KAUNO TECHNOLOGIJOS UNIVERSITETAS

MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS

Laimonas Andriulis

JURIDINIŲ ASMENŲ KREDITO RIZIKOS MODELIAVIMAS: BANKO KLIENTŲ ATVEJIS

Baigiamasis magistro projektas

Vadovai
Doc. dr. Aura Drakšaitė
Lekt. dr. Evaldas Vaičiukynas

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS

MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS

JURIDINIŲ ASMENŲ KREDITO RIZIKOS MODELIAVIMAS: BANKO KLIENTŲ ATVEJIS

Baigiamasis magistro projektas

Didžiųjų verslo duomenų analitika (621G12002)

Vadovai

Doc. dr. Aura Drakšaitė

Lekt. dr. Evaldas Vaičiukynas

Recenzentai

Lekt. dr. Lina Sinevičienė

Lekt. dr. Mindaugas Kavaliauskas

Projektą atliko

Laimonas Andriulis

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Matematikos ir gamtos mokslų fakultetas
(Fakultetas)

Laimonas Andriulis
(Studento vardas, pavardė)

Didžiųjų verslo duomenų analitika, 621G12002
(Studijų programos pavadinimas, kodas)

„JURIDINIŲ ASMENŲ KREDITO RIZIKOS MODELIAVIMAS: BANKO KLIENTŲ ATVEJIS“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 ___ m. _____ ____ d.
Kaunas

Patvirtinu, kad mano, **Laimono Andriulio**, baigiamasis projektas tema „JURIDINIŲ ASMENŲ KREDITO RIZIKOS MODELIAVIMAS: BANKO KLIENTŲ ATVEJIS“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

_____ _____
(vardą ir pavardę įrašyti ranka) *(parašas)*

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

Andriulis, Laimonas. Credit Risk Scoring For Legal Entities: The Case Of Bank Customers: / Master‘s thesis in Business Big Data Analytics / supervisors assoc. prof. Aura Drakšaitė, lect. Evaldas Vaičiukynas. The Faculty of Mathematics and Natural Sciences, Kaunas University of Technology.

Research area and field: Mathematics, physical science

Key words: Probability of default, bankruptcy, machine learning

Kaunas, 2017. 73 p.

SUMMARY

Small and medium enterprises take important place in nowadays economics, yet due to dynamic market they are often considered to be on high risk to go bankrupt. Therefore, to guarantee functioning of an enterprise a strong position in financial safety is essential. To be fully prepared for all planned expanses enterprises are willing to take loans from banks. Banks wants to lower the risk of giving loans to companies that will not be able to pay their payments on time or will go bankrupt.

This paper is going to investigate the issue of enterpresis going bankrupt by creating model that would predict probability of default based on behavior data. The main objectives of this paper are: define the bakruptcy of small and medium enterprises problem and review ralated works to gather knowledge about used methods to predic probability of default an lastly create a binary classification model for bankruptcy of enterprise.

Models were built on one of Scandinavian bank‘s behavior and transactions history data. Three classification algorithms (random forest, neural networs, xgboost) were tested using 182 variable. Best model was obtained by using xgboost algorithm with 5 fold cross validation with the result AUC = 0,817, while random forest‘s AUC = 0,763, neural network‘s AUC = 0,69. After investigating most important variables extracted from best performed model there can be made a conclusion that out of 182 variables related with costumers behavior, dutifulness for paying payments for their loans are more important and informative than behavior with money compared with liabilities and income.

Andriulis, Laimonas. Juridinių asmenų kredito rizikos įvertinimas: banko klientų atvejis. / Magistro baigiamasis projektas / vadovai doc. dr. Aura Drakšaitė, lekt. Dr. Evaldas Vaičiukynas; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

Mokslo kryptis ir sritis: Matematika, gamtos mokslai

Reikšminiai žodžiai: *bankroto tikimybė, mašininis mokymasis.*

Kaunas, 2017. 73 p.

SANTRAUKA

Mažos ir vidutinės įmonės yra svarbi ekonomikos dalis, kuri dėl didelės dinamikos rinkoje yra laikoma sąlyginai rizikinga bankroto klausimu. Todėl, kad įmonė garantuotų savo atliekamų funkcijų vykdymą labai reikalingas finansinis saugumas. Kad įmonė būtų pasiruošusi visoms planuotoms išlaidoms ji yra pasiryžusi imti paskolas iš bankų. O tuo tarpu, bankai norėdami išvengti rizikingų klientų, kurie negalės sumokėti visos paskolos arba skelbs bankrotą, turi gebėti įvertinti kiekvieno kliento bankroto tikimybę.

Šiame darbe tiriama įmonių bankroto problema, kuriant modelį, kuris prognozuotų bankroto tikimybę naudojantis banko pateiktais elgsenos duomenimis. Pagrindiniai darbo uždaviniai yra: apibrėžti mažų ir vidutinių įmonių bankroto problemą ir padaryti susijusių darbų literatūros apžvalgą ir surinkus žinias išskirti metodus, kuriais būtų galima sukurti bankroto įvykio klasifikavimo modelį.

Modeliai kurti naudojantis Skandinavijos banko elgsenos ir įmonių atliekamų įmokų istoriniais duomenimis. Panaudoti trys klasifikavimo algoritmai (atsitiktinis miškas, neuroniniai tinklai, ekstremalaus gradientinio sustiprinimo medžių algoritmas) buvo išbandyti naudojant 182 kintamąjį. Atliekant 5 dalių kryžminę patikrą, geriausias modelis buvo sudarytas naudojant ekstremalaus gradientinio sustiprinimo medžių algoritmą, kurio AUC įvertis lygus 0,817, lyginant su atsitiktiniu mišku (AUC = 0,763) ir neuroniniais tinklais (AUC = 0,69). Po svarbiausių kintamųjų, kurie buvo naudojami geriausio modelio sudarymo metu, analizės daromos išvados, kad iš 182 kintamųjų susijusių su klientų elgsena, įmokų mokėjimo pareigingumas yra svarbesnis ir informatyvesnis nei įmonės elgsena su turimais finansais lyginant su turimais įsipareigojimais ir įplaukomis.

ĮŽANGA

Savo veikloje, įmonės neišvengiamai susiduria su įvairiomis rizikos rūšimis. Norint užtikrinti įmonės veiklos stabilumą, būtina gebėti objektyviai įvertinti esamą situaciją rinkoje ir visus gresiančius pavojus. Šiais, greitų pokyčių, laikais nesvarbu kokią strategiją įmonė pasirinks norėdama išsilaikyti rinkoje, neišvengiamai tam reikės piniginių resursų. Tokiu atveju įmonės dažnai kreipiasi į komercinius bankus ir ryžtasi įsiskolinti, kad galėtų garantuoti savo veiklos tęstinumą.

Kitoje šio, įmones gelbėjančio, sandorio pusėje yra komerciniai bankai (toliau bankai), kurie yra suinteresuoti sudaryti sutartis su sėkmingai veiklą vykdančiomis įmonėmis, kurios visą paimto kredito mokėjimo laikotarpį būtų mokios. Todėl bankams svarbu įvertinti potencialių ir esamų skolininkų būklę. Bankui svarbiausia, ar įmonė galės sumokėti įnašus nurodytais laikotarpiais ir ar nebankrutuos negrąžinusi viso kredito. Todėl bankui aktualu gebėti prognozuoti įsipareigojimų nevykdymo (nemokumo arba bankroto) tikimybę (angl. probability of default).

Bankai kaupia įmonių (ir banko su įmonėmis) elgsenos duomenis, todėl gali atlikti kompleksinį bankroto tikimybės vertinimą, įvertinant ne tik juridinio asmens finansinius veiklos rodiklius, bet ir jų elgesį kredito mokėjimo metu. Šioje vietoje susiduriama su keliomis didžiųjų duomenų savybėmis: didelė įvairovė ir dideli duomenų kiekiai.

Objektas – mažų ir vidutinių įmonių bankroto tikimybės prognozavimo modelis.

Darbo tikslas – naudojantis banko elgsenos su paskolas paėmusiomis mažomis ir vidutinėmis įmonėmis ir įmonių atliktų transakcijų duomenis sukurti bankroto tikimybės prognozavimo modelį.

Darbo uždaviniai:

1. Atlikti mažų ir vidutinių įmonių bankroto problematikos analizę ir bankroto tikimybės prognozavimo naudotinų metodų ir tų metodų pritaikymo elgsenos duomenų atveju literatūros analizę;
2. Parinkti ir aprašyti šiame darbe naudojamus mašininio mokymosi metodus;
3. Parengti duomenų rinkinį modeliavimui ir, apmokant mašininio mokymosi algoritmus, sukurti kelis bankroto tikimybės prognozavimo modelius, bei išrinkti geriausią.

Šiame darbe apžvelgiami tiek klasikiniai, tiek pasinaudojus mašininio mokymosi metodus sukurti bankroto tikimybės prognozavimo modeliai, jų skirtumai, trūkumai ir pranašumai. Naudojantis Skandinavijos banko elgsenos duomenimis ir mašininio mokymosi algoritmais

kuriami bankroto tikimybės prognozavimo modeliai, interpretuojami ir lyginami jų rezultatai. Darbo rezultatai buvo pristatyti Kauno technologijos universiteto Matematikos ir gamtos mokslų fakulteto organizuojamoje 15-toje studentų konferencijoje „Matematika ir gamtos mokslai: teorija ir taikymas“. Taip pat, tyrimo eiga ir rezultatai buvo pristatyti Skandinavijos banko organizuojamo duomenų analizės konkurso metu, kaip bankui aktualios problemos sprendimo būdas.

1. ĮMONIŲ BANKROTO PROGNOZAVIMO PROBLEMA IR SUSIJUSIŲ DARBŲ LITERATŪROS APŽVALGA

Šiame skyriuje apibrėžiama ekonominė mažų ir vidutinių įmonių bankroto rizika ir jo tikimybės prognozavimo svarbumas. Susipažinus su problema atliekama susijusių darbų literatūros apžvalga, kur išskiriami klasikinių bankroto prognozavimo metodų taikymo privalumai ir trūkumai, taip pat vis dažniau literatūroje minimų mašininio mokymosi metodų taikymo pavyzdžiai, kurie leidžia atsižvelgti į didesnę duomenų įvairovę prognozuojant bankroto įvykius. Skyriuje aprašomas įvairių duomenų panaudojimas kuriant prognozavimo modelius. Taip pat skyriuje apžvelgiamos mašininio mokymosi algoritmų taikymo metodologijos, kurios pritaikomos ir šiame darbe analizuojamai problemai.

1.1. Kredito rizikos vertinimo problema

Mažos ir vidutinės įmonės yra didelė ekonomikos varomoji jėga kuriamomis darbo vietomis, kuriamų produktų kiekiu, suteikiamomis paslaugomis ar sumokamais mokesčiais. Pagal Europos Komisiją, vidutinėms įmonėms priskiriamos įmonės, turinčios mažiau nei 250 darbuotojų ir įmonės metinės pajamos neviršija 50 mln. eurų arba įmonės balanse nurodyto turto vertė neviršija 43 mln. Eurų (1). Šio darbo tiriamojoje imtyje yra Skandinavijos įmonės, neviršijančios šių rodiklių – vidutinės, mažos ir labai mažos įmonės (šiame darbe toliau MVĮ). Europos Sąjungos šalyse mažos ir vidutinės įmonės sudaro daugiau kaip 99 % visų įmonių ir steigia 67 % pilno etato darbo vietų. Taip pat, tokios įmonės sukuria apie 57 % bendrosios pridėtinės vertės (BPV) Europos Sąjungoje (2).

Kadangi tokių įmonių yra labai daug, jos dažnai taiko į nedidelį vartotojų rinkos segmentą ir neišvengia konkurencijos, mažoms ir vidutinėms įmonėms sunku įsitvirtinti rinkoje su keliomis paslaugomis ar keliais produktais. Taip pat, tokios įmonės dažniausiai yra užima vieną ir nedidelį geografinį regioną. Tokios įmonės dažniausiai yra silpnesnės finansine prasme ir labiau priklausomos nuo finansinės būklės. Dažniausiai bankrotas skelbiamas dėl vienos iš dviejų priežasčių: arba einamuoju laikotarpiu pritrūksta apyvartinių lėšų, kurios gal būti skirtos susimokėti įmoką turimam kreditiniam įsiskolinimui padengti arba įmonės įsipareigojimai viršija įmonės turtą (3). Tokias situacijas gali iššaukti tiek vidiniai tiek išoriniai veiksniai. Tokių veiksmų apibendrinimas pateiktas J. Karalevičienės ir R. Bužinskienės 2012 metų straipsnyje (4) remiantis autorių I. A. Blanko (1999), J. Mackevičiaus ir A. Rakštelienės (2005), J. Mackevičiaus (2007), S. Liučvaičio (2003) ir K. Garškaitė (2002) darbais. Apibendrinus šiuos šaltinius prie vidinių, bankrotą sukeliančių, veiksnių beveik visi autoriai priskiria prastą,

neprofesionalų vadovavimą, įmonės strategijų parinkimą, ypač srityse tiesiogiai susijusiose su finansine veikla (neefektyvus finansų valdymas, didelė skolinto kapitalo dalis ir tokių įsipareigojimų didinimas), investicijomis (užtęsti produkto tobulinimo darbai, nepasiteisinanti marketingo strategija ar kitos prastos investavimo strategijos) ar neapskaičiuotu išteklių naudojimu (neefektyvūs gamybos planai). Išoriniai veiksniai taip pat gali būti skirstomi į kelias grupes – ekonominius veiksnius (šalies bendro vystymosi veiksniai: BVP mažėjimas, infliacijos augimas, mokesčių sistemos nestabilumas, šešėlinės ekonomikos didėjimas, nedarbas, gyventojų pajamos) ir politinę arba socialinę padėti nusakančius veiksnius (demografinė situacija, santykiai su kaimyninėmis šalimis, kintantys mokesčių įstatymai, dviprasmiški teisiniai aktai).

S. Liučvaitis 2003 (5) teigia, kad jeigu šalyse, kuriose situacija ekonomine ir politine prasme yra stabili, sėkminga įmonės veikla priklauso tik vienu trečdaliu nuo išorinių veiksnių, kurių įmonė nepakeis, todėl didžioji dalis dėmesio siekiant užtikrinti įmonės veiklą ir norint išvengti bankroto tenka vidiniams įmonės procesams, kurie jau buvo apibendrinti į vadovavimo problemas, investicines klaidas ir prastą išteklių panaudojimą. Du pastarieji veiksniai neatsiejami nuo piniginiu išteklių. Visoms mažoms ir vidutinėms įmonėms, ypač naujoms, viena pagrindinių problemų tampa kapitalas, naudojamas įsitvirtinti rinkoje, atlikti pradines investicijas į produktą ar paslaugą. Vieni pagrindinių pajamų šaltinių yra bankai ir jų teikiamos paskolos. Tačiau, žinant sunkumus, su kuriais susiduria MVĮ, atsiranda kredito rizika ir jeigu ji laiku nepašalinama – gali privesti įmonę prie bankroto, dėl šios priežasties bankai rizikuoja duodami paskolas ir turi apsidrausti, numatyti kam ir kada duoti paskolą.

Juridinių asmenų kreditingumo rizikai nusakyti galima rasti daug apibrėžimų. Šiame darbe kredito rizika laikoma rizika susijusi su bet kokiais kreditavimo teisiniais santykiais, tokiais, kaip kredito kokybinių savybių pokyčiu (kredito reitingo kritimu), rizikai atsiradusiai dėl pakitusios ekonominės aplinkos ar įmonės valdymo ar bankroto atveju (6). Šiame darbe tiriamas paskutinis atvejis, bankroto (įsipareigojimų nevykdymo) rizika. Lietuvos Respublikos įmonių bankroto įstatyme (7) tai apibrėžiama kaip „teisės aktų nustatyta tvarka pripažinta įmonės nemokumo būsena, kai siekiama šios būsenos pabaigos iš įmonės turto tenkinant kreditorių reikalavimus ir užtikrinant kreditorių ir įmonės interesų pusiausvyrą“. Šios rizikos tikėtinumo įvertinimui naudojama įsipareigojimų nevykdymo (nemokumo arba bankroto) tikimybė, kuri apibrėžiama kaip, tikimybė, kad viena sutarties šalis nevykdys įsipareigojimų.

Bankrutavusi įmonė, bent kurį laiką tarpą, turi stabdyti savo veiklą, taip panaikindama sukurtas darbo vietas ir griaudama rinkos ciklus, kurios dalyvė buvo, taip padidinama bankroto tikimybė įmonėms, kurios buvo bent dalinai priklausomos nuo bankrutavusios. Mažos ar vidutinės įmonės bankrotas leidžia užimti didesnę rinkos dalį konkuruojančioms įmonėms, taip

atsiranda galimybė kelti produkcijos ar paslaugos kainas regione. Tokie išoriniai veiksniai gali turėti įtaką kitų įmonių veiklai ir finansinei situacijai. Kadangi mažos ir vidutinės įmonės kuria apie du trečdalius darbo vietų, tai gali vesti prie dar didesnio nedarbo lygio, demografinių pokyčių ar kitų veiksnių veikiančių šalies bendro vidaus produkto lygį. Turint omenyje išvardintas galimas problemas ir norint jų išvengti įmonės bankroto rizika turi būti įvertinama kuo anksčiau.

Dažnai literatūroje įsipareigojimų nevykdymo tikimybės prognozavimo, įvertinimo metodai skirstomi į dvi grupes: klasikinius ir modernius, mašininio mokymosi pagrindu, sukurtus modelius.

Pirmajai grupei priskirti modeliai dažnai remiasi įmonių veiklos finansiniais duomenimis ir naudojant juos išvestais rodikliais. Tai gan stipriai apriboja naudingos informacijos kiekį ir šiais, didžiųjų duomenų laikais, kai informacijos įvairumas dažnai lemia ir gaunamo rezultato tikslumą, galima galvoti kaip apie didelį trūkumą. Taip pat prie apribojimų prisideda ir klasikiniams metodams būdingos kompaktiškos, nesudėtingos matematinės išraiškos, kurių sudarymas ir naudojimas yra labai paprastas, tačiau sunkiai adaptuojamos prie greitai kintančios informacijos, kuria nusakoma rinkos situacija.

Antrajai grupei priskiriami mažiau apriboti ir dinamiški modeliai, kurie gali įgauti sudėtingesnes, nei daugiamatės diskriminantinės analizės išraiškas, todėl dažnu atveju tokie modeliai laikomi „juodosiomis dėžėmis“, kur visa matematinė logika, pagal kurią atliekami skaičiavimai naudojant pradinę informaciją, yra sunkiai interpretuojama. Tačiau naudojant šiuos metodus galima laisviau žiūrėti į naudojamus rodiklius, eksperimentuoti pridedant papildomos skaitinės, kategorinės ar tekstinės informacijos. Todėl galima panaudoti šių laikų duomenų įvairovę, taip pat adaptuotis prie greitų situacijos pokyčių iš naujo apmokant mašininio mokymosi algoritmus.

Šiame skyriuje aptariami būdai, kuriais galima apskaičiuoti bankroto tikimybę. Nusakomos klasikinių ir modernių, mašininio mokymosi algoritmais pagrįstų, metodų pagrindinės prielaidos, privalumai ir trūkumai.

1.2. Klasikiniai bankroto tikimybės vertinimo metodai

Maždaug XX a. antroje pusėje pradėti kurti klasikiniai nemokumo vertinimo modeliai, paremti finansinėse ataskaitose teikiama informacija. Todėl daugelis mokslininkų bankroto tikimybės vertinimą sieja su finansinės analizės metodikomis, kurių sudarymui naudojami finansinės analizės metodai, daugiamatė diskriminantinė analizė (MDA), regresijos modeliai (8). Bendra ir dažniausiai pasitaikanti tokių modelių išraiška:

$$Z = \mathrm{a} + b_1 \cdot X_2 + b_2 \cdot X_3 + \cdots + b_n \cdot X_n \quad \textbf{(1.1)}$$

čia:

Z – indekso reikšmė;

a – konstanta;

X_n – finansinis (nepriklausomas) kintamasis;

b_i – diskriminanto koeficientas, nusakantis ryšį tarp finansinio kintamojo ir indekso reikšmės. T.y. X_i kintamajam pakitus per 1 vienetą, indekso Z reikšmė pakinta dydžiu b_i.

Pirmuoju bankroto modeliu laikomas daugiamatės diskriminantinės analizės klasifikacijos metodu sudarytas modelis, kurį 1986 metais sukūrė E. I. Altmanas naudodamas 5 finansinių rodiklių santykinius dydžius, išreikštus naudojant (1) lygtį (9). Modelis buvo sukurtas turint 66 įmonių įrašus, kur pusė imties buvo duomenys apie bankrutavusias įmones ir pusė imties – duomenys apie nebankrutavusias. Taip pat, iš pradžių buvo naudojami 22 finansiniai kintamieji, tačiau pabaigoje tik 5.

Skirtinguose sektoriuose įmonės gyvavimas gali priklausyti nuo skirtingų veiksnių ir aplinkybių, todėl gali kilti įtarimas dėl fiksuotų finansinių rodiklių ir prie jų esančių koeficientų universalumo. Todėl atsiranda vis naujesnių mokslinių publikacijų tiek tiriant tą patį Altmano Z įverčio modelį, tiek jo modifikacijas, kurios taip pat paremtos diskriminantinės analizės metodu sudaryta lygtimi iš finansinių (nepriklausomų) kintamųjų.

1977 metais mokslininkai Taffleris ir Tisshawas tęsdami diskriminantinės tiesinės analizės metodo taikymą bankroto tikimybei įvertinti išanalizavo 46 bankrutavusių ir 46 veikiančių įmonių 80 finansinių rodiklių reikšmių iš kurių atsirinko 4 rodiklius ir sudarė dar vieną klasifikacijos modelį pateiktą 1.1 lentelėje.

Ilgai buvo manyta, kad MDA modeliavimas yra vienas tinkamiausių bankroto tikimybės prognozavimo atveju. Tačiau MDA metodo prielaidos dažnai pažeidžiamos realių duomenų atveju, todėl reikėjo atlikti tyrimus naudojant kitus metodus. Vieni pirmųjų logistinės regresijos modelių taikymų bankroto tikimybei buvo atlikti Chesserio (1974), Ohlsono (1980) ir Zavgre (1985) naudojant Z įvertį, gautą naudojant tiesinę funkciją ir tuomet tikimybę apskaičiuojant pagal išraišką:

$$\mathrm{P} = \frac{1}{(1 + e^{-Z})} \quad \textbf{(1.2)}$$

čia:

P – bankroto tikimybė intervale [0,1];

e ≈ 2.71828.

Pirmieji, klasikiniai bankroto prognozavimo modeliai sukurti naudojant MDA ir logistinės regresijos metodus pateikti 1.1 lentelėje.

1.1 lentelė. Klasikiniai bankroto prognozavimo modeliai

Autorius	**Modelis**	**Modelio elementai**
Altman	$Z = 1{,}2 \cdot X_1 + 1{,}4 \cdot X_2 + 3{,}3 \cdot X_3 + 0{,}6 \cdot X_4 + 0{,}999 \cdot X_5$ Bankroto tikimybė maža, kai Z > 2,9; Bankrotas galimas, kai 1,8 < Z ≤ 2,9; Bankroto tikimybė didelė, kai Z ≤ 1,8.	X_1 – apyvartinis turtas / turtas, X_2 – nepaskirstytas pelnas / turtas, X_3 – pelnas prieš apmokestinant / turtas, X_4 – akcinio kapitalo rinkos kaina / įsipareigojimai, X_5 – pardavimų pajamos / turtas.
Taffler ir Tisshaw	$Z = 0{,}53 \cdot X_1 + 0{,}13 \cdot X_2 + 0{,}18 \cdot X_3 + 0{,}16 \cdot X_4$ Bankroto tikimybė maža, kai Z > 0,3; Bankrotas galimas, kai 0,2 < Z ≤ 0,3; Bankroto tikimybė didelė, kai Z ≤ 0,2.	X_1 – pelnas prieš apmokestinant / trumpalaikiai įsipareigojimai, X_2 – trumpalaikis turtas / įsipareigojimai, X_3 – trumpalaikiai įsipareigojimai / turtas, X_4 – (trumpalaikis turtas - trumpalaikiai įsipareigojimai) / veiklos sąnaudos,
Chesser	$Z = -2{,}0432 - 5{,}24 \cdot X_1 + 0{,}0053 \cdot X_2 + 6{,}65 \cdot X_3 + 4{,}4 \cdot X_4 - 0{,}079 \cdot X_5 - 0{,}102 \cdot X_6$ Bankroti tikimybė maža, kai P ≤ 0,5; Bankroto tikimybė didelė, kai P > 0,5.	X_1 – pinigai / turtas, X_2 – pardavimų pajamos / pinigai, X_3 – pelnas prieš apmokestinant / turtas, X_4 – įsipareigojimai / turtas, X_5 – ilgalaikis materialusis turtas / nuosavas kapitalas, X_6 – grynasis apyvartinis kapitalas / pardavimų pajamos.

Šaltinis: Rasa Budrikienė, Irena Paliulytė (10).

Visi minėti metodai buvo kuriami naudojant sąlyginai nedideles subalansuotų (bankroto ir ne bankroto klasėms priklausančių įrašų kiekiai skiriasi nežymiai) duomenų imtis, fiksuojant kintamųjų kiekius ir koeficientus prie jų.

Tokie metodai turi kelis privalumus:

1. Lengvai taikomi ir interpretuojami;
2. Lengvai diegiami esamose sistemose, skaičiavimai atliekami greitai ir nereikalauja daug resursų;
3. Ištestuoti laiko, patikimi;
4. Pritaikomi visoms įmonėms, nes naudojami viešai prieinami finansinių ataskaitų duomenys.

Kita vertus, šiais laikais greitai besikeičiančiame duomenų sraute galima rasti vis naujos, svarbios informacijos, kas neįvertinama klasikiniuose modeliuose. Apibendrintai galima išskirti pagrindinius klasikinių modelių trūkumus:

1. Fiksuoti kintamieji neleidžia panaudoti naujos informacijos, kuri gaunama kitokiais būdais ir yra kitokios struktūros;
2. Fiksuoti koeficientai nėra lankstūs tiriant skirtingomis veiklomis užsiimančias įmones, skirtingų ekonomikos lygių šalyse ar skirtingu laikotarpiu;
3. Modeliai buvo sudaryti naudojant subalansuotus duomenis, kurie neatitinka realios situacijos dabartyje;
4. Naudojami finansiniai rodikliai privalomai teikiami tik kartą per metus, dėl to tokias lygtis galima taikyti tik ilguoju laikotarpiu, t.y. jos nėra dinamiškos, prisitaikančios prie greitų rinkos pokyčių.

1.3. Mašininio mokymosi bankroto tikimybės vertinimo metodai

Besivystant kredito rizikos vertinimo metodams, atsiranda nauji ir sudėtingesni klausimai, kurie susiję ne tik su finansiniais rodikliais, bet ir informacinės technologijos klausimais, dideliais duomenų kiekiais, procesų optimizavimu, automatizavimu ir lankstumu. Sprendžiant iškilusias problemas buvo sukurti ir pradėti taikyti nauji kredito rizikos (bankroto tikimybės) mokslo srityje mašininio mokymosi algoritmai, kurie leidžia daugelį skaičiavimų atlikti greičiau ir, dažnu atveju, tiksliau. Pastaruoju metu šie algoritmai laikomi geresniais, už klasikinius (11), (12), (13). Šių metodų taikymas išsprendžia daugelį klasikinių metodų trūkumų, kurie paminėti 1.2 skyriuje, ir netgi tampa privalumais:

1. Leidžia naudoti įvairiapusišką informaciją, neapsiribojant tik retai teikiamų finansinių rodiklių duomenimis. Šiuose metoduose galima naudoti tiek kategorinius, tiek tekstinius kintamuosius, kurie nusakytų ne tik įmonės deklaruojamą finansinę būklę, bet ir jos elgseną. Kaip įmonė pozicionuoja save rinkoje t.y. kokius sprendimus atlieka rinkodaros srityje, kokią informaciją pateikia spaudoje. Taip pat, į šiuos duomenis galima reaguoti iš karto, nelaukiant finansinių metų pabaigos, kada paviešinamos įmonių finansinės ataskaitos;
2. Mašininio mokymosi algoritmai atlieka duomenų diferencijavimą pagal daugelį kriterijų. Kategoriniai kriterijai, kaip rinka, šalis, įmonės dydis gali lemti visiškai skirtingų faktorių svarbą įmonės bankrotui. Taip pat, metoduose turinčiuose derinamus reguliarizacijos parametrus ir susitvarkančiuose su duomenimis, turinčiais daug

dimensijų, nėra svarbūs pertekliniai kintamieji, todėl sutaupomas reikšmingų kintamųjų, kuriuos galima naudoti modelyje, paieškos laikas;

3. Dalis metodų (sudaryti remiantis sprendimų medžiais arba klasterizavimo teorija) gali sukurti modelius, kurie yra adaptuoti nesubalansuotiems duomenims. Priešingu atveju pasitelkus apmokymo imties modifikavimo metodus galima dirbtinai sudaryti imtį, kuri tiktų visiems mašininio mokymosi algoritmams ir nekeistų verslo duomenų. T.y šie metodai arba kelių metodų naudojimas vieno paskui kitą leidžia dirbti su realiais duomenimis tiesiogiai, be papildomos atrankos (14).

Šie nauji metodai leidžia kitaip įvertinti kredito rizikos problemas, tarp jų ir bankrotą. Pagrindinis tikslas – naudojantis istoriniais įvykiais surasti kriterijus, pagal kuriuos visus naujus įrašus galima suklasifikuoti į dvi grupes, pirmoji – įmonės, kurios panašios į tas, kurios jau bankrutavo, antroji – įmonės, kurios panašios į sėkmingai vykdančias savo įsipareigojimus pagal sutartis. Tam, kad atskirti, į kurią klasę turi patekti įrašas, metodai grąžina įrašo priklausymo kiekvienai grupei tikimybes, šią tikimybę randančio modelio sukūrimas yra šio darbo tikslas. Skirtingi klasifikavimo metodai grąžina ne vienodai pasiskirsčiusias tikimybes, todėl slenkstis, nuo kurio įrašą galima priskirti kuriai nors klasei, negali būti fiksuotas iš anksto. Dėl šių priežasčių, šiame darbe modelio galimybės klasifikuoti įrašus vertinimui naudojamas ROC (angl. receiver operating curve) grafikas ir ploto po ROC kreive AUC (angl. area under the curve) metrika, identifikavimo procentine paklaida.

Naudojantis šiomis metrikomis kredito rizikų vertinimo srityje atlikta ir aprašyta nemažai bandymų. Tokių bandymų sąrašas pateiktas Stefan Lessmann 2015 metų straipsnio literatūros analizėje (15), kurioje galima pastebėti, kad kiekvienu atveju bandymai daromi su skirtingais duomenų rinkiniais ir skirtingais kintamaisiais, todėl vienas modelis su fiksuotais kintamaisiais, kaip tai buvo daroma klasikiniu atveju, nėra universalus ir ypač šiais, skirtingų tendencijų rinkose, laikais reikėtų rinktis metodus, kuriais modeliai gali būti kuriami išlaikant kiekvienos tiriamos grupės specifiškumą, unikalumą. Taip pat rezultatai yra geri ir tokie tyrimai pasiteisina, nepaisant kai kurių praleidžiamų klasikinės duomenų analizės žingsnių – prielaidų duomenims tikrinimo, ar duomenų disbalanso tvarkymo prieš modeliavimą. Toliau apžvelgsime populiariausius mašininio mokymosi metodus, su kuriais buvo atlikti tyrimai bankroto tikimybės prognozavimui.

Neuroniniai tinklai (angl. neural networks) – mašininio mokymosi algoritmas, kurį sudaro nurodytas skaičius sluoksnių ir mazgų kiekviename sluoksnyje. Įėjimo signalai pereidami per visus mazgus su numatytais arba patikslintais svoriais yra modifikuojami ir tinklo pabaigoje atliekamas paskutinis agregavimas, norint gauti vieną, paskutinę, reikšmę – prognozę. A. Velido

ir kt. (12) literatūros apžvalgoje atlikta neuroninių tinklų ir klasikinių metodų palyginimų apžvalga ir išvados, kuriose paminima, kad šie algoritmai dažniau veikia geriau nei klasika, yra gan svarūs neuroninių tinklų taikymo argumentas.

Vienas pirmųjų neuroninių tinklų taikymų bankroto tikimybei rasti buvo pristatytas 1990 metais Marcuso D. Odomo ir Ramesho Shardo (16), kur autoriai naudodami Altmano atrinktais kintamaisiais aprašytais 1.1 lentelėje apmoko neuroninio tinklo, su vienu paslėptuoju sluoksniu ir 5 mazgais jame, algoritmą ir palygina rezultatus su Altmano siūlomu MDA metodu gautomis prognozėmis. Bandymų metu buvo naudojami skirtingi bankroto ir ne bankroto įvykiu disbalansai. MDA modelio padarytos klaidos klasifikuojant svyravo nuo 21,9 % iki 31 %, tuo tarpu neuroninių tinklų modelio klaidos klasifikuojant pateko į intervalą nuo 18,1 % iki 21,8 %. Taip pat, neuroniniai tinklai pastebimai geriau klasifikavo apmokyti naudojant nesubalansuotus duomenis (bankroto ir ne bankroto įvykiai imtyje atitiko 1:9 santykį), gautos paklaidos: 18,25 % neuroninių tinklų atveju ir 31 % MDA atveju. Pamela K. Coasts ir Franklinas Fantas (17) taip pat lygino neuroninių tinklų algoritmą su MDA, prognozės buvo kuriamos ilgajam laikotarpiui – trims metams ir trumpajam – iki vienerių metų. MDA metodo klasifikavimo tikslumas buvo intervale nuo 83,7 % iki 87.9 %, neuroninių tinklų tikslumas svyravo intervale nuo 81,9 % iki 95 %. Šiuo atveju, neuroninių tinklų metodas pasirodė taip pat geriau.

K. Tamas ir M. Kiang (18) bankų bankroto atvejų tyrimui palygino kelis metodus: anksčiau minėtą MDA, logistinę regresiją, k artimiausių kaimynų, sprendimų medžių metodą bei vieno sluoksnių ir kelių sluoksnių neuroninius tinklus. Naudojant šiuos metodus geriausiai pasirodė kelių sluoksnių neuroninis tinklas buvo tiksliausias. Taip pat pastabėta, kad neuroniniai tinklai ir sprendimų medžių klasifikavimo algoritmas pasirodė geriau nei visi kiti minėti.

Kitas mašininio mokymosi algoritmas, kurio pagalba stengiamasi spręsti bankroto tikimybės prognozės uždavinius – atsitiktinis miškas. Šis algoritmas paremtas taisyklių, kurių pagalba apmokymo duomenų imtis dalinama į dalis pagal tam tikrus kriterijus. Atsitiktinis miškas yra sudarytas iš sprendimų medžių t.y. iš silpnesnių klasifikatorių. Kinijos autorių kolektyvas (Gang Wang, Jian Ma, Lihua Huang, Kaiquan Xu) 2012 metais atliko bandymus palygindami vieno sprendimų medžio ir kelių atsitiktinių miškų modifikacijų rezultatus kredito rizikos įvertinimo uždaviniui (19). Gauti rezultatai parodė, kad silpnesnių klasifikatorių kolektyvai, atsitiktiniai miškai, veikia geriau, nei pavieniai sprendimų medžiai. Kitas atsitiktinių miškų algoritmo taikymas aprašytas autorių Imad Bou-Hamad Deniso Larocque, Hatemo Ben–Ameuro 2011 metais norint pritaikyti šį algoritmą finansiniams santykiniams rodikliams sprendžiant išgyvenimo uždavinį diskretaus laiko atveju – prognozuojant likusį laiką iki įmonės bankroto (20).

Atliekant praktinius tyrimus vis dažniau įrodoma, kad išvardinti mašininio mokymosi algoritmai geriau veikia, nei klasikiniai diskriminantinė analizė ar logistinė regresija bendrai kredito rizikos ir bankroto prognozavimo uždaviniuose. Tačiau, naujausi tyrimai siūlo vis dar vieną priėjimą prie šios problemos – klasifikatorių kolektyvus. Tai modeliai, kurių prognozė priimama atsižvelgiant į kelių skirtingų modelių prognozių agreguotą rezultatą. Ispanijos ir Didžiosios Britanijos autoriai (A. Alfaro, N. Garcia, M. Gamez, D. Elizondo) 2008 metais lygino gradientinio sustiprinimo klasifikatorių kolektyvo metodą „AdaBoost“ ir dirbtinių neuroninių tinklų veikimą naudodami Europos įmonių finansinius rodiklius ir kelias papildomas įmonių charakteristikas, tokias kaip dydis ar veiklos sritis (21). Tyrimo metu gauti rezultatai - naudojant minėta klasifikatorių kolektyvo metodą gautą paklaida (8,9 %) yra mažesnė nei neuroninių tinklų (12,7 %). Kinijos autoriai (J. Sun, H Li) 2012 (22) tikrino atraminių vektorių mašinų ir atraminių vektorių mašinų kolektyvo modelių skirtumus naudojant Kinijos akcijų rinkų ir buhalterinių tyrimų duomenų bazes. Gauti rezultatai rodo, kad klasifikatorių kolektyvo vidutinis tikslumas – 82.55 %, o atraminių vektorių mašinų modelio vidutinis tikslumas – 79.81 %. Taip pat plati hibridinių prognozavimo sistemų, skirtų identifikuoti problemas prieš bankroto paskelbimą, apžvalga pateikta Ispanų autorių (F. Sanchez-Lasheras, J. de Andres, P. Lorca, F. J. De Cos Juez) 2012 metų straipsnio (23) pirmose keturiose lentelėse.

1.4. Bankroto tikimybės modeliavimas naudojant elgsenos duomenis

Atlikus klasikinių ir mašininio mokymosi metodų taikymo bankroto rizikos prognozavimo uždaviniams spręsti pastebėta, kad dažniausiai sudarant tokius modelius naudojami finansiniai rodikliai. Šiais laikais, apie įmones galime fiksuoti daugiau informacijos ir ją panaudoti siekiant patikslinti bankroto rizikos prognozes. Dėl neapibrėžtų duomenų naudojimo gali būti sunku palyginti literatūroje siūlomus sprendimus, kadangi daroma prielaida, kad kuriant modelius galima naudoti tuos elgsenos rodiklius, kuriuos turime konkrečiu atveju, bet tai nėra privalomi kintamieji, kuriuos galima surinkti apie visas įmones. Kita vertus, dėl tokios informacijos naudojimo dažniausiai gali kilti problemų susijusių su konfidencialumu. Tokie modeliai nėra universalūs, jie gali būti kuriami ir taikomi vienai rinkai, ekonominiam sektoriui, geografiniam regionui ar netgi vienos konkrečios įmonės ribose. Susiaurinus modelio taikymo sritį, išvardintų kriterijų atžvilgiu, galima padidinti naudojamos informacijas įvairovę, taip padidinant kuriamų modelių tikslumą.

Kadangi įmonėse vykstantys procesai yra gan sudėtingi – jų elgsena apibendrinama finansine būkle, todėl tokių tyrimų, kurių metu būtų naudojama komunikacija su skolintoju arba įmonės vėlavimu sumokėti įmokas rasti sunku. Taip pat, tokios informacijos naudojimas

tyrimuose ir jos viešinimas gali kelti įmonių problemų susijusių su konfidencialumu. Tačiau panašių tyrimu galima rasti fizinių asmenų bankroto tyrimo atveju. Bankai tokius tyrimus gali atlikti naudodama informaciją apie kredito kortelių naudojimą. Sumito Agarvalo ir Chunlino Liu 2003 metų straipsnyje (24) ir Davido Groso ir Nikolo Soulelo 2002 metų straipsnyje (25) sprendžiamas fizinių asmenų bankroto rizikos uždavinys logistinės regresijos metodu naudojant sudarytos kredito kortelės sutarties amžių, kredito likutį, kredito panaudojimo lygį per nurodytą laikotarpį, mokėjimai, pirkimai, keli išoriniai veiksliai, nedarbingumas, draudimo polisas, neturto lygis, vidutinė būsto kaina regione ir kt. Tyrimo eiga ir išvados parodo, kad šių kintamųjų naudojimas bankroto prognozavimo uždaviniuose yra racionalus sprendimas ir tai gali praturtinti turimus modelius.

Tačiau galima numanyti, kad elgsenos, bendravimo efektas bankrotui ar nemokumui skiriasi lyginant mažus (fizinius asmenis) ir didelius skolininkus. Mitčelas Petersonas 2002 metais (26) pasiūlė kelis skirtumus, kurie susiję su paskolos kaina skolintojui, artimo ryšio palaikymu su įmonių kontaktiniais asmenimis ir skirtumo fizinių asmenų ir įmonių skaidrumo prasme. Dėl šių priežasčių bankai gali rinktis kaip vertinti skolininką ir kuriais rodikliais geriausiai prognozuoti jų gebėjimą mokėti įmokas ateityje. Ir kada balansų ataskaitų duomenis pakeisti bendravimo su klientais duomenimis. Mitčelas Berilnas ir Loreta Mester 1998 metais (27) komentuodamas santykiais grįsto skolinimo rizikingumą nurodė, kad nors tai ir padeda sudaryti stabilias sutartis su skolintojais, tai bankui gali per brangiai kainuoti laiko ir žmogiškųjų resursų klausimu.

Kitas sprendimas – riziką įvertinti remiantis praeities transakcijomis. Naudojant tokį metodą galima atsižvelgti į tai ar skolintojas laiku moka įmokas, kaip dažnai ir kiek ilgai vėluoja. Skolintojo patikimumas tvarkingų transakcijų atžvilgiu gali atskleisti bankroto artėjimo tikimybę geriau nei finansinės ataskaitos. Naudojantis šiais kintamaisiais Bilas Fairas ir Erlas Isakas 1956 metais įkūrė kompaniją „FICO“, kuri siūlė kredito rizikos įvertinimo metodus. 1990 metais Sistemos išanalizavo 17 bankų surinktus duomenis apie 5000 mažų įmonių paimtas paskolas ir sukurė mažų įmonių kredito rizikos įvertinimo sistemą (angl. Small Business Scoring System) SBSS. Tokios sistemos pagrindu kuriami ir patentuojami kiti sprendimai. Amerikos išradėjų kolektyvas (Debashis Ghosh, William A. Nobili, Arun R. Pinto, Kurt D. Newman, David N. Joffe, Sudeshna Banerjee) 2007 metais užpatentavo sistemą įvertinančia ar įmonė gebės vykdyti įsipareigojimus (28). Ši sistema atsižvelgia į kredito naudojimą 44 skirtingoms kategorijoms į kurias įeina ir pramogos, būtinosios prekės, prabangos prekės ir kt. tai dar kartą parodo, kad skolininkų elgsena yra svarbus faktorius kredito rizikos problemoms spręsti.

1.5. Duomenų tyrybos metodologija

Duomenų tyryba tai procesas apimantis tris pagrindines dalis: dalykinės srities pažinimą, darbą su duomenimis ir rezultatų diegimą produkcinėje aplinkoje. Visame procese dažnai dalyvauja skirtingų sričių žinovai – dalykinės srities, modeliavimo, informacinių sistemų specialistai. Pagrindinių žingsnių, apimančių visas šias dalis, sekai nusakyti pagrinde naudojamos kelios metodologijos, arba jų modifikacijos daromos prisitaikant prie projekto apribojimų. Keli metodologijų pavyzdžiai:

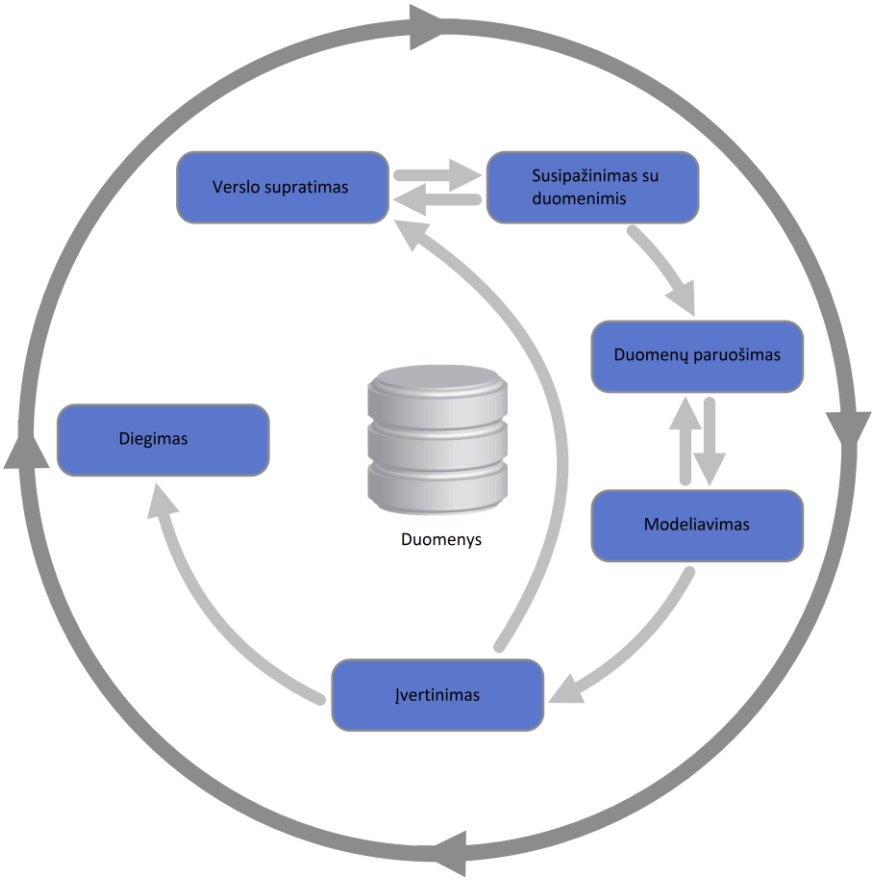
SEMMA žodžių Sample, Explore, Modify, Model, Assess (vertimas atrinkti pavyzdžius, ištirti, modifikuoti, modeliuoti, įvertinti) akronimas. SAS institutas siūlo duomenų tyrybos ciklą susidedantį iš penkių dalių (29):

1. Atrinki pavyzdžius – ši pasirenkama metodologijos dalis labiau būtina dirbant su dideliais duomenų kiekiais, kai naudojant mažesnį duomenų poimtį norima sumažinti laiko sąnaudas atliekant papratus skaičiavimus ar pradinę duomenų apžvalgą. Taip pat, šios stadijos metu atliekama ir pagrindinio duomenų skaidymo į apmokymo, testavimo ir validacijos imtis, kurios naudojamos kuriant modelius ir vertinant jų gerumą, modelius kuriant naudojant apmokymo imtį, o jų veikimą išbandome modelio kūrimo metu nenaudotais duomenimis iš testavimo ir validacijos imčių.
2. Ištirti – šioje dalyje stengiamasi surasti dalį išskirčių ir šablonų duomenyse. Naudotini metodai: aprašomoji statistika, vizualizacijos, faktorinė analizė, klasterizavimas. Šios dalies metu probleminė sritis susiejama su turimais duomenimis tam, kad būtų iškelti teisingi, pagrįsti klausimai.
3. Modifikuoti – šioje dalyje kintamieji paruošiami modeliavimui – atrenkami reikalingi, sukuriami nauji ir modifikuojami kintamieji, kad jie tiktų naudotiniems metodams. Naudojantis įžvalgomis iš 1.2 metodologijos dalies galima sukurti kintamuosius su atliktu grupavimu ar pogrupiais, pašalinami kintamieji dėl didelio išskirčių ar tuščių reikšmių kiekio. Taip pat, atrenkami reikšmingiausi kintamieji.
4. Modeliavimas – programinių priemonių naudojimas norint rasti šablonus duomenyse pagal kuriuos galima atlikti prognozavimą. Naudojami mašininio mokymosi metodai: sprendimų medžiai, atsitiktiniai miškai ir kiti sprendimų medžių pagrindu pagrįsti metodai, neuroniniai tinklai, logistinė regresija, laiko eilutės, principinių komponenčių metodai.
5. Įvertinti – šioje dalyje įvertinami duomenų tyrybos proceso rezultatai, įžvalgos sujungiamos su dalykinės srities žiniomis ir patikrinamas jų validumas, naudingumas verslui.

Norint gauti gerą rezultatą, šie penki žingsniai kartojami kelis kartus, kol sukuriamas geriausias modelis ir jis yra pritaikomas produkcinėje aplinkoje, kur matoma investicijos į duomenų tyrybos procesą grąža.

CRISP-DM – Cross Industry Standarf Process for Data Mining (vertimas: tarp industrinis duomenų tyrybos proceso standartas), pasiūlytas 1996 metais kaip Europos Sąjungos projektas ir buvo įtvirtintas kelių duomenų analizės įmonių: SPSS, Teradata, Daimer AG. Procesas skaidomas į šešis žingsnius (30), kurie taip pat pavaizduoti 1.1 paveiksle:

1. Verslo supratimas – pradinė dalis, kurioje stengiamasi suprasti projekto esmę, preliminarius planus, iškeltus klausimus ir paversti verslo problemą į duomenų tyrybos uždavinį.
2. Susipažinimas su duomenimis – ši dalis apima duomenų surinkimą ir pradinę analizę. T.y. stengiamasi įvertinti duomenis kokybiniais aspektais, gauti pirmas įžvalgas ir rasti ryšį tarp verslo problemos.
3. Duomenų paruošimas – naudojantis gautomis įžvalgomis antroje dalyje modifikuojami duomenys taip, kad būtų juos galima naudoti tolimesniame duomenų tyrybos procese. Taip pat, kuriami nauji, išvestiniai kintamieji, formatuojami apjungiami ir valomi duomenys.
4. Modeliavimas – naudojamos įvairios modeliavimo technikos, kalibruojami jų parametrai, kad būtų gautas kuo geresnis modelis, atliekamas testavimas (kryžminė validacija), modelio prielaidų tikrinimas.
5. Įvertinimas – vertinama ar gauti modeliai atsako į visus dalykinės srities iškeltus klausimus pirmame žingsnyje.
6. Diegimas – modelis diegiamas produkcinėje aplinkoje, kur jo naudojimas pateisintų investicijas.



1.1 pav. CRISP-DM metodologijos diagrama

Šios dvi metodologijos nusakančios panašų priėjimą prie duomenų tyrybos uždavinių tinka ir bankroto tikimybės prognozavimo uždaviniui.

Atlikus literatūros analizę daromos išvados, kad nors problema nėra nauja, tačiau aktuali ir svarbi ekonomikoje, ypač tarp mažų ir vidutinių įmonių, kurios yra vienas pagrindinių šios dienos ekonomikos variklių. Žinant, kad vienos įmonės bankrotas gali sužadinti išorinių veiksnių pokyčius, nuo kurių priklauso šalies bendrojo vidaus produkto lygis ir kitų įmonių stabilumas, bankroto riziką reikia identifikuoti kuo anksčiau. Taip pat, literatūroje galima rasti naujų metodų pritaikymų, kurių pagalba galima lengviau adaptuotis prie mažų ir vidutinių įmonių dinamikos ir bankroto rizikos numatymo, toliau parenkami atsitiktinio miško, neuroninių tinklų ir ekstremalaus gradientinio sustiprinimo medžių metodai. Pastaruosius dešimtmečius didelius pokyčius įnešė ne tik mašininio mokymosi algoritmai ar jų taikymo metodologijos, bet ir vis didėjantys duomenų kiekiai bei jų įvairovė. Tačiau, dėl didelės naudojamų duomenų įvairovės tenka susiaurinti modelių taikymo ribas, todėl bankroto tikimybės modelių kūrimas naudojantis mašininio mokymosi algoritmais gali būti susiaurintas netgi iki vienos įmonės, šiuo atveju, iš Skandinavijos banko paėmusių paskolą smulkaus ir vidutinio verslo klientų.

2. METODŲ ANALIZĖ

Atlikus literatūros analizę toliau apžvelgsime parinktus metodus, kurie tinkami bankroto tikimybės prognozavimo uždaviniui. Šiame skyriuje pateikiami darbe naudojamų mašininio mokymosi metodų detalūs aprašymai bei kuriamų modelių kokybės įvertinimo kriterijai.

2.1. Modelių įvertinimas

Tarkime, kad z – prognozuojama stebėjimo būsena, kuri gali įgyti dvi reikšmes: B – bankroto, N – ne bankroto. Paprastu atveju, būtų galima fiksuoti aibę A = [S; 1], kur S ∈ [0; 1] ir yra laikomas slenksčiu, nuo kurio stebėjimui priskiriama prognozuojama bankroto būsena, tuomet:

$$s \in \mathrm{A} \Rightarrow \mathrm{z} = B$$
$$s \notin \mathrm{A} \Rightarrow \mathrm{z} = \mathrm{N} \quad \textbf{(2.1)}$$

Dažniausiai tikimasi, kad S galima priskirti reikšmę 0,5 ir interpretuoti, kad įrašui priskyrus tikimybę virš 0,5, galima jam prognozuoti bankrotą ir priskirti būseną B. Tačiau, tai priklauso nuo dalykinės srities, kainos dėl padarytos klaidos ar kitų aspektų tai gali būti pakeista.

Kitas siūlomas būdas parinkti S įvertį yra optimizuoti daromas klaidas (31):

- I tipo klaida: klaidingai atmetama būsena;

$$P[I\ tipo\ klaida] = P[S \notin \mathrm{A}|\mathrm{B}] \quad \textbf{(2.2)}$$

- II tipo klaida: klaidingai priimama būsena B.

$$P[II\ tipo\ klaida] = P[S \in \mathrm{A}|\mathrm{N}] \quad \textbf{(2.3)}$$

Tuomet parinkus pasikliautinumo lygmenį α ∈ [0; 1] (dažniausiai mažas skaičius: 0,01, 0,05, 0,1), parenkamas optimalus S minimizuojant II tipo klaidą laikantis šio apribojimo:

$$P[I\ tipo\ klaida] \leq \alpha \quad \textbf{(2.4)}$$

Tyrimo rezultatams vertinti naudojamas ROC grafikas ir ploto po ROC kreive AUC metrika (31).

1. ROC – grafinė vizualizacija, skirta klasių atskirčiai įvertinti. Abscisių ašyje dažniausiai atvaizduojamas klaidingo identifikavimo dažnis, tai yra sąlyginė tikimybė, rodanti, kad nebankrutuojančių klientų slenksčio įvertis S yra mažesnis arba lygus įverčiui s:

$$P[S \leq s|N] = F_N(s) \quad \textbf{(2.5)}$$

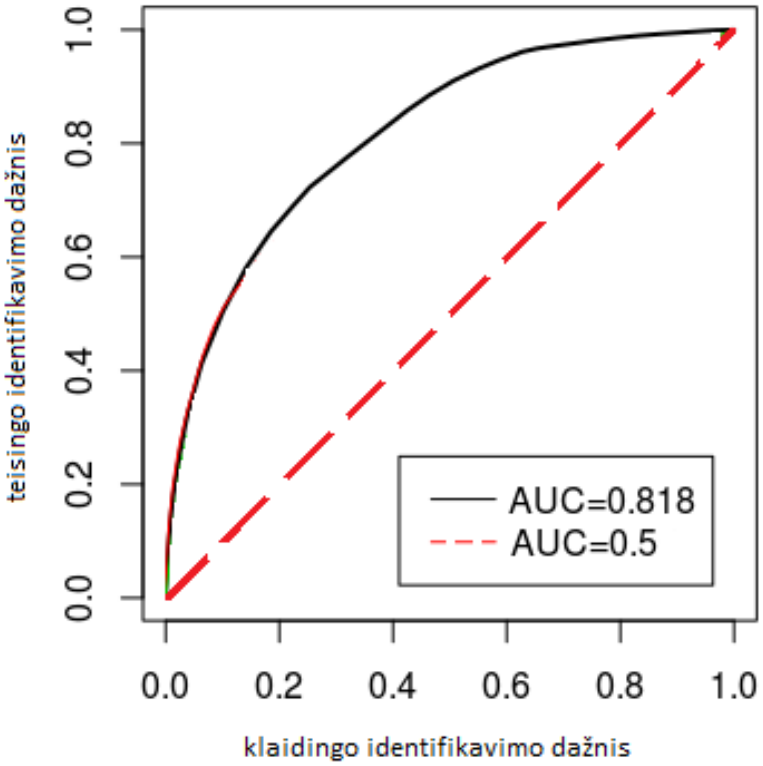
Tuomet klaidingo identifikavimo dažnis atstoja dalį nebankrutavusių populiacijos, kuri būtų neteisingai identifikuota jei būtų parinktas slenkstis s. ROC kreivės funkcija užrašoma tokia išraiška:

$$ROC(u) = F_D\left(F_N^{-1}(u)\right) \tag{2.6}$$

čia:

u ∈ (0, 1).

ROC kreivės grafikas gali būti vaizduojamas kaip visų taškų (u, ROC(u)), u ∈ (0, 1) arba visų taškų ($F_N(s)$,$F_D(s)$), s ∈ R. Kuo aukščiau taškai išsidėstę virš 45 laipsnių tiesės einančios per taškus (0, 0) ir (1, 1), tuo geresnis modelis t.y. kokybiškiau klasifikuoja įrašus, kai skirtingų modelių kreivės kertasi – nėra vienareikšmiško paaiškinimo, kuris modelis geresnis. Tiesė jungianti taškus (0, 0) ir (1, 1) nusako atsitiktinį spėjimą.



2.1 pav. ROC kreivės pavyzdys su AUC įverčiu

2. AUC – įvertis, leidžiantis skaitine išraiška įvertinti modelius pagal ROC kreivę, t.y. plotą tarp ROC kreivės ir abscisių ašies. Šis plotas apskaičiuojamas kaip ROC funkcijos apibrėžtinis integralas intervale [0, 1]:

$$AUC = \int_0^1 ROC(u)du \tag{2.7}$$

Taip pat, AUC įvertis gali būti apibrėžiamas kaip tikimybė, kad atsitiktinai parinkto įrašo su būsena N įvertis SN yra didesnis už nepriklausomai parinkto įrašo su būsena D:

$$AUC = P[S_D - S_N] \quad \textbf{(2.8)}$$

Didesnė AUC reikšmė reiškia, kad modelis geba geriau atskirti klases. Kai AUC = 1 – modelis atskiria klases neklysdamas. Kai AUC = 0,5 – klasės priskiriamos atsitiktinai.

Sudarant ROC kreivę ir apskaičiuojant AUC įvertį yra įvertinami visi galimi bankroto priskyrimo slenksčiai S. Tačiau, tai nepasako kaip elgtis praktikoje t.y. nenurodo kurį slenkstį naudoti atskiriant klases.

Parinkus slenkstį galima įverti teisingai ir klaidingai identifikuotus įvykius sudarant sumaišymų matricą. Mokymosi su mokytoju binarinio klasifikavimo atveju sumaišymo matrica pavaizduota 2.1 lentelėje.

2.1 lentelė. Sumaišymų matrica

		Prognozė		**Iš viso atvejų**
		1	**0**	
Realios klasės	**1**	TP	FN	P
	0	FP	TN	N

Sumaišymo matricoje pavaizduotos keturios celės, nusakančios realių klasių ir modelio prognozuojamų klasių sutapimų arba ne sutapimų kiekius. Žymėjimui lentelėje naudojami angliškų terminų akronimai:

- TP. Teisingas identifikavimas (angl. True positive) – teisingai identifikuojamas įvykio buvimas.
- FP. Klaidingas identifikavimas (angl. False positive) – klaidingai identifikuojamas įvykio būvimas. Dar vadinama pirmo tipo klaida.
- FN. Klaidingas atmetimas (angl. False negative) – klaidingai atmetamas įvykio būvimas. Dar vadinama antro tipo klaida.
- TN. Teisingas atmetimas (angl. True negative) – teisingai atmetamas įvykio būvimas.
- P – visi įvykio būvimo atvejai.
- N- visi įvykio nebūvimo atvejai.

Slenkstis turi būti parenkamas taip, kad minimizuotų FP arba FN, priklausomai nuo dalykinės srities. Naudojant šiuos keturis dažnius apibrėžiamos papildomos metrikos, kurias galima naudoti modelio įvertinimui arba modelio parametrų optimizavimui pagal svarbiausią metriką.

- Ypatumas (angl. specificity) – teisingo atmetimo dažnis.

$$TNR = \frac{TN}{TN + FP} = \frac{TN}{N} \quad \textbf{(2.9)}$$

- Jautrumas (angl. sensitivity) – teisingo identifikavimo dažnis.

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{P} \quad \textbf{(2.10)}$$

- Tikslumas (angl. precision) – teisingai identifikuotų įvykio būvimų ir visų prognozuojamų įvykių santykis

$$PPV = \frac{TP}{TP + FP} \quad \textbf{(2.11)}$$

- Bendrasis tikslumas (angl. accuracy) – teisingai identifikuotų įvykių ir teisingai atlikto įvykio atmetimo sumos santykis su bendru stebėjimų kiekiu.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{P + N} \quad \textbf{(2.12)}$$

- F1 įvertis – harmoninis tikslumo ir jautrumo vidurkis.

$$F1 = 2 \cdot \frac{1}{\frac{1}{TPR} + \frac{1}{PPV}} = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} \quad \textbf{(2.13)}$$

Naudojantis šiomis metrikomis galima rasti optimalų slenkstį. Šiame darbe slenkstis bus parenkamas maksimizuojant F1 įvertį arba ieškant tikslumo ir jautrumo sankirtos koordinačių, nes praktikoje šie du atvejai grąžina artimas slenksčio reikšmes. Tokiu atveju gaunamas slenkstis nesubalansuotų duomenų atveju padės tiksliau išskirti bankrutuojančias įmones, tačiau nebus konservatyvus. Kadangi slenksčio parinkimas glaudžiai susijęs su dalykine sritimi ir konkrečia problema konservatyvesnio modelio variantą galima gauti slenkstį parenkant pagal jautrumo ir ypatumo metrikų susikirtimo tašką. Tokiu atveju stengiamasi atsižvelgti į kuo didesnį mažesnės klasės procentinį atitikimą ir kuo mažesnį didesnės klasės procentinį neatitikimą.

Norint gauti kuo objektyvesnį modelio vertinimą išvardintų metrikų pagalba naudojama stratifikuota kryžminė patikra. Dažna mašininio mokymosi metodų klaida yra persimokymas – sukurtas modelis atkartoja apmokymo imties duomenis t.y. modelis puikiai prognozuoja tik tuos įrašus, kurie buvo naudoti apmokymo metu, tačiau prastai prognozuoja naujus, nematytus stebėjimus. Tokių modelių vidinė paklaida būna labai maža, tačiau testavimo imties paklaida labai didelė.

Norint išvengti šios problemos kuriant modelius būtina atlikti modelio patikrinimą naudojant testavimo imtį. Tam, kad išvengti atsitiktinumo parenkant per daug panašias apmokymo ir testavimo imtis, kuomet tiek vidinė, tiek testavimo paklaida būtų mažos, šią operaciją patartiną atlikti kelis kartus. Jeigu apsibrėžiame n dalių kryžminė patikra, tokiu atveju bus sukuriami n modelių naudojant n skirtingų apmokymo imčių ir jie testuojami naudojant n

skirtingų testavimo imčių – geriausiu modeliu laikomas tas, kurio vidutinė testavimo paklaida yra mažiausia.

Kryžminę patikrą galime apibrėžti kelių žingsnių ciklu. Tarkime, turime imtį A ir ji dalinama į n lygių dalių A_i, i = 1,…,n.

1. Parenkame metodo parametrus $\alpha_k = (\alpha_1, \alpha_2,..., \alpha_k)$
2. Apmokymo imčiai parenkamas duomenų poaibis $A \backslash A_i$, ir testavimo imčiai paliekamas poaibis A_i.
3. Apmokomas modelis su parametrų rinkiniu α_i naudojant apmokymo imtį.
4. Įvertinama modelio paklaida naudojant testavimo imtį A_i.

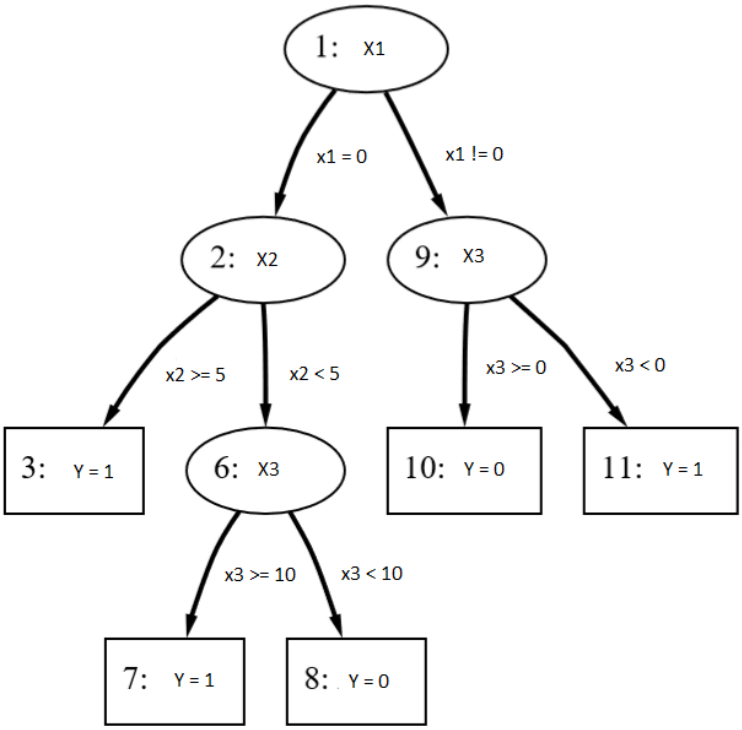
Kartojant 1-4 žingsnius n kartų gauname n paklaidų įvertinimų. Modelio su α_k parametrais vidutinę paklaidą įvertinama kaip n paklaidų vidurkis. Norint palyginti K skirtingų modelių reikia šį ciklą atlikti K kartų su skirtingais α_k parametrų rinkiniais.

2.2. Sprendimų medis

Sprendimų medis (angl. decision tree) – fundamentalus mašininio mokymosi algoritmas, skirtas spręsti regresijos ir klasifikacijos uždaviniams (32). Šio algoritmo pagrindiniai privalumai:

1. Sugeneruojami naudotojui suprantami taisyklių rinkiniai, kurie gali būti kitos analizės pagrindu;
2. Gali išspręsti regresijos ar klasifikavimo uždavinius be papildomų dalykinės srities žinių;

Algoritmas paremtas rekursiniu duomenų skaidymu į grupes pagal apibrėžtus naudos arba algoritmo stabdymo kriterijus. Sprendimų medis, kurio priklausomas kintamasis yra kategorinis turi baigtinį skaičių klasių, dažnai vadinamas klasifikavimo medžiu, kurių lapuose atvaizduojama viena iš klasių (žr. 2.2 paveikslėlį).



2.2 pav. Sprendimų medžio pavyzdys

Tarkime, turime N stebėjimų apmokymo imtyje $(x^{(1)},y^{(1)})$, $(x^{(2)},y^{(2)})$,..., $(x^{(N)},y^{(N)})$, kur $x^{(i)}$ – n atributų vektorius ir $y^{(i)} \in \{a_1, a_2,..., a_C\}$ – viena iš C klasių, kuri atitinka $x^{(i)}$ įėjimo vektorių. Pačioje pirmoje dalyje yra $N^{(0)}$ stebėjimu, kiekvienas tolimesnis medžio mazgas v turės $N^{(v)}$ stebėjimų. Tuomet, apmokymo imties stebėjimų kiekis, priklausantis klasei a_k, mazge v žymimas $N_{a_k}^{(v)}$. Tuomet

$$\sum_{k=1}^{C} N_{a_k}^{(v)} = N^{(v)} \tag{2.14}$$

Vienas iš naudingumo kriterijų yra pagrįstas entropijos matu (33). Skeliant duomenis pasirenkamas toks kriterijus, kuriuo labiausiai sumažinama entropija. Tarkime mazgo u, kuris turi V vaikinių mazgų po dalinimo, naudos kriterijus užrašomas taip:

$$Gain(x_j) = \left[\sum_{k=1}^{C} -\left(\frac{N_{a_k}^{(u)}}{N^{(u)}}\right) \log\left(\frac{N_{a_k}^{(u)}}{N^{(u)}}\right)\right] - \left[\sum_{v=1}^{V}\left(\frac{N^{(v)}}{N^{(u)}}\right)\sum_{k=1}^{C} -\left(\frac{N_{a_k}^{(v)}}{N^{(v)}}\right) \log\left(\frac{N_{a_k}^{(v)}}{N^{(v)}}\right)\right] \tag{2.15}$$

Lygties pirmoji dalis atstoja tėvinio mazgo u entropiją, o antroji dalis svorinė vaikinių mazgų entropija. Šių dviejų dalių skirtumas – informacijos nauda. Atributas x_j, kuris teikia didžiausią informacijos naudą parenkamas dalinimui mazge u.

Antras populiarus būdas parinkti optimalų kintamąjį duomenų dalinimui vadinamas Gini „užterštumo“ indeksu, kuris išreiškia kaip dažnai atsitiktinai parinktas stebėjimas $x^{(i)}$ bus neteisingai priskirtas vienai iš C klasių. Gini indeksas išreiškiamas:

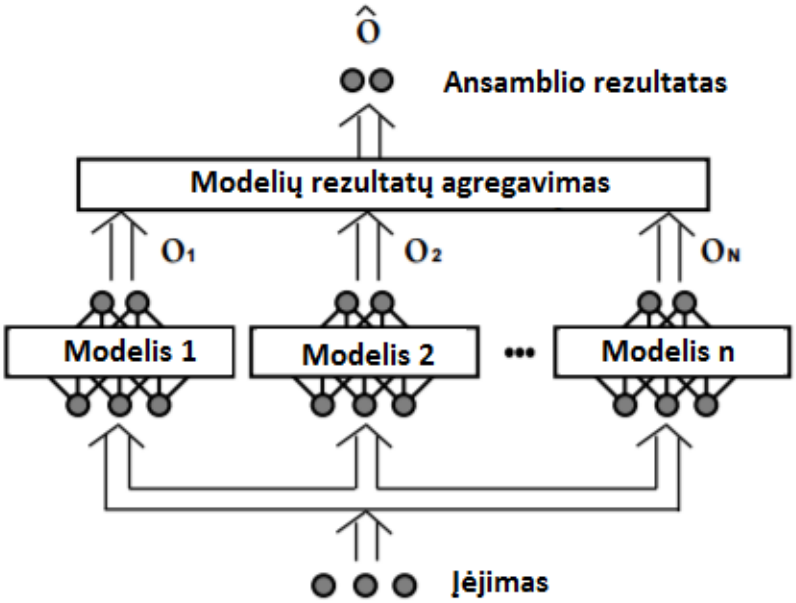
$$Gini(x_j) = \frac{1}{N}\left[\sum_{k=1}^{C}\sum_{v=1}^{V}\frac{N_{a_k}^{(v)^2}}{N^{(v)}} - \sum_{k=1}^{C}\frac{N_{a_k}^{(u)^2}}{N^{(u)}}\right] \quad \textbf{(2.16)}$$

Šiuo atveju parenkamas tas atributas x_j, kuris suteikia didžiausią „užterštumo“ sumažinimą.

Sprendimo medžio sudarymas tęsiamas kol mazge yra kelių priklausomo kintamojo klasių stebėjimų, iki fiksuoto medžio gylio, arba kol atitinka tam tikrus stabdymo kriterijus. Sprendimų medis yra genimas (angl. pruning), kad būtų išvengiamos permokymo (angl. overfitting) problemos, kuri atsiranda, paskutinius duomenų skėlimus atliekant pagal išskirtinius apmokymo imties kriterijus, kurie dažniausiai nėra universalūs. Taip pat, genint sprendimų medžius sumažinamas perteklinių veiksmų kiekis sudarant modelį ir modelį nusakančių taisyklių ilgiai. Genėjimo principas – nebeskaidyti $N^{(i)}$ duomenų į vaikinius mazgus (iš viršaus į apačią), arba pastebėjus, kad dalinimas nesuteikė daug informacijos – apjungti į vieną mazgą (iš apačios į viršų) (34).

2.3. Klasifikatorių kolektyvai

Klasifikatorių kolektyvai (angl. ensemble methods) – kelių atskirai apmokytų modelių grupė, kurios prognozuojama priklausomo kintamojo reikšmė yra visų modelių prognozių agregavimo rezultatas (žr. 2.3 paveikslėlį). Kelių skirtingų modelių sudarymas ir jų rezultatų agregavimas yra naudingas tik tuo atveju, jeigu modelių rezultatai skiriasi (jeigu rezultatai sutampa, tai papildomų modelių kūrimui gaištamas laikas ir kiti resursai neduoda papildomos naudos modelio kokybei). Hansenas ir Salamonas (1990) (35) parodė, kad kiekvieno neuroninio tinklo klasifikatoriaus, priklausančio klasifikatorių kolektyvui, paklaidai esant mažesnei nei 50 % ir kolektyvui priklausančių modelių kiekiui artėjant į begalybę bendra paklaida artėja į 0 %, tačiau tolimesniais tyrimais tokių rezultatų patvirtinti nepavyko. Šiuos tyrimus apibendrinus galima teigti, kad idealų klasifikatorių kolektyvą sudaro gerai prognozuojantys modeliai, kurie daro klaidai skirtingose vietose (modelių rezultatai nesutampa) (36). Tam, kad gauti gerus, bet besiskiriančius modelius sukuriami dirbtini apribojimai arba modeliai dirbtinai sustiprinami apmokymo metu. Populiariausi klasifikatorių kolektyvų metodai yra atsitiktinis miškas (angl. bagging) ir skirtingų svorių modeliams skyrimas siekiant sustiprinti tam tikrus grupėje esančius modelius (angl. boosting) (37).



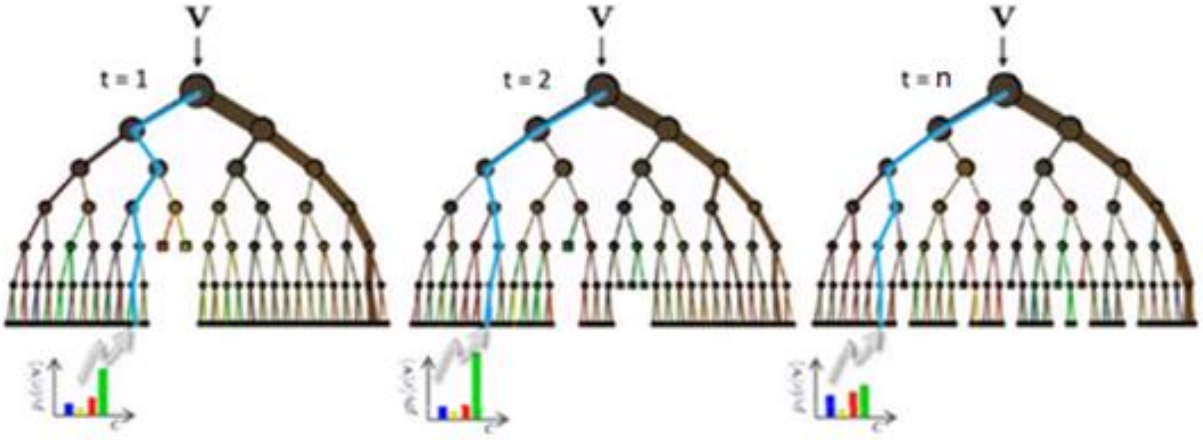
2.3 pav. Klasifikatorių kolektyvo metodo schema

2.4. Atsitiktinis miškas

Atsitiktinis miškas (angl. random forest) – kolektyvinio tipo mašininio mokymosi metodas klasifikavimo ir regresijos uždaviniams. Modeliavimo metu sukuriama daug negenėtų (angl. unpruned) sprendimo medžių ant savirankos agregavimo būdu (angl. bootstrap aggregating) parinktų apmokymo imties poimčių. Kiekviename medyje sprendimai priimami naudojant kintamųjų atrankos būdu atrinktus kintamuosius.

Atsitiktinio miško algoritmas:

1. Savirankos būdu parenkama n poaibių A_i, i = 1…n iš pradinės aibės A_0.
2. Kiekvienai iš n imčių sukuriamas negenėtas sprendimo medis. Kiekviename medžio sprendimo mazge iš kintamųjų atrankos metu atrinktų m kintamųjų pasirenkamas kintamasis ir to kintamojo kategorija (jei kintamasis yra kategorinis) arba lūžio taškas (jei kintamasis yra skaitinis), kuris geriausiai padalintų mazgą pasiekusią n-tojo poimčio duomenų dalį t.y. geriausiai atskirtų skirtingas klases esančias šioje duomenų dalyje.
3. Norint gauti atsitiktinio miško prognozes, naujas įrašas v turi būti apskaičiuota kiekviename iš n sprendimų medžių ir prognozė suagreguota. Klasifikacijos atveju galutinis sprendimas priimamas balsų dauguma, regresijos atveju realių skaičių agregavimo funkcijomis (vidurkis, moda ir kt.) (žr. 2.4 paveikslėlį).



2.4 pav. Atsitiktinio miško rezultatų skaičiavimo pavyzdys

Kadangi kiekvienas iš n sprendimo medžių buvo sudarytas naudojant atskirą poaibį A_i, tai kiekvienas i-tasis sprendimo medis apmokymo imtyje turi duomenų dalį, kuri nebuvo naudota apmokymo metu $A_0 \backslash A_i$, i = 1...n. Šia dalį likusią duomenų dalį (angl. out of bag) galima naudoti testavimui ir vidinės, apmokymų metu daromos paklaidos įvertinimui, kurios pagalba galima įvertinti skirtingus modelius nenaudojant testavimo imties arba kryžminės validacijos (38).

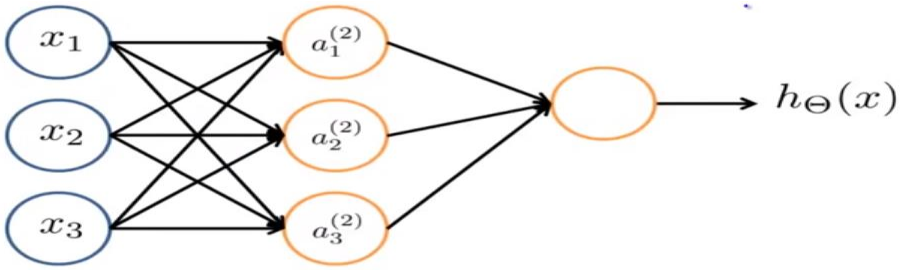
2.5. Sprendimų medžių sustiprinti algoritmai

Sprendimų medžių pagrindu kuriami algoritmai gali būti sustiprinami (angl. boosting) atsižvelgiant į svorinius koeficientus, kurie suteikiami kiekvienam sprendimo medžiui. Jeigu apmokymo metu i-tasis spendimo medis teisingai prognozavo įrašo v priklausomąjį kintamąjį, o j-tasis sprendimo medis suklydo. Tuomet i-tojo medžio sprendimas galutiniam rezultatui turės didesnę įtaką, nei j-tojo medžio rezultatas agreguojant ar balsuojant dėl galutinės prognozuojamos priklausomojo kintamojo reikšmės (39). Kitaip tariant, sprendimų medžiai sudaromi sekoje, kurioje paskesnio medžio sudaryme naudojama informacija apie praeitų medžių padarytas klaidas. Klasifikacijos atveju galime išskirti tris pagrindinius parametrus, kuriuos reikėtų optimizuoti, siekiant sukurti gerą modelį (40):

1. Medžių kiekis. Priešingai tradiciniam atsitiktinio miško variantui – didinant medžių kiekį gali atsirasti permokymo problemos.
2. Apmokymo greitis. Dažniausiai mažas teigiamas skaičius, kuris turi priešingą ryšį su medžių kiekiu t.y. parinkus mažą apmokymo greitį reikės didelio medžių kiekio, kad būtų gautas gerai klasifikuojantis modelis.
3. Dalinimų kiekis medyje. Nuo to priklauso kokio gylio medžiai bus ir kiek maksimaliai kintamųjų gali būti panaudota sudarinėjant medį.

2.6. Neuroniniai tinklai

Neuroniniai tinklai – bandymas atkartoti žmogaus smegenyse esančių neuronų struktūrą, kuri galėtų pati išmokti spręsti regresijos ir klasifikavimo uždavinius. Tinklas turi įeinamąjį, paslėptuosius ir išeinamąjį sluoksnius. Įeinamasis sluoksnis sudarytas iš tiek mazgų kiek turima kintamųjų, binarinio klasifikavimo atveju išeinamasis sluoksnis turi vieną mazgą, kuriame grąžinama tikimybė priklausyti įvykių klasei. Pilnai sujungto tinklo atveju visi sluoksnio mazgai sujungti su visais kito sluoksnio mazgais, pradedant nuo įėjimo sluoksnio kaip pavaizduota 2.5 paveikslėlyje.



2.5 pav. Neuroninio tinklo pavyzdys

Pavyzdyje pateiktą tinklą sudaro įėjimo sluoksnis x, paslėptieji sluoksniai, kuriuose ir slypi neuronai (šiame pavyzdyje yra vienas paslėptasis sluoksnis su trimis neuronais), išeities sluoksnis, kuris grąžina paskutinę apskaičiuotą reikšmę. Kiekvieno neurono reikšmę galima apskaičiuoti pagal nurodytas lygtis:

$$a_1^{(2)} = g(\theta_{10}^{(1)} \cdot x_0 + \theta_{11}^{(1)} \cdot x_1 + \theta_{12}^{(1)} \cdot x_2 + \theta_{13}^{(1)} \cdot x_3) \quad \textbf{(2.17)}$$

$$a_2^{(2)} = g(\theta_{20}^{(1)} \cdot x_0 + \theta_{21}^{(1)} \cdot x_1 + \theta_{22}^{(1)} \cdot x_2 + \theta_{23}^{(1)} \cdot x_3)$$

$$a_3^{(2)} = g(\theta_{30}^{(1)} \cdot x_0 + \theta_{31}^{(1)} \cdot x_1 + \theta_{32}^{(1)} \cdot x_2 + \theta_{33}^{(1)} \cdot x_3)$$

$$h_\theta(x) = a_1^{(3)} = g(\theta_{10}^{(2)} \cdot a_0^{(2)} + \theta_{11}^{(2)} \cdot a_1^{(2)} + \theta_{12}^{(2)} \cdot a_2^{(2)} + \theta_{13}^{(2)} \cdot a_3^{(2)})$$

čia:

x – įeities kintamųjų vektorius (x_0, x_1, x_2, x_3), kur $x_0 = 1$, laisvas kintamasis.

θ – svoriai arba parametrai, kurie nurodo į neuroną įeinančių kintamųjų svarbą.

Kiekvieno matematinio modelio esmė yra sumažinti paklaidą (skirtumo tarp prognozuojamos ir tikrosios reikšmės sumą), t.y. parinkti tokius parametrus, kad su mokymosi imtimi kainos funkcija būtų kuo mažesnė. Kadangi neuroniniai tinklai dažnai naudojami klasifikavimo uždaviniams spręsti galima tinklo sprendimą užrašyti naudodami (2.18) kainos funkciją, kurios neuronuose yra eksponentinės funkcijos (2.19).

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \cdot \log\left(h_\theta\left(x^{(i)}\right)\right)_k + (1 - y_k^{(i)}) \cdot \log(1 - \left(h_\theta\left(x^{(i)}\right)\right)_k)\right] \quad \textbf{(2.18)}$$

$$+ \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(\theta_{ji}^{(l)})^2$$

čia:

K – klasių kiekis;

M – stebėjimų mokymo imtyje kiekis;

L – sluoksnių skaičius;

S_l – neuronų kiekis l-tajame sluoksnyje

λ – reguliarizacijos parametras.

$$h_\theta(x) = \frac{1}{1 + x^{-\theta^T x}} \quad \textbf{(2.19)}$$

Norint sudaryti tikslų neuroninį tinklą reikia parinkti tinkamus svorinius parametrus. Neuroninio tinklo privalumas yra, tas, kad svorių matricos θ parametrai turi būti perskaičiuojami pagal apmokamąją imtį t.y. pačios pirmos iteracijos metu parametrai parenkami atsitiktinai, tuomet perėjus per visą tinklą gautas rezultatas palyginamas su tikru rezultatu ir pagal tai pakoreguojama svorių matrica θ.

3. EMPIRINIS TYRIMAS

Šiame darbe bandoma į problemą pažiūrėti ne iš finansinių ataskaitų pusės, bet iš kreditorių bendravimo, elgsenos su mažomis ir vidutinėmis įmonėmis, kurioms buvo suteiktos paskolos ir tokių įmonių įmokų mokėjimų pareigingumu. Šiame skyriuje aprašomi duomenys, duomenų tvarkymo žingsniai, bankroto tikimybės modeliavimas naudojant kelis metodus, geriausio modelio atrinkimas ir rezultatų palyginimas pagal ROC kreivę ir AUC metriką.

Iš susijusių darbų apžvalgos parinkti mašininio mokymosi algoritmai, kurie dažnai naudojami bankroto tikimybės prognozavimui naudojant finansinius rodiklius, bet ne elgsenos ar transakcijų. Taip pat, šie metodai yra taikomi nesubalansuotų duomenų atveju, todėl jiems nereikia dirbtinai išplėsti (angl. upsampling) arba sumažinti (angl. downsampling) imties, norint dirbtinai sudaryti subalansuotą apmokymo imtį. Pasirinkti metodai:

- Neuroniniai tinklai;
- Atsitiktinis miškas;
- Ekstremalaus gradientinio sustiprinimo medžių metodas (XGBoost).

Norint taikyti šiuos metodus duomenyse negali būti trūkstamų reikšmių, taip pat XGBoost algoritmo realizacija R įrankyje veikia tik su skaitinio tipo kintamaisiais. Atlikti duomenų pakeitimai taip pat pateikti šiame skyriuje.

Tyrimo planas sudaromas pagal literatūros apžvalgoje atliktą mašininio mokymosi algoritmų taikymo metodologiją:

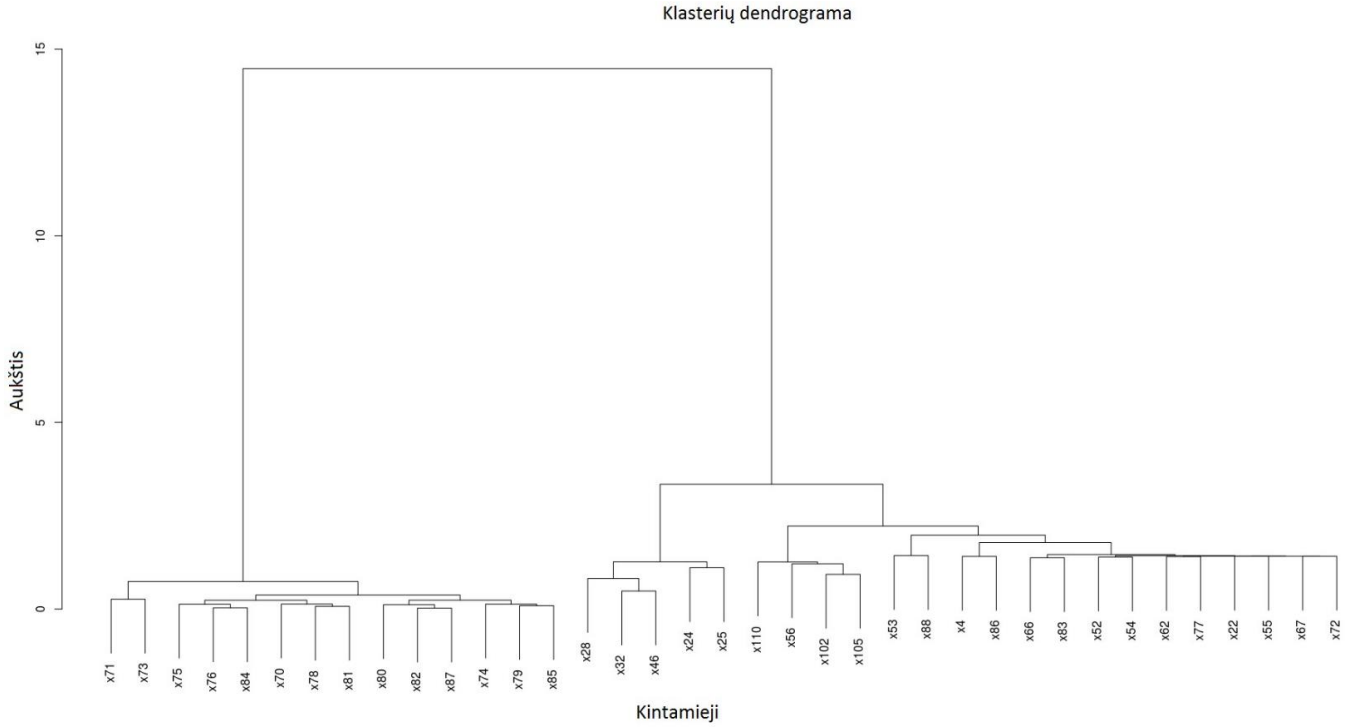
1. Atliekamas duomenų paruošimas;
2. Atliekamas pasirinktų metodų parametrų derinimas, siekiant rasti geriausią parametrų rinkinį turimiems duomenims;
3. Naudojantis kryžminės patikros metodu parenkamas geriausias iš antrame punkte gautų modelių;
4. Aptariamas geriausias modelis, svarbiausi kintamieji, jo prognozių gerumas įvertinamas AUC metrika, sumaišymo matrica, tikslumo, jautrumo ir F1 įverčio, bendrojo tikslumo metrikomis;
5. Svarbiausi kintamieji įvertinami iš dalykinės srities pusės.

3.1. Duomenys

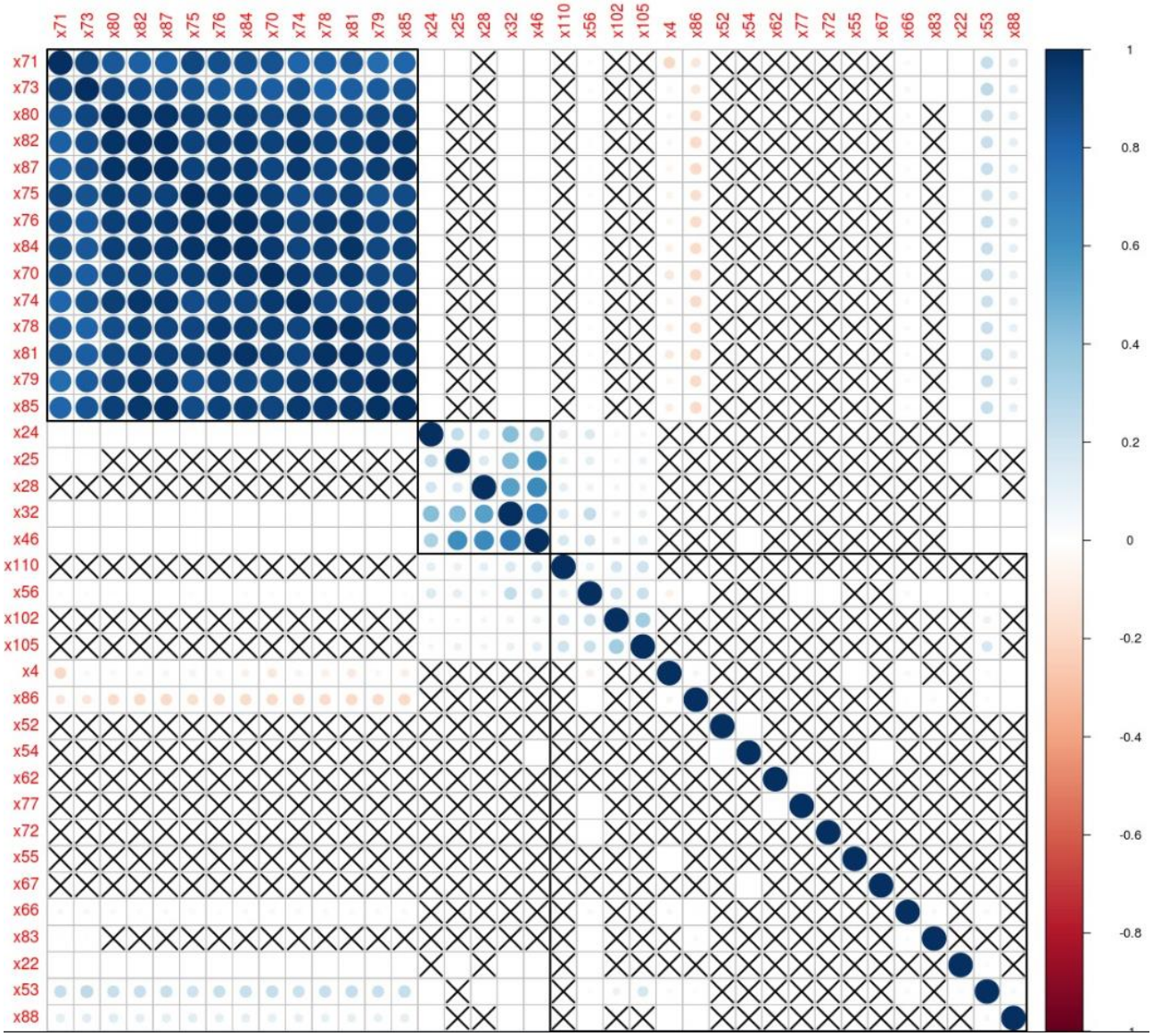
Duomenys šiam tyrimui buvo gauti iš vieno iš Skandinavijos šalių bankų. Tai yra agreguoti bankų elgsenos ir mažų ir vidutinių įmonių, kurios turi įsiskolinimų bankui, atliktų įmokų duomenys. Tyrimui naudojami 182 kintamųjų apie įvairių pranešimų išsiuntimo kiekius,

informacija apie vėlavimą sumokėti įmokas, įmokų dydžius paskolų mokėjimo laikotarpiu. Imtyje yra 128336 stebėjimų tarp jų 2057 fiksuotų bankroto atvejų ir 126279 ne bankroto. Šioje vietoje problema iškiltų dėl didelio disbalanso tarp klasių, todėl tyrimui parenkami algoritmai, kurie gan tiksliai identifikuoja esamas išskirtis neatliekant papildomų transformacijų duomenims.

Skaitiniai kintamieji su kuriais bus dirbama (atlikus modifikacijas aprašytas 3.2 Duomenų paruošimas skyriuje) gali būti grupuojami į kelis klasterius, kurie pavaizduoti 3.1 paveikslėlyje klasterių dendrograma pagal J. H. Vardo pasiūlytą hierarchinį klasterizavimą ir 3.2 paveikslėlyje kintamųjų grupę apibrėžiant stačiakampiu koreliacijos matricoje: pirmasis, susidarantis kairėje pusėje iš smarkiai koreliuojančių kintamųjų, klasteris susijęs su vidutinėmis išlaidomis ir įplaukomis per paskutinius 1, 4, 8, 12 mėnesių, neįskaitant perlaidų tarp vidinių sąskaitų. Antrasis kintamųjų klasteris, esantis centre, sudarytas iš informacijos apie banko siunčiamus pranešimus dėl vėlavimo arba fiksuotą įmonių vėlavimą sumokėti įmoką. Trečiasis skaitinių kintamųjų klasteris, sudarytas iš visų kitų įmonių piniginių srautų valdymo, t.y. įplaukų ir išlaidų, įsiskolinimo dydžio, atliktų planuotų ir neplanuotų paskolos įmokų, šių rodiklių santykinių dydžių ar vidurkių per paskutinius 1, 4, 8, 12 mėnesių.



3.1 pav. Skaitinių kintamųjų dendrograma



3.2 pav. Koreliacijos, reikšmingų skirtumų, klasterizavimo skaitinių kintamųjų grafikas

3.2 paveikslėlyje mėlyna spalva pažymėta stipri teigiama koreliacija, raudona stipri neigiama koreliacija, „X“ ženklu statistiškai reikšmingas skirtumas tarp kintamųjų, kvadratai esantys matricos viduje žymi sudarytus klasterius pagal J. H. Vardo hierarchinį klasterizavimą.

3.2. Duomenų paruošimas

Duomenų paruošimui naudota Python 3.5.2 programavimo kalba parašytą kodą pateiktą 5.3 priede. Tam, kad duomenų rinkinį būtų galima naudoti su visais klasifikatoriais buvo atlikti nurodyti duomenų rinkinio pakeitimai:

1. 138 kintamieji, kurie turėjo daugiau nei 70 % trūkstamų reikšmių pakeisti į binarinius kintamuosius pagal taisyklę:

$$x = \begin{cases} 0, kai\ x\ tr\bar{u}kstama\ reikšmė \\ 1\ ,\ \ priešingu\ atveju \end{cases} \quad \textbf{(3.1)}$$

2. Išskiriami 145 kategoriniai ir 37 skaitiniai kintamieji, kuriems atitinkamai suvienodinami duomenų tipai į tekstinį ir realų skaičių.
3. Vietoje trūkstamų reikšmių 37 skaitiniams kintamiesiems priskiriamas to kintamojo apmokymo imties vidurkis.
4. Kadangi, naudotas ekstremalaus gradientinio sustiprinimo medžių algoritmas geba apdoroti tik skaitines įėjimo reikšmes – visi kategoriniai kintamieji pakeičiami į binarinius pseudo kintamuosius. 145 kategoriniai kintamieji pakeisti į 349 binarinių pseudo kintamųjų, kurie įgauna reikšmę lygią 1 prie konkrečios vienos kategorijos ir reikšmę lygią 0 visais kitais atvejais.
5. Skaitinių kintamųjų standartizuotos ir normalizuotos t.y. iš kintamojo reikšmės atimtas vidurkis ir padalinta iš kintamojo standartinio nuokrypio, kad visi kintamieji būtų vertinami intervale [0; 1].

3.3. Modeliavimas

Šioje dalyje aprašomas metodų parametrų derinimas ir kryžminė patikra geriausiems modeliams. Šie veiksmai buvo atliekami naudojant R 3.2.3 versijos programinę įrangą ir programinį kodą, kuris pateiktas 5.4 priede.

3.3.1. Parametrų derinimas

Siekiant, kad modelis kuo geriau atskirtų klases jis turi būti kuriamas derinant metodų parametrus. Kaip Deividas Volpertas 1996 metais (41) teigė, kad nėra vieno algoritmo, kuris visais atvejais geriau prognozuotų. Taip pat nėra universalaus metodo parametrų rinkinio, su kuriuo būtų gaunami geresni prognozės rezultatai. Todėl kiekvienam metodui randame geriausius parametrų rinkinius:

1. Atsitiktinis miškas – modeliai sudaromi esant skirtingiems medžių skaičiams iš pradžių didinant po 200 medžių, vėliau po 300 iki kol bus pastebimas nereikšmingas modelio pagerėjimas. Fiksuoti parametrai: atsitiktinai atrenkamų kintamųjų kiekis kiekvienam medžiui – 18. Klasių svoriai – 0,98396 ne bankroto klasei ir 0,016036 bankroto klasei. Slenksčio derinimo parametras (angl. cutoff), kurio pagalba priskiriant bankroto tikimybę atsižvelgiama į klasių disbalansą – vektorius 0,98396 ir 0,016036. Parinkta stratifikuota poimčių atranka. Rezultatai pateikiami 3.1 lentelėje.

3.1 lentelė. Atsitiktinio miško metodo parametrų derinimas

Medžių kiekis	AUC metrikos vertė
1400	0,7865819

1100	0,7863391
800	0,7822562
500	0,7698119
300	0,7665372
100	0,7421518

Didinant medžių kiekio parametrą gaunami vis geresni rezultatai. Tačiau pridėjus papildomus 300 sprendimų medžių gaunami minimalus pagerėjimas, todėl procesas stabdytas ir geriausiu atsitiktinių miškų AUC įverčiu laikoma 0,78658 reikšmė.

2. Neuroniniai tinklai – modeliai sudaromi esant skirtingiems neuronų skaičiams trijuose paslėptuose sluoksniuose. Jeigu sluoksnis egzistuoja, tuomet viename sluoksnyje neuronų kiekis kinta intervale [1; 10]. Dešimt geriausių rezultatų pateikiami 3.2 lentelėje.

3.2 lentelė. Neuroninių tinklų metodo parametrų derinimas

Paslėptų sluoksnių vektorius	AUC metrikos vertė
2, 2, 0	0,7833663477
3, 8, 0	0,7806674547
3, 5, 3	0,7754503788
1, 5, 3	0,7750771501
1, 1, 5	0,7722803813
5, 5, 0	0,7720102672
1, 7, 1	0,7720098350
4, 10, 0	0,7705931323
1, 0, 0	0,7687631
5, 6, 0	0,76768337

Geriausias rezultatas gaunamas su sąlyginai nedideliu neuroniniu tinklu turinčiu du paslėptuosius sluoksnius ir po du neuronus kiekviename jų. Geriausias AUC įvertis lygus 0,78337 yra truputi mažesnis nei atsitiktinio miško atveju.

3. XGBoost – modeliai sudaromi esant skirtingiems sprendimų medžių gyliams intervale [2; 6], mokymosi parametrai (intervale [0,1;1]), L1 reguliarizacijos parametrui kintant intervale [0,1; 1], nuo nekonservatyvaus iki konservatyvaus modelio. Fiksuoti parametrai: maksimalus medžių skaičius lygus 100 fiksuojamas, nes XGBoost algoritmas prie didelio medžių kiekio persimoko, paralelių skaičiavimo gijų

naudojimas, stiprinimo stadijų skaičiui priskirta reikšmė 10. Dešimt geriausių rezultatų pateikiami 3.3 lentelėje.

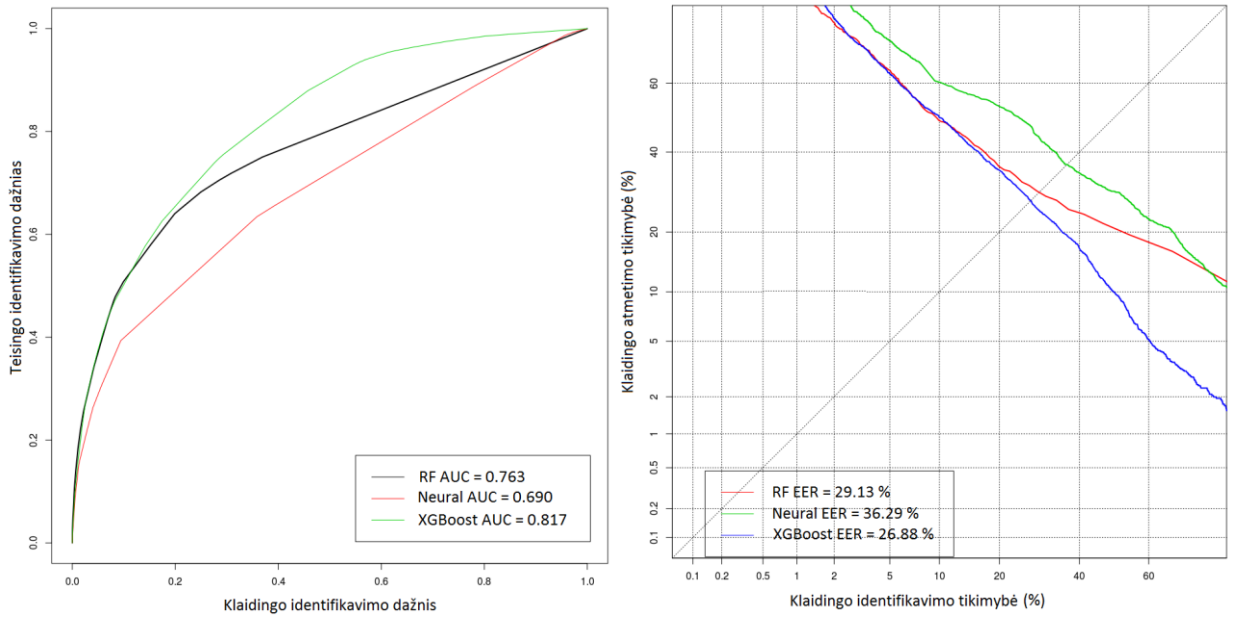
3.3 lentelė. XGBoost metodo parametrų derinimas

Medžių gylis	Mokymosi greitis	L1 reguliarizacijos parametras	AUC metrikos vertė
5	0,8	0,3	0,8309436
4	0,7	0,6	0,8290390
5	0,8	0,6	0,8286309
5	0,7	0,8	0,8275915
6	0,7	0,2	0,8274257
3	0,8	1	0,8273927
3	0,8	0,9	0,8273362
3	0,8	0,8	0,8272829
4	0,9	0,3	0,8272084
3	0,9	0,2	0,8254609

Geriausias modelis šiuo atveju turintis AUC reikšmę 0,83943 yra 10 sprendimo medžių, kuriu gyliai 4-5 lygiai, bei reguliarizacijos parametras L1 = 0,3, reiškiantis, kad modelis yra ne konservatyvus. Šis modelis pasirodė geriausiai derinimo metu.

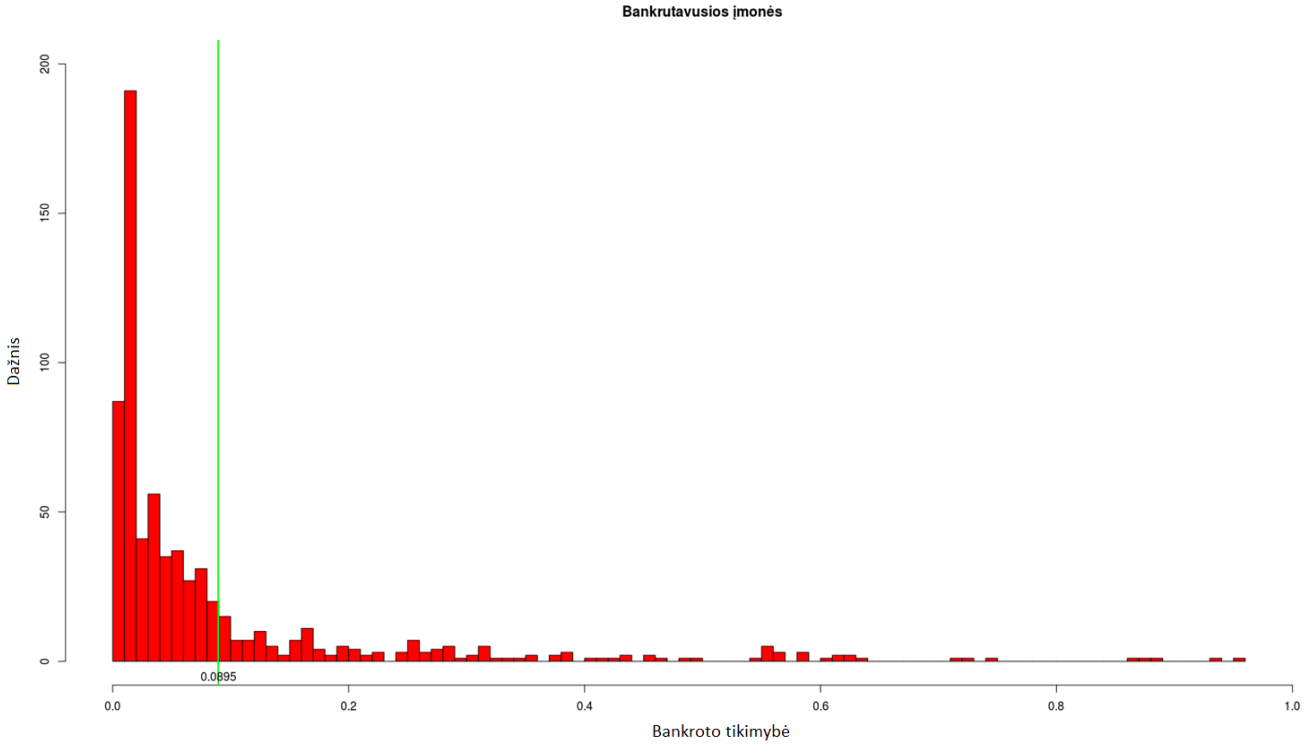
3.3.2. Kryžminės patikra ir rezultatai

Išrinkus geriausius parametrų rinkinius kiekvienam mašininio mokymosi algoritmui atliekama 5 dalių kryžminė patikra. Ir kiekvienam nubraižomos vidutinės metodo ROC kreivės. Apskaičiuojama vidutinė AUC reikšmė.



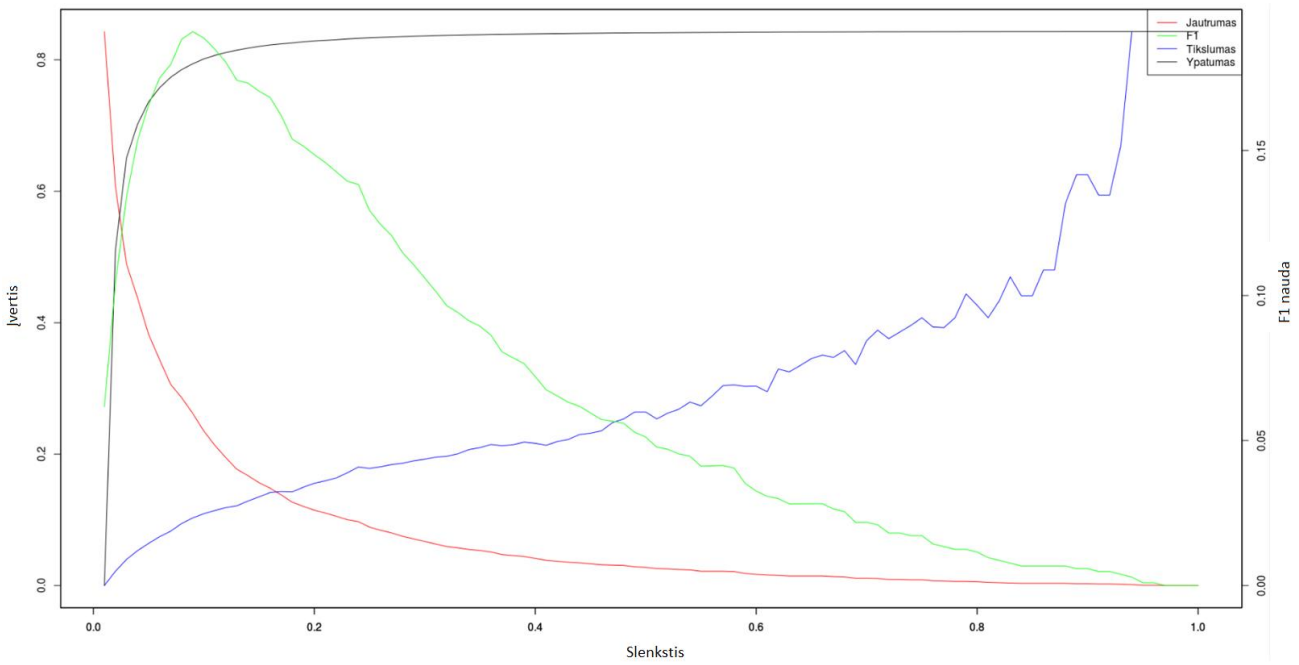
3.3 pav. Kryžminės patikros ROC ir DET kreivės su AUC ir EER metrikomis

Pagal AUC metriką daroma išvada, kad XGBoost algoritmas (AUC = 0,817, EER = 26,88 %) lenkia Neuroninius tinklus (AUC = 0,69, EER = 36,29 %) ir Atsitiktinį mišką (AUC = 0,817, EER = 29,13 %). Kadangi ROC kreivė gaunama išbandžius visas galimas bankroto tikimybės slenksčio, nuo kurio būtų fiksuojamas bankroto įvykis, reikšmes. Norint gauti kiekybinius teisingai identifikuotų ir atmestų įvykių išraiškas reikia parinkti slenkstį.



3.4 pav. Testavimo imties bankrutavusių įmonių prognozuojamos bankroto tikimybes histograma grafikas

Nubraižius testavimo imties bankrutavusių įmonių poaibio prognozuojamų bankroto tikimybių histogramą matoma, kad tikimybių vidurkis 0,0895 yra didesnis už visų nebankrutavusioms įmonėms priskirtų tikimybių vidurkį – 0,0155. Atsižvelgiant į duomenų disbalansą reikia konservatyvesnio slenksčio, nei teisingai identifikuotų įvykių tikimybės vidurkis. Tam naudojamos metodinėje dalyje aprašytos jautrumo ir ypatumo (žr. 3.5 paveikslėlyje) metrikos, kurių pagalba galima nustatyti optimalų slenkstį.



3.5 pav. Jautrumo, tikslumo, ypatumo ir F1 įverčių grafikas

Slenksčio reikšmė parenkama pagal jautrumo ir ypatumo metrikų susikirtimo tašką. Slenksčiui priskiriama reikšmė lygi 0,013. Gaunama sumaišymų matrica:

3.4 lentelė. XGBoost kryžminės patikros sumaišymų matrica, kai slenkstis lygus 0,013

		Prognozė		**Iš viso atvejų**
		1	**0**	
Realios klasės	**1**	1526	531	2057
	0	35376	90903	126279

Naudojant sudarytą sumaišymų matricą galima daryti kelias išvadas. Pirma, turint nesubalansuotą imtį labai lengva gauti vienų metrikų gerus įverčius ir joms priešingų metrikų prastus įverčius. Šiuo atveju apskaičiavus ypatumo metriką, kuri yra lygi 0,72, galima interpretuoti kaip 72 % nebankrutavusių buvo identifikuota teisingai, t.y. 28 % nebankrutavusių įmonių buvo identifikuotos kaip bankrutavusios. Jautrumo metrika šiuo atveju lygi 0,74, tai reiškia, kad 74 % bankrutavusių įmonių buvo identifikuotų teisingai. Šios dvi metrikos nusako, kokioms klasių dalims siūloma priskirti bankrotą, tai būtų griežtesnis paskolų išdavimo filtras, kurio laikantis būtų teisingai identifikuojamos 74 % įmonių, kurioms gresia bankrotas, kita vertus būtų siūloma neduoti paskolos 28 % nebankrutuojančių įmonių.

Apskaičiavus bendrąjį tikslumą, kuris parodo visų teisingai atliktų prognozių santykį su visų stebėjimų kiekiu, gaunama vertė lygi 0,72, tai leistų teigti, kad 72 % atvejų modelis prognozuoja tiksliai.

Norint modelį naudoti taip, kad dėmesys būtų orientuotas į teisingą bankrutuojančių įmonių identifikavimą reikėtų didinti slenksčio reikšmę. Kitas naudojamas slenksčio parinkimo įvertis –

lokalus F1 kreivės maksimumas, kai slenksčio reikšmė = 0,15, šis abscisių ašies taškas beveik sutampa su jautrumo ir tikslumo kreivių sankirta ties slenksčiu 0,165. Gaunama sumaišymų matrica:

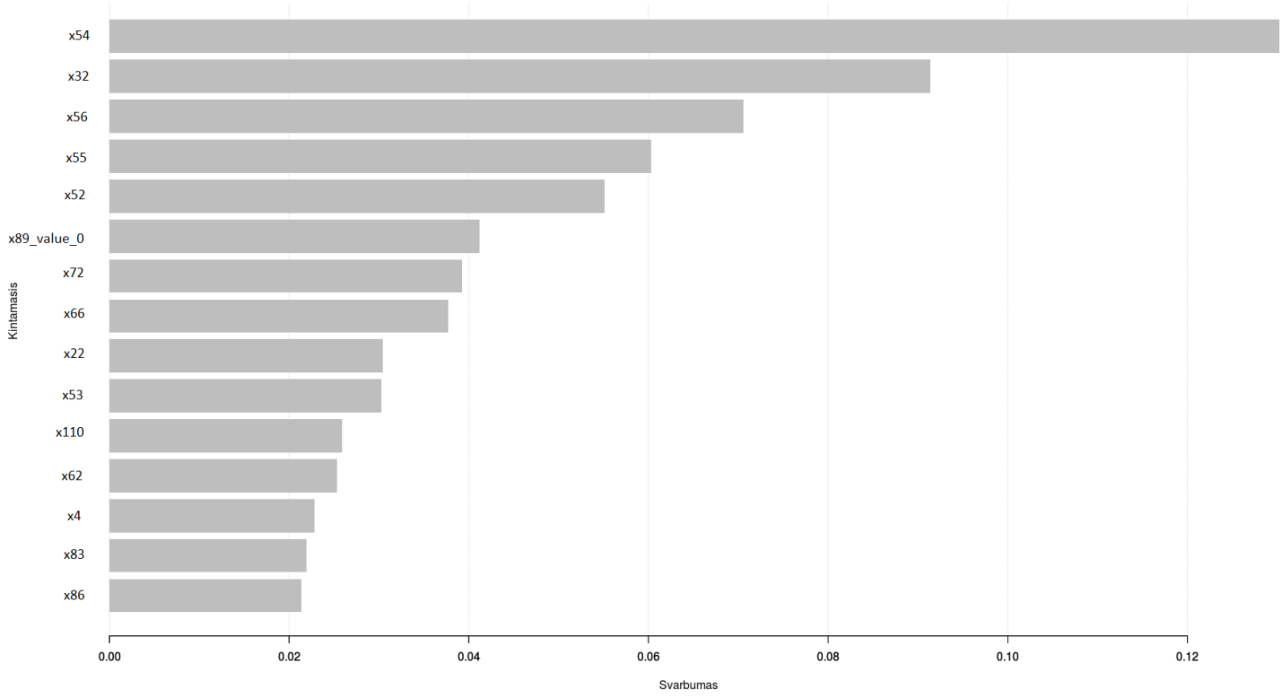
3.5 lentelė. XGBoost kryžminės patikros sumaišymų matrica, kai slenkstis lygus 0,165

		Prognozė		**Iš viso atvejų**
		1	**0**	
Realios klasės	**1**	305	1752	2057
	0	1263	125016	126279

Šiuo atveju apskaičiavus ypatumo metriką, kuri yra lygi 0,988, galima teigti, kad klasifikatorius beveik neklysdamas identifikuoja nebankrutuosiančias įmones. Taip pat, apskaičiavus bendrąjį tikslumą, kuris parodo visų teisingai atliktų prognozių santykį su visų stebėjimų kiekiu, gaunama vertė lygi 0,976, tai leistų teigti, kad 97,6 % atvejų modelis prognozuoja tiksliai. Tačiau šioje vietoje svarbiausia kaip identifikuojama bankroto klasė ir tai parodo tikslumo ir jautrumo metrikos, kurių apskaičiuotos vertės 0,19 ir 0,15 atitinkamai. Šios metrikos interpretuojamos, kaip procentinė išraiška teisingai identifikuotų bankroto įvykių santykiai su visais prognozuojamais įvykiais arba su visais realiais bankroto įvykiais atitinkamai. Tai reiškia, kad 18,9 % įmonių, kurioms modelis prognozavo bankrotą iš tikrųjų bankrutavo. Palyginus su testavimo imties duomenų disbalansu (0,016 % bankroto atveju ir 0,984 % ne bankroto atveju), modelis turintis tokį tikslumą leidžia pagerinti prognozuojamų bankrotų tikslumą 8,4 karto, t.y. iš modelio atrinktų įmonių 19 % bankrutuos, kai remiantis paprasta įmonių statistika galima pasakyti, kad 1,6 % įmonių esančių imtyje bankrutuos.

3.3.3. Kintamųjų svarba

Mašininio mokymosi algoritmai kartu su rezultatais grąžina ir sąrašus kintamųjų, kurie buvo svarbiausi atliekant klasifikavimą. Tai reiškia, kad šių kintamųjų pokytis turi daugiausiai įtakos ne tik sudarant modelį, bet ir iš dalykinės srities pusės. Išskirsime kelis kintamuosius, kuriuos nurodė, geriausiai klases atskyręs, XGBoost modelis. Penkiolikos svarbiausių kintamųjų sąrašas pateiktas 3.6 paveikslėlyje ir 3.6 lentelės apraše žemiau. Penkiasdešimties dažniausiai naudotų kintamųjų, reliatyviai lyginant su pirmuoju ir sugrupuojant į 8 klasterius, grafinė vizualizacija pateikta 5.2 priede.



3.6 pav. XGBoost 15 svarbiausių kintamųjų sąrašas

3.6 paveikslėlyje pavaizduota penkiolikos kintamųjų atnešamos naudos (angl. gain) grafikas, t.y. pavaizduojama kiek pagerėja bendrasis tikslumas panaudojant konkretų kintamąjį medyje. Prieš atliekant papildomą dalinimą pagal kintamąjį medyje atlikti dalinimai gali būti nusakyti kokybės įvertinimo metrikomis, todėl galima įvertinti ir kaip pagerėjo tikslumas atlikus dar vieną dalinimą.

3.6 lentelė. Svarbiausių kintamųjų aprašas

Kintamasis	Atnešama nauda	Aprašymas
x54	0.1302041019	Vidutinis pastarųjų 4 mėnesių einamosios sąskaitos limito panaudojimo procentas nuo bendros limito sumos.
x32	0.0914018414	Priminimų skaičius (svertinis rodiklis) per pastaruosius 12-a mėnesių.
x56	0.0706143712	Vidutinis vėluojamų sumokėti dienų skaičius per pastaruosius 8 mėnesius.
x55	0.0602669996	Likutis sąskaitoje pastarąjį mėnesį lyginant su vidutine įplaukų suma per pastaruosius 12 mėnesių.
x52	0.0551194611	Vidutinis pastarųjų 4 mėnesių einamosios sąskaitos ir kortelių sąskaitų limito panaudojimo procentas nuo bendros limito sumos.
x89	0.0411906244	Maksimalus skaičius dienų, kuomet vėluojama sumokėti įmoką per pastaruosius 4 mėnesius (kai įmoka didesnė nei 50 eurų)
x72	0.0392363493	Pastarųjų 4-ių mėnesių debeto apyvarta lyginant su 5-12 mėnesių apyvarta, atmetant didžiausias stebėtas vertes ir pinigų pervedimus

		tarp vidinių sąskaitų.
x66	0.0377288709	Vidutiniai pavieniai indėliai visoms sąskaitoms, 4-ių mėnesių laikotarpiu. Suskaičiuota įmonei, esančiai aukščiausiai verslo įmonių grupės hierarchijoje.
x22	0.0304250333	Vidutiniai pavieniai indėliai visoms sąskaitoms, 12-os mėnesių laikotarpiu. Suskaičiuota visiems privatiems asmenims grupės struktūroje turintiems daugiau nei 50% nuosavybės teisės.
x53	0.0302571612	Vidutiniai pavieniai indėliai visoms sąskaitoms, 4-ių mėnesių laikotarpiu.
x110	0.0259346922	Vėluojamų įmokų dienų vidurkio palyginimas tarp pastarųjų 4 mėnesių ir prieš tai buvusių 4 mėnesių laikotarpiu.
x62	0.0252979093	Indėlių vidurkis/ vidutinės įplaukos atmetant paskutinį didžiausią įplaukų stebėjimą, per pastaruosius metus.
x4	0.0228499208	Įsiskolinimų dydis einamosiose sąskaitose.
x83	0.0219028078	Įplaukų pokytis per pastaruosius 4-is mėnesius lyginant su 5-8 mėnesiais.
x86	0.0213295594	Įsiskolinimai pagrindinėse kliento sąskaitose.

Didžioji dalis stipriausių kintamųjų yra skaitiniai ir susiję su skolininko vėlavimo arba banko siunčiamų perspėjimų dėl vėlavimo statistika. Taip pat, svarbūs rodikliai yra susiję su santykiu tarp naudojamo kredito sumos ir įmonės turimų įplaukų, išlaidų.

3.4. Empirinės dalies rezultatų apibendrinimas

Prieš modeliavimą visi kintamieji buvo paversti į skaitinius t.y. atrinkti labai reti įvykiai paverčiami į binarinius egzistavimo kintamuosius, kategoriniai kintamieji, turintys n kategorijų, paverčiami į n-1 binarinį kintamąjį. Tuomet, skaitiniams atributams užpildomos trūkstamos reikšmės kintamųjų vidurkiais, standartizuojami ir normalizuojami duomenys atimant vidurkį ir dalinant iš standartinio nuokrypio.

Įmonių bankroto modeliavimo, naudojant elgsenos duomenis, uždaviniui spręsti buvo parinkti trys metodai – neuroniniai tinklai, atsitiktinis miškas, ekstremalaus gradientinio sustiprinimo medžių algoritmas (XGBoost). Atlikus parametrų derinimą ir penkių dalių kryžminę patikrą geriausiai klasifikuojantis modelis buvo sudarytas naudojantis XGBoost algoritmu (AUC = 0,817, EER = 26,88 %), o atsitiktinio miško ir neuroninių tinklų AUC reikšmės 0,763 ir 0,69 atitinkamai. Taip pat, parinkus optimalų tikimybės slenkstį, pagal ypatumo ir jautrumo kreivių sankirtos tašką, gaunamos metrikų reikšmė lygios 0,72 ir 0,74

atitinkamai, reiškia, kad teisingai identifikuojami 74 % bankrutavusių ir 72 % nebankrutavusių, todėl vadovaujantis modelio paskolos išdavimo metu, būtų atsakoma 74 % rizikingų įmonių ir 28 % nerizikingų įmonių.

Parenkant ne tokį konservatyvų slenksčio variantą lygų 0,165 gaunama tikslumo metrikos reikšmė lygi 0,19. Tai rodo, kad modelis 8,4 karto geriau prognozuoja mažesniąją, bankroto įvykio, klasę nei atsitiktinis spėjimas. Tai padėtų patikimiau išfiltruoti mažesnę rizikingų klientų dalį.

Jeigu būtų galima įvertinti patirtus vidutinius nuostolius, kuriuos patiria bankas, kai įmonė bankrutuoja negražinus paskolos, arba vidutinį uždarbi iš sėkmingai veiklą vykdančių įmonių, tuomet slenksti galimą parinkti maksimizuojant pelno funkciją naudojant EMP įvertį (angl. expected maximum profit) (42), (43).

Atlikus modelyje naudojamų kintamųjų, kurie sudarymo metu nešė daugiausiai informacijos apie įmonės bankrotą, pastebėta, kad vieni svarbiausių rodiklių yra susiję su įmonių vėlavimu atlikti įmokas arba banko pranešimų išsiuntimo dėl vėlavimo pastarojo laikotarpio statistika. Šie kintamieji priklauso 3.1 skyriuje aprašytam kintamųjų apie vėlavimą klasteriui. Šiai kintamųjų grupei priklausantys reikšmingi kintamieji:

- Priminimų skaičius (svertinis rodiklis) per pastaruosius 12-a mėnesių.
- Vidutinis vėluojamų sumokėti dienų skaičius per pastaruosius 8 mėnesius.
- Egzistavimo sąlyga nurodanti ar buvo fiksuotas dienų skaičius, kuomet vėluojama sumokėti įmoką per pastaruosius 4 mėnesius (kai įmoka didesnė nei 50 eurų).
- Vėluojamų įmokų dienų vidurkio palyginimas tarp pastarųjų 4 mėnesių ir prieš tai buvusių 4 mėnesių laikotarpiu.

Šie kintamieji yra tiesiogiai susiję su Skandinavijos banko renkama informacija, kuri ir apibrėžiama kaip elgsenos duomenys. Tradiciniai modeliai aprašyti pirmojoje darbo dalyje neleidžia pasinaudoti tokia informacija, nors atliktas tyrimas parodo, kad ji yra reikšminga sprendžiant bankroto prognozavimo uždavinį. Todėl tai patvirtina šiame tyrime naudojamų metodų reikalingumą ir veiksmingumą. Tačiau išlieka apribojimas, kad modeliai sukurti naudojant šiuos metodus yra pritaikomi tik įmonėms, kurių aprašytus kintamuosius galime fiksuoti. Tai reiškia, kad susiauriname taikymo sritį ir modelius aktualu atnaujinti tiek kintant laikui, tiek keičiantis fiksuojamiems kintamiesiems.

Minėtų elgsenos kintamųjų grupė buvo nežymiai reikšmingesnė už kito kintamųjų pogrupio mažų ir vidutinių įmonių elgesį su turimu kapitalu, kurį nusako kintamieji susiję įmonės lėšų valdymu ir ypač pinigų panaudojimu lyginant su turimomis įplaukomis ir įsiskolinimu. Keli reikšmingiausi šios grupės kintamieji:

- Vidutinis pastarųjų 4 mėnesių einamosios sąskaitos limito panaudojimo procentas nuo bendros limito sumos.
- Likutis sąskaitoje pastarąjį mėnesį lyginant su vidutine įplaukų suma per pastaruosius 12 mėnesių.
- Vidutinis pastarųjų 4 mėnesių einamosios sąskaitos ir kortelių sąskaitų limito panaudojimo procentas nuo bendros limito sumos.
- Pastarųjų 4-ių mėnesių debeto apyvarta lyginant su 5-12 mėnesių apyvarta, atmetant didžiausias stebėtas vertes ir pinigų pervedimus tarp vidinių sąskaitų.

Ši informacija glaudžiai susijusi su kito klasterio kintamaisiais, kurie nepateko tarp penkiolikos reikšmingų kintamųjų. Tai buvo 3.1 skyriuje aprašytas pirmasis kintamųjų klasteris, susidarantis iš smarkiai koreliuojančių kintamųjų, šis klasteris vienareikšmiškai susijęs su vidutinėmis išlaidomis ir įplaukomis per paskutinius 1, 4, 8, 12 mėnesių, neįskaitant perlaidų tarp vidinių sąskaitų. Tai rodo, kad informacija apie turimas įplaukas, išlaidas nėra labai svarbi, kol jos nepalyginame su turimais įsipareigojimais arba neimame jų santykinių dydžių. Ši įžvalga išplaukianti iš atlikto tyrimo glaudžiai susijusi su klasikiniais bankroto tikimybės vertinimo modeliais, kuriuose pagrindinė, naudingiausia informacija buvo – santykiniai finansiniai rodikliai, kurie gaunami iš pelno, nuostolių ir balanso ataskaitų. Skaičiuojant santykinius rodiklius geriau įvertinama įmonės dinamika, nepriklausomai nuo jos dydžio, taip pat sumažinamas tarp industrinis skirtumas rodiklių atžvilgiu. Ši savybė išlieka svarbi ir kuriant naujus išvestinius kintamuosius, naudojant slenkamus kelių mėnesių vidurkius ir lyginant juos su einamojo mėnesio rodikliais.

Tarp svarbiausių kintamųjų nepateko kategoriniai kintamieji ar kintamojo egzistavimo sąlygą nurodantys kintamieji, kurių gavimas aprašytas 3.2 skyriuje. Daroma išvada, kad retai įvykstantys įvykiai, šiame tyrime nėra reikšmingi.

Sudarytas modelis gali būti atvaizduotas kaip taisyklių sąrašas, todėl gilesnei analizei būtų galima žiūrėti kaip svarbiausi kintamieji įkomponuojami į šiuos taisyklių sąrašus, prie kokių sąlygų jie tampa svarbūs ir kokie duomenų dalinimo pagal kintamąjį taškai parenkami. Nors modelis sudarytas iš daugelio taisyklių, tokia analizė padėtų išskirti pačias pagrindines sąsajas, į kurias būtų galima reaguoti nenaudojant viso modelio.

IŠVADOS

1. Atlikus susijusių darbų literatūros apžvalgą daromos išvados, kad didėjant surenkamų duomenų kiekiui ir įvairovei mašininio mokymosi metodų pagrindu kuriamų bankroto tikimybės modelių taikymas yra efektyvesnis nei klasikinių bankroto prognozavimo modelių. Nors mašininio mokymosi pagrindu sukurti modeliai yra pritaikyti siauresniam rinkos segmentui ar geografinei vietovei, tai padidina tikslumą ir patikimumą.
2. Bankroto įvykių klasifikavimui ir bankroto tikimybės įvertinimui, naudojantis banko elgsenos ir skolininkų atliekamų transakcijų duomenimis, kurie yra nesubalansuoti ir turi multikolinearumo problemą, siūloma naudoti atsitiktinio miško, neuroninių tinklų ir ekstremalaus gradientinio sustiprinimo medžių algoritmus, kurie veikia nepaisydami šių duomenų trūkumų.
3. Atlikus empirinį tyrimą su Skandinavijos banko elgsenos duomenimis naudojantis parinktus metodus daromos išvados, kad po parametrų derinimo ir penkių dalių kryžminės patikros geriausiai klasifikuojantis modelis buvo sudarytas naudojant XGBoost algoritmą (AUC = 0,817, EER = 26,88 %), o atsitiktinio miško ir neuroninių tinklų AUC reikšmės 0,763 ir 0,69 atitinkamai.
4. Parinkus optimalų tikimybės slenkstį lygų 0,013, gaunamos ypatumo ir jautrumo metrikų reikšmės lygios 0,72 ir 0,74 atitinkamai, reiškia, kad teisingai identifikuojami 74 % bankrutavusių ir 72 % nebankrutavusių, todėl vadovaujantis modelio paskolos išdavimo metu, būtų atsakoma 74 % rizikingų įmonių ir 28 % nerizikingų įmonių. Parenkant ne tokį konservatyvų slenksčio variantą lygų 0,165 gaunama tikslumo metrikos reikšmė lygi 0,19. Tai rodo, kad modelis 8,4 karto geriau prognozuoja mažesniąją, bankroto įvykio, klasę nei atsitiktinis spėjimas.
5. Atlikus modelyje naudojamų kintamųjų, kurie sudarymo metu nešė daugiausiai informacijos apie įmonės bankrotą, pastebėta, kad svarbiausi rodikliai yra susiję su įmonių vėlavimu atlikti įmokas arba banko pranešimų išsiuntimo dėl vėlavimo pastarojo laikotarpio statistika. Svarbūs elgsenos kintamieji: priminimų skaičius (svertinis rodiklis) per pastaruosius 12-a mėnesių, vidutinis vėluojamų sumokėti dienų skaičius per pastaruosius 8 mėnesius, egzistavimo sąlyga nurodanti ar buvo fiksuotas dienų skaičius, kuomet vėluojama sumokėti įmoką per pastaruosius 4 mėnesius (kai įmoka didesnė nei 50 eurų) ir vėluojamų įmokų dienų vidurkio palyginimas tarp pastarųjų 4 mėnesių ir prieš tai buvusių 4 mėnesių laikotarpiu.

Tradiciniai bankroto modeliai neleidžia pasinaudoti tokia informacija, nors atliktas tyrimas parodo, kad ji yra reikšminga sprendžiant bankroto prognozavimo uždavinį. Todėl tai patvirtina šiame tyrime naudojamų metodų reikalingumą ir veiksmingumą.

4. LITERATŪRA

1. Europos Komisija. [Tinkle] [Cituota: 2017 m. 04 19 d.] http://ec.europa.eu/growth/smes/business-friendly-environment/sme-definition_lt.

2. **Aarno Airaksinen, Henri Luomaranta, Pekka Alajääskö, Anton Roodhuijzen.** Eurostat Statistics Explained. [Tinkle] 2015 m. Rugsėjis. [Cituota: 2017 m. 03 28 d.] http://ec.europa.eu/eurostat/statistics-explained/index.php/Statistics_on_small_and_medium-sized_enterprises.

3. **Uhrig-Homburg, Marliese.** Cash-flow shortage as an endogenous bankruptcy reason. *Journal of Banking and Finance.* 2005 m., p. 1509–1534.

4. **Jurgita Karalevičienė, Rita Bužinskienė.** Moderniųjų bankroto modelių tinkamumo įvertinimasįmonių bankroto diagnozavimui. *Vadyba.* 2012 m., p. 45-54.

5. **Stanislovas Liučvaitis.** Rizikos valdymas ir jos analizės svarba verslo plėtotei. *Verslas: teorija ir praktika.* 2003 m., p. 25-35.

6. **Tomasz R. Bielecki, Marek Rutkowski.** *Credit Risk: Modeling, Valuation and Hedging.* s.l. : Springer, 2002.

7. **Seimas, Lietuvos Respublikos.** *Lietuvos Respublikos įmonių bankroto įstatymas.* 2001.

8. **Kuskytė, Skaistė.** Diskriminantinės analizės bankroto prognozavimo modeliai. Altman Z ir Springate modelių taikymas Lietuvos įmonėms. *Tiltas į ateitį.* Eglė Butkevičienė, 2015 m., T. 9, p. 303-307.

9. *Financial ratios, discriminant analysis and the prediction of corporate bankruptcy.* **Altman, Edward I.** 1968 m., The journal of finance.

10. **Rasa Budrikienė, Irena Paliulytė.** Bankroto prognozavimo modelių pritaikomumas skirtingo mokumo ir pelningumo įmonėms. *Ekonomika ir vadyba: aktualijos ir perspektyvos.* 2012 m., p. 90-103.

11. **Tsai, Chih-Fong.** Combining cluster analysis with classifier ensembles to predict financial. *Information Fusion.* 2014 m., p. 46-58.

12. **A. Vellido, P.J.G. Lisboa, J. Vaughan.** Neural networks in business: a survey of applications (1992–1998). *Expert Systems with Applications.* 1999 m., p. 51-70.

13. **Michael J. Shaw, James A. Centry.** Inductive Learning for Risk Clssification. 1990 m.

14. **Iain Brown, Christophe Mues.** An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications.* 2012 m.

15. **Stefan Lessmann, Bart Baesens, Hsin-Vonn Seow, Lyn C. Thomas.** Benchmarking state-of-the-art classification algorithms for credit scoring: An update of researc. *European Journal of Operational Research.* 2015 m., p. 124-136.
16. **Marcus D. Odom, Ramesh Sharda.** A neural network model for bankruptcy prediction. 1990 m.
17. **Pamela K. Coast, Franklin Fant.** Recognizing Financial Distress Patterns Using a Neural Network Tool. *Financial Management.* 1993 m., p. 142-155.
18. **Kar Yan Tam, Melody Y. Kiang .** Managerial applications of the neural networks: The case of bank failure predictions. *Management Science.* 1992 m., p. 416-430.
19. **Gang Wang, Jian Ma, Lihua Huang, Kaiquan Xu.** Two credit scoring models based on dual strategy ensemble trees. *Knowledge-Based Systems.* 2012 m., p. 61-68.
20. **Imad Bou–Hamad, Denis Larocque, Hatem Ben–Ameur.** Discrete-time survival trees and forests with time-varying covariates: application to bankruptcy data. *Statistical Modelling.* 2011 m., p. 429-446.
21. **Esteban Alfaro, Noelia García, Matías Gámez, David Elizondo.** Bankruptcy forecasting: An empirical comparison of AdaBoost and neural networks. *Decision Support Systems.* 2008 m., p. 110-122.
22. **Jie Sun, Hui Li.** Financial distress prediction using support vector machines: Ensemble vs. individual. *Applied Soft Computing.* 2012 m., p. 2254-2265.
23. **Fernando Sánchez-Lasheras, Javier de Andrés, Pedro Lorca, Francisco Javier de Cos Juez.** A hybrid device for the solution of sampling bias problems in the forecasting of firms' bankruptcy. *Expert Systems with Applications.* 2012 m., p. 7512-7523.
24. **Sumit Agarwal, Chunlin Liu.** Determinants of Credit Card Delinquency and Bankruptcy: Macroeconomic Factors . *Journal of economics and finance.* 2003 m.
25. **David B. Gross, Nicholas S. Souleles.** An Empirical Analysis of Personal Bankruptcy and Delinquency. *The Review of Financial Studies Spring.* 2002 m., p. 319-347.
26. **Mitchell A. Peterson, Raghuram G. Rajan.** Does Distance Still Matter? The Information Revolution in Small Business Lending. *the journal of finance.* 2002 m.
27. **Mitchell Berlin, Loretta J. Mester.** On the pro®tability and cost of relationship lending. *Journal of Banking & Finance.* 1998 m., p. 873-897.
28. **Ghosh, Debashis, et al., et al.** *Calculating credit worthiness using transactional data. US 7,734,539 B2* United States of America, 2007 m.
29. **Thomas B. Fomby, Ph.D.** SEMMA, SAS. [Tinkle] [Cituota: 2017 m. May 1 d.] http://faculty.smu.edu/tfomby/eco5385_eco6380/data/SPSS/SAS%20_%20SEMMA.pdf.

30. **Shearer, Colin.** The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of data warehousing.* 2000 m., T. 5.
31. *Validation of internal rating systems and PD estimates.* **Tasche, Dirk.** 2006 m. May.
32. *Multi-valued attribute and multi-labeled data decision tree algorithm.* **Weiguo Yi, Mingyu Lu, Zhi Liu.** 2, 2011 m., International Journal of Machine Learning and Cybernetics, p. 67-74.
33. *A new node splitting measure for decision tree construction.* **B. Chandra, Ravi Kothari, Pallath Paul.** 2010 m., Pattern Recognition.
34. *Pruning Decision Trees and Lists.* **Frank, Eibe.** 2000 m.
35. *Neural Network Ensembles.* **Lars Kai Hansen, Peter Salamon.** 1990 m., IEEE Transactions on Pattern Analysis and Machine Intelligence.
36. *Generating Accurate and Diverse Members of a Neural-Network Ensemble.* **David w. Opitz, Jude W. Shavlik.** 1996 m., Advances in Neural Information Precessing Systems, p. 535-541.
37. *Popular Ensemble Methods: An Empirical Study.* **David Opitz, Richard Maclin.** 1999 m., Journal of Artificial Intelligence Research 11, p. 169-198.
38. **Bylander, Tom.** Estimating Generalization Error on Two-Class Datasets Using Out-of-Bag Estimates. *Machine Learning.* 2002 m., p. 287-297.
39. *Boosting the margin: A new explanation for the effectiveness of voting methods.* **Robert E. Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee.** 1998 m., Annals of Statistics, p. 1651-1686.
40. **Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani.** *An Introduction to Statistical Learning with Applications in R.* s.l. : Springer, 2013.
41. **Wolpert, David.** The Lack of A Priori Distinctions between Learning Algorithms. *Neural Computation.* 1996 m., p. 1341-1390.
42. **Thomas Verbraken, Cristian Bravo, Richard Weber, Bart Baesens.** Development and application of consumer credit scoring models using profit-based classification measures. *European Journal of Operational Research.* 2014 m., p. 505-513.
43. —. Development and application of consumer credit scoring models using profit-based classification measures. *European Journal of Operational Research.* 2014 m.
44. *Distressed Firm and Bankruptcy Prediction in an International Context: A Review and Empirical Analysis of Altman's Z-Score Model* . **Edward I. Altman, Małgorzata Iwanicz-Drozdowska, Erkki K. Laitinen, Arto Suvas.** 2014 m. August 10 d., p. 47.
45. *MDL-based Decision Tree Pruning.* **Manish Mehta, Jorma Rissanen, Rakesh Agrawal.** 1995 m.

5. PRIEDAI

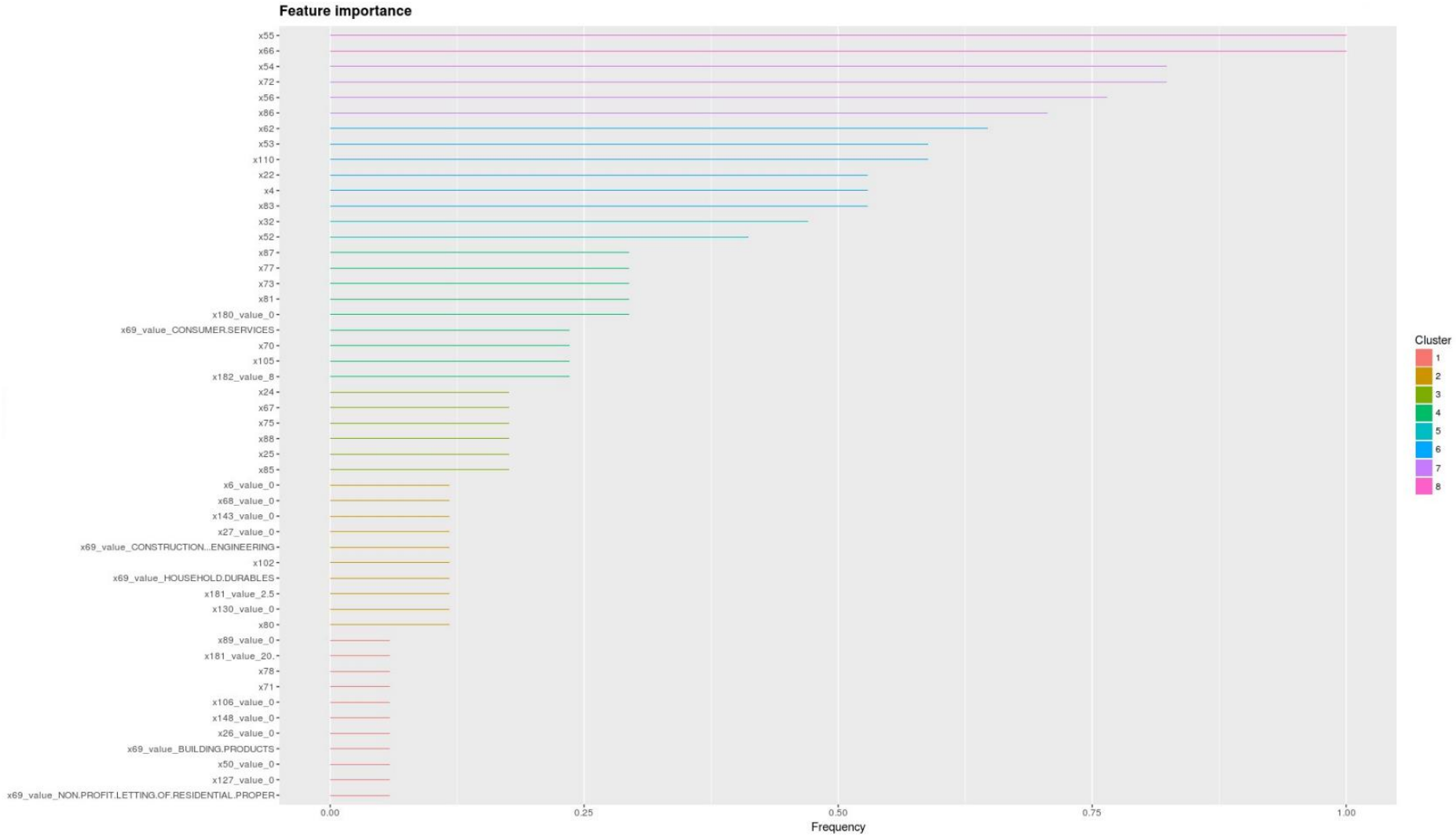
5.1. priedas. ES MVĮ statistika

5.1 lentelė. 2016 metų ES MVĮ statistika.

	Įmonės		VDDE		BPV	
	Bendrai	% MVĮ	Bendrai	% MVĮ	Bendrai	% MVĮ
ES	22 346 729	99.8	133 767 348	67.0	6 184 825	57.5
Belgija	566 006	99.8	2 718 355	70.1	189 086	62.2
Bulgarija	312 608	99.8	1 872 997	75.5	18 246	62.3
Čekija	1 007 441	99.9	3 521 520	69.8	84 142	56.0
Danija	213 358	99.7	1 602 105	65.0	119 936	62.5
Vokietija	2 189 737	99.5	26 401 395	62.5	1 385 501	53.3
Estija	58 408	99.7	393 545	78.1	9 338	74.9
Graikija	726 581	99.9	2 198 986	86.5	54 703	72.8
Ispanija	2 385 077	99.9	10 923 323	73.9	434 156	63.0
Prancūzija	2 882 419	:	15 495 621	:	890 597	:
Kroatija	148 573	99.7	1 002 905	68.3	19 115	54.8
Italija	3 825 458	:	14 715 132	:	646 476	:
Kipras	46 139	99.9	224 915	:	7 864	:
Lietuva	141 893	99.8	835 630	76.2	12 155	68.5
Latvija	91 939	99.8	573 580	78.8	9 269	69.2
Liuksemburgas	29 265	99.5	242 533	68.3	19 250	70.7
Vengrija	528 519	:	2 430 681	:	46 497	:
Malta	26 796	99.8	119 224	79.3	3 548	74.9
Olandija	862 697	99.8	5 359 446	66.7	310 022	62.9
Austrija	308 411	99.7	2 671 477	68.0	164 976	60.5
Lenkija	1 519 904	99.8	8 326 839	68.9	171 627	50.1
Portugalija	793 235	99.9	2 942 895	:	66 360	:
Rumunija	425 731	99.6	3 837 868	66.4	48 432	:
Slovėnija	119 644	99.8	574 479	72.3	17 140	62.8
Slovakija	398 392	99.9	1 417 228	69.7	32 922	60.5
Suomija	226 373	99.7	1 457 599	63.0	86 957	59.6
Švedija	661 822	99.8	3 025 006	65.4	210 859	58.5
Didžioji Britanija	1 703 562	99.7	17 784 620	53.0	1 037 293	50.9
Norvegija	278 899	99.8	1 510 838	67.6	230 661	58.6

: Trūksta duomenų

5.2. Kintamųjų svarba XGBoost



5.1 pav. XGBoost 50 dažniausiai naudotų kintamųjų sąrašas

5.3. Duomenų paruošimo kodas „Python“ kalba

```
import pandas as pd
import numpy as np
import sklearn as sk
from sklearn.model_selection import train_test_split
from collections import defaultdict, Counter

from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
import sys

import seaborn as sns

# testPart is a ration between test and train sets. I have set it to 0.0000001 so I would keep all transformet records in train part so I could use them in R.
# exiting this srcipt at line 180

ThreshForNull = 0.7
testPart = 0.00001

trainData = pd.read_csv("dataBehaviorTrain.csv", decimal = ',')
testData = pd.read_csv("dataBehaviorTest.csv", decimal = ',')
rnum = trainData.shape[0]

print(trainData.shape[0], trainData.shape[1])
print(testData.shape[0], testData.shape[1])

testData['default_total'] = 'test'

ml_dataset = trainData

# adding month column
ml_dataset['month'] = pd.DatetimeIndex(ml_dataset['RUN_TS']).month
testData['month'] = pd.DatetimeIndex(testData['RUN_TS']).month

print( 'Base data has {0} rows and {1} columns'.format(ml_dataset.shape[0], ml_dataset.shape[1]))
```

```
print(list(ml_dataset))

names = ['x1']
for i in range(2,ml_dataset.shape[1]+1):
        n = 'x{0}'.format(i)
        names.extend([n])

ml_dataset.columns = names
testData.columns = names
print(list(ml_dataset))

# getting a lists of categorical and numerical features and setting data types accordingly.
# first part of categorical features are made from numerical features, which contais a lot of NULL values in train set.
# if column has bigger part than ThreshForNull NULL values we replace that with presence indicator. 1 if exist, 0 if NULL.
categorical_features = [x for x in list(trainData) if (trainData[x].isnull().sum()/rnum >= ThreshForNull)]

print(len(categorical_features))

for x in categorical_features:
        if (trainData[x].isnull().sum()/rnum >= ThreshForNull):
                print(x, trainData[x].isnull().sum()/rnum)

# features for which we replace the feature by a indicating whether the value was present
for feature in categorical_features:
   ml_dataset[feature] = ml_dataset[feature].map(lambda x: 0 if pd.isnull(x) else 1).astype(np.uint8)
   testData[feature] = testData[feature].map(lambda x: 0 if pd.isnull(x) else 1).astype(np.uint8)

#categorical_features.extend(['LE_UNIT_MASTER', 'RUN_TS', 'MAN_LE_CUSTINFRA_CUSTOMER_TIME_ESTABLISHMENT_DATE_FROM_1', 'MAN_CP_SUM_SHARE_OWSHAR', 'flag_for_df', 'default_total', 'Industry', 'month'])

categorical_features.extend(['x1', 'x2', 'x181', 'x180', 'x8', 'x9', 'x69', 'x182'])

numerical_features = set(list(ml_dataset)).symmetric_difference(set(categorical_features))

print(numerical_features)
```

```
for feature in categorical_features:
        ml_dataset[feature] = ml_dataset[feature].astype('unicode')
        testData[feature] = testData[feature].astype('unicode')

for feature in numerical_features:
        ml_dataset[feature] = ml_dataset[feature].astype('double')
        testData[feature] = testData[feature].astype('double')

print('Number of categorical features', len(set(categorical_features)))
print('Number of numerical features', len(set(numerical_features)))

print(categorical_features)
print(numerical_features)



# creating a target variable mappings
target_map = {'0': 0, '1': 1}
ml_dataset['__target__'] = ml_dataset['x9'].map(str).map(target_map)
#del ml_dataset['x9']
#del testData['x9']
#testData['X__target__'] = 'test'

# remove rows for which the target is unknown.
ml_dataset = ml_dataset[~ml_dataset['__target__'].isnull()]

print( 'after target map data has {0} rows and {1} columns'.format(ml_dataset.shape[0], ml_dataset.shape[1]))

# splitting data to train and test
train, test = train_test_split(ml_dataset, test_size= testPart, random_state=1, stratify = ml_dataset['__target__'])

print( 'Train data has {0} rows and {1} columns'.format(train.shape[0], train.shape[1]))
print( 'Test data has {0} rows and {1} columns'.format(test.shape[0], test.shape[1]))

# filling missing values with means from train set (to keep test set completely unseen for modelling part)
```

```
for feature in numerical_features:
        v = train[feature].mean()
        train[feature] = train[feature].fillna(v)
        #test[feature] = test[feature].fillna(v)
        #testData[feature] = testData[feature].fillna(v)
        #print( 'Imputed missing values in feature {0} with value {1}'.format(feature, v))

for x in categorical_features:
        if (trainData[x].isnull().sum()/rnum >= ThreshForNull):
                print(x, trainData[x].isnull().sum()/rnum)
```

creating dummies for categorical variables (again we are using only train data, this may cause conflicts if test set will contain unseen category.)

```
# a binary column is created for each of the 100 most frequent values.
LIMIT_DUMMIES = 100

categorical_features_dum = set(categorical_features).symmetric_difference(set(['x1', 'x2', 'x9']))

def select_dummy_values(train, features):
    dummy_values = {}
    for feature in categorical_features_dum:
        values = [
            value
            for (value, _) in Counter(train[feature]).most_common(LIMIT_DUMMIES)
        ]
        dummy_values[feature] = values
    return dummy_values

DUMMY_VALUES = select_dummy_values(train, categorical_features)

def dummy_encode_dataframe(df):
    for (feature, dummy_values) in DUMMY_VALUES.items():
        for dummy_value in dummy_values:
            dummy_name = '{0}_value_{1}'.format(feature, dummy_value)
            df[dummy_name] = (df[feature] == dummy_value).astype(float)
        del df[feature]
        #print( 'Dummy-encoded feature {0}'.format(feature))

dummy_encode_dataframe(train)
```

```
#dummy_encode_dataframe(test)

#dummy_encode_dataframe(testData)

# rescale numerical features using mean and std from train set.

for feature in numerical_features:
        shift = train[feature].mean()
        scale = train[feature].std()
        if scale == 0.:
                del train[feature]
                #del test[feature]
                #del testData[feature]
                #print( 'Feature {0} was dropped because it has no variance'.format(feature))
        else:
                #print( 'Rescaled {0}'.format(feature))
                train[feature] = (train[feature] - shift).astype(np.float64) / scale
                #test[feature] = (test[feature] - shift).astype(np.float64) / scale
                #testData[feature] = (testData[feature] - shift).astype(np.float64) / scale


#train_X = train.drop(['__target__', 'LE_UNIT_MASTER', 'RUN_TS'], axis=1)
#test_X = test.drop(['__target__', 'LE_UNIT_MASTER', 'RUN_TS'], axis=1)
train_X = train.drop(['__target__', 'x1', 'x2', 'x9'], axis=1)
#test_X = test.drop(['__target__', 'x1', 'x2'], axis=1)

train_Y = np.array(train['__target__'])
#test_Y = np.array(test['__target__'])

print(train_X.shape[0], train_X.shape[1])
#print(test_X.shape[0], test_X.shape[1])
print(testData.shape[0], testData.shape[1])

train.to_csv("trainx.csv")
#testData.to_csv("testx.csv")

sys.exit(0)
```

5.4. Modeliavimo kodas „R“ kalba

```
#install.packages("randomForest")
#install.packages("FNN")
#install.packages("neuralnet")
#install.packages("RSNNS")
#install_github("davidavdav/ROC")
#install.packages("RRF")
#install.packages("xgboost")
#install.packages("fdrtool")
#install.packages("ggplot2")
#install.packages("devtools")
#install.packages("caret")
#install.packages("DiagrammeR")
#install.packages("Ckmeans.1d.dp")
#install.packages("inTrees")
#install.packages("OptimalCutpoints")
#devtools::install_github("jandob/ccf")
#install.packages("lift")
#install.packages('party')
#install.packages('ROCR')

library(fdrtool)
library(lattice)
library(ggplot2)
library(devtools)
library(ROC)
library(caret)
library(randomForest)
library(MASS)
library(FNN)
library(nnet)
library(neuralnet)
library(RSNNS)
library(RRF)
library(xgboost)
library(DiagrammeR)
library(ccf)
```

```
library(Ckmeans.1d.dp)
library(inTrees)
library(doParallel)
library(OptimalCutpoints)
library(lift)
library(party)
library(ROCR)

rootFolder <- "/home/laimonas/try2/TestTrainSplit"
setwd(rootFolder)

colY <- "X__target__"
D <- read.csv('trainx.csv',header=T)

table(D$X__target__)

myData <- D[,-which(colnames(D) %in% c("x1","x2", "x9", "X"))]

colnames(myData)[colnames(myData) %in% "X__target__"] <- colY
myData[,colY] <- factor(myData[,colY],labels=c("False","True"))

hiddenSize <- c(2, 2)

k <- 5 # number of CV folds
myFolds <- createFolds(myData$X__target__, k)

cl <- makeCluster(detectCores())
registerDoParallel(cl)

for (i in 1:length(myFolds)) {
  tstInd <- myFolds[[i]]
  trnIdx <- as.logical(rep(1,1,nrow(myData)))
  trnIdx[tstInd] <- FALSE
  trnInd <- which(trnIdx)
  write.csv(myData[trnInd,],sprintf('CVfold%d-train.csv',i),row.names=FALSE)
  write.csv(myData[tstInd,],sprintf('CVfold%d-test.csv',i),row.names=FALSE)
}
```

```
stopCluster(cl)


i = 1

trainData <- read.csv(sprintf('CVfold%d-train.csv',i))
trainData[,colY] <- trainData[,colY]
Xtrain <- trainData[,-which(colnames(trainData) %in% colY)]
XtrainNorm <- trainData[,-which(colnames(trainData) %in% colY)]
Ytrain <- trainData[,colY]

testData <- read.csv(sprintf('CVfold%d-test.csv',i))
testData[,colY] <- testData[,colY]
Xtest <- testData[,-which(colnames(testData) %in% colY)]
XtestNorm <- testData[,-which(colnames(testData) %in% colY)]
target <- as.logical(testData[,colY])


dtrain <- xgb.DMatrix(data = as.matrix(Xtrain), label = as.integer(as.logical(Ytrain)))
dtest <- xgb.DMatrix(data = as.matrix(Xtest), label = as.integer(as.logical(testData[,colY])))


myResultsXGB <- NULL

for(i in 2:6){
  for(j in seq(0.1,1, 0.1)){
    #for(k in seq(1,25,4)){
    # xgboost
    bst <- xgboost(data = dtrain, max.depth = i, eta = j, ntreelimit = 100, nthread = 20, nround = 10, alpha
= 0.2, objective = "binary:logistic")
    name <- paste("bst_", i, "_", j)
    model <- rep(name,length(target))
    pred <- predict(bst, dtest)
    score <- pred
    myResultsXGB <- rbind(myResultsXGB,data.frame(model,score,target))
    #}
  }
}
```

```
myModels <- levels(myResultsXGB[,"model"])

myModelNames <- NULL
df <- data.frame(name = character(),
                 auc = double(),
                 stringsAsFactors=FALSE)
performance <- roc.plot(myResultsXGB[myResultsXGB[,"model"]==myModels[i],],i,traditional=TRUE)
myModelNames[i] <- sprintf('%s AUC=%5.3f',myModels[i],1-performance['pAUC'])
df[1,1] <-myModels[i]
df[1,2] <- 1-performance['pAUC']
for (i in 2:length(myModels)) {
  performance <- roc.plot(myResultsXGB[myResultsXGB[,"model"]==myModels[i],],i,traditional=TRUE)
  myModelNames[i] <- sprintf('%s AUC=%5.3f',myModels[i],1-performance['pAUC'])
  df[i,1] <-myModels[i]
  df[i,2] <- 1-performance['pAUC']
}
legend(0.25,0.7,myModelNames,lty=rep(1,1,length(myModels)),col=1:length(myModels))

df[order(-df$auc),]

###################################################################### neuralnet


myResultsXGB <- NULL
for(i in seq(1,10,2)){
  for(j in seq(1,10,2)){
    for(k in seq(1,10,2)){
      hiddenSize <- c(i, j, k)

      # neuralnet function from neuralnet package
      n <- names(trainData)
      # https://www.r-bloggers.com/fitting-a-neural-network-in-r-neuralnet-package/
      f <- as.formula(paste(paste(colY," ~",sep=""), paste(n[!n %in% colY], collapse = " + ")))
      tD <- trainData
      tD$X__target__ <- as.numeric(tD$X__target__)-1
```

```
            neural_model <- neuralnet(f, data = tD, hidden = hiddenSize, threshold = 0.5, err.fct='sse', linear.output = FALSE, lifesign = 'minimal')
            name <- paste("neurnet", i, "_", j, "_", k)
            model <- rep(name,length(target))
            soft <- compute(neural_model, XtestNorm)
            score <- soft$net.result
            myResultsXGB <- rbind(myResultsXGB,data.frame(model,score,target))

          }
        }
      }

      myModels <- levels(myResultsXGB[,"model"])
      i <- 1
      myModelNames <- NULL
      df <- data.frame(name = character(),
                       auc = double(),
                       stringsAsFactors=FALSE)
      performance <- roc.plot(myResultsXGB[myResultsXGB[,"model"]==myModels[i],],i,traditional=TRUE)
      myModelNames[i] <- sprintf('%s AUC=%5.3f',myModels[i],1-performance['pAUC'])
      df[1,1] <-myModels[i]
      df[1,2] <- 1-performance['pAUC']
      for (i in 2:length(myModels)) {
        performance <- roc.plot(myResultsXGB[myResultsXGB[,"model"]==myModels[i],],i,traditional=TRUE)
        myModelNames[i] <- sprintf('%s AUC=%5.3f',myModels[i],1-performance['pAUC'])
        df[i,1] <-myModels[i]
        df[i,2] <- 1-performance['pAUC']
      }
      legend(0.25,0.7,myModelNames,lty=rep(1,1,length(myModels)),col=1:length(myModels))

      df[order(-df$auc),]


      ############################################################################ random forest
```

```
myResultsXGB <- NULL
for(i in seq(800,1400,300)){
  print(i)
  model_classwt <- prop.table(table(Ytrain))
  rf_model <- randomForest(Xtrain, Ytrain, ntree = i,
                           classwt = model_classwt, cutoff = model_classwt,
                           strata = Y, replace = FALSE, importance=FALSE, do.trace = TRUE)
  name <- paste("rf", i)
  model <- rep(name,length(target))
  soft <- predict(rf_model,Xtest,type="prob")
  score <- soft[,2] - soft[,1]
  myResultsXGB <- rbind(myResultsXGB,data.frame(model,score,target))
}

myModels <- levels(myResultsXGB[,"model"])
i <- 1
myModelNames <- NULL
df <- data.frame(name = character(),
                 auc = double(),
                 stringsAsFactors=FALSE)
performance                                                                    <-
roc.plot(myResultsXGB[myResultsXGB[,"model"]==myModels[i],],i,traditional=TRUE)
myModelNames[i] <- sprintf('%s AUC=%5.3f',myModels[i],1-performance['pAUC'])
df[1,1] <-myModels[i]
df[1,2] <- 1-performance['pAUC']
for (i in 2:length(myModels)) {
  performance                                                                  <-
roc.plot(myResultsXGB[myResultsXGB[,"model"]==myModels[i],],i,traditional=TRUE)
  myModelNames[i] <- sprintf('%s AUC=%5.3f',myModels[i],1-performance['pAUC'])
  df[i,1] <-myModels[i]
  df[i,2] <- 1-performance['pAUC']
}
legend(0.25,0.7,myModelNames,lty=rep(1,1,length(myModels)),col=1:length(myModels))

df[order(-df$auc),]
```

```
##############################Cross fold

myResults <- NULL

for (i in 1:k) {
  trainData <- read.csv(sprintf('CVfold%d-train.csv',i))
  trainData[,colY] <- trainData[,colY]
  Xtrain <- trainData[,-which(colnames(trainData) %in% colY)]
  XtrainNorm <- trainData[,-which(colnames(trainData) %in% colY)]
  Ytrain <- trainData[,colY]

  testData <- read.csv(sprintf('CVfold%d-test.csv',i))
  testData[,colY] <- testData[,colY]
  Xtest <- testData[,-which(colnames(testData) %in% colY)]
  XtestNorm <- testData[,-which(colnames(testData) %in% colY)]
  target <- as.logical(testData[,colY])

  model_classwt <- prop.table(table(Ytrain))
  rf_model <- randomForest(Xtrain, Ytrain, ntree = 900,
                           classwt = model_classwt, cutoff = model_classwt,
                           strata = Y, replace = FALSE, importance=FALSE, do.trace = TRUE)
  model <- rep("RF",length(target))
  soft <- predict(rf_model,Xtest,type="prob")
  score <- soft[,2] - soft[,1]
  myResults <- rbind(myResults,data.frame(model,score,target))


  # neuralnet function from neuralnet package
  n <- names(trainData)
  # https://www.r-bloggers.com/fitting-a-neural-network-in-r-neuralnet-package/
  f <- as.formula(paste(paste(colY," ~",sep=""), paste(n[!n %in% colY], collapse = " + ")))
  tD <- trainData
  tD$X__target__ <- as.numeric(tD$X__target__)-1
  neural_model <- neuralnet(f, data = tD, hidden = hiddenSize, threshold = 0.5, err.fct='sse', linear.output
= FALSE, lifesign = 'minimal')
  model <- rep("neural",length(target))
  soft <- compute(neural_model, XtestNorm)
  score <- soft$net.result
```

```
  myResults <- rbind(myResults,data.frame(model,score,target))


  # xgboost
  dtrain <- xgb.DMatrix(data = as.matrix(Xtrain), label = as.integer(as.logical(Ytrain)))
  dtest <- xgb.DMatrix(data = as.matrix(Xtest), label = as.integer(as.logical(testData[,colY])))

  bst <- xgboost(data = dtrain, max.depth = 5, eta = 0.8, ntreelimit = 100, nthread = 20, nround = 10,
objective = "binary:logistic")
  model <- rep("xgboost",length(target))
  pred <- predict(bst, dtest)
  score <- pred
  myResults <- rbind(myResults,data.frame(model,score,target))

}

#####################################    ROC curve
myModels <- levels(myResults[,"model"])

myModelNames <- NULL
i <- 1
performance <- roc.plot(myResults[myResults[,"model"]==myModels[i],],i,traditional=TRUE)
myModelNames[i] <- sprintf('%s AUC=%5.3f',myModels[i],1-performance['pAUC'])
for (i in 2:length(myModels)) {
  performance <- roc.plot(myResults[myResults[,"model"]==myModels[i],],i,traditional=TRUE)
  myModelNames[i] <- sprintf('%s AUC=%5.3f',myModels[i],1-performance['pAUC'])
}
legend(0.55,0.17,myModelNames,lty=rep(1,1,length(myModels)),col=1:length(myModels))

#####################################    det curve

myModelNames <- NULL
det.plot(NULL,1,xmax=75,ymax=75)
for (i in 1:length(myModels)) {
  performance <- det.plot(myResults[myResults[,"model"]==myModels[i],],nr=i+1)
  myModelNames[i] <- sprintf('%s EER=%5.2f%%',myModels[i],performance['eer'])
}
legend(-3,-2.6,myModelNames,lty=rep(1,1,length(myModels)),col=2:(length(myModels)+1))
```

```
################################################ histograms
df1 = as.data.frame(cbind(pred, target))

df1$target

is_def <- df1[df1$target == 1, 'pred']
no_def <- df1[df1$target == 0, 'pred']

hist(no_def, col = 'blue', breaks = 100, xlab = 'Bankroto tikimybÄ—', ylab = 'DaÅ¾nis', main = 'Nebankrutavusios Ä¯monÄ—s', xlim = c(0, 1), ylim = c(0, 25000))
abline(v = mean(no_def), col = "green", lwd = 2)
text(mean(no_def), -400 , round(mean(no_def), 4))


hist(is_def, col = 'red', breaks = 100, xlab = 'Bankroto tikimybÄ—', ylab = 'DaÅ¾nis', main = 'Bankrutavusios Ä¯monÄ—s', xlim = c(0, 1), ylim = c(0, 200))
abline(v = mean(is_def), col = "green", lwd = 2)
text(mean(is_def), -5 , round(mean(is_def), 4))


############################ lift

plotLift(myResultsXGB$score, myResultsXGB$target, cumulative = TRUE, n.buckets = 100)
TopDecileLift(myResultsXGB$score, myResultsXGB$target)

##################################
################################## best xgb

myResultsXGB <- NULL
k <- 5

for (i in 1:k) {

  trainData <- read.csv(sprintf('CVfold%d-train.csv',i))
  trainData[,colY] <- trainData[,colY]
  Xtrain <- trainData[,-which(colnames(trainData) %in% colY)]
```

```
  XtrainNorm <- trainData[,-which(colnames(trainData) %in% colY)]
  Ytrain <- trainData[,colY]

  testData <- read.csv(sprintf('CVfold%d-test.csv',i))
  testData[,colY] <- testData[,colY]
  Xtest <- testData[,-which(colnames(testData) %in% colY)]
  XtestNorm <- testData[,-which(colnames(testData) %in% colY)]
  target <- as.logical(testData[,colY])

  dtrain <- xgb.DMatrix(data = as.matrix(Xtrain), label = as.integer(as.logical(Ytrain)))
  dtest <- xgb.DMatrix(data = as.matrix(Xtest), label = as.integer(as.logical(testData[,colY])))

  bst <- xgboost(data = dtrain, max.depth = 5, eta = 0.8, ntreelimit = 100, nthread = 20, nround = 10, objective = "binary:logistic")

  model <- rep('xgb',length(target))
  pred <- predict(bst, dtest)
  typeof(pred)
  score <- pred
  myResultsXGB <- rbind(myResultsXGB,data.frame(model,score,target))

}



importance_matrix <- xgb.importance(colnames(dtrain), model = bst)

print(importance_matrix)
xgb.plot.importance(importance_matrix = importance_matrix, top_n = 15,  xlab = "Svarbumas", ylab = "Kintamasis")



(gg <- xgb.ggplot.importance(importance_matrix, measure = "Frequency", top_n = 70, rel_to_first = TRUE, xlab = "Svarbumas", ylab = "Kintamasis"))

gg + ggplot2::ylab("Frequency")

gg + theme_minimal()
```

```
#DaÅ¾nis

xgb.dump(bst, with_stats = T)
xgb.plot.tree(model = bst)


length(importance_matrix$Feature)


xgb.plot.deepness(bst)
xgb.ggplot.deepness(bst)
xgb.plot.deepness(bst, which='max.depth', pch=16, col=rgb(0,0,1,0.3), cex=2)
xgb.plot.deepness(bst, which='med.weight', pch=16, col=rgb(0,0,1,0.3), cex=2)

xgb.ggplot.deepness(model = bst, which = c("2x1", "max.depth", "med.depth", "med.weight"))


#cutpointInfo <- optimal.cutpoints(X="score", status="target",tag.healthy=FALSE,data=myResultsXGB[,-which(names(myResultsXGB)=="model")] ,methods="SpEqualSe")
#threshold <- cutpointInfo$SpEqualSe$Global$optimal.cutoff$cutoff


threshold <- 0.01354045
#threshold <- 0.09
confMat <- confusionMatrix(myResultsXGB$target, myResultsXGB$score>threshold)
print(confMat)

plot(cutpointInfo)



test_df1 <- data.frame(matrix(ncol = 5))
colnames(test_df1)[1]="threshold"
colnames(test_df1)[2]="recall" #sens
colnames(test_df1)[3]="precision"
colnames(test_df1)[4]="f1"
```

```
colnames(test_df1)[5]="specificity"

th=c(seq(0.01, 1, by=0.01))

for(i in 1:length(th)){
  thres = th[i]
  predi = ifelse (myResultsXGB$score >= thres, 1,0)
  cm<-table(myResultsXGB$target, predi)
  if (dim(cm)[1]+dim(cm)[2]==4){
    rec = cm[2,2]/(cm[2,1]+cm[2,2])
    prec = cm[2,2]/(cm[1,2]+cm[2,2])
    f1 = 2*prec*rec/(prec + rec)
    spec = cm[1,1]/(cm[1,1]+cm[1,2])
  } else {
    rec = 0
    prec = 1
    f1 = 0
    spec = 1
  }
  test_df1 = rbind(test_df1,c(thres,rec, prec, f1, spec))
}

plot(test_df1$threshold,test_df1$recall,type="l",col="red", ylab="Ä®vertis", xlab = "Slenkstis")
par(new=TRUE)
plot(test_df1$threshold, test_df1$precision,type="l",col="blue",xaxt="n",yaxt="n" ,xlab="",ylab="")
par(new=TRUE)
plot(test_df1$threshold, test_df1$specificity,type="l",col="black",xaxt="n",yaxt="n" ,xlab="",ylab="")
par(new=TRUE)
plot(test_df1$threshold,test_df1$f1,type="l",col="green", xlab ="", ylab="", yaxt="n")
axis(4)
mtext("F1",side=4,line=3)
legend("topright",col=c("red","green","blue",   "black"),lty=1,legend=c("Jautrumas","F1",   "Tikslumas",
"Ypatumas"))

predi = ifelse (myResultsXGB$score >= 0.01354045, 1,0)
cm<-table(myResultsXGB$target, predi)
cm
```

```
plotLift(myResultsXGB$score, myResultsXGB$target, cumulative = TRUE, n.buckets = 100)

TopDecileLift(myResultsXGB$score, myResultsXGB$target)

typeof(as.prediction(myResultsXGB$score))
# And then a lift chart
perf <- performance(myResultsXGB$score,"lift","rpp")
plot(perf, main="lift curve", colorize=T)
```