KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Mantas Gaižauskas

Prognozavimo metodų analizės sistema

Magistro darbas

Darbo vadovas
prof. L. Nemuraitė

Kaunas, 2009

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Mantas Gaižauskas

Prognozavimo metodų analizės sistema

Magistro darbas

Recenzentas
dr. Audronė Janavičiūtė
2009-05-25

Vadovas
prof. L. Nemuraitė
2009-05-25

Atliko
IFM-3 gr. stud.
Mantas Gaižauskas
2009-05-25

Kaunas, 2009

TURINYS

1. ĮVADAS

Duomenų gavyba yra šiuolaikinė informacijos analizės sritis, atsiradusi duomenų bazių technologijų, dirbtinio intelekto ir statistinės duomenų analizės sankirtoje. Duomenų gavyba yra labai plati sritis, apimanti daug jai skirtų metodų, algoritmų bei taikomųjų sistemų. Duomenų gavyba naudojama daugelyje sričių, kuriose siekiama pasiekti verslo tikslus, konkurencinį pranašumą, tikslius ir pagrįstus sprendimus, pasiekti geresnių darbo rezultatų.

Duomenų analizavimas gali turėti lemiamos įtakos kompanijos augimui, ligų diagnozavimui ir mokslo atradimams. Rankiniai metodai žinioms iš duomenų išgauti naudojami jau šimtmečius, tačiau atsiradus duomenų gavybos procesui paplito automatiniai metodai. Ir duomenų, ir iš duomenų išgaunamos informacijos apimtys nuolat didėja, todėl tiesioginė duomenų analizė palaipsniui pakeičiama netiesioginiu, automatizuotu duomenų apdorojimu, naudojant sudėtingesnes ir modernesnes priemones, metodus ir modelius. Nuolat besikeičiant ir didėjant duomenų kiekiui sistemose, tik automatizuotos mokymosi sistemos gali padėti išspręsti žinių paieškos ir sprendimų priėmimo problemas.

Duomenų gavybos technologijos leidžia aptikti duomenų saugyklose esančių duomenų dėsningumus, panašumus ir atlikti duomenų prognozavimą. Duomenų gavybos technologijomis realizuotais algoritmais sprendžiamus uždavinius pagal sprendimo būdus galima skirstyti į tokias grupes: asociacijos, regresijos bei klasifikavimo. Klasifikavimo uždavinių grupė apibrėžiama kaip duomenų atvejų narystės grupėje prognozavimas.

Atsiradus naujiems statistiniams ir mokymosi algoritmams vis dažniau kyla klausimas, kaip efektyviai panaudoti algoritmus duomenų analizei ir organizacijų priimamų sprendimų bei strategijų kokybės gerinimui. Platus algoritmų spektras leidžia duomenų gavybą panaudoti daugelyje sričių – duomenų gavybos metodologija gali būti taikoma ne tik ten, kur sprendžiamos duomenų klasifikacijos ir ryšių tarp duomenų bei informacinių modelių identifikavimo problemos, bet ir numatomos galimybės dirbti su meta duomenimis, perspektyviose duomenų sistemose.

Magistro darbo tyrimo sritis yra duomenų gavyba, konkrečiau – klasifikuojantys prognozavimo algoritmai, jų taikymas bei panaudojimas, kuriant naujas, tobulesnes organizacijose naudojamas veiklos intelekto sistemas. Tyrimo objektas – klasifikavimo / prognozavimo algoritmų įvertinimo, parinkimo ir parengimo veikti informacinėje sistemoje procesas.

Darbo tikslas – sukurti veiklos intelekto technologijomis pagrįstą prognozavimo metodų analizės sistemą, kuria parodoma, kaip darbui informacinėse sistemose turėtų būti sukuriami, analizuojami, parenkami ir paruošiami prognozavimo metodai.

Šiam tikslui pasiekti reikėjo išspręsti šiuos uždavinius:

- atlikti duomenų gavybos metodų, algoritmų ir jų taikymo aspektų analizę;
- atlikti duomenų gavybos technologijų, procesų ir architektūros analizę;
- sukurti algoritmų vertinimo ir palyginimo sistemos modelį;
- realizuoti sistemą algoritmams palyginti;
- atlikti eksperimentą, pritaikant sistemą konkrečiam prognozavimo uždaviniui.

Norint tinkamai panaudoti ir palyginti duomenų gavybos algoritmus, išanalizuota literatūra, susijusi su duomenų gavybos algoritmų savybėmis ir taikymo aspektais. Taip pat analizuota literatūra, kurioje pateikiama metodika, kaip turėtų būti atliekamas algoritmų įvertinimas. Renkantis vieną ar kitą duomenų gavybos algoritmą, atsižvelgiama į turimus duomenis. Duomenims keičiantis, sukurtos informacinės sistemos, kuriose naudojami prognozavimo algoritmai, veikimas gali neatitikti pasikeitusios situacijos duomenyse, todėl gali prireikti kartoti duomenų gavybos procesą. Taip pat gali tekti perrinkti ir kitą duomenų gavybos algoritmą. Todėl taip pat analizuojama su duomenų gavybos procesais susijusi literatūra [2, 7, 8, 14, 15].

Veiklos intelekto technologijos kuriamos remiantis standartizuotais duomenų gavybos procesais: SEMMA, KDD, CRISP-DM. Remiantis šiais procesais Microsoft, Oracle, IBM, SAP sukūrė technologijas veiklos intelekto sprendimams. Kiekviena iš šių kompanijų duomenų gavybos algoritmų taikymui pateikia skirtingą technologijos architektūrą. Kuriant sistemas, kuriose norima panaudoti prognozavimo algoritmus, svarbu išsiaiškinti veiklos intelekto technologijų architektūrą. Tam išanalizuota literatūra [9, 10, 11, 12, 13] susijusi su veiklos intelekto priemonių technologiniais aspektais.

Pirmoje darbo dalyje atliekama duomenų gavybos proceso, algoritmų savybių, algoritmų įvertinimo ir testavimo metodikų bei veiklos intelekto technologijų analizė.

Antroje darbo dalyje aprašomas prognozavimo algoritmų palyginimo sistemos koncepcinis modelis. Šioje darbo dalyje pagal pirmoje dalyje išanalizuotą informaciją aprašoma sukurta algoritmų palyginimo sistemos duomenų bazės schema, architektūra ir detalizuotos panaudos atvejų diagramos. Algoritmų palyginimo sistemos realizacijai buvo pasirinkta Microsoft Analysis Services 2008 duomenų gavybos, Microsoft SQL Server 2008 ir ASP.NET technologijos. Palyginimo sistemos duomenų bazės schema sukurta atsižvelgiant į Microsoft Analysis services technologinius darbo su algoritmų modeliais DMX kalbos aspektus ir dažniausiai naudojama prognozavimo algoritmų įvertinimo metodika. Šioje dalyje, remiantis prognozavimo algoritmų įvertinimo metodika, pateikiami DMX ir SQL darbo su prognozavimo algoritmais šablonų pavyzdžiai, sukurti konkrečiam klasifikavimo uždaviniui. Taip pat aprašomas sistemos įvertinimo skaičiavimo panaudojant šablonus

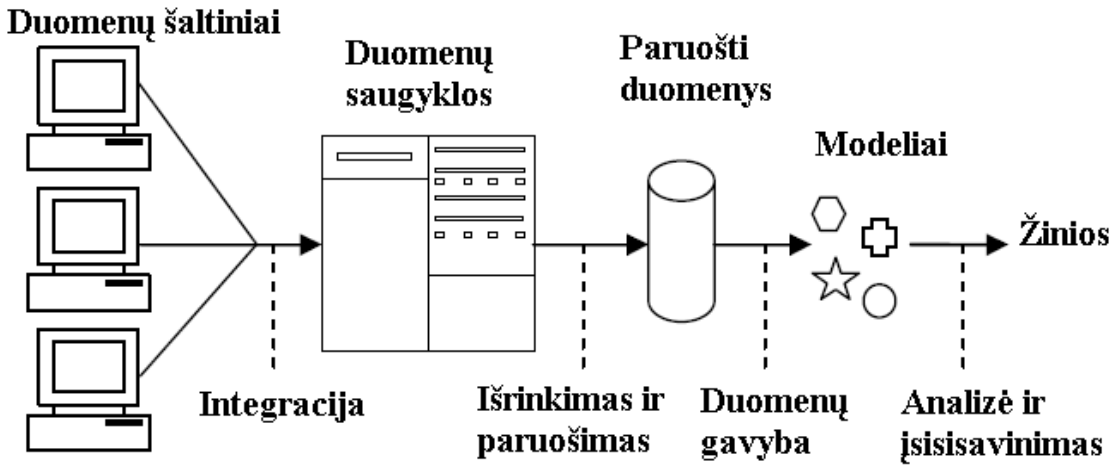
veikimo principas.

Trečioje darbo dalyje sukurta algoritmų palyginimo sistema pritaikoma konkretaus klasifikavimo uždavinio prognozavimo algoritmo išrinkimui. Pagal koncepcinėje darbo dalyje aprašytą sistemos modelį sukurta sistema išbandoma pacientų, besiskundžiančių kraujotakos ligomis, klasifikavimo uždaviniui. Klasifikavimo algoritmai suskirsto pacientus į dvi klases „Sveiki“ ir „Sergantys“. Remiantis įvertinimo metodika ir atsižvelgiant į pacientų klasifikavimo uždavinį, sukurti pavyzdiniai prognozavimo algoritmų įvertinimo šablonai. Pagal šiuos šablonus sukurta sistema skaičiuoja įvertinimus, reikalingus prognozavimo algoritmams palyginti. Išanalizavus sistemos grafiškai pateiktus įvertinimo duomenis, išrenkamas geriausiai klasifikavimo uždaviniui tinkantis algoritmas ir nustatomas algoritmo apmokymo duomenų kiekis, būtinas jo taikymui informacinėje sistemoje.

2. DUOMENŲ GAVYBOS PROCESO ANALIZĖ

2.1 Duomenų gavybos procesai

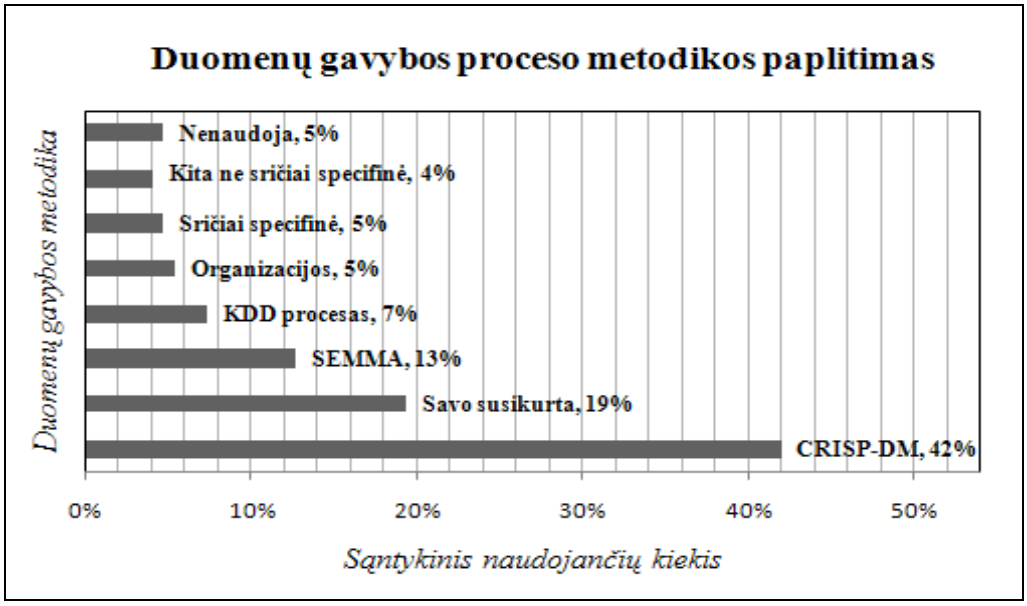
Duomenų gavyba apibrėžiama kaip anksčiau nežinotų ir galimai naudingų žinių išgavimas iš duomenų.



2.1 pav. Žinių suradimo procesas

Kaip matyti 2.1 paveiksle, duomenys į duomenų saugyklas surenkami iš įvairių šaltinių. Duomenys išrenkami ir pervedami į standartinę formą. Paruošti duomenys perduodami duomenų gavybos algoritmui, kuris sugeneruoja rezultatus taisyklių ar tam tikrų modelių pavidalu. Rezultatai analizuojami naujų ir galimai naudingų žinių atradimui.

Duomenų gavybos analizė dažniausiai taikoma naudojant vieną iš kelių duomenų gavybos procesų metodikų. Pagal tinklapyje www.kdnuggets.lt pateiktus apklausos duomenis sudaryta populiariausių duomenų gavybos metodikų diagrama patiekiama 2.2 paveiksle.



2.2 pav. Populiariausios duomenų gavybos metodikos

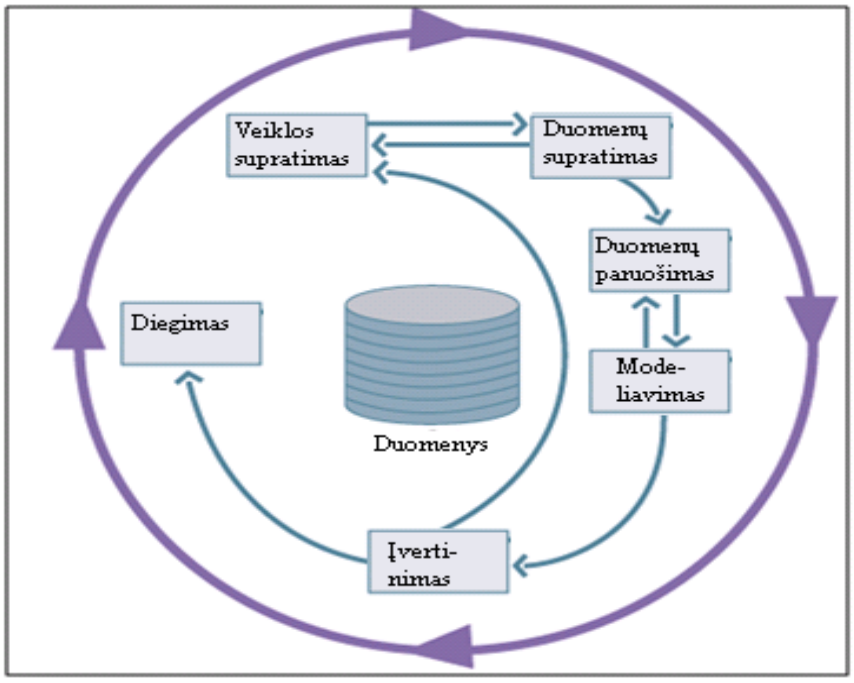
Šaltinis: Sudaryta pagal KDNuggets tinklapio apklausos duomenis

Pagal 2.2 paveiksle pateiktą diagramą galima spręsti apie populiariausias duomenų gavybos metodikas.

Kaip rodo tinklapio www.kdnuggets.lt lankytojų apklausa, gana dažnas yra sistemingos metodikos naudojimas. Kadangi šiuo metu populiariausia laikoma CRISP-DM metodika, ją paanalizuoti reikėtų detaliau.

2.1.1 CRISP-DM duomenų gavybos metodika

Dažniausiai CRISP-DM metodika duomenų gavybai naudojama pramonėje. Tinklapyje www.crisp-dm.org aprašytame modelyje ciklišk ai naudojami šeši etapai pavaizduoti 2.3 paveiksle.



2.3 pav. CRISP-DM proceso modelio etapai

Kaip matyti 2.3 paveiksle, CRISP-DM proceso modelį sudaro šie etapai:

1. Veiklos supratimas. Šis etapas susideda iš esamos situacijos įvertinimo, tikslų nustatymo, duomenų gavybos tikslų nustatymo ir projekto plano sudarymo;

2. Duomenų supratimas. Šiame etape nustatomi veiklos tikslai ir sukuriamas projekto planas. Reikalavimai duomenims nustatomi duomenų analizės metu. Šis etapas gali susidėti iš pradinių duomenų surinkimo, duomenų apibrėžimo, duomenų analizės ir duomenų kokybės įvertinimo.

3. Duomenų paruošimas. Šis etapas atliekamas nustačius ir surinkus duomenis iš šaltinių. Duomenys turi būti išrinkti, išvalyti, sutvarkyti ir suformatuoti. Prieš atliekant duomenų modeliavimą atliekamas neteisingų ir netikslių duomenų pašalinimas, taisymas ir transformavimas. Šiame etape, panaudojant papildomus duomenų gavybos modelius, detaliau gali būti atliekama duomenų analizė ir remiantis veiklos supratimu duomenyse nustatomi šablonai;

4. Modeliavimas. Šiame etape pradinei analizei dažnai naudojami duomenų vizualizavimo, klasterių analizės įrankiai. Tokie įrankiai kaip apibendrintos taisyklių indukcijos gali pateikti pradines asociacijų taisykles. Kai įgyjamas didesnis duomenų supratimas (įprastai peržiūrint modelio rezultatus ir aptinkant šablonus duomenyse), gali būti pritaikyti detalesni duomenims tinkantys duomenų gavybos modeliai. Taip pat šiame etape duomenys suskirstomi į apmokymo ir testavimo duomenų grupes;

5. Įvertinimas. Modelio rezultatai turi būti įvertinti atsižvelgiant į veiklos tikslus, nustatytus pirmame etape. Tai leidžia nustatyti kitus poreikius, dažnai grįžtant prie pradinių CRISP-DM etapų. Duomenų gavyboje veiklos supratimas yra iteracinis procesas, kurio metu rezultatai gaunami atliekant įvairias peržiūras, taikant statistinius ir dirbtinio intelekto įrankius. Naudojant duomenų gavybos įrankius vartotojui pateikiami nauji sąryšiai, kurie praplečia veiklos taisyklių supratimą;

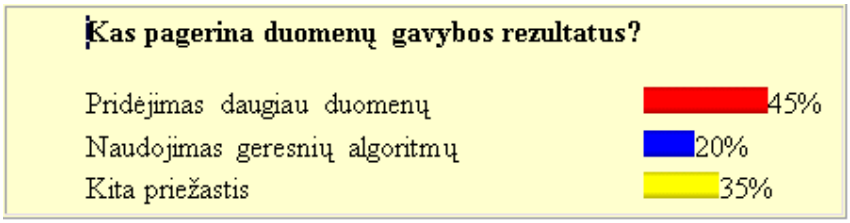
6. Diegimas. Duomenų gavyba gali būti naudojama tiek anksčiau iškeltai hipotezei patvirtinti, tiek naujoms žinioms surasti, taip pat ir netikėtiems bei naudingiems sąryšiams atrasti. Radus naują informaciją ankstesniuose etapuose, nauji modeliai gali būti sukurti ir pritaikyti veiklos operacijoms, prognozavimui. Šie modeliai turi būti prižiūrimi, kadangi keičiantis veiklos sąlygoms, keičiasi duomenys. Jei įvyksta žymūs pasikeitimai, modeliai turėtų būti įvertinti iš naujo [2].

Kaip matome iš duomenų gavybos proceso ciklo, turint istorinius duomenims nuolat yra kartojami šeši etapai, kol pasiekiamas iškeltas tikslas. Realiose informacinėse sistemose diegiant sukurtą produktą, situacija gali keistis, todėl toks produktas turėtų būti perdarytas iš naujo, atsižvelgiant į naujus pasikeitusius duomenis. Sukurtose informacinėse sistemose galėtų būti panaudoti keli duomenų gavybos modeliai. Paprastesnis modelio pasirinkimas ir įvertinimas galėtų būti atliekamas jau sukurtoje sistemoje iteraciniu būdu, atsižvelgiant į šiuo metu žinomus duomenų gavybos algoritmus ir turimus duomenis. Tokiu atveju jau sukurtoje ir įdiegtoje informacinėje sistemoje modelio parinkimas ir įvertinimas taptų ciklišku procesu.

Norint sukurti duomenų gavybos modelio parinkimo įrankį, reikia detaliau paanalizuoti modeliavimo ir įvertinimo etapuose naudojamas metodikas ir algoritmus.

2.2 Duomenų gavybos algoritmų analizė

Sistemos duomenų gavybos modelio veikimas priklauso tiek nuo duomenų, tiek nuo algoritmo. Pagal tinklapyje www.kdnuggets.lt pateiktos apklausos duomenis (2.4 pav.) 65 procentų duomenų gavybą naudojančių specialistų mano, kad labiausiai duomenų gavybą veikia algoritmo parinkimas ir duomenys.[1]



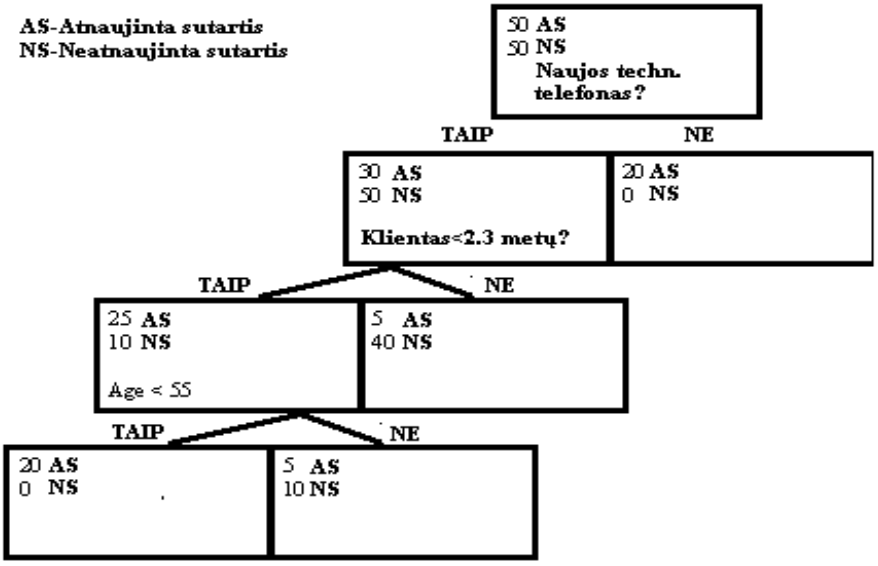
2.4 pav. Priežastys gerinančios duomenų gavybos rezultatus

Jei duomenys turi didelę įtaką rezultatams, akivaizdu, kad jų pasikeitimas turės įtakos prognozavimo ar kito duomenų gavyboje sprendžiamo uždavinio rezultatams. Kaip matome 2.4 paveiksle, penktadalis apklaustųjų teigia, kad labai svarbūs yra duomenų gavybos algoritmai. Daugelyje sistemų ar siūlomų sprendimų daugiau negu vieno algoritmo panaudojimas, priklausomai nuo kintančių duomenų, gali pagerinti sistemos rezultatus.

Akivaizdu, kad norint naudoti duomenų gavybos metodus bei algoritmus to paties uždavinio sprendimui, pirmiausiai reikia išanalizuoti algoritmus ir nustatyti, kokį uždavinį jie gali spręsti.

2.2.1 Sprendimų medžio metodas

Sprendimų medis – tai prognozavimo metodas, turintis medžio formą. Kiekviena medžio šaka yra klasifikavimo klausimas. Pavyzdžiui, jeigu klasifikuojami klientai, kurie neatnaujina savo telefonų sutarčių, sudarytas sprendimų medis atrodytų taip (2.5 pav.).



2.5 pav. Sprendimų medis

Sprendimų medis pasižymi šiomis savybėmis:

1. Išskirsto duomenis kiekvienai šakai neprarandant jokių duomenų. Bendras įrašų skaičius tam tikrame tėviniame mazge yra lygus vaikų mazguose esančių įrašų sumai;
2. Skaičius atnaujinusių sutartis ir neatnaujinusių išlieka tas pats einant medžiu žemyn ar aukštyn;
3. Labai lengva suprasti, kaip modelis sudaromas (priešingai nei neuroninių tinklų modeliai);
4. Galima intuityviai jausti klientus. Pavyzdžiui, klientai esantys tinkle keletą metų ir turintys naujus telefonus yra lojalūs, ištikimi;
5. Sprendimų medžio struktūra ir lengvas taisyklių medžio generavimas – mėgstama metodika kuriant suprantamus modelius. Dėl savo aukšto automatizavimo lygio ir paprasto sprendimų medžio pakeitimo SQL užklausomis algoritmą lengviau integruoti IT sistemose.[5]

Dauguma sprendimo medžių formuojami tokio dydžio, kad atitiktų šiuos kriterijus:

1. Segmentas turi vieną įrašą;
2. Visi įrašai segmente turi vienodas charakteristikas (nėra prasmės klausti kito klausimo, kadangi likę įrašai tokie patys);
3. Gerinti nėra pakankamai pagrindo atliekant išskyrimą.

Praktikoje sprendimų medžiams sudaryti naudojami trys pagrindiniai algoritmai: CART („Classification and Regression Trees“), C4.5 ir CHAID. Visi trys algoritmai iš duomenų formuoja taisyklėmis paremtą medį. Pagrindinis skirtumas yra medžio formavimo procesas. Visi trys algoritmai mėgina apriboti medžio dydį. CHAID medžio augimą nutraukia naudodamas statistikos sustojimo taisyklę. Priešingai CART ir C4.5 pirmiausiai augina visą medį, o vėliai atlieka medžio mažinimą. Medžio genėjimas atliekamas nagrinėjant medžio efektyvumą, naudojant apmokymo duomenis. Medis mažinamas kol pasiekiamas vienodas efektyvumas. CHAID ir C4.5 galutinio medžio suformavimui naudoja duomenų rinkinį, tuo tarpu CART medžio sumažinimui papildomai naudoja apmokymo duomenų rinkinį. CART algoritmas formuoja tik binarinius medžius, tuo tarpu CHAID ir C4.5 gali turėti daugiau nei dvi šakas.[5]

2.2.2 K- vidurkių grupavimas

K- vidurkių grupavimo (angl. K-Means Clustering) algoritmas narystę grupėje nustato minimizuodamas skirtumus tarp narių grupėje ir maksimizuodamas atstumą tarp grupių. K – vidurkių grupavimo algoritme iš anksto nustatoma, kiek grupių bus naudojama. K – nusako grupių skaičių. Tuomet k taškų atsitiktinai išrenkama grupių centrais. Skaičiuojant Euklidinį atstumą, visi taškai priskiriami artimiausią centrą turinčiai grupei. Tuomet kiekvienoje grupėje esantiems taškams apskaičiuojama centroidė arba vidurkis. Šios centroidės laikomos naujais grupių centrais. Grupėms visas procesas pakartojamas su naujais centrais. Iteracijos kartojamos tol, kol tie patys taškai priskiriami kiekvienai grupei.[14]

K-vidurkių algoritmas priskiria kiekvieną duomenų tašką vienai grupei. Priklausymas grupei išreiškiamas atstumu iki centroidės.

Įprastai k-vidurkių algoritmas naudojamas tolygias reikšmes turinčių atributų grupavimui, kur atstumas iki vidurkio skaičiuojamas tiesiogiai. Tačiau panaudojant tikimybes galima skirstyti į grupes ir diskrečius atributus.

Grupavimo metodas paprastas ir efektyvus. K - vidurkių grupavimo algoritmas efektyviai dirba, kai grupės gerai atsiskiria. Paprastai atstumų skaičiavimas k – vidurkių grupavimo algoritme atliekamas naudojant keletą iteracijų, kuriose skaičiuojant atstumus kiekvienam taškui ieškomi k grupių centrų, tačiau yra daug algoritmo modifikacijų, kurios pagreitina atstumų skaičiavimus [4].

2.2.3 Neuroninių tinklų metodas

Neuroniniai tinklai – biologinės sistemos, kurios aptinka modelius, atlieka prognozavimą ir mokymąsi. Moderniam modelių aptikimui ir prognozavimo modelių sudarymui, iš didelės apimties istorinių duomenų bazių, naudojami dirbtiniai neuroniniai tinklai. Jie realizuojami kompiuterio programomis. Nors mokslininkai dar toli nuo visiško supratimo, kaip veikia žmogaus smegenys, dirbtiniai neuroniniai tinklai jau gali kai kuriuos dalykus atlikti lygiai taip pat kaip ir žmonės.

Dažnai neuroniniais tinklais siekiama pasiekti tokį automatizavimo lygį, kad vartotojas galėtų išsiversti su minimaliomis žiniomis. Taip pat norint naudoti neuroninius tinklus siekiama, kad nereiktų tvarkyti ar modifikuoti duomenų.

Dažnai būna atvirkščiai. Norint veiksmingai panaudoti neuroninius tinklus reikia atlikti gana daug svarbių projektinių sprendimų. Pavyzdžiui:

- Kaip neuroniniame tinkle turėtų būti sujungti mazgai?
- Kiek neuroninių apdorojimo mazgų turėtų būti naudojama?
- Kada apmokymas turėtų būti sustabdytas, norint išvengti nereikalingų parametrų?

Taip pat yra daug svarbių žingsnių, reikalingų išankstiniam duomenų paruošimui. Pavyzdžiui, dažnai reikia normalizuoti skaitinius duomenis nuo 0.0 iki 1.0.

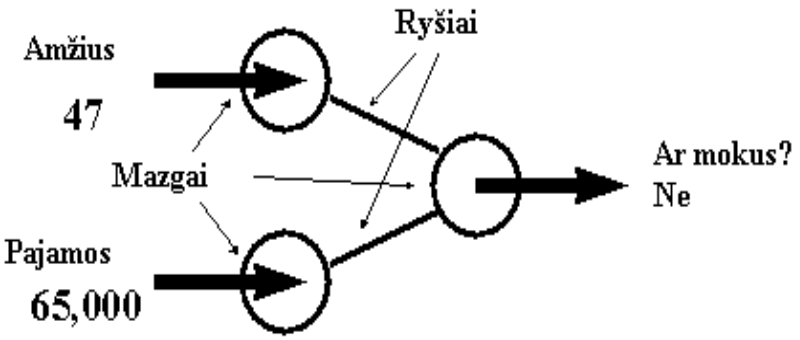
Neuroniniai tinklai puikiai tinka prognozavimo modeliavimui, bet juos sudėtinga naudoti. Neuroniniais tinklais galima sukurti labai sudėtingus modelius, kuriuos beveik visada sunku suprasti net ekspertams. Šiame modelyje skaičiavimai paremti skaitinėmis reikšmėmis, todėl duomenis reikia pateikti skaičių formatu. Neuroniniu tinklu prognozuoti rezultatai yra skaitiniai, todėl juos reikia konvertuoti į suprantamą formą.

Norint sėkmingai taikyti neuroninį tinklą modelis apribojamas konkrečiam sprendimui, pavyzdžiui, sukčiavimo prognozavimui. Tai leidžia neuroninį tinklą tinkamai pritaikyti vienai konkrečiai problemai spręsti. Jei pavyksta patvirtinti modelio teisingumą, modelis gali būti naudojamas daug kartų, nesigilinant, kaip jis veikia.

Neuroniniai tinklai sudaromi iš dviejų pagrindinių struktūrų:

1. Mazgas, kuris lyg ir atitiktų neuroną žmogaus smegenyse;
2. Jungčių, kurios atitiktų sąsajas tarp neuronų žmogaus smegenyse.

Paveiksle 2.6 pavaizduotas paprastas neuroninis tinklas. Apskritimai žymi mazgus, o jungiančios linijos – jungtis. Neuroninis tinklo prognozavimo veikimas paremtas skaičiavimais mazge priklausančiais nuo įėjimo reikšmių.

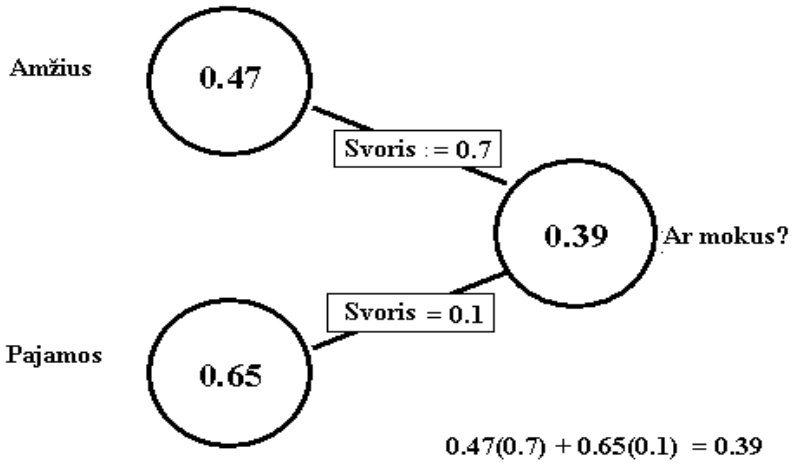


2.6 pav. Paskolų išdavimo neuroninis tinklas

Tam, kad neuroninis tinklas atliktų prognozavimą iš pradžių priimamos reikšmės į įėjimo mazgus. Šiuose mazguose įėjimo reikšmės padauginamos iš tam tikro svorio. Tuomet įėjimo mazguose gautos reikšmės siunčiamos išėjimo mazgui, kuriame suformuojamas

prognozavimo rezultatas.

Paveiksle 2.7 pateiktame neuroninio tinklo pavyzdyje prognozuojama reikšmė 0 reiškia, kad klientui galima suteikti paskolą, o 1 reiškia, kad paskolos suteikimas yra rizikingas. Kadangi prognozuota reikšmė artimesnė 0, įrašas priskiriamas nerizikingam.



2.7 pav. Normalizuotos įėjimo reikšmės padauginimas iš ryšių svorių paskolos davimo reikšmei apskaičiuoti

Neuroninis tinklas konstruojamas pateikiant daugybę pavyzdžių kuomet įėjimo signalai atitinka tam tikrą prognozuojamą reikšmę. Lyginant teisingą rezultatą, paimtą iš apmokymo įrašų ir keičiant ryšių svorius, galima lėtai keisti neuroninio tinklo elgseną. Kaip pavyzdys galėtų būti mokytojo klausimas moksleiviui ir pataisymas, jei moksleivis suklydo. Kuo didesnė buvo klaida, tuo didesnė korekcija atliekama.

Įvesties mazgų ir išvesties mazgų paskirtį paprastai yra gana lengva suprasti. Tačiau dar naudojamos ir kitos viršūnės, kurių paskirtį sunku nusakyti. Dėl to kyla keletas problemų:

- Sunku pasitikėti prognozavimo rezultatais, jeigu mazgų paskirtis nėra pakankamai gerai suprantama;
- Prognozavimas atliekamas išėjimo mazge. Jei skiriasi prognozavimo ir apmokymo reikšmės, reikia keisti svorius. Kaip perduoti svorių pakeitimus per kitus viršūnių sluoksnius?

Kitaip nei sprendimų medžiai ar artimiausių kaimynų metodai, kurie gali pasiekti aukštą prognozavimo tikslumą apmokymo duomenų bazėje, neuroniniai tinklai gali būti ilgai mokomi ir vis tiek nesugebės pasiekti 100% tikslumo apmokymo duomenims [5,15].

2.2.4 SQL SERVER 2008 analizės servisų duomenų gavybos algoritmai ir jų palyginimas

Tik keletas duomenų gavybos algoritmų šiuo metu turi standartus. Kiekvienas iš duomenų gavybos programinės įrangų tiekėjų algoritmuose įgyvendina jų pačių sukurtus patobulinimus. Pavyzdžiui, Microsoft sukūrė keletą savų patobulinimų algoritmams: medžių panaudojimas regresijai ar lizdiniai atvejai. Todėl įprasto duomenų gavybos algoritmo pavadinimo priekyje turime prierašą „Microsoft“.[13]

Duomenų gavybos algoritmais galima spręsti įvairius uždavinius. Svarbu žinoti ar sprendžiamam uždaviniui tinkamai pasirinktas algoritmas. Pagal uždavinio sprendimo būdą galima nustatyti, ar algoritmas gali būti pritaikytas uždavinio sprendimui. Algoritmai gali būti grupuojami pagal sprendimo būdus į tokias grupes:

- Klasifikavimas;
- Regresija;
- Asociacija;
- Ir kiti.

Palyginsime algoritmus pagal uždavinių sprendimo būdus (2.1 lentelė).

2.1 lentelė

Duomenų gavybos algoritmų palyginimas

Algoritmo realizacija / **Naudojimo paskirtis**	**Microsoft Sprendimų medžiai**	**K- vidurkių grupavimas**	**Neuroniniai tinklai**
Klasifikacija	**+**	**+**	**+**
Regresija	**+**	**+**	**+**
Asociacija	**+**	**-**	**-**
Veikimo principas	Sukuria po medį kiekvienam prognozuojamam atributui	Sudaromos panašių įrašų grupės	Mazgų įvertinimai perduodami sekantiems mazgams
Trūkumai	Blogai dirbai, kai duomenys reti, nepakankamas duomenų kiekis	Gerai dirba, kai grupės atsiskiria. Reikia nurodyti grupių skaičių	Sunku interpretuoti rezultatus
Pliusai	Pakankamai greitas, lengvai suprantamas	Greitas ir paprastas	Puikiai tinka komplikuotų sąryšių tarp atributų radimui

Norint palyginti skirtingų algoritmų veikimą, būtina juos analizuoti ir testuoti vienodomis sąlygomis. Šios duomenų gavybos algoritmų analizės metu paaiškėjo, kad visi algoritmai tinka klasifikacijos uždaviniams spręsti. Todėl kuriant bendrą modelį duomenų gavybos algoritmo pasirinkimo problemai spręsti buvo pasirinktas klasifikavimo uždavinys.

Klasifikavimo uždavinys paprastai apibrėžiamas kaip duomenų atvejų grupės narystės prognozavimas. Klasifikavimo uždavinio pavyzdys – oro spėjimas tam tikrą dieną, kuris gali būti „saulėtas“, „lietingas“, „apsiniaukęs“[6].

Regresija yra žinomiausia duomenų gavybos metodika. Regresijos uždavinyje imamas skaitinių reikšmių duomenų rinkinys ir sudaroma matematinė formulė, kuri apibrėžia duomenis. Pagrindinis apribojimas šio tipo uždaviniams, kad turi būti naudojami tolygūs atributai, pavyzdžiui, svoris, greitis ar amžius. Jei duomenyse naudojami diskretūs atributai tokie kaip spalva vardas ar kiti geriau rinktis kita metodika.[6]

2.3 Prognozavimo modelių tikslumo įvertinimas klasifikavimo uždaviniui

2.3.1 Duomenų gavybos algoritmų įvertinimo metodikos

Duomenų gavybos modeliavimo etape pasirinkus duomenų gavybos algoritmą ir iš istorinių duomenų sukūrus modelį gana svarbu įvertinti, kaip modelis dirba su duomenimis. Galima būtų pamėginti panaudoti netgi kelis duomenų gavybos modelius tai pačiai prognozavimo problemai spręsti ir pagal esamą ateities situaciją bei pagal prognozavimo efektyvumo, tikslumo įvertį pasirinkti tinkamesnį modelį. Yra keletas prognozavimo modelio našumo įvertinimo strategijų, kurias reikėtų analizuoti. Galima išskirti dvi įvertinimo metodų grupės: vaizdinės priemonės ir skaitiniai įvertinimo metodai.

Norint tinkamai sumodeliuoti duomenų gavybos algoritmų įvertinimo įrankį, reikia detaliau paanalizuoti skaitines metodų įvertinimo metodikas. Kadangi pasirinktu atveju spręsime klasifikacijos uždavinį, reikia išnagrinėti, kaip įvertinti skirtingus duomenų gavybos metodus klasifikacijos uždaviniui ir kokios yra modelių apmokymo strategijos.

2.3.2 Standartinės paklaidos įvertinimas

Atliekant klasifikavimo algoritmo įvertinimą apskaičiuotas įvertinimas įgyja paklaidą kadangi negali būti panaudojami visi galimi uždavinio duomenų atvejai. Naudojami statistiniai metodai apskaičiuoti intervalą, kuriame įvertinimas yra teisingas. Prognozavimo algoritmo įvertinimo ir paklaidos skaičiavimas priklauso nuo šių testavimo strategijų:

- duomenų padalinimas į apmokymo ir testavimo rinkinius;
- k-kartų kryžminis patvirtinimas;
- n-kartų kryžminis patvirtinimas (vieną praleidžiant).

Jeigu klasifikuosime skirtingus testinius duomenų rinkinius, tai gausime skirtingus prognozavimo įvertinimus p. Su įvertinimu p susijusi standartinė paklaida. Jei p apskaičiuojama naudojant tesintį duomenų rinkinį iš N atvejų tuomet standartinė paklaida yra $S = \sqrt{p(1-p)/N}$. Standartinė paklaida su tam tikra pasirinkta tikimybe leidžia nusakyti, kad teisingas klasifikatoriaus įvertinimas yra virš ar žemiau pamatuotos įvertinimo reikšmės p. Standartinės paklaidos skaičius žymimas Z_{CL}. Kuo tikslesni norime būti, tuo didesnis Z_{CL}. Pasirinkta tikslumo tikimybė vadinama pasitikėjimo lygiu ir žymima CL. Tuomet klasifikatoriaus įvertinimas yra intervale $p \pm Z_{CL} \cdot S$ [8].

Ryšys tarp dažnai naudojamų reikšmių CL ir Z_{CL} pateiktas lentelėje 2.2.

2.2 lentelė

CL ir Z_{CL} sąryšis

Pasitikėjimo lygis (CL)	0.9	0.95	0.99
Z_{CL}	1.64	1.96	2.58

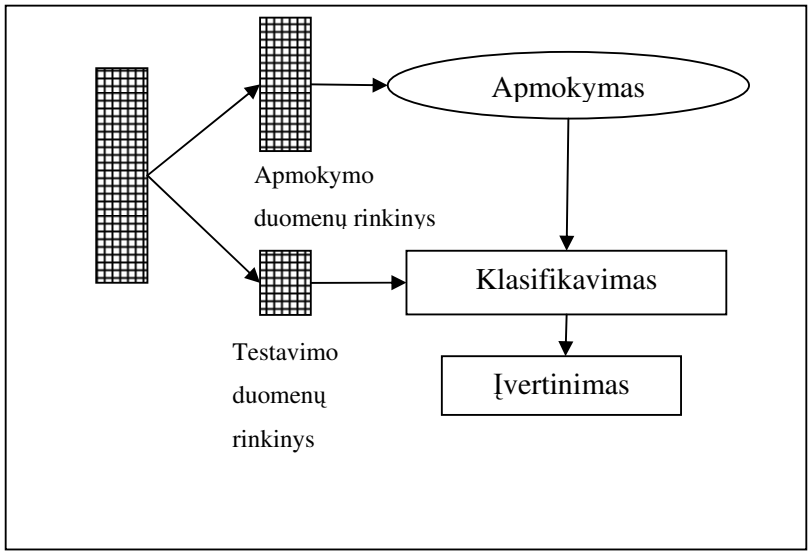
Pavyzdžiui, jei klasifikavimas 80 atvejų iš 100 prognozuotas teisingai, tuomet prognozavimo įvertinimas testiniam rinkiniui yra 80/100=0.8. Standartinė paklaida $\sqrt{0.8 \times 0.2/100} = 0.04$. Galime teigti, kad teisingas klasifikatoriaus įvertinimas su tikimybe 0.95 yra intervale $0.8 \pm 1.96 \times 0.04$.

K - kryžminio testavimo strategijos atveju klasifikatorius naudojamas klasifikuoti k testinių duomenų rinkinių. Jei visi rinkiniai yra vienodo dydžio N, tuomet bendras prognozavimo įvertinimas p skaičiuojamas panaudojant vidurkį visiems k testiniams rinkiniams. Kadangi bendras galimų atvejų skaičius testiniuose rinkiniuose yra kN, tuomet standartinė paklaida $S = \sqrt{p(1-p)/kN}$.[8]

2.3.3 Apmokymo ir testavimo duomenų atskyrimo strategija

Apmokymo ir testavimo strategijoje visi duomenys padalinami į dvi grupes, pavadintas „Apmokymo rinkiniu“ ir „Testavimo rinkiniu“ (2.8 pav.). Pirmiausia apmokymo rinkinys naudojamas konstruojant klasifikatorių (pavyzdžiui sprendimų medį ar neuroninį tinklą). Tuomet klasifikatorius yra naudojamas nuspėti, kaip suklasifikuoti testavimo duomenų

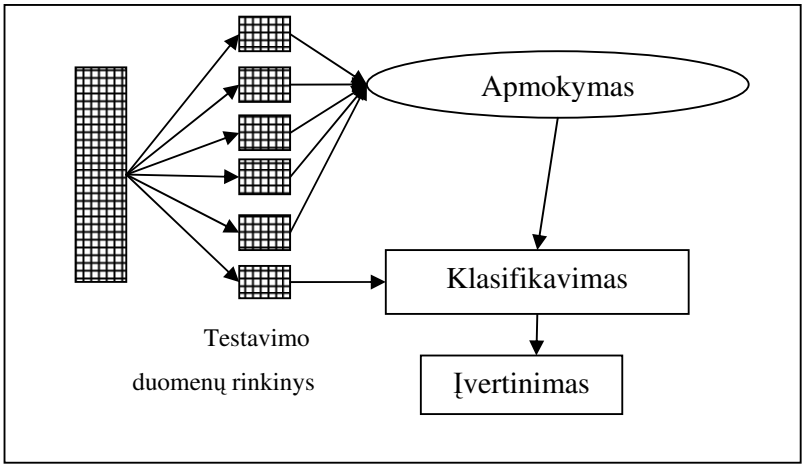
rinkinį. Skaičiuojant prognozavimo teisingumą testavimo rinkinys turi N atvejų iš kurių C atvejų teisingai suklasifikuoti, tuomet prognozavimo teisingumas klasifikatoriui apskaičiuojamas p=C/N. Taip įvertinamas klasifikatorius bet kuriam „Testavimo rinkiniui“.[8]



2.8 pav. Apmokymo ir testavimo duomenų atskyrimo strategija

2.3.4 K ir N kartų kryžminis patvirtinimas

Šios įvertinimo strategijos atveju turint N klasifikuojamų atvejų jie padalinami į K lygių dalių. K paprastai yra mažas skaičius, pavyzdžiui, 5 ar 10. Apmokymui naudojama serija K duomenų rinkinių. Kiekvienas iš K duomenų rinkinių paeiliui naudojamas kaip testavimo rinkinys. Likę K-1 rinkiniai naudojami kaip apmokymo duomenų rinkiniai (2.9 pav.).



2.9 pav. K - kartų kryžminis patvirtinimas

Kiekvienam testavimo rinkiniui teisingai suklasifikuotų atvejų skaičius dalinamas iš visų atvejų skaičiaus N tam, kad būtų galima apskaičiuoti teisingumo įvertinimą p su standartine paklaida $S = \sqrt{p(1-p)/N}$.

N-kartų kryžminio įvertinimo strategijoje kaip ir k-kartų kryžminėje strategijoje visas duomenų rinkinys padalinamas į tiek dalių, kiek yra klasifikavimo atveju N. N klasifikatorių sukuriama iš N-1 apmokymo duomenų atvejų. Kiekvienas jų naudojamas suklasifikuoti vienintelį likusį klasifikuojamą duomenų atvejį. Prognozavimo teisingumas kiekvienam testavimo rinkimui p yra bendras teisingai suklasifikuotų atvejų skaičius padalintas iš N su standartine paklaida $S = \sqrt{p(1-p)/N}$.

N-kartų kryžminė strategijai reikia atlikti daug skaičiavimų, todėl esant dideliems duomenų kiekiams sunkiai pritaikoma. Praktiškai metodas dažniausiai naudojamas ten, kur reikia kuo daugiau duomenų panaudoti klasifikatoriui apmokyti.[8]

2.3.5 Sumaišymo matrica (Confusion matrix)

Sumaišymo matrica parodo teisingai ir neteisingai suklasifikuotų duomenų atvejų skaičių. Dažnai pasitaiko, kad modelis gali būti labai geras vienų reikšmių klasifikatorius, o kitas gali labai prastai klasifikuoti. Sumaišymo matricos paskirtis yra identifikuoti, kokios rūšies klaidos būdingos prognozavimo modeliui. Vienos rūšies klaidų gali būti svarbiau išvengti nei kitos. Sumaišymo matrica suteikia galimybę pamatyti ir įvertinti klasifikatoriaus prognozavimo klaidas[7].

Sumaišymo matrica susideda iš keturių celių, kurios gali būti pažymėtos TP, FP, FN ir TN (2.3 lentelė).

2.3 lentelė

Sumaišymo matrica

		Prognozuojamos klasės		**Iš viso atvejų**
		+	**-**	
Tikrosios klasės	**+**	TP	FN	P
	-	FP	TN	N

TP - pozityvių atvejų, kurie suklasifikuojami kaip pozityvūs, skaičius;

FP - negatyvių atvejų, kurie suklasifikuojami kaip pozityvūs, skaičius;

FN – pozityvių atvejų, kurie suklasifikuojami kaip negatyvūs, skaičius;

TN – negatyvių atvejų, kurie suklasifikuojami kaip negatyvūs, skaičius;

N - bendras negatyvios klasės atvejų skaičius;

P – bendras pozityvios klasės atvejų skaičius.

FP ir FN yra klasifikuojančio modelio daromos klaidos. Priklausomai nuo uždavinio tipo, kiekviena iš šių klaidų gali turėti skirtingą svarbumą.

Priklausomai nuo TP, FP, FN, TN naudojimo išskiriami tokie klasifikuojančio modelio įvertinimai:

Pavadinimas	Formulė	Aprašymas
TP dažnis, jautrumas, pataikymų dažnis	TP/P	Santykis pozityvių atvejų, kurie teisingai klasifikuojami
FP dažnis arba netikro pavojaus dažnis	FP/N	Negatyvių atvejų, kurie klaidingai klasifikuojami kaip pozityvūs, santykis su negatyvios klasės atvejų skaičium
FN dažnis	FN/P	Santykis pozityvių atvejų, kurie klaidingai klasifikuojami kaip negatyvūs
TN dažnis	TN/N	Santykis negatyvių atvejų, kurie teisingai klasifikuojami kaip negatyvūs
Tikslumas (Precision)	TP/(TP+FP)	Santykis pozityvių prognozuotų reikšmių klasifikuotų kaip pozityvių, kurios iš tiesų yra pozityvios
F1 Score	(2 × TP)/(TP+FP)× TP/P)/ (TP/(TP+FP)+TP/P)	Įvertinimas, kuris apima tikslumą ir pataikymų dažnį
Teisingumas arba prognozavimo teisingumas	(TP + TN)/(P + N)	Santykis atvejų, kurie teisingai klasifikuoti
Klaidų dažnis	(FP + FN)/(P + N)	Santykis atvejų, kurie klaidingai klasifikuoti

Priklausomai nuo sprendžiamo klasifikavimo uždavinio tipo vieni įvertinimai gali būti svarbesni už kitus. Blogų įmonių investavimo uždavinyje FP dažnis (blogų įmonių skaičius klasifikuojamų kaip gerų) turėtų būti kuo mažesnis, idealiu atveju lygus nuliui. Kadangi yra daugybė įmonių į kurias galima investuoti, mažiau svarbu jei geros įmonės klasifikuojamos kaip blogos.[7]

Nustatant smegenų auglį paciento galvos peršvietimo uždavinyje daktaras siunčia pacientą pasidaryti nuotrauką. Pateikus klasifikavimo modeliui pacientui atliktų tyrimų duomenis prognozuojama ar siųsti pacientą darytis galvos nuotrauką. Šiame klasifikavimo uždavinyje leistinas aukštas FP dažnis (pacientai peršviesti bereikalingai). Tačiau FN dažnis (pacientų, turinčių smegenų auglį, neperšvietimas) turėtų būti kuo mažesnis.

2.4 Duomenų gavybos technologijų analizė

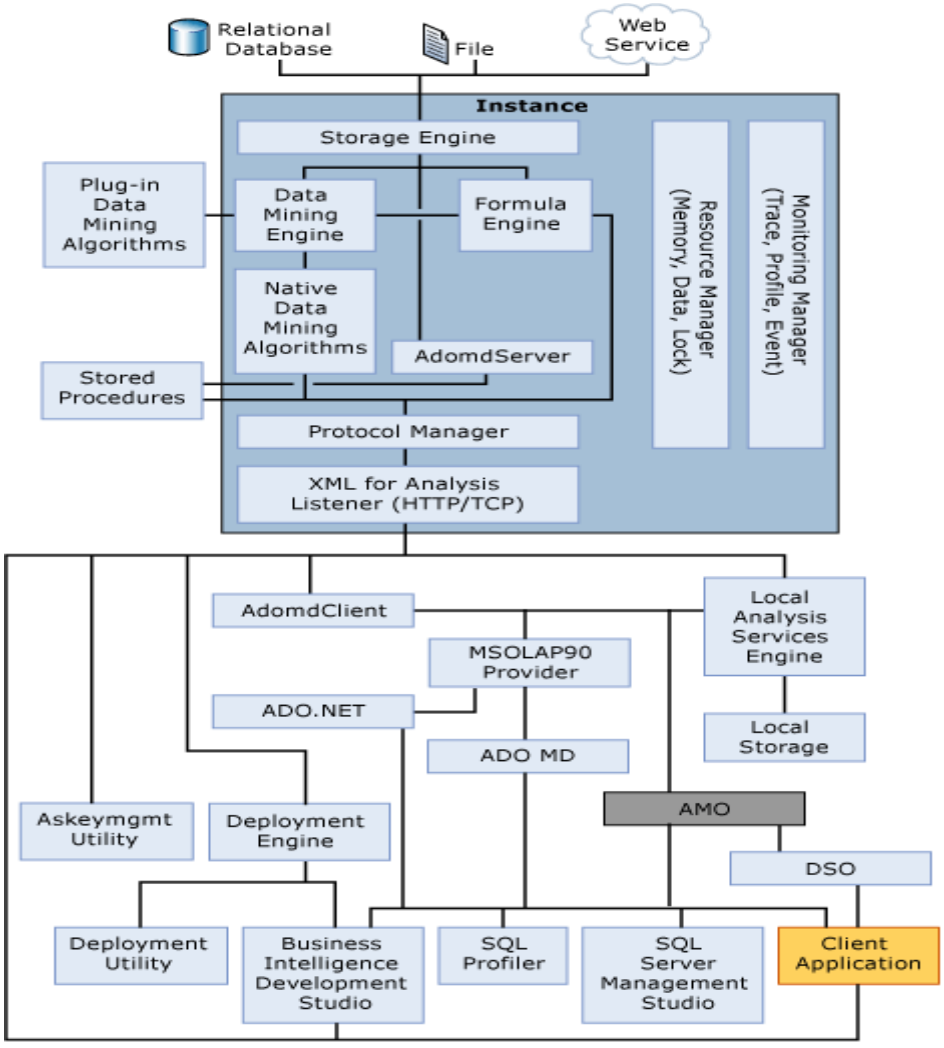
2.4.1 Microsoft SQL Server ir Oracle duomenų gavybos technologijos

Oracle reliacinėje duomenų bazėje realizuota daug įvairių duomenų gavybos algoritmų. Viskas realizuota tiesiogiai Oracle duomenų bazės branduolyje (angl. kernel) ir lokaliai operuojama su duomenimis laikomais reliacinės duomenų bazės lentelėse. Lokaliai naudojamos funkcijos duomenų gavybos modelių sukūrimui, testavimui ir manipuliavimui. Modeliai sukuriami ir saugomi kaip duomenų bazės objektai. Jų valdymas panašus į duomenų bazės lentelių, views, indeksų valdymą [9].

Microsoft analizės servisų pagalba Microsoft SQL serveryje atliekama duomenų analizė. Microsoft analizės servisų serverio ir klientų komponentai naudojami duomenų gavybos funkcionalumo realizacijai veiklos intelekto sistemose. Naudodami viešą analizės servisų XML standartą (XMLA), klientai komunikuoja su analizės servisais SOAP paremtu protokolu. Web serviso pagalba išsiunčiamos komandos ir gaunami atsakymai. Naudojant DMX kalbą vykdomos servisų užklausos. Taip pat gali būti naudojama skriptų kalba (ASSL) darbui su analizės servisų duomenų bazės objektais [10].

Analizės servisai veikia nepriklausomai, komunikavimas su servisu atliekamas per HTTP ar TCP protokolus. AMO yra sluoksnis tarp klientų programų ir analizės servisų. Šis sluoksnis suteikia priėjimą prie analizės servisų administravimo objektų. AMO yra klasių biblioteka, kuri komandas gautas iš kliento programų paverčia XMLA formato pranešimais analizės servisams [10].

Microsoft analizės serviso pagalba gali būti atliekama duomenų gavyba panaudojant duomenis reliacinių duomenų bazių lentelėse ar naudojant bet kurį kitą duomenų šaltinį, kuris apibrėžiamas analizės servisuose. Gali būti panaudojami ankstesnių SQL serverio versijų duomenų bazių, failų ir sistemų pasiekiamų per Web servisus duomenys [10].



2.10 pav. „Microsoft Analysis services“ architektūrinė schema

Norint Microsoft SQL serverio duomenų bazėje panaudoti duomenų gavybos įrankius reikia papildomai panaudoti analizės servisus. Microsoft analizės servisų atskyrimas nuo SQL serverio reliacinės duomenų bazės leidžia neprisirišti vien tik prie reliacinių duomenų bazių. Tačiau tokia Microsoft technologijų duomenų gavybos architektūra daro duomenų gavybos panaudojimą sudėtingesnį informacinėse sistemose, naudojančiose reliacines duomenų bazes. Negalima tiesiogiai taip kaip Oracle duomenų bazėse manipuliuoti duomenų gavybos objektais. MS SQL serverio reliacinėje duomenų bazėje negalima tiesiogiai vykdyti DMX prognozavimo užklausų ir valdyti modelių sukūrimą ir apmokymą.

2.4.2 DMX kalbos darbui su prognozavimo modeliais analizė

Microsoft Analizės servisuose DMX kalba leidžia sukurti naujus ir dirbti su esančiais duomenų gavybos modeliais. Priklausomai nuo klasifikavimo uždavinio tipo naudojant DMX kalba galima sukurti naujas duomenų gavybos modelių struktūras ir pačius modelius, apmokyti šiuos modelius, atlikti prognozavimą arba klasifikavimą. DMX susideda iš duomenų apibrėžties kalbos DDL komandų, duomenų manipuliavimo komandų DML ir funkcijų ir operatorių [11].

Microsoft SQL Serveryje analizės servisuose duomenų gavyba atliekama naudojant šias pagrindines operacijas su duomenų gavybos modeliais:

- Duomenų gavybos struktūros ir modelio sukūrimas;
- Duomenų gavybos modelių apmokymas;
- Duomenų gavybos modelių ir struktūrų ištrynimas;
- Duomenų gavybos modelių kopijavimas;
- Duomenų gavybos modelių peržiūra;
- Apmokytų duomenų gavybos modelių panaudojimas prognozavimui.

Norint atlikti prognozavimą arba klasifikavimą pirmiausia DDL komandomis sukuriama modelio struktūra, tuomet atliekamas modelio sukūrimas, vėliau apmokymas ir modelio panaudojimas prognozavimui. 2.4 lentelėje pateiktos pagrindinės DMX kalbos DDL komandos darbui su duomenų gavybos modeliais.[11]

2.4 lentelė

DDL komandos

Komanda	Aprašymas
CREATE MINING STRUCTURE (DMX)	Sukuria naują duomenų gavybos struktūrą analizės servisų duomenų bazėje
ALTER MINING STRUCTURE (DMX)	Prideda duomenų gavybos modelį į egzistuojančią duomenų gavybos struktūrą
CREATE MINING MODEL (DMX)	Sukuria naują duomenų gavybos struktūrą ir modelį analizės servisų duomenų bazėje
DROP MINING MODEL (DMX)	Ištrina duomenų gavybos modelį iš analizės servisų duomenų bazės
DROP MINING STRUCTURE (DMX)	Ištrina duomenų gavybos struktūra iš analizės servisų duomenų bazės
SELECT INTO (DMX)	Sukuria egzistuojančio duomenų gavybos modelio kopiją

Sukurti duomenų gavybos modeliai turi būti apmokomi prieš atliekant prognozavimą. Darbui su modelių duomenimis naudojamos DML komandos DMX kalboje. 2.5 lentelėje

pateiktos pagrindinės manipuliavimo modelių duomenimis komandos [11].

2.5 lentelė

DML komandos

Komanda	Aprašymas
DELETE (DMX)	Ištrina apmokymo duomenis iš duomenų gavybos modelio
INSERT INTO (DMX)	Apmoko duomenų gavybos modelį
SELECT (DMX)	Apmokyto duomenų gavybos modelio duomenų peržiūra

Prognozavimo užklausos DMX kalba vykdomos Microsoft analizės servisuose. Taip naujiems duomenų rinkiniams galima nustatyti nežinomų atributų reikšmes, kurios priklauso nuo pasirinkto apmokyto duomenų gavybos modelio. 2.6 lentelėje pateikti galimi prognozavimo užklausų variantai.

2.6 lentelė

Prognozavimo užklausų tipai

Prognozavimo tipas	**Aprašymas**
Prediction join	Naudojamas prognozavimo užklausai kai reikalingas ON operatorius, kuris nusako sujungimo sąlygas tarp duomenų gavybos modelio stulpelių ir įėjimo duomenų atributų
Natural prediction join	Naudojama kai sutampa modelio struktūroje esančių stulpelių pavadinimai su įėjimo duomenų stulpelių pavadinimais. Šio tipo užklausoje nereikalinga ON sintaksė, nes tarp duomenų gavybos modelio stulpelių ir įėjimo duomenų stulpelių sujungimo sąlygos sudaromos automatiškai pagal sutampančius pavadinimus.
Empty prediction join	Šio tipo prognozavime nenaudojami įėjimo duomenys. Gražina prognozes tik duomenų gavybos modelyje esantiems apmokymo duomenims.

DMX užklausos struktūra

Prognozavimo užklausa DMX kalboje atliekama panaudojant šių elementų kombinacijas:

- **SELECT [FLATTENED]**
- **TOP**
- **FROM <*model*> PREDICTION JOIN**
- **ON**
- **WHERE**
- **ORDER BY**

Select operatorius prognozavimo užklausoje apibrėžia stulpelius ir išraiškas, kurios bus matomos rezultatų rinkinyje. Operatoriais *Predict* ir *PredictOnly* nurodomi prognozuojami stulpeliai duomenų gavybos modelyje. Taip pat nurodomi visi matomi stulpeliai ar funkcijos vykdomos stulpelio duomenims.

Sintaksės **FROM <*model*> PREDICTION JOIN** elementas apibrėžia duomenų šaltinį kuriam atliekama prognozė.

ON elementas susieja duomenų šaltinio atributus su duomenų gavybos modelio atributais. Šis elementas nenaudojamas kai užklausos tipai yra "empty prediction join" arba "natural prediction join".

Naudojant **WHERE** sąlyga atliekamas prognozavimo užklausos rezultatų filtravimas. Taip pat galima naudoti **TOP** ir **ORDER BY** operatorius kaip ir SQL užklausose.

2.5 Analizės išvados

1. CRISP-DM – duomenų gavybos proceso modelis, kuris aprašo dažniausiai naudojamus metodus, kuriuos patyrę ekspertai naudoja duomenų gavybos problemoms spręsti.
2. Atlikus „K – vidurkių grupavimo", „Neuroninių tinklų", „Sprendimų medžių" algoritmų analizę paaiškėjo, kad šie algoritmai bendrai gali spręsti klasifikavimo uždavinį, kuris apibrėžiamas kaip duomenų narystės grupėje prognozavimas.
3. Atlikus metodų, leidžiančių įvertinti algoritmus, analizę paaiškėjo, kad prognozavimo algoritmus galima įvertinti naudojant skirtingas įvertinimo strategijas ir metodus. Dažniausiai naudojamos dvi įvertinimo strategijos „Apmokymo ir testavimo atskyrimas" ir „K – kryžminis patvirtinimas". Klasifikavimo uždaviniams įvertinti tinkamiausi yra sumaišymo matrica paremti įvertinimo metodai.
4. Daugkartinis DMX kalbos šablonų panaudojimas skirtingiems klasifikavimo uždaviniams spręsti naudojant DMX šablonus gana sudėtingas. Nėra vieningo modelio, kaip galima būtų naudoti DMX šablonus prognozavimo modelių sukūrimui, įvertinimui ir geriausio klasifikavimo modelio parinkimui. Vieningo modelio sukūrimas ir daugkartinis naudojimas pagreitintų duomenų gavybos CRISP-DM procesų modeliavimo įvertinimo ir diegimo etapus.
5. Priklausomai nuo sprendžiamo klasifikavimo uždavinio tipo, turėtų būti naudojami skirtingi įvertinimo metodai. Atsižvelgiant į klasifikavimo uždavinį ir naudojant skirtingus įvertinimo metodus, reikia išrinkti geriausiai tinkantį prognozavimo modelį. Kadangi modelio įvertinimas priklauso taip pat ir nuo pasirinktos modelio testavimo strategijos, vieningas modelis, leidžiantis operuoti DMX šablonais duomenų gavyboje, taip pat turėtų leisti rinktis skirtingas strategijas modelių įvertinimui.
6. DMX kalbos naudojimas analizės servisuose remiasi DML ir DDL komandomis. Operuojant reliacinės duomenų bazės duomenimis šių komandų ar prognozavimo užklausų

Microsoft SQL serveryje tiesiogiai vykdyti negalima. Todėl duomenų gavybos panaudojimas klasifikavimo uždavinių sprendimui informacinėse sistemose su reliacinėmis duomenų bazėmis reikalauja sudėtingesnės architektūros. Papildomai reikia naudoti analizės servisus. Šiuo atveju įrankis, realizuotas SQL serverio reliacinėje duomenų bazėje, taikant vieningą modelį, palengvintų klasifikavimo uždaviniams naudojamų prognozavimo modelių DMX šablonais sukūrimą, įvertinimą bei išsirinkimą.

3. DUOMENŲ GAVYBOS PROGNOZAVIMO MODELIŲ VALDYMO IR ĮVERTINIMO SISTEMA

3.1 Sistemos paskirtis

Naudojant vieningą struktūrą, palengvinti klasifikavimo uždaviniams skirtų duomenų gavybos modelių, paremtų DMX šablonais, sukūrimą, apmokymą, ir tinkamiausio modelio išrinkimą.

3.2 Funkciniai ir nefunkciniai reikalavimai

Nefunkciniai reikalavimai

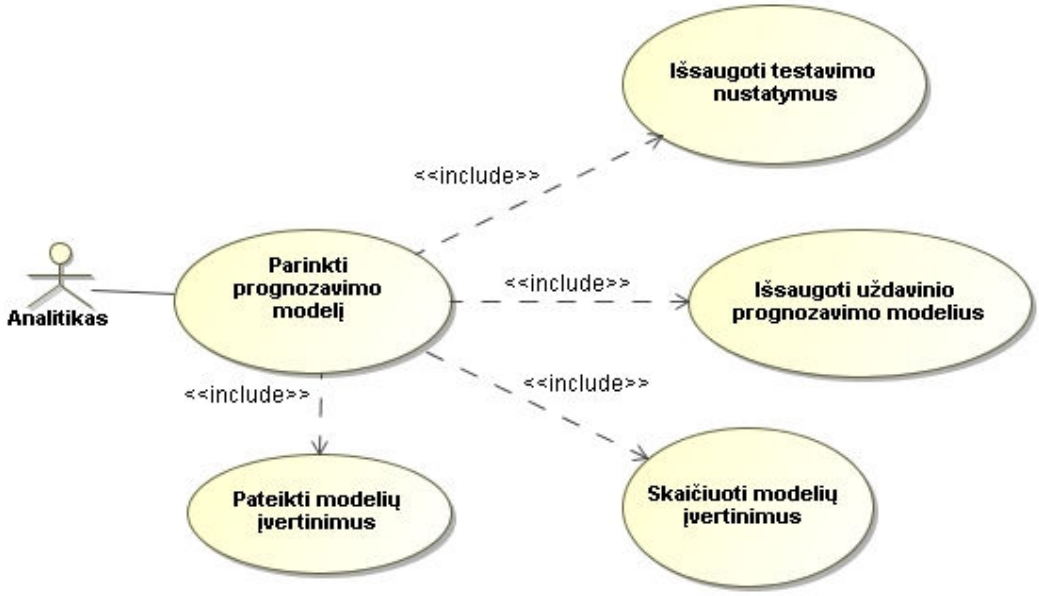
Duomenų gavybos metodo išrinkimas gali būti taikomas tik sprendžiant to paties tipo uždavinį (pavyzdžiui, klasifikavimą). Todėl svarbu, kad visi duomenų gavybos metodai tiktų konkretaus vieno uždavinio sprendimui. Su konkrečiu uždaviniu yra susiję sistemoje besikeičiantys duomenys. Šių duomenų dalis turi būti išrinkta apmokymui ir testavimui. Sistema naudojanti duomenų gavybos metodo išrinkimą turi saugoti naujus duomenis, iš kurių vėliau gali būti atliekamas apmokymas ir klasifikavimas. Iškart ar vėliau turėtų būti įvertinta ar klasifikavimas teisingai išspręstas išsaugant teisingo sprendimo rezultatą.

Funkciniai reikalavimai

Skirtingose sistemose klasifikavimo uždaviniams spręsti gali reikti naudoti įvairias skirtingų duomenų gavybos metodų kombinacijas, todėl modelis turi leisti rinktis, kuriuos duomenų gavybos metodus naudoti. Priklausomai nuo pasirinkto kriterijaus - sistemos duomenų kaitos greičio, laiko ar kito, turi būti vykdomas geriausio metodo įvertinimas, apmokymas ir išrinkimas sistemos naudojimui. Įrankis turi turėti galimybę keisti klasifikavimo metodo išrinkimo ir įvertinimo kriterijus, apmokymo ir testavimo duomenų kiekius ar santykius bei testavimo strategijas.

3.3 Panaudojimo atvejų diagramos ir jų specifikacija

Analitiko panaudojimo atvejų diagrama



3.1 pav. Analitiko panaudojimo atvejų diagrama

Panaudojimo atvejis “Parinkti prognozavimo modelį”

3.1 lentelė.

Panaudojimo atvejo “Parinkti prognozavimo modelį ” specifikacija

Pavadinimas	**Reikšmė**
Prieš sąlyga	
Po sąlyga	Išrinktas ir paruoštas prognozavimo modelis
Tikslas	Išrinkti prognozavimo modelį, nustatyti algoritmų modelių apmokymo duomenų kiekį klasifikavimo uždavinio realizacijai
Įgyvendinimo atvejai	Atliekama klasifikavimo uždavinio prognozavimo algoritmų analizė ir valdymas
Pastabos	Visi analizuojami prognozavimo modelių algoritmai turi tikti spręsti tą patį klasifikavimo uždavinį

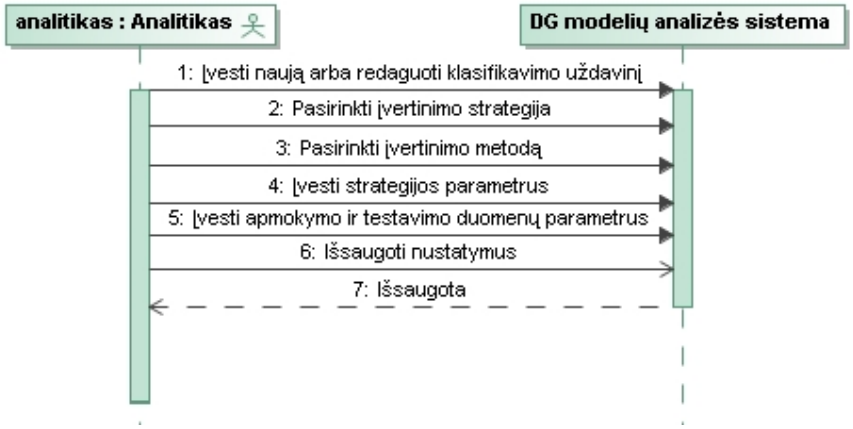
Šis panaudojimo atvejis apima kitus keturis panaudojimo atvejus: “Išsaugoti testavimo nustatymus”, “Išsaugoti uždavinio prognozavimo modelius”, “Skaičiuoti modelių įvertinimus” ir “Pateikti modelių įvertinimus”.

Panaudojimo atvejis “Išsaugoti testavimo nustatymus”.

3.2 lentelė

“Išsaugoti testavimo nustatymus ”

Pavadinimas	Reikšmė
Prieš sąlyga	Reikalingas naujo uždavinio įvertinimas arba uždavinio nustatymų keitimas
Po sąlyga	Išsaugoti nustatymai įvertinimo skaičiavimams
Tikslas	Nustatyti klasifikavimo uždavinio įvertinimo skaičiavimo strategija, įvertinimo metodą. Nustatyti apmokymo duomenų kiekių kitimą skaičiuojant įvertinimus ir testavimo duomenų kiekius
Įgyvendinimo atvejai	Turi būti išsaugota informacija apie klasifikavimo uždavinį, apmokymo ir testavimo duomenų kiekius. Nustatytas koks įvertinimo metodas turi būti panaudotas ir pasirinkta testavimo strategija



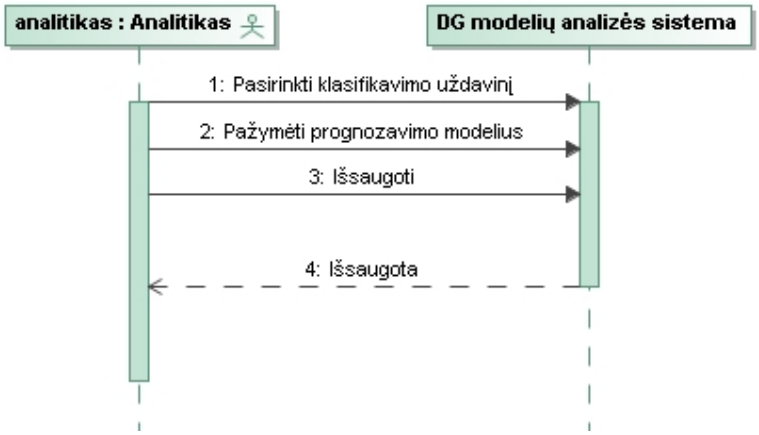
3.2 pav. Panaudojimo atvejo “ Išsaugoti testavimo nustatymus ” sekų diagrama

Panaudojimo atvejis “Išsaugoti uždavinio prognozavimo modelius”

3.3 lentelė

“ Išsaugoti uždavinio prognozavimo modelius”

Pavadinimas	Reikšmė
Prieš sąlyga	Egzistuoja sukurtas klasifikavimo uždavinys
Po sąlyga	Išsaugoti klasifikavimo uždaviniui analizuojami prognozavimo modeliai
Tikslas	Nustatyti klasifikavimo uždavinyje naudojamus prognozavimo modelius įvertinimui
Įgyvendinimo atvejai	Pasirenkami klasifikavimo uždaviniui įvertinamai prognozavimo modeliai



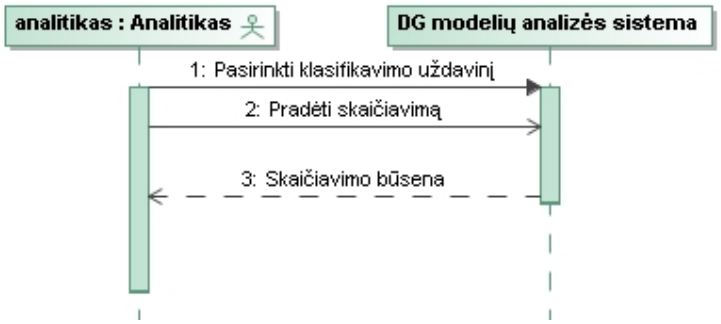
3.3 pav. Panaudojimo atvejo“ Išsaugoti uždavinio prognozavimo modelius ” sekų diagrama

Panaudojimo atvejis “Skaičiuoti modelių įvertinimus”

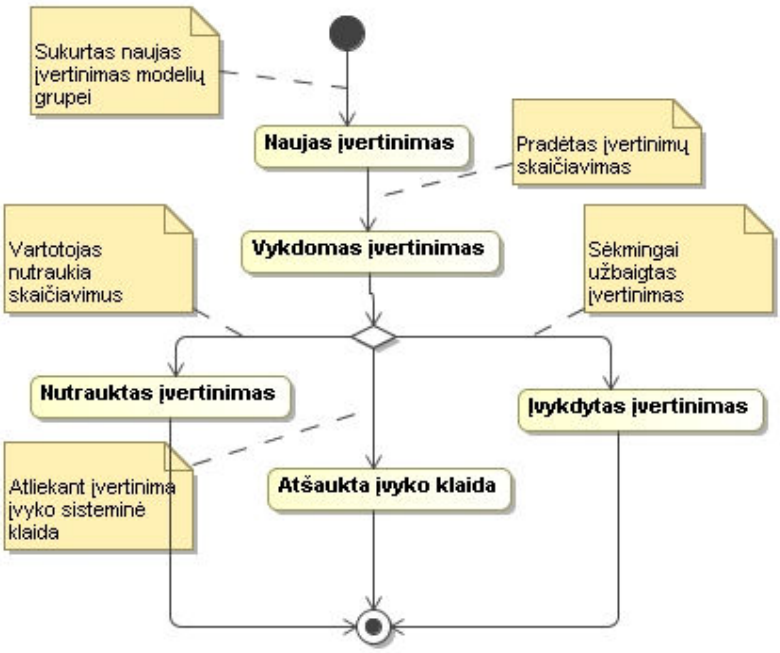
3.4 lentelė

“Skaičiuoti modelių įvertinimus”

Pavadinimas	**Reikšmė**
Prieš sąlyga	Pasirinkti prognozavimo modeliai klasifikavimo uždaviniui (pav. 3.1). Sukurti DMX modelių šablonai, apmokymo šablonai ir įvertinimo šablonai (pav. 3.7)
Po sąlyga	Prognozavimo modelių įvertinimų skaičiavimo būsena
Tikslas	Įvertinti prognozavimo modelius
Įgyvendinimo atvejai	Atliekamas klasifikavimo uždavinio prognozavimo modelių įvertinimų skaičiavimas
Pastabos	Įvertinimas iš naujo turėtų būti atliekamas pasikeitus apmokymo ar prognozavimo duomenims ar po įtraukimo naujų prognozavimo modelių



3.4 pav. Panaudojimo atvejo “Skaičiuoti modelių įvertinimus” sekų diagrama



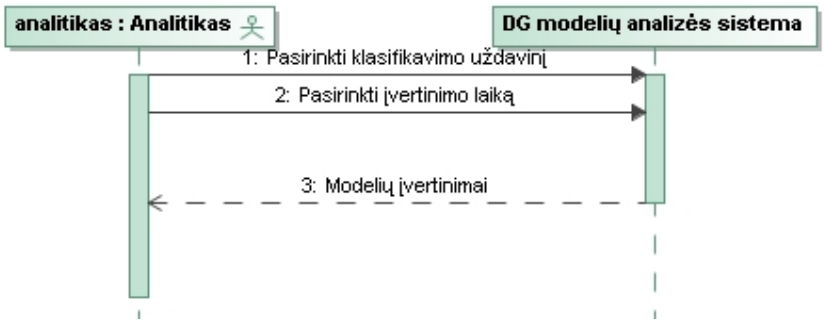
3.5 pav. Panaudojimo atvejo "Skaičiuoti modelių įvertinimus" būsenų diagrama

Panaudojimo atvejis "Pateikti modelių įvertinimus"

3.5 lentelė

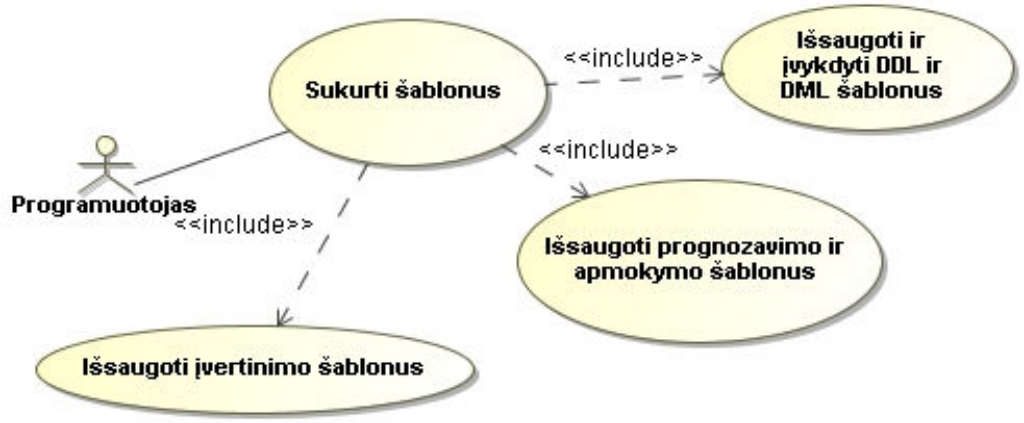
"Pateikti modelių įvertinimus"

Pavadinimas	**Reikšmė**
Prieš sąlyga	Apskaičiuoti klasifikavimo uždavinio prognozavimo modelių įvertinimai
Po sąlyga	Pateikti modelių įvertinimai pagal, kuriuos galima spręsti apie tinkamiausio modelio išrinkimą klasifikavimo uždaviniui
Tikslas	Pateikti prognozavimo modelių įvertinimus grafinei analizei.
Įgyvendinimo atvejai	Pagal pateiktus klasifikavimo uždavinio įvertinimus atliekama klasifikavimo uždavinio prognozavimo modelių analizė, nustatomas modelio apmokymo duomenų kiekis, išrenkamas geriausias modelis



3.6 pav. Panaudojimo atvejo "Pateikti modelių įvertinimus" sekų diagrama

Programuotojo panaudojimo atvejų diagrama



3.7 pav. Programuotojo panaudojimo atvejų diagrama

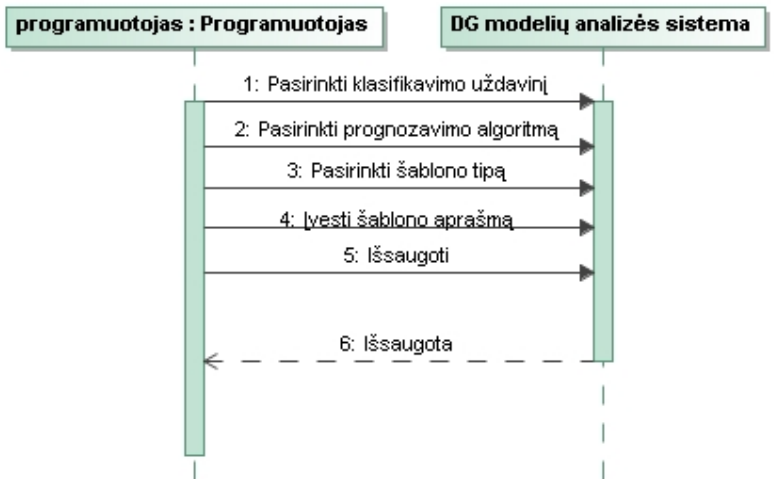
Panaudojimo atvejis „Sukurti šablonus“

3.6 lentelė

„Sukurti šablonus“

Pavadinimas	**Reikšmė**
Prieš sąlyga	Sukurtas naujas klasifikavimo uždavinys ir pasirinkti prognozavimo įvertinami algoritmai (panaudojimo atvejų diagrama 3.1 pav.)
Po sąlyga	Sukurtos modelių struktūros Microsoft Analysis serveryje ir išsaugoti šablonai modelių apmokymui ir prognozavimui
Tikslas	Paruošti klasifikavimo uždavinyje naudojamus modelius įvertinimui
Įgyvendinimo atvejai	Prieš atliekant klasifikavimo uždavinio prognozavimo modelių įvertinimo skaičiavimą paruošiami operacijų su modeliais šablonai ir patys modeliai.

Panaudojimo atvejis „Sukurti šablonus“ susideda iš dviejų panaudojimo atvejų: „Išsaugoti ir įvykdyti DDL ir DML šablonus“, „Išsaugoti prognozavimo ir apmokymo šablonus“ ir „Išsaugoti įvertinimo šablonus“



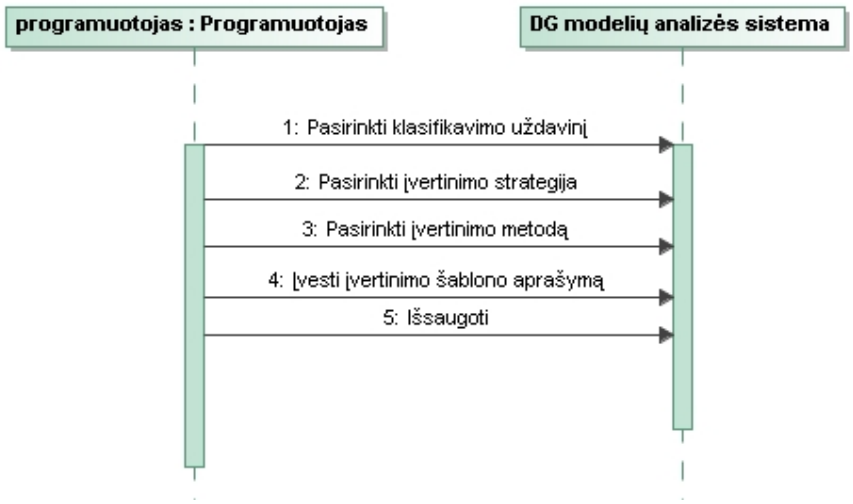
3.8 pav. Panaudojimo atvejų „Išsaugoti ir įvykdyti DDL ir DML šablonus“, „Išsaugoti prognozavimo ir apmokymo šablonus“ sekų diagrama

Panaudojimo atvejis „Išsaugoti įvertinimo šablonus“

3.7 lentelė

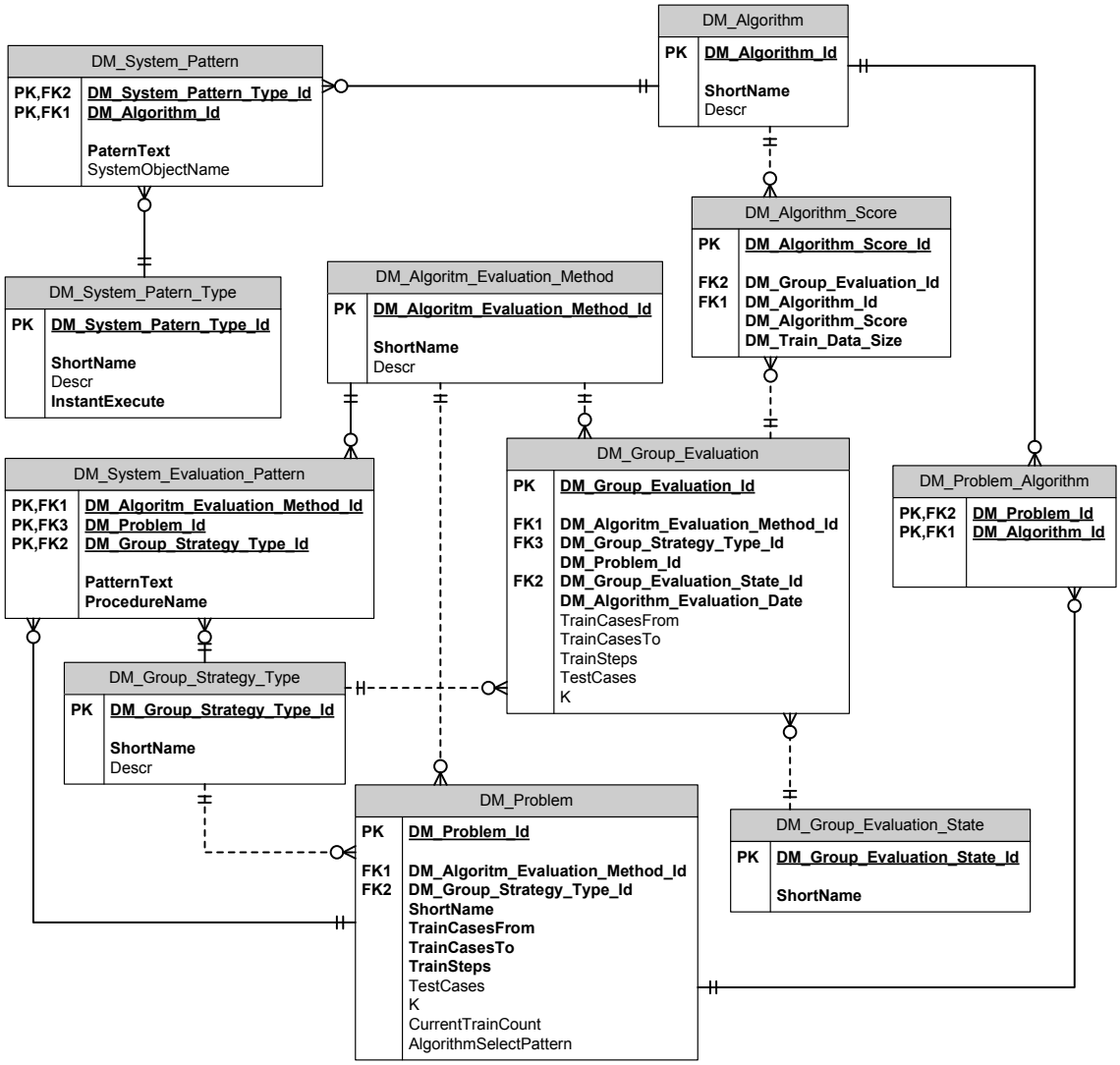
„Išsaugoti įvertinimo šablonus“

Pavadinimas	**Reikšmė**
Prieš sąlyga	Sukurti DMX šablonai prognozavimui, modelio apmokymui ir struktūros sukūrimui
Po sąlyga	Išsaugotas šablonas klasifikavimo uždavinio prognozavimo modelių įvertinimui
Tikslas	Paruošti klasifikavimo uždavinyje naudojamus modelius įvertinimui, pateikiant bendrą modelių įvertinimo šabloną
Įgyvendinimo atvejai	Prieš atliekant algoritmų modelių įvertinimą, pagal reikiama įvertinimo metodą ir strategija, sudaromas bendras prognozavimo algoritmų modelių įvertinimo šablonas



3.9 pav. Panaudojimo atvejo “ Išsaugoti įvertinimo šablonus” sekų diagrama

3.4 Klasifikavimo uždavinių, prognozavimo algoritmų, įvertinimo modelio duomenų bazės schema



3.10 pav. DB schema klasifikavimo uždavinių įvertinimui

Lentelė „DM_Algorithm“ aprašo visus galimus prognozavimo algoritmus

Lentelės atributai:

ShortName – trumpas prognozavimo algoritmo pavadinimas
Descr – algoritmo aprašymas

Lentelė „DM_Algorithm_Score“ aprašo algoritmų modelių įvertinimus

Lentelės atributai:

DM_Algorithm_Score – algoritmo įvertinimas
DM_Train_Data_Size – naudojamas apmokymo duomenų kiekis įvertinant

Lentelė „DM_Algoritm_Evaluation_Method“ aprašo galimus įvertinimo metodus

Lentelės atributai:

ShortName – trumpas įvertinimo pavadinimas
Descr – įvertinimo aprašymas

Lentelė „DM_Group_Evaluation“ aprašo klasifikavimo uždaviniui atliekamo įvertinimo nustatymus

Lentelės atributai:

DM_Algorithm_Evaluation_Date – modelių įvertinimo atlikimo data ir laikas
TrainCasesFrom – modelių apmokymo duomenų kiekis nuo
TrainCasesTo – modelių apmokymo duomenų kiekis iki
TrainSteps – modelių apmokymui naudojamų duomenų kiekio kitimo intervalų skaičius
TestCases – modelio testavimui naudojamų duomenų kiekis
K – kryžminės strategijos atveju apmokymo duomenų sudalinimų į intervalus skaičius

Lentelė „DM_Group_Evaluation_State“ aprašo galimas modelių įvertinimo skaičiavimo būsenas

Lentelės atributai:

ShortName – įvertinimo skaičiavimo būsenos aprašymas

Lentelė „DM_Group_Strategy_Type“ aprašo galimas strategijas modelių įvertinimo skaičiavimui

Lentelės atributai:

ShortName – trumpas įvertinimo strategijos pavadinimas
Descr – įvertinimo strategijos aprašymas

Lentelė „DM_Problem“ aprašo klasifikavimo uždavinį ir prognozavimo modelių įvertinimo nustatymus

Lentelės atributai:

ShortName – trumpas klasifikavimo uždavinio pavadinimas
TrainCasesFrom – modelių apmokymo duomenų kiekis nuo
TrainCasesTo – modelių apmokymo duomenų kiekis iki
TrainSteps - modelių apmokymui naudojamų duomenų kiekio kitimo intervalų skaičius
TestCases – modelio testavimui naudojamų duomenų kiekis
K – kryžminės strategijos atveju apmokymo duomenų sudalinimų į intervalus skaičius

Lentelė „DM_Problem_Algorithm“ aprašo, kurie prognozavimo modeliai naudojami skaičiuojant įvertinimus klasifikavimo uždaviniui

Lentelė „DM_System_Evaluation_Pattern“ aprašo prognozavimo modelių, įvertinimo skaičiavimui, naudojamą šabloną

Lentelės atributai:

PatternText – šablono modelių įvertinimui aprašymas
ProcedureName – sql server procedūros, kuri atlieka įvertinimą pavadinimas

Lentelė „DM_System_Patern_Type“ aprašo šablonų tipus, kuriais gali būti modelio ar jo struktūros sukūrimo, pašalinimo šablonai, modelio apmokymo šablonas, modelio prognozavimo šablonas, apmokymo duomenų šablonas ir testavimo duomenų šablonas

Lentelės atributai:

ShortName – trumpas šablono tipo pavadinimas
Descr – šablono tipo aprašymas
InstantExecute – žymi ar šabloną išsaugant iškart įvykdyti modelio, struktūros sukūrimą arba pašalinimą

Lentelė *„DM_System_Pattern“* aprašo prognozavimo modelių šablonus pagal jų tipus aprašytus lentelės *„DM_System_Patern_Type“*

Lentelės atributai:

PaternText – prognozavimo šablono aprašymas priklausomai nuo šablono tipo
SystemObjectName – prognozavimo modelio arba struktūros sisteminis pavadinimas, jei šablono tipas yra struktūros arba modelio sukūrimas

3.5 Įvertinimo modelyje naudojami šablonai ir jų parametrai

Prognozavimo algoritmų modeliai priklauso nuo konkretaus klasifikavimo uždavinio. Todėl šablonai modelių sukūrimui, apmokymui ir įvertinimui turi būti sudaromi atsižvelgiant į konkretų klasifikavimo uždavinį. Šiuo atveju įvertinimo modelis bus naudojamas turint ligonių klasifikavimo uždavinį. Šiam uždaviniui analitikas turėtų parinkti koks įvertinimo metodas, strategija turi būti naudojama modelių palyginimui ir nustatyti testavimo duomenų kiekius, jų kitimą bei apmokymo duomenų kiekius. Programuotojas atsižvelgdamas į tai kokia yra klasifikavimo uždavinio duomenų struktūra, kur laikomi ir kokie turi būti naudojami modelio apmokymo, testavimo duomenys sudaro šešių tipų šablonus:

- Struktūros duomenų gavybai sukūrimo arba redagavimo šablonas
- Duomenų gavybos modelio sukūrimo šabloną
- Duomenų gavybos modelio apmokymo šablonas
- Duomenų gavybos modelio prognozavimo šablonas
- Apmokymo duomenų šablonas
- Testavimo duomenų šablonas

Taip pat turi būti sukurtas šablonas modelių įvertinimui, kuris gali priklausyti tiek nuo

pasirinktos įvertinimo metodikos, tiek nuo įvertimo strategijos.

3.5.1 Struktūros duomenų gavybai sukūrimo arba redagavimo šablonas

Struktūros duomenų gavybai sukūrimas ir redagavimas atliekamas CREATE ir ALTER DDL komandomis. Parametras @MiningStruct apibrėžia duomenų gavybos struktūros pavadinimą. Jei duomenų gavybos modelyje naudojame „MICROSOFT_NEURAL_NETWORK“ algoritmą tuomet duomenų struktūra galėtų būti pavadinta „[HM_MICROSOFT_NEURAL_NETWORK_Structure]“. @MiningStruct parametras įgis reikšmę „[HM_MICROSOFT_NEURAL_NETWORK_Structure]“. Skliausteliuose už struktūros pavadinimo išvardijami duomenų atributai, jų tipai ir diskretumas(tolygumas)(žr. pavyzdį esantį žemiau).

Duomenų gavybos struktūros sukūrimo šablonas ligonių klasifikavimo uždaviniui pavyzdys:

```
CREATE MINING STRUCTURE @MiningStruct
(   Age LONG CONTINUOUS,
    Angina Boolean DISCRETE,
    Cholesteral LONG CONTINUOUS,
    Class LONG DISCRETE,
    Ecg TEXT DISCRETE,
    kId LONG KEY,
    Oldpeak DOUBLE CONTINUOUS,
    Pain TEXT DISCRETE,
    Rate LONG CONTINUOUS,
    SEX TEXT DISCRETE,
    Slope TEXT DISCRETE,
    Sugar Boolean DISCRETE,
    Thal TEXT DISCRETE,
    Trestbps LONG CONTINUOUS,
    Vessels LONG DISCRETE
)
```

Duomenų gavybos modelio sukūrimo metu, duomenų gavybos struktūra susiejama su duomenų gavybos modeliu. Modeliai privalo turėti rakto atributą, kuris apibrėžiamas duomenų gavybos struktūroje “KEY” operatoriumi einančiu po atributo tipo.

Norint pašalinti sukurtą duomenų gavybos modelį arba struktūra naudojama „DROP“ DMX DDL komanda:

```
DROP MINING STRUCTURE [HM_MICROSOFT_NEURAL_NETWORK_Structure]
```

„[HM_MICROSOFT_NEURAL_NETWORK_Structure]“ aprašo struktūros, kuri bus pašalinta pavadinimą.

3.5.2 Duomenų gavybos modelio sukūrimo šablonas

Duomenų gavybos modelio šablonas apibrėžia, kurie duomenų atributai ir koks algoritmas turi būti naudojami duomenų gavybos modelio sukūrimui. Šablone parametru @MiningModel nusakomas duomenų gavybos modelio pavadinimas. Parametras @MiningStruct, kaip ir struktūros sukūrimo šablone, nusako duomenų gavybos struktūros pavadinimą. Skliausteliuose, už duomenų gavybos modelio pavadinimo, nurodomi prognozavimui atlikti naudojami duomenų atributai. PREDICT arba PREDICT ONLY operatoriais pažymimi atributai, kurie bus prognozuojami pagal visus kitus modelio šablone aprašytus atributus. Už skliaustelių po operatoriaus USING turi būti nurodytas prognozavimo algoritmas. Šiuo atveju modelis sukuriamas naudojant „MICROSOFT_NEURAL_NETWORK“ algoritmą (toliau pateiktas pavyzdys).

Duomenų gavybos modelio sukūrimo šablonas ligonių klasifikavimo uždaviniui pavyzdys:

```
ALTER MINING STRUCTURE  @MiningStruct
ADD MINING MODEL @MiningModel
(
    Age ,
    Angina,
    Cholesteral,
    Class  PREDICT_ONLY,
    Ecg,
    Oldpeak ,
    Pain ,
    kId,
    Rate,
    SEX ,
    Slope ,
    Sugar ,
    Thal ,
    Trestbps,
    Vessels
)
USING MICROSOFT_NEURAL_NETWORK
```

Prognozavimui galima naudoti ir kitus klasifikavimui tinkančius algoritmus, pavyzdžiui:„MICROSOFT_CLUSTERING“,“MICROSOFT_DECISION_TREES“. Algoritmų aprašymai pateikiami šaltinyje [12].

Norint pašalinti sukurtą duomenų gavybos modelį, naudojama „DROP“ DMX DDL komanda:

```
DROP MINING MODEL [HM_MICROSOFT_NEURAL_NETWORK]
```

„[HM_MICROSOFT_NEURAL_NETWORK]“ žymi modelio, kuris bus pašalintas pavadinimą

3.5.3 Duomenų gavybos modelio apmokymo šablonas

Sukurtas duomenų gavybos modelis panaudojant modelio sukūrimo šabloną turi būti apmokytas prieš atliekant prognozavimą arba įvertinimą. Modelio apmokymas atliekamas DMX DML komanda „INSERT INTO“. Parametru @MiningModel apibrėžiamas apmokomo modelio pavadinimas. Skliausteliuose už apmokomo modelio pavadinimo nurodomi atributai , kurie naudojami apmokant modelį panaudojant duomenų šaltinį(žr pavyzdį esantį žemiau). Šiame šablono pavyzdyje naudojamas duomenų šaltinis yra SQL serveryje esanti reliacinės duomenų bazės lentelė. Duomenų perdavimas iš SQL serverio duomenų bazės į Analizės servisus atliekamas OPENQUERY sintakse. OPENQUERY pirmuoju parametru nurodomas duomenų šaltinis(SQL serverio duomenų bazė), antruoju reliacinės duomenų bazės SQL komanda duomenų išrinkimui. SQL komanda šiuo atveju nusakoma parametru @TrainDataPattern, kuris žymi apmokymo duomenų šabloną.

Modelio apmokymo šablono pavyzdys:

```
INSERT INTO @MiningModel (
    Age ,
    Angina ,
    Cholesteral ,
    Class ,
    Ecg ,
    kId ,
    Oldpeak ,
    Pain ,
    Rate ,
    SEX ,
    Slope ,
    Sugar ,
    Thal ,
   Trestbps ,
    Vessels
    )
 OPENQUERY([Heart],'@TrainDataPattern ')
```

3.5.4 Duomenų gavybos modelio prognozavimo šablonas

Duomenų gavybos MDX prognozavimo šablonas naudojamas skaičiuojant modelio įvertinimą. Priklausomai nuo klasifikavimo uždavinio gali būti pasirikti skirtingi prognozuojami atributai. Šiuo atveju AClass atributas žymi tikrąją atributo reikšmę, o

atributas PClass prognozuojamą reikšmę. Parametru @MiningModel nurodomas duomenų gavybos modelio pavadinimas. OPENQUERY sintakse kaip ir duomenų gavybos modelio apmokymo šablone atliekamas duomenų perdavimas iš SQL serverio duomenų bazės į Analizės servisus. Parametras @TestDataPattern žymi apmokymo duomenų šabloną(žr pavyzdį esantį žemiau).

Prognozavimo šablono pavyzdys:

```
SELECT
   t.Class as AClass,
   Predict(@MiningModel.[Class]) as PClass
FROM
  @MiningModel
NATURAL PREDICTION JOIN
OPENQUERY([Heart],'@TestDataPattern') as t
```

3.5.5 Apmokymo ir testavimo duomenų šablonai

Duomenys modelio apmokymui turi būti paimti iš tam tikro šaltinio. Šiuo atveju ligonių klasifikavimo uždavinio duomenys saugomi SQL Server reliacinėje duomenų bazėje. Todėl šiuo atveju apmokymo duomenų šablonas yra SQL užklausa(žr. žemiau esantį pavyzdį).

Apmokymo duomenų šablonas yra naudojamas modelio apmokymo šablone ir žymimas parametru @TrainDataPattern. Šiame šablone parametras @TrainDataCnt nusako apmokymo duomenų kiekį. Šiuo parametru valdomas modelio apmokymui naudojamų duomenų kiekis. K-kryžminės strategijos modelių įvertinimo atveju parametrais @KPartitions ir @CurrentPartition valdomas duomenų suskaidymas intervalais(žr. skyrių K-kryžminio patvirtinimo strategija). Parametras @KPartitions žymi intervalų skaičių. Parametras @CurrentPartition žymi testavimo duomenų intervalą.

Apmokymo duomenų šablonas:

```
select TOP(@TrainDataCnt)
    Age ,
    Angina ,
    Cholesteral ,
    Class ,
    Ecg ,
    kId ,
    Oldpeak ,
    Pain ,
    Rate ,
    SEX ,
    Slope ,
    Sugar ,
    Thal ,
    Trestbps ,
    Vessels
```

```
 from
(
select
(ROW_NUMBER()OVER(ORDER BY kId))%@KPartitions as [Partition],
* from dbo.[training_data]
) as D
where [Partition] <> @CurrentPartition
```

Testavimo duomenų šablonas kaip ir apmokymo duomenų šablonas yra SQL užklausa reliacinėje duomenų bazėje. Testavimo šablonas yra naudojamas prognozavimo šablone ir žymimas parametru `@TestDataPattern`. Šiame šablone parametru @TestDataCnt nusakomas testavimo duomenų kiekis(žr pavyzdį esantį žemiau).

Testavimo duomenų šablonas:

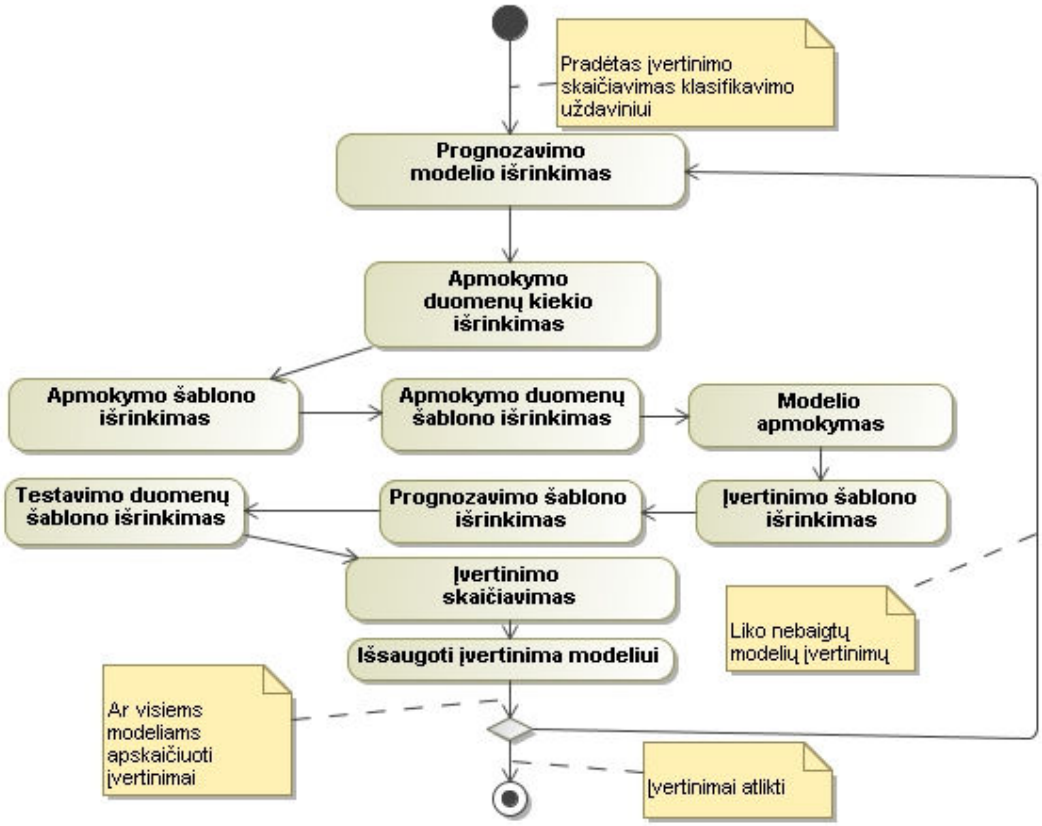
```
Select  TOP(@TestDataCnt)
    Age ,
    Angina ,
    Cholesteral ,
    Class ,
    Ecg ,
    kId ,
    Oldpeak ,
    Pain ,
    Rate ,
    SEX ,
    Slope ,
    Sugar ,
    Thal ,
    Trestbps ,
    Vessels
 from
(
select
* FROM [testing_data]
) as D
```

3.6 Modelio naudojamo prognozavimo algoritmų įvertinimui veikimo principas

Prognozavimui naudojamų algoritmų įvertinimo modelis paremtas reliacinės duomenų bazės schema. 3.4 skyriuje pateikta šio modelio duomenų bazės schema. Naudojantis šia schema sukurtoje reliacinėje duomenų bazėje saugomi modelių sukūrimui, apmokymui, prognozavimui, apmokymo ir testavimo duomenų išrinkimui ir įvertinimui reikalingi šablonai. Vieną kartą sukurti ir toliau modifikuojant šie šablonai pakartotinai gali būti panaudoti prognozavimo modelių įvertinimui panašiems klasifikavimo uždaviniams.

Atliekant įvertinimą, panaudojant modelyje esančia informacija apie šablonus, turi

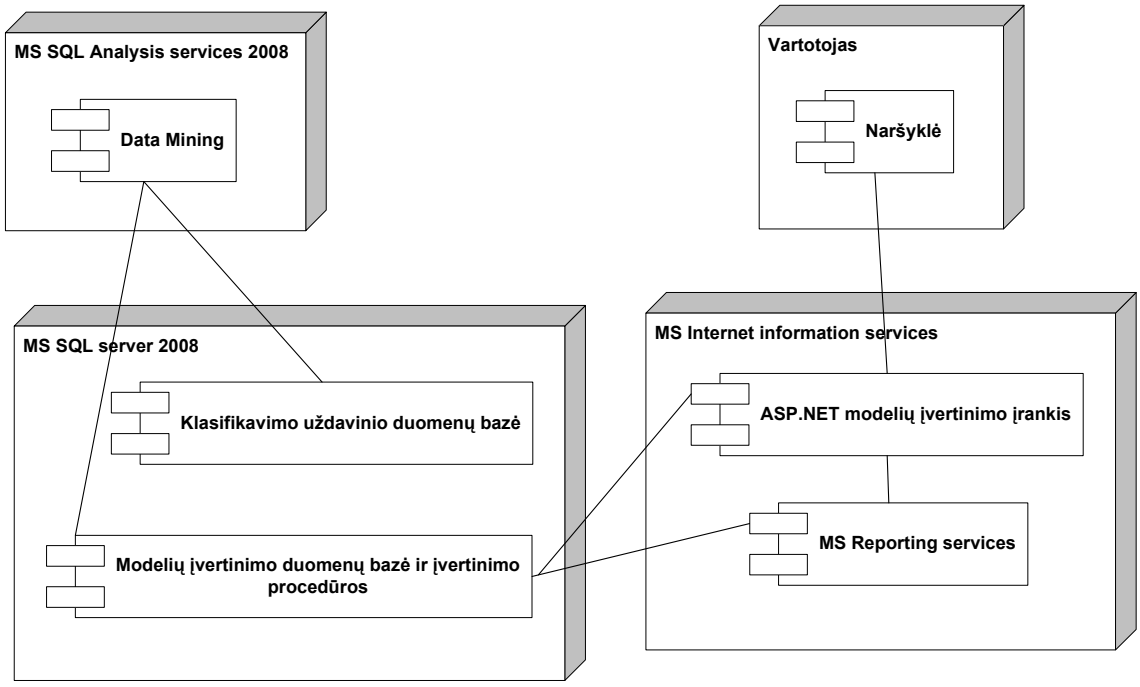
būti atlikti šie veiksmai: klasifikavimo uždavinyje įvertinimui naudojamų prognozavimo modelių išrinkimas; modelio apmokymo duomenų kiekio kitimo išrinkimas; prognozavimo modelio apmokymo šablono išrinkimas; prognozavimo modelio apmokymo duomenų šablono išrinkimas; prognozavimo modelio apmokymas; prognozavimo modelio įvertinimo šablono išrinkimas; prognozavimo šablono išrinkimas modeliui; testavimo duomenų šablono išrinkimas modeliui; įvertinimo skaičiavimas ir išsaugojimas. Modelių įvertinimo veiklos diagramoje galima pamatyti kaip susiję veiksmai(3.11 pav.).



3.11 pav. Įvertinimo veiklos diagrama

3.7 Algoritmų analizės sistemos architektūra

Klasifikavimo uždavinių modelių valdymas ir įvertinimas vartotojo kompiuteryje atliekamas naršykle. Naršyklė komunikuoja su ASP.NET programa, skirta modelių valdymui, įvertinimui ir rezultatų užkrovimui. „MS SQL Analysis services“ vykdomi DMX šablonai, kurie saugomi modelių įvertinimų duomenų bazėje. Modelių įvertinimų ataskaitos generuojamos naudojant „MS Reporting services“. Duomenų šaltinis modelių apmokymui ir testavimui šiuo atveju yra reliacinė klasifikavimo uždavinio duomenų bazė. 3.12 paveiksle pateikta algoritmų analizės sistemos architektūrinė schema.



3.12 pav. Algoritmų analizės sistemos architektūrinė schema

4. SUKURTOS SISTEMOS TAIKYMAS KLASIFIKAVIMO UŽDAVINIUI

4.1 Kraujotakos ligomis sergančių pacientų klasifikavimo uždavinys

Yra daugybė klasifikavimo uždavinių. Tačiau norint atlikti prognozavimo modelių įvertinimą konkrečiam uždaviniui reikia turėti su uždaviniu susijusius istorinius duomenis modelių apmokymui ir testavimui. Kraujotakos ligomis sergančių pacientų klasifikavimo uždavinio duomenys viešai prieinami internete adresu http://cervisia.org/machine_learning_data.php. Šiame puslapyje galima rasti ir kitų duomenų tinkančių klasifikavimo algoritmų palyginimui.

Pacientų klasifikavimo uždavinyje pagal pacientų informaciją ir jiems atliktų kraujotakos tyrimų rezultatus reikia nustatyti, ar pacientas serga kraujotakos ligomis. Pacientų klasifikavimo duomenys buvo importuoti į SQL Server 2008 duomenų bazės lentelę „Pacientų lentelės atributų aprašymai“.

4.1 lentelė

Pacientų lentelės atributų aprašymai

Atributo pavadinimas	Tipas	Diskretumas	Aprašymas
Age	LONG	Tolygus	Paciento amžius metais
Sex	Boolean	Diskretus	Paciento lytis. Galimos reikšmės: taip - vyras; ne – moteris.
Pain	TEXT	Diskretus	Paciento jaučiamo krūtinės skausmo tipas. Galimi atvejai: tipiška krūtinės angina, netipiška krūtinės angina, nebūdingas krūtinės anginos skausmas, neturintis ligos požymių skausmas.
Trestbps	LONG	Tolygus	Kraujo spaudimas ramybės būsenoje
Cholesteral	LONG	Tolygus	Cholesterolio koncentracija
Sugar	Boolean	Diskretus	Ar gliukozės koncentracija kraujyje < 120 mg/dl. Galimi reikšmės: 1 - taip; 0 – ne.
ECG	TEXT	Diskretus	Elektrokardiogramos rezultatas ramybės būsenoje. Galimos reikšmės: normali, nukrypimai, dideli nukrypimai;
Rate	LONG	Tolygus	Treniruotės metu pasiektas maksimalus širdies susitraukimų dažnis
Angina	Boolean	Diskretus	Ar treniruotė sukėlė krūtinės skausmą. Galimos reikšmės: 1 - taip; 0 – ne.
Oldpeak	LONG	Tolygus	Treniruočių metu kardiogramoje rastų nukrypimų (ST depression) skaičius
Slope	TEXT	Diskretus	Intensyviausios treniruotės metu širdies susitraukimų dažnio kitimas. Galimi kitimo reikšmės: didėjantis, tolygus, mažėjantis.
Vessels	LONG	Diskretus	Arterijų, venų ir limfagyslių įvertinimas. Galimos reikšmės: 0 – visos sveikos, 1-3 pakitusių kiekis.
Thal	TEXT	Diskretus	Treniruočių metu užfiksuoti širdies veiklos defekto tipai. Defektų tipai: nerasta defektu, nuolatinis defektas, atsistatantis defektas.
Class	LONG	Diskretus	Klasė, kuriai priskiriamas pacientas. Galimos klasės: 1 - serga širdies ligomis, 0 - sveikas.

4.2 Klasifikavimo modelių įvertinimo šablonų sudarymas

Sudarant įvertinimo šablonus, svarbu žinoti, kokia įvertinimo metodika bus naudojama. Širdies ligomis sergančių pacientų uždavinio atveju klasifikavimo modeliams palyginti naudojama sumaišymo matrica (2.3.5 sk. „Sumaišymo matrica“). Pacientų klasifikavimo uždaviniui apibrėšime TP, TN, FP ir FN sumaišymo matricos įvertinimų formulėse naudojamus kintamuosius:

TP – sergančių pacientų skaičius prognozavimo modelio pripažinti sergančiais;

TN- sveikų pacientų skaičius prognozavimo modelio pripažinti sveikais;

FP- sergančių pacientų skaičius prognozavimo modelio pripažinti sveikais;

FN – sveikų pacientų skaičius prognozavimo modelio pripažinti sergančiais.

Pacientų klasifikavimo uždaviniui svarbu teisingai parinkti modelio įvertinimo skaičiavimą. Reikia atsižvelgti į tai, kad modelis kuo geriau klasifikuotų tikrai sergančius ligonius ir kad kuo mažiau sergančių pacientų būtų pripažinti sveikais. Sveikų pacientų pripažinimo sergančiais atvejų skaičius mažiau svarbus, bet geriau, kad jis būtų kuo mažesnis, nes tuomet bereikalingai gaištamas gydytojo laikas ir taip gali būti pakenkta sveikam ligoniui. Kadangi svarbūs ir TP, ir TN kintamieji, klasifikavimo uždavinio modeliai bus palyginami naudojant prognozavimo teisingumo įvertinimą $P = \frac{TP + TN}{TP + TN + FP + FN}$.

4.2.1 K - kryžminės strategijos įvertinimo šablonas

K- kryžminės strategijos modelio įvertinimo atveju apmokymas ir testavimas atliekamas skaidant turimą apmokymo duomenų rinkinį į k intervalų. Kiekvienas šių intervalų paeiliui naudojamas testavimui(2.3.4 sk. „K ir N kartų kryžminis patvirtinimas“). Bendras modelio įvertimas skaičiuojamas pagal formulę $P = \frac{\sum_{j=1}^{K} (TP_j + TN_j)}{\sum_{j=1}^{K} (TP_j + TN_j + FP_j + FN_j)}$, kur TP_j , TN_j ir FP_j , FN_j j-tojo testavimo duomenų rinkinio teisingai ir klaidingai klasifikuotų pacientų skaičiai.

TP_j , TN_j ir FP_j , FN_j apskaičiavimo j-oje įvertinimo iteracijoje SQL šablono pavyzdys:

```
select @TP=SUM(TP),@TN=SUM(TN),@FP=SUM(FP),@FN=SUM(FN)
 from(
 Select CASE WHEN AClass=PClass and AClass =1 THEN 1 ELSE 0 END as TP,
           CASE WHEN AClass=PClass and PClass =0 THEN 1 ELSE 0 END as TN,
           CASE WHEN AClass<>PClass and PClass =1 THEN 1 ELSE 0 END as FP,
           CASE WHEN AClass<>PClass and PClass =0 THEN 1 ELSE 0 END as FN
 from
 OPENQUERY(ASServer,@PredictPattern)
 ) as T1
```

SQL šablone naudojama OPENQUERY sintaksė. Jos pagalba įvykdoma MDX prognozavimo užklausa „MS SQL Analysis servises“. MDX prognozavimo užklausa aprašoma parametru @PredictPattern (3.5.4 sk. „Duomenų gavybos modelio prognozavimo šablonas“). Pilną įvertinimo procedūros aprašymą, naudojant k - kryžmine strategija, galima rasti prieduose (8.1 sk. „Pacientų klasifikavimo uždavinio prognozavimo modelių įvertinimo procedūros“).

4.2.2 Testavimo ir apmokymo strategijos įvertinimo šablonas

Testavimo ir apmokymo strategijos naudojami du duomenų šaltiniai vieni skirti apmokymui, kiti testavimui(2.3.3 sk. „Apmokymo ir testavimo duomenų atskyrimo strategija“).

Pacientų klasifikavimo uždavinyje naudojamų modelių prognozavimo teisingumo įvertinimas skaičiuojamas naudojant formulę $P = \frac{TP + TN}{TP + TN + FP + FN}$.

Įvertinimo P apskaičiavimo SQL šablono pavyzdys:

```
select @Calculated_Value=(0.000000+SUM(TP)+SUM(TN))/(SUM(TP)+SUM(TN)+SUM(FP)+SUM(FN))
from(
Select CASE WHEN AClass=PClass and AClass =1 THEN 1 ELSE 0 END as TP,
     CASE WHEN AClass=PClass and PClass =0 THEN 1 ELSE 0 END as TN,
     CASE WHEN AClass<>PClass and PClass =1 THEN 1 ELSE 0 END as FP,
     CASE WHEN AClass<>PClass and PClass =0 THEN 1 ELSE 0 END as FN
from
OPENQUERY(ASServer,@PredictPattern)
) as T1
```

Įvykdžius SQL užklausą parametrui @Calculated_Value priskiriama paskaičiuota įvertinimo P reikšmė. Parametras @PredictPattern nusako MDX prognozavimo šabloną. Pilną įvertinimo procedūros aprašymą, panaudojant apmokymo ir testavimo strategija, galima rasti prieduose (8.1 sk. „Pacientų klasifikavimo uždavinio prognozavimo modelių įvertinimo procedūros“).

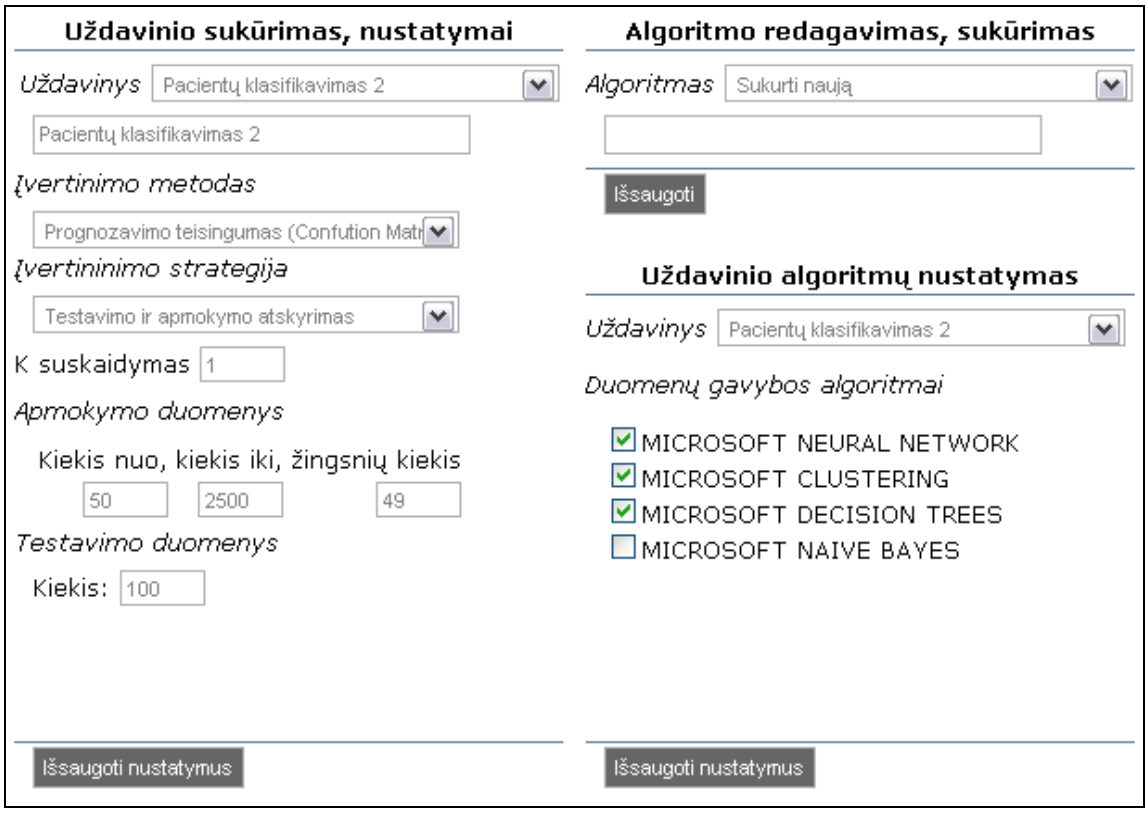
4.3 Pacientų klasifikavimo modelio pasirinkimas

Pacientų klasifikavimo uždaviniui spręsti galima rinktis iš keleto prognozavimo algoritmų „Neuroninių tinklų“, „Sprendimų medžių“, „K - vidurkių grupavimas“ ir kitų. Gali būti daug įvairių būdų, kaip pasirinkti algoritmą. Šiame darbe sukurto įrankio tikslas pagreitinti duomenų gavybos modelių pasirinkimą naudojant įvertinimo metodikas ir strategijas, bei naudojant DMX ir SQL modelių šablonus.

Klasifikavimo modelių įvertinimo rezultatai gali priklausyti nuo pasirinktos įvertinimo metodikos ir strategijos, nuo modelio apmokymo duomenų kiekio ir modelyje naudojamų atributų. Prieš naudojant prognozavimo modelį informacinėse sistemose, būtina įsitikinti, ar naudojamas geriausias algoritmas, ar prognozavimo modelis tinkamai apmokytas pacientų klasifikavimui. Šiame skyriuje aprašomas eksperimentas ir mėginama atsakyti į svarbiausius prognozavimo modelio išrinkimo ir paruošimo veikti informacinėje sistemoje klausimus.

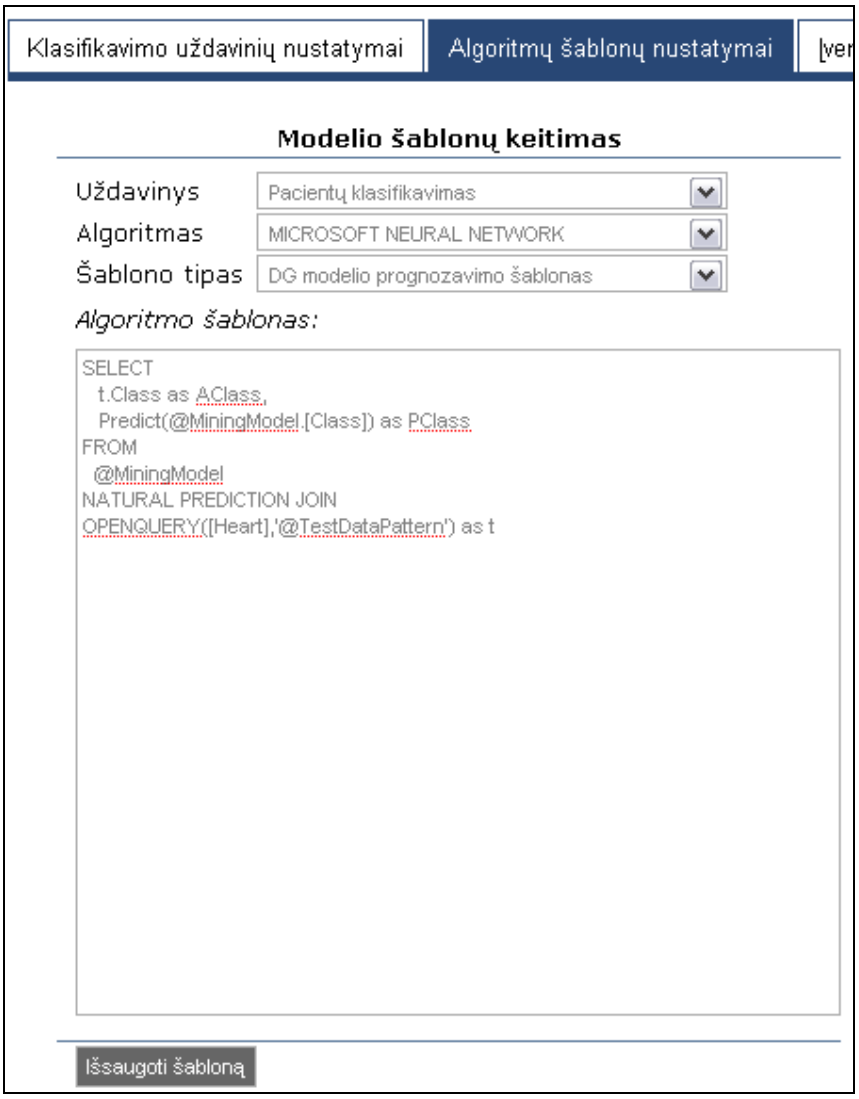
4.3.1 Įvertinimų skaičiavimas pacientų klasifikavimo uždavinio modeliams

Naudojant sukurtą algoritmų analizės sistemą galima nustatyti, kaip keičiantis apmokymo duomenų kiekiui kinta pacientų klasifikatorių prognozavimo teisingumas. Norėdami atlikti eksperimentą pasirenkame apmokymo duomenų kiekį nuo ir iki, bei analizuojamus duomenų gavybos algoritmus (4.1 pav.). Šiame pacientų klasifikatorių palyginimo eksperimente naudosime prognozavimo teisingumo įvertinimo metodą, bei testavimo ir apmokymo įvertinimo strategiją. Algoritmų modelių apmokymui ir testavimui naudojami skirtingi duomenys, tačiau duomenų kiekis yra toks pats. Šis kiekis apmokymo ir testavimo duomenų šablonuose apibrėžiamas parametru @TrainDataCnt. Parametras @TrainDataCnt kinta priklausomai nuo pasirinkto apmokymo duomenų kiekio nuo, iki ir žingsnių skaičiaus. Kadangi apmokymo ir testavimo duomenų išrinkimo šablonuose naudojamas tas pats parametras @TrainDataCnt, į nurodytą testavimo duomenų kiekį šio eksperimento metu nebus atsižvelgiama.



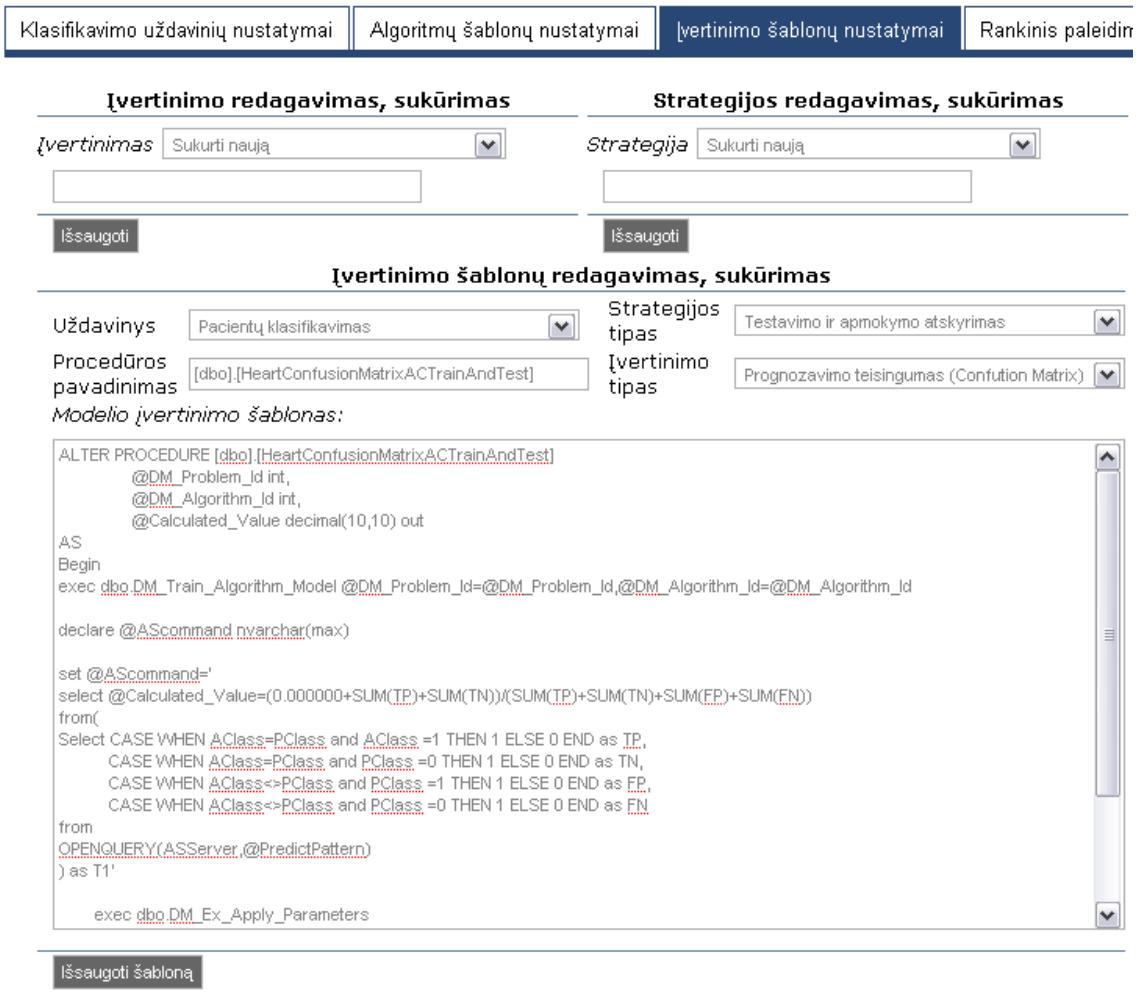
4.1 pav. Algoritmų modelių įvertinimo nustatymai

Pirmą kartą sukūrus klasifikavimo uždavinį ir nurodžius testavimo parametrus turi būti sukurti ir išsaugoti prognozavimo algoritmų valdymo ir įvertinimo šablonai (žr. 3.5 sk., 4.2 sk.). Norint išsaugoti algoritmų valdymo šabloną pasirenkamas prognozavimo uždavinys, prognozavimo algoritmas, šablono tipas, aprašomas šablonas ir išsaugomas (4.2 pav.).



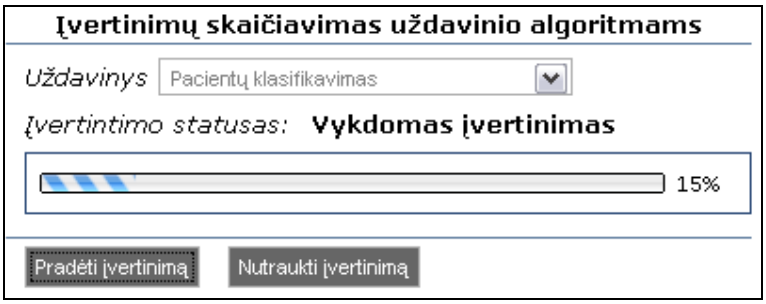
4.2 pav. Algoritmo valdymo šablono išsaugojimas

Prieš atliekant prognozavimo uždavinio algoritmų įvertinimų skaičiavimą reikia išsaugoti įvertinimo šabloną. Pasirenkamas prognozavimo uždavinys, įvertinimo strategijos tipas, įvertinimo šablonas, aprašomas procedūros, kuri skaičiuos įvertinimą pavadinimas, aprašas ir išsaugoma (4.3 pav.).



4.3 pav. Prognozavimo uždavinio algoritmų įvertinimo šablono išsaugojimas

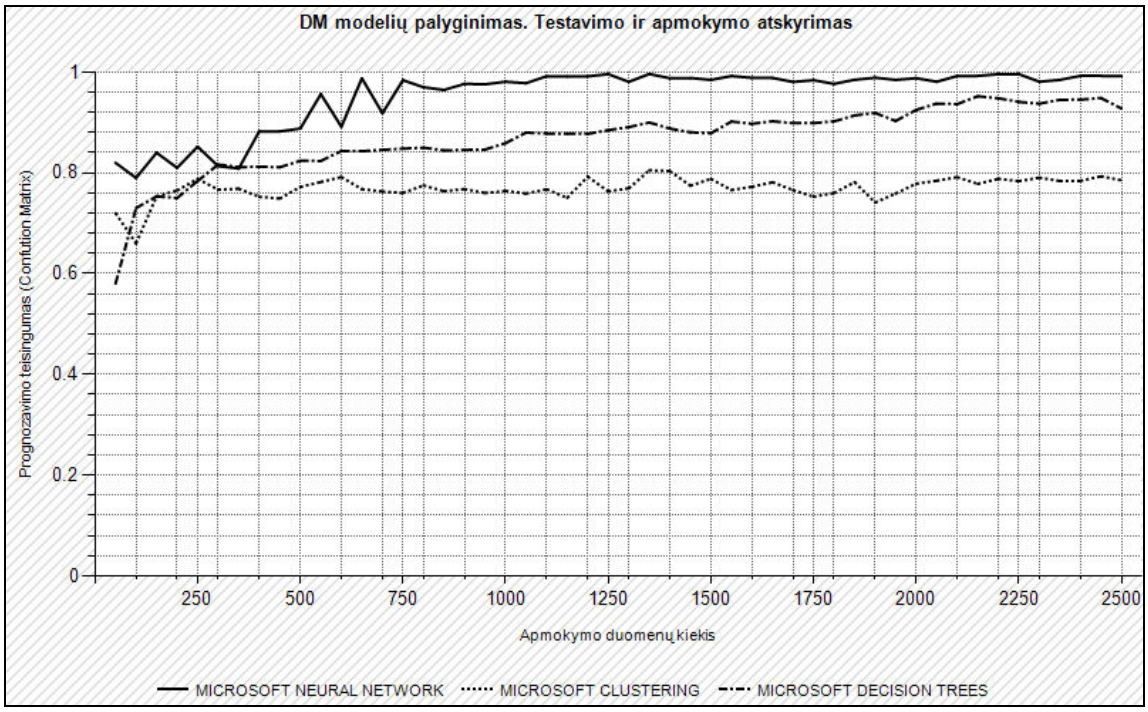
Po pacientų klasifikavimo uždavinio testavimo nustatymų ir šablonų išsaugojimo paleidžiamas įvertinimų skaičiavimas (4.4 pav.)



4.4 pav. Įvertinimų skaičiavimo paleidimas

Pasibaigus įvertinimų skaičiavimui rezultatai pateikiami grafiškai (4.5 pav.)

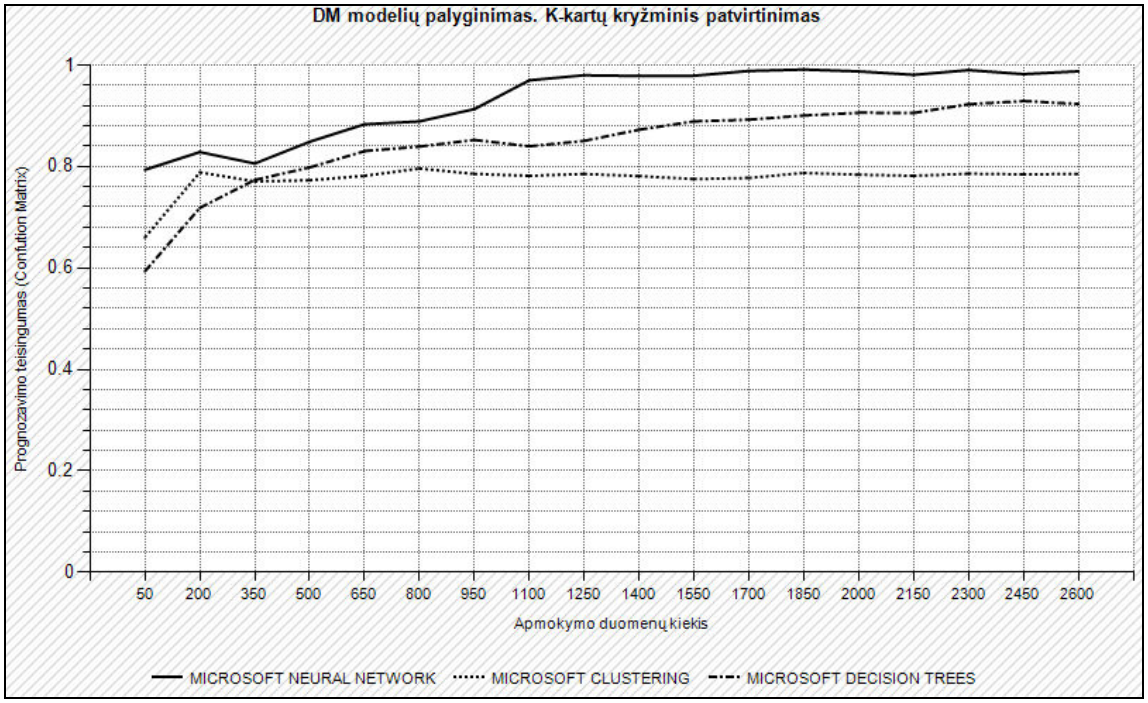
4.3.2 Klasifikavimo modelių įvertinimų priklausomybės nuo modelių apmokymo duomenų kiekio rezultatai



4.5 pav. Duomenų gavybos modelių palyginimo grafikas. Testavimo ir apmokymo strategija

Pacientų klasifikavimo uždavinio palyginamų modelių grafike (4.5 pav.) matyti, kad geriausias prognozavimo tikslumas pasiekiamas naudojant neuroninių tinklų algoritmą. Kaip matyti 4.5 paveiksle, didėjant neuroninio tinklo modelio apmokymo duomenų kiekiui, prognozavimo teisingumo įvertinimas didėja ir pradedant nuo tam tikro apmokymo duomenų kiekio išlaiko maždaug vienodą reikšmę. Sprendimų medžio algoritmo prognozavimo teisingumas taip pat didėja didėjant apmokymo duomenų kiekiui, tačiau ženkliai atsilieka lyginant su neuroninių tinklų algoritmu tam pačiam duomenų kiekiui. K vidurkių grupavimo algoritmas („MICROSOFT CLUSTERING“) išlaiko maždaug vienodą tikslumą ir nesikeičia.

Atlikę dar vieną eksperimentą modelių prognozavimo teisingumo palyginimui naudodami K – kryžminę įvertinimo strategiją gavome panašius rezultatus (4.6 pav.). Šio eksperimento metu buvo naudojamas tik apmokymo duomenų šaltinis. K - kryžminės strategijos įvertinimo metu duomenys apmokymui ir testavimui buvo skaidomi į 3 dalis. Dvi apmokymo duomenų dalys buvo naudojamos apmokymui, o likusi testavimui.



4.6 pav. Duomenų gavybos modelių palyginimas. K - kryžminė įvertinimo strategija

K – kryžminės įvertinimo strategijos atveju matome, kad neuroninių tinklų modelio prognozavimo teisingumo įvertinimas kinta panašiai kaip ir naudojant apmokymo ir testavimo įvertinimo strategiją (4.5 pav.). Sprendimų medžio algoritmo modelio („Microsoft Decision Trees“) įvertinimas didėja didėjant apmokymo duomenų kiekiui. K - vidurkių grupavimo algoritmas („MICROSOFT CLUSTERING“) kaip ir ankstesniame eksperimente išlaiko panašų įvertinimą.

4.3.3 Tinkamiausio prognozavimo modelio ligonių klasifikavimui pasirinkimas ir paklaidų įvertinimas

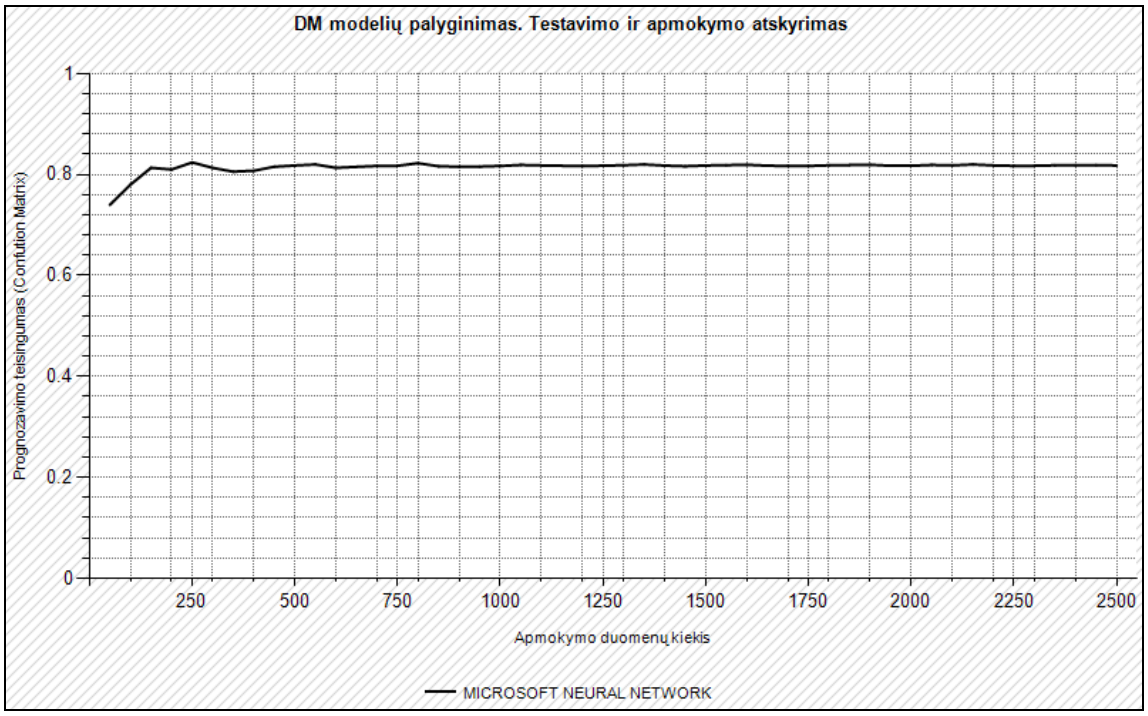
Apskaičiavus pacientų klasifikavimo uždavinio įvertinimus „Neuroninių tinklų“, „Sprendimų medžių“ ir „K – vidurkių grupavimo“ algoritmams paaiškėjo, kad geriausiai klasifikavimą atlieka neuroninių tinklų algoritmo modelis. Tačiau pasirinkus netinkama apmokymo duomenų kiekį net ir neuroninių tinklų algoritmas gali pasiekti panašų prognozavimo teisingumą kaip ir kiti du algoritmai. Iš grafikų 4.5 ir 4.6 paveiksluose matyti, kad pacientų klasifikavimo uždaviniui naudojamo neuroninių tinklų modelio įvertinimas pradedant duomenų kiekiu $dk_0 = 1100$ išlieka maždaug pastovus. Tarkim, kad pacientų klasifikavimo uždavinio modelio prognozavimo teisingumo įvertinimas turi būti didesnis už

$\varepsilon_0 = 0.96$. Neuroninio tinklo modelio įvertinimo reikšmė didesnė už ε_0 yra $\varepsilon_1 = 0.979733$, kai modelis buvo vertinamas naudojant K – kryžminio patvirtinimo strategija ir apmokymo duomenų kiekį $dk_1 = 1250$.

Paskaičiuosime prognozavimo teisingumo įvertinimo ε_1 paklaidą. Teisingo įvertinimo intervalas apibrėžiamas $p \pm Z_{CL} \cdot S$, kur p – prognozavimo modelio teisingumo įvertinimas. Standartinis nuokrypis S apskaičiuojamas pagal formulę $\sqrt{p(1-p)/N}$. N yra bendras K-kryžminėje strategijoje naudotų testavimo duomenų skaičius. Reikšmė Z_{CL} nustatoma priklausomai nuo pasirinkto patikimumo lygio (2.3.2 skyrius). Kai pasirinktas patikimumo lygis 99%, teisingumo įvertinimas neuroninių tinklų prognozavimo modeliui yra intervale 0.979 ± 0.010. Kai $\varepsilon_0 = 0.96$ ir $\varepsilon_0 < 0.979 \pm 0.010$ neuroninių tinklų modelio apmokymui turėtų būti naudojamas apmokymo duomenų kiekis $dk_1 = 1250$.

4.3.4 Įvertinimų priklausomybės nuo modelyje naudojamų atributų eksperimentas

Atliksime neuroninių tinklų prognozavimo modelio įvertinimų kitimo, nuo atributo eksperimentą. Iš modelio pašalinsime atributą „Angina“, kuris žymi ar treniruotė sukėlė pacientui krūtinės skausmą. Šiam eksperimentui naudojama apmokymo ir testavimo strategija. Naudodami tuos pačius apmokymo duomenų kitimo rėžius kaip ir ankstesniuose eksperimentuose (4.3.1 skyrius) įrankiu buvo apskaičiuoti teisingumo įvertinimai, kurie matomi apačioje.



4.7 pav. Neuroninių tinklų modelio įvertinimų priklausomybė nuo apmokymo duomenų kiekio. Iš modelio pašalintas atributas “Angina”.

Kaip matome 4.7 paveiksle atributo “Angina” neįtraukimas į neuroninių tinklų prognozavimo modelį žymiai sumažina teisingumo įvertinimus, kurie yra mažesni už 0.82. Modelio įvertinimai, šiame eksperimente, pradeda lėtai kisti nuo žymiai mažesnio apmokymo duomenų kiekio lyginant su 4.3.2 skyriuje atliktais eksperimentais. Pasirinkus neuroninių tinklų modelį kai nenaudojamas atributas “Angina” modelio apmokymo duomenų skaičius galėtų būti mažesnis, tačiau modelis prognozuos kur kas prasčiau, apskaičiuotas mažesnis teisingumo įvertinimas.

5. IŠVADOS

1. Pateikiant sistemos modelį ir sukuriant bandomąją sistemą, kuri leidžia atlikti kelių algoritmų analizę, parodyta, kaip duomenų analizės procese konkrečiam prognozavimo uždaviniui reikia pasirinkti prognozavimo algoritmus.

2. Darbe duomenų gavybos modelių sukūrimui, įvertinimui ir geriausio klasifikavimo metodo parinkimui sukurtas vieningas modelis. Šio modelio daugkartinis panaudojimas pagreitintų DMX šablonų taikymą ir duomenų gavybos CRISP-DM proceso modeliavimo, įvertinimo ir diegimo etapus.

3. Sukurtoje prognozavimo algoritmų palyginimo sistemoje naudojami šešių tipų šablonai: duomenų gavybos modelio struktūros sukūrimo šablonas; duomenų gavybos modelio sukūrimo šablonas; duomenų gavybos modelio apmokymo šablonas; duomenų gavybos modelio prognozavimo šablonas; apmokymo duomenų šablonas; testavimo duomenų šablonas. Taip pat naudojamas modelių įvertinimo šablonas, kuris priklauso nuo pasirinkto įvertinimo metodo ir įvertimo strategijos.

4. Pacientų klasifikavimo uždaviniui sukūrus DMX ir SQL algoritmų modelių valdymo, apmokymo, įvertinimo šablonus ir atlikus algoritmų analizę, naudojant sumaišymo matricos teisingumo įvertinimą, bei įvertinimo strategijas „K – kryžminis patvirtinimas“ ir „Apmokymo ir testavimo atskyrimas“, paaiškėjo, kad geriausiai klasifikuoja „Neuroninių tinklų“ algoritmas.

5. Algoritmo modelio prognozavimo teisingumo įvertinimas priklauso nuo apmokymo duomenų kiekio. Diegiant prognozavimo modelį informacinėje sistemoje apmokymo duomenų kiekis turi būti parinktas atsižvelgiant į norimą teisingumo įvertinimą.

6. Atlikus teisingumo įvertinimo nuo „Neuroninių tinklų“ prognozavimo modelyje naudojamų atributų priklausomybės analizę paaiškėjo, kad teisingumo įvertinimas gali mažėti arba didėti priklausomai nuo teisingai parinktų atributų.

7. Pagal koncepcinį modelį sukurta prognozavimo algoritmų palyginimo sistema sėkmingai gali būti panaudota daugkartiniam klasifikavimo uždavinių algoritmų parinkimui. Kadangi algoritmų analizės sistemoje naudojami šablonai, sistema neapribota konkretaus prognozavimo uždavinio algoritmų analizei. Tereikia perrašyti šablonus atsižvelgiant į kito uždavinio reikalavimus.

8. Perskaičiuojant įvertinimus galima stebėti algoritmų darbo informacinėse sistemose efektyvumą. Tai galima atlikti aprašius testavimo duomenų šabloną, kuriuo turėtų būti išrenkami sistemoje besikeičiantys duomenys.

6. LITERATŪRA

1. Duomenų gavybos apklausos. Nuoroda internete. http://www.kdnuggets.com/polls/index.html
2. Informacija apie CRISP-DM procesą. Nuoroda internete. http://www.crisp-dm.org/
3. CRAWFORD, J. ir CRAWFORD, F. „Data Mining in a Scientific Environment“. Nuoroda internete. http://www.csu.edu.au/special/auugwww96/proceedings/crawford/crawford.html.
4. IAN H. WITTEN ir EIBE FRANK „Data Mining Practical Machine Learning Tools and Techniques“ 2005 Puslapių skaičius – 525 psl.
5. Informacija apie duomenų gavybos metodus. Nuoroda internete. http://www.thearling.com/text/dmtechniques/dmtechniques.htm
6. Straipsniai apie duomenų gavybą, terminų apibrėžimai. Nuoroda internete. http://databases.about.com/od/datamining/a/datamining.htm
7. MAX BRAMER „Principles of Data Mining“ 2007 Puslapių skaičius – 343 psl.
8. DAVID L. OLSON ir DURSUN DELEN „Advanced Data Mining Techniques“ 2009 Puslapių skaičius – 180 psl.
9. Informacija apie duomenų gavybą Oracle duomenų bazėje. Nuoroda internete http://www.filibeto.org/sun/lib/nonsun/oracle/11.1.0.6.0/B28359_01/datamine.111/b28129/intro_concepts.htm#i1023970
10. Informacija apie duomenų gavybą naudojant Microsoft SQL Server analizės servisus. Nuoroda internete. http://msdn.microsoft.com/en-us/library/bb510517.aspx
11. Informacija apie DMX kalba darbui su prognozavimo modeliais. Nuoroda internete. http://msdn.microsoft.com/en-us/library/ms132058.aspx
12. Informacija apie DMX užklausas prognozavimui. Nuoroda internete http://msdn.microsoft.com/en-us/library/ms131992.aspx
13. Informacija apie DMX algoritmų modelius. Nuoroda internete. http://msdn.microsoft.com/en-us/library/bb895228.aspx
14. DAVID HAND; HEIKIKI MANNILA ir PADHRAIC SMYTH „Principles of Data Mining“ 2001 Puslapių skaičius – 546 psl.
15. Paolo Giudici „Applied Data Mining“ 2003 Puslapių skaičius – 364 psl.

7. TERMINU IR SANTRUMPŲ ŽODYNAS

7.1 SANTRUMPOS

Santrumpa	Apibūdinimas
DDL	Data Definition Language
DML	Data Manipulation Language
DMX	Data Mining Extensions
SQL	Structured Query Language
XMLA	XML for Analysis is the industry standard for data access in analytical systems
HTTP	Hypertext Transfer Protocol
TCP	Transmission Control Protocol
SOAP	Simple Object Access Protocol

7.2 TERMINAI

Terminas	Terminas anglų kalba	Apibūdinimas
Duomenų gavyba	Data Mining	Duomenų analizės procesas anksčiau nežinotų ir galimai naudingų žinių išgavimui ir taikymui.
Veiklos intelektas	Buisness Intellegence	Technologijos ir programinės įrangos taikomos duomenų surinkimui, analizei, integracijai ir atvaizdavimui.
Duomenų gavybos modelis	Data Mining Model	Objektas savo struktūra panašus į reliacinę lentelę, kuris turi raktų, įėjimų ir prognozavimo atributus.

8. PRIEDAI

8.1 Pacientų klasifikavimo uždavinio prognozavimo modelių įvertinimo procedūros

Algoritmų modelių panaudojant k-kryžminę įvertinimo strategija prognozavimo teisingumo įvertinimo procedūra

```
ALTER PROCEDURE [dbo].[HeartConfusionMatrixACKCross]
        @DM_Problem_Id int,
        @DM_Algorithm_Id int,
        @Calculated_Value decimal(10,10) out
AS
Begin

  declare @KPartitions int
  Select @KPartitions=K from dbo.DM_Problem where DM_Problem_Id=@DM_Problem_Id
  declare @CurrentPartition int
  declare @AScommand nvarchar(max)
  declare @AScommandPattern nvarchar(max)

  set @AScommandPattern='
  select @TP=SUM(TP),@TN=SUM(TN),@FP=SUM(FP),@FN=SUM(FN)
  from(
  Select CASE WHEN AClass=PClass and AClass =1 THEN 1 ELSE 0 END as TP,
           CASE WHEN AClass=PClass and PClass =0 THEN 1 ELSE 0 END as TN,
           CASE WHEN AClass<>PClass and PClass =1 THEN 1 ELSE 0 END as FP,
           CASE WHEN AClass<>PClass and PClass =0 THEN 1 ELSE 0 END as FN
  from
  OPENQUERY(ASServer,@PredictPattern)
  ) as T1'

  declare @TPS int=0
  declare @TNS int=0
  declare @FPS int=0
  declare @FNS int=0

  set @CurrentPartition = 0
  while(@CurrentPartition<@KPartitions)
  begin
        declare @TP int=0
        declare @TN int=0
        declare @FP int=0
        declare @FN int=0

        exec dbo.DM_Train_Algorithm_Model @DM_Problem_Id=@DM_Problem_Id,
        @DM_Algorithm_Id=@DM_Algorithm_Id,
           @CurrentPartition= @CurrentPartition

        set @AScommand=@AScommandPattern
```

```
          exec dbo.DM_Ex_Apply_Parameters @DM_Problem_Id=@DM_Problem_Id,
          @DM_Algorithm_Id=@DM_Algorithm_Id,@AScommand=@AScommand out,
             @CurrentPartition= @CurrentPartition

          exec sp_executeSql @ASCommand,
          N'@TP int OUTPUT,@TN int OUTPUT,
          @FP int OUTPUT,@FN int OUTPUT',
          @TP output,@TN output,@FP output,@FN output

          set @TPS=@TPS+@TP
          set @TNS=@TNS+@TN
                 set @FPS=@FPS+@FP
          set @FNS=@FNS+@FN

        set @CurrentPartition =@CurrentPartition +1

  end

  set @Calculated_Value=(0.000000+@TPS+@TNS)/(@TPS+@TNS+@FPS+@FNS)

END
```

Algoritmų modelių panaudojant apmokymo ir įvertinimo strategija prognozavimo teisingumo įvertinimo procedūra

```
ALTER PROCEDURE [dbo].[HeartConfusionMatrixACTrainAndTest]
         @DM_Problem_Id int,
         @DM_Algorithm_Id int,
         @Calculated_Value decimal(10,10) out
AS
Begin
Execdbo.DM_Train_Algorithm_Model @DM_Problem_Id=@DM_Problem_Id,
@DM_Algorithm_Id=@DM_Algorithm_Id
declare @AScommand nvarchar(max)
set @AScommand='
select @Calculated_Value=(0.000000+SUM(TP)+SUM(TN))/(SUM(TP)+SUM(TN)+SUM(FP)+SUM(FN))
from(
Select CASE WHEN AClass=PClass and AClass =1 THEN 1 ELSE 0 END as TP,
     CASE WHEN AClass=PClass and PClass =0 THEN 1 ELSE 0 END as TN,
     CASE WHEN AClass<>PClass and PClass =1 THEN 1 ELSE 0 END as FP,
     CASE WHEN AClass<>PClass and PClass =0 THEN 1 ELSE 0 END as FN
from
OPENQUERY(ASServer,@PredictPattern)
) as T1'

  exec dbo.DM_Ex_Apply_Parameters @DM_Problem_Id=@DM_Problem_Id,
        @DM_Algorithm_Id=@DM_Algorithm_Id, @AScommand=@AScommand out

exec sp_executeSql @ASCommand,  N'@Calculated_Value decimal(10,10) OUTPUT',
  @Calculated_Value output
END
```