VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

Justinas Mockus

DUOMENŲ GAVYBOS TECHNOLOGIJŲ TAIKYMAS INTERNETINIO PIRATAVIMO DUOMENIMS KLASIFIKUOTI

APPLICATION OF DATA MINING CLASSIFICATION TECHNIQUES ON ONLINE PIRACY DATA

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 621E14004
Duomenų gavybos technologijų specializacija
Informatikos inžinerijos studijų kryptis

Vilnius, 2016

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

TVIRTINU
Katedros vedėjas
_____
(Parašas)
_____
(Vardas, pavardė)
_____
(Data)

Justinas Mockus

DUOMENŲ GAVYBOS TECHNOLOGIJŲ TAIKYMAS INTERNETINIO PIRATAVIMO DUOMENIMS KLASIFIKUOTI

APPLICATION OF DATA MINING CLASSIFICATION TECHNIQUES ON ONLINE PIRACY DATA

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 621E14004
Duomenų gavybos technologijų specializacija
Informatikos inžinerijos studijų kryptis

Vadovas	doc. dr. Jelena Mamčenko	_____	_____
	(Moksl. laipsnis/pedag. vardas, vardas, pavardė)	(Parašas)	(Data)
Konsultantas	_____	_____	_____
	(Moksl. laipsnis/pedag. vardas, vardas, pavardė)	(Parašas)	(Data)
Konsultantas	_____	_____	_____
	(Moksl. laipsnis/pedag. vardas, vardas, pavardė)	(Parašas)	(Data)
Lietuvių kalbos konsultantas	dr. Vaida Buivydienė	_____	_____
	(Moksl. laipsnis/pedag. vardas, vardas, pavardė)	(Parašas)	(Data)

Vilnius, 2016

Vilniaus Gedimino technikos universiteto
egzaminų sesijų ir baigiamųjų darbų rengimo
bei gynimo organizavimo tvarkos aprašo
2 priedas

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Justinas Mockus, 20075707
(Studento vardas ir pavardė, studento pažymėjimo Nr.)
Fundamentinių mokslų fakultetas
(Fakultetas)
Informacinės technologijos, DGTfm-14
(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO (PROJEKTO)
SĄŽININGUMO DEKLARACIJA

2016 m. gegužės 25 d.

Patvirtinu, kad mano baigiamasis darbas tema „Duomenų gavybos technologijų taikymas internetinio pirątavimo duomenims klasifikuoti" patvirtintas 2014 m. lapkričio 11 d. dekano potvarkiu Nr. 330fm, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Mano darbo vadovas doc. dr. Jelena Mamčenko.

Kitų asmenų indėlio į parengtą baigiamąjį darbą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).

(Parašas)

Justinas Mockus
(Vardas ir pavardė)

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Informacinių technologijų katedra

ISBN ISSN
Egz. sk. .....
Data .....-.....-.....

Antrosios pakopos studijų **Informacinių technologijų** programos magistro baigiamasis darbas 4

Pavadinimas	**Duomenų gavybos technologijų taikymas internetinio piratavimo duomenims klasifikuoti**
Autorius	**Justinas Mockus**
Vadovas	**doc. dr. Jelena Mamčenko**

Kalba: lietuvių

Anotacija

Baigiamajame magistro darbe nagrinėjamos internetinio piratavimo priežastys, teisiniai instrumentai ir autorių teisių gynimo būdai. Pirmoje analitinėje dalyje pateikiama įstatymų, kurie gina autorių teises Lietuvoje ir pasaulyje, apžvalga. Glaustai apžvelgiami dažniausiai pasitaikantys pažeidimų tipai ir kaip P2P tinkluose vyksta piratavimas. Antrajame analitinės dalies skyriuje nagrinėjamos duomenų gavybos technologijos. Analizuojamos MS programinės įrangos galimybės, kuri buvo pasirinkta duomenų gavybos modeliams kurti.

Praktinėje dalyje kuriamas duomenų gavybos modelis, kuris sprendžia P2P duomenų klasifikavimo uždavinį. Modelio tikslas – nufiltruoti duomenis, kurie nėra tinkami tolesniam apdorojimui. Modelis kuriamas pagal CRISP-DM duomenų gavybos metodologiją. Naudojamos duomenų ir tekstinių duomenų gavybos technologijos. Modeliui apmokyti ir testuoti naudojami iš P2P tinklų surinkti duomenys.

Darbo pabaigoje pateikiamos baigiamojo darbo išvados. Darbą sudaro 6 dalys: įvadas, autorių teisių apžvalga, duomenų gavybos technologijų analizė, duomenų gavybos modelio kūrimas, išvados ir literatūros sąrašas.

Darbo apimtis – 55 p. teksto be priedų, 15 pav., 2 lent., 33 bibliografiniai šaltiniai.

Prasminiai žodžiai: duomenų gavyba, teksto gavyba, autorių teisės, intelektinės nuosavybės piratavimas, P2P tinklai, MS SQL Server.

Vilnius Gediminas Technical University
Faculty of Fundamental Sciences
Department of Information Technologies

ISBN ISSN
Copies No. .....
Date .....-.....-.....

Master Degree Studies **Information Technologies** study programme Master Graduation Thesis

Title	**Application of Data Mining Classification Techniques on Online Piracy Data**
Author	**Justinas Mockus**
Academic supervisor	**doc. dr. Jelena Mamčenko**

Thesis language: Lithuanian

Annotation

In this master thesis are examined online piracy causes, legislative measures and ways of defending copyright. First analytical part provides a review of the laws, which are defending copyright in Lithuania and other states of the World. Also, there is a brief review of common infringement types and how piracy is working in P2P networks. Second part analyses capabilities of MS software, which was used for data mining model creation.

Project part describes how data mining model was created, which is solving a task of P2P data classification. A goal of the model is to filter out data, which is not useful for further processing. Model creation was based on CRISP-DM data mining methodology. In the project were used data and text mining technologies. For model training and testing were used data collected from P2P networks.

At the end of the paper there are conclusions. The master thesis consists of 6 parts: introduction, copyright review, analysis of data mining technologies, data, creation of data mining model, conclusions and references.

Work size: 55 p. text without appendixes, 15 fig., 2 tab., 33 references.

Keywords: data mining, text mining, copyright, intellectual property, online piracy, P2P networks, MS SQL Server.

Turinys

Iliustracijų sąrašas

Lentelių sąrašas

Santrumpos

BI (angl. *Business Intelligence*) – verslo intelektika

DG – duomenų gavyba

ETL (angl. *Extract Transform Load*) – išgavimas, transformavimas, įkėlimas

ISP (angl. *Internet Service Provider*) – interneto serviso tiekėjas

MS – *Microsoft* kompanija. Programinės įrangos kūrėja

P2P (angl. *Peer to Peer*) – tinklų architektūra, kurioje keitimasis resursais vyksta tiesiogiai tarp naudotojų

TDG – tekstinių duomenų gavyba

Įvadas

Nuolat tobulėjant kompiuterinei technikai, spartėjant interneto greičiui ir augant naudotojų skaičiui, vis aktualesnė tampa intelektinės nuosavybės saugumo problema. Įvykdoma vis daugiau autorių teisių pažeidimų internete. Intelektinį turinį kuriančios kompanijos, autoriai ir leidėjai patiria didžiulių nuostolių. Šiuolaikinės technologijos leidžia daryti muzikos, filmų, žaidimų, elektroninių knygų bei programinės įrangos kopijas ir tai palengvina kūrinių platinimą internetinėje erdvėje. Autorių teisių turėtojams tampa sunkiau apsaugoti savo kūrinius nuo tokių kopijų atsiradimo bei platinimo internetu. Siekdami mažinti piratavimo mastus internete autorių teisių turėtojai imasi įvairių priemonių, siekdami apsaugoti intelektinę nuosavybę bei apsunkindami saugomos informacijos prieigą internete. Naudotojai, kurie platina nelegaliai įgytą informaciją, pažeidžia įstatymą ar įstatymus – tampa nusikaltėliais.

Autorių teisių turėtojai yra suinteresuoti mažinti ir panaikinti nuostolius. Viena iš priemonių yra kompanijų samdymas, kurios renka duomenis apie piratavimą. Šios kompanijos tiekia duomenis verslo intelektikos tikslams pasiekti bei šalinti piratinį turinį iš interneto. Tokios priemonės gali labai sumažinti piratavimą ir suteikti žinių, kurios padėtų plėtoti rinkodarą.

Tyrimo objektas

Tyrimo objektui pasirinkta antipiratines paslaugas teikianti įmonė. Kompanija, naudodama įvairias duomenų rinkimo technologijas, surenka didžiulius duomenų kiekius iš P2P tinklų. Kiekis yra labai didelis, todėl darbuotojai yra nepajėgūs patikrinti visų surinktų duomenų – įvertinti, ar jie tinkami pateikti klientams, todėl būtinas klasifikavimo automatizavimas. Įdiegus klasifikavimo modelį, automatizuotas duomenų gavybos procesas atrinktų netinkamus įrašus, todėl darbuotojams tektų mažesnis krūvis tikrinant duomenis. Netinkamas įrašas – tai įrašas apie piratinį turinį, kuris gali būti:

1. Pornografinis turinys.
2. Turinys nėra piratinis.
3. Bylos dydis neįprastai mažas arba didelis to turinio tipui.
4. Failas yra aplikacija, nors jo pavadinimas deklaruoja, kad yra filmas.

Darbo tikslas ir uždaviniai

Šio darbo tikslas – transformuoti iš įmonės duomenų sandėlio gautus P2P tinklų duomenis į tokią formą, kad juos sugebėtų apdoroti duomenų gavybos algoritmai. Pritaikius klasifikavimo algoritmus, juos palyginti ir išrinkti tiksliausią. Panaudojus tiksliausią algoritmą,

sukurti klasifikavimo sistemą, kuri apdorotų iš P2P tinklų surenkamus duomenis ir nustatytų ar duomenys tinakmi tolesniam apdorojimui sistemoje. Siekiant realizuoti užsibrėžtą tikslą buvo suformuluoti šie uždaviniai:

1. Įsigilinti į tiriamosios veiklos aplinką apžvelgiant intelektinės nuosavybės sąvoką, jos rūšis, autorių teisių gynimo įstatymus bei P2P tinklų veikimo principus.
2. Atlikti mokslinės literatūros analizę, kurioje aprašomi duomenų gavybos algoritmai ir nustatyti, kuriuos algoritmus būtų galima taikyti duomenų iš P2P tinklų tinkamumo nustatymui, nusprendžiant ar jie tinkami tolesniam apdorojimui įmonės sistemoje.
3. Sukurti duomenų gavybos modelį, į jį įdiegti netinkamų įrašų nustatymui tinkamus algoritmus ir palyginti jų veikimo tikslumą.
4. Radus tiksliausią modelio algoritmą, sukurti jo įdiegimo planą įmonės duomenų apdorojimo sistemoje. Tam, kad būtų svarstomas algoritmo įdiegimas sistemoje, jis turėtų veikti bent 90–ies procentų tikslumu.

Tyrimo naujumas bei jo praktinė vertė

Duomenų klasifikavimo problema yra plačiai išnagrinėta, jai spręsti yra sukurta nemažai metodų, pritaikant sprendimų medžio, *Naive Bayes* ar k–artimiausio kaimyno algoritmus. Tyrimo naujumas grindžiamas tuo, kad klasifikavimo modelis yra kuriamas panaudojant naujausią programinę įrangą MS *Visual Studio 2015* ir MS *SQL Server 2016*. Praktinė vertė remiama tuo, kad modelį galima pritaikyti antipiratine veikla užsiimančioje įmonėje. Darbo pabaigoje pateikiamas patvirtinimo raštas iš skyriaus vadovo. Šis tyrimas ypač aktualus verslui, nes įdiegus sukurtą duomenų gavybos modelį labai sumažėtų žmogiškųjų išteklių poreikis P2P duomenų klasifikavimui atlikti.

1. Intelektinė nuosavybė ir piratavimas internete

Šiame skyriuje išaiškinama intelektinės nuosavybės sąvoka, jos rūšys, kaip intelektinė nuosavybė skirstoma pagal turinio tipus tarptautiniame lygmenyje. Apžvelgiami Lietuvos įstatymai, kurie nurodo autorių teisių gynimo būdus. Taip pat aprašomas internetinis piratavimas ir kokias būdais jis atliekamas. Glaustai apžvelgiama P2P tinklų istorija ir architektūra.

Šio skyriaus tikslas – padėti suvokti veiklos sritį ir susidaryti bendrą vaizdą apie internetinį piratavimą. Šiuo tikslu įgyvendinamas pirmasis žingsnis CRISP–DM duomenų gavybos metodologijoje (žr. 2.2 CRISP–DM duomenų gavybos metodologija).

1.1. Intelektinė nuosavybė

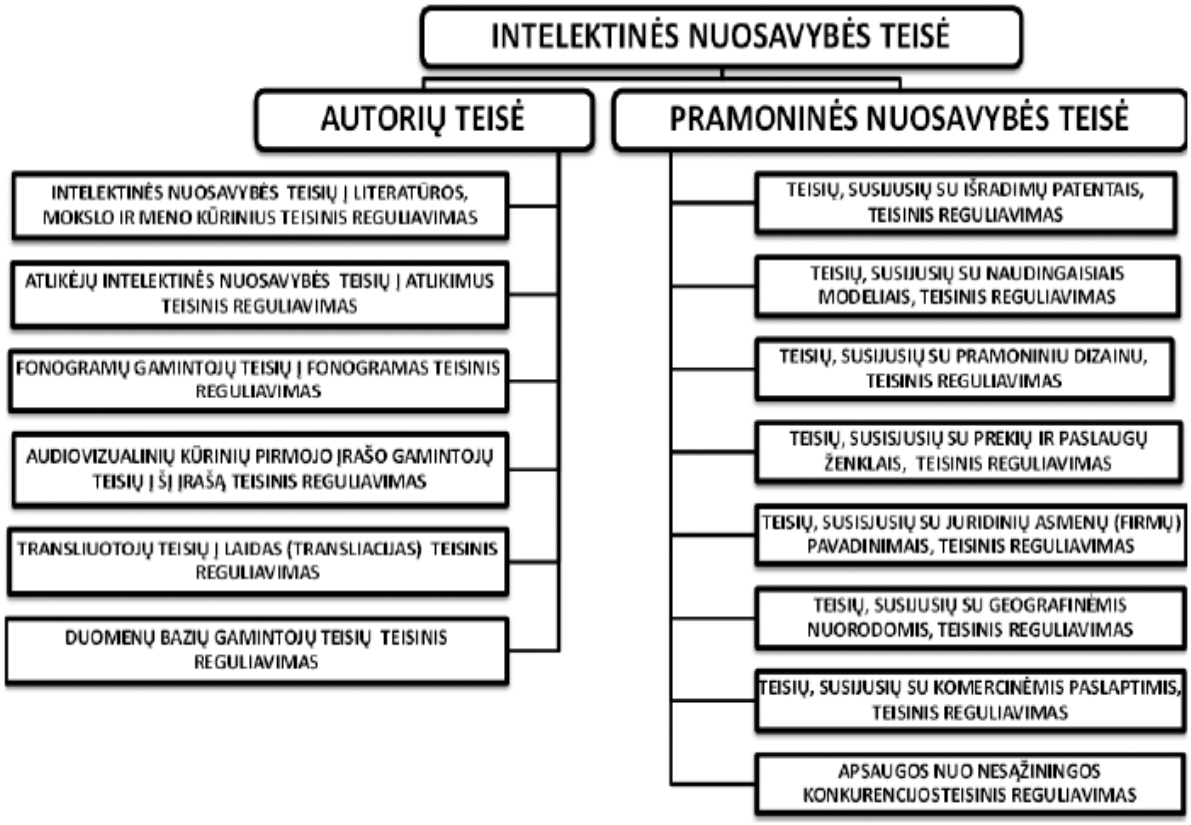
Teisiniu terminu „intelektinė nuosavybė“ (angl. *intellectual property*) išreiškiama nematerialios vertybės – intelektinės kūrybinės veiklos produkto savinimosi, nuosavybės apsaugos idėja. Šiuo terminu pabrėžiamas svarbiausias objektų, kuriems taikoma teisės apsauga, savitumas, parodoma, kuo šie objektai skiriasi nuo daiktinės nuosavybės teisės objektų. Intelektinės nuosavybės sąvoka parodoma, kad nuosavybės santykių objektais yra ne materialūs daiktai, kad nuosavybės santykiai atsiranda dėl kitokio pobūdžio – intelektinių objektų (nematerialių daiktų), kurių gamybos būdas skirtingas nuo daiktų gamybos (ar pasisavinimo iš gamtos) būdų. Šie objektai kuriami išimtinai žmogaus protine (intelekto, kūrybos) veikla ar atsiranda iš šios veiklos. Šiems objektams apibūdinti yra naudojamos nematerialios vertybės, nematerialaus turto sampratos. Nematerialumas yra išskiriamasis intelektinės nuosavybės objekto ypatumas [30].

Plačiajuosčio interneto ir P2P technologijos per pastarąjį dešimtmetį pavertė internetą pagrindine intelektinės nuosavybės teisių pažeidimų bei daugelio kitų neteisėtų veikų priemone ir terpe. Kovai su šiuo globaliu reiškiniu per pastaruosius kelis dešimtmečius priimta keletas tarptautinių ir regioninių (Europos Sąjungos) teisinių instrumentų. Tarp jų – 2001 m. Europos Tarybos Konvencija dėl elektroninių nusikaltimų, kurioje unifikuojamos materialiosios teisės normos, reglamentuojančios neteisėtų veikų internete kontrolę bei specialiai intelektinės nuosavybės teisių gynimui skirta Europos Parlamento ir Tarybos direktyva 2004/48/EB. Nepaisant šių iniciatyvų, esami teisių gynimo mechanizmai, kuriais gali naudotis intelektinės nuosavybės teisių turėtojai, dažnai laikomi nepakankamais ir neefektyviais. Empiriniu požiūriu minėti teisės aktai menkai įvertina didėjančio piratavimo skaitmeninėje erdvėje priežastis ir ypatumus, todėl modernių intelektinės nuosavybės teisių gynimo internete priemonių poreikis tampa itin aktualus [8].

1.1.1. Samprata

Materialiems daiktams apibrėžti, apibūdinti taikomi metodai ir būdai (fiziniai matavimai – kiekis, svoris etc.) yra netaikytini nematerialiems objektams apibrėžti. Nematerialaus objekto nuosavybės teisinei apsaugai svarbūs kriterijai: intelektinės kūrybinės veiklos rezultato raiška, jo išraiška materialia laikmena, kiti išskyrimui, atribojimui nuo kitų intelektinės kūrybinės rezultatų svarbūs kriterijai – originalumas, naujumas etc. Intelektinės nuosavybės objektų savitumas, jų apibrėžties ypatumai lemia, kad intelektinės nuosavybės objektais laikomi tik tie intelektinės kūrybos rezultatai, kurie nurodomi valstybėje galiojančiose bendrosiose teisės normose ir atitinka šiose normose aprašomus nematerialių vertybių identifikavimo reikalavimus, kriterijus. Perleidžiant nuosavybės teises į materialų daiktą – kūrybinės veiklos rezultato laikmeną, visos intelektinės nuosavybės teisės savaime neperleidžiamos. Intelektinės nuosavybės teisės turi būti perleidžiamos atskirai [30].

1.1.2. Rūšys



1 pav. Intelektinės nuosavybės teisės rūšys [30]

Pramonės nuosavybės samprata tarptautiniu lygiu aptarta 1883 m. Paryžiaus konvencijoje – pramonės nuosavybė apibrėžta kaip išradimų patentai, naudingieji modeliai, pramonės pavyzdžiai (arba pramoninis dizainas), prekių ženklai, juridinių asmenų (firmų) pavadinimai,

geografinės nuorodos (arba kilmės šalis), komercinės paslaptys, apsauga nuo nesąžiningos konkurencijos. Plečiantis pramoninės nuosavybės objektų katalogui šiuo metu pramonės nuosavybės objektais gali būti laikomos ir augalų, gyvūnų veislės, puslaidininkių gaminių topografijos, neskleistina komercinė informacija [30]. 1 pav. pateikiama, pagal kokius intelektinės nuosavybės teisės ginamus objektus galima išskirti intelektinės nuosavybės rūšis: autorių teisę ir pramoninės nuosavybės teisę.

1.1.3. Teisinės intelektinės nuosavybės apsaugos teritoriškumas

Intelektinės nuosavybės teisinė apsauga skiriasi nuo daiktinės nuosavybės apsaugos. Šiuos skirtumus lemia valstybių teisės sistemų ypatumai, pasireiškiantys intelektinės nuosavybės objektų, intelektinės nuosavybės teisių teisinėse identifikavimo schemose, formuojantys skirtingas įvairiose valstybėse taikomas intelektinės nuosavybės apsaugos praktikas. Tai leidžia išskirti vieną iš svarbiausių daiktinės nuosavybės ir intelektinės nuosavybės teisių teisinės apsaugos skirtumų – teritorinį intelektinės nuosavybės apsaugos pobūdį. Intelektinės nuosavybės teisių teisinės apsaugos teritoriškumas mažinamas pasirašant tarptautinius, regioninius susitarimus (dvišalius, daugiašalius) dėl intelektinės nuosavybės teisių apsaugos. Tai leidžia išskirti skirtingus (pagal galiojimo aprėptį) intelektinės nuosavybės teisinės apsaugos teisės aktus, intelektinės nuosavybės teisės šaltinius, intelektinės nuosavybės apsaugos lygius [30]:

1. Nacionalinį intelektinės nuosavybės teisinės apsaugos lygį.
2. Regioninį intelektinės nuosavybės teisinės apsaugos lygį.
3. Tarptautinį intelektinės nuosavybės teisinės apsaugos lygį.

1.1.4. Tarptautiniu lygiu pripažįstami intelektinės teisės objektai

1967 m. liepos 14 d. buvo pasirašyta konvencija dėl Pasaulinės intelektinės nuosavybės organizacijos įsteigimo (vadinamoji 1967 metų 6–oji Stokholmo konvencija; pakeista 1979 m. spalio 2 d.). Lietuva Pasaulinės intelektinės nuosavybės organizacijos (angl. *World Intellectual Property Organization, WIPO*) nare tapo 1992 m. balandžio 30 dieną. 1967 metų Stokholmo konvencija – tarptautinė daugiašalė sutartis, kurią pasirašė 184 valstybė. Intelektinė nuosavybė apima teises, susijusias su [11]:

1. Literatūros, meno ir mokslo kūriniais.
2. Atlikėjų pasirodymais, fonogramomis, radijo ir televizijos laidomis.
3. Išradimais visose žmogaus veiklos srityse.
4. Moksliniais atradimais.
5. Pramoniniu dizainu.

6. Prekių ženklais, paslaugų ženklais, komerciniais vardais ir pavadinimais.
7. Apsauga nuo nesąžiningos konkurencijos.
8. Taip pat visas kitas teises, atsirandančias vykdant intelektinę veiklą pramonės, mokslo, literatūros ar meno srityse.

Atsižvelgiant į naujausias intelektinės nuosavybės tendencijas, prie intelektinės nuosavybės teisių taip pat galima priskirti teises, susijusias su [7]:

1. Konfidencialia informacija.
2. Naudingais modeliais (mažaisiais arba inovaciniais patentais).
3. Kompiuterių programomis.
4. Duomenų bazėmis.
5. Puslaidininkių gaminių topografijomis.
6. Mikroorganizmais.
7. Techninėmis apsaugos priemonėmis.

1.2. Autorių teisių gynimo būdai Lietuvoje

Autorių teisių, gretutinių teisių ir tam tikrų teisių subjektai, gindami savo teises, išimtinių licencijų licenciatai, gindami jiems suteiktas teises, taip pat kolektyvinio teisių administravimo asociacijos, gindamos administruojamas teises, įstatymų nustatyta tvarka turi teisę kreiptis į teismą ir reikalauti [10]:

1. Pripažinti teises.
2. Įpareigoti nutraukti neteisėtus veiksmus.
3. Uždrausti atlikti veiksmus, dėl kurių gali būti realiai pažeistos teisės arba atsirasti žala.
4. Atkurti pažeistas asmenines neturtines teises (įpareigoti padaryti reikiamus taisymus, apie pažeidimą paskelbti spaudoje ar kitokiu būdu).
5. Išieškoti atlyginimą už neteisėtą naudojimąsi kūriniu, gretutinių teisių ar tam tikrų teisių objektu.
6. Atlyginti turtinę žalą, įskaitant negautas pajamas ir kitas turėtas išlaidas, o šio Įstatymo 84 straipsnyje numatytais atvejais – ir neturtinę žalą.
7. Sumokėti kompensaciją.
8. Taikyti kitus šio ir kitų įstatymų nustatytus teisių gynimo būdus.

Kai asmens, kuriam priimamas įpareigojimas nutraukti neteisėtus veiksmus ar taikomos 82 straipsnyje nurodytos atkuriamosios priemonės, veiksmuose dėl šio Įstatymo saugomų teisių pažeidimo nėra kaltės, teismas šio asmens prašymu gali įpareigoti jį sumokėti nukentėjusiai šaliai

piniginę kompensaciją, jeigu taikant šio straipsnio 1 dalyje nurodytus gynimo būdus atsirastų neproporcingai didelė žala tam asmeniui ir jeigu piniginė kompensacija nukentėjusiai šaliai yra priimtina ir pakankama. Laikoma, kad piniginę kompensaciją sumokėjęs asmuo įgyja teisę naudoti kūrinį, gretutinių teisių ar tam tikrų teisių objektą [10].

1.3. Intelektinės nuosavybės piratavimas

Pagal Kembridžo universiteto žodyną, internetinis piratavimas – tai praktika, kai pasinaudojant internetu sukuriama nelegali programinės įrangos kopija ir ji dalinama kitiems žmonėms [3].

Intelektinės nuosavybės pažeidimu laikomas intelektinės nuosavybės produktų atgaminimas, platinimas, naudojimas, laikymas ir gabenimas be teisių turėtojo sutikimo ar sutarties su teisių turėtoju (ar jo atstovu). Pažeidimu laikytinas ir bet koks intelektinės nuosavybės įstatymų pažeidimas, įskaitant neturtinių teisių pažeidimus. Pažeidimai gali būti padaromi komerciniais tikslais, taip pat ne komerciniais tikslais. Komerciniais tikslais laikomi tie atvejai – kai siekiama betarpiškos pasipelnymo, taip pat tie atvejai, kai intelektinės nuosavybės teisių objektas naudojamas kitai veiklai, susijusiai su komercija [7].

1.3.1. Intelektinės nuosavybės teisių pažeidimai

Intelektinės nuosavybės pažeidimai elektroninėje erdvėje yra specifiniai intelektinės nuosavybės teisių pažeidimai, tarp jų:

1. Tradiciniai intelektinės nuosavybės pažeidimai, kurių padarymo terpė ar įrankis yra elektroninė erdvė arba elektroninės erdvės technologijos.
2. Specifiniai intelektinės nuosavybės pažeidimai padaromi tik elektroninėje erdvėje:
    - Nuorodų į neteisėtas intelektinės nuosavybės kopijas talpinimas ir platinimas.
    - Intelektinės nuosavybės neteisėtų kopijų talpinimas ir platinimas kompiuterių tinklais.
    - Techninių apsaugos priemonių ir teisių valdymo informacijos pažeidimai.
    - Įrankių intelektinės nuosavybės pažeidimams elektroninėje erdvėje platinimas.

Intelektinės nuosavybės pažeidimais gali būti laikomi bet kokie intelektinės nuosavybės teisių pažeidimai, kadangi dauguma jų gali būti padaromi tiek elektroninėje erdvėje, tiek ir tradiciniais būdais. Pažeidimu laikytinas bet koks intelektinės nuosavybės įstatymų pažeidimas ar intelektinės nuosavybės licencinės sutarties pažeidimas, tarp jų intelektinės nuosavybės teisių objekto naudojimas be teisių turėtojo sutikimo (sutarties) [7].

Intelektinės nuosavybės pažeidimai įtraukti į daugelio išsivysčiusių valstybių baudžiamuosius ir administracinius kodeksus, tame tarpe Europos Sąjungos bei Jungtinių tautų organizacijos rekomendacijas dėl neteisėtų ir nusikalstamų veikų kvalifikavimo. Intelektinės nuosavybės pažeidimai visose valstybėse, tarp jų ir Lietuvoje, taip pat laikomi civilinių teisių pažeidimu. Lietuvos Respublika yra pasirašiusi ir ratifikavusi pagrindines tarptautines konvencijas, reglamentuojančias intelektinės nuosavybės teisių pažeidimus, bei ES direktyvas intelektinės nuosavybės klausimais [7].

1.3.2. Programinės įrangos piratavimo rūšys

Skiriamos 5 pagrindinės nelegalios programinės įrangos naudojimo rūšys [2]:

1. Galutinio naudotojo piratavimas. Tai nutinka, kai bendrovės darbuotojas atgamina kompiuterių programų kopijas be leidimo.
2. Klientas–Serveris: per didelis naudotojų skaičius. Šis piratavimo būdas atsiranda tada, kai per daug darbuotojų tinkle naudoja centrinę kompiuterių programos kopiją vienu metu. Jei turite vietinį tinklą ir įdiegiate programas serveryje kelių žmonių naudojimui, turite būti tikri, jog pagal licenciją galite tai daryti.
3. Piratavimas internete. Tai atsitinka, kai programinė įranga atsisiunčiama internetu. Tos pačios pirkimo taisyklės turi būti taikomos tiek perkant programinę įrangą internetu, tiek ir įprastais būdais.
4. Įrašymas į kietąjį diską. Tai atsitinka, kai naujų kompiuterių pardavėjai atgamina nelicencijuotas kompiuterių programų kopijas į kietąjį diską, kad parduodami kompiuteriai būtų patrauklesni.
5. Programinės įrangos klastojimas. Šis piratavimo tipas – tai neteisėtas autorių teisės saugomų kūrinių kopijavimas ir pardavimas siekiant tiesiogiai imituoti originalų kūrinį. Programinės įrangos pakuotės atveju, paprastai kompaktinių diskų (CD) ar diskelių kopijos, bei pati pakuotė, naudojimo vadovai, licencinės sutartys, etiketės, registracijos kortelės ir saugumo elementai yra suklastoti.

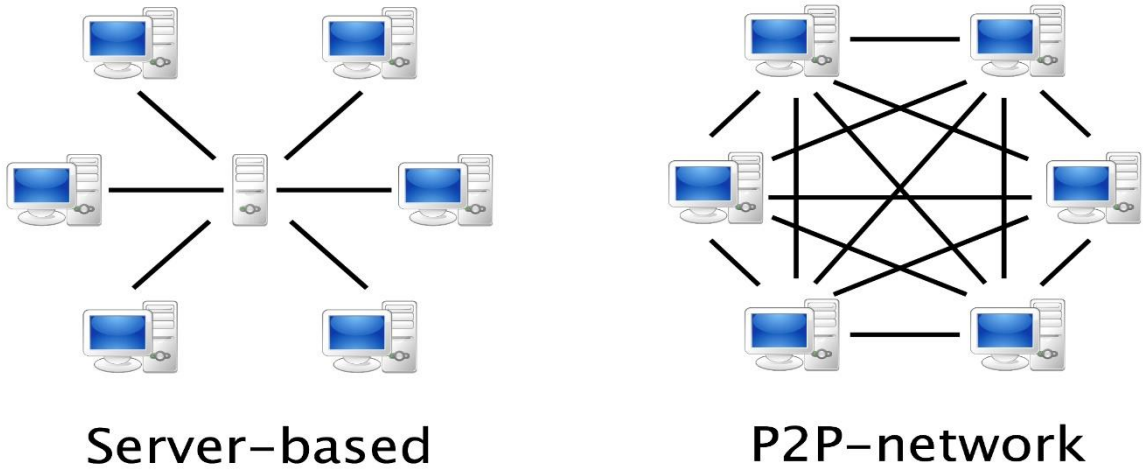
Šiame poskyryje įvardijami su programine įranga susiję piratavimo būdai, tačiau 3–ajame ir 5–ajame punkte apibrėžti piratavimo veiksmai yra taikomi ir visiems kitiems turinio tipams, t.y., muzikai, filmams, elektroninėms knygoms etc.

1.4. P2P tinklai

1.4.1. Samprata

P2P tinklai yra decentralizuoti ir išskirstyti kompiuterių tinklai, kuriuose individualūs tinklo mazgai (angl. peers) veikia ir kaip serveriai ir kaip jų naudotojai. Tinklams priklausantys asmeniniai kompiuteriai dalinasi turimais resursais (procesoriumi, atmintimi) ir šie resursai yra pasiekiami tiesiogiai, be tarpinių serverių. Priešingai nei centralizuotose sistemose, kuriose už tinklo veiklą yra atsakingi pagrindiniai serveriai. Tipinis centralizuotos sistemos pavyzdys – FTP servisas, kuriame klientai ir serveriai yra atskirti [33].

2 pav. pateikiami centralizuoto ir P2P tinklo schemų pavyzdžiai.



2 pav. Centralizuotas ir P2P tinklas [32]

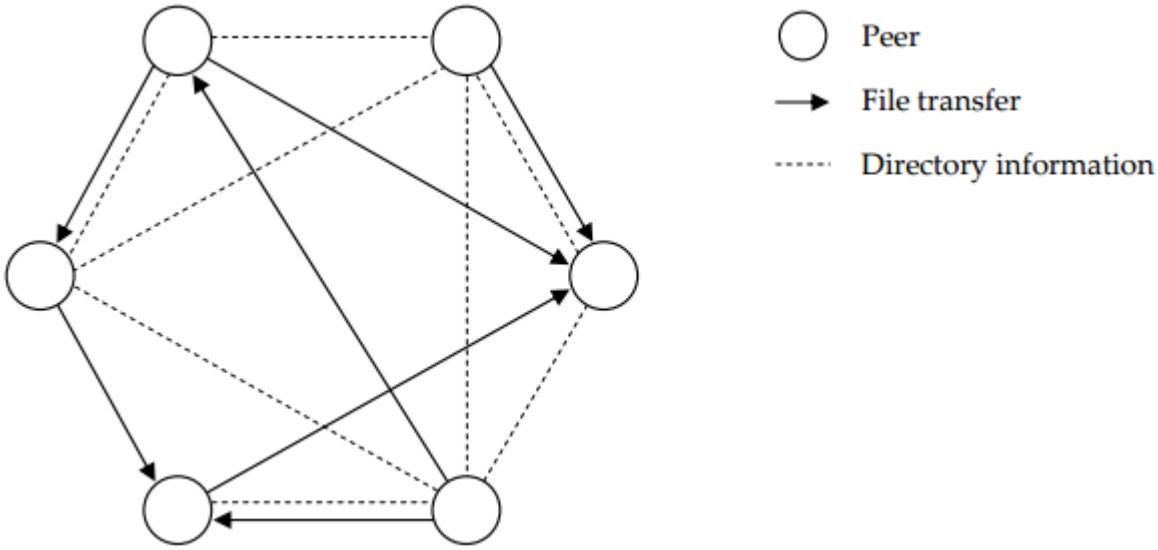
P2P tinklai pagal architektūrą skirstomi į grynuosius ir hibridinius. Grynasis P2P tinklas – pirmiausia atitinka apibūdinimą, kuris pateiktas anksčiau, taip pat – jeigu bet kuris kompiuteris gali būti atjungtas nuo tinklo be jokios žalos tinklo teikiamoms paslaugoms. T.y. grynasis P2P tinklas neturi centrinio, kitus tinklo kompiuterius aptarnaujančio serverio. Hibridiniais P2P tinklais laikomi tokie tinklai, kuriems palaikyti (teikti tam tikrą dalį tinklo siūlomų paslaugų) reikalingas centrinis serveris. Pastarosios P2P sistemos dar vadinamos centralizuotomis [31].

1.4.2. P2P tinklų kartos

P2P technologijomis paremtas bylų dalijimosi sistemas galima skirstyti į tris kartas. Pirmajai kartai priskiriama programa *Napster*. Programos naudotojams nereikėdavo žinoti vienas kito, jiems užtekdavo prisijungti prie serverio ir susirasti pageidaujamą turinį bendrame archyve, kuris buvo sudaromas pagal kitų tuo metu prisijungusių naudotojų bendrinamas direktorijas.

Naudotojui prisijungus programa gaudavo naudotojo bendrinamų bylų sąrašą ir nusiųsdavo jį į serverį ir šios bylos tapdavo matomos ir kitiems naudotojams. Norint gauti bylą programa susijungdavo su nariu turinčių ieškomą bylą ir pradėdavo siuntimą. *Napster* serverio infrastruktūra palyginus su įprastinėmis failų saugyklomis buvo nedidelė ir nesudėtinga, nes bylos buvo saugomos naudotojų kompiuteriuose, o ne serveryje. *Napster* buvo ypatingai populiari tarp muzikos mėgėjų. Šio tinklo veikla buvo sustabdyta po poros metų nuo veiklos pradžios [31].

Antros kartos P2P sistemoms priskiriamos gana įvairios sistemos, iš esmės jau decentralizuoto veikimo modelio. Šios kartos P2P sistemos jau ganėtinai skyrėsi nuo pirmojo centralizuoto *Napster* modelio. Iš antros kartos P2P sistemų paminėtinos tokios kaip *Gnutella,* kuri veikė visiškai decentralizuotai, tiesiogiai jungdama naudotojų kompiuterius, taip išvengiant centrinio



3 pav. Gnutella – FastTrack architektūra [9]

serverio dalyvavimo. Taip pat paminėtinos sistemos, veikiančios vadinamojo *FastTrack* protokolo pagrindu: tokios kaip *KaZaA, Grokster, Morpheus*. Tokios sistemos naudojo dviejų pakopų sistemą, susidedančią iš paprastų prie tinklo prijungtų kompiuterių (angl. *nodes*) ir specialių aukštesnės pakopos kompiuterių (angl. *super nodes*), įgyvendinusių *Napster* centrinio aptarnaujančio serverio failų indeksavimo funkcijas: tokioje sistemoje paprastas naudotojo kompiuteris prisijungia prie aukštesniojo kompiuterio, pateikdamas jam užklausą apie dominantį failą, ir aukštesnysis kompiuteris atsiunčia paprastam kompiuteriui jo užklausą atitinkančių failų sąrašą. Tuomet naudotojas jau gali tiesiogiai kurti ryšį su pasirinktu kito naudotojo kompiuteriu, iš kurio bus parsisiunčiamas ieškomas failas [31].

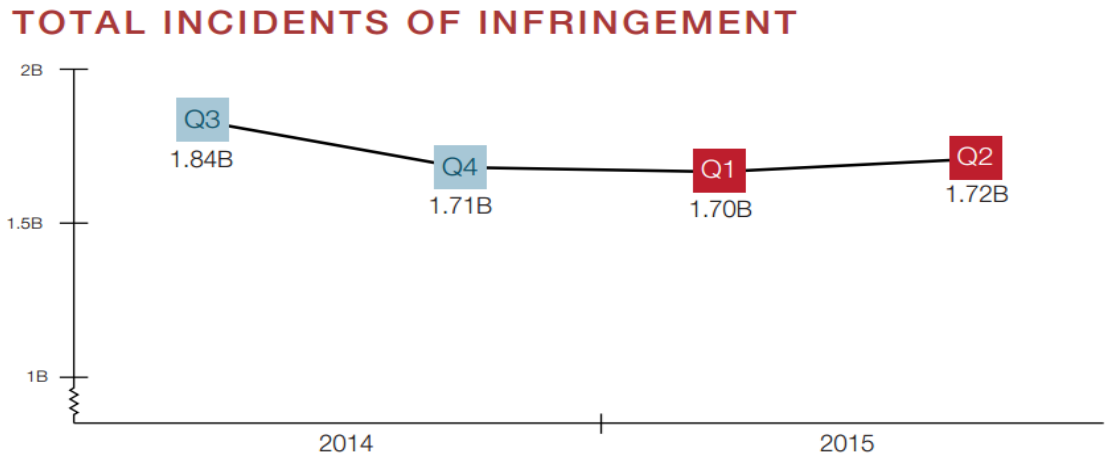
Trečiajai kartai priskiriamos sistemos veikiančios *BitTorrent* protokolo pagrindu. Taip kaip ir *Gnutella* ar *FastTrack, BitTorrent* vykdo P2P bylų perdavimą be centrinio serverio poreikio, tačiau skiriasi nuo pastarųjų dviem esminiais bruožais. Pirmiausia, *BitTorrent* padalina siuntimo procesą į didelį kiekį P2P jungčių per kurias yra parsiunčiami maži bylos fragmentai iš šaltinių, kurie

turi bylą. Parsiųstus fragmentus sujungia programinė įranga, kol galiausiai parsiunčiama visa rinkmena. Daugelio šaltinių naudojimas labai padidina didelių bylų siuntimo patikimumą. Pvz., atsijungus vienam šaltiniui siuntimo procesas tęsiamas, nes automatiškai panaudojami kiti laisvi šaltiniai. Antra, *BitTorrent* protokolas reikalauja kiekvieno naudotojo, kuris siunčiasi bylą, taip pat būti ir bylos dalinimosi šaltiniu nuo pat siuntimosi pradžios. Tokiu būdu užtikrinama, kad turinio pasiūla tinkle tenkintų paklausą. Pvz., kokia nors rinkmena tampa populiari ir didelis kiekis naudotojų nusprendžia ją parsisiųsti. Pradėję siųsti ir gavę dalį fragmentų, naudotojai iškarto tampa ir bylos šaltiniais, taip patenkindami ir padidėjusią paklausą [9].

Paieškos mechanizmas šiose naujausios kartos P2P sistemose taip pat yra kitoks. Vietoj to, kad įdiegus savo kompiuteryje tokią programą ir prisijungus prie interneto būtų tiesiogiai ieškoma norimo failo kitų naudotojų kompiuterių kietuosiuose diskuose, *BitTorrent* naudotojas pirmiausia turi ieškoti tinklalapio, kuris talpina taip vadinamą torrento failą (angl. *torrent*) failą, susietą su failu ar failais, kuriuos naudotojas nori gauti. Torrento failas talpina informaciją apie naudotojo kompiuterio, kuris skleidžia tikrąjį failų rinkinį, adresą, ir taip pat – adresą serverio, vadinamo trakerio (angl. *tracker*), kuris koordinuoja mainus konkretaus failo dalimis tarp naudotojas. Taigi torrento byla ir speciali pritaikyta kompiuterinė programa leidžia naudotojui pradėti siuntimosi procesą. Tokios hibridinės P2P sistemos pagrindu veikia ir žinomiausias Lietuvoje duomenų apsikeitimo tinklalapis Linkomanija (www.linkomanija.net) [31].

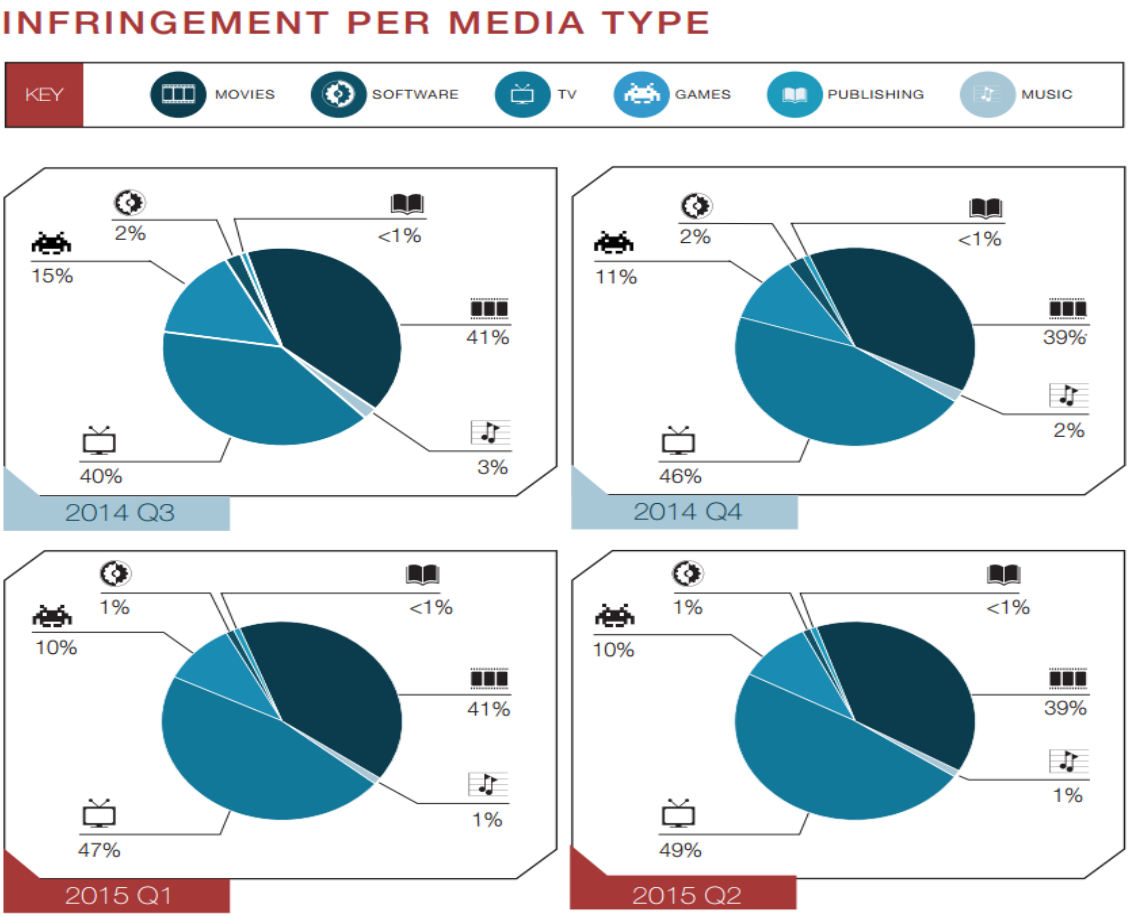
1.4.3. P2P tinklų piratavimo statistika

Dėl globalaus piratavimo ir padirbinėjimo įmonės kasmet patiria daugiau nei 350 milijardų dolerių nuostolių. 4 pav. matome stabilų piratavimo lygį P2P tinkluose nuo 2014 metų trečio ketvirčio iki 2015 metų 2 ketvirčio. Grafike pateikiama unikalių pažeidimų kiekių suma kiekvienam ketvirčiui iš visų turinio tipų [13].



4 pav. 2014 - 2015 metų piratavimo lygio grafikas [13]

5 pav. pateikiami grafikai pagal turinio tipus – kino filmus, programinę įrangą, televizijos laidas ir serialus, kompiuterinius žaidimus, elektronines knygas ir muziką. Didžiausia dalis pažeidimų yra įvykdoma siunčiantis filmus ir televizijos laidas. Grafikuose matomas didėjantis pirataujamų TV laidų parsisiuntimų procentas. Viena iš labiausiai įtakojančių priežasčių – P2P svetainių patobulinimas, leidžiant vaizdinį turinį žiūrėti iš karto, be reikalavimo pirmiausia parsisiųsti turinį kompiuterį. Kita priežastis – nuolatinis interneto greičio didėjimas, nes lėtas internetas nėra pakankamas aukštos kokybės vaizdinės medžiagos peržiūrai [13].



5 pav. Pažeidimų santykis pagal turinio tipus [13]

2. Duomenų gavybos metodai

Įmonė, kuriai kuriamas duomenų gavybos modelis, naudojasi MS programine įranga, todėl šiame darbe nėra konkurentų siūlomos programinės įrangos apžvalgos ir palyginimo. Naujos programinės įrangos pasirinkimas pareikalautų daug išlaidų – tektų pirkti programinę įrangą, skirti laiką darbuotojų apmokymams. MS *SQL Server Data Tools* turi savo duomenų gavybos programinę įrangą bei algoritmus.

Tiriamojo duomenų rinkinio atributuose yra tiek sveikuosius skaičius turinčių atributų, tiek ir saugančių nestruktūrizuotą informaciją. Todėl būtina apžvelgti literatūrą, kurioje DG taikoma įvairaus tipo informacijai.

2.1. Duomenų gavyba

Efektyvus informacijos, slypinčios duomenyse, atskleidimas ir panaudojimas yra svarbiausias konkurencingumo didinimo veiksnys šiuolaikinėje dinamiškoje tyrimų ir verslo aplinkoje. Duomenų gavyba yra šiuolaikinė informacijos analizės sritis, atsiradusi duomenų bazių technologijų, dirbtinio intelekto ir statistinės duomenų analizės sankirtoje. DG yra labai plati sritis, apimanti daug metodų, algoritmų bei taikomųjų programinių sistemų. Įprasti duomenų analizės metodai padeda atskleisti tiriamų kintamųjų priklausomumą, o DG unikali tuo, kad analizės rezultatas yra naujų priklausomybių, apie kurių egzistavimą buvo, ar net nebuvo įtariama, radimas. Šiuolaikinės DG technologijos pagrindas yra šablonų, atvaizduojančių daugiabriaunius duomenų tarpusavio santykius, koncepcija. Šie šablonai atskleidžia vidinę duomenų struktūrą bei dėsningumus, būdingus duomenų poaibiams, kurie išreiškiami naudotojui suprantamu pavidalu. Svarbi DG savybė yra ieškomų šablonų netrivialumas. Tai reiškia, kad atskleisti šablonai turi atvaizduoti neakivaizdžius, netikėtus duomenų reguliarumus, kurie sudaro taip vadinamas paslėptas žinias. Nemažai žmonių duomenų gavybą labiau laiko metodologinių principų rinkiniu, arba netgi verslo filosofijos ar matematikos sritimi, negu praktinių sprendimų rinkiniu, skirtų verslo problemoms [28].

2.1.1. Duomenų gavybos uždaviniai ir pritaikymai

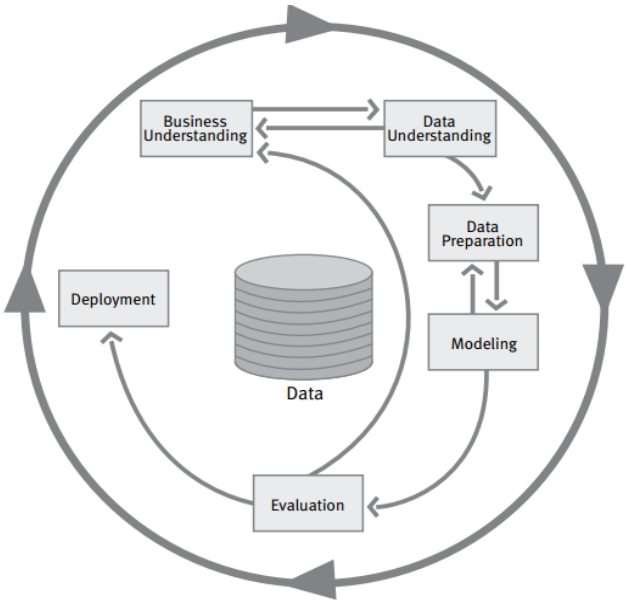
Šiuolaikinė duomenų analizė dažniausiai sprendžia šiuos uždavinius [14]:

1. Spėjimai – pardavimų, orų, serverių gedimų numatymas.
2. Rizika ir tikimybė – tinkamiausių klientų atrinkimas naujiems pasiūlymams, diagnozių tikimybių nustatymas.
3. Rekomendacijos – prekių, kurios dažnai perkamos kartu, sąrašų kūrimas. Pasiūlymų generavimas.

4. Sekų ieškojimas – pirkėjų krepšelio tyrimas, įvykių sekų nuspėjimas.
5. Grupavimas – klientų ar įvykių grupavimas pagal susijusius atributus.

2.2. CRISP–DM duomenų gavybos metodologija

CRISP–DM (angl. *Cross–Industry Standard Process for Data Mining*) – standartinis duomenų gavybos proceso modelis, kuris buvo sudarytas 1996 metais. Šio modelio sudarytojai, duomenų gavybos pradininkai – *Daimler Chrysler, SSPS, Teradata*. Pagrindinis šių organizacijų tikslas integruoti duomenų gavybą į verslo aplinką, plėtojant duomenų gavybos principus. Modelis buvo kuriamas remiantis ne tiek teorija, kiek praktika, todėl jis yra toks paplitęs ir dažnai naudojamas [4]. Paplitimą patvirtina, o taip pat ir įtakoja metodologijos pasirinkimą ir pritaikymą šiame darbe, internetinio puslapio *KDnuggets* 2014 metų pabaigoje atlikta apklausa [6].



6 pav. CRISP-DM duomenų gavybos procesų metodologija [4]

Duomenų gavybos ciklą galime suskirstyti į 6 pagrindinius žingsnius:

1. Veiklos suvokimas
2. Duomenų surinkimas ir suvokimas.
3. Duomenų transformacijos.
4. Duomenų gavybos modelių pritaikymas.
5. Rezultatų analizavimas ir vertinimas.
6. Sukurtų modelio ar modelių įdiegimas.

6 pav. paveiksle matome, kad procesas yra ciklišKas, tai reiškia, kad duomenų gavyba yra dinamiškas ir grįstas iteracijomis procesas. Pavyzdžiui, duomenų rinkiniui pritaikius modelius gali paaiškėti, kad rezultatai yra nepakankami žinių išgavimui, todėl gali tekti perkonfigūruoti modelį ar papildyti duomenų rinkinį naujais atributais [4].

2.2.1. Veiklos (verslo) suvokimas

Pirmasis žingsnis pagal CRISP–DM modelį – įsigilinti ir suvokti turimą verslo informaciją. Tada iškelti tikslius klausimus ir tikslus. Pradinėje fazėje akcentuojamas duomenų analizės projekto tikslų supratimas ir reikalavimų iš dalykinės srities perspektyvos, taip pat šios suformuluotos problemos vertimas į duomenų tyrybos problemos apibrėžimą. Šioje fazėje nustatomas preliminarus planas, kaip bus siekiama tikslo [27].

2.2.2. Duomenų surinkimas ir suvokimas

Šioje fazėje pirmiausia surenkami projektui reikalingi duomenys, tada atliekami šie veiksmai [4]:

1. Atliekama duomenų rinkinio analizė, kuriama ataskaita apie atributus ir juose saugomus duomenis.
2. Tikrinama duomenų kokybė, jei randama trūkumų – teikiami pasiūlymai kaip juos šalinti.
3. Atlikus pirminę analizę apibendrinami pirmieji pastebėjimai, formuluojama prielaida, koks duomenų rinkinys gali būti tinkamas duomenų gavybai.

2.2.3. Duomenų transformacija ir valymas

Duomenų parengimo fazė apima visas veiklas, reikalingas galutiniam duomenų rinkiniui parengti. Šios fazės veiksmai priklauso nuo turimų pradinių neapdorotų duomenų. Tipiniai duomenų parengimo uždaviniai yra lentelės, įrašo ir atributo projekcijų parinkimas, atributų transformacija, normalizavimas, kategorizavimas, triukšmų šalinimas, diskretizavimas [27].

2.2.4. Modelio taikymas

Šioje fazėje parenkami tinkami modeliavimo metodai, algoritmai arba jų deriniai. Toliau parenkamos optimalios algoritmų parametrų reikšmės. Dažniausiai galimi keli modeliavimo metodai tam pačiam uždaviniui išspręsti. Šis etapas vykdomas iteraciniu būdu, kol pasiekiamas užsibrėžtas modelio kokybės kriterijus [27]. Taip pat šiame žingsnyje tiriamasis duomenų rinkinys padalinamas į mokymo ir testavimo rinkinius. Modelis treniruojamas naudojantis mokymo duomenimis ir testuojamas naudojantis testavimo rinkinį [4].

2.2.5. Modelio įvertinimas

Šioje fazėje jau suformuotas aukštos kokybės modelis (arba keli modeliai). Prieš galutinai įdiegiant modelį labai svarbu nuodugniai jį įvertinti, peržiūrėti modelio konstravimo žingsnius, įsitikinti, kad veiklos tikslai yra pasiekti tinkamai. Modelio kokybei įvertinti naudojamos DG ir statistikoje populiarios metrikos: bendras tikslumas, jautrumas, specifiškumas, ROC kreivė, teigiamas ir neigiamas tikimybinis santykis, teigiama ir neigiama prognostinė vertė. Galutinis šios fazės rezultatas – sprendimas, ar duomenų tyrybos rezultatai gali būti naudojami [27].

2.2.6. Modelio įdiegimas

Modelio sukūrimas nėra duomenų tyrybos projekto pabaiga. Netgi tuo atveju, jei duomenų tyrybos projekto tikslas buvo sužinoti daugiau apie turimus duomenis, gautos žinios turi būti organizuotos ir pristatytos galutiniam naudotojui suprantama forma. Priklausomai nuo reikalavimų, paprasčiausiu atveju diegimo fazę gali sudaryti ataskaitos parengimas arba pasikartojančio duomenų tyrybos proceso įdiegimas. Labai svarbu, kad galutinis naudotojas iš anksto suprastų, kokie veiksmai turės būti atlikti siekiant gauti praktinę naudą iš sukurto DG modelio [27].

2.3. Tekstinių duomenų gavyba

Tekstinių duomenų gavybos tikslas yra nestruktūrizuotos (tekstinės) informacijos apdorojimas, reikšmingų indeksų ištraukimas iš teksto, ir išgautų duomenų paruošimas duomenų gavybos algoritmams. TDG algoritmai gali analizuoti žodžius, žodžių rinkinius naudojamus dokumentuose, taip pat gali analizuoti dokumentus ir rasti panašumus tarp jų arba kaip jie yra susiję su kitais duomenų gavybos modelio kintamaisiais [29].

Tekstinių duomenų gavyboje, analogiškai kaip ir duomenų gavyboje, siekiama išgauti naudingą informaciją identifikuojant ir tiriant įdomius šablonus ar pasikartojimus. TDG atveju duomenų šaltiniai yra dokumentų rinkiniai, įdomūs šablonai randami ne normalizuotose duomenų struktūrose, o nestruktūrizuotuose tekstiniuose duomenyse. Architektūriškai abi sistemos yra ganėtinai panašios – abiem yra bendros duomenų apdorojimo rutinos, šablonų atpažinimo algoritmai ir vizualizavimo sluoksnio elementai [5].

2.3.1. TDG problemos

Kuriant TGD sistemą susiduriama su šiomis problemomis [29]:

1. Didelis kiekis mažų dokumentų ir mažas kiekis didelių dokumentų. Susiduriama, kai tarkime, turime tik kelias didelės apimties knygas. Tokiu atveju statistinė analizė greičiausiai bus

netiksli, nes atvejų skaičius (dokumentų) yra mažas, nors kintamųjų (ištrauktų žodžių) skaičius yra didžiulis.

2. Ženklų, skaičių, trumpų žodžių šalinimas. Analizuojant dokumentus būtina atlikti tekstų valymą: išmesti skaičius, skyrybos ir kitus ženklus, ženklų sekas, trumpus ar retai pasikartojančius žodžius.

3. Nereikalingų žodžių sąrašo sudarymas. Specifinių žodžių sąrašo sudarymas gali būti naudingas, kai tekste ieškoma atitinkamų žodžių, kurie gali padėti, pvz., dokumentų klasifikavimui. „STOP“ žodžių sąrašą dažniausiai sudaro asmenvardžiai, jungtukai ir kiti labai dažnai pasitaikantys nereikšmingi žodžiai.

4. Sinonimai ir frazės. Sinonimai ir frazės, kurie turi tą pačią reikšmę, gali būti apjungti, taip sumažinant kintamųjų skaičių.

5. Žodžių šaknų ištraukimas. Atlikus žodžių šaknų ištraukimą, taip pat apjungiami panašias reikšmes turintys žodžiai, taip sumažinant kintamųjų skaičių. Pavyzdžiui, atlikus šaknies ištraukimą, žodžiai „keliautojas“ ir „keliaudavo“ bus atpažinti kaip tas pats žodis.

6. Skirtingų kalbų palaikymas. Šaknys, sinonimai, raidės naudojamos žodžiuose yra labai priklausomi nuo kalbos. Todėl kuriant sistemą verta įtraukti skirtingų kalbų palaikymą.

2.4. Tekstinių duomenų transformacijos su MS *SQL Server Data Tools*

Šiame skyriuje nagrinėjamos dvi duomenų srauto transformacijos, kurios palengvina teksto gavybos užduotį: *Term Extraction* ir *Term Lookup*.

MS duomenų gavybos algoritmai palaiko tekstinių duomenų tipą, bet jie yra nepakankamai atlikti prasmingą teksto analizę. Algoritmų perspektyvoje, tekstinių duomenų tipą turintys stulpeliai yra traktuojami kaip diskretieji *LONG* duomenų tipo stulpeliai, kurių duomenys saugomi kaip įvairių būsenų rinkiniai. Norint atlikti teksto gavybą su *SQL Server Data Tools* pirmiausia reikia tekstinius atributus suformuoti taip, kad juos galėtų apdoroti duomenų gavybos algoritmai. Kiekvienas tekstas traktuojamas kaip žodžių ar frazių rinkinys, duomenų gavyba atliekama skaičiuojant tam tikrų žodžių ar frazių dažnumą. Kai kiekvienas dokumentas yra suskaidomas į raktinių žodžių rinkinius, galima pritaikyti šių modelių duomenų gavybos algoritmus [12]:

1. Klasifikavimo modeliai, kurie kaip įvesties duomenis naudoja raktinius žodžius ir frazes ir pagal juos nustato dokumento klasę.

2. Klasterizavimo modeliai, kurie grupuoja panašius dokumentus pagal pasikartojančius terminus.

3. Asociacijų modeliai, kurie ieško panašumų tarp raktinių žodžių ir frazių.

Teksto gavyba su MS *SQL Server Data Tools* dažniausiai susideda iš mažiausiai trijų žingsnių [12]:

1. Žodyno sudarymo iš raktinių žodžių ir frazių iš analizuojamo duomenų rinkinio. Užduotis dažniausiai atliekama panaudojant *Term Extraction* transformaciją.
2. Remiantis žodynu, išgaunami reikšmingi žodžiai ir frazės kiekvienam dokumentui. Užduočiai spręsti taikoma *Term Lookup* transformacija.
3. Duomenų gavybos modeliai apmokomi panaudojus transformuotus tekstinius duomenis.

2.4.1. *Term* Extraction transformacija

Term Extraction transformacija naudojama žodyno kūrimui iš raktinių žodžių ir frazių. Dažniausiai tai yra pirmasis žingsnis tekstinių duomenų gavybos projekte. Transformacija taikoma duomenų srautui, kuris turi turėti vieną tekstinių duomenų tipo atributą. Jos tikslas yra išanalizuoti atributo duomenis ir sukurti žodyną iš gautų reikšminių žodžių. Raktinių terminų išgavimas nėra trivialus uždavinys, nes yra taikomos sudėtingos technikos, tokios kaip žodžių šaknų išrinkimas ar gramatikos nagrinėjimas [12].

Term Extraction transformacija veikia tik su anglų kalbos žodžiais ir gali ištraukti iš teksto daiktavardžius, daiktavardžių frazes (angl. noun phrases), arba ir daiktavardžius ir daiktavardžių frazes. Daiktavardžių frazės – mažiausiai du žodžiai, kurių vienas yra daiktavardis, o kitas yra daiktavardis arba būdvardis. Esant poreikiui, transformacija palaiko nereikalingų terminų išmetimą, tokie terminai turi būti saugojami atskiroje bazės lentelėje. Transformacija sugeneruoja du atributus – *Term* ir *Score*. Term atribute saugomi terminai, *Score* saugomas termino rezultatas, tai gali būti pasikartojimų skaičius analizuotuose dokumentuose arba *TFIDF* rezultatas, kuris gaunamas padalinus termino pasikartojimų skaičių iš visų dokumentų skaičiaus [25].

2.4.2. *Term* Lookup transformacija

Po žodyno sukūrimo, kiekvienas analizuojamas dokumentas turi būti transformuojamas į terminų rinkinį. Tai atliekama remiantis išgautų terminų žodynu. *Term Lookup* transformacija įvesties tekste ieško reikšminių terminų pagal žodyną, kurį sukuria *Term Extraction* transformacija. Kaip ir *Term Extraction* transformacijai, *Term Lookup* transformacijos įvesčiai yra reikalingas vienas tekstinio tipo atributas. *Term Lookup* transformacija į lentelę išveda du atributus – *Term* ir *Frequency* (šis atributas nurodo kiek kartų terminas pasikartojo analizuojamame duomenų rinkinyje) [12].

Term Lookup transformacija yra naudinga nestandartinio žodžių sąrašo kūrimui panaudojant įvesties tekstinius duomenis. Prieš paiešką transformacija išgauna žodžius ir teksto tokiu pačiu metodu kaip ir *Term Extraction* transformacija [26]:

1. Tekstas suskaidomas į sakinius.
2. Sakiniai suskaidomi į žodžius.
3. Terminai yra normalizuojami.

Term Lookup transformacija atlieka paiešką ir gražina rezultatą pagal šias taisykles [26]:

1. Jei transformacija yra sukonfigūruota skirti didžiąsias ir mažąsias raides, tapatinimai kurie nesutaps bus atmesti. Pvz., *studentas* ir *STUDENTAS* bus traktuojami kaip skirtingi žodžiai.
2. Jei nuorodų lentelėje egzistuoja daiktavardžio daugiskaitos arba daiktavardžio frazės forma – paieška sutapatins tik su daugiskaita arba daiktavardžių frazėmis. Pvz., visi žodžiai *studentai* bus skaičiuojami atskirai nuo *studentas.*
3. Jei rinkinyje egzistuoja tik vienaskaitinis daiktavardis, tiek vienaskaitinės, tiek daugiskaitinės formos bus priskaičiuojamos vienaskaitinei formai.
4. Jei įvestis yra daiktavardžio frazė – normalizacija bus taikoma tik paskutiniam žodžiui.

2.5. MS *Full–Text Search* technologija

MS *Full–Text Search* technologija skirta tekstinės informacijos paieškos palengvinimui *SQL Server* lentelėse. Šio tipo užklausos atlieka lingvistinę paiešką tekstiniuose duomenyse pasinaudodamos *full–text* indeksais ir operuodamos žodžiais ar frazėmis pagal naudojamos kalbos, pvz., anglų ar japonų, taisykles. *Full–Text* užklausa grąžina dokumentus, kuriuose yra bent vienas sutapimas su ieškomu tekstu. Prieš pradedant naudoti *Full–Text Search* technologiją, būtina sukurti *full–text* indeksus *full–text* kataloguose. Po indekso sukūrimo lentelės tekstiniams atributams, galima atlikti šias paieškas [1]:

1. Paprastų terminų – tai yra, vieno ar daugiau žodžių ar terminų.
2. Pagal žodžio pradžią – tai yra, panaudoti ne pilną žodį, o tik jo pradžią.
3. Vienaskaitos, daugiskaitos ar linksniuotų terminų.
4. Panašiai skambančių žodžių.
5. Tezauro arba sinonimų.
6. Terminų, pagal individualizuotą svorį.
7. Statistinę semantinę paiešką.
8. Panašių dokumentų.

LIKE operatorius, priešingai nei f*ull–text* paieška, veikia tik pagal simbolių šablonus. *LIKE* operatoriaus negalima taikyti suformatuotiems dvejetainiams duomenims. Tęsiant, užklausa naudojanti *LIKE* operatorių yra gerokai lėtesnė dirbant su nestruktūrizuotais duomenimis. *LIKE*

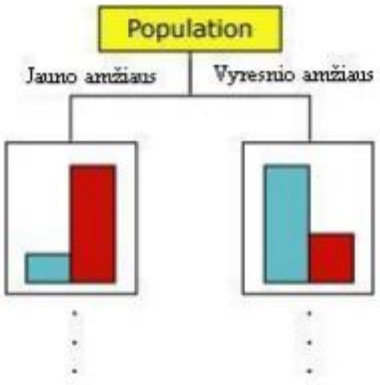
operatorius apdorodamas keletą milijonų įrašų gali užtrukti kelias minutes, o f*ull–text* paieška tai atlieka per kelias sekundes ar mažiau, priklausomai nuo to, kiek įrašų yra grąžinama [1].

2.6. MS duomenų gavybos algoritmai

SQL Server Data Tools for Visual Studio 2015 paketas turi 9 duomenų gavybos algoritmus, kurie glaustai aprašomi šio skyriaus poskyriuose. Skyriuje analizuojami integruoti algoritmai, pateikiant jų veikimo principą, taikymo galimybes bei parametrus. Taip pat MS *Visual Studio* turi galimybę naudoti ir trečiųjų šalių algoritmus, tačiau informacija apie juos šiame darbe nebus pateikiama.

2.6.1. Sprendimų medžių algoritmas

Pavadinimas pakete – *Microsoft Decision Trees Algorithm*. Tai klasifikavimo ir regresijos algoritmas, tiek diskrečių, tiek tolydžių požymių spėjimui. Diskrečių dydžių atveju, algoritmas ieško ryšio tarp keleto atributų reikšmių ar būsenų ir priklausomai nuo dažniausiai pasitaikančių ryšių, juos skirsto į dvi šakas. Tarkime, turime įvairaus amžiaus tam tikros prekės pirkėjus. Mums reikia nuspręsti, kuris pirkėjas greičiausiai pirktų šią prekę, jei 8 iš 10 jaunuolių pirko analogišką prekę ir 2 iš 10 vyresnio amžiaus žmonių pirko taip pat (*7 pav. Sprendimų medžio šakojimas*. Mėlynas stulpelis – amžius; raudonas stulpelis – nupirktos prekės). Algoritmas nustato, kad amžius yra tinkamas spėjimo požymis, prekių pirkimui. Sprendimų medyje spėjimas daromas remiantis išvadomis, apie esamų ryšių tendencijas. Giliau gali būti šakojama pagal kitus požymius [15].



7 pav. Sprendimų medžio šakojimas [15]

Siekdamas nustatyti kur sprendimų medis išsišakos, tolydžių požymių grupavimui algoritmas naudoja tiesinę regresiją. Naudojama dviejų tiesinių lygčių laužtė, kurios lūžio taškas ir yra medžio išsišakojimo vieta. Šiuo atveju kiekvienos šakos charakteringas požymis yra tiesinės regresijos lygtis. Turint sprendimų medį – apmokytą algoritmą, jau galima naudoti tokius duomenis,

pagal kurių požymius galime priimti mus dominantį sprendimą. Toliau pateikiami šio algoritmo parametrai – nurodomas pavadinimas, trumpas aprašymas bei pagal nutylėjimą nustatyta parametro reikšmė [15]:

1. *COMPLEXITY_PENALTY* – apibrėžia išsišakojimo tikimybę. Numatytoji reikšmė priklauso nuo įvesties atributų skaičiaus. 0,5 kai atributų mažiau nei 10, 0,9 kai atributų nuo 10 iki 99 ir 0,99 kai parametrų daugiau nei 100.
2. *FORCE_REGRESSOR* – galima priverstinai nurodyti, kuriuo atributu remiantis bus skaičiuojama medžio skilimo regresijų lygtys. Numatytosios reikšmės nėra.
3. *MAXIMUM_INPUT_ATTRIBUTES* – maksimalus atributų skaičius skirtas algoritmo apmokymui. Jei yra daugiau atributų, automatiškai parenkami charakteringiausi. Numatytoji reikšmė – 255.
4. *MAXIMUM_OUTPUT_ATTRIBUTES* – maksimalus atributų skaičius ryšių atvaizdavimui ir būsenų charakterizavimui. Numatytoji reikšmė – 255.
5. *SCORE_METHOD* – nurodoma, kokiu metodu apskaičiuojamas šakojimo taškas. Galimos reikšmės: 1 – šakojama pagal C. Shannono entropinį pasiskirstymą. 3 – Bayeso K2 labiausio tikėtinumo; 4 – pagal Bayeso – Dirichleto labiausio tikėtinumo pasiskirstymą. Numatytoji reikšmė – 4.
6. *SPLIT_METHOD* – nurodo kaip šakoti sprendimų medį. Galimos reikšmės – 1 – binarinis, 2 – pilnas, 3 – kombinuotas šakojimas. Numatytoji reikšmė – 3.

2.6.2. Grupavimo algoritmas

Pavadinimas pakete – *Microsoft Clustering Algorithm*. Grupavimo (angl. clustering) algoritmas duomenis skirsto į tam tikras kategorijas automatiškai pagal skirstymo požymius. Tai yra iteracinis algoritmas, kiekvieną kartą vis tikslinant objekto priklausomumą vienai ar kitai grupei, kol pasiekiamas norimas tikslumas. Algoritmas naudingas duomenų tyrimui, ieškant įvairių anomalijų, bei spėjimams apie naujų duomenų aibes atlikti. Taikant galima atrasti iš pirmo žvilgsnio nepastebimų dėsningumų. Toliau pateikiami šio algoritmo parametrai – nurodomas pavadinimas, trumpas aprašymas bei pagal nutylėjimą nustatyta parametro reikšmė [16]:

1. *CLUSTER_COUNT* – apibrėžiamas klasterių skaičius. Numatytoji reikšmė – 10.
2. *CLUSTER_SEED* – atsitiktinis skaičius, įtakojantis atributų pasiskirstymą pirmoje apmokymo iteracijoje. Numatytoji reikšmė – 0.
3. *CLUSTERING_METHOD* – vienas iš 4 metodų priskirti atributą vienam ar kitam klasteriui. Galimos reikšmės: 1 – tikėtinumo maksimizavimo metodas su pakartotina peržiūra; 2 – tikėtinumo maksimizavimo iš karto; 3 – *K–means* algoritmas, apskaičiuojant atstumus tarp klasterių

su pakartotina peržiūra atitikimui klasteriui; 4 – *K–means* algoritmas, apskaičiuojant atstumus tarp klasterių. Numatytoji reikšmė – 1.

4. *MAXIMUM_INPUT_ATTRIBUTES* – maksimalus atributų skaičius skirtas algoritmo apmokymui. Jei yra daugiau atributų, automatiškai parenkami charakteringiausi. Numatytoji reikšmė – 255.

5. *MAXIMUM_STATES* – nurodoma galimų atributo reikšmių skaičių. Numatytoji reikšmė – 100.

6. *MINIMUM_SUPPORT* – apibrėžia minimalų atvejų skaičių, kiek jų turi būti kiekviename klasteryje. Numatytoji reikšmė – 1.

7. *MODELLING_CARDINALITY* – nurodoma kiek kartų tobulinti klasterius, kol modelis ištreniruojamas. Numatytoji reikšmė – 10.

8. *SAMPLE_SIZE* – nurodoma kiek įrašų įtraukiama į modelio apmokymą. Numatytoji reikšmė lygi 50000. Gali būti automatiškai parinkta kita reikšmė, priklausomai nuo duomenų rinkinio įrašų kiekio. Numatytoji reikšmė – 50000.

9. *STOPPING_TOLERANCE* – dydis, kuris apibūdina modelio treniravimo pabaigos sąlygą. Kitaip tariant nustatomas modelio tikslumas. Numatytoji reikšmė – 10.

Pradinių duomenų aibėje identifikuojami ryšiai ir generuojamos požymių grupės pagal tuos ryšius. Tuomet skaičiuojama, kaip tiksliai sukurtosios grupės atspindi išsidėsčiusių duomenų požymius. Yra du būdai paskaičiuoti priklausomumą kuriai nors grupei. Pirmasis – priklausymo vienai ar kitai grupei tikimybės skaičiavimas (kur tikimybė didžiausia – tai grupei objektas priklausys). Kitas būdas – artimiausio kelio iki atitinkamos grupės radimas [16].

2.6.3. Naivusis Bajeso algoritmas

Pavadinimas pakete – *Microsoft Naive Bayes Algorithm*. Naivusis Bajeso (angl. Naive Bayes) algoritmas naudojamas spėjimų modeliavimui. Kadangi skaičiavimų prasme vienas paprasčiausių – taikomas preliminariam įvertinimui, o vėliau spėjimus galima tikslinti pasitelkiant kitus algoritmus. Veikimas pagrįstas tikimybių skaičiavimu, spėjant kokia būsena gali būti kiekviename požymių stulpelyje. Taikant *Naive Bayes* algoritmą traktuojama, kad visi įėjimo stulpelių atributai yra nepriklausomi. Šis algoritmas taikomas tik diskretiems arba diskretizuotiems požymiams tirti. Toliau pateikiami šio algoritmo parametrai – nurodomas pavadinimas, trumpas aprašymas bei pagal nutylėjimą nustatyta parametro reikšmė [17]:

1. *MAXIMUM_INPUT_ATTRIBUTES* – maksimalus atributų skaičius skirtas algoritmo apmokymui. Jei yra daugiau atributų, automatiškai parenkami charakteringiausi. Numatytoji reikšmė – 255.

2. *MAXIMUM_OUTPUT_ATTRIBUTES* – maksimalus atributų skaičius ryšių atvaizdavimui ir būsenų charakterizavimui. Numatytoji reikšmė – 255.

3. *MAXIMUM_STATES* – maksimalus atributų įgyjamų reikšmių (būsenų) skaičius. Numatytoji reikšmė – 100.

4. *MINIMUM_DEPENDECY_PROBABILITY* – minimali tikimybė tarpusavio priklausomumo atributų, kurie įtraukiami į modelį. Tikimybei mažėjant daugėja mažiau susijusių atributų, kurie bus įtraukti. Numatytoji reikšmė – 0,5.

2.6.4. Asociacijų algoritmas

Pavadinimas pakete – *Microsoft Association Algorithm*. Naudojamas tiek pirkinių krepšelio analizei, tiek rekomenduojant pirkėjams naują produkciją, žinant ką jie jau pirko. Algoritmo veikimas: peržiūrimi visi duomenų įrašai ir atrenkami tokie, kurie turi panašių sąsajų. Jie grupuojami į tyrėjo apibrėžtą grupių skaičių. Kiekvienai grupei generuojama charakterizuojanti taisyklė. Pagal tai galima aptikti dėsningumus naujiems atvejams – nuspėti, kokiai kategorijai jie priklausys. Toliau pateikiami šio algoritmo parametrai – nurodomas pavadinimas, trumpas aprašymas bei pagal nutylėjimą nustatyta parametro reikšmė [18]:

1. *MAXIMUM_ITEMSET_COUNT* – visų elementų aibių modelyje skaičius. Numatytoji reikšmė – 2000000.

2. MIN_SUPPORT – minimalus palaikymas bet kokiai vienai elementų aibei. Ši vertė gali skirtis nuo vertės, kuri nurodoma *MINIMUM_SUPPORT* parametre. Numatytoji reikšmė – 0,03.

3. *MAX_SUPPORT* – maksimalus palaikymas bet kokiai vienai elementų aibei. Ši vertė gali skirtis nuo vertės, kurią nurodote *MAXIMUM_SUPPORT* parametre. Numatytoji reikšmė – 1.

4. *MIN_ITEMSET_SIZE* – mažiausios elementų aibės dydis, apibūdinamas kaip elementų skaičius. 0 reikšmė rodo, kad trūkstama būsena buvo laikoma nepriklausomu elementu. Numatytoji reikšmė – 1.

5. *MAX_ITEMSET_SIZE* – rodo didžiausios elementų aibės, kuri buvo surasta, dydį. Ši vertė yra ribojama vertės, kurią jūs nustatėte *MAX_ITEMSET_SIZE* parametrui, kai kūrėte modelį. Ši vertė niekada negali viršyti tos vertės; tačiau, gali būti mažesnė. Numatytoji reikšmė – 3.

6. *MIN_PROBABILITY* – Minimali tikimybė, aptikta bet kokiai vienai elementų aibei ar taisyklei modelyje. Elementų aibei ši vertė yra visada didesnė negu vertė, kurią jūs nustatėte *MINIMUM_PROBABILITY* parametrui, kai kūrėte modelį. Numatytoji reikšmė – 0,4.

2.6.5. Sekų grupavimo algoritmas

Pavadinimas pakete – *Microsoft Sequence Clustering Algorithm*. Panašus į grupavimo algoritmą, tačiau jeigu grupavimo algoritme užteko surasti skirtingų požymių grupes, čia reikia išskirti panašių veiksmų grupes. Tai gali būti žiniatinklio puslapių navigacijos veiksmų seka, prekių krepšelio papildymo seka ir panašiai. Toliau pateikiami šio algoritmo parametrai – nurodomas pavadinimas, trumpas aprašymas bei pagal nutylėjimą nustatyta parametro reikšmė [19]:

1. *CLUSTER_COUNT* – apibrėžiamas klasterių skaičius. Numatytoji reikšmė – 10.
2. *MINIMUM_SUPPORT* – apibrėžia minimalų atvejų skaičių, kiek jų turi būti kiekviename klasteryje. Numatytoji reikšmė – 10.
3. *MAXIMUM_SEQUENCE_STATES* – nustatomas maksimalus būsenų kiekis. Numatytoji reikšmė – 64.
4. *MAXIMUM_STATES* – maksimalus atributų įgyjamų reikšmių (būsenų) skaičius. Numatytoji reikšmė – 100.

Naudojamas tikimybių maksimizavimas, identifikuoti grupėms ir joms būdingas dažniausiai pasitaikančias veiksmų sekas. Veiksmų seka suprantama kaip aibė būsenų, sujungtų perėjimais iš vienos į kitą. Taikant šį algoritmą vienu metu galima aprašyti tik vienokio tipo būsenas, tai reiškia, kad vienu metu negalima ištirti pvz. ir prekių krepšelio papildymo ir puslapių navigacijos sekų [19].

2.6.6. *Time* Series algoritmas

Pavadinimas pakete – *Microsoft Time Series Algorithm*. Tai regresinis algoritmas, skirtas prognozuoti tolydžių dydžių pokyčiams. Tai gali būti produkto pardavimo tendencijos ar kainų augimo prognozavimas. *Time Series* algoritmas pritaikęs autoregresijos sprendimų medžio principą, gali numatyti tendencijas pagal esamų duomenų reikšmes, tai yra – prognozuoti. MS *Time Series* algoritmas turi svarbią savybę: galima apmokyti algoritmą dvejomis poromis skirtingų stebėjimų, ir gauti išvestinę tendenciją. Pavyzdžiui, vieno produkto stebėjimo rezultatai gali pasitarnauti kito giminingo produkto pardavimų prognozei. Toliau pateikiami šio algoritmo parametrai – nurodomas pavadinimas, trumpas aprašymas bei pagal nutylėjimą nustatyta parametro reikšmė [20]:

1. *AUTO_DETECT_PERIODICITY* – reikšmė turi būti tarp 0 ir 1, kuo reikšmė arčiau 0, tuo stipresnio periodiškumo duomenys yra atrenkami. Numatytoji reikšmė – 0,6.
2. *COMPLEXITY_PENALTY* – kontroliuoja sprendimų medžio augimą. Reikšmės mažinimas padidiną tikimybę, kad šaka bus perskelta. Numatytoji reikšmė – 0,1.

3. *FORECAST_METHOD* – nustatoma, kuris spėjimų algoritmas bus naudojamas. Galimi algoritmai: *ARTXP, ARIMA* arba mišrus. Pagal nutylėjimą naudojamas mišrusis.

4. *HISTORIC_MODEL_COUNT* – nustatoma kiek istorijos modelių bus generuojama. Numatytoji reikšmė – 1.

5. *HISTORICAL_MODEL_GAP* – nurodo laiko vienetus, kurie gaunami iš modelio ir nustato laiko tarpą tarp dviejų istorijos modelių. Numatytoji reikšmė – 10.

6. INSTABILITY_SENSITIVITY – kontroliuoja spėjimų variacijos ribą, kurią viršijęs algoritmas *ARTXP* sustabdo spėjimų generavimą. Numatytoji reikšmė – 1.

7. MAXIMUM_SERIES_VALUE – parametras naudojamas kartu su *MINIMUM_SERIES_VALUE* parametru kai reikia apibrėžti spėjimus iki numatyto dydžio.

8. MINIMUM_SERIES_VALUE – parametras naudojamas kartu su *MINIMUM_SERIES_VALUE* parametru kai reikia apibrėžti spėjimus iki numatyto dydžio.

9. MINIMUM_SUPPORT – nurodo minimalų reikalingą laiko rėžių kiekį, kuris yra reikalingas šakojimų generavimui kiekviename *Time Series* medyje. Numatytoji reikšmė – 10.

10. MISSING_VALUE_SUBSTITUTION – nurodo, kaip užpildomi tušti laiko rėžiai. Pagal nutylėjimą, tušti laiko rėžiai yra neleidžiami.

11. PERIODICITY_HINT – nurodo algoritmui duomenų rinkinio periodiškumą. Numatytoji reikšmė – 1.

12. *PREDICTION_SMOOTHING* – nurodo, kaip turi būti sudaromas miršus *ARTXP* ir *ARIMA* algoritmas. Galimos reikšmės yra tarp 0 ir 1. Numatytoji reikšmė – 0,5.

2.6.7. Neuroninių tinklų algoritmas

Pavadinimas pakete – *Microsoft Neural Network Algorithm*. Naudojamas didelio kompleksiškumo duomenų analizei. Sudaromas keleto sluoksnių neuroninis tinklas, pasitelkiant klasifikavimo ir regresijos algoritmus. Panašiai kaip sprendimų medžio algoritme, kiekvienam įvesties atributui paskaičiuojama tikimybė, kai žinomi visi galimi išėjimai. Įėjimo sluoksnyje mazgai nėra tarpusavyje surišti, o rišasi tik su viduriniu sluoksniu, o jei jo nėra – su išėjimo sluoksniu. Kiekvienas rezultatas yra netiesinė funkcija, susumuota iš įėjimo funkcijų. Išskiriami neuronų tipai:

1. Įėjimo neuronai – gauna stebėjimo duomenų atskirus atributus; juos gali paskirstyti vienam ar keletui vidurinio sluoksnio neuronams.

2. Vidurinio sluoksnio neuronai – priima duomenis iš įėjimo neuronų ir juos paskirsto išėjimo neuronams.

3. Išėjimo neuronai – generuoja spėjamas atributų reikšmes prie naujų duomenų, kurie taip pat ateina iš vidurinio ar pirmojo sluoksnio neuronų.

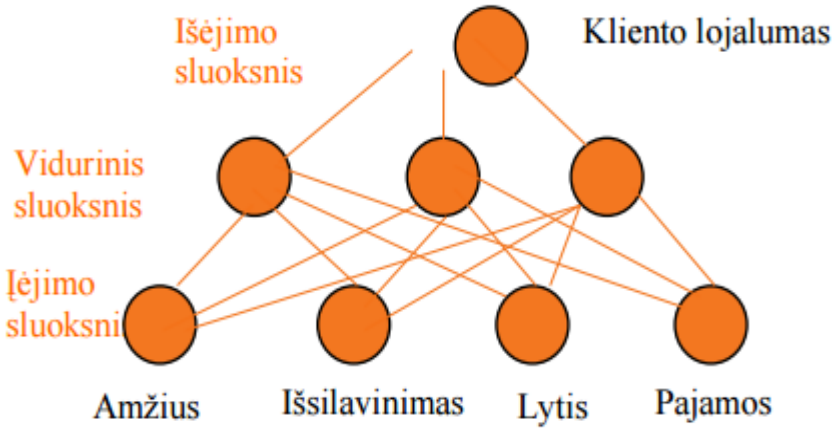
Toliau pateikiami šio algoritmo parametrai – nurodomas pavadinimas, trumpas aprašymas bei pagal nutylėjimą nustatyta parametro reikšmė [21]:

1. *HIDDEN_NODE_RATIO* – daugiklis, apsprendžiantis kiek paslėptojo sluoksnio neuronų bus naudojama. Numatytoji reikšmė – 4.
2. *HOLDOUT_PERCENTAGE* – nurodoma procentinė dalis atvejų, nuo visų, kurie panaudojami algoritmo apmokymo pabaigos sąlygoms rasti. Numatytoji reikšmė – 30.
3. *HOLDOUT_SEED* – pseudo atsitiktinis kintamasis, aukščiau aprašyto parametro duomenų atrinkimui. Numatytoji reikšmė lygi 0. Tai reiškia, kad pseudo atsitiktinumo kriterijus visada priklauso nuo gavybos struktūros pavadinimo. Numatytoji reikšmė – 0.
4. *MAXIMUM_INPUT_ATTRIBUTES* – maksimalus atributų skaičius skirtas algoritmo apmokymui. Jei yra daugiau atributų, automatiškai parenkami charakteringiausi. Numatytoji reikšmė – 255.
5. *MAXIMUM_OUTPUT_ATTRIBUTES* – maksimalus atributų skaičius ryšių atvaizdavimui ir būsenų charakterizavimui. Numatytoji reikšmė – 255.
6. *MAXIMUM_STATES* – maksimalus atributų įgyjamų reikšmių (būsenų) skaičius. Numatytoji reikšmė – 100.
7. *SAMPLE_SIZE* – nurodoma, kiek įrašų bus naudojama algoritmo treniravimu. Numatytoji reikšmė – 10000.

Algoritme išskiriamos tokios fazės:

1. Duomenų išskleidimas ir įvertinimas.
2. Apibrėžiama tinklo struktūra, priklausanti nuo esamų bei spėjamų atributų skaičiaus.
3. Nustatoma kiekvieno požymio galimos būsenos.
4. Apsprendžiama, ar bus, ir jei bus, tai kiek naudojama vidurinio sluoksnio mazgų.
5. Modeliui paduodama papildoma aibė duomenų, patikslinti parametrams, taip pat nustatyti klaidų lygį [21].

8 pav. pateikiamas pavyzdys, kaip formuojamas neuroninis tinklas iš 4 įėjimo sluoksnio atributų ir tarpinės veiklos logikos prognozuoja ekonomiškai naudingą atributą – kliento lojalumą. Taigi, šiuo lojalų klientą apsprendžia jo amžius, išsilavinimas, lytis bei pajamos. Paslėptojo sluoksnio ryšiai algoritmui identifikuojami automatiškai [21].



8 pav. Neuroninio tinklo principinė schema verslo klientų lojalumui įvertinti [21]

2.6.8. Logistinės regresijos algoritmas

Pavadinimas pakete – *Microsoft Logistic Regression Algorithm*. Logistinė regresija yra gerai žinomas statistinis metodas, kuris naudojamas dvejetainiams rezultatams modeliuoti. Panaudojant skirtingus metodus statistikos tyrimuose egzistuoja įvairių logistinės regresijos atmainų. Logistinės regresijos algoritmas paremtas modifikuotu neuroninių tinklų algoritmu. Šis algoritmas turi daug neuroninių tinklų savybių, tačiau yra paprasčiau formuojamas. Logistinė regresiją turi privalumą, nes algoritmo lankstumas leidžia naudoti bet kurio tipo duomenis, be to palaiko keletą skirtingų analitinių užduočių [22]:

1. Demografinių rezultatų prognozavimas pvz., tam tikriems susirgimams prognozuoti.
2. Ištirti ir įvertinti faktorius, kurie įtakoja galutinį rezultatą, pvz. nustatant faktorius, kurie įtakoja naudotojus apsilankant pakartotinai tam tikroje parduotuvėje.
3. Klasifikuoti dokumentus, e–laiškus, arba kitus objektus, turinčius daug atributų.

Taikant šį algoritmą tiesiogiai apsprendžiama, koks ryšys yra tarp įėjimo atributų stulpelių, kad būtų galima nuspėti išėjimo atributo reikšmes. Šiam algoritmui įėjimais gali būti tiek diskretūs, tiek tolydūs atributai. Logaritminės regresijos algoritmas taikomas tekstinės analizės, klasifikacijos bei įvertinimo uždaviniams išspręsti. Toliau pateikiami šio algoritmo parametrai – nurodomas pavadinimas, trumpas aprašymas bei pagal nutylėjimą nustatyta parametro reikšmė [22]:

1. *HOLDOUT_PERCENTAGE* – nurodoma procentinė dalis atvejų, nuo visų, kurie panaudojami algoritmo apmokymo pabaigos sąlygoms rasti. Numatytoji reikšmė – 30.
2. *HOLDOUT_SEED* – pseudo atsitiktinis kintamasis, aukščiau aprašyto parametro duomenų atrinkimui. Numatytoji reikšmė lygi 0. Tai reiškia, kad pseudo atsitiktinumo kriterijus visada priklauso nuo gavybos struktūros pavadinimo. Numatytoji reikšmė – 0.

3. *MAXIMUM_INPUT_ATTRIBUTES* – maksimalus atributų skaičius skirtas algoritmo apmokymui. Jei yra daugiau atributų, automatiškai parenkami charakteringiausi. Numatytoji reikšmė – 255.

4. *MAXIMUM_OUTPUT_ATTRIBUTES* – maksimalus atributų skaičius ryšių atvaizdavimui ir būsenų charakterizavimui. Numatytoji reikšmė – 255.

5. *MAXIMUM_STATES* – maksimalus atributų įgyjamų reikšmių (būsenų) skaičius. Numatytoji reikšmė – 100.

6. *SAMPLE_SIZE* – nurodoma, kiek įrašų bus naudojama algoritmo treniravimu. Numatytoji reikšmė – 10000.

Ruošiant duomenis logistinės regresijos modeliui reikia žinoti konkretaus algoritmo reikalavimus, t.y., kiek reikia duomenų ir kaip duomenys yra naudojami. Logistinės regresijos modelio reikalavimai [22]:

1. Vieno rakto stulpelis (ang. *single key column*). Kiekvienas modelis turi turėti vieną skaitinį arba tekstinį stulpelį, kuris unikaliai identifikuoja kiekvieną įrašą. Neleistini apjungti raktai.

2. Įvesties stulpeliai (angl. *input columns*). Kiekvienas modelis turi turėti bent vieną įvesties stulpelį, turintį reikšmes, kurios analizėje naudojamos kaip faktoriai. Įvesties atributų galima turėti iki 65 tūkstančių, tačiau priklausomai nuo reikšmių skaičiaus kiekviename stulpelyje kiekvienas papildomas stulpelis gali ilginti laiką, kurio reikia modeliui išpildyti.

3. Mažiausiai vienas prognozinis stulpelis. Modelis turi turėti bent vieną prognozinį bet kurio duomenų tipo (įskaitant baigtinių skaičių (angl. *continuous numeric*) duomenis) stulpelį. Prognozinio stulpelio reikšmės taipogi gali būti traktuojamos kaip modelio įvestys, arba galima tiesiog laikyti, jog jos naudojamos tik prognozavimui. Prognoziniams stulpeliams neleistinos įdėtinės (angl. *nested*) lentelės, tačiau jos gali būti naudojamos kaip įvestys.

2.6.9. Tiesinės regresijos algoritmas

Pavadinimas pakete – *Microsoft Linear Regression Algorithm*. Sprendimų medžių algoritmo atmaina, kai žinomų galimų būsenų skaičius yra mažesnis, arba lygus, nei numatoma prognozuojant naujų duomenų porciją. Šiuo atveju niekada negausime dviejų tiesių lygčių, o tik vieną, taigi sprendimų medis niekur neišsišakos. Paprasčiausias taikymo pavyzdys: turime ledų paklausos duomenis, priklausančius nuo temperatūros. Uždavinys nustatyti, kiek reikia užsakyti ledų, jei prognozuojama tam tikra temperatūra, kad nebūtų nei pertekliaus, nei stygiaus [23].

Tiesinę regresiją galima panaudoti norint išaiškinti ryšį tarp dviejų tolydžių stulpelių. Pavyzdžiui, galima panaudoti tiesinę regresiją tendencijų linijos apskaičiavimui gamybos arba pardavimo duomenyse. Tiesinę regresiją galima naudoti ir kaip sudėtingesnių intelektualios analizės

modelių pagrindą, jos pagalba galima nustatyti ryšį tarp dviejų duomenų stulpelių. Nors ir egzistuoja daug tiesinės regresijos skaičiavimo būdų nenaudojant intelektualios duomenų analizės priemonių, tačiau MS tiesinės regresijos algoritmas turi pranašumą sprendžiant šį uždavinį – visų galimų sąryšių tarp kintamųjų skaičiavimas ir patikrinimas yra automatinis. Skaičiavimo metodą, pavyzdžiui mažiausių kvadratų skaičiavimą užduoti nebūtina. Tiesinė regresija gali ženkliai palengvinti ryšius scenarijuose, kur rezultatą įtakoja keletas faktorių. Ruošiant duomenis tiesinės regresijos modeliui reikia atsižvelgti į konkrečių algoritmų reikalavimus. Reikia atsižvelgti į duomenų apimtį ir tai, kaip jie naudojami. Duotam modelio tipui keliami tokie reikalavimai [23]:

1. Vieno rakto atributas**.** Kiekvienas modelis privalo turėti vieną tekstinį arba skaitmeninį stulpelį, kuris unikaliu būdu apibūdina kiekvieną įrašą. Sudėtiniai raktai draudžiami.
2. Prognozuojamas atributas**.** Būtinas bent vienas prognozuojamas stulpelis. Į modelį galima įjungti keletą prognozuojamų atributų, bet jie privalo turėti nuolatinius, skaitmeninius duomenų tipus. Datos duomenų tipo – *datetime* negalima naudoti prognozavimo atributui, net jei saugojimo formatas yra skaitmeninis.
3. Įvesties atributai. Šiuose stulpeliuose privalo būti nuolatiniai skaitmeniniai duomenys ir jie privalo turėti tinkamą duomenų tipą.

Toliau pateikiami šio algoritmo parametrai – nurodomas pavadinimas, trumpas aprašymas bei pagal nutylėjimą nustatyta parametro reikšmė [23]:

MAXIMUM_INPUT_ATTRIBUTES – maksimalus atributų skaičius skirtas algoritmo apmokymui. Jei yra daugiau atributų, automatiškai parenkami charakteringiausi. Numatytoji reikšmė – 255.

MAXIMUM_OUTPUT_ATTRIBUTES – maksimalus atributų skaičius ryšių atvaizdavimui ir būsenų charakterizavimui. Numatytoji reikšmė – 255.

FORCE_REGRESSOR – algoritmui priverstinai nurodomi atributas ar atributai, kurie bus naudojami kaip regresoriai (angl. regressors) nepaistant to, kad algoritmas laiko svarbiais kitus atributus.

2.7. *Microsoft* duomenų gavybos algoritmų taikymai

Tinkamiausio algoritmo parinkimas gali tapti dideliu iššūkiu. Tai pačiai verslo problemai spręsti gali būti taikomi skirtingi algoritmai, pvz., vieną algoritmą pritaikius reikšmingų atributų suradimui, o kitu juos analizuoti. Kiekvienas algoritmas grąžina skirtingus rezultatus, o kai kurie algoritmai gali sugeneruoti ir skirtingų tipų rezultatus, pvz., sprendimų medžių algoritmas gali būti naudojamas ne tik spėjimams, bet ir atributų mažinimui tiriamų duomenų rinkiniuose, nes gali

identifikuoti modeliui nereikšmingus atributus. MS siūlo pačius populiariausius, geriausiai išnagrinėtus duomenų gavybos algoritmus [24].

1 lentelė. MS duomenų gavybos algoritmų taikymas [24]

Užduočių pavyzdžiai	**Tinkami algoritmai**
Tikslių atributų nuspėjimas: 1. Galimų klientų sąrašo suskirstymas į perspektyvius ir neperspektyvius. 2. Serverio sutrikimo tikimybės numatymas 6 mėnesių laikotarpyje. 3. Pacientų rezultatų kategorizacija ir susijusių faktorių radimas.	*Microsoft Decision Trees Algorithm* *Microsoft Naive Bayes Algorithm* *Microsoft Clustering Algorithm* *Microsoft Neural Network Algorithm* *Microsoft Logistic Regression Algorithm*
Tolydžiųjų atributų numatymas: 1. Sekančių metų pardavimų numatymas. 2. Naudotojų nuspėjimas pagal istorinius duomenis. 3. Rizikos taškų skaičiavimas pagal demografinius duomenis.	*Microsoft Decision Trees Algorithm* *Microsoft Time Series Algorithm* *Microsoft Linear Regression Algorithm*
Sekų numatymas: 1. Paspaudimų interneto puslapiuose analizė. 2. Faktorių įtakojančių serverio sutrikimus analizė. 3. Pacientų apsilankymų sekų surinkimas ir analizavimas siekiant suformuluoti geriausias praktikas.	*Microsoft Sequence Clustering Algorithm*
Panašių grupių paieška transakcijose: 1. Pirkinių krepšelio analizė sprendžiant prekių išdėliojimą. 2. Susijusių prekių pasiūlymų kūrimas. 3. Apklausų analizavimas siekiant rasti kurios veiklos ar kabinos koreliuoja.	*Microsoft Association Algorithm* *Microsoft Decision Trees Algorithm*
Panašių grupių paieška: 1. Rizikos grupių kūrimas pagal elgseną ar demografinius duomenis. 2. Naudotojų analizavimas pagal paieškos ir pirkimo atvejus. 3. Serverių identifikavimas, kurie turi panašias charakteristikas.	*Microsoft Clustering Algorithm* *Microsoft Sequence Clustering Algorithm*

1 lentelėje pateikiami algoritmai, sugrupuoti pagal šiuos tipus:

1. Klasifikavimo algoritmai skirti numatyti vienai ar daugiau diskrečių reikšmių.
2. Regresijos algoritmai prognozuoja vieno ar daugiau skaitinių kintamųjų reikšmes.

3. Segmentacijos algoritmai dalina duomenis į grupes ar klasterius, kurie turi panašias ypatybes.

4. Asociacijų algoritmai ieško koreliacijų tarp skirtingų atributų rinkinyje. Dažniausiai taikomi pirkėjų krepšelio analizavimui.

5. Sekų algoritmai ieško pasikartojančių sekų, pvz., pasikartojančio paspaudimų eiliškumo internetinėje svetainėje [24].

3. Duomenų gavybos metodų pritaikymas piratinio turinio analizavimui

ĮSPĖJIMAS: modelis yra kuriamas privačiai kompanijai, norint nepažeisti konfidencialumo sutarties, duomenys bei kai kurie atributai nebus labai tiksliai aprašomi.

Šiame skyriuje aprašomas duomenų gavybos modelio kūrimas taikant CRISP–DM duomenų gavybos metodologiją. Modelis skirtas duomenims iš P2P tinklo klasifikuoti, siekiant nustatyti ar duomenys tinkami pateikti klientams. Darbas atliktas pagal šiuos žingsnius:

1. Pirmame šio darbo skyriuje (žr. 1. Intelektinė nuosavybė ir piratavimas internete) atliekamas tiriamosios veiklos srities aprašymas. Tiriama intelektinės nuosavybės sąvoka, jos rūšys bei autorių teisių gynimo su teisiniais aspektais bei P2P tinklais.
2. Iš kompanijos duomenų sandėlio buvo ištrauktas 7 gigabaitų, 17 atributų duomenų rinkinys, kurį sudaro 20 milijonų įrašų. Darbui sukurta nauja duomenų bazė, gautieji duomenys užkrauti į vieną duomenų bazės lentelę. Atlikta atributų analizė ir jų atrinkimas tolesniam naudojimui.
3. Duomenų lentelės transformacijos ir normalizacijos metu buvo atlikti šie veiksmai:
    - Ištrinti nereikšmingi atributai.
    - Ištrinti pasikartojantys įrašai.
    - Ištrinti įrašai, kurių atributuose nebuvo reikšmių.
    - Sukurtos atskiros lentelės tekstiniams duomenis laikyti ir susietos ryšiais *vienas su daug* su pagrindine lentele.
    - Pagrindinėje lentelėje sukurti raktiniai atributai susieti su išorinėmis lentelėmis, o pagrindinės lentelės tekstiniai laukai pašalinti.
    - Daliai lentelių su tekstiniais atributais sukurti papildomi atributai su tekstinio atributo kopija. Nukopijuotieji atributai išvalyti nuo netekstinių ir neskaitmeninių reikšmių, taip paliekant tik žodžius ir skaičius.
    - Panaudojant išvalytus tekstinius atributus ir pritaikius MS *Term Extracion* transformaciją, buvo sukurtas neigiamų terminų žodynas, kuriame sukaupti žodžiai identifikuojantys apgaulingus (angl. *fake*) P2P bylų įrašus. Dažniausiai tai pornografinį turinį apibrėžiantys terminai.
    - Pasinaudojus MS technologija *Full–Text Search* buvo sukurti terminų katalogai ir indeksai tekstinių duomenų lentelėms.
    - Pasinaudojus *FREETEXT* funkcija ir terminais, gautais iš neigiamų terminų žodyno, atribute FilenameType buvo sužymėti įrašai, kurie potencialiai yra apgaulingi.

- Pasinaudojus *FREETEXTTABLE* funkcija, atlikti tekstinių laukų panašumo skaičiavimai. Įrašai su aukštu rezultatu atribute FilenameType buvo sužymėti kaip galimai tinkami tolesniam apdorojimui sistemoje.
- Pagrindinėje lentelėje dalis skaitinių atributų turėjo labai išsklidusias reikšmes. Išsklidusios reikšmės buvo sugrupuotos į keletą panašaus dydžio grupių, gautoji grupių informacija įrašyta į lentelėje sukurtus papildomus atributus.
- Sukurtos dvi lentelės su duomenimis duomenų gavybos modelių treniravimui ir testavimui.

4. Remiantis 2.7 skyriaus informacija buvo parinkti ir pritaikyti 5 algoritmai tinkantys duomenų klasifikavimui.

5. Pateikus modeliui testavimo duomenis ir pasinaudojus MS *Lift Chart* ir *Classification Matrix* įrankiais, nustatyta kokiu tikslumu veikia algoritmai.

6. Sukurtas modelis pasiūlytas įdiegimui kompanijos duomenų bazių sistemoje.

7. Klasifikavimo modelio tikslas – naudojantis gaunamais duomenimis, kuo tiksliau surūšiuoti torentus, nusprendžiant ar jie yra tinkami tolesniam apdorojimui sistemoje. Tam naudojamas duomenų rinkinys, turintis ir skaitinės, ir tekstinės informacijos. Duomenys tekstiniuose atributuose yra nestruktūrizuoti, t.y., juose daug neraidinių ir neskaitinių simbolių, todėl būtinas atributų valymas.

3.1. Duomenų surinkimas ir suvokimas

Iš kompanijos duomenų sandėlio buvo ištrauktas 7 gigabaitų, 17 atributų duomenų rinkinys, kurį sudaro 20 milijonų įrašų. Darbui sukurta nauja duomenų bazė *ResearchAndDevelopment* ir schema *RRD*. Sukuriama lentelė *P2pDataMining* ir į ją įkrauti gautieji duomenys. Priede A esančioje 2 lentelėje pateikiami *P2pDataMining* lentelės atributai, jų duomenų tipai ir saugomų duomenų aprašymu.

P2pDataMining lentelėje pateikiami duomenys apie dalijamas bylas *BitTorrent* bei kituose P2P tinkluose. Dalis informacijos apie jas yra gaunama vos tik aptikus veiklą tinkle, kita dalis – užregistruojama per ilgesnį laiką toliau renkant informaciją apie dalijimąsi.

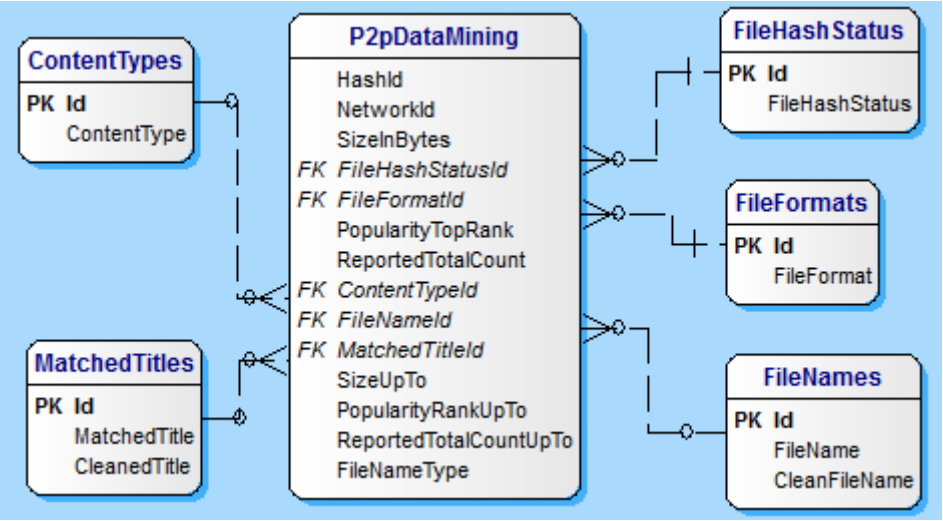
Didžiausią duomenų rinkinio dalį, 70 procentų, sudaro niekada nepatikrini įrašai. Tai tik įrodo, kad būtinas tikrinimo automatizavimas, nes dabartinės sistemos bei žmonės nesugeba peržiūrėti viso atkeliaujančio duomenų srauto. Patikrintų ir įvertintų kaip tinkamų įrašų kiekis – 20%, įvertintų, kaip apgaulingų ar netinkamų – 10%.

3.2. Duomenų transformavimas ir struktūros kūrimas

3.2.1. Lentelės P2pDataMining normalizacija

Pirmiausia buvo ištrinti atributai *VerifiedTitle*, *VerifiedContentType*, nes juose nėra informacijos, kol įrašai nėra apdoroti analitikų ar vidinių sistemų. Taip pat ištrinti atributai *HashText, FileHashStatus, FileFormat,* nes juos identifikuojantys atributai jau egzistuoja lentelėje. Atributai *CreatedAt* ir *FirstSeenAt* ištrinami, nes po poros pirmųjų duomenų gavybos modelio taikymų, buvo nustatyta, kad jie mažai reikšmingi. Taip duomenų rinkinio atributų skaičius sumažinamas iki dešimties.

Ištrynus nereikšmingus atributus, taip pat buvo ištrinti įrašai, kurie bent viename atribute turėjo *NULL* būseną. Pirminiu raktu buvo pasirinktas atributas *HashId*, tačiau dėl prieš tai egzistavusių atributų, duomenų rinkinyje egzistavo pasikartojantys įrašai, todėl dalis jų buvo ištrinti, taip paverčiant *HashId* atributą unikaliu.



9 pav. Normalizuotų lentelių esybių ryšių diagrama

Taikant antrosios normalinės formos reikalavimus, buvo sukurtos lentelės ir ryšiai tarp jų ir pagrindinės lentelės. 9 pav. pateikiama galutinė duomenų bazės esybių ryšių diagramos versija su visomis lentelėmis ir ryšiais tarp atributų, išskyrus dvi lenteles, kurios skirtos duomenų gavybos modelio treniravimo ir testavimo duomenims.

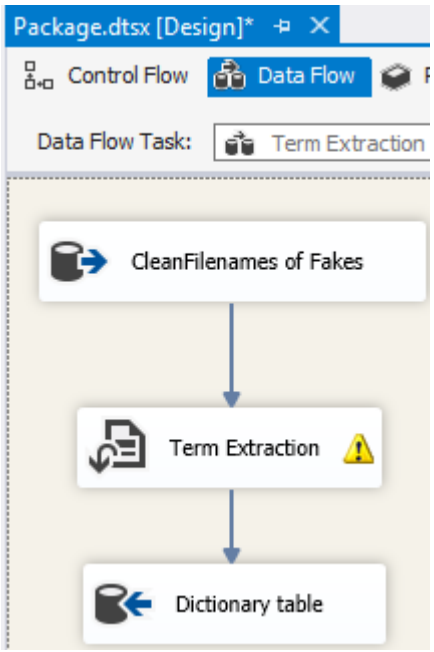
3.2.2. Tekstinių atributų duomenų valymas

Lentelėse *MatchedTitles* ir *FileNames* sukurti papildomi atributai su tekstinio atributo duomenų kopija, atitinkamai *MatchedTitle* į *CleanedTitle* ir *FileName* į *CleanFileName*. Nukopijuotieji atributai išvalyti nuo netekstinių ir neskaitmeninių reikšmių, taip paliekant tik žodžius ir skaičius. Valymui buvo sukurta universali funkcija *dbo.f_RemovePatternFromString*, kuri

konvertuoja ne raidinius skaitmeninius (angl. *nonalphanumeric*) simbolius į tarpo simbolį, taip paliekant tik atskirus žodžius ir tekstus. Sekantis žingsnis buvo pasikartojančių tarpo simbolių šalinimas, nes valymo funkcija prigeneravo pasikartojančių tarpo ženklų duomenyse, kuriuose buvo keletas iš eilės ne raidinių skaitmeninių simbolių. Ši transformacija pritaikyta pastebėjus, kad technologija *Full–Text Search* bei transformacija *Term Extraction* veikia gerokai tiksliau su rankiniu būdu išvalytais duomenimis, nors jų pačių algoritmai sugeba duomenyse išskirti tekstinę ir skaitmeninę informaciją.

3.2.3. *Term Extraction* transformacija

Toliau sekė duomenų, kurie sistemoje jau buvo identifikuoti kaip netinkami, analizė. Pritaikius MS *Term Extracion* transformaciją *CleanFileName* atributui, buvo sukurtas neigiamų terminų žodynas, kuriame sukaupti terminai identifikuojantys apgaulingus P2P tinkle rastus torentų bei kitų bylų įrašus. Dažniausiai tai pornografinį turinį apibrėžiantys terminai. Terminų išgavimui buvo sukurtas MS *SQL Server Integration Services* paketas *Term Extraction*.



10 pav. Term Extracion transformacija

10 pav. pavaizduota terminų išgavimo transformacijos duomenų srauto schema, kuri atliko šiuos veiksmus:

1. Sukuriama užklausa, kuri susieja *FileNames* ir *P2pDataMining* lentas, iš *CleanFileName* ištraukiami tik tiek duomenys, kurie *P2pDataMining* lentoje yra pažymėti kaip netinkami. Duomenys ištraukiami ir pateikiami transformacijai pirmame žingsnyje *CleanFilenames of Fakes.*

2. Term Extraction transformacija iš pateiktų duomenų sudaro terminų žodyną. Transformacijos konfigūracijoje nustatoma, kad būtų ieškomi daiktavardžiai ir daiktavardžių frazės ir skaičiuojami jų pasikartojimai. Nustatoma, kad maksimalus termino ilgis butų 30 simbolių ir rezultatuose būtų pateikiami bent 5 kartus rasti terminai.

3. Gauti rezultatai įrašomi į rezultatų lentelę, kurios dviejuose atributuose saugomi termino ir jo pasikartojimo duomenys.

Įvykdžius transformaciją buvo gautas daugiau nei 4 tūkstančių terminų rinkinys. Ne visi gauti terminai apibrėžė netinkamą turinį, todėl sąrašas buvo peržiūrėtas rankiniu būdu ir atrinkti didžiausią svorį turintys, netinkamą turinį apibrėžiantys terminai. Kaip jau buvo minėta anksčiau, tai dažniausiai su pornografiniu turiniu susiję terminai.

3.2.4. Full–Text Search pritaikymas

Surinkus terminus reikėjo įgyvendinti tekstinių duomenų palyginimą. Testavimui buvo pasirinkta *Full–Text Search* technologija, *Fuzzy Lookup* transformacija bei *LIKE* operatorius. Pastarasis lengvai pritaikomas tačiau, kaip jau minėta skyrelyje 2.5, labai lėtas dirbant su nestruktūrizuotais duomenimis. *Fuzzy Lookup* transformacija pasirodė tinkama, tačiau jos kūrimas ir palaikymas būtų daug pastangų reikalaujanti užduotis. *Full–Text Search* technologija net ir po trumpų nagrinėjimų pasirodė lengvai pritaikoma ir parodė tinkamą rezultatą.

Pasinaudojus MS technologija *Full–Text Search* buvo sukurtas tekstinių duomenų katalogas *TitlesFilenames* ir indeksai tekstinių duomenų lentelėms MatchedTitles ir FileNames. Lentelėje *P2pDataMining* sukuriamas sveikojo skaičiaus tipo atributas *FilenameType*. Reikšmė pagal nutylėjimą – 0. Parašius užklausą su *FREETEXT* operatoriumi, kurio parametrui buvo panaudoti terminai, gauti iš neigiamų terminų žodyno, lentelėje *P2pDataMining* buvo sužymėti beveik 50 tūkstančių įrašų, kurie potencialiai yra netinkami. Tokiems įrašams stulpelyje *FilenameType* suteikta reikšmė – *3*.

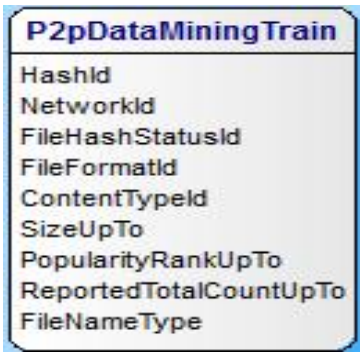
Potencialiai naudingų įrašų radimui buvo pritaikyta *FREETEXTTABLE* funkcija, kuri lygino atributą *CleanedTitle* su *CleanFileName*. Atliekant funkcijos testavimą, buvo pastebėta, kad lyginamųjų reikšmių panašumas yra labai aukštas, jei funkcijos grąžinamuose duomenyse, atributo *RANK* reikšmė yra ne mažesnė nei 150. Tai pastebėjus, ši reikšmė buvo pritaikyta mažai sutampančių rezultatų nufiltravimui. Lentelės *P2pDataMining* įrašai su aukštu *RANK* rezultatu buvo sužymėti kaip galimai tinkami tolesniam apdorojimui sistemoje suteikiant reikšmę *1*.

3.2.5. Išsklidusių duomenų grupavimas

P2pDataMining lentelėje atributai *SizeInBytes, PopularityTopRank* ir *ReportedTotalCount* turi labai išsklidusias reikšmes. Todėl buvo sukurti atributai *SizeUpTo, PopularityRankUpTo, ReportedTotalCountUpTo* ir išsklidusios reikšmės atributuose rankiniu būdu buvo sugrupuotos į 10 panašaus dydžio grupių. Tai buvo atlikta todėl, kad išsklidę duomenys šiame modelyje neatneša naudos, o algoritmams lengviau apdoroti mažesnį kiekį reikšmių.

3.2.6. Treniravimo ir testavimo duomenų rinkinių sudarymas

11 pav. vaizduojama treniravimo duomenų lentelės *P2pDataMiningTrain* struktūra. Verta paminėti, kad po transformacijų ir normalizacijos lentelės dydis sumažėjo nuo 20 Gb iki 800 Mb. Po nereikalingų duomenų šalinimo ir transformacijų, mokymui ir testavimui liko šiek tiek virš milijono įrašų.



11 pav.
Treniravimo duomenų lentelės

Iš lentelės *P2pDataMining* į atskiras, vienodos struktūros lenteles, buvo ištraukti duomenys modelio treniravimui ir testavimui. Atskiras lenteles nuspręsta sukurti tam, kad modelis galėtų greičiau užsikrauti duomenis, be poreikio skanuoti visą *P2pDataMining* lentelę. Mokymui paskirta 75 procentai, testavimui – 25.

3.3. Duomenų gavybos modelio kūrimas

Šiame skyriuje aprašoma kaip kuriama duomenų gavybos modelio struktūra ir kaip panaudojami po transformacijų sukurti duomenų šaltiniai. Į modelio struktūrą įtraukti 5 algoritmai, kurie gali būti taikomi duomenų klasifikavimo uždaviniui spręsti.

3.3.1. Duomenų gavybos projekto kūrimas

Pirmiausia, naudojantis *Visual Studio,* buvo sukurtas *Analysis Services Multidimensional and Data Mining* tipo projektas pavadinimu *P2pDataClassification.* Jo duomenų

šaltinio konfigūracijoje prijungiama anksčiau sukurta duomenų bazė *ResearchAndDevelopment* ir parenkamos lentelės su mokymo ir treniravimo duomenimis.

3.3.2. Modelio struktūros kūrimas ir algoritmų parinkimas

Toliau sekė modelio struktūros *P2p Data Mining Train* sukūrimas. Remiantis 2.7 skyriaus informacija pritaikyti šie 5 algoritmai tinkantys duomenų klasifikavimui:

1. Logistinės regresijos.
2. Grupavimo.
3. Neuroninių tinklų.
4. Sprendimų medžio.
5. *Naive Bayes*.

Mining model structure:

Tables/Columns	Key	Input	Predict...	Content Type	Data Type
P2pDataMiningTrain					
ContentTypeId	☐	☑	☐	Discrete	Long
FileFormatId	☐	☑	☐	Discrete	Long
FileHashStatusId	☐	☐	☑	Discrete	Long
FileNameType	☐	☑	☐	Discrete	Long
HashId	☑	☐	☐	Key	Long
NetworkId	☐	☑	☐	Discrete	Long
PopularityRankUpTo	☐	☑	☐	Discrete	Text
ReportedTotalCountUpTo	☐	☑	☐	Discrete	Text
SizeUpTo	☐	☑	☐	Discrete	Text

12 pav. Modeliui parinktų atributų tipai

12 pav. matome, kad iš lentelės *P2pDataMiningTrain* panaudojami visi atributai. Raktiniu parenkamas atributas *HashId*, prognozavimo – *FileHashStatusId*, visi kiti pažymimi kaip įvesties atributai. Visiems įvesties laukams pažymimas *Discrete* turinio tipas, nes visi atributai turi baigtinį reikšmių skaičių. Jei nors vienas įvesties atributas turi daugiau nei 100 skirtingų reikšmių ir jam priskiriamas *Discrete* požymis, ir jei algoritmas turi *MAXIMUM_STATES* parametrą, būtina jį sukonfigūruoti priskiriant didesnę skaičių, nei atributas turi reikšmių. To neatlikus, algoritmai negalės sėkmingai įvykdyti duomenų apdorojimo, apdorojimo procesas bus sustabdytas. Šiame modelyje didžiausią reikšmių kiekį turi *FileFormatId* atributas, jų yra 220, todėl *MAXIMUM_STATES* parametre nustatoma reikšmė – 220.

Algoritmams būtina pateikti normalizuotus duomenis, raktinis laukas turi būti unikalus. Atliekant pirmus bandymus, algoritmams buvo pateikiama gerokai daugiau duomenų ir atributų, todėl jų apdorojimas truko gerokai ilgiau. Tarpinių bandymų metu apdorojimui buvo naudojami apie 4 milijonai įrašų. Dauguma algoritmų užtrukdavo apie pusvalandį, o neuroninių tinklų algoritmas

užtrukdavo net penkias. Tačiau, atlikus transformacijas ir išrinkus reikšmingus atributus, duomenų apdorojimas sutrumpėjo iki kelių minučių. Algoritmų mokymui panaudota 1,2 milijono unikalių įrašų.

3.4. Modelio tikslumo vertinimas

MS *Visual Studio 2015* programinės įrangos paketas turi *Lift Chart* ir *Classification Matrix* įrankius, kuriais naudojantis lengva nustatyti kokiu tikslumu veikia algoritmai. Prieš pradedant jais naudotis, reikia apmokyti algoritmus atliekant visų treniravimo duomenų apdorojimą. Tada *Input Selection* lange parenkami algoritmai, kurių tikslumas bus lyginamas ir nurodoma, iš kokio šaltinio imami duomenys testavimui. Šiame darbe duomenys gaunami iš *P2pDataMiningTest* lentelės. Atlikus konfigūraciją ir norint pamatyti rezultatus, užtenka perjungti langą pasirinktinai į *Lift Chart* ir *Classification Matrix*. Rezultatų išvedimo į langą trukmė priklauso nuo testavimui pateiktų duomenų kiekio.

13 pav. ir 14 pav. pateikia algoritmų klasifikavimo tikslumo legendą ir grafiką. Iš grafiko matome, kad visi, išskyrus klasterizavimo algoritmą, gavo vienodą įvertinimo balą – 0,99. Prastesnį klasterizavimo algoritmo pasirodymą nulėmė nesugebėjimas nuspėti netinkamų įrašų statuso, tai matoma *15 pav.*, tiek pataikymo, tiek nepataikymo rezultatai lygūs 0.

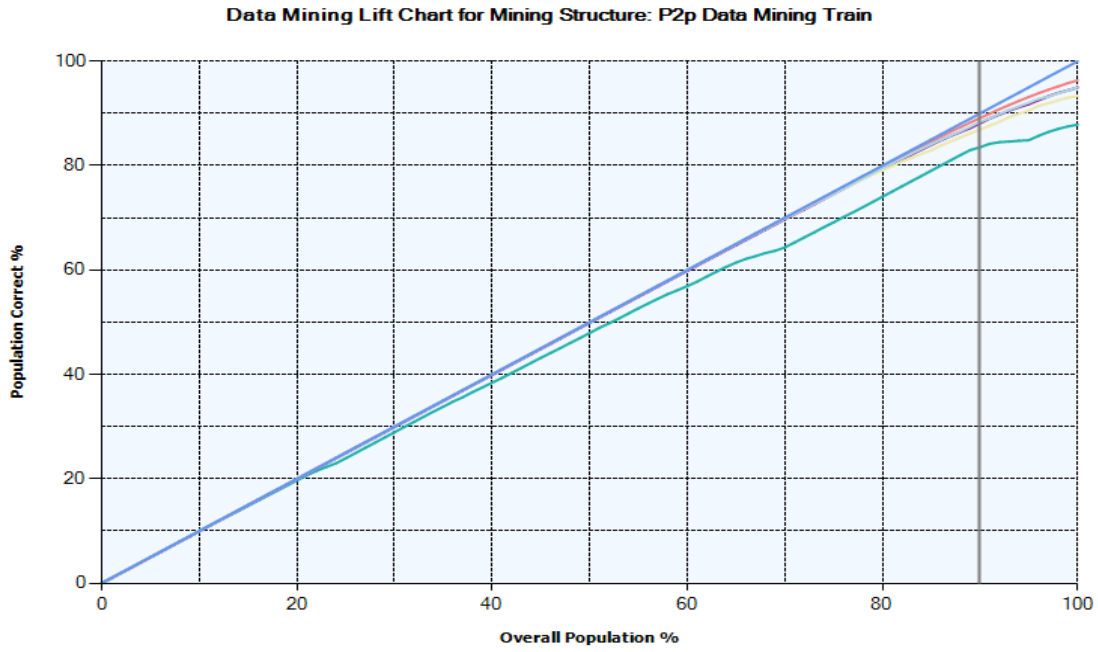
Mining Legend

Population percentage: 90.00%

Series, Model	Score	Population correct	Predict probability
Dtrees	0.99	89.12%	86.62%
Clustering	0.94	83.53%	76.49%
LogReg	0.99	88.15%	74.30%
NBayes	0.99	86.90%	76.07%
NNetwork	0.99	88.39%	79.17%
Ideal Model		90.00%	

13 pav. Lift Chart diagramos legenda

Legendos interpretavimas – pasirenkame testavimo populiacijos dydį (angl. *population percentage*), šiuo atveju 90 procentų, legendos atribute *Population Correct* parodoma kiek iš tų 90–ies procentų algoritmas nustatys teisingai ir su kokia tikimybe. Pvz., sprendimų medžio algoritmas *Dtrees* iš 90–ies procentų teisingai numatys 89,12 procentų populiacijos su 86,62 procento tikimybe. *Lift chart* diagrama vizualizuoja šią informaciją.



14 pav. Lift Chart diagarama

15 pav. pateikiamos nesutapimų matricos kiekvienam algoritmui, kuriose matoma kiek reikšmių buvo atspėta teisingai ir kiek neteisingai. Pagal jų informaciją galima paskaičiuoti kokiu tikslumu procentais veikia algoritmai: sprendimų medžių – 96,33, neuroninių tinklų – 95,01, logistinės regresijos – 94,97, *Naive Bayes* – 93,34, grupavimo – 87,85.

Input Selection | Lift Chart | Classification Matrix | Cross Validation

Columns of the classification matrices correspond to actual values; rows correspond to predicted values

Counts for Dtrees on [File Hash Status Id]:

Predicted	2 (Actual)	14 (Actual)
2	341220	8564
14	5938	39437

Counts for Clustering on [File Hash Status Id]:

Predicted	2 (Actual)	14 (Actual)
2	347158	48001
14	0	0

Counts for LogReg on [File Hash Status Id]:

Predicted	2 (Actual)	14 (Actual)
2	339979	12715
14	7179	35286

Counts for NBayes on [File Hash Status Id]:

Predicted	2 (Actual)	14 (Actual)
2	336407	15577
14	10751	32424

Counts for NNetwork on [File Hash Status Id]:

Predicted	2 (Actual)	14 (Actual)
2	340752	13331
14	6406	34670

15 pav. Klasifikavimo algoritmų nesutapimų matricos

Įvertinus *Lift chart* grafiką ir nesutapimų matricas, galima daryti išvadą, kad tiksliausiai, 96,33 procentų tikslumu, šiame duomenų klasifikavimo modelyje veikia sprendimų medžio algoritmas.

3.5. Modelio įdiegimo planas

Jei įmonė nuspręstų įsidiegti P2P tinklo duomenų klasifikavimo modelį, įdiegimui sistemoje reikėtų sukurti nuoseklų procesą, kuris iš duomenų sandėlio gautų naujus, neklasifikuotus duomenis, juos transformuotų ir suteikęs tinkamą formatą nusiųstų duomenų gavybos modeliui, o šis gautus rezultatus turėtų užregistruoti sistemoje. Šiame darbe sukurta sistema nėra vientisa – duomenų transformavimą atlieka keletas atskirų užklausų, modelis taip pat iškviečiamas rankiniu būdu. Tačiau, tai nebūtų sudėtinga įgyvendinti, nes visi veiksmai atliekamai MS platformoje. Greičiausiai užtektų sukurti vieną kompleksinę duomenų bazės procedūrą, kuri, panaudodama jau sukurtas užklausas ir paketus, vykdytų naujų duomenų klasifikavimą. Taip pat reiktų išspręsti dar keletą verslo klausimų, pvz., ar duomenis reiktų klasifikuoti vos tik juos užregistravus sistemoje. Idealiu atveju klasifikavimą reiktų atlikti visiems naujiems duomenims tačiau, kai kuriuose modelio naudojamuose atributuose duomenų akumuliacija užtrunka ne vieną dieną. Apibendrinant, reikėtų įgyvendinti šiuos žingsnius:

1. Apjungti jau suprogramuotas duomenų transformacijų procedūras ir funkcijas ir sukurti ETL procedūrą, kuri išgavusi iš duomenų bazių reikiamus duomenis gavybos modeliui pateiktų atributus tinkamu formatu.
2. Duomenų gavybos modelyje palikti tik tiksliausiai veikiantį algoritmą, įdiegti modelį į duomenų bazių sistemą atlikti duomenų klasifikavimą.

Įdiegus modelį į sistemą, jo tikslumą reiktų tikrinti nustatytu reguliarumu. Taip pat būtų galima įdiegti automatizuotą klasifikavimo modelio apmokymą iš naujo, nes laikui bėgant atributuose atsirastų naujų kintamųjų ir dėl to modelio tikslumas nukristų.

Išvados

1. Apžvelgus intelektinės nuosavybės sąvoką, jos rūšis, autorių teisių gynimo įstatymus bei P2P tinklų veikimo principus, įsigilinama į piratinės ir antipiratinės veiklos esmę.

2. Atlikus mokslinių straipsnių analizę, kurie nagrinėja duomenų gavybos algoritmus nustatyta, kad įrašų tinkamumo nustatymui reikia pritaikyti duomenų klasifikavimo algoritmus. Kadangi tiriamajame duomenų rinkinyje yra atributų su nestruktūrizuotais duomenimis, taip pat reikėjo išnagrinėti tekstinės duomenų gavybos literatūrą. Tai atlikus, tekstinės duomenų gavybos daliai buvo nuspręsta panaudoti MS programinės įrangos *Term Lookup* ir *Full–Text Search* įrankius. Atlikus transformacijas ir normalizaciją sukurtas struktūrizuotas duomenų rinkinys, kuris tinkamas duomenų gavybos algoritmų apmokymui bei testavimui.

3. Sudarius struktūrizuotą duomenų rinkinį, buvo sukurtas duomenų gavybos modelis ir pritaikyti 5 duomenų klasifikavimo algoritmai:

- Sprendimų medžių – 96,33.
- Neuroninių tinklų – 95,01.
- Logistinės regresijos – 94,97.
- Naivusis Bajeso – 93,34.
- Grupavimo – 87,85.

Greta algoritmo pavadinimo pateikiama jo klasifikavimo tikslumo procentinė išraiška

4. Atlikus klasifikavimo algoritmų testus nustatyta, kad tiksliausiai modelyje veikia sprendimų medžio algoritmas, kurio tikslumas yra 96,33 procento. Taip pat, sukurtas planas, kuriuo vadovaujantis įmonė galėtų įsidiegti duomenų klasifikavimo sistemą.

Literatūros sąrašas:

1. Ben–Gan, I.; Sarka, D.; Talmage, R. 2012. *Querying Microsoft SQL Server 2012*. Sebastopol. 191–215.

2. Business Software Alliance. 2016. *Piratavimo rūšys* [žiūrėta 2016 m. gegužės 02 d.] Prieiga per internetą:
<http://ww2.bsa.org/country/Anti-Piracy/What-is-Software-Piracy/Types%20of%20Piracy.aspx>

3. *Meaning of "internet piracy" in the English Dictionary*. Cambridge dictionaries online. [žiūrėta 2016 m. sausio 25 d.] Prieiga per internetą: <http://dictionary.cambridge.org/dictionary/english/internet-piracy>.

4. Chapman, P., et al. 2000. *CRISP–DM 1.0*. 76 p.

5. Feldman, R.; Sanger, J. 2007. *The text mining handbook*. New York. 423 p.

6. KDnuggets. 2014. *CRISP-DM, still the top methodology for analytics, data mining, or data science projects*. [interaktyvus]. [žiūrėta 2016 m. balandžio 25 d.]. Prieiga per internetą: <http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html>

7. Kiškis M. 2011. *Intelektinės nuosavybės teisinė apsauga elektroninėje erdvėje*. Vilnius. 52–56.

8. Kiškis M. 2012. *Intelektinės Nuosavybės Teisių Gynimas Elektroninėje Erdvėje Pasitelkiant Alternatyvius Internetinius Ginčų Sprendimo Mechanizmus*. Vilnius. 1–10.

9. Klumpp, T. 2013. *File Sharing, Network Architecture, and Copyright Enforcement: An Overview*. Canada. 3–20.

10. Lietuvos Respublikos Seimas. 1999. *Lietuvos respublikos autorių teisių ir gretutinių teisių įstatymas.*Vilnius. 6 p.

11. Lietuvos Vyriausybės kanceliarijos administracinis departamentas. Autentiškas vertimas. 2015. *Konvencija, kuria įsteigiama pasaulinė intelektinės nuosavybės organizacija*. Stokholmas. 15 p.

12. MacLenan, J.; Tang, Z.; Crivat, B. 2009. *Data Mining with Microsoft SQL Server 2008.* Indianapolis. 464–468.

13. MarkMonitor. 2015. *Pirated digital content.* [interaktyvus]. [žiūrėta 2016 m. gegužės 10 d.]. Prieiga per internetą:
<http://www.fox.temple.edu/cms_blogs/wp-content/uploads/sites/20/2016/03/MarkMonitor.infographic.IntellectualPiracy.pdf>

14. Microsoft. 2016. *Data mining concepts.* [interaktyvus]. [žiūrėta 2016 m. gegužės 3 d.]. Prieiga per internetą:
<https://msdn.microsoft.com/en-us/library/ms174949.aspx>
15. Microsoft. 2016. *Microsoft Decision Trees Algorithm.* [interaktyvus]. [žiūrėta 2016 m. gegužės 3 d.]. Prieiga per internetą:
<https://msdn.microsoft.com/en-us/library/ms175312.aspx>
16. Microsoft. 2016. *Microsoft Clustering Algorithm.* [interaktyvus]. [žiūrėta 2016 m. gegužės 3 d.]. Prieiga per internetą:
<https://msdn.microsoft.com/en-us/library/ms174879.aspx>
17. Microsoft. 2016. *Microsoft Naive Bayes Algorithm.* [interaktyvus]. [žiūrėta 2016 m. gegužės 3 d.]. Prieiga per internetą:
<https://msdn.microsoft.com/en-us/library/ms174806.aspx>
18. Microsoft. 2016. *Microsoft Association Algorithm.* [interaktyvus]. [žiūrėta 2016 m. gegužės 3 d.]. Prieiga per internetą:
<https://msdn.microsoft.com/en-us/library/ms174916.aspx>
19. Microsoft. 2016. *Microsoft Sequence Clustering Algorithm.* [interaktyvus]. [žiūrėta 2016 m. gegužės 3 d.]. Prieiga per internetą:
<https://msdn.microsoft.com/en-us/library/ms175462.aspx>
20. Microsoft. 2016. *Microsoft Time Series Algorithm.* [interaktyvus]. [žiūrėta 2016 m. gegužės 3 d.]. Prieiga per internetą:
<https://msdn.microsoft.com/en-us/library/ms174923.aspx>
21. Microsoft. 2016. *Microsoft Neural Network Algorithm.* [interaktyvus]. [žiūrėta 2016 m. gegužės 3 d.]. Prieiga per internetą:
<https://msdn.microsoft.com/en-us/library/ms174941.aspx>
22. Microsoft. 2016. *Microsoft Logistic Regression Algorithm.* [interaktyvus]. [žiūrėta 2016 m. gegužės 3 d.]. Prieiga per internetą:
<https://msdn.microsoft.com/en-us/library/ms174828.aspx>
23. Microsoft. 2016. *Microsoft Linear Regression Algorithm.* [interaktyvus]. [žiūrėta 2016 m. gegužės 3 d.]. Prieiga per internetą:
<https://msdn.microsoft.com/en-us/library/ms174824.aspx>
24. Microsoft. 2016. *Data Mining Algorithms (Analysis Services - Data Mining).* [interaktyvus]. [žiūrėta 2016 m. gegužės 3 d.]. Prieiga per internetą:
<https://msdn.microsoft.com/en-us/library/ms175595.aspx>
25. Microsoft. 2016. *Term Extraction Transformation.* [interaktyvus]. [žiūrėta 2016 m. gegužės 19 d.]. Prieiga per internetą:

< https://msdn.microsoft.com/en-us/library/ms141809.aspx>

26. Microsoft. 2016. *Term Lookup Transformation.* [interaktyvus]. [žiūrėta 2016 m. gegužės 19 d.]. Prieiga per internetą:

< https://msdn.microsoft.com/en-us/library/ms137850.aspx>

27. Niakšu, O. 2014. Duomenų tyryba medicinoje: taikymas, problemos ir galimybės, *Visuomenės sveikata,* Vilnius, 4(67) 12–13

28. Sakalauskas, L. 2012. *Duomenų gavyba*. Vilnius. 29 p.

29. Statsoft, Inc. 2013. *Electronic Statistics Textbook.* [interaktyvus]. [žiūrėta 2016 m. gegužės 16 d.]. Prieiga per internetą:

<http://www.statsoft.com/Textbook/Text-Mining/button/3>

30. Stonkienė, M. 2011. *Intelektinės nuosavybės teisė. Autorių teisė.* Vilnius. 298 p.

31. Tatarūnaitė, A. 2010. *File sharing sistemos: autorių teisių aspektai.* Vilnius. 83 p.

32. Terndurp. 2015. *Building a P2P Peer-Client with Node.js.* [interaktyvus]. [žiūrėta 2016 m. gegužės 3 d.]. Prieiga per internetą:

<http://www.terndrup.net/2015/10/27/Building-a-P2P-Peer-Client-with-Node-js>

33. Vale, Z.; Ramos, C.; Robles, R. 2014. Effective Use of Multiple Random Walks in P2P Networks, *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology.* 4(1) 1–2

Priedas A

2 lentelė. P2pDataMining duomenų lentelė

Atributo pavadinimas	**Atributo tipas**	**Aprašymas**
HashId	[int]	Unikalaus torento maišos kodo identifikatorius.
CreatedAt	[smalldatetime]	Nurodo kada torentas buvo užregistruotas sistemoje.
HashText	[varchar](75)	Unikalus maišos kodas (angl. *hash*), kuris P2P tinkluose nurodo koks failų rinkinys yra torento rinkmenoje.
NetworkId	[tinyint]	Nurodo kokiame P2P tinkle rastas maišos kodas. Tinklo reikšmės saugomos kaip identifikatoriai.
VerifiedTitle	[nvarchar](300)	Nurodo po torento patikrinimo užregistruotą turinio pavadinimą.
VerifiedContentType	[varchar](64)	Nurodo po torento patikrinimo užregistruotą turinio tipą (video, el. knyga, muzika, programinė įranga).
MatchedTitle	[nvarchar](300)	Turinio pavadinimas užregistruotas torentui radimo metu.
ContentType	[varchar](64)	Turinio tipas užregistruotas torentui radimo metu.
Filename	[nvarchar](256)	Torento bylos pavadinimas tinkle.
SizeInBytes	[bigint]	Torento dydis baitais.
FileHashStatusId	[tinyint]	Torento statusas duomenų bazėje.
FileHashStatus	[varchar](40)	Tekstinė statuso reikšmė.
FirstSeenAt	[smalldatetime]	Kada torentas pirmą kartą užregistruotas tinkle.
FileFormatId	[smallint]	Torento failo formato ID sistemoje.
FileFormat	[varchar](40)	Torento failo formatas.
PopularityTopRank	[int]	Nurodo torento populiarumą.
ReportedTotalCount	[int]	Nurodo kiek turinio pažeidimų surasta tinkle iš viso.