182|+|0,0832-0,0727|+\\+|0,1012-0,1273|+|0,1109-0,2182|+|0,1326-0,2727|\end{array}}{12} = 0,0504.$$

Tālāk pēc formulas (2.19) tika aprēķināta vidējā absolūtā kļūda (*MAE*) katrā klasterī, kuras vērtība sastāda 0,061 (skatīt P1.3. attēlu, kur kļūdas novērtējums attēlots ar sarkanu krāsu). Pilnus vidējās absolūtās novirzes un vidējās absolūtās kļūdas aprēķinus katrā no klasterim skatīt P1.1. tabulā.

P1.1. tabula

Vidējās absolūtās novirzes un vidējās absolūtās kļūdas aprēķini klasteros

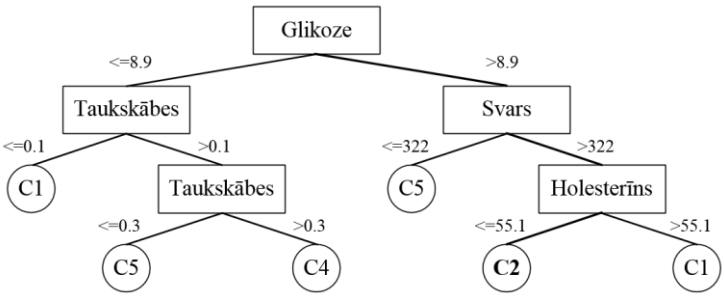
	Klasteris									
	1	2	3	4	5	6	7	8	9	10
Ierakstu skaits	28	15	3	3	26	16	10	12	23	13
Vidējā absolūtās novirzes summa katrā klasterī	1,010	0,963	0,115	0,101	1,207	1,076	0,576	0,770	1,779	0,789
Vidējā absolūtā kļūda	0,036	0,064	0,038	0,034	0,046	0,067	0,058	0,064	0,077	0,061

Pēc vidējās absolūtās kļūdas aprēķina var noteikt kopējo absolūto kļūdu (*TAE*) pēc formulas (2.20), kuras aprēķins parādīts zemāk:

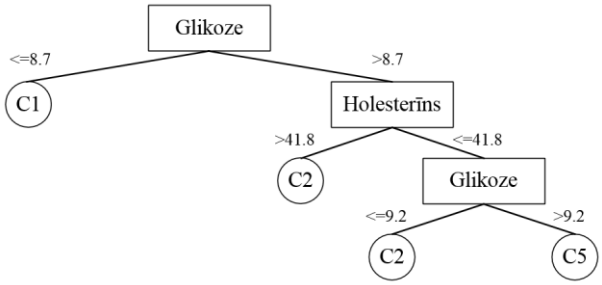
$$TAE = \frac{\begin{array}{c}0,036+0,064+0,038+0,034+0,046+\\+0,067+0,058+0,064+0,077+0,061\end{array}}{10} = 0,0546.$$

2. pielikums

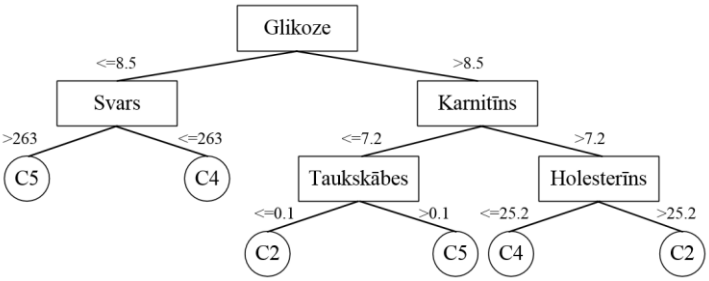
SNRPS klasifikatora iegūtie rezultāti dažādām apakškopām



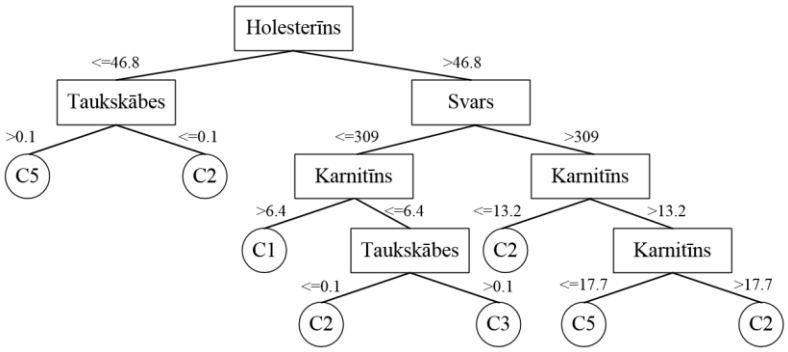
P2.1. att. Klasifikācijas rezultātā iegūtais induktīvais lēmuma koks 2. apakškopai



P2.2. att. Klasifikācijas rezultātā iegūtais induktīvais lēmuma koks 3. apakškopai



P2.3. att. Klasifikācijas rezultātā iegūtais induktīvais lēmuma koks 4. apakškopai



P2.4. att. Klasifikācijas rezultātā iegūtais induktīvais lēmuma koks 5. apakškopai