KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Vaidotas Brukštus

Intelektuali universiteto akademinių duomenų analizė MS SQL Server 2008 priemonėmis

Magistro darbas

Darbo vadovas
Vigintas Šakys

Kaunas, 2009

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Intelektuali universiteto akademinių duomenų analizė MS SQL Server 2008 priemonėmis

Programų sistemų inžinerijos magistro baigiamasis darbas
Studijų programa M404DF11

Vadovas:
doc. dr. V. Šakys
2009 05

Recenzentas:
doc. dr. A. Lenkevičius
2009 05

Atliko:
IFM-3/2 gr. stud.
V. Brukštus
2009 05 25

Kaunas, 2009

Brukštus V. (2009) *Intelektuali universiteto akademinių duomenų analizė MS SQL Server 2008 priemonėmis*. Programų sistemų inžinerijos magistro baigiamasis darbas. Studijų programa M404DF11. Vadovas doc. dr. V. Šakys. Kaunas: Kauno technologijos universitetas,Informatikos fakultetas.

Santrauka

Šiame darbe tiriama galimybė analizuoti įtakas, lemiančias studentų mokymosi universitete sėkmę. Remiamasi duomenų gavybos algoritmais. Sukurtas būdas, kaip prognozuoti, ar būsimas studentas, remiantis jo turimais stojimo balais bei ankstesnės kartos patirtimi, sėkmingai užbaigs studijas.

Pradžioje aptariamos galimos duomenų gavybos taikymo sritys, būtini etapai, tam skirta programinė įranga. Detalizuojami Microsoft SQL Server 2008 palaikomi duomenų gavybos algoritmai. Keturi iš jų sėkmingai pritaikyti pasirinktos dalykinės srities analizei. Sukurta analitinė sistema, sugebanti įvertinti stojimo balų įtakas, universitete dėstomų dalykų įtakas galimybei sėkmingai baigti studijas. Atliktas tyrimas, nustatyti, kuris duomenų gavybos algoritmas yra tinkamiausias prognozuoti studentų iškritimą.

Raktiniai žodžiai: Duomenų gavyba, MS SQL Server 2008, akademinių duomenų analizė, būsenų prognozavimas.

Brukštus V. (2009) *Intelligent analysis of university data with MS SQL Server 2008 tools*. Master's Work in Software Engineering. Study Programme M404DF11. Supervisor dr. V. Šakys. Kaunas: Faculty of Informatics, Kaunas University of Technology.

Summary

This paper describes a research of evaluation of influences, causing a success to graduate university. The research is based on the data mining algorithms. There has been developed way to predict if a prospective student will successfully graduate the university or not. The prediction is based on the data of earlier generation of students, and school's marks of prospective student.

First part of paper describes the spheres, where data mining is adapted. Then, there is detailed stages used in data mining process; reviewed most popular data mining tools. After Microsoft SQL Server data mining algorithms were analyzed, it became clear witch ones are most suitable for the selected research area. The realization part explains how data mining can serve to improve the study process in university. This can be achieved by analyzing influences of different study disciplines to the ability to graduate the university. The last part of paper describes the performed experiment, witch showed the most appropriate algorithm to make predictions about ability to graduate the university.

Key words: Data Mining, Microsoft SQL Server, academic data analysis, state prediction.

Turinys

ĮVADAS 10
1. GAVYBOS CIKLO, ĮRANKIŲ, METODIKŲ ANALIZĖ 11
1.1 Taikymo sritys 11
1.2.1. Duomenų surinkimas 12
1.2.2. Duomenų transformacija ir valymas 13
1.2.3. Modelio taikymas 13
1.2.4. Modelio įvertinimas 14
1.2.5. Rezultatų prognozavimas ir paskelbimas 14
1.2.6. Modelio atnaujinimas 14
1.3. Esami įrankiai 14
1.3.1. SAS 15
1.3.2. SPSS 15
1.3.3. IBM 15
1.3.5. Oracle Data Mining 16
1.3.6. Weka 16
1.3.7. Orange 16
1.4. MS SQL Server duomenų gavybos algoritmai 17
1.4.1. Sprendimų medžių algoritmas 17
1.4.2. Grupavimo algoritmas 19
1.4.3. Naive Bayes algoritmas 21
1.4.4. Asociacijų algoritmas 23
1.4.5. Sekų grupavimo algoritmas 23
1.4.7. Neuroninių tinklų algoritmas 24
1.4.8. Logaritminės regresijos algoritmas 26
1.4.9. Tiesinės regresijos algoritmas 26
1.5. Duomenų gavybos algoritmų palyginimas 27
1.6. DMX kalbos struktūra 27
1.7. Analitinės dalies išvados 28
2. SISTEMOS PROJEKTAVIMAS 29

2.1. Tikslai 29
2.2. Sistemos technologinis kontekstas 29
2.3. Panaudojimo atvejai 30
2.3.1 Sistemos administravimo panaudojimo atvejai 30
2.3.2 Sistemos taikymo panaudojimo atvejai 32
2.4 Sistemos elgsenos aspektai 33
2.5 Funkciniai reikalavimai 37
2.6 Nefunkciniai reikalavimai 38
2.7 Reikalavimai duomenims, ER schema 39
3. DUOMENŲ GAVYBOS REALIZACIJA 40
3.1. Saugyklos formavimas 40
3.2. Duomenų vaizdiniai 42
3.3. Analysis Services projektas 43
3.3.1. Nayve Bayes algoritmo taikymas 43
3.3.2. Grupavimo algoritmo taikymas 46
3.3.3. Sprendimų medžio algoritmo taikymas 51
3.3.4. Neuroninio tinklo taikymas būsenų prognozavimui 53
3.4. Algoritmų taikymo apibendrinimas 54
4. GAVYBOS ALGORITMŲ PROGNOZAVIMO KOKYBĖS TYRIMAS 55
4.1. Tyrimo tikslas ir apžvalga 55
4.2. Prognozių rezultatų įvertinimas 55
4.2. Prognozavimo rezultatų apibendrinimas 58
5. IŠVADOS 59
6. NAUDOTA LITERATŪRA 60
7. TERMINŲ IR SANTRUMPŲ ŽODYNAS 62
PRIEDAI 63

Paveikslų turinys

1.1 pav. Pagrindiniai duomenų gavybos žingsniai [2] 12
1.2 pav. Sprendimų medžio šakojimas 17
1.3 pav. Sprendimų medžio algoritmo parametrų langas 18
1.4 pav. Būsenos „1" pasiskirstymas klasteriuose, remiantis stojimo pažymiais 20
1.5 pav. Klasterių savybių detalizavimo fragmentas 20
1.6 pav. Nayve Bayes algoritmo priklausomybių įtakos atvaizdavimas 22
1.7 pav. Tendencijų vizualizavimas su „Time Series" algoritmu[9] 24
1.8 pav. Neuroninio tinklo principinė schema verslo klientų lojalumui įvertinti[19] 25
2.1 pav. Sistemos komponentų pasiskirstymas 30
2.2 pav. Sistemos panaudojimo atvejų diagrama analitikui 31
2.3 pav. Galutinių vartotojų panaudojimo atvejų diagrama 32
2.4 pav. Duomenų paruošimo veiklos diagrama 33
2.5 pav. Duomenų gavybos taikymų veiklos diagrama 34
2.6 pav. Duomenų analizės serviso rengimo veiklos diagrama 35
2.7 pav. Dalyko įvertinimo scenarijus 37
2.8 pav. ER diagrama, charakterizuojanti būsimą saugyklą. 39
3.1 pav. Sėkmingo duomenų importo langas. 40
3.2 pav. Duomenų saugyklos schema. 41
3.3 pav. Vaizdinys brandos atestato dalykų analizei. 43
3.4 pav. Gavybos struktūra brandos atestato dalykų analizei. 44
3.5 pav. Naive Bayes algoritmo sugeneruotas priklausomybių tinklas. 44
3.6 pav. Atributų savybių fragmentas. 45
3.7 pav. Vaizdinys pirmojo semestro analizei. 45
3.8 pav. Pirmojo semestro modulių įtaka būsenai. 46
3.9 pav. Pirmojo semestro modulių charakteristikų fragmentas. 46
3.10 pav. Vaizdinys būsenų prognozei. 47
3.11 pav. Bendras studentų pasiskirstymas klasteriuose. 48
3.12 pav. Prognozės rengimas Mining Model Prediction komponente. 48
3.13 pav. Būsenų prognozės fragmentas. 49

3.14 pav. Vaizdinys pirmojo ir antrojo semestro analizei.....49
3.15 pav. Pirmojo semestro diskrečios analizės fragmentas.....50
3.16 pav. Pirmojo semestro tolydinės analizės fragmentas.....50
3.17 pav. Valstybinių egzaminų rezultatų ir „1“ būsenos pasiskirstymas.....51
3.18 pav. Priklausomybių tinklas būsenos įtakai.....52
3.19 pav. Atestato dalykų ir visos imties pasiskirstymas.....53
3.20 pav. Įėjimų su galimomis reikšmėmis įtaka būsenai.....54
4.1 pav. Būsenų prognozių problematika.....55
4.2 pav. prognozių pasiskirstymas pagal Naive Bayes algoritmą 56
4.3 pav. prognozių pasiskirstymas pagal grupavimo algoritmą 56
4.4 pav. prognozių pasiskirstymas taikant sprendimų medžio algoritmą 57
4.5 pav. prognozių pasiskirstymas taikant neuroninių tinklų algoritmą 57
4.6 pav. Algoritmų efektyvumo prognozavimui palyginimas.....58

Lentelių turinys

1 lentelė. Duomenų gavybos algoritmų taikymo galimybių palyginimas 27

ĮVADAS

Bėgant laikui, įvairios įmonės, organizacijos, ar valstybinės institucijos sukaupia vis daugiau ir daugiau informacijos duomenų saugyklose. Per pastaruosius dešimtmečius nuolat krito ir tebemažėja informacijos kaupiklių kainos, taigi, saugoti didelės apimties duomenis gali vis daugiau ir daugiau įvairių organizacijų. Prieinama prie tokios situacijos, kada yra turima daug duomenų, tačiau juose sudėtinga įžvelgti naudingą informaciją; iš jų mažai naudos. Tada ir pasitelkiama duomenų gavyba.

Duomenų gavyba (angl. Data Mining) – tai surinktų duomenų analizė, norint atrasti užslėptus įvairius ryšius ir dėsningumus, kurie yra naudingi duomenų turėtojui [1]. Remiamasi statistika, priklausomybių teorija, sprendimų medžiais, neuroninių tinklų žiniomis.

Šio darbo tikslas naudojantis automatizuotomis duomenų gavybos priemonėmis atlikti universiteto studentų įvertinimų analizę. Remiantis analizės rezultatais, prognozuoti būsimų studentų mokymosi universitete galimybes. Darbe apžvelgiama intelektualios duomenų analizės programinė įranga, jos taikymo sritys. Detaliai analizuojama Microsoft SQL Server 2008 Developer Editon programinė įranga duomenų gavybai. Būtent jos pagrindu plėtojamas šis projektas. Atlikus analizę nustatyta, kurie duomenų gavybos algoritmai yra tinkamiausi pasirinktai dalykinei sričiai tirti.

Sistemos reikalavimų analizės ir projektavimo etapus atspindi projektinis skyrius. Jame pateikti svarbiausieji aspektai, pasitelkiant UML diagramas.

Duomenims saugoti suprojektuota duomenų saugykla, su išvalytais informatikos fakulteto dviejų studentų kartų stojimo bei mokymosi universitete dalykais ir pažymiais. Studentams priskirtas atributas, charakterizuojantis universiteto baigimą – iškritimą.

Akademinių universiteto duomenų analizei pasirinkta Bayeso, sprendimų medžių, klasterizacijos bei neuroninių tinklų algoritmai. Išsamiai aprašomas jų taikymas bei gauti rezultatai. Analizuoti aspektai: stojimo pažymių įtaka, pirmojo semestro pažymių įtaka galimybei sėkmingai užbaigti studijas.

Atliktas tyrimas, nustatyti, kuris iš panaudotų duomenų gavybos keturių algoritmų yra tinkamiausias prognozuoti studentų galimybę baigti studijas, remiantis ankstesnės kartos modeliu bei stojimo pažymiais.

1. GAVYBOS CIKLO, ĮRANKIŲ, METODIKŲ ANALIZĖ

1.1 Taikymo sritys

Duomenų gavyba, kaip inžinerinė informatikos mokslo šaka taikoma daugelyje sričių, tiek globaliniams tiek lokaliems uždaviniams spręsti. Pagrindinės intelektualios duomenų gavybos taikymo sritys:

1.Marketingas. Turint ilgo laikotarpio pirkėjų duomenis, galima padidinti pardavimus, žinant pirkėjų įpročius, poreikius ir panašiai. Paprastas pavyzdys: pirkėjas žiniatinklio pagalba nuolat perka kompiuterinius žaidimus. Tokiam pirkėjui, sistema gali automatiškai parinkti reklaminius skydelius būtent su naujausiais pasirodžiusiais žaidimais. Visada didesnė tikimybė, kad pirkėjas pirks tokio tipo prekių, kokių yra daugiausiai pirkęs.

2. Apgaulių aptikimas (angl. fraud detection). Pasitelkus duomenų gavybą galima aptikti, kurie sandėriai yra potencialiai apgaulingi. Taikoma draudimo kompanijose, elektroninėje bankininkystėje.

3. Operatyviniai tyrimai. Galima pastebėti įvairias anomalijas įvairiose sistemose, kaupiančiose įvairius gyventojų registrus, klasifikatorius ir kitus duomenis. Taip pat naudojama antiteroristiniams tikslams.

4. Sportas. Įvairiose sporto šakose fiksuojama daugybė įvairiausių parametrų. Pvz. krepšinyje kiekvienam žaidėjui stebimi pataikymo procentai, pražangos, išprovokuotos pražangos, žaidimo trukmės, blokai, perdavimai, atkovojimai ir panašiai. Komandų strategai pasitelkiant duomenų analizę gali optimaliau paskirstyti žaidėjų resursus, pergalei pasiekti.

5. Įvykių prognozavimas. Analizuojami tam tikrame laike sukaupti faktai, ir remiantis jais prognozuojama tam tikro dar neesamo laikotarpio įvykis. Taikoma tiek ekonomikoje, tiek fizikoje, bei kitose mokslo srityse.

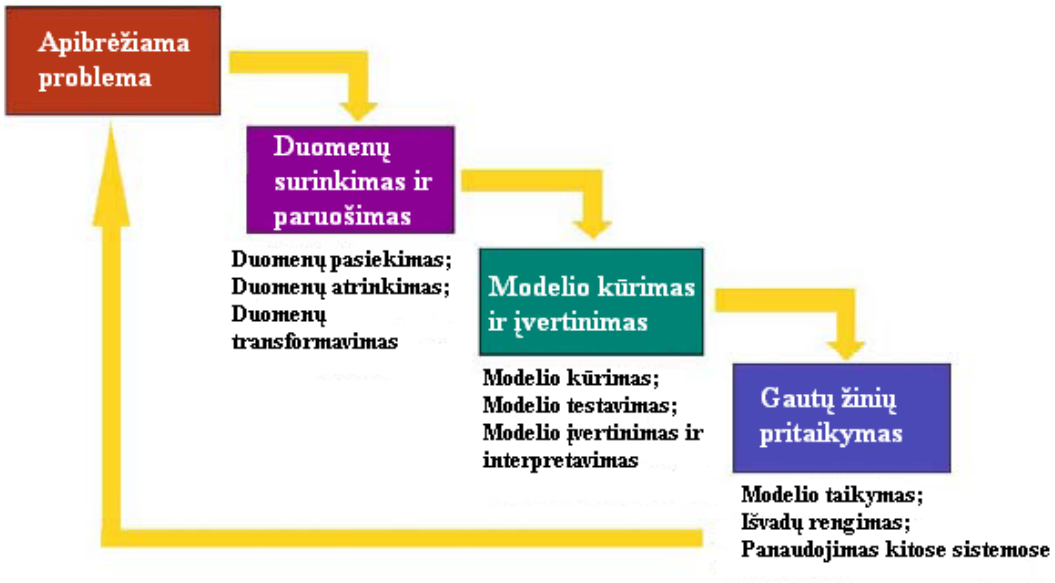
Egzistuoja daugelis kitų, pačių įvairiausių duomenų gavybos taikymo sričių. Kadangi pasaulyje vis daugėja saugomų duomenų apimtis, duomenų gavybos galimybės ir taikymai nuolat plečiasi, kartu evoliucionuojant ir atitinkamai programinei įrangai. Šiame darbe aptariama intelektuali universiteto akademinių duomenų analizė.

1.2 Duomenų gavybos ciklas

Duomenų gavybos ciklą galime suskirstyti į keturis pagrindinius žingsnius[2]: dalykinės srities problemos apibrėžimą, analizės duomenų surinkimą ir paruošimą, modelio pritaikymą problemos analizei, bei tyrimo rezultatų pritaikymą problemai išspręsti (1.1 pav.).

Kiekvienas išvardintų žingsnių turi smulkesnius etapus, kurie yra reikalingi, norint sėkmingai įgyvendinti intelektualios duomenų analizės uždavinį.

Šiame skyriuje bus apžvelgti duomenų gavybos pagrindiniai etapai, taikomajam duomenų gavybos uždaviniui pasiekti[3]. Šiais žingsniais paremtas ir magistrinio darbo projektas. Pradedama pradinių duomenų surinkimu, ir pabaigiama konkrečių naudingų rezultatų išgavimu.



1.1 pav. Pagrindiniai duomenų gavybos žingsniai [2]

1.2.1. Duomenų surinkimas

Reikalingi tyrimui, to paties tipo duomenys gali būti išsidėstę keletoje įvairų mažai susijusių sistemų. Norint juos ištirti, reikia pirmiausia viską surinkti į vieną vietą. Surinkimo vieta gali būti konkreti, dedikuota duomenų bazė. Pavyzdžiui, reikia ištirti žiniatinklio puslapio peržiūrinėjimo apkrovimus, kuomet naudojama daug serverių. Tuomet pradžioje reikia persikelti iš visų jų lankomumo statistiką į vieną vietą.

1.2.2. Duomenų transformacija ir valymas

Dažniausiai iš pradinių duomenų negalima daryti išvadų, nes jie nėra tinkamai paruošti. Tam tikri duomenys gali būti nenaudingi, ir įtakoti rezultatų klaidingumą. Taigi svarbu pasiruošti reprezentatyvią tiriamųjų duomenų aibę. Pagrindiniai paruošiamieji darbai:

Duomenų tipo keitimas. Paprasta operacija, dažnai reikalinga efektyviau realizuoti vieną ar kitą duomenų gavybos algoritmą. Pavyzdžiui, loginė reikšmė keičiama sveikaisiais skaičiais, arba atvirkščiai.

Grupavimas. Naudojamas, kai reikia sumažinti tam tikrų požymių skaičių. Tarkime vienas iš kintamųjų yra konkrečios profesijos ir mums pakanka abstraktesnės aibės su mažiau narių – profesinės krypties (humanitariniai mokslai, inžinerija, menai ir t.t.). tokiu būdu modelis tampa mažiau kompleksiškas ir jį lengviau interpretuoti.

Agregacija. Naudojama, kuomet reikalingas išvestinis tiriamasis požymis iš keleto esančių, kurie tiesiogiai mažai naudingi. Pavyzdžiui, turima abonentų telefoninių pokalbių duomenys. Yra reikalinga suskirstyti vartotojus pagal naudojimosi intensyvumą. Tuomet galima surasti, kiek vidutiniškai kuris skambino, ir kokia vidutinė skambučio trukmė.

Trūkstamų reikšmių apdorojimas. Dažnai nutinka, kad dėl vienokių ar kitokių priežasčių nebūna reikalingų reikšmių įrašuose. Yra keletas būdų, kaip tokias reikšmes atstatyti. Vienas iš jų – įrašyti dažniausiai pasitaikančią reikšmę. Kitas panašus būdas – įrašyti atitinkamo atributo vidutinę reikšmę. Tačiau dažnai tokie pakeitimai gali iškreipti galutinius rezultatus. Vienas iš sudėtingesnių būdų – pasitelkti atskirą duomenų gavybos uždavinį tokioms reikšmėms gauti. O paprasčiausiu atveju – įrašus su trūkstamais duomenimis tiesiog pašalinti iš tyrimo.

Anomalijų šalinimas. Ekstremalios reikšmės visada iškreipia dėsnį, tinkantį dažniausiai pasitaikantiems atvejams. Taigi, eilutes su atitinkamo atributo nestandartiškai didelėmis ir mažomis reikšmėmis galima arba pašalinti, arba priskirti atskirai kategorijai.

1.2.3. Modelio taikymas

Vienas svarbiausių etapų. Pasirinkus netinkamą modelį (duomenų gavybos algoritmą),vėliau, kituose etapuose galime sugaišti daug laiko tačiau gauti klaidingas išvadas. Modelio pasirinkimas priklauso nuo ieškomo dėsnio, dalykinės srities bei kitų veiksnių. Dažnai tinka keletas algoritmų. Tačiau būna, kad kuris nors vienas duoda geresnį tikslumą. Norint

įsitikinti, kas yra tiksliausia, geriausiai taikyti keletą algoritmų. Modelio pritaikymas nereikalauja tiek laiko, kaip duomenų paruošimas.

1.2.4. Modelio įvertinimas

Turint apmokytą algoritmą – suradus pradiniuose duomenyse dėsningumą, reikia įsitikinti, kiek tas dėsningumas yra efektyvus. Įvertinimo reikšmė dar labiau sustiprėja, jei taikoma keletas modelių ir reikia pasirinkti tiksliausią. Įvertinti atrastų dėsningumų prasmę gali padėti išmanantys dalykinę sritį, bei testiniai duomenų rinkiniai. Atliekant testą su žinomais duomenimis, gaunamas įvertinimas su atitinkamomis prognozių tikimybėmis, ir šie rezultatai sulyginami su testinių duomenų iš anksto žinomu rezultatu. Tuomet vizualizavus šiuos rezultatus nesunkiai galime pasakyti, ar pasirinktas modelis naudingas, ar reikia grįžti į ankstesnį etapą keisti arba algoritmą, arba kažkuriuos tyrimo parametrus.

1.2.5. Rezultatų prognozavimas ir paskelbimas

Kartais duomenų gavybos ciklas baigiasi atrasto dėsningumo ataskaitos parengimu. Tai gali būti vienas iš duomenų gavybos tikslų. Tačiau gerokai dažniau tie dėsningumai pasitarnauja prognozavimui. Tam reikalinga naujų duomenų aibė ir apmokytas modelis. Šiuo atveju ataskaitoje paskelbiama prognozė, lūkesčiai, prie naujų sąlygų – apmokytam modeliui pateiktų duomenų.

Sukurtasis modelis taip pat gali būti integruojamas į verslo valdymo ar kitokias sistemas, pastoviam naudojimui.

1.2.6. Modelio atnaujinimas

Atsirandant vis naujiems duomenų rinkiniams, gali pakisti ir dėsningumai. Kartais pridėjus naujų duomenų dėsnis išliks stabilus, o kartais, (ypač aktualu verslo aplinkose) – smarkiai pasikeis. Todėl svarbu nuolat stebėti padėtį ir periodiškai kartoti duomenų gavybą. Taigi duomenų gavybos procesas yra iteracinis

1.3. Esami įrankiai

Skyriuje bus trumpai apžvelgti šiuo metu rinkoje esantys, lyderiaujantys duomenų gavybos įrankiai ir jų gamintojai [2]. Tai padės geriau suvokti, kurioje vietoje, lyginant su

analogais, yra Microsoft kompanijos duomenų gavybos priemonės, kokios jų galimybės. Norint iliustruoti kuo platesnį spektrą programinės įrangos, pateikiami komerciniai rinkos lyderiai, bei populiaresni nemokami, laisvai platinami atvirojo kodo įrankiai, skirti intelektualiai duomenų analizei atlikti.

1.3.1. SAS

SAS kompanija laikoma viena didžiausių duomenų gavybos įrankių gamybos ir tobulinimo srityje. SAS įrankiuose gausu įvairių statistinių funkcijų duomenų analizei įvairiais aspektais. Naujausias produktas SAS Enterprise BI Server. Sukurta atskiri moduliai tiek mažoms tiek didelėms įmonėms, priklausomai nuo poreikių ir galimybių. [4]

1.3.2. SPSS

SPSS kompanija duomenų intelektualiai analizei siūlo Clementine įrankį. Naujausios Clementine 12 versijos duomenų gavybos algoritmus galima naudoti kartu su IBM DB2, Oracle, bei Microsoft SQL Server duomenų bazių valdymo sistemomis per Clementine Server [5]. Originalus būdas verslo efektyvumui padidinti, naudojant pagrindinių gamintojų duomenų bazių valdymo sistemas.

Clementine server yra trys pagrindiniai algoritmų moduliai – asociacijų, klasifikacijos ir segmentacijos.

1.3.3. IBM

2006 m. IBM nutraukė Intelligent Miner gamybą ir palaikymą DB2 duomenų bazių valdymo sistemai. Dabar IBM duomenų gavyba integruota į InfoSphere Warehouse. Gamintojo nurodoma paskirtis - melagingų operacijų atlikinėjimo aptikimai, vartotojų kategorizavimas, bei internetinių apsipirkinėjimų analizė.

2008 m. IBM įsigijo Cognos bendrovę, vieną iš buvusių duomenų gavybos įrankių lyderių. Pilnai duomenų analizei IBM siūlo naudotis InfoSphere Warehouse ir Cognos 8 BI [6].

1.3.4. Microsoft SQL Server

Microsoft SQL Server 2008 yra pirmaujantis serveris tarp OLAP srityje. Microsoft SQL Server 2008 Developer Edition Business Intelligence Developement Studio siūlo 9 duomenų gavybos algoritmus. Analizuoti galima tiek reliacines duomenų bazes, tiek duomenų kubus. Algoritmai aprašyti atskiroje dalyje.

Duomenų grafiniam apipavidalinimui Microsoft kompanija paskutinės serverio versijos Reporting Servines komponentą papildė Dundas Data Visualisation komponentais [7].

1.3.5. Oracle Data Mining

Kompanijos Oracle duomenų bazių valdymo sistemos pakete 11g esantis komponentas Oracle Data Mining (ODM) turi 11 įvairių duomenų gavybos algoritmų. Skirstomi į prižiūrimus (dėsningumo požymį pasirenka duomenų išgavėjas) ir neprižiūrimus apsimokymo algoritmus[2].

Oracle Data Mining palaikomi algoritmai: Naivo-Bayeso, Sprendimų medžio, regresijų, keleto algoritmų kombinaciją – SVM, du klasterizacijos algoritmai, asociacijų, atributų prastinimo, anomalijų aptikimo, atributų „svorio“ radimo.

1.3.6. Weka

Naujosios Zelandijos mokslininkų duomenų gavybos įrankis. Weka yra nemokama, ir atvirojo kodo. Naujausia versija Weka 3.6, kaip ir ankstesnės yra sukurtos taikyti JAVA kalba paremtuose projektuose, įskiepų pavidalu, bei savarankiškai. [8]. Gali būti dirbama tiek komandiniame, tiek vaizdiniame režimuose. Weka turi klasterizacijos, klasifikacijos, asociacijų bei skirstymo pagal spėjamus atributus algoritmus.

1.3.7. Orange

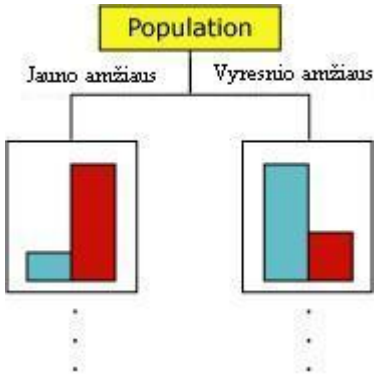
Nemokamas ir laisvai platinamas duomenų analizės paketas. Komponentai parengti C++ kalba, gali būti laisvai integruojami į kuriamus projektus. Turi 7 analizės ir prognozių modeliavimo algoritmus: sprendimų medžio, Naivo – Bayeso, k-„artimiausio kaimyno“, grupavimo pagal dažniausius atributus, logaritminės regresijos, grupavimo pagal nurodytas taisykles, taip pat iš keleto kombinuotą SVM algoritmą. Orange įrankis turi algoritmo įvertinimo metodus, nustatyti jų panaudojimo tinkamumui [10].

1.4. MS SQL Server duomenų gavybos algoritmai

Microsoft SQL Server Analysis Services yra 9 duomenų gavybos algoritmai. Yra galimybė kaip įskiepus naudoti ir kitus, trečiųjų šalių algoritmus, kurie yra pritaikyti MS SQL Server. Skyriuje analizuojami integruotieji algoritmai, pateikiant kiekvieno veikimo principą bei taikymo galimybes. Aptarta algoritmų taikymui skirta DMX užklausų kalba.

1.4.1. Sprendimų medžių algoritmas

Tai klasifikavimo ir regresijos algoritmas, tiek diskrečių, tiek tolydžių požymių spėjimui. Diskrečių dydžių atveju, algoritmas ieško ryšio tarp keleto atributų reikšmių ar būsenų ir priklausomai nuo dažniausiai pasitaikančių ryšių, juos skirsto į dvi šakas. Tarkime, turime įvairaus amžiaus tam tikros prekės pirkėjus. Mums reikia nuspręsti, kuris pirkėjas greičiausiai pirktų šią prekę, jei 8 iš 10 jaunuolių pirko analogišką prekę ir 2 iš 10 vyresnio amžiaus žmonių pirko taip pat (1.2. pav. Mėlynas stulpelis – amžius; raudonas stulpelis – nupirktos prekės). Algoritmas nustato, kad amžius yra tinkamas spėjimo požymis, prekių pirkimui. Sprendimų medyje spėjimas daromas remiantis išvadomis, apie esamų ryšių tendencijas. Giliau gali būti šakojama pagal kitus požymius[10].



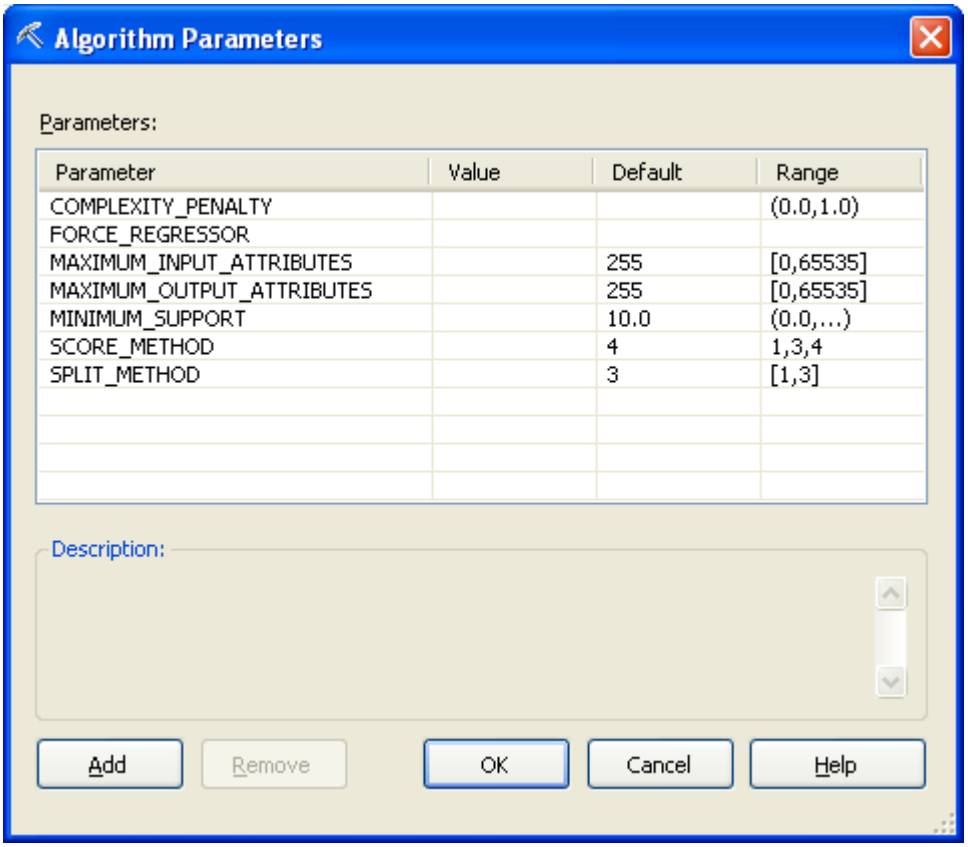
1.2 pav. Sprendimų medžio šakojimas

Tolydžių požymių grupavimui, algoritmas naudoja tiesinę regresiją, nustatyti, kur sprendimų medis išsišakos. Naudojamos dviejų tiesinių lygčių laužtė, kurios lūžio taškas ir yra medžio išsišakojimo vieta. Šiuo atveju kiekvienos šakos charakteringas požymis yra tiesinės regresijos lygtis.

Turint sprendimų medį – apmokytą algoritmą, jau galima paduoti tokius duomenis, iš kurių požymių galime priimti mus dominantį sprendimą.

Sprendimų medžio algoritmo vizualinė išraiška susideda iš dviejų dalių. Pirma - medžio, kuriame galima stebėti tiek populiacijos pasiskirstymą, tiek pasirinkto atributo konkrečios būsenos pasiskirstymą. Antra – priklausomybių įtakos. Čia stebima, kurie dydžiai labiausiai įtakoja analizuojamą atributą. Silpnesnius ryšius galima eliminuoti.

Taikant sprendimų medžio algoritmą, jį galima įtakoti koreguojant įvairius parametrus. 1.3 paveiksle atvaizduojama sprendimų medžio algoritmo parametrai.



1.3 pav. Sprendimų medžio algoritmo parametrų langas

COMPLEXITY_PENALTY – parametras, apibrėžiantis šakojimosi tikimybės apribojimus. Numatytoji reikšmė priklauso nuo įėjimo atributų skaičiaus. 0,5 kai atributų mažiau, ne 10; 0,9 kai atributų nuo 10 iki 99, ir 0,99 kai parametrų daugiau nei 100.

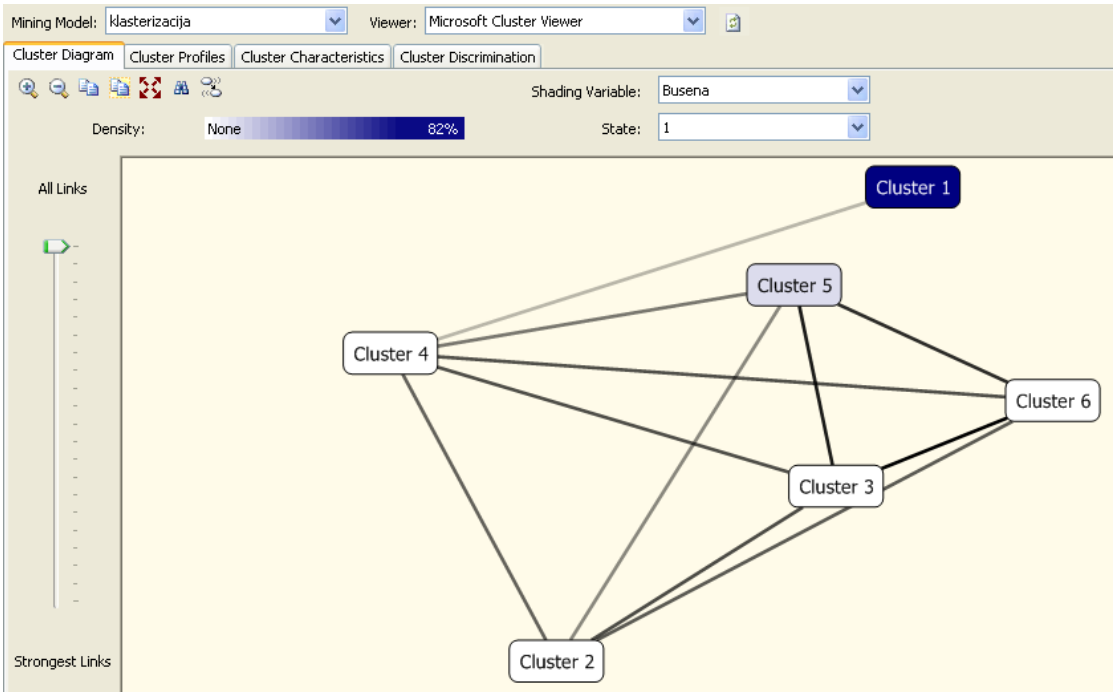
FORCE_REGRESSOR – galima priverstinai nurodyti, kuriuo atributu remiantis bus skaičiuojama medžio skilimo regresijų lygtys.

MAXIMUM_INPUT_ATTRIBUTES – galimas maksimalus skirtingų atributų skaičius, paduodamas sprendimų medžio kalkuliavimui. Esant daugiau atributų, automatiškai atrenkami charakteringiausi.
MAXIMUM_OUTPUT_ATTRIBUTES – galimas maksimalus skirtingų atributų skaičius, paduodamas sprendimų medžio generavimui.
MINIMUM_SUPPORT – apibrėžia minimalų atvejų skaičių, kiek jų turi būti kiekviename šakos mazge. Jei reikšmės nurodomos intervale [0..1], traktuojama kaip procentinė dalis (0=0%; 1=100%). Jei nurodomi skaičiai yra didesni – traktuojami kaip absoliutiniai atvejų skaičiai.
SCORE_METHOD – nurodoma, kokiu metodu apskaičiuojamas šakojimo taškas. Galimos tikimybinės reikšmės – 1 – šakojama pagal C. Shannon‘o entropinį pasiskirstymą; 3 – Bayeso K2 labiausio tikėtinumo; 4 – pagal Bayeso - Dirichleto labiausio tikėtinumo pasiskirstymą. Numatytoji reikšmė – 4.
SPLIT_METHOD – nurodo kaip šakoti sprendimų medį. Galimos reikšmės – 1 – binarinis; 2 – pilnas; 3 – kombinuotas šakojimas.

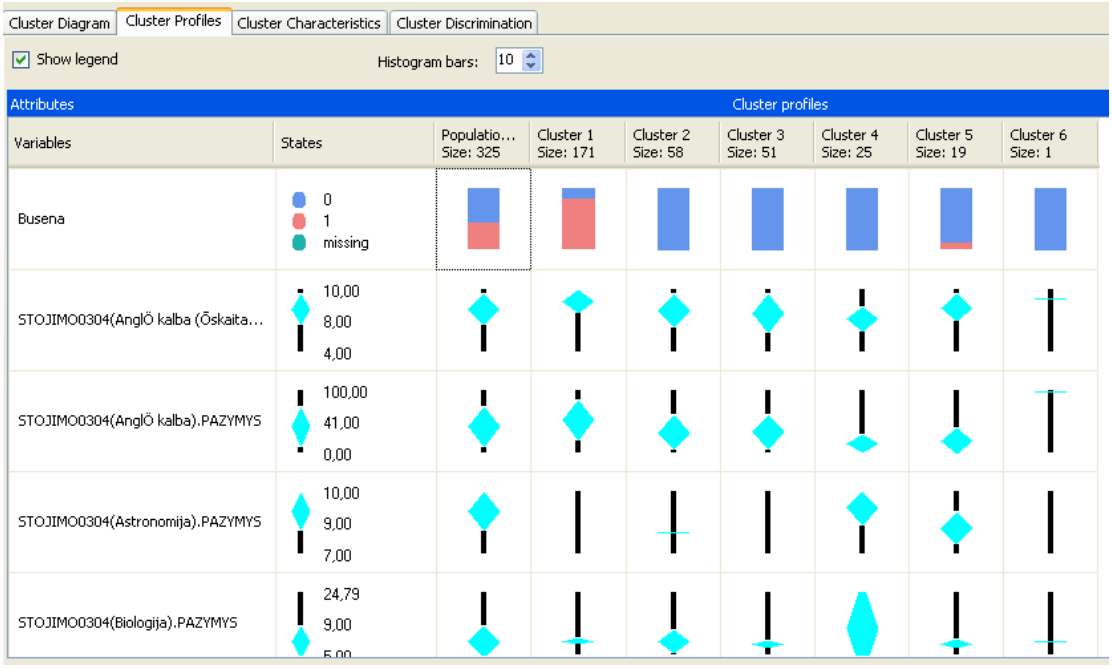
1.4.2. Grupavimo algoritmas

Grupavimo (angl. clustering) algoritmas duomenis skirsto į tam tikras kategorijas automatiškai atrandant skirstymo požymius. Tai yra iteracinis algoritmas, kiekvieną kartą vis tikslinant objekto priklausomumą vienai ar kitai grupei, kol pasiekiamas norimas tikslumas. Algoritmas naudingas duomenų tyrimui, ieškant įvairių anomalijų, bei spėjimams apie naujų duomenų aibes atlikti[11]. Taikant galima atrasti iš pirmo žvilgsnio nepastebimų dėsningumų, kurie gali būti naudingi.

Grupavimo algoritmo veikimas. Pradinių duomenų aibėje identifikuojami ryšiai ir generuojamos požymių grupės pagal tuos ryšius. Tuomet skaičiuojama, kaip tiksliai sukurtosios grupės atspindi išsidėsčiusių duomenų požymius. Yra du būdai paskaičiuoti priklausomumą kuriai nors grupei. Pirmasis – tikimybės priklausymo vienai ar kitai grupei skaičiavimas (kur tikimybė didžiausia – tai grupei objektas priklausys). Kitas būdas – artimiausio kelio iki atitinkamos grupės radimas.



1.4 pav. Būsenos „1” pasiskirstymas klasteriuose, remiantis stojimo pažymiais



1.5 pav. Klasterių savybių detalizavimo fragmentas

1.4 paveiksle grupavimo algoritme parinkta atvaizdavimui sėkmingai baigusiųjų studijas pasiskirstymas. Žinant kad didžioji jų dalis yra 1 klasteryje, galima nesunkiai pastebėti, kokiomis

savybėmis jie pasižymi. 1.5 paveiksle pateiktas fragmentas, kuriame matomi dalykų įvertinimų vidurkiai bei jų nuokrypiai, charakterizuojantys atskirus klasterius.

Grupavimo algoritmas turi šiuos parametrus:

CLUSTER_COUNT – apibrėžiamas klasterių skaičius. Numatytoji reikšmė – 10.

CLUSTER_SEED – atsitiktinis skaičius, įtakojantis atributų pasiskirstymą pirmoje apmokymo iteracijoje.

CLUSTERING_METHOD – vienas iš 4 metodų priskirti atributą vienam ar kitam klasteriui. Reikšmės – 1 – tikėtinumo maksimizavimo metodas su pakartotina peržiūra; 2 – tikėtinumo maksimizavimo iš karto; 3 – „K-means“ algoritmas, apskaičiuojant atstumus tarp klasterių su pakartotina peržiūra atitikimui klasteriui; 4 - „K-means“ algoritmas, apskaičiuojant atstumus tarp klasterių.

MAXIMUM_INPUT_ATTRIBUTES – galimas maksimalus skirtingų atributų skaičius, paduodamas skirstymui į klasterius. Esant daugiau atributų, automatiškai atrenkami charakteringiausi.

MAXIMUM_STATES – nurodo galimų atributo reikšmių skaičių. Numatytoji reikšmė yra 100.

MINIMUM_SUPPORT – apibrėžia minimalų atvejų skaičių, kiek jų turi būti kiekviename klasteryje.

MODELLING_CARDINALITY – nurodomas skaičius, kiek kartų tobulinti klasterius, kol pilnai ištreniruojamas modelis.

SAMPLE_SIZE – nurodo kiek atributų atvejų įtraukiama į modelio apmokymą, kai CLUSTERING_METHOD reikšmės parinktos su pakartotina peržiūra. Numatytoji reikšmė 50000.

STOPPING_TOLERANCE – dydis apibūdinantis, modelio treniravimo pabaigos sąlygą. Kitaip tariant užduodamas modelio tikslumas. Numatytoji reikšmė 10.

1.4.3. Naive Bayes algoritmas

Klasifikavimo algoritmas, naudojamas spėjimų modeliavimui. Kadangi skaičiavimų prasme vienas paprasčiausių – taikomas preliminariems įvertinimams, o vėliau spėjimus galima tikslinti pasitelkiant kitus algoritmus[12].

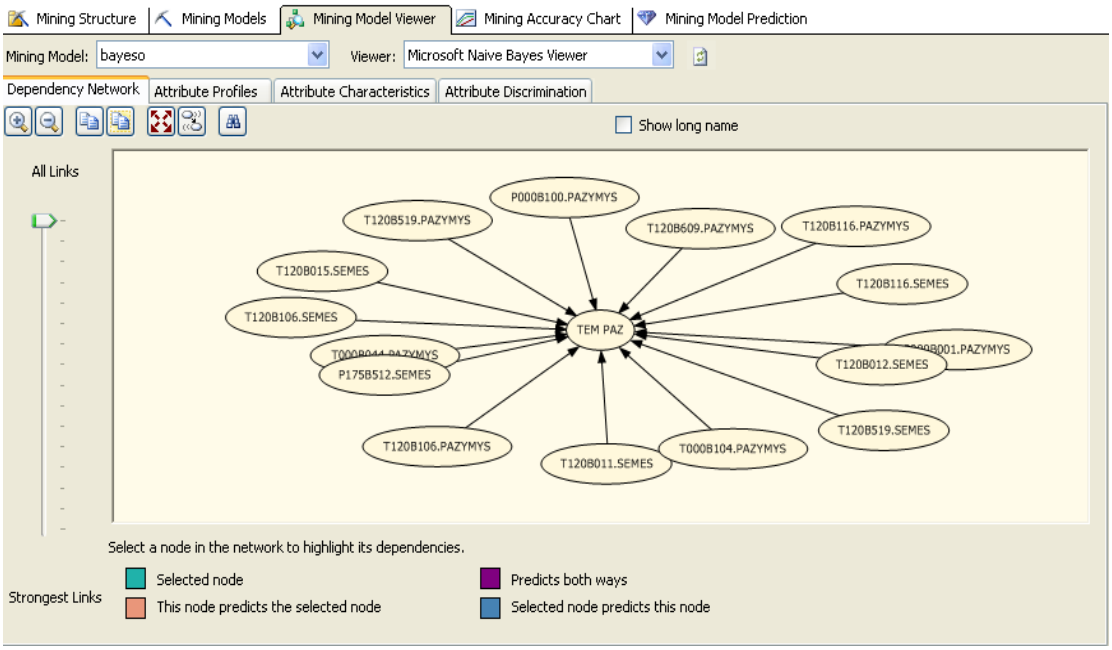
Veikimas pagrįstas skaičiavimu tikimybių , kokia gali būti būsena kiekviename požymių stulpelyje. Taikoma Bayes‘o tikimybinė išraiška:

$$P(H \mid E) = \frac{P(E \mid H) x P(H)}{P(E)}.$$

Čia P(H|E) – sąlyginė tikimybė, hipotezei H, turint aplinkybes (sąlygas) E.

Taikant Naive Bayes algoritmą traktuojama, kad visi įėjimo stulpelių atributai yra nepriklausomi. Šis algoritmas taikomas tik diskretiems arba diskretizuotiems požymiams tirti.

Microsoft SQL Server Business Intelligence Studio turi Microsoft Naive Bayes Viewer duomenų vizualizavimo komponentą, kuriame matoma, su kokiomis tikimybėmis yra pasiskirstę atitinkamų požymių kategorijos (pvz. amžiaus grupės ir jų tikimybės; moksliniai laipsniai ir jų tikimybės ir panašiai). Eliminavus mažiau įtakingų atributų ryšius, galima įvertinti „stipriųjų“ savybes atskiroje legendoje. Tačiau algoritmas vizualiai neapibūdina būtent kokia atributo reikšmė (būsena) yra įtakojama. 1.6 paveiksle iliustruotas pavyzdys, kokie universitete dėstomi moduliai labiausiai įtakoja baigiamojo darbo pažymį. Naive Bayes priklausomybių tinklas neatvaizduoja, kokiu būdu – kokiomis atributų reikšmėmis yra įtakojamas pasirinktas dydis. Įtakas detalizuoja atributų profilių komponentas.



1.6 pav. Nayve Bayes algoritmo priklausomybių įtakos atvaizdavimas

Nayve Bayes algoritmo parametrai:

MAXIMUM_INPUT_ATTRIBUTES – galimas maksimalus skirtingų atributų skaičius, paduodamas algoritmo apmokymui. Esant daugiau atributų, automatiškai atrenkami charakteringiausi. Numatytoji reikšmė 255.

MAXIMUM_OUTPUT_ATTRIBUTES – galimas maksimalus skirtingų atributų skaičius, ryšių atvaizdavimui ir būsenų charakterizavimui. Numatytoji reikšmė 255.
MAXIMUM_STATES – galimas maksimalus atributų įgyjamų reikšmių (būsenų) skaičius. Numatytoji reikšmė – 100.
MINIMUM_DEPENDECY_PROBABILITY – minimali tikimybė tarpusavio priklausomumo atributų, kurie įtraukiami į modelį. Tikimybei mažėjant daugėja mažiau susijusių atributų, kurie bus įtraukti.

1.4.4. Asociacijų algoritmas

Naudojamas tiek pirkinių krepšelio analizei, tiek pirkėjams rekomenduoti naują produkciją, žinant ką jie jau pirko. Algoritmo veikimas: peržiūrimi visi duomenų įrašai, atrenkant tokius, kurie turi panašių sąsajų. Jie grupuojami į tyrėjo apibrėžtą grupių skaičių. Kiekvienai grupei generuojama charakterizuojanti taisyklė[13]. Pagal tai galima aptikti dėsningumus naujiems atvejams – nuspėti, kokiai kategorijai jie priklausys. Asociacijų algoritmas taikomas tik diskretiems dydžiams.

Pavyzdžiui, žinant pirkėjo įpročius, jam galima pasiūlyti tokių produktų, kuriuos, labiausiai tikėtina, kad jis pirks. Tyrimo rezultatus pritaikyti galima per reklaminius skydelius, įvairias papildomos informacijos nuorodas ir panašiai.

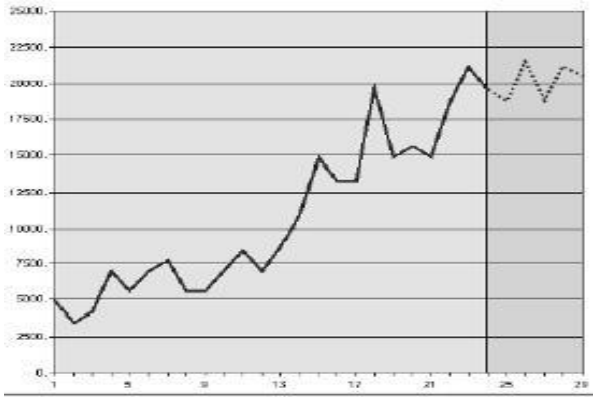
1.4.5. Sekų grupavimo algoritmas

Panašus į grupavimo algoritmą, tačiau jeigu grupavimo algoritme užteko surasti skirtingų požymių grupes, čia reikia išskirti panašių veiksmų grupes. Tai gali būti žiniatinklio puslapių navigacijos veiksmų seka, prekių krepšelio papildymo seka ir panašiai[14].

Veikimas. Naudojamas tikimybių maksimizavimas, identifikuoti grupėms ir joms būdingas dažniausiai pasitaikančias veiksmų sekas. Veiksmų seka suprantama, kaip aibė būsenų, sujungtų perėjimais iš vienos į kitą. Taikant šį algoritmą, vienu metu galima aprašyti tik vienokio tipo būsenas, tai reiškia, kad vienu metu negalima ištirti pvz. ir prekių krepšelio papildymo ir puslapių navigacijos sekų.

1.4.6. „Time Series“ algoritmas

Tai regresinis algoritmas, skirtas prognozuoti tolydžių dydžių kitimams. Tai gali būti produkto pardavimo tendencijos, kainų augimo prognozavimas ir panašiai. Iki šiol aptarti algoritmai prognozėms pasitarnaudavo, apmokytam modeliui pateikus porciją naujų duomenų. „Time Series“ algoritmas iš esamų duomenų pritaikius autoregresijos sprendimų medžio principą, gali išvesti tendencijas apie esamų duomenų būsimas reikšmes, kas ir yra prognozė [15].



1.7 pav. Tendencijų vizualizavimas su „Time Series“ algoritmu[9]

1.7 paveiksle pavaizduotas grafikas padalintas į dvi dalis. Kairiojoje pusėje esamų duomenų priklausomybės grafikas, dešiniojoje pusėje – prognozė, iš esamų duomenų.

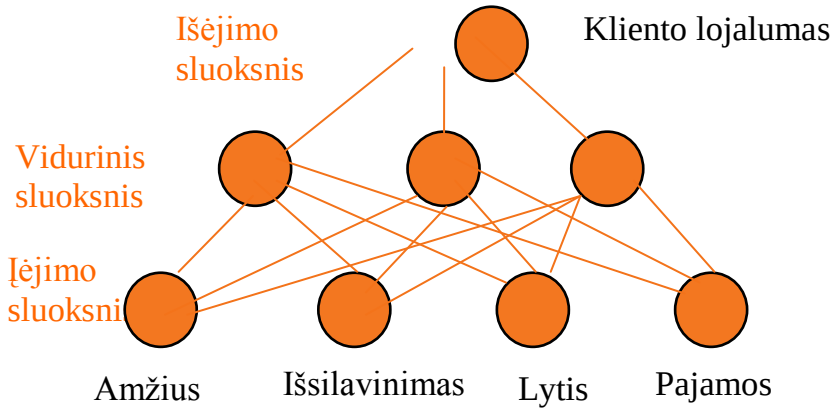
Microsoft Time Series algoritmas turi svarbią savybę: galima apmokyti algoritmą dvejomis poromis skirtingų stebėjimų, ir gauti išvestinę tendenciją. Pavyzdžiui, vieno produkto stebėjimo rezultatai gali pasitarnauti kito giminingo produkto pardavimų prognozei.

1.4.7. Neuroninių tinklų algoritmas

Naudojamas didelio kompleksiškumo duomenų analizei. Sudaromas keleto sluoksnių neuroninis tinklas, pasitelkiant klasifikavimo ir regresijos algoritmus. Panašiai kaip sprendimų medžio algoritme, kiekvienam įėjimo atributui paskaičiuojama tikimybė, kai žinomi visi galimi išėjimai. Sluoksniai yra susiję tik vienas su kitu, tačiau ne tarpusavyje. Tai reiškia kad įėjimo sluoksnyje mazgai nėra tarpusavyje surišti, o rišasi tik su viduriniu sluoksniu, o jei jo nėra – su išėjimo sluoksniu. Kiekvienas išėjimas yra netiesinė funkcija, susumuota iš įėjimo funkcijų. Išskiriami neuronų tipai: 1) įėjimo neuronai – gauna stebėjimo duomenų atskirus atributus; juos

gali paskirstyti vienam ar keletui vidurinio sluoksnio neuronams; 2) vidurinio sluoksnio neuronai – priima duomenis iš įėjimo neuronų ir juos paskirsto išėjimo neuronams; 3) išėjimo neuronai – generuoja spėjamas atributų reikšmes prie naujų duomenų, kurie taip pat ateina iš vidurinio ar pirmojo sluoksnio neuronų[16].

Algoritmo panaudojime išskiriamos tokios fazės: 1) duomenų išskleidimas ir įvertinimas; 2) apibrėžiama tinklo struktūra, priklausanti nuo esamų bei spėjamų atributų skaičiaus; 3) nustatoma kiekvieno požymio galimos būsenos 4) apsprendžiama, ar bus, ir jei bus, tai kiek naudojama vidurinio sluoksnio mazgų; 5) modeliui paduodama papildoma aibė duomenų, patikslinti parametrams, taip pat nustatyti klaidų lygį.



1.8 pav. Neuroninio tinklo principinė schema verslo klientų lojalumui įvertinti[19]

1.8 pav. pateikiamas pavyzdys, kaip formuojamas neuroninis tinklas iš 4 įėjimo sluoksnio atributų ir tarpinės veiklos logikos prognozuoja ekonomiškai naudingą atributą – kliento lojalumą. Taigi, šiuo lojalų klientą apsprendžia jo amžius, išsilavinimas, lytis bei pajamos. Viduriniojo sluoksnio ryšiai algoritmui identifikuojami automatiškai.

Neuroninio tinklo veikimą įtakojantys algoritmo parametrai:

HIDDEN_NODE_RATIO - daugiklis, apsprendžiantis kiek viduriniojo sluoksnio neuronų bus naudojama. Vidurinio sluoksnio neuronų skaičius S_{mid} apskaičiuojamas pagal šią formulę:

$$S_{mid} = HIDDEN\_NODE\_RATIO \times \sqrt{S_{in}} \times S_{out};$$

Kur S_{in} – įėjimo neuronų skaičius, S_{out} – išėjimo neuronų skaičius. Numatytoji reikšmė lygi 4.

HOLDOUT_PERCENTAGE – nurodoma procentinė dalis atvejų, nuo visų, kurie panaudojami algoritmo apmokymo pabaigos sąlygoms rasti. Numatytoji reikšmė – 30%.

HOLDOUT_SEED – pseudo atsitiktinis kintamasis, aukščiau aprašyto parametro duomenų atrinkimui. Numatytoji reikšmė lygi 0. Tai reiškia, kad pseudo atsitiktinumo kriterijus visada priklauso nuo gavybos struktūros pavadinimo.
MAXIMUM_INPUT_ATTRIBUTES – galimas maksimalus skirtingų atributų skaičius, paduodamas algoritmo apmokymui. Esant daugiau atributų, automatiškai atrenkami charakteringiausi. Numatytoji reikšmė 255.
MAXIMUM_OUTPUT_ATTRIBUTES – galimas maksimalus skirtingų atributų skaičius, neuroninio tinklo veiklos logikai organizuoti. Numatytoji reikšmė 255.
MAXIMUM_STATES – galimas maksimalus atributų įgyjamų reikšmių (būsenų) skaičius. Numatytoji reikšmė – 100.
SAMPLE_SIZE – nurodo kiek atributų atvejų įtraukiama į modelio apmokymą. Numatytoji reikšmė lygi 10000. Gali būti automatiškai parinkta kita reikšmė, priklausomai nuo visų galimų atvejų skaičiaus.

1.4.8. Logaritminės regresijos algoritmas

Neuroninių tinklų algoritmo atmaina, kai nenaudojamas vidurinysis neuronų sluoksnis. Taikant šį algoritmą tiesiogiai apsprendžiama, koks ryšys yra tarp įėjimo atributų stulpelių, kad būtų galima nuspėti išėjimo atributo reikšmes. Šiam algoritmui įėjimais gali būti tiek diskretūs tiek tolydūs atributai [17].

Logaritminės regresijos algoritmas taikomas tekstinės analizės, klasifikacijos bei įvertinimo uždaviniams išspręsti.

Algoritmo parametrai iš esmės tokie patys, kaip ir neuroninio tinklo. Nėra tik viduriniojo sluoksnio neuronų skaičių apsprendžiančio parametro.

1.4.9. Tiesinės regresijos algoritmas

Sprendimų medžio algoritmo atmaina, kai žinomų galimų būsenų skaičius yra mažesnis, arba lygus, nei numatoma prognozuojant naujų duomenų porciją. Šiuo atveju niekada negausime dviejų tiesių lygčių, o tik vieną, taigi sprendimų medis niekur neišsišakos. Paprasčiausias taikymo pavyzdys: turime ledų paklausos duomenis, priklausančius nuo temperatūros. Uždavinys nustatyti, kiek reikia užsakyti ledų, jei prognozuojama tam tikra temperatūra, kad nebūtų nei pertekliaus, nei stygiaus[18].

1.5. Duomenų gavybos algoritmų palyginimas

MS SQL Server Analysis Services palaikomi duomenų gavybos algoritmai palyginami 1 lentelėje [19]. Kaip matome, visi algoritmai turi arba išsamią duomenų analizės galimybę, ar bent jau dalinę realizaciją. Projekte, pagal pasirinktą dalykinę sritį, kaip tinkamiausius numatoma panaudoti grupavimo, sprendimų medžio, Naive Bayes bei neuroninių tinklų algoritmus.

1 lentelė. Duomenų gavybos algoritmų taikymo galimybių palyginimas

Algoritmas	Klasifikavimas	Segmentavimas	Įvertinimas	Asociavimas	Tendencijų prognozė	Teksto analizė	Išsami duomenų analizė
Asociacijų	●			✓		✓	✓
Grupavimo	●	✓	●	●		●	✓
Sprendimų medžio	✓		✓	✓		●	✓
Tiesinės regresijos			✓				●
Logaritminės regresijos	✓		✓			✓	●
Naive Bayes	✓			●		✓	✓
Neuroninių tinklų	✓		✓			✓	●
Sekų grupavimo	●	✓	●	●			✓
„Time series“					✓		✓
Legenda: ✓ - Pritaikymas palaikomas; ● - Pritaikymas dalinai palaikomas							

1.6. DMX kalbos struktūra

Duomenų gavybos išplėstinė kalba susideda iš dviejų blokų. Tai yra duomenų apibrėžimo dalis (DDL) ir manipuliacijos su duomenimis dalis (DML). Taikymo metu yra sukuriama duomenų gavybos struktūra. Tam nurodoma, kokios lentelės, jų atributai įtraukiami į gavybą. Taip pat nurodoma, kokio tipo yra atributai, nes juos skirtingai traktuojant gaunami skirtingi modelio rezultatai. Jau apibrėžus struktūrą, yra nurodomas duomenų gavybos modelis, ar keletas modelių. Tai yra duomenų gavybai skirtų algoritmų apibrėžimas. Šiame etape nurodomi kokie bus naudojami algoritmo parametrai duomenų apmokymui. Duomenų apibrėžimo etapas BI

Developement Studio aplinkoje vykdomas dizainerio pagalba. Duomenų apibrėžimo dalyje taip pat aprašomos ir sąlygos, kada struktūra ar modelis yra pašalinami.

Antrąjį kalbos bloką – manipuliavimą su duomenimis vykdome kai turime apmokytą modelį. Čia galima išskirti du tikslus – pačio modelio analizė – vartotojo požiūrių – tai yra algoritmo rezultatų interpretavimas; bei DMX kalbos panaudojimą nurodyto atributo prognozavimui, prie aprašytų sąlygų.

Bendra DMX struktūra duomenų gavybai yra tokia:

```
CREATE MINING STRUCTURE  --  (+ sąlygos) sukuriama struktūra
        CREATE MINING MODEL -- (+ sąlygos) sukuriamas modelis
                INSERT INTO -- (+ sąlygos) modelio apmokymas
                DELETE -- (+ sąlygos) naikinami apmokymo duomenys
        DROP MINING MODEL  -- (+ sąlygos) naikinamas modelis
---------------------------------
PREDICTION JOIN-- (+ sąlygos + gavybos funkcijos) – rezultatų prognozavimas
```

Rezultatų prognozavimui DMX kalboje yra specifinių funkcijų, kurios būdingos vienam ar kitam algoritmui [20]. Šiame darbe numatoma modelį kurti automatizuotai, o prognozavimui naudoti tiesiogines DMX užklausas.

1.7. Analitinės dalies išvados

1. Apžvelgtos pagrindinės intelektualiosios duomenų gavybos taikymo sritys. Iš tokio konteksto atsiskleidžia naujų taikymo sričių potencialas, tame tarpe ir šio darbo.
2. Suvoktas ir struktūrizuotas duomenų gavybos ciklas. Aprašytais etapais vadovaujamasi šiame projekte.
3. Apžvelgti esami duomenų gavybos įrankiai, bei jų galimybės. Dėmesys skirtas tiek komerciniams tiek laisvai platinamiems produktams.
4. Detaliai išanalizuoti Microsoft SQL Server 2008 Developer Edition integruoti duomenų gavybos algoritmai. Gilesnei analizei pateikiama algoritmo veikimą iliustruojantys pavyzdžiai. Pateikiamas algoritmų palyginimas, taikymo struktūra. Pasirinkti tinkamiausi gavybos algoritmai projektui įgyvendinti.

2. SISTEMOS PROJEKTAVIMAS

2.1. Tikslai

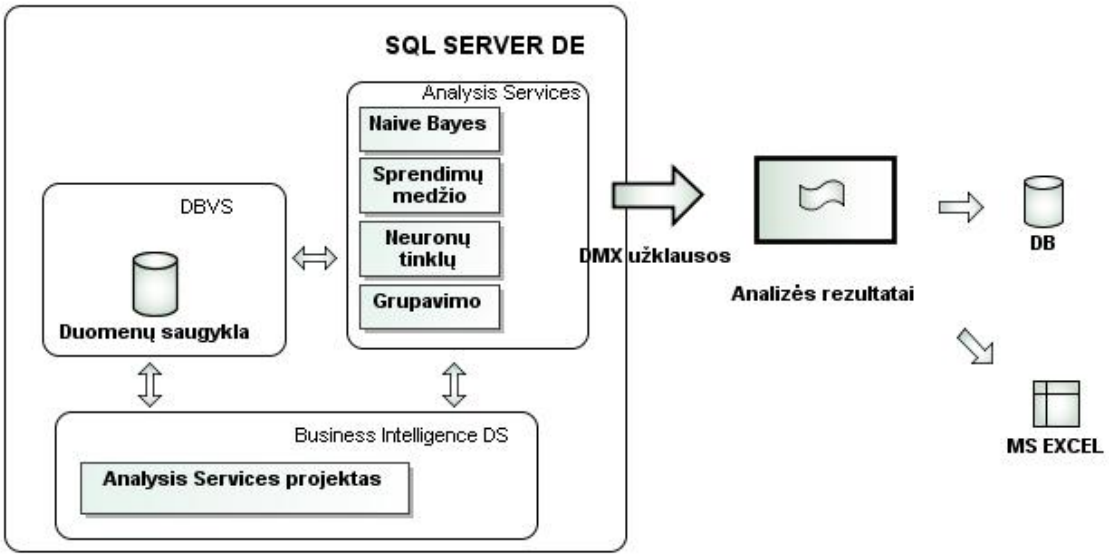
Skyriaus tikslas – atskleisti esminius sistemos projektavimo aspektus. Apžvelgiami funkciniai ir nefunkciniai reikalavimai, apribojimai sistemai, detalizuojami panaudos atvejai, veiklos aspektai pasitelkiant UML diagramas. Apžvelgiama kokie duomenys bus naudojami ir kokiais ryšiais tarpusavyje bus susiję. Iliustruojama pasitelkiant ER diagramą.

Praktine prasme, tikslas yra panaudojant MS SQL Server programinę įrangą, įgyvendinti galimybę analizuoti ir prognozuoti studentų mokymosi universitete galimybę, remiantis brandos atestato bei universitete gautais pažymiais.

2.2. Sistemos technologinis kontekstas

Projektuojamos duomenų analizės sistemos išdėstymas apsiriboja SQL Server Developer Edition naudojamose paslaugose (2.1 pav.). Duomenų bazių variklyje suprojektuojama duomenų saugykla su analizei reikalingais duomenimis. Tuomet Business Intelligence Developement Studio aplinkoje konstruojami duomenų gavybos algoritmų taikymai. Suprojektuotieji algoritmų taikymai registruojami Analysis Services dalyje, kur jau gali būti panaudoti gavybai. Modelio interpretavimas vykdomas per algoritmų rezultatų peržiūros priemones Business Intellignece DS aplinkoje, o prognozės įgyvendinamos per DMX užklausas. Jas galima vykdyti arba Analysis Services dalyje, arba Business Intellignece Developement Studio aplinkoje.

Rezultatai toliau apdorojami arba ataskaitų rengyklėmis arba skaičiuoklėmis, arba tiesiog išsaugomi tam skirtoje duomenų bazėje. Tai priklauso nuo potencialių sistemos vartotojų norų, galimybių bei reikalingumo. Šiame darbe rezultatai buvo apdorojami MS Excel programa, tiesiogiai juos įkėlus. Tačiau yra galimybė importuoti duomenis tiesiai iš duomenų bazės, nurodant duomenų šaltinį.



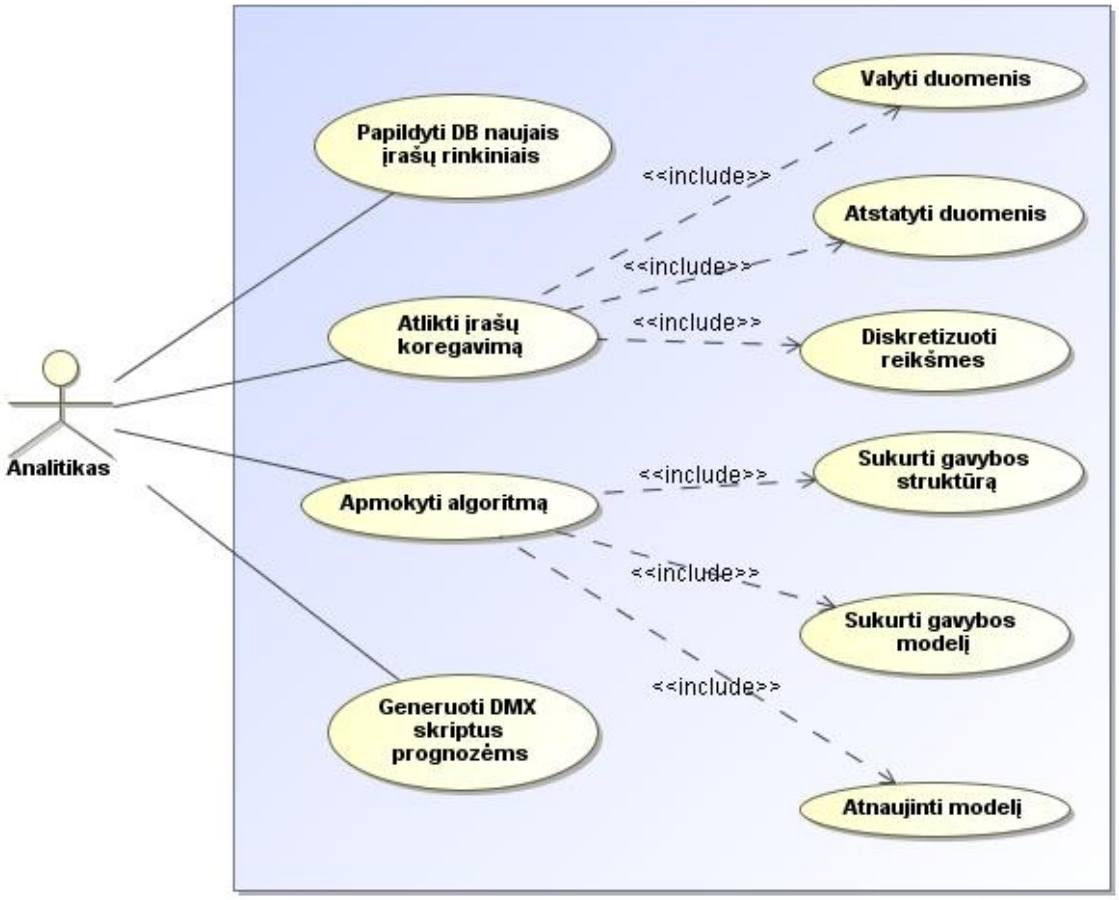
2.1 pav. Sistemos komponentų pasiskirstymas

2.3. Panaudojimo atvejai

Panaudojimo atvejuose išskiriami du aktorių - vartotojų tipai. Tai sistemos analitikas ir galutiniai vartotojai. Analitikas atsakingas už sistemos technologinius aspektus. Galutiniai vartotojai atsako už organizacines veiklas ir sprendimų priėmimą remiantis prognozių bei analizės rezultatais. Toliau trumpai aprašomi panaudojimo atvejai.

2.3.1 Sistemos administravimo panaudojimo atvejai

Sistemos analitikas atsakingas už duomenų bazės bei duomenų gavybos organizavimo techninius aspektus. Laikoma, kad duomenų saugykla jau suprojektuota, ir yra reikalinga tik papildyti jos turinį. Tai būtų administratoriaus darbas, ir toliau jo neaptarinėsime. Analitiko atliekamos veiklos iliustruotos 2.2 pav.



2.2 pav. Sistemos panaudojimo atvejų diagrama analitikui

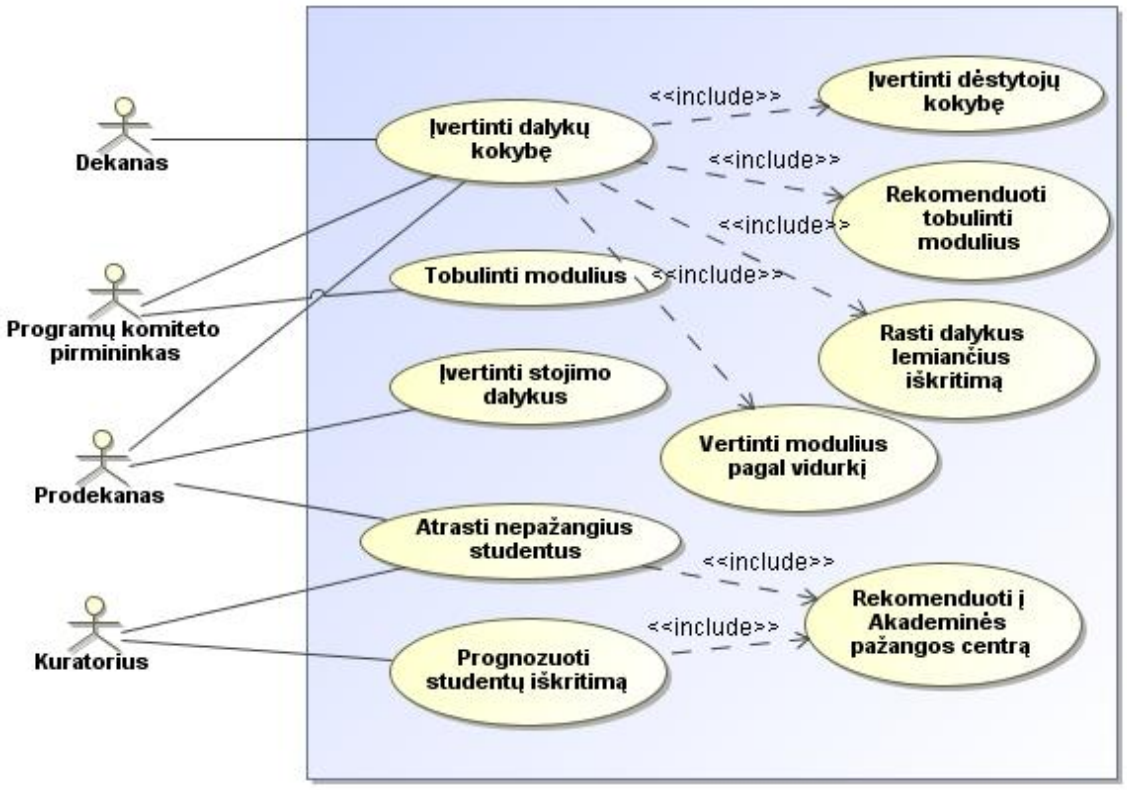
Norint atnaujinti duomenų gavybos modelį, ar struktūrą naujais duomenų rinkiniais reikalinga DB atnaujinti. Pridėtieji įrašai ne visada tiesiogiai gali būti panaudoti analizei. Tiek dėl atsiradusių anomalijų, tiek dėl netinkamos formos. Tam įrašai koreguojami – nenaudingi įrašai yra išmetami – išvalomi. Jei įrašas yra naudingas, tačiau turi sugadintą atributo reikšmę – atliekamas atstatymas, priklausomai nuo tipo ir aplinkybių. Jeigu atitinkamas atributas įgyja perteklinį skaičių galimų būsenų, galima tokias būsenas generalizuoti į stambesnius – diskretinius blokus.

Turint paruoštus duomenis, algoritmas treniruojamas pradiniais duomenimis. Tam reikalinga sukurti duomenų gavybos struktūrą, sukurti bent vieną modelį struktūroje. Esant galimybei atnaujinti DB naujais duomenų rinkiniais, gauti tikslesnei prognozei modelis atnaujinamas. Galiausiai, jau turint apmokytą – ištreniruotą pradiniais duomenimis modelį – generuojamas DMX skriptas, kurį paleidus gaunamos prognozuojamos studentų būsenos.

2.3.2 Sistemos taikymo panaudojimo atvejai

Pateikiama diagrama, su plačiu panaudojimo atvejų kontekstu, kaip galimas būdas, panaudoti kuriamą analitinę sistemą.

Galutiniais vartotojais laikomi aktoriai: fakulteto dekanas, programų komiteto primininkas, prodekanai bei grupių kuratoriai. Aktorių atliekamos veiklos atvaizduotos 2.3 paveiksle. Dekanas, programų komiteto primininkas bei prodekanas atlieka dalykų kokybės vertinimą. Galimybė įvertinti tiek pagal vidurkius, tiek pagal studentų iškritimą tam tikrame dalyke. Radus įtartinų rezultatų (didelis neišlaikiusių skaičius, prastas vidurkis) galima teikti rekomendacijas tobulinti modulį. Programų komiteto pirmininkas atsako už modulių tobulinimą, jei apibendrinus analizės rezultatus to reikia.



2.3 pav. Galutinių vartotojų panaudojimo atvejų diagrama

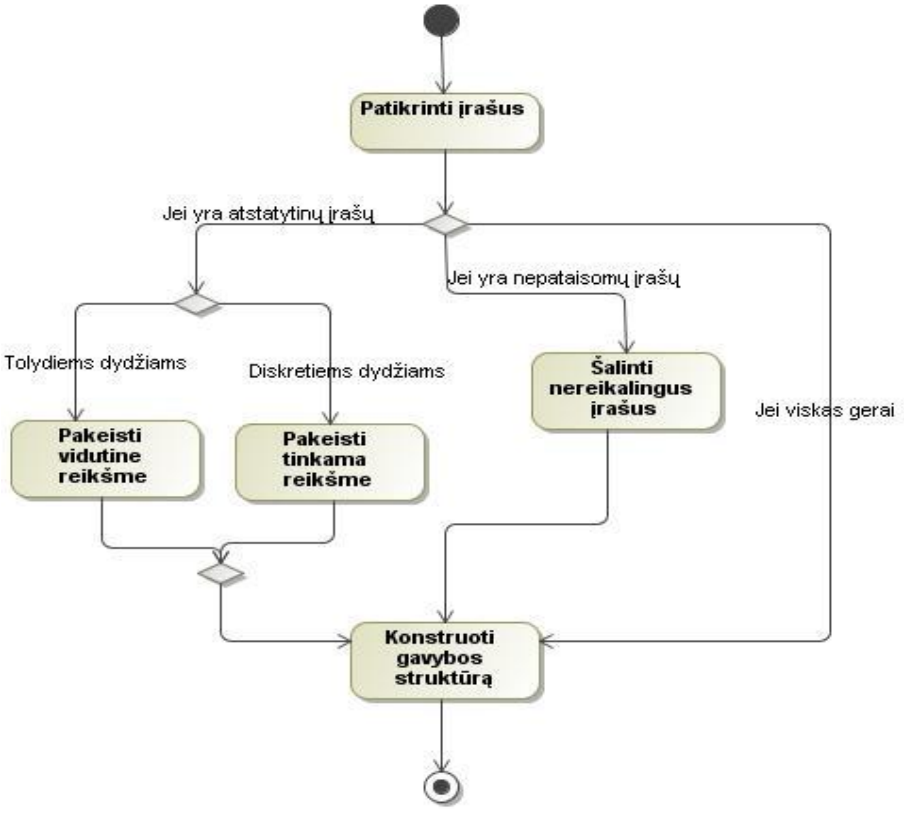
Prodekanas taip pat dalyvauja atrandant nepažangius studentus, kuriems, pasiekti geresnių rezultatų gali būti rekomenduota pasinaudoti akademinės pažangos centro paslaugomis, pagerinti žinių lygį. Tai glaudžiai susiję ir su stojimo balais, iš kurių taip pat galima numatyti, ar

bus reikalingas papildomas dėmesys studentams. Kuratorius be kita ko, atlieka prognozes naujai įstojusiems studentams, kokia jų galimybė baigti studijas. Tai papildoma profilaktinė – motyvacinė priemonė, mažiau pažangiems studentams.

2.4 Sistemos elgsenos aspektai

Skyriuje bus aptarta duomenų paruošimo veiksmų, sistemos funkcijų vykdymo, bei analizės serviso projektavimo veiklos schemos. Tam iliustruoti, pasitelkiama UML veiklos diagramos.

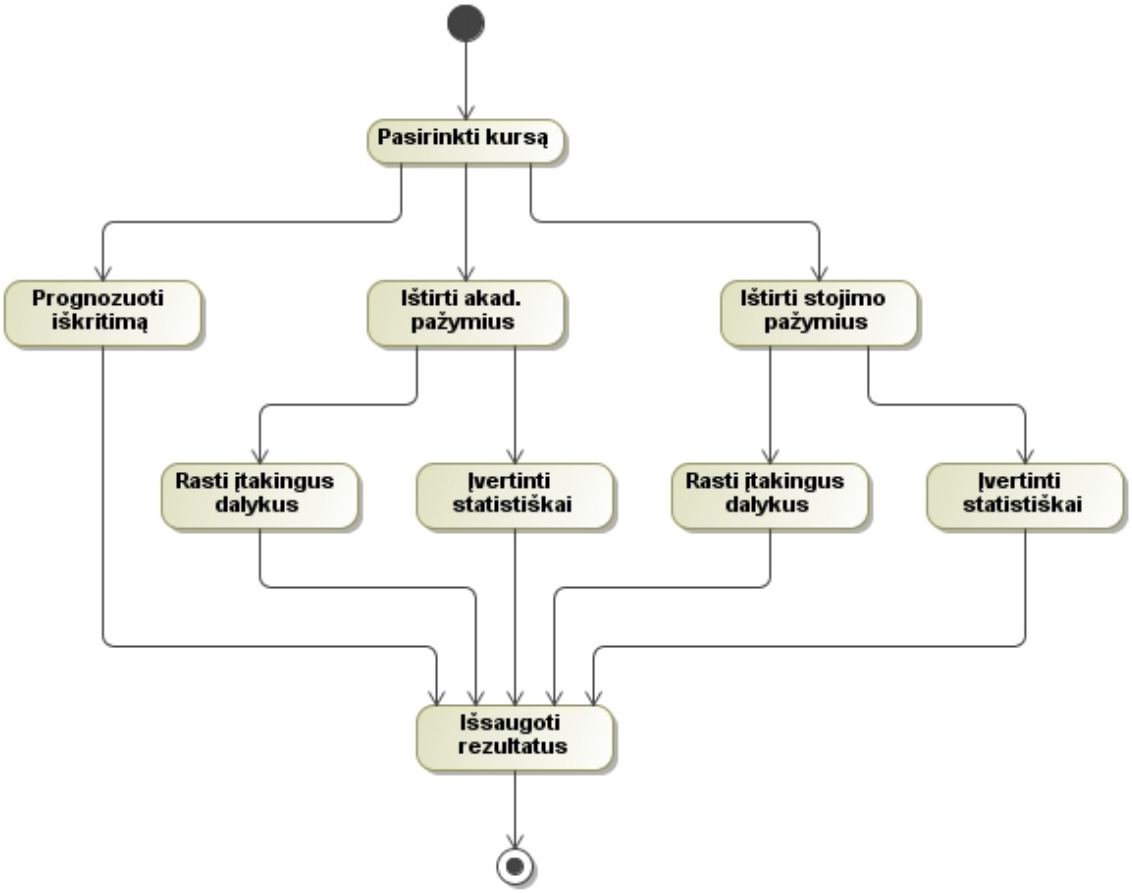
Toliau aptarta duomenų paruošimo veiksmų ir galimų alternatyvų schema. Pradiniai įrašai dėl įvairių priežasčių ne visuomet yra tinkami analizei. Juose esančios anomalijos gali neigiamai įtakoti tiek jų interpretaciją, tiek prognozavimo tikslumą. Duomenų paruošimo veiksmų seka pateikta 2.4. paveiksle.



2.4 pav. Duomenų paruošimo veiklos diagrama

Paprasčiausiu atveju, patikrinus įrašus, ir jiems esant tinkamiems jau galima ruošti duomenų gavybos modelį, arba modelių struktūrą. Esant žalingiems įrašams – juos galima paprasčiausiai išmesti. Turint nedidelę imtį, toks sprendimas yra pakankamai jautrus galutiniams

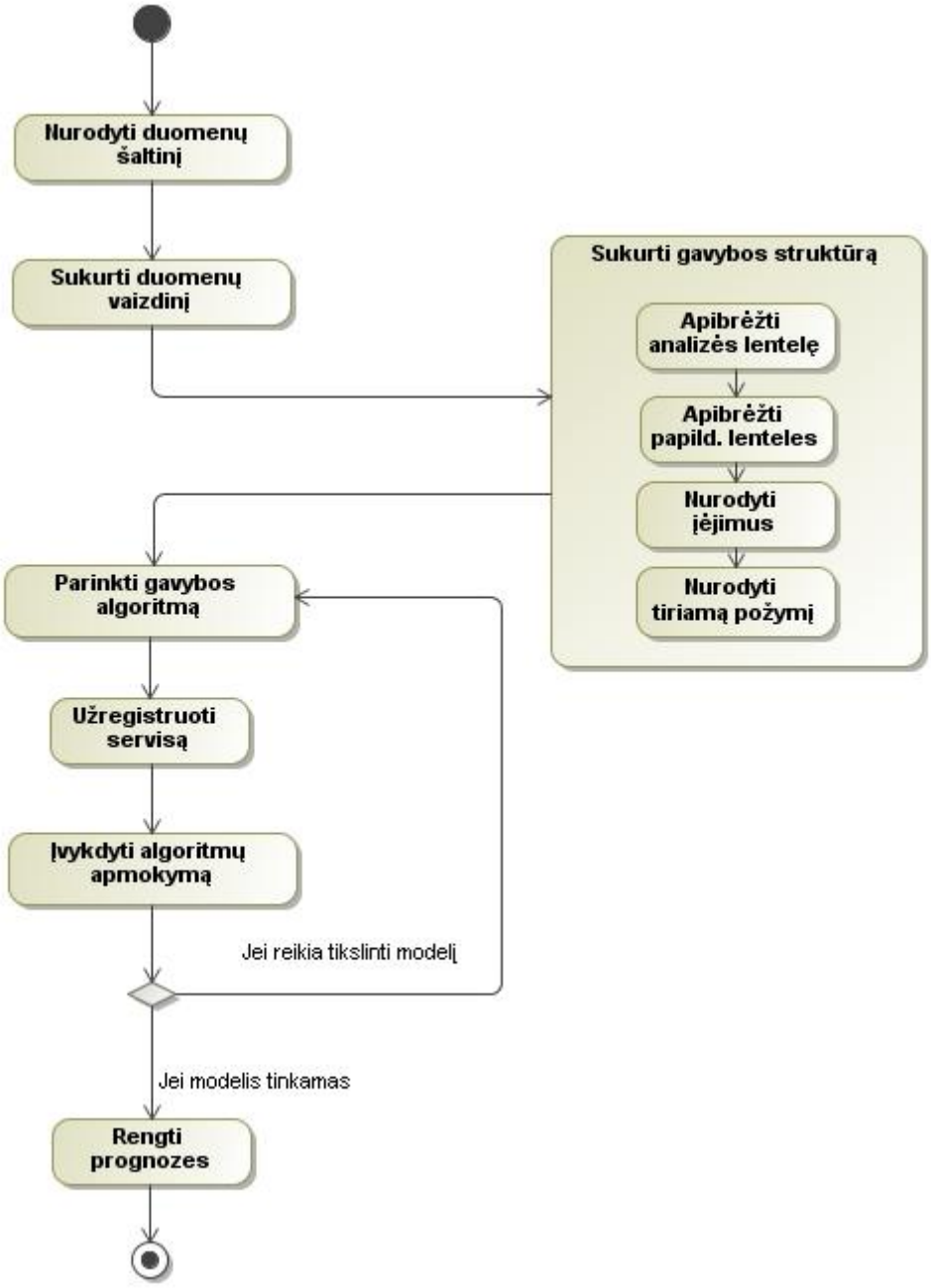
rezultatams. Tuomet, priklausomai nuo tikrinamo atributo tipo – ar diskretus – ar tolydus – galima įrašo lauką rekonstruoti arba vidutine, arba dažniausia reikšme. Priklausomai nuo lauko pobūdžio, gali būti pritaikomas ir kitoks atstatymo būdas. Duomenų paruošimas įgyvendinamas SQL kalba.



2.5 pav. Duomenų gavybos taikymų veiklos diagrama

Sistemos apibendrintos taikymo galimybės pateikiamos 2.5 paveiksle. Iškritimo prognozėms gauti, apmokytam pradiniais duomenimis gavybos modeliui įvykdoma DMX užklausa, kurios rezultatas – reliacinė lentelė, su galimybe jos duomenis išsaugoti pasirinktoje DB arba tiesiog eksportuoti į norimą aplinką.

Tiek studijų modulių, tiek stojimo pažymių įtaką iškrentamumui galima įvertinti keletu aspektų: skaičiuojant standartines statistikas, bei atrandant pačius jautriausius dalykus. Šakų vykdymą pasirenka vartotojas, eiliškumas ir vykdymo būtinumas nėra griežtai apibrėžtas.



2.6 pav. Duomenų analizės serviso rengimo veiklos diagrama

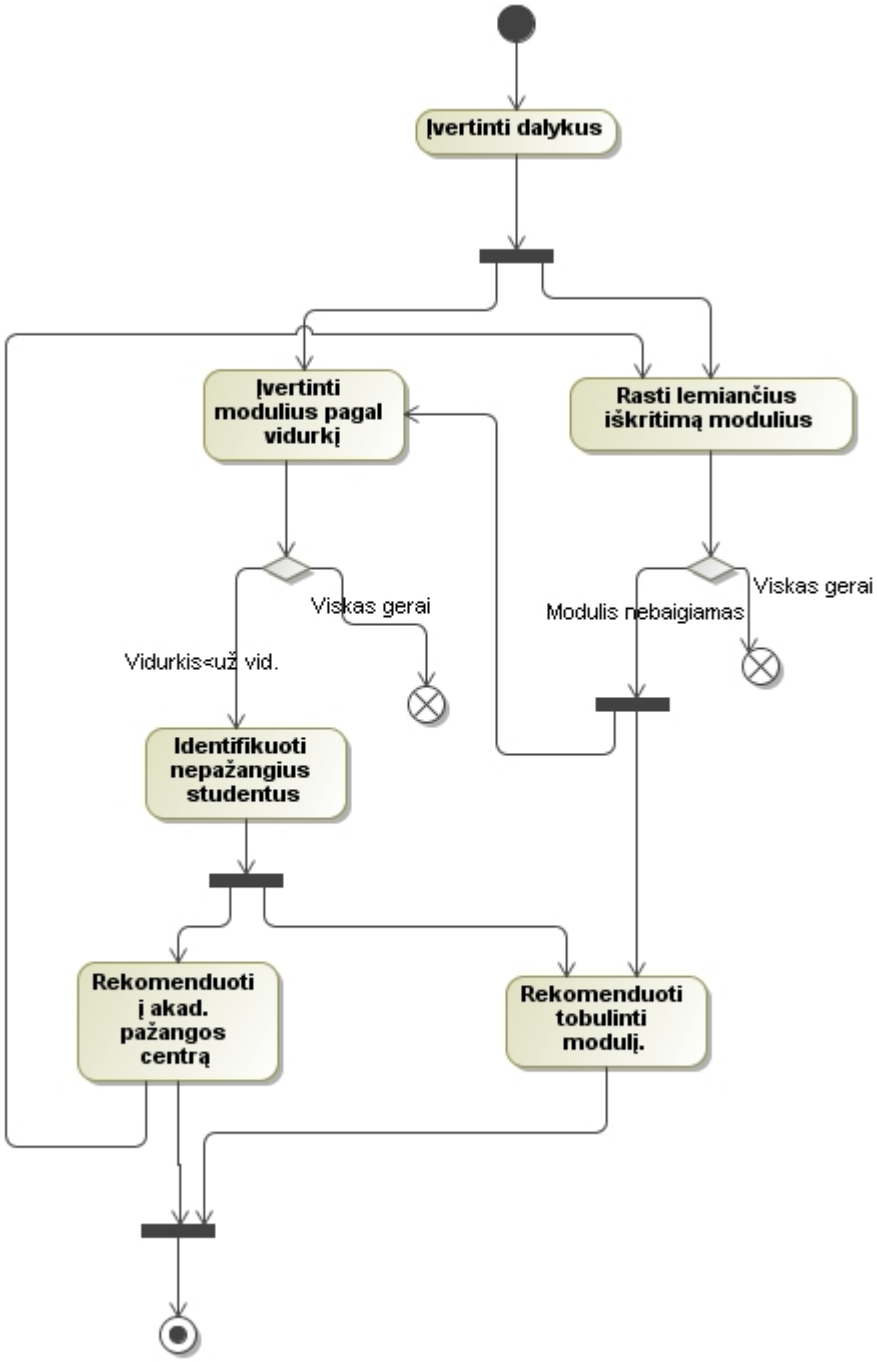
2.6 pav. pateikiama veiksmų schema, charakterizuojanti analizės serviso parengimo etapus. Veiksmai atliekami BI Developement Studio. Nurodžius duomenų šaltinį sukuriamas tinkamas gavybai duomenų vaizdinys. Tuomet kuriama gavybos struktūra. Šis veiksmas skaidomas į smulkesnius etapus: analizės (Case table) bei papildomų lentelių (Nested table) apibrėžimas, įėjimų (bei jų tipų) nurodymas; analizuojamo atributo nurodymas (Case). Sukūrus struktūrą, apibrėžiamas naudojamas algoritmas analizei, bei jo parametrai. Užregistravus servisą

įvykdomas algoritmo apmokymas, ir, jei rezultatai tenkina – rengiama prognozavimo užklausa. Jei reikalinga tikslinti parametrus, arba keisti gavybos algoritmą – grįžtama į prie algoritmo modifikavimo / parinkimo žingsnio.

2.7 pav. pateikiamas galimas scenarijus, kaip gali būti įvertinamas vienas ar keletas universitete dėstomų modulių, siekiant pagerinti tiek modulio kokybę, tiek studento žinių lygį. Vertinant modulį pagal vidurkį (ar pagal pažymių pasiskirstymą), radus anomalinę situaciją (čia griežtai neapibrėžiama, kriterijų pasirenka vartotojas) priimamas vienas ar kitas sprendimas: tobulinti modulį, arba rekomenduoti studentui pagilinti žinias. Jei pasirinktas kriterijus tenkinamas – šakos vykdymas užbaigiamas, jei ne, rekomendavus vieną iš sprendimų galima pereiti į modulių, lemiančių iškritimą paiešką, nes modulis su prastu vidurkiu jokiu būdu dar nereiškia, kad studentai jo neišlaikys, nesvarbu iš kelinto karto. Radus modulius, kuriuose koncentruojasi studijas nebaigę studentai, galima analizuoti tokius modulius pagal aukščiau aprašytą šaką, arba iš karto teikti rekomendacijas tobulinti modulį.

Įvertinant dalykus gali būti pasirinkta tik viena ar kita šaka, priklausomai nuo tikslo ir poreikių. Tą gali lemti ir skirtingo tipo aktoriai. Nutraukus šakos vykdymą, galima veiksmus pradėti iš naujo, pasirinkus tarkime kitą modulį, kitą semestrą, ar kitus kriterijus.

Veiksmų eiliškumo (sequence diagram) diagramos nepateikiamos, kaip menkai charakterizuojančios sistemos projektavimo bei panaudojimo sprendimus.



2.7 pav. Dalyko įvertinimo scenarijus

2.5 Funkciniai reikalavimai

Skyrelyje pateikiami funkciniai reikalavimai sistemai. Prie kiekvieno iš jų trumpai pakomentuojama kam jie reikalingi, kaip bus įgyvendinti.

- Sistemos galutinis vartotojas turi turėti galimybę prognozuoti studento būseną (iškris /neiškris). Reikalavimas įgyvendinamas ištreniravus modelį ir sugeneravus DMX skriptą.
- DB turi turėti galimybę būti atnaujintai. Galima realizuoti esamas lenteles papildyti naujais įrašais, kurių struktūra yra iš anksto suderinta.
- Turi būti galimybė atsikratyti įrašų su klaidingomis reikšmėmis. Pasiekiama per SQL užklausas MS SQL Server Management Sturio aplinkoje.
- Turi būti galimybė atstatyti trūkstamas reikšmes įrašuose. Trūkstamos reikšmės priklausomai nuo aplinkybių gali būti atstatomos dažniausiai pasitaikančia arba vidutine reikšme.
- Turi būti galimybė diskretizuoti reikiamo atributo reikšmes. Tai pasiekiama BI Developement Sturio aplinkoje, vaizdinyje įterpiant išvestinį stulpelį (angl. named calculation).
- Turi būti galimybė apmokyti algoritmą naujais duomenimis / atnaujinti modelį. Turint sukurtą struktūrą, modifikavus pradinius duomenis pakartotinai treniruojamas pasirinktas algoritmas.
- Turi būti galimybė gavybos struktūrą papildyti naujais modeliais. Skirtingi modeliai panaudojant skirtingus algoritmus, gali padėti identifikuoti naujus dėsningumus pradiniuose duomenyse.

2.6 Nefunkciniai reikalavimai

Pateikiami įvairių aspektų nefunkciniai sistemos reikalavimai. Kiekvienas trumpai pakomentuojamas. Nefunkciniai reikalavimai:

- Vartotojas turi turėti galimybę koreguoti parametrus, įtakojančius duomenų gavybą. BI Dvelelopement Studio galima keisti algoritmo parametrus, nuo kurių priklauso modelio tikslumas.
- Sistema turi būti pasiekiama tik jai skirtiems vartotojams. SQL Server Management Studio jungiantis prie analizės servisų ir DB variklio vartotojai autentifikuojami. Anglysis Servines projekte nurodant duomenų šaltinį ir jo pasiekiamumą.
- Sistemos prognozių rezultatas – būsena (iškris / neiškris iš studijų) turi būti vertinamas tik rekomendaciniu pobūdžiu.

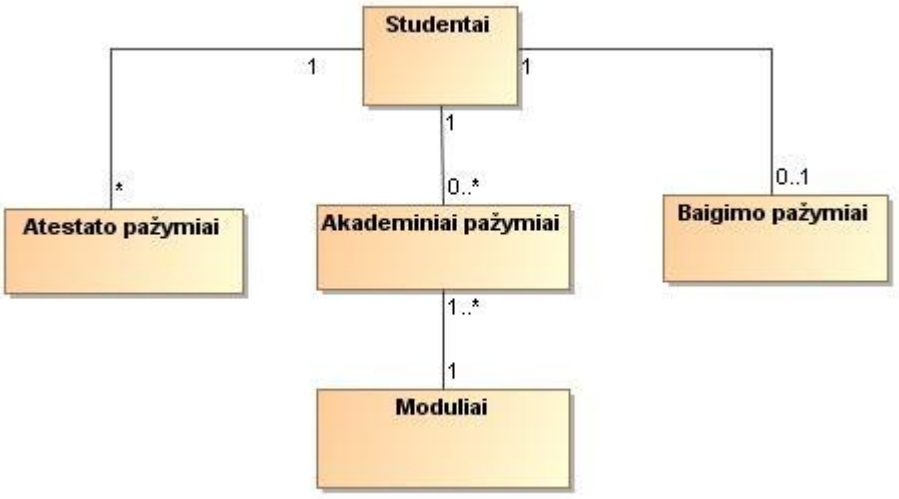
- Duomenų apsaugos sumetimais nenaudoti asmens kodų, vardų, pavardžių. Studentui identifikuoti pilnai pakaks jo unikalaus vidinio kodo iš studijų informacinių sistemų.

2.7 Reikalavimai duomenims, ER schema

Darbas įgyvendinamas naudojant Kauno technologijos universiteto informatikos fakulteto dieninio skyriaus bakalaurų pažymius. Analizei pasirinkti 2003 bei 2004 metais įstoję studentai (baigę atitinkamai 2007 ir 2008 metais). Atmetami po akademinių atostogų sugrįžę studentai, bei kiti mažiau tipiniai atvejai, kaip potencialiai iškreipsiantys standartinį dėsnį. Taigi turima dvejų metų stojusiųjų brandos atestato pažymiai, šių stojusiųjų mokymosi universitete pažymiai ir baigiamojo darbo pažymiai. Papildomai aprašomi universitete dėstomi moduliai.

Akademinei informacijai saugoti bus projektuojama MS SQL Server DBVS saugoma duomenų saugykla. Esybių ryšių diagramoje pateiktoje 2.8 paveiksle identifikuojami ryšiai bei jų kardinalumas tarp būsimų saugyklos (DB) lentelių. Pagrindinė unikalius studentus identifikuojanti esybė „studentai“. Siejamasi su trijų tipų pažymiais. Akademiniai pažymiai – su dalyko apibūdinimais „Moduliai“.

Projekto realizacijai skirtų duomenų šaltinis – Kauno technologijos universiteto informacijos sistemų tarnybos informacijos sistemų aptarnavimo skyrius.



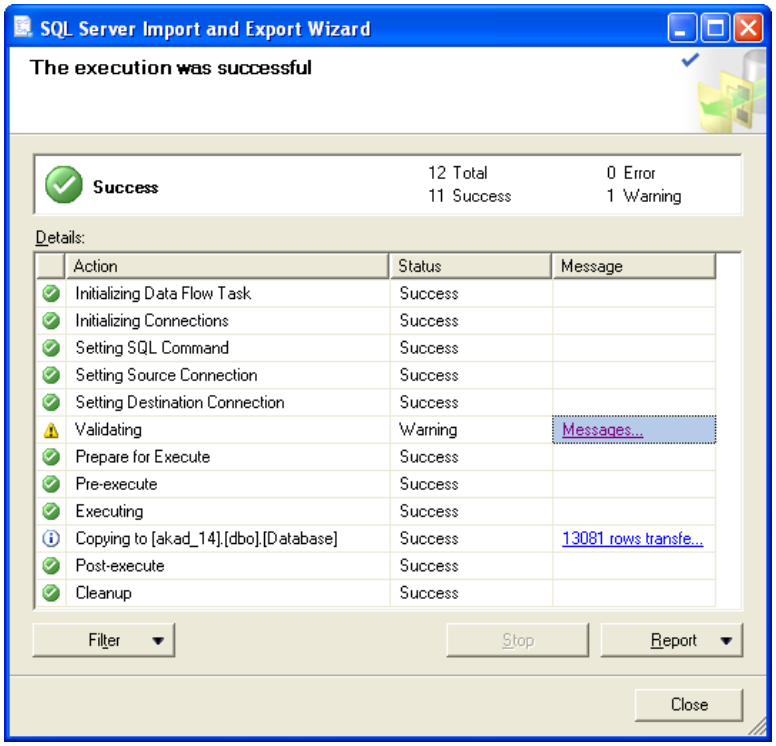
2.8 pav. ER diagrama, charakterizuojanti būsimą saugyklą.

Studentų esybė turės identifikuoti studento statusą studijų atžvilgiu. Galimo statuso būsenos – įstojęs į studijas, studijuoja, yra mainų programos studentas, išbrauktas iš sąrašų, kartoja semestrą, yra akademinėse atostogose, bei kita. Šiame darbe numatoma formuoti dvi galimas studento statuso būsenas. Jos yra „sėkmingai baigė studijas“ ir „iškrito iš studijuojančiųjų sąrašo“. Toliau jos bus vadinamos tiesiog būsenomis.

3. DUOMENŲ GAVYBOS REALIZACIJA

3.1. Saugyklos formavimas

Turima duomenų forma – DB3 tipo reliacinės lentelės, suderintos su MS Excel formatu, importuotos į MS SQL Server. Kadangi kiekvienam mokslo metų ciklui buvo po atskirą lentelę, jos importuojant (3.1. pav.) buvo sujungtos į vieną (Append). Studento lentelė suformuota stojusiųjų į informatikos fakultetą 2003-2004 metais pagrindu.



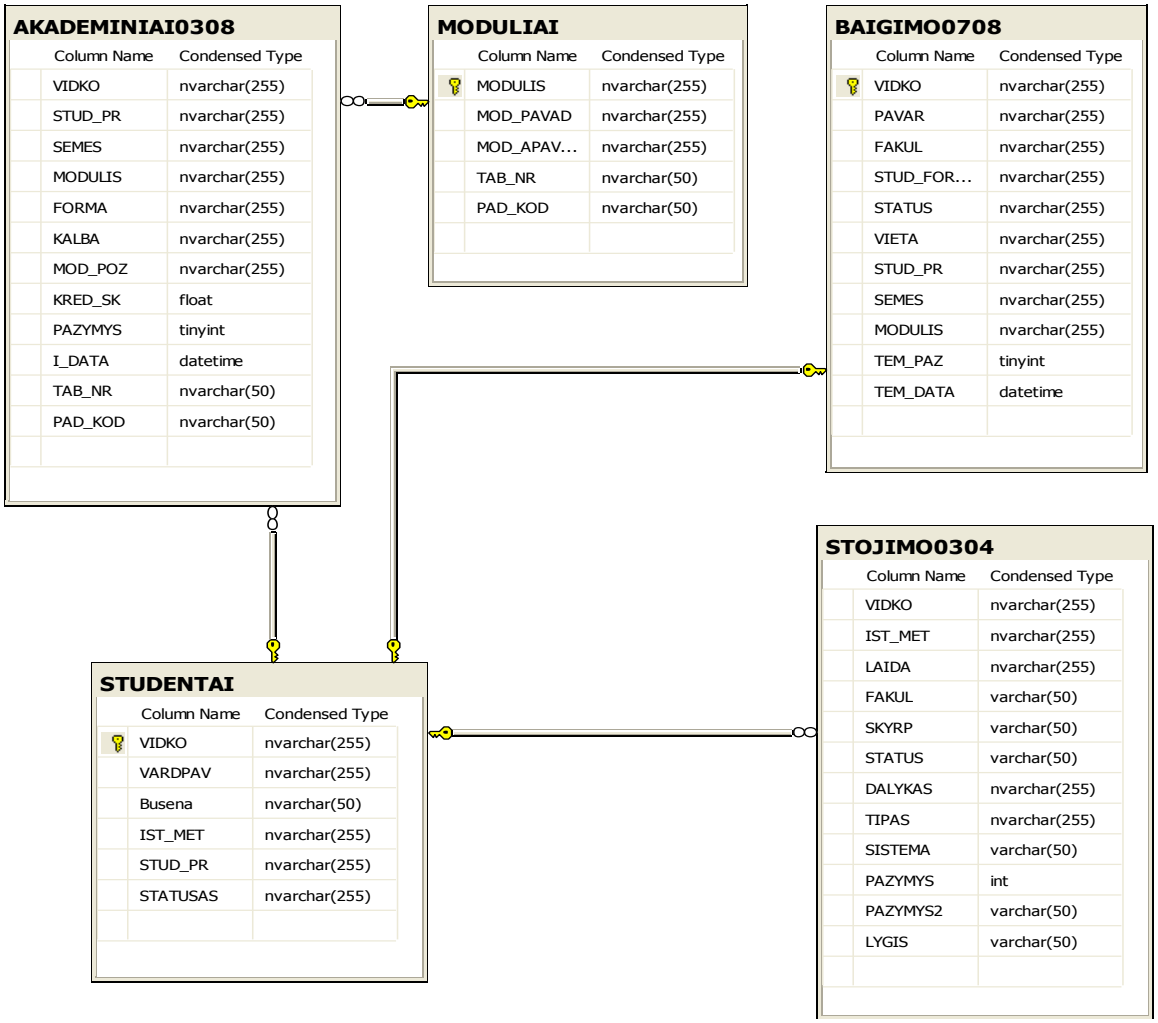
3.1 pav. Sėkmingo duomenų importo langas.

Visų lentelių importavimas į MS SQL Server buvo sėkmingas. Stojimo į universitetą abiturientų pažymiai buvo importuoti kaip *.csv failai. Tai yra tekstinis lentelės formatas, kuriame laukai atskiriami pasirinktu simboliu. Importuojant atskirų laukų tipai buvo modifikuoti. Duomenų apsaugos tikslais, visose lentelės panaikintas laukas su asmens kodais. Unikaliu

identifikatorium pasirinktas VIDKO atributas, kuris naudojamas įvairiose KTU sistemose studentams (darbuotojams) identifikuoti.

Vienas iš tikslų yra prognozuoti, remiantis brandos atestato pažymiais, ar būsimas studentas baigs universitete studijas. Tam studentų lentelėje įvedamas būsenos atributas. Laikysime, kad studentai, kurie baigė universitetą, įgis atributo reikšmę „1“; kurie iškrito įgis reikšmę „0“. Apie studentų būseną indikuoja absolventų lentelė. Tai reiškia, kad visiems įrašams studentų lentelėje, kurių VIDKO atributas sutampa su absolventų lentelės atitinkamu atributu suteikiama būsena „1“. Visiems likusiems – „0“.

Saugyklos formavimo etape realizuota ir dalis duomenų valymo: palikti tik į informatikos fakultetą stojusiųjų stojimo dalykai ir balai, atmesti anksčiau įstoję studentai, kaip necharakteringi būsimam tyrimui duomenys.



3.2 pav. Duomenų saugyklos schema

3.2. paveiksle pavaizduota MS SQL Server DBVS suprojektuota ir užpildyta duomenų saugykla. Lentelė „STUDENTAI“ – kiekvieną studentą apibrėžianti lentelė. „STOJIMO0304“ – studentų brandos atestato dalykai ir pažymiai. „AKADEMINIAI0308“ – pagal studijų individualiuosius planus studentų mokymosi universitete dalykai ir gautieji įvertinimai. „MODULIAI“ – studijų dalykus aprašanti lentelė. „BAIGIMO0708“ – absolventų lentelė.

Kuriant duomenų saugyklą, nebuvo tikslo minimizuoti jos dydžio, taigi duomenų tipai bei jų užimamas dydis buvo pritaikyti tik tiems laukams, kurie yra esminiai intelektualiajai duomenų analizei įgyvendinti.

3.2. Duomenų vaizdiniai

MS SQL Server Management Studio aplinkoje sukurti duomenų vaizdiniai, skirti išskaidyti stojimo duomenis į tris kategorijas. Tai yra valstybiniai egzaminai, mokykliniai egzaminai, bei metiniai pažymiai. Taip atlikta, norint objektyviau įvertinti atitinkamų kategorijų dalykų įtakas. Juolab kad valstybinių pažymių skalė yra šimtabalė, o mokyklinių, bei metinių pažymių – dešimtbalė. Dalykai vertinami įskaitomis į vaizdinius neįtraukiami.

Valstybinių egzaminų vaizdinio formavimo užklausa:

```
SELECT    VIDKO, IST_MET, DALYKAS AS VE_Dal, PAZYMYS AS VE_Paz, TIPAS, SISTEMA
FROM      dbo.STOJIMO0304
WHERE    (TIPAS = 'valstybinis egz.') AND (SISTEMA = '100')
```

Mokyklinių egzaminų vaizdinio užklausa:

```
SELECT    VIDKO, IST_MET, DALYKAS AS MEg_Dal, PAZYMYS AS MEg_Paz, TIPAS, SISTEMA
FROM      dbo.STOJIMO0304
WHERE    (TIPAS = 'mokyklinis egz.') AND (SISTEMA = '10')
```

Metinių pažymių vaizdinio užklausa:

```
SELECT    VIDKO, IST_MET, DALYKAS AS Met_Dal, PAZYMYS AS Met_Paz, TIPAS, SISTEMA
FROM      dbo.STOJIMO0304
WHERE    (TIPAS = 'metinis') AND (SISTEMA = '10')
```

Formuojant vaizdinius, dalinai realizuojamas duomenų koregavimas. Kitas koregavimo etapas – anomalinių reikšmių atmetimas – įgyvendintas BI Developement Studio sukurtųjų vaizdinių išvestiniuose stulpeliuose. Pateiktuose kodo fragmentuose, dėl aiškumo, kiekvienos kategorijos dalykai ir pažymiai pervadinti skirtingai.

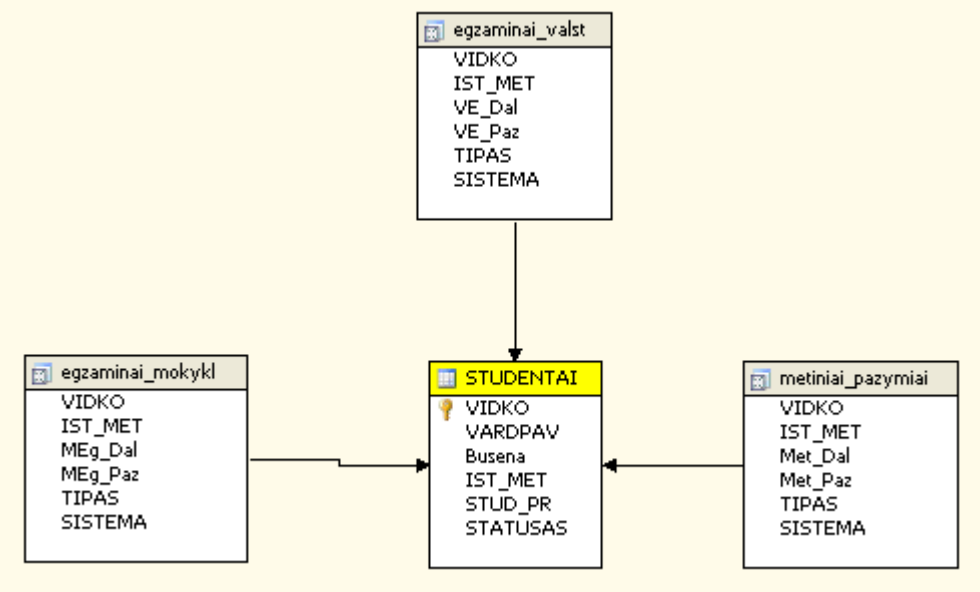
3.3. Analysis Services projektas

Projektas įgyvendintas Business Intelligence Developement Studio aplinkoje. Ji paveldi standartinę Visual Studio vartotojo sąsają, tik naudojamos skirtingos įrankių juostos. Projektas saugomas XML formatu. Šio kuriamo projekto pagrindinės aprašomos dalys yra duomenų šaltinis, duomenų šaltinio vaizdinys, bei duomenų gavybos struktūra. Duomenų gavybos struktūra gali susidėti iš vieno ar daugiau duomenų gavybos algoritmų (modelių). Apmokius modelį pradiniais duomenimis ir nustačius, kad jis yra tinkamas, kuriama DMX užklausa, kurios rezultatas – prognozė jau iš naujai paduotų duomenų. Ši užklausa gali būti vykdoma tiek BI Developement Studio tiek SQL Server Management Studio aplinkose.

Sukūrus projektą, jis yra įtraukiamas į MS SQL Server DBVS kaip analizės servisas. Tam įvykdomas išdėstymo (angl. Deployment) procesas.

3.3.1. Nayve Bayes algoritmo taikymas

Taikant šį algoritmą, turimas pirmas tikslas – išanalizuoti, kurie kiekvienos kategorijos, aprašytos 3.2 skyriuje dalykų pažymiai labiausiai įtakoja studentų mokimąsi universitete, arba iškritimą. Antras tikslas – pažiūrėti, kurie pirmojo semestro dalykai lemia, ar studentai baigs studijas. Iš analitinės dalies žinome, kad šiam algoritmui reikalinga mažiausia skaičiavimų, taigi jo apmokymas reikalauja mažiausiai laiko. BI Developement Studio suformuotas gavybos duomenų vaizdinys pateikiamas 3.3 paveiksle.



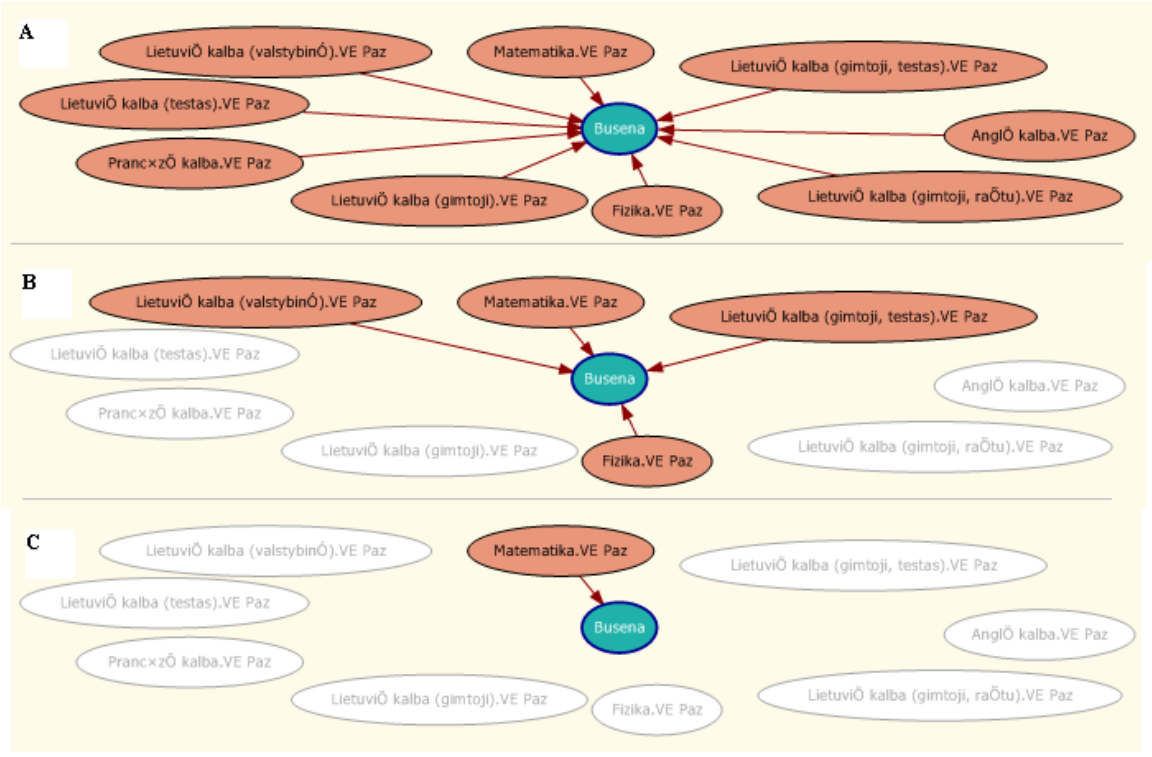
3.3 pav. Vaizdinys brandos atestato dalykų analizei

Norint įvertinti visus tris stojimo pažymių tipus, buvo sukurta gavybos struktūra, įtraukiant 3.3 pav. pateiktas lenteles. Struktūroje sukurti trys identiški modeliai, kuriuose iš gavybos išjungta nereikalingi įėjimai. 3.4 pav. matoma gavybos struktūra su visais trimis modeliais, Naive Bayes algoritmo panaudojimui.

Structure	Mag_Bayeso_Valst	Mag_Bayeso_Mok	Mag_Bayeso_Met
	Microsoft_Naive_Bayes	Microsoft_Naive_Bayes	Microsoft_Naive_Bayes
Busena	Predict	Predict	Predict
Egzaminai Mokykl	Ignore	Input	Ignore
M Eg Dal	Ignore	Key	Ignore
M Eg Paz	Ignore	Input	Ignore
Egzaminai Valst	Input	Ignore	Ignore
VE Dal	Key	Ignore	Ignore
VE Paz	Input	Ignore	Ignore
Metiniai Pazymiai	Ignore	Ignore	Input
Met Dal	Ignore	Ignore	Key
Met Paz	Ignore	Ignore	Input
VIDKO	Key	Key	Key

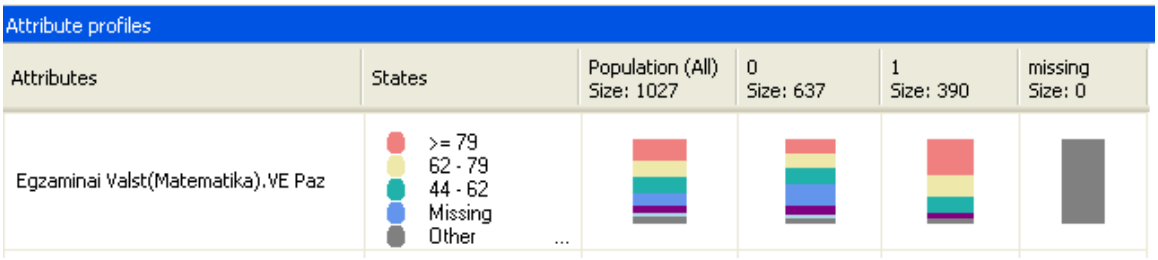
3.4 pav. Gavybos struktūra brandos atestato dalykų analizei

Gauti rezultatai – grafinė algoritmo interpretacija atvaizduojama gavybos modelio rodymo lange. Skirtingų atributų įtaka tiriamai būsenai yra nevienareikšmiška. Naive Bayes algoritmo rezultatų peržiūros lange ryšių stiprumus galima įvertinti, eliminuojant silpniausius.



3.5 pav. Naive Bayes algoritmo sugeneruotas priklausomybių tinklas

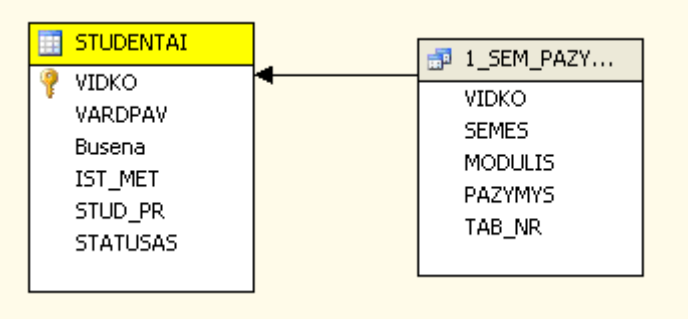
3.5 paveiksle A atvejis atvaizduoja visus studentų būseną įtakojančius valstybinių egzaminų dalykus. B atvejis tolydžiai paliekant įtakingiausius. Iš C atvejo matome, kad svarbiausias yra valstybinis matematikos egzaminas. Priklausomybių tinklas dar nenusako, kokią būseną ir kurios atributų reikšmės labiausiai įtakoja. Tą vizualizuoja atributų savybių langas. 3.6 paveiksle pateiktame fragmente matyti, kaip egzamino rezultatų intervalai pasiskirstę tarp baigusiųjų studijas, iškritusiųjų, bei visų kartu studentų. Atributų savybių lange analogiškai detalizuoti visi priklausomybių tinkle esantys dalykai. Šių duomenų skaitinę versiją nesunkiai galima įkelti į MS Excel skaičiuoklę įvairialypiam apdorojimui.



3.6 pav. Atributų savybių fragmentas

Taikant algoritmą brandos atestato pažymių analizei, sumažintos parametro MINIMUM_DEPENDECY_PROBABILITY reikšmės, siekiant įtraukti daugiau įvertintų dalykų. Duomenų rezervavimo integruotam testavimui, dėl itin jautrios imties, atsisakyta.

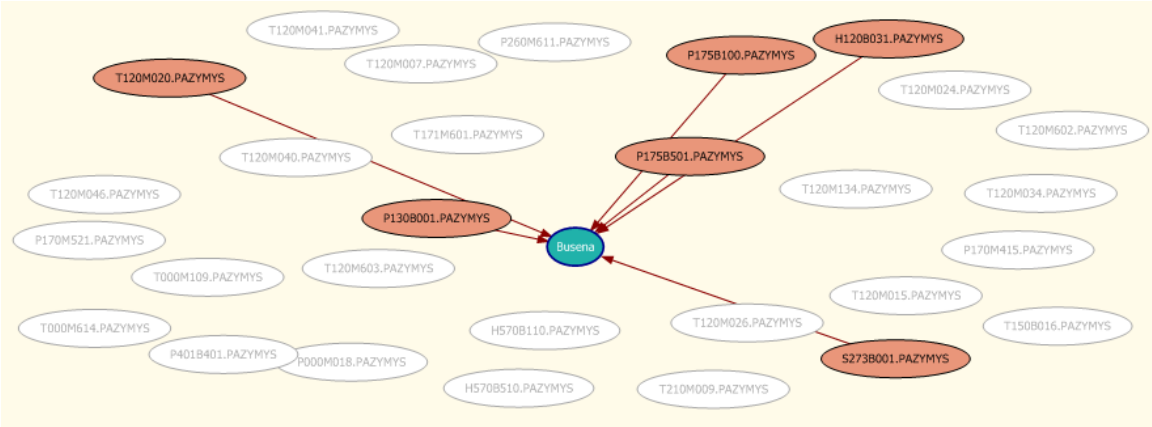
Naive Bayes algoritmas buvo panaudotas ir universitete įgytų pažymių analizei, bei modulių įtakai baigti studijas įvertinti. Apsiribota pirmuoju semestru. Duomenų gavybos struktūros vaizdinys pateikiamas 3.7 paveiksle.



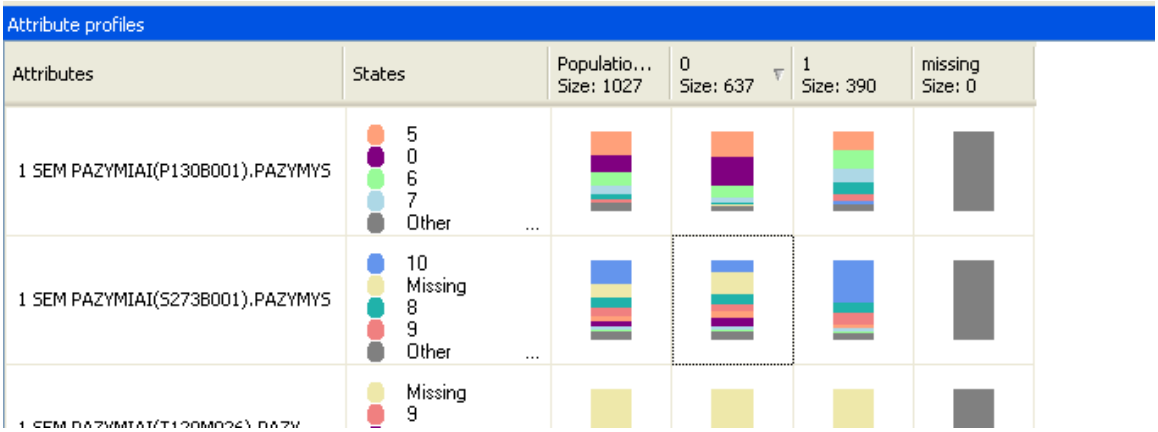
3.7 pav. Vaizdinys pirmojo semestro analizei

Gauti rezultatai – priklausomybių tinklas 3.8 pav., bei atributų charakteristikos 3.9 pav. Stipriausias ryšys pritaikius Naive Bayes algoritmą nurodomas su moduliu P130B001 (Tiesinė

algebra ir diferencialinis skaičiavimas), toliau seka S273B001(Kūno kultūra 1). Iš atributų charakteristikų matome, kad P130B001 modulio neigiama įtaka pasireiškia didele dalimi „0“ reikšmės įgijimu – tai reiškia, kad už šį modulį neatsiskaityta. Tuo tarpu P130B001 modulyje didžiausią dalį užimanti „missing“ reikšmė nurodo, kad daugelis iškritusiųjų studentų šio modulio net nelankė, arba nebuvo įtraukti į jį. Baigusiųjų stulpelyje (Būsena =„1“) tokių iš viso nėra.



3.8 pav. Pirmojo semestro modulių įtaka būsenai



3.9 pav. Pirmojo semestro modulių charakteristikų fragmentas

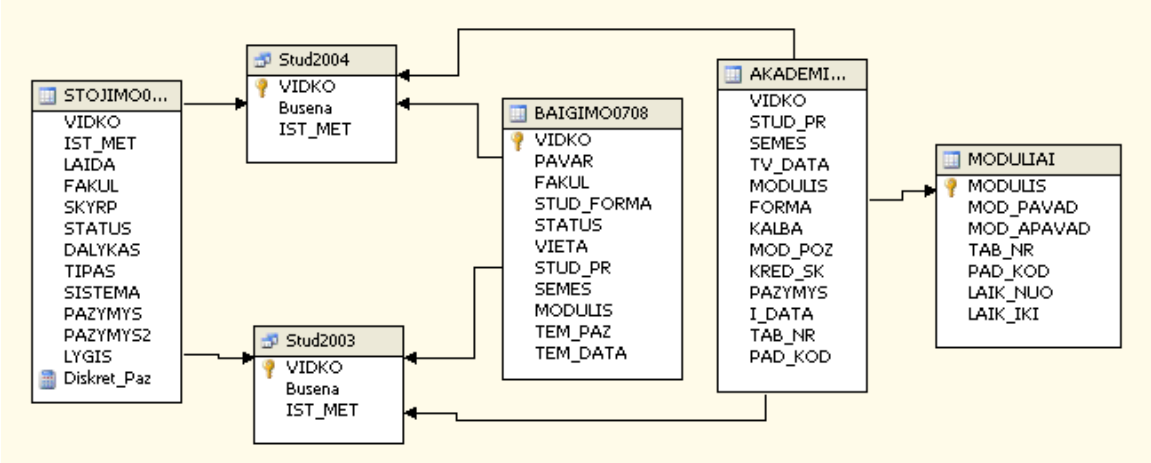
Taikant algoritmą pirmojo semestro modulių analizei, parametro MINIMUM_DEPENDECY_PROBABILITY reikšmė buvo padidinta iki 0,6 siekiant išmesti mažiau įtakingus modulius iš analizės.

3.3.2. Grupavimo algoritmo taikymas

Grupavimo arba kitaip klasterizacijos algoritmas neparodys atskirų dalykų įtakų stiprumo studento būsenai. Taikant šį algoritmą stojimo ir mokymosi universitete pažymiai suskirstomi į

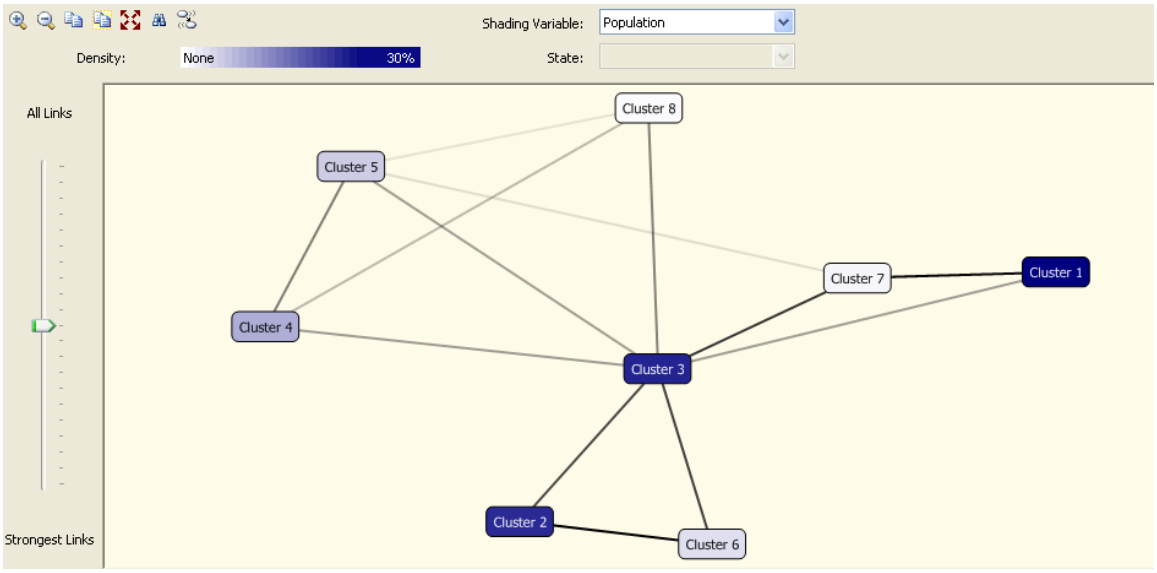
grupes, pagal dalykus, jų pažymius, bei tų pažymių pasiskirstymą. Šiame taikyme turimas tikslas ne tik įvairių pasiskirstymų analizė, tačiau ir būsenų prognozė. Reikia padalinti studentų lentelę pagal stojimo metus. 2003 metų stojimo studentus panaudosime modeliui treniruoti. 2004 metų stojimo studentams bus prognozuojama iškritimo arba studijų baigimo faktas.

3.10 paveiksle suformuotame vaizdinyje iš lentelės „STUDENTAI“ atliktas skaidymas į dvi, atskiriančias studentus pagal stojimo metus (Stud2004 ir Stud2003). Šį kartą visi atestato dalykai, ar tai būtų metinis pažymys, ar valstybinis egzaminas, traktuojami vienodai.



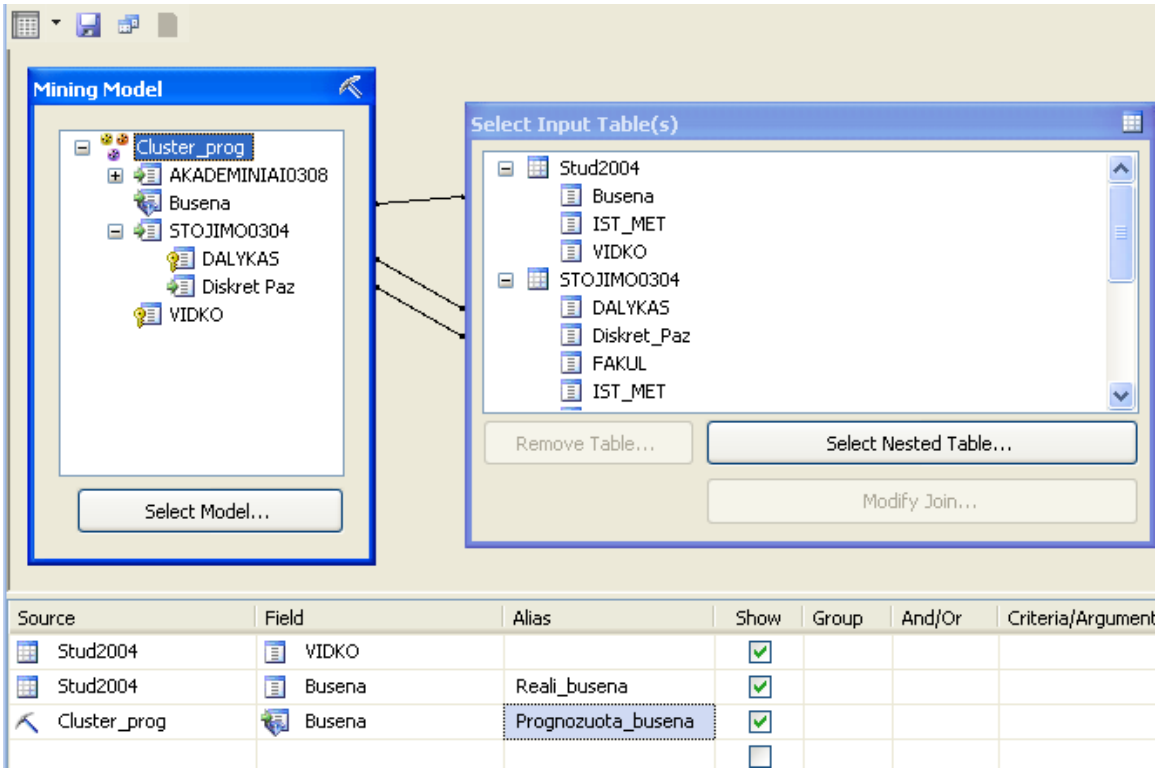
3.10 pav. Vaizdinys būsenų prognozei

Taikant grupavimo algoritmą, buvo pakeisti šie parametrai: MAXIMUM_STATES - 128; MAXIMUM_INPUT_ATTRIBUTES – 1024; CLUSTER_COUNT – 10. padidinus įėjimų skaičių, bei galimų būsenų skaičių, į algoritmo apmokymą buvo įtraukti visi atributai su visomis būsenomis. 3.11 paveiksle pavaizduota, kaip nepriklausomai nuo būsenos, pasiskirsto visi studentai į klasterius. Matome, kad didžiausia imtis yra pirmame klasteryje. Iš pirmo žvilgsnio tai mažai reikšminga, tačiau žinant tam tikros būsenos pasiskirstymą tam tikrame klasteryje, galima įvertinti jų savybes, analogiškai kaip Naive Bayes algoritmo panaudojime (kaip 3.6 pav). Atributų reikšmių pasiskirstymas tarp klasterių yra jau tam tikros taisyklės, pagal kurias galima įvertinti naujus duomenų rinkinius, kurie nėra įtraukti į algoritmo apmokymą. Duomenų rezervavimo integruotam testavimui, dėl itin jautrios imties, atsisakyta. Rezervuojant net ir mažiau nei 10% duomenų, jau sumažėja prognozavimo kokybė.



3.11 pav. Bendras studentų pasiskirstymas klasteriuose

Sekantis etapas – būsenų prognozavimas 2004 metų studentams, kai „žinomi“ tik jų stojimo į universitetą, tai yra brandos atestato pažymiai.



3.12 pav. Prognozės rengimas Mining Model Prediction komponente

Prognozei parengti - nurodomas koks algoritmas ir struktūra bus naudojama, bei kokiai atvejų lentelei bus atliekama prognozė. Šiuo atveju, vienas iš prognozės įėjimų yra stojimo

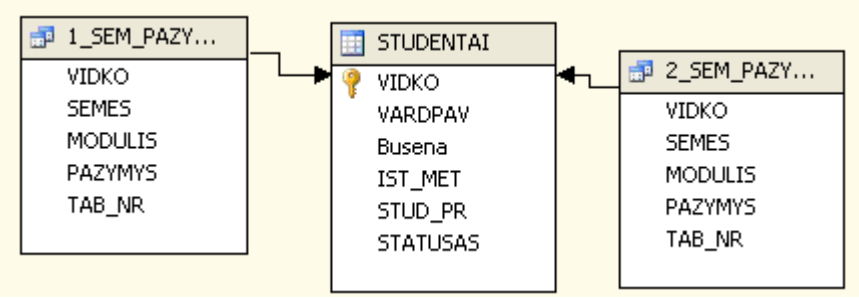
duomenys taigi, reikia nurodyti papildomą „Nested“ lentelę (3.12 pav.). Prognozės rezultatas – unikalus studento identifikatorius VIDKO, studento tikroji būsena, bei prognozuojama būsena. Tikroji būsena pasirenkama, kad būtų galima palyginti ir įvertinti, kaip tiksliai pavyko suprognozuoti.

Parengta prognozės užklausa – DMX skriptas gali būti išsaugota faile. Šį skriptą galima naudoti tiek SQL Server Management Studio, tiek BI Developement aplinkoje. Užklausos rezultatą galima išsaugoti kaip lentelę duomenų pasirinktoje DB. Užklausos rezultato fragmentas pateikiamas 3.13 paveiksle.

VIDKO	Reali_busena	Prognozuota_b...
83197	0	0
83198	1	1
83199	0	0
83200	1	0
83201	0	0
83202	1	1
83203	0	0
83204	0	0

3.13 pav. Būsenų prognozės fragmentas

Kitas grupavimo algoritmo taikymo atvejis – skirtas pirmojo semestro universitete dalykų įtakai iškritimui įvertinti. Tam formuojamas vaizdinys su visais turimais studentais, bei pasirinktojo semestro (pirmojo) pažymių filtravimo lentele. 3.14 paveiksle pateiktame vaizdinys yra pirmųjų metų analizei, išskaidant į atskirus semestrus.

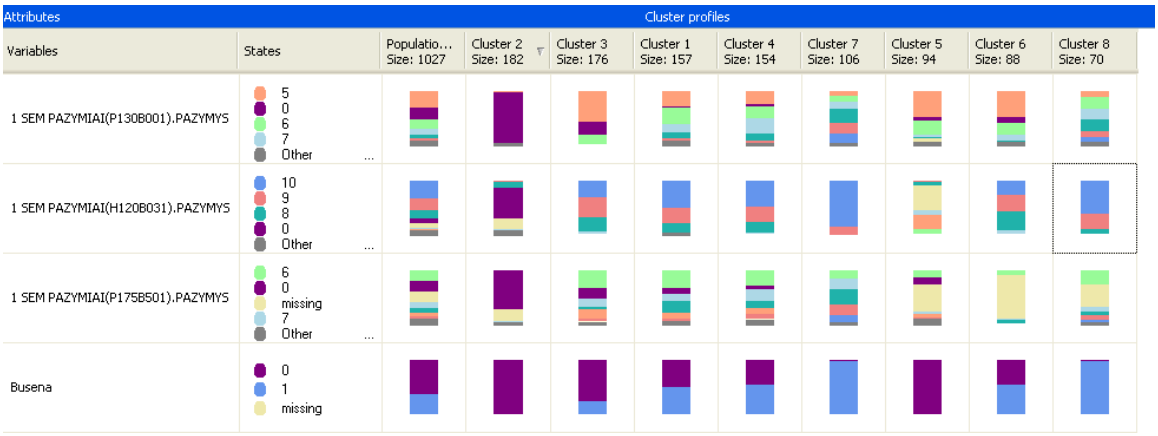


3.14 pav. Vaizdinys pirmojo ir antrojo semestro analizei

Studentų gautuosius pažymius – kaip atributą galima traktuoti tiek kaip diskretųjį dydį, tiek kaip tolydųjį. Sugeneravus klasterius nurodant skirtingus tipus, analizei gauname skirtingas statistikas. Taigi, kiekvienu atveju atskleidžiami nauji analizės aspektai.

3.15 Paveiksle matome, grupavimo algoritmo rezultatą – atskirų klasterių atributų

reikšmių ir galimų studentų būsenų pasiskirstymą. Šiame pavyzdyje matome, kad daugiausiai iškritusių studentų lyginant su neiškritusiais yra 2 ir 5 klasteriuose. Atlikus lygiavimą pagal antrąjį klasterį, matome, kad modulio P130B001 neišlaikė beveik visi studentai, priskirti tam klasteriui, o tokių yra 182 iš visų 1027 studentų, kurie įstojo 2003 ir 2004 metais. Aptariamas modulis yra „Tiesinė algebra ir diferencialinis skaičiavimas“.



3.15 pav. Pirmojo semestro diskrečios analizės fragmentas

3.16 pav. Pirmojo semestro tolydinės analizės fragmentas

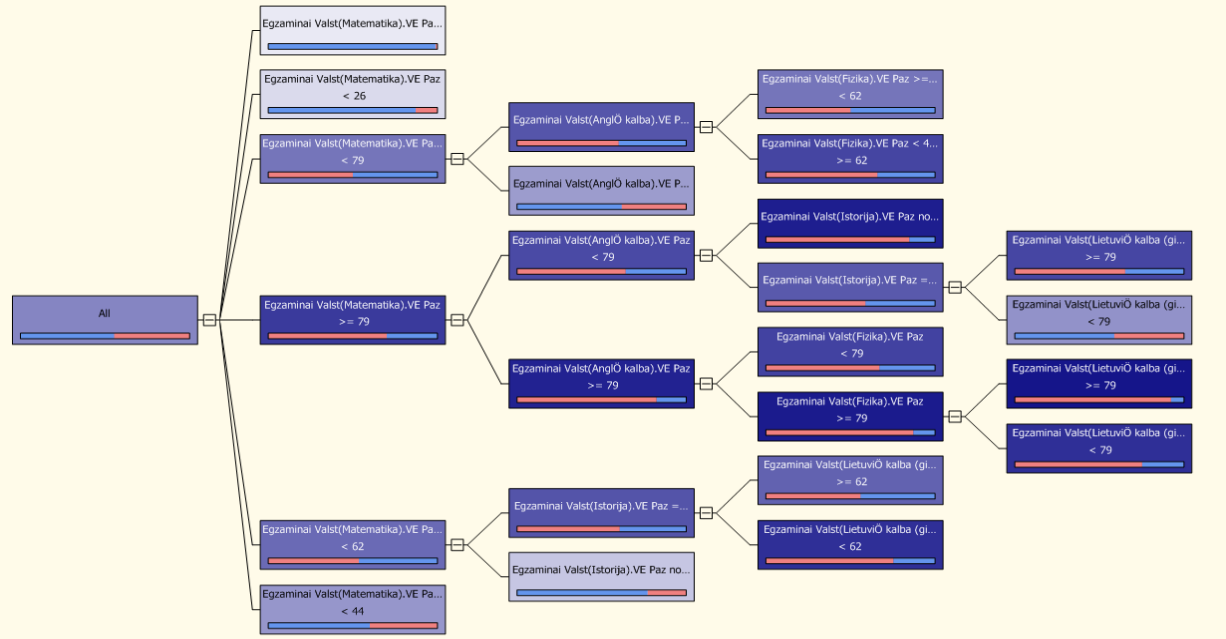
Kada atributai – pažymiai yra traktuojami, kaip tolydieji dydžiai, grupavimo algoritmas duoda jau kitokio pobūdžio informaciją. 3.16 paveiksle pavaizduotame fragmente būsenos dydis paliktas diskretus, o dalykų įvertinimai – tolydūs. Susiradę klasterį su didžiausia dalimi baigusiųjų studijas (1 klasteris), matome, kad aukščiau aptartojo P130B001 modulio vidurkis yra

7,09, su galimu 1,43 standartiniu nuokrypiu. Grafiškai – rombo vidurys – vidurkis, aukštis – galimas nuokrypis. Šios statistikos taip pat yra paskaičiuojamos visiems, nepriklausomai nuo pasiskirstymo klasteriuose, dalykams (Population stulpelyje).

Taikant grupavimo algoritmą semestro pažymių analizei galima nuodugniai suvokti, tiek gautųjų įvertinimų pasiskirstymą, tiek pagrindines statistikas. Visa tai apibendrinus, fakulteto atstovai gali priimti studijų procesą teigiamai įtakojančius sprendimus.

3.3.3. Sprendimų medžio algoritmo taikymas

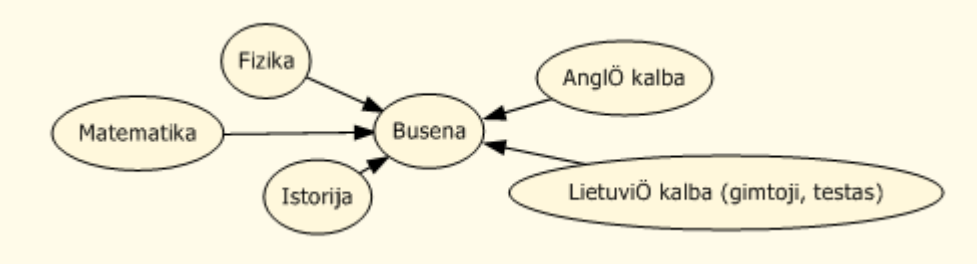
Sprendimų medžio algoritmas pritaikytas dviems uždaviniams. Pirmiausia – kaip ir Naive Bayes – patyrinėti atskirų tipų atestato pažymius, ir patikrinti, ar rezultatai atitinka. Kitas tikslas – panaudoti visus atestato pažymius generuojant sprendimų medį, kuris, kaip ir grupavimo algoritmas pasitarnautų studentų būsenų prognozavimui. Taigi, pirmuoju atveju buvo naudotasi vaizdiniu pateiktu 3.3 paveiksle, antruoju – prognozavimo atveju – 3.10 paveiksle pateiktu vaizdiniu.



3.17 pav. Valstybinių egzaminų rezultatų ir „1“ būsenos pasiskirstymas

3.17 pateiktame sprendimų medžio paveiksle atvaizduota, kaip charakterizuojami studentai, sėkmingai baigę universitetą. Kuo lapas tamsesnės spalvos, tuo įgijusių būseną „1“ studentų daugiau. Bendrą santykį su nebaigusiais parodo horizonalioji skalė (mėlyna spalva – iškritę, raudona spalva – baigę studijas). Atskirų dalykų įtakas parodo priklausomybių tinklas

(3.18 pav.).



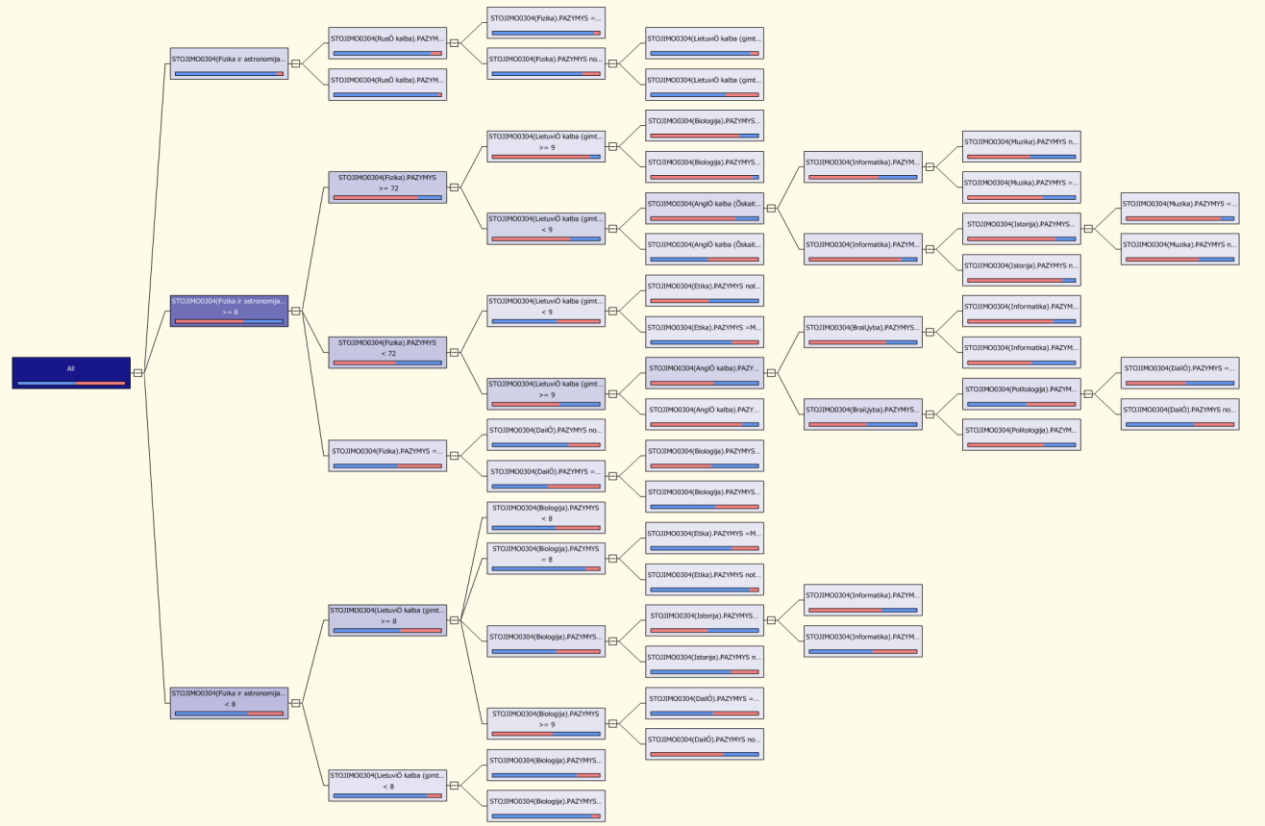
3.18 pav. Priklausomybių tinklas būsenos įtakai

Kaip ir Naive Bayes algoritmo taikyme, stipriausias ryšys išlieka Matematikos valstybinis egzaminas. Sekantis pagal stiprumą egzaminas – lietuvių kalbos testas. Taikant šį algoritmą parametro SCORE_METHOD reikšmė buvo pakeista į 1 (Lūžio taškai pagal C. Shanon populiacijos entropinio pasiskirstymo tikimybę).

Būsenų prognozių taikymo atveju, algoritmo treniravimui buvo paduoti visi nekategorizuoti brandos atestato pažymiai. Kadangi įėjimų skaičius viršija numatytąjį, pakoreguoti šie įėjimo parametrai – MINMUM_INPUT_ATTRIBUTES – 1024; MINIMUM_OUTPUT_ATTRIBUTES – 1024. Siekiant įtraukti į algoritmą daugiau dalykų, parametro COMPLEXITY_PENALTY tikimybė sumažinta iki fiksuotos 0,25 reikšmės. SCORE_METHOD reikšmė – 1. Integruotų testavimo atvejų atsisakyta.

3.19 paveiksle sugeneruotame sprendimų medyje vaizduojama, visų tipų brandos atestato pažymių pasiskirstymas, bei visa imtis, nežiūrint, ar 0 ar 1. Populiacijos koncentraciją charakterizuoja mėlynos spalvos tamsumas; pasiskirstymą tarp būsenų – horizontali skalė (raudona – 1; mėlyna – 0).

Kaip ir grupavimo algoritmui, sprendimų medžio algoritmo taikymui prognozėms, sugeneruota DMX užklausa.

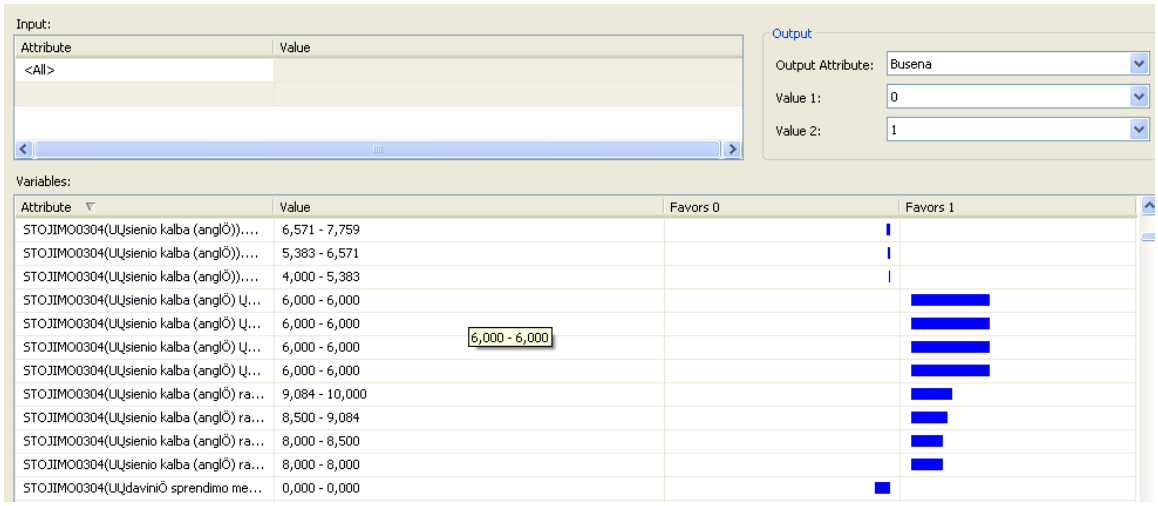


3.19 pav. Atestato dalykų ir visos imties pasiskirstymas.

3.3.4. Neuroninio tinklo taikymas būsenų prognozavimui

Neuroninio tinklo įėjimams aprašyti naudotas vaizdinys, pateiktas 3.10 paveiksle. Taikant neuroninių tinklų algoritmą grafinės prasmingos jo interpretacijos nėra. Tačiau galima įvertinti, koks įėjimas, su kokiomis galimomis būsenomis, bei su kokia tikimybine išraiška įtakoja tiriamą kintamąjį – neuroninio tinklo išėjimą (3.20 pav.). Šio taikymo tikslas yra pasinaudoti algoritmu, kad būtų galimybė analogiškai prognozuoti studentų būseną - ar iškris, ar baigs universitetą.

Realizuojant algoritmą padidintos algoritmo parametrų MAXIMUM_INPUT_ATTRIBUTES bei MAXIMUM_OUTPUT_ATTRIBUTES reikšmės, kad į modelį būtų įtraukti visi dalykai. Kadangi valstybinio egzamino skalė šimtabalė, padidinta MAXIMUM_STATES parametro reikšmė. Taikant neuroninių tinklų algoritmą užteko apibrėžti įėjimų mazgus bei išėjimo mazgą – būsenos atributą. Viduriniojo sluoksnio neuronų logika ir kiekis S_{mid} apspręstas pagal nutylėjimą, tai yra $S_{mid} = 4 \times \sqrt{230 + 663} \times 1 \approx 120$. Algoritmo treniravimas buvo imliausias kompiuterio resursams, lyginant su aukščiau aprašytais taikymais.



3.20 pav. Įėjimų su galimomis reikšmėmis įtaka būsenai

Prognozės DMX užklausa parengta analogiškai, kaip ir grupavimo bei sprendimų medžio algoritmams.

3.4. Algoritmų taikymo apibendrinimas

Buvo sėkmingai pritaikyti 4 duomenų gavybos algoritmai universiteto akademinių duomenų analizei. Viena taikymo sritis – pačių duomenų intelektuali analizė, kita sritis – galimybė pritaikyti algoritmus prognozavimui.

Išsamiai pačių duomenų analizei geriausiai tinka grupavimo bei Naive Bayes algoritmai. Grupavimo algoritmas labiausiai tinkamas analizei dėl galimybės duomenis apdoroti tiek kaip diskrečius, tiek kaip tolydžius. Tuo būdu atskleidžiama daugiau įvairesnių statistinių aspektų. Naive Bayes algoritmo rezultatus lengviau interpretuoti (konkrečiau apibrėžtos pasiskirsčiusių dydžių kategorijos).

Norint atlikti greitą tam tikrų įtakų pasiskirstymą, geriausia naudoti Naive Bayes ir sprendimų medžio algoritmus, dėl galimybės atvaizduoti priklausomybių tinką. Tai lengviausiai interpretuojama apibendrinta rezultatų išraiška.

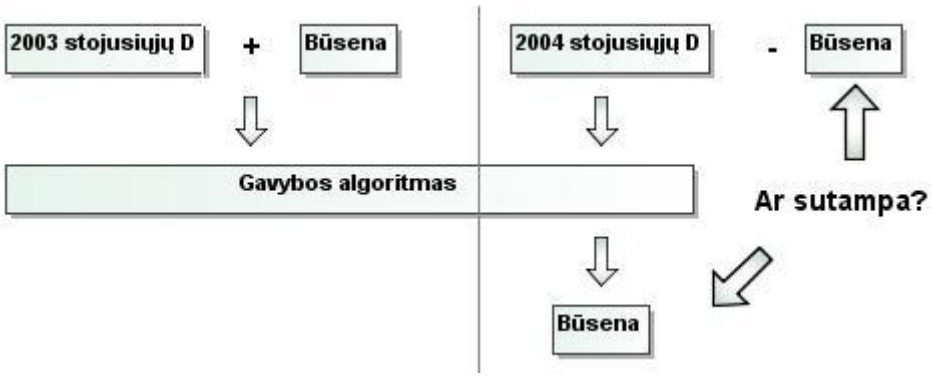
Būsenų prognozavimui galima panaudoti visus keturis aptartuosius algoritmus. Prognozavimo kokybės savybių tyrimas aprašytas tolimesniame skyriuje.

Atliekant algoritmų treniravimą pradiniais duomenimis, koreguoti numatytieji jų parametrai, taip išgaunant didesnį tikslumą, arba atitinkamai supaprastinant modelį.

4. GAVYBOS ALGORITMŲ PROGNOZAVIMO KOKYBĖS TYRIMAS

4.1. Tyrimo tikslas ir apžvalga

Trečiame skyriuje išdėstyta ir pademonstruota, kaip duomenų gavybos algoritmai pritaikyti ne tik akademinių universiteto duomenų analizei, tačiau ir prognozuoti ar iškris ar baigs studijas studentai, kai žinomi tik jų brandos atestato pažymiai. Taigi, turimos keturios prognozavimo galimybės, pagal kiekvieną algoritmą. Tyrimo tikslas yra nustatyti, kurį duomenų gavybos algoritmą taikant studento būsenos prognozuojamos tiksliausiai. Tyrimo problema atvaizduota 4.1. paveiksle. Norint ištirti, kuris algoritmas tiksliausiai vykdo prognozę, reikalinga turėti du pilnus dalykų / pažymių rinkinius. Pirmąjį rinkinį panaudoti gavybai, antrą – prognozėms. Prognozuojamo rinkinio būsenos laikoma, kad yra nežinomos ir kiekvieną kartą jas reikės sulyginti su faktiniais duomenimis.



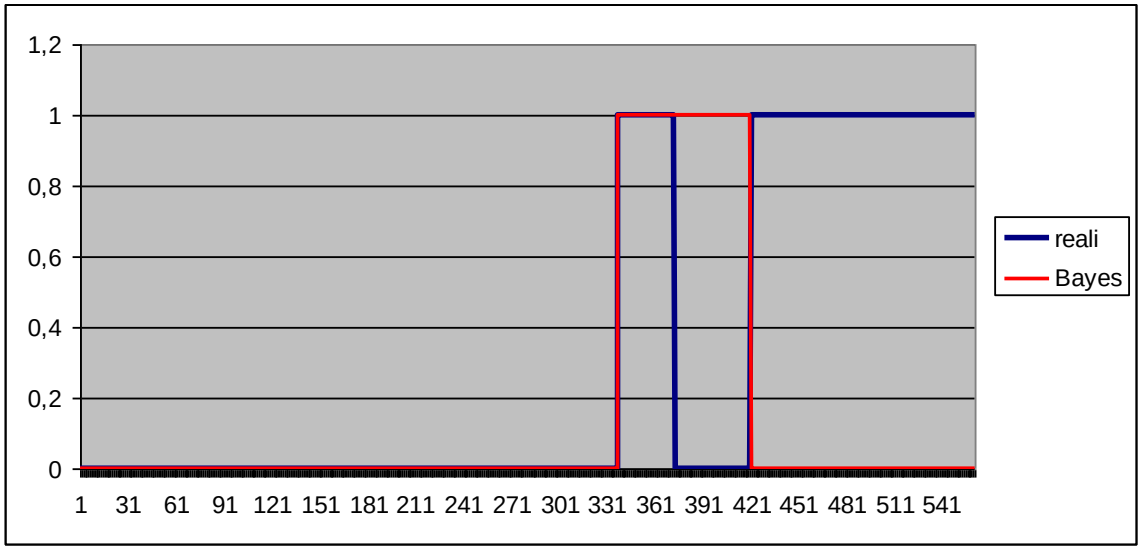
4.1 pav. Būsenų prognozių problematika

Prognozuojamos būsenos gaunamos įvykdžius DMX užklausas. Tyrimui duomenys bus įkelti į MS Excel programą, kur ir bus statistiškai apdorota, palyginta, bei apibendrinta.

4.2. Prognozių rezultatų įvertinimas

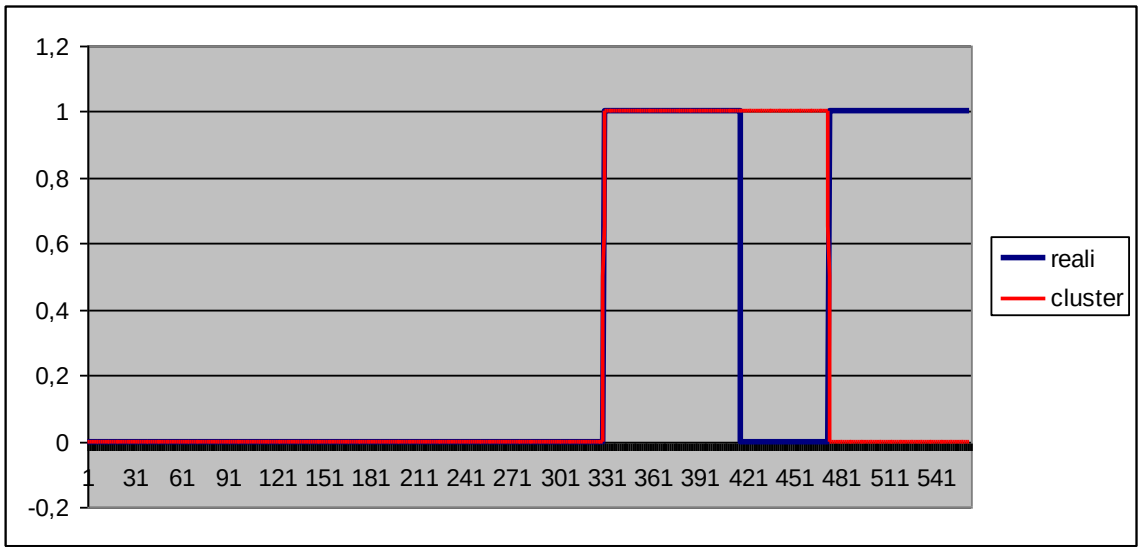
Naive Bayes algoritmo rezultatai pateikiami 4.2 paveiksle. Aiškumo sumetimais, grafike pradžioje nurodomos sutampančios iškritusiųjų studentų būsenos, toliau sutampančios baigusiųjų studijas būsenos. Paskiausiai eina nebesutampančios būsenos. 374 atvejais iš 562 būsenų prognozė sutampa su realia. Gauname, kad Naive Bayes algoritmo taikymo efektyvumas yra

66%. Iš grafiko matyti, kad algoritmas geriau prognozuoja „0“ būseną.



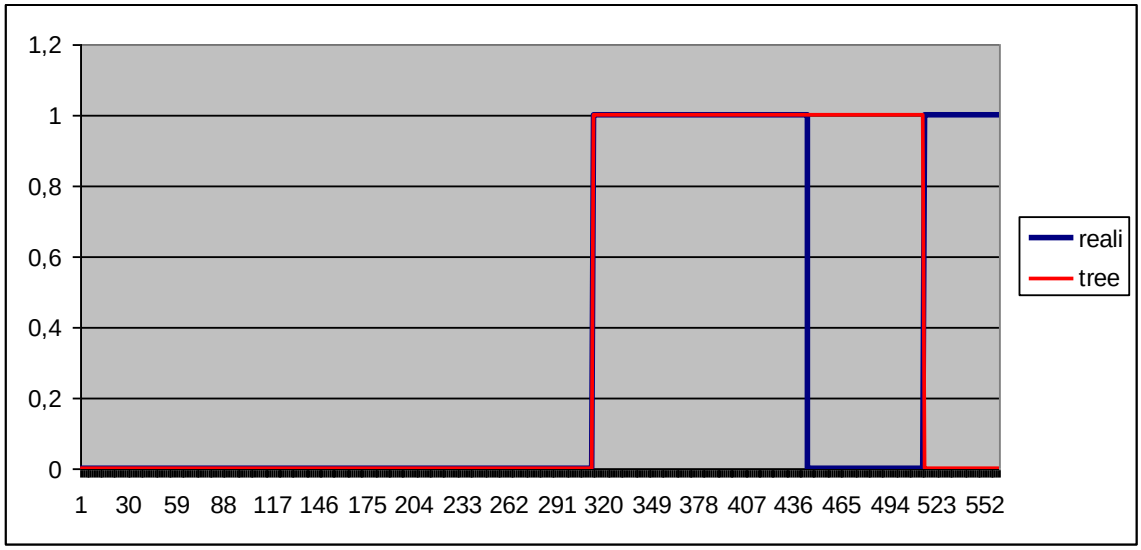
4.2 pav. prognozių pasiskirstymas pagal Naive Bayes algoritmą

Kitas įvertinamas algoritmas – grupavimo (4.3 pav.). Patikrinus būsenas, gauta, kad prognozės sutampa 419 iš 562 atvejų. Tai apytiksliai lygu 74 procentams.



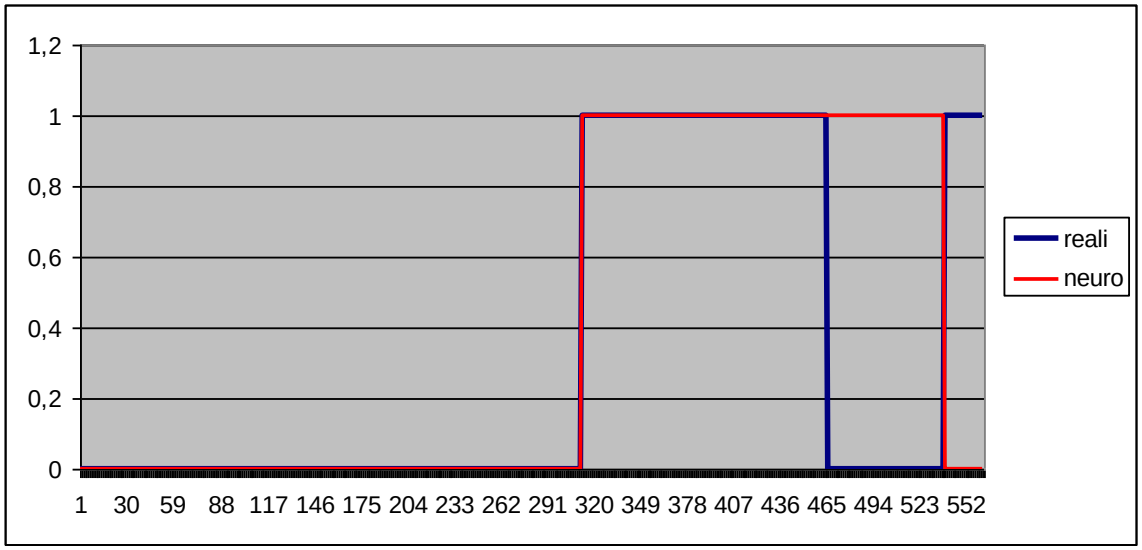
4.3 pav. prognozių pasiskirstymas pagal grupavimo algoritmą

Trečias vertintas algoritmas – sprendimų medžio. Jo prognozių grafinė išraiška pateikta 4.4 paveiksle. Rezultatai sutapo 445 iš 562 atvejų. Tai yra apytikriai 79 procentai sutampančių atvejų. Taip pat pastebima „optimistiško“ prognozavimo tendencija.



4.4 pav. prognozių pasiskirstymas taikant sprendimų medžio algoritmą

Paskutinis įvertintas algoritmas – neuroninių tinklų (4.5 pav.). Algoritmo prognozės sutapo 466 atvejais iš 562. Tai yra 83 porcentai. Akivaizdus algoritmo „optimistiškumas“ prognozuojant būsenas.

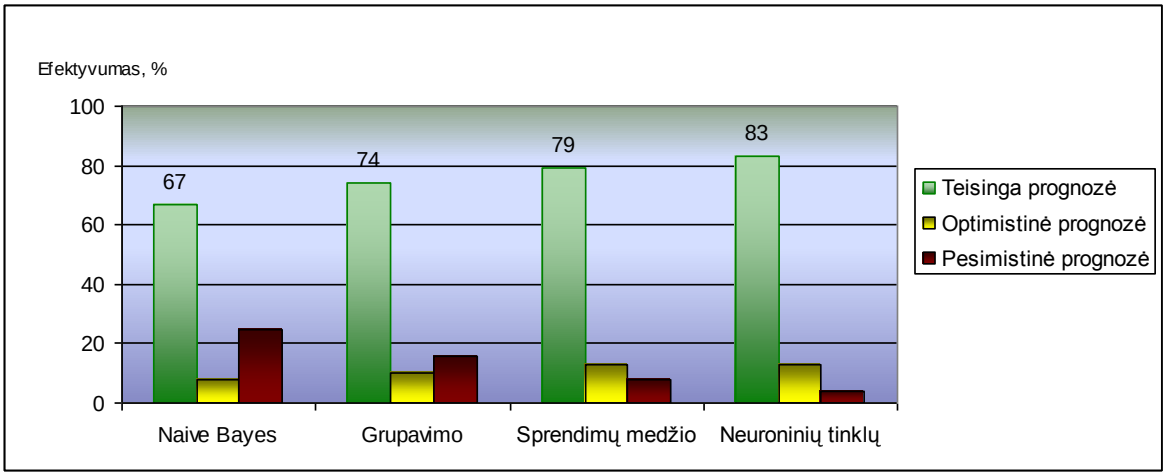


4.5 pav. prognozių pasiskirstymas taikant neuroninių tinklų algoritmą

4.2. Prognozavimo rezultatų apibendrinimas

Praktiškai įsitikinta, kad efektyviausiai būsenas prognozuoja neuroninių tinklų algoritmas. Algoritmų efektyvumas tarpusavyje palygintas 4.6 diagramoje. Pateikiama, kokia dalis buvo teisingai suprognozuota, ir kaip pasiskirstė neteisingi atvejai. Optimistinė prognozė – realiai iškritusiems prognozuota sėkmė pabaigti studijas. Pesimistinė prognozė – sėkmingai pabaigusiems studijas prognozuotas iškritimas. Matome, kad su kiekvienu algoritmu gerėjant prognozavimo kokybei tolydžiai mažėja pesimistinių prognozių, tuo tarpu optimistinės išlieka apytiksliai stabilios. Naive Bayes algoritmo taikymas buvo mažiausiai efektyvus. Iš to išplaukia rekomendacija, kad jo geriau nenaudoti studentų būsenoms prognozuoti.

Grupavimo ir sprendimų medžio algoritmų panaudojimas gali būti papildomas, patvirtinti, arba paneigti neuroninio tinklo modeliu gautoms prognozėms.



4.6 pav. Algoritmų efektyvumo prognozavimui palyginimas

Rengiant panašaus pobūdžio prognozes, efektyvumą galima padidinti įtraukiant į algoritmo treniravimą ne tik pažymius, tačiau ir kitus atributus. Tai galėtų būti lytis, baigtos mokyklos tipas, bei vieta (miestas, gyvenvietė, kaimas...) bei kiti atributai. Efektyvumui teigiamos įtakos taip pat turėtų didesnis kiekis apmokymui skirtų duomenų. Tarkime, jeigu šiame darbe remiantis 2003 metais įstojusiais buvo prognozuota galimybė baigti studijas 2004 metais įstojusiems, tai į algoritmo apmokymą įtraukus dar ir stojusius į universitetą 2002 metais, prognozavimo kokybė turi gerėti.

5. IŠVADOS

1. Magistrinis darbas skirtas šiuo metu ypač aktualiai informacinių technologijų sričiai – verslo intelektikai (angl. Business Intelligence). Darbe suprojektuota ir eksperimentiškai išbandyta universiteto akademinių duomenų intelektualios analizės sistema tampa ypač svarbi reformuojant Lietuvos aukštojo mokslo sistemą, didinant jos efektyvumą ir gerinant studijų kokybę.

2. Išanalizuotas duomenų gavybos ciklas sėkmingai pritaikytas naujai sričiai - intelektualiai akademinių universiteto duomenų analizei. Projektavimui pasinaudota UML diagramomis.

3. Suprojektuota duomenų saugykla, užpildyta išvalytais dviejų kartų informatikos fakulteto studentų duomenimis. Įvairiapusei šių duomenų analizei pritaikyta Naive Bayes, grupavimo, sprendimų medžio ir neuroninių tinklų algoritmai. Analizė atlikta Microsoft SQL Server Business Intelligence Developement Studio įrankiu.

4. Naive Bayes ir grupavimo algoritmai akademinių universiteto duomenų analizei pasitarnavo išsamiausi. Sprendimų medžio, Naive Bayes ir grupavimo algoritmų pritaikymai parodė brandos atestatų atskirų dalykų įtakas studijų baigimo galimybei. Grupavimo algoritmas duomenų analizei pritaikytas dviem būdais: paduodant atributus kaip diskrečius ir kaip tolydinius dydžius, atitinkamai atskleidžiant skirtingas dydžių charakteristikas.

5. Realizuota galimybė prognozuoti būsimų studentų mokymosi universitete galimybes. Atliktas tyrimas parodė kad efektyviausiai studijų būsenas prognozuoja neuroninių tinklų algoritmas. Iš to išplaukia, kad būtent jį reikėtų naudoti analogiškoms užduotims spręsti.

6. Atlikta pirmojo semestro dalykų analizė atskleidė, kurie moduliai labiausiai įtakoja studentų iškritimą. Tam pasinaudota Naive Bayes ir grupavimo algoritmais. Analogiškai galima įvertinti pasirinktojo kito semestro, metų arba visų studijų dalykų teigiamas ir neigiamas įtakas studijų baigimui.

7. Sukurta duomenų analizės metodika gali pasitarnauti studijų proceso gerinimui. Tą galima pasiekti įvertinant universitete dėstomų dalykų charakteristikas. Kitas aspektas – galima teikti rekomendacijas būsimiems studentams, dėl jų studijavimo potencialo, taikant būsenų prognozes. Sistemą nesunkiai galima adaptuoti ir kitų fakultetų duomenų analizei.

6. NAUDOTA LITERATŪRA

1. Daniel T. Larose - ,,Discovering Knowledge in Data. An Introduction to Data Mining" Wiley – Interscience, 2005;
2. ,,Oracle Data Mining 11g - Know More, Do More, Spend Less" – An Oracle White Paper, 2007;
3. ZhaoHui Tang and Jamie MacLennan - ,,Data Mining with SQL Server 2005" Wiley, 2005;
4. „SAS 9.2 – Transforming your business with Enterprise Intelligence" http://www.sas.com/software/sas9/index.html (2009.02.20);
5. ,,Clementine – What's New" http://www.spss.com/Clementine/whats_new.htm (2009.02.20);
6. ,, Integrate InfoSphere Warehouse data mining with IBM Cognos reporting" http://www.ibm.com/developerworks/data/library/techarticle/dm-0810wurst/index.html (2009.02.21);
7. ,,Dundas Data Visualisation" http://www.dundas.com/Company/Media/PressSQL2008.aspx (2009.02.22);
8. ,, Data Mining with Open Source Machine Learning Software" http://www.cs.waikato.ac.nz/ml/weka/ (2009.02.23);
9. ,,Orange Data Mining" http://www.ailab.si/orange/ (2009.02.23);
10. Sprendimų medžio algoritmas - MSDN. http://msdn2.microsoft.com/en-us/library/ms175312.aspx (2009.02.23);
11. Grupavimo algoritmas- MSDN http://msdn2.microsoft.com/en-us/library/ms174879.aspx (2009.02.25);
12. Nayve – Bayes algoritmas - MSDN http://msdn2.microsoft.com/en-us/library/ms174806.aspx (2009.02.25) ;
13. Asociacijų algoritmas- MSDN http://msdn2.microsoft.com/en-us/library/ms174916.aspx (2009.02.25);
14. Sekų grupavimo algoritmas – MSDN http://msdn2.microsoft.com/en-us/library/ms175462.aspx (2009.02.25);
15. „Time Series" algoritmas - MSDN http://msdn2.microsoft.com/en-us/library/ms174923.aspx (2009.02.25);

16. Neuroninio tinklo algoritmas – MSDN http://msdn2.microsoft.com/en-us/library/ms174941.aspx (2009.02.25);
17. Logaritminės regresijos algoritmas – MSDN http://msdn2.microsoft.com/en-us/library/ms174828.aspx (2009.02.25);
18. Tiesinės regresijos algoritmas – MSDN http://msdn2.microsoft.com/en-us/library/ms174824.aspx (2009.02.25);
19. Rafal Lukawiecki - ,,Finding Hidden Intelligence with Predictive Analysis of Data Mining" Seminaro medžiaga.
20. ,,Data Mining Extentions (DMX) reference". http://technet.microsoft.com/en-us/library/ms132058.aspx (2009.05.15).

7. TERMINŲ IR SANTRUMPŲ ŽODYNAS

DMX – (angl. Data Mining Extention) Microsoft SQL Server DBVS palaikoma užklausų kalba, skirta duomenų gavybai ir prognozių rengimui.

DB – Duomenų bazė.

OLAP - (angl. Online Analytical Processing) DB technologija tiesioginei duomenų analizei.

UML – (angl. Unified Modelling Language) – Unifikuota modeliavimo kalba sistemų ir procesų projektavimui.

Duomenų gavybos struktūra - (angl. Mining Structure) - duomenų analizės blokas su iš anksto apibrėžtais įėjimais, ir laukiamais išėjimais.

PRIEDAI

1. Būsenų prognozės DMX užklausa Neuroninio tinklo algoritmui (standartiniai pažymiai):

```
SELECT
  t.[VIDKO],
  (t.[Busena]) as [reali_busena],
  ([Neuro_prog].[Busena]) as [Neuro_busena],
  (PredictProbability([Neuro_prog].[Busena])) as [busenos_tikimybe]
From
  [Neuro_prog]
PREDICTION JOIN
  SHAPE {
  OPENQUERY([Mag Stud14],
    'SELECT
      [VIDKO],
      [Busena]
    FROM
      (SELECT     VIDKO, Busena, IST_MET
FROM         dbo.STUDENTAI
WHERE     (IST_MET = ''2004'')) as [Stud2004]
    ORDER BY
      [VIDKO]')}
  APPEND
  ({OPENQUERY([Mag Stud14],
    'SELECT
      [DALYKAS],
      [PAZYMYS],
      [VIDKO]
    FROM
      [dbo].[STOJIMO0304]
    ORDER BY
      [VIDKO]')}
    RELATE
      [VIDKO] TO [VIDKO])
    AS
      [STOJIMO0304] AS t
ON
  [Neuro_prog].[Busena] = t.[Busena] AND
  [Neuro_prog].[STOJIMO0304].[DALYKAS] = t.[STOJIMO0304].[DALYKAS] AND
       [Neuro_prog].[STOJIMO0304].[PAZYMYS] = t.[STOJIMO0304].[PAZYMYS]
```

2. Būsenų prognozės DMX užklausa Neuroninio tinklo algoritmui, esant diskretizuotiems pažymiams:

```
SELECT
  t.[VIDKO],
  (t.[Busena]) as [Reali_busena],
  ([Neuro_prog].[Busena]) as [Neuro_busena],
  (PredictProbability([Neuro_prog].[Busena])) as [Neuro_tikimybe]
From
  [Neuro_prog]
PREDICTION JOIN
  SHAPE {
  OPENQUERY([Mag Stud14],
    'SELECT
      [VIDKO],
      [Busena]
    FROM
      (SELECT     VIDKO, Busena, IST_MET
FROM         dbo.STUDENTAI
WHERE     (IST_MET = ''2004'')) as [Stud2004]
    ORDER BY
      [VIDKO]')}
  APPEND
  ({OPENQUERY([Mag Stud14],
    'SELECT
      [DALYKAS],
      (CASE
 WHEN [TIPAS] = ''mokyklinis egz.''  THEN [PAZYMYS]
 WHEN [TIPAS] = ''metinis''  THEN [PAZYMYS]
 WHEN [PAZYMYS] > 100   THEN 0
 WHEN [PAZYMYS] <= 14 AND tipas=''valstybinis egz.''  THEN ''1''
 WHEN [PAZYMYS] <= 24 AND tipas=''valstybinis egz.''  THEN ''2''
 WHEN [PAZYMYS] <= 34 AND tipas=''valstybinis egz.''  THEN ''3''
 WHEN [PAZYMYS] <= 44 AND tipas=''valstybinis egz.''  THEN ''4''
 WHEN [PAZYMYS] <= 54 AND tipas=''valstybinis egz.''  THEN ''5''
 WHEN [PAZYMYS] <= 64 AND tipas=''valstybinis egz.''  THEN ''6''
 WHEN [PAZYMYS] <= 74 AND tipas=''valstybinis egz.''  THEN ''7''
 WHEN [PAZYMYS] <= 84 AND tipas=''valstybinis egz.''  THEN ''8''
 WHEN [PAZYMYS] <= 94 AND tipas=''valstybinis egz.''  THEN ''9''
 WHEN [PAZYMYS] <=100 AND tipas=''valstybinis egz.''  THEN ''10''
END) AS [Diskret_Paz],
      [VIDKO]
    FROM
      [dbo].[STOJIMO0304]
    ORDER BY
      [VIDKO]')}
    RELATE
      [VIDKO] TO [VIDKO])
    AS
      [STOJIMO0304] AS t
ON
  [Neuro_prog].[Busena] = t.[Busena] AND
  [Neuro_prog].[STOJIMO0304].[DALYKAS] = t.[STOJIMO0304].[DALYKAS] AND
  [Neuro_prog].[STOJIMO0304].[Diskret Paz] =
t.[STOJIMO0304].[Diskret_Paz]
```