Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas

Socialinės žiniasklaidos rinkodaros sprendimų tobulinimas panaudojant didžiųjų verslo duomenų analitikos sprendimus

Baigiamasis magistro projektas

Edvinas Radvilavičius
Projekto autorius

dr. Asta Tarutė
Vadovas
dr. Mindaugas Kavaliauskas
Vadovas

Kaunas, 2018

Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas

Socialinės žiniasklaidos rinkodaros sprendimų tobulinimas panaudojant didžiųjų verslo duomenų analitikos sprendimus

Baigiamasis magistro projektas
Didžiųjų verslo duomenų analitika (621G12002)

Edvinas Radvilavičius
Projekto autorius

dr. Asta Tarutė
Vadovas
dr. Mindaugas Kavaliauskas
Vadovas

doc. dr. Rimantė Hopenienė
Recenzentas
doc. dr. Tomas Ruzgas
Recenzentas

Kaunas, 2018

Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas
Edvinas Radvilavičius

Socialinės žiniasklaidos rinkodaros sprendimų tobulinimas panaudojant didžiųjų verslo duomenų analitikos sprendimus

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Edvino Radvilavičiaus, baigiamasis projektas tema „Socialinės žiniasklaidos rinkodaros sprendimų tobulinimas panaudojant didžiųjų verslo duomenų analitikos sprendimus“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

(vardą ir pavardę įrašyti ranka)	(parašas)

Turinys

Paveikslų sąrašas

Lentelių sąrašas

Radvilavičius, Edvinas. Socialinės žiniasklaidos rinkodaros sprendimų tobulinimas panaudojant didžiųjų verslo duomenų analitikos sprendimus. Magistro baigiamasis projektas / vadovai: dr. Asta Tarutė, dr. Mindaugas Kavaliauskas, Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Fiziniai mokslai, Matematika (01 P).

Reikšminiai žodžiai: Socialinė žiniasklaida, rinkodara, socialiniai tinklai, įsitraukimas

Kaunas, 2018. 56 p.

Santrauka

Šiame darbe buvo siekiama optimizuoti socialinės žiniasklaidos sprendimus panaudojant didžiųjų verslo duomenų analitikos sprendimus. Darbe buvo išsiaiškinti socialinės žiniasklaidos rinkodaros ypatumai bei identifikuota optimizacijos reikalaujanti problema - vartotojo įsitraukimo nesulaukiantys įrašai skelbiami socialiniame tinkle. Identifikavus vartotojų įsitraukimo į įrašą socialiniame tinkle svarbą buvo siekiama sukurti modelį, kuris leistų numatyti ar įrašas taps įtraukiančiu vartotoją. Tokio pobūdžio problema buvo apibrėžta kaip klasifikavimo uždavinys, kuriam spręsti buvo identifikuoti tinkamiausi klasifikavimo metodai – sprendimų medis, atsitiktiniai miškai, neuroniniai tinklai ir logistinė regresija. Sudarius klasifikavimo modelius buvo nustatyta, kad atsitiktinių miškų modelis yra tiksliausiai įrašo įsitraukimą prognozuojantis modelis. Atlikus literatūros ir tyrimo metodų analizę bei įvertinus sudaryto modelio tikslumą, galima daryti išvadą, kad socialinės žniasklaidos sprendimai socialiniame tinkle „Facebook“ gali būti optimizuoti didžiųjų verslo duomenų analitikos sprendimais.

Radvilavičius, Edvinas. Social media marketing decisions improvement through big data analytics decisions. Master's Final Degree Project / supervisors: dr. Asta Tarutė, dr. Mindaugas Kavaliauskas; The Faculty of Mathematics and Natural Sciences, Kaunas University of Technology.

Research area and field: Natural Sciences, Mathematics (01 P)

Keywords: social media, marketing, social network, engagement

Kaunas, 2018. 56 pages.

Summary

The aim of Master's final project was to improve social media marketing decisions through big business data analytics. The work revealed the peculiarities of social media marketing and identified an issue requiring optimization – posts that do not get enough engagement from social network users. By identifying the importance of consumer engagement, the aim was to create a model that would predict whether the post would become engaged. This kind of problem was defined as a classification task the most appropriate classification methods were found to solve it - decision tree, random forest, neural networks and logistic regression. By building the classification models, it has been found that the random forest model is the most predictive model for predicting the engagement of a post. After analyzing the literature and research methods and evaluating the accuracy of the model, it can be concluded that social media marketing solutions in the social network „Facebook“ can be improved by using big data analytics solutions.

Įvadas

Socialinių ryšių mezgimas ir palaikymas yra svarbi kiekvieno žmogaus gyvenimo dalis. Dėl technologinės pažangos vartotojai turi priėjimą prie socialinių tinklų įvairiausiose gyvenimo situacijose. Socialiniai tinklai sukuria galimybę palaikyti ryšius ne tik vartotojams, bet ir prekės ženklams pasiekti vartotojus. Dauguma socialinių tinklų sukuria galimybę verslams turėti savo paskyrą, kuria galima pristatyti save bei savo produktus ar teikiamas paslaugas bei reikšti nuomonę kuriant pranešimus. Tokių veiksmų vykdymas priskiriamas socialinės žiniasklaidos rinkodarai.

Socialinė žiniasklaida tampa vis svarbesne rinkodaros dalimi ir leidžia įmonėms pasiekti didelę auditoriją. Pagrindinis socialinės žiniasklaidos kanalas yra internetiniai socialiniai tinklai. Socialinių tinklų atsiradimo pradžioje šis rinkodaros kanalas buvo laikomas pigiu ir lengvu būdu pasiekti vartotoją. Tačiau per keletą pastarųjų metų socialiniai tinklai tapo itin populiariu rinkodaros kanalu į kurį įsitraukė beveik visas komercinis pasaulis.

Dėl didelės konkurencijos socialiniuose tinkluose, tam, kad turėti įsitraukusią auditoriją ir patrauklią socialinio tinklo paskirą, reikia skirti laiko ir pastangų. Verslams turint ribotus resursus socialinei žiniasklaidai, tampa itin aktualu šio kanalo rinkodaros procesų optimizavimas.

Socialiniai tinklai, juose vykdoma socialinė žiniasklaida bei vartotojų interakcija su ja sukuria didelius duomenų kiekius. Šių duomenų analizė bei didžiųjų verslo duomenų analitikos sprendimų taikymas gali padėti pagerinti socialinės žiniasklaidos rinkodaros sprendimus.

Problema:

Kaip ir kokie socialinės žiniasklaidos rinkodaros sprendimai gali būti tobulinami panaudojant didžiųjų duomenų analitikos sprendimus?

Tikslas:

Pagrįsti socialinės žiniasklaidos rinkodaros sprendimų tobulinimo galimybes panaudojant didžiųjų verslo duomenų analitikos priemones.

Uždaviniai:

1. Išsiaiškinti socialinės žiniasklaidos ypatumus bei identifikuoti optimizacijos reikalaujančias problemas;
2. Remiantis atliktos mokslinės literatūros analizės rezultatais, parengti tyrimo metodiką;
3. Sudaryti matematinį modelį, galintį optimizuoti socialinės žiniasklaidos rinkodaros sprendimus;

4. Pateikti socialinės žiniasklaidos rinkodaros sprendimų tobulinimo galimybes panaudojant didžiųjų verslo domenų analitikos priemones apibendrinančias išvadas ir rekomendacijas.

1. Teoriniai socialinės žiniasklaidos panaudojimo rinkodaroje sprendimai

1.1. Socialinės žiniasklaidos samprata ir jos ypatumai

Ralfe ir Renklodo atliktos apklausos rodo, kad 70% vartotojų tiesioginę reklamą laiko nepatikima [1]. Tuo tarpu Hali atliktas tyrimas atskleidė, kad socialinės žiniasklaidos rinkodara skatina klientų pasitikėjimą ir turi tiesioginį poveikį vartotojo ketinimui įsigyti prekę ar paslaugą [2]. Tobulėjančios ir lengvai prieinamos technologijos sudaro sąlygas greitai kurti, talpinti ir platinti informaciją. Socialinė žiniasklaida yra pritaikoma rinkodaroje ir keičia tiesioginę reklamą.

Rinkodaroje socialinė žiniasklaida naudojama siekiant vartotoją pritraukti kuriamu turiniu. Pagal Steenburgh, rinkodaros strategiją galima suskirstyti į įeinančią (ang. *Inbound*) ir išeinančią (ang. *Outbound*) [3]. Scott tokius tradicinius rinkodaros metodus, kaip skambučius klientams, elektroninio pašto rinkodarą, TV reklamą bei reklamą paštu, priskiria išeinančiai rinkodarai [4]. Pasirinkus tokį rinkodaros metodą yra tiesiogiai siūloma pirkti tam tikrą produktą ar paslaugą. Įeinančios rinkodaros tikslas yra kliento pritraukimas netiesogiai, kad jis pats atrastų prekės ženklą [4]. To siekiama kuriant vertingą ir įtraukiantį turinį ir skelbiant jį įvairiu formatu – straipsniais, vaizdo įrašais, elektroninėmis knygomis. Šiam turiniui skleisti pasitelkiama interneto paieškos variklių optimizacija (angl. *serch engine optimization)* ir socialinė žiniasklaida [4].

Scott teigimu, įeinanti rinkodara yra efektyvesnė nei išeinanti, nes vartotojus per dieną pasiekia didelis pranešimų kiekis, siūlančių įsigyti prekes, tačiau šie pranešimai dažnai yra nepastebimi [4]. Scott taip pat pabrėžia, kad interneto naudotojai dažnai tikslingai vengia ir blokuoja tiesioginės reklamos turinį naudodami įvairias technines priemones, tokias kaip „adblock" įskiepius ir „spam" filtrus, o reikalingas prekes ir paslaugas susiranda patys, naudodami paieškos sistemas interneto tinkle ar socialiniuose tinkluose [4]. Luke teigia, kad socialinė žiniasklaida yra efektyvesnis būdas pasiekti didelį skaičių klientų nei skambučiais, e. laiškais ar susitikimais [5]. Mize išskiria vieną pagrindinį socialinės žiniasklaidos privalumą lyginant su tradiciniais rinkodaros kanalais – mažesnius kaštus ir tą taip pat grindžia tuo, kad tokia rindaros forma leidžia lengviau ir greičiau pasiekti plačią tikslinių vartotojų auditoriją [6]. Tačiau pabrėžia, kad rinkodaros kanalai turi būti diversifikuoti ir negalima pasikliauti tik vienu [6]. Pagal Kelly rinkodaros plėtojimas naudojant socialinę žiniasklaidą bus sėkmingas tik tuo atveju, jeigu bus suderintas su kitais rinkodaros kanalais [7]. Blanchard papildo šią nuomonę teigdamas, kad socialinė žiniasklaida ir jos skleidžiamas turinys turėtų būti suderinta ne tik su kitais rinkodaros kanalais, bet ir su visomis įmonės funkcijomis bei strategija [8]. Dėl šios priežasties įmonei svarbu turėti aiškią turinio kryptį.

Laskey siūlo rinkodaros turinį skirstyti į informatyvų ir transformacinį [9]. Informacinis turinys pabrėžia faktinius duomenis apie parduodamą produktą ar paslaugą, o transformacinis orientuotas į kliento emocijas ir patirtį. Lee ir Hosanagar siūlo klasifikuoti turinį į informacinį ir įtikinantį [10]:

- Informacinis turinys sudarytas iš informacijos apie produktus, akcijas, pasiūlymus, likučius ir visa kita reikalinga informacija, palengvinančia įsigijimo sprendimą.
- Įtikinamu turiniu siekiama paveikti žmogaus emocijas – naudojamos įžymybės, labdara, socialinės akcijos.

Lee ir Hosanagar taip pat pabrėžia, kad įtikinantis turinys turi teigiamą įtaką socialinės žiniasklaidos pasiekiamų vartotojų įsitraukimui ir pripažįsta, kad informatyvus turinys traukia vartotoją tik tuo atveju, jeigu toks turinys yra naudojamas kartu su įtikinančiu [10]. Socialinės žiniasklaidos rinkodarai tinka abu turinio tipai. Scott teigimu, socialinės žiniasklaidos rinkodarą galima vystyti šiais kanalais [4]:

- Diskusijų forumai - virtuali platforma internete keistis mintimis ir patirtimi kuriant temas ir įrašus. Dažnai forumai vystosi lengvai, nereikalaudami ypatingos priežiūros ar reklamos, ypač būdami lankytojus pritraukiančių informacinių portalų dalimis.
- Tinklaraščiai – internetiniai dienoraščiai, kuriuose vienas arba keli autoriai išsako savo mintis apie faktus ir įvykius, skleidžia idėjas.
- Elektroninio pašto komunikacija – turimai klientų bazei reguliariai siunčiami elektroniniai laiškai su jiems aktualia informacija.
- Internetiniai socialiniai tinklai – interneto platforma vienijanti tam tikrą, bendrų interesų turinčią narių grupę, kuri kuria svetainės turinį ir virtualiai bendrauja tarpusavyje.

Interneto socialinių tinklų atsiradimas sudarė sąlygas kiekvienam asmeniui ar įmonei bendrauti su šimtais ar net tūkstančiais kitų socialinio tinklo vartotojų. Tuo tarpu Įmonės pasiekia vartotojus naudodamos socialinę žiniasklaidą socialiniuose tinkluose. Dėl suteikiamos bendravimo su klientais galimybės bei vis augančio internetinio turinio populiarumo Mangaulas ir Fauldsas teigia, kad socialinė žiniasklaida socialiniuose tinkluose turėtų būti laikoma rinkodaros komplekso (angl. *promotional mix*) dalimi [11]. Atsižvelgiant į tai šiame darbe išskirtinis dėmesys skiriamas socialiniams tinklams.

Pagal Goldenberą, vartotojai kuria profilius socialiuose tinkluose ne tam, kad pirktų prekes ar domėtųsi prekės ženklais [12]. Visgi, pagal Ertell, vartotojai tikisi rasti savo mėgstamą prekės ženklą populiariausiuose socialiniuose tinkluose [13]. Autorius taip pat teigia, kad sėkminga komunikacija

su tokiu vartotoju galėtų padidinti jo lojalumą prekės ženklui bei padidinti tikimybę įsigyti produktą. Iresonas prideda, kad socialinės žiniasklaidos rinkodara turėtų sukurti jaukų draugiškumo jausmą [14]. Taigi, socialinės žiniasklaidos rinkodaros egzistavimas socialiniuose tinkluose vartotojui nėra nepriimtinas, tačiau reikalauja subtilesnių priemonių iš prekės ženklų nei tradicinė tiesioginė rinkodara. Tuo tarpu prekės ženklai įgyvendindami savo socialinės žiniasklaidos rinkodaros tikslus turi prisitaikyti prie vartotojo lūkesčių.

Pagal Kelly, socialinės žiniasklaidos rinkodara socialiniuose tinkluose turi tris pirminius tikslus – susidomėjimo didinimas (angl. *brand awareness*), susidomėjimo pirkimu generavimas (angl. *generating leads*) ir esamų klientų išlaikymas (angl. *retaining customers*) [7].

- Žinomumo didinimas yra skirtas atkreipti vartotojų dėmesį į prekės ženklą. Žinomumo didinimas yra dažniausiai pasirenkamas socialinės rinkodaros tikslas, nes socialinė žiniasklaida leidžia pasiekti plačią tikslinių vartotojų auditoriją mažesnėmis pastangomis nei tradiciniais rinkodaros kanalais [7]. Pagrindinis būdas įgyvendinti šį tikslą yra kurti reklamines kampanijas, kurios būtų įsimenamos [7]. Kelly teigimu, prekinio ženklo žinomumo didinimas yra sudėtingas procesas, reikalaujantis pastangų, siekiant sudominti tikslinę vartotojų auditoriją ir priversti ją dalintis prekės ženklo kuriamu turiniu, tačiau šį tikslą įvardina kaip lengviausiai pamatuojamą [7].

- Susidomėjimo didinimo tikslas - rasti kuo daugiau pardavimo galimybių. Tai yra socialinės žiniasklaidos rinkodaros panaudojimas, skatintis paslaugų ar prekių pardavimą [7]. Dauguma kompanijų siekia padidinti savo pardavimus, todėl susidomėjimo didinimo tikslas gali atrodyti itin patrauklus. Socialinė žiniasklaida taip pat gali būti pritaikyta šiam tikslui. Susidomėjęs klientas yra tas, kuris išreiškė susidomėjimą teikiama paslauga ar produktu palikdamas savo kontaktus ar kokia nors kita forma, priklausomai nuo verslo modelio ir industrijos [7].

- Klientų išlaikymas skirtas išsisaugoti esamus klientus, kad jie toliau naudotųsi teikiamomis paslaugomis arba toliau reguliariai pirktų prekes. Paskutinis Kelly pateikiamas socialinės žiniasklaidos tikslas - esamų klientų išlaikymas. Peppardo teigimu esamų klientų išlaikymo kaštai yra mažesni, nei naujų klientų pritraukimo [15]. Todėl ne visada efektyvu susitelkti tik į klientų bazės didinimą ir gavus naują klientą jį iškart pamiršti. Esamų klientų duomenų bazė gali būti panaudojama pajamų didinimui, jeigu yra žinoma, kaip šiems klientams suteikti tai ko jie nori [7]. Galiausiai, Kelly socialinę žiniasklaidą įvardina kaip puikų kanalą siekiant abipusės komunikacijos ir grįžtamojo ryšio iš esamos klientų bazės ir išskiria dvi šio tikslo

siekimo teikiamas naudas – klientų aptarnavimo gerinimą ir pajamų iš esamų klientų didinimą [7].

Kiekvienam iš apibrėžtų tikslų pasiekti reikalingos skirtingos socialinės žiniasklaidos rinkodaros priemonės. Įmonės strategija gali siekti įgyvendinti visus tris tikslus, tačiau tai būtų labai sunkiai įgyvendinama dėl resursų apribojimų [7].

Įmonių vadovai nori suprasti, kaip socialiniuose tinkluose vykdoma socialinė žiniasklaida bei jos tikslų siekimas prisideda prie pagrindinių įmonės finansinių rodiklių. Pagal Kelly, dažnai yra siekiama įvertinti, kokią įtaką socialinė žiniasklaida turi pardavimams, pajamoms ir kaštams [7]. Porterfieldo atliktas tyrimas parodė, kad 65% apklaustų įmonės vadovų mano, jog jų kompanijos nepajautė padidėjusių pardavimų ar pelno dėl socialinės žiniasklaidos rinkodaros [16]. Tačiau toks vertinimas yra abejotinas, nes tame pačiame tyrime 36% respondentų teigė, jog jie neturi pakankamai duomenų, kad teisingai išanalizuoti socialinės žiniasklaidos investicijų gražą. Galiausiai tyrimas parodė, kad įmonės, kurios įvardijamos kaip sėkmingos, buvo du kartus labiau tikėtinos turėti aktyvią socialinio tinklo paskyrą bei priskirtą socialinės žiniasklaidos rinkodaros vadybininką [16]. Remiantis šiuo tyrimu, galima teigti, kad socialinę žiniasklaidą ir jos įtaką finansiniams rodikliams yra itin sunku įvertinti. Tačiau, pagal Kelly, socialiniai tinklai suteikia galimybę sekti kiekvieną kliento interakciją su prekės ženklo leidžiamu turiniu, priešingai nei tradiciniuose rinkodaros kanaluose, tokiuose kaip laikraščiai, televizija, radijas [7]. Taigi, nors ir sunkiai įvertinima finansiniais rodikliais, socialinės žiniasklaidos rinkodara socialiniuose tinkluose yra vienas iš labiausiai pamatuojamų rinkodaros kanalų.

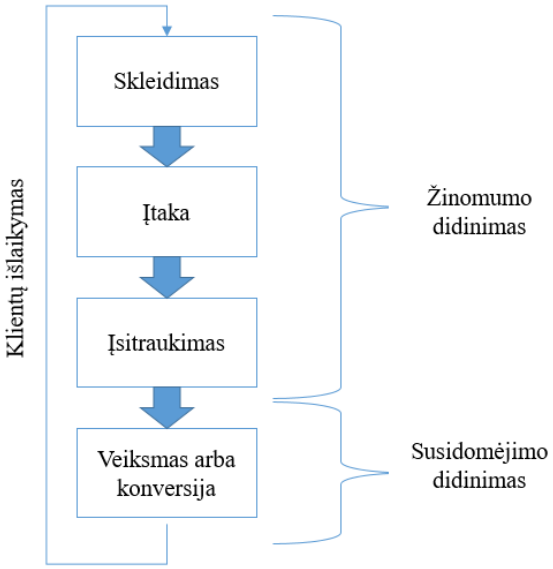
Esant ribotoms galimybėms taikyti finansinius rodiklius, kitas socialinės žiniasklaidos vertinimo būdas yra pardavimų piltuvėlio principo taikymas (angl. *sales funnel*). Klasikinis pardavimų piltuvėlis susideda iš trijų žingsnių – susipažinimo, apsvarstymo ir pirkimo [17]. Tai metodologija, kuri leidžia kiekybiškai įvertinti kliento žingsnius nuo susipažinimo su prekės ženklu iki prekės įsigijimo. Kiekviename žingsnyje lieka vis mažesnis vartotojų skaičius. Įvairūs autoriai pateikia skirtingą piltuvėlio detalumo lygį. Piltuvėlio žingsniai, per kuriuos turės praeiti klientais, taip pat gali skirtis priklausomai nuo industrijos ir pardavimų ciklo [7]. Havenas, kaip paskutinį piltuvėlio lygį, siūlo vertinti kliento lojalumą, o ne prekės ar paslaugos įsigijimą [18]. Tačiau Haveno teigimu tradicinis pardavimų piltuvėlis netinka vertinti socialinei žiniasklaidai dėl šių priežasčių [18]:

- Socialinės žiniasklaidos atveju piltuvėlio vidurys susideda iš itin daug skirtingų faktorių.
- Vertingas klientas ne visada yra tas, kuris perka arba naudojasi paslauga dažnai. Kartais klientas, kuris perka rečiau, tačiau visada įvertiną prekę ir palieka atsiliepimą gali tapti

vertingesniu negu dažnai perkantis, kadangi jo paliktas grįžtamasis ryšys gali daryti įtaką dideliam skaičiui būsimų pirkėjų, atsižvelgiančių į atsiliepimą.

- Tradicinio piltuvėlio rodikliai negali aprėpti visų socialinės žiniasklaidos faktorių – nėra įvertinamas klientų įsitraukimas, nuomonės formuotojai, grįžtamasis ryšys, komentarai ir diskusijos.

Kelly pritaikė pardavimų piltuvėlio principą socialinei žiniasklaidai atsižvelgdamas į socialinės žiniasklaidos tikslus (žr 1 pav) [7].



1 pav. Pardavimų piltuvėlio schema, pritaikyta socialinės žiniasklaidos rinkodarai

Matomumo, įtakos ir įsitraukimo rodikliai siejasi su žinomumo didinimo tikslu, o veiksmas arba konversija - su susidomėjimo didinimo tikslu. Pirmi trys piltuvėlio žingsniai skirti žinomumo didinimo tikslui pasiekti. Kiekvieno rodiklio paaiškinimas:

- Matomumas (angl. *exposure*) yra ženklo pasiekiamumo matas. Pirmasis piltuvėlio rodiklis vertina klientų pasiekiamumą. Kuo daugiau prekės ženklas yra matomas, tuo daugiau žmonių jį įsimena, todėl didinant pasiekiamumą, didėja tikimybė, jog įvyks pardavimas [7].

- Įtaka (angl *influence*) – tai matas, parodantis, kaip įtaką darantys vartotojai padėjo skleisti turinį. Įtakos rodikliu vertinami vartotojai, kuriuos pasiekia prekės ženklas per nuomonės formuotojus (srities ekspertai, žinomi žmonės) [7]. Frebergas, Grahamas ir McGaughey socialinių tinklų nuomonės formuotojus apibrėžia kaip trečiąją šalį, esančią tarp vartotojo ir prekinio ženklo, kuri formuoja auditorijos požiūrį naudojantis socialinių tinklų suteikiamomis galimybėmis [19]. Nuomonės formuotojai turi ištikimus sekėjus, gerbėjus ir skaitytojus, kurie pasitiki jų nuomone. Nuomonės formuotojo paveikti klientai yra labiau linkę pirkti produktą,

dėl šios priežasties Kelly šiuos žmones, veikiamus nuomonės formutojų, išksiria į atskirą piltuvėlio rodiklį [7].

- Įtraukimas (angl. *engagement*) – tai rodiklis, parodantis kiek žmonių turėjo kontaktą su prekės ženklu socialinio tinkle platformoje. Šiuo rodikliu vertinamas žmonių įsitraukimas į prekinio ženklo rinkodarą socialiniame tinkle. Tai yra itin svarbus socialinės žiniasklaidos rinkodaros efektyvumo rodiklis [7].

- Sekančiame piltuvėlio žingsnyje vertinami su susidomėjimo tikslu sietini rodikliai. Veiksmo arba konversijos rodiklis leidžia įvertinti susidomėjimo didinimo tikslo siekimo rezultatus. Šis piltuvėlio rodiklis vertina galutinį veiksmą, kurio siekia prekės ženklas kurdamas socialinio tinklo rinkodaros kampaniją [7]. Tačiau jis gali būti skirtingas priklausomai industrijos srities bei verslo modelio. Kelly įvardina du populiariausius prekės ženklų siekiamus klientų veiksmų – pirkimas arba kontaktų pasidalinimas [7].

Socialinė žiniasklaida naudojama siekiant pritraukti vartotoją kuriamu turiniu, vietoje to, kad būtų kreipimasi į vartotoją tiesiogiai su tikslu parduoti prekę ar paslaugą. Socialinio tinklo platforma yra pagrindinis kanalas vystyti socialinės žiniasklaidos rinkodarą. Socialinės žiniasklaidos rinkodaros vystymas socialiniuose tinkluose yra vartotojui priimtinas reiškinys bei galintis turėti teigiamos įtakos parduodant prekę ar paslaugą. Tačiau tikslas, dėl kurio prekės ženklai pasirenka socialinės žiniasklaidos rinkodarą, yra ne tik pardavimų didinimas, bet ir žinomumo didinimas bei esamų klientų išlaikymas. Nors ir ne visada lengvai atsispindinti finansinėse ataskaitose, socialinės žiniasklaidos rinkodaros kuriama nauda ir efektyvumas yra išmatuojami rodikliais, tokiais kaip matomumas, įtaka, įsitraukimas.

1.2. Socialinių interneto tinklų panaudojimo galimybės: socialinio tinklo „Facebook“ atvejis

Pagal rinkos tyrimų ir konsultacijų bendrovės „TNS“ atliktą tyrimą, 2015 m. „Facebook“ buvo populiariausias socialinis tinklas Lietuvoje [20]. Tyrimo duomenimis prie savo „Facebook“ paskyros kasdien prisijungia beveik 37 proc. 15-74 m. amžiaus šalies gyventojų. To paties tyrimo metu nustatyta, kad antras populiariausias socialinis tinklas Lietuvoje yra „Youtube“ su 21 proc. kasdien prisijungiančių 15-74 m. amžiaus Lietuvos gyventojų. Į „Facebook“ socialinio tinklo populiarumą Lietuvoje reaguoja ir įmonės skirdamos reklamos biudžetą šiam socialiniam tinklui (žr 1 lentelę) [21].

1 lentelė. Interneto reklamos pasiskirstymas interneto kanaluose

Interneto reklamos kanalas	**2017 m. skirta interneto reklamos biudžeto dalis, %**	**2018 m. planuojama interneto reklamos biudžeto dalis, %**
Banerinė reklama lietuviškuose portaluose	36	32
Reklama “Facebook” soc. tinkle	15	14
Reklama paieškos sistemose (“Google”)	11	12
Mobilioji reklama	10	11
Banerinė reklama užsienio portaluose („Google Display Network“)	9	10
Video reklama lietuviškuose vaizdo portaluose	8	9
Video reklama “Youtube” kanale	7	8
Kiti reklamos kanalai	5	5

Lentelėje pateikti Lietuvos įmonių išlaidų socialinių tinklų reklamai duomenys pagal „TNS“ 2018 metais atliktą tyrimą. Reklama „Facebook“ socialiniame tinkle sudarė 14% bendro Lietuvos internetinės reklamos biudžeto. „Youtube“ reklamai tenkanti dalis 8%, o „Google“ tenkanti dalis sudarė 11% numatyto biudžeto. Palyginus su JAV rinka, „eMarketer“ duomenimis, „Google“ sudaro 40.7% o „Facebook“ sudaro 19.7% JAV internetinės rinkodaros reklamos išlaidų [22]. Atsižvelgiant į „Facebook“ socialinio tinklo populiarumą Lietuvoje, šiame darbe pasirinkta tobulinti socialinės žiniasklaidos sprendimus „Facebook“ socialiniame tinkle.

Kiekvienas internetinis socialinis tinklas turi skirtingą funkcionalumą. Plačiąja prasme, neišskiriant asmeninių Facebook paskyrų nuo verslo, Acaras ir Polonsky „Facebook“ panaudojimo galimybes suagreguoja iki galimybės susikurti profilius bei tyrinėti kitų profilius, taip įgaunant informacijos apie kitų gyvenimo stilių ir pomėgius [23]. Kircova ir Enginkaya „Facebook“ socialinio tinklo galimybes apibendrina, kaip komunikacijos, kooperacijos ir dalinimosi informacija tarp asmenų, bendruomenių ir verslų, įgalinimo priemonę [24]. Ryanas in Jonesas išskiria panaudojimą ryšio ir santykio palaikymui ir kūrimui [25]. Bussas ir Straussas akcentuoja žmonių būrimąsi į bendruomenes „Facebook“ socialiniame tinkle, su tikslu dalintis informacija tarp panašių interesų žmonių [26]. Pagal Eley ir Tilley ši informacija gali būti įvairios formos – nuotraukos, būsenos pasikeitimai, vaizdo įrašai ir nuorodos į svetaines [27]. Pagal Kietzmaną, Kristopherį ir Silvestrą socialiniai tinklai panaudojami diskutuoti ir dalintis šia įvairia informacija [28]. Asuro ir Hubermano

teigimu socialiniai tinklai, tarp jų ir „Facebook“, panaudojami kaip erdvė, kurioje vartotojai gali kuria turinį ir juo dalintis [29]. Gretzelis taip pat pabrėžia, kad pagrindinė socialinio tinklo panaudojimo galimybė yra turinio kūrimas [30]. Apibendrintos nagrinėjamo socialinio tinklo panaudojimo galimybės pateiktos 2 lentelėje.

2 lentelė. „Facebook“ socialinio tinklo panaudojimo galimybės vartotojui

Autorius, metai	**„Facebook“ socialinio tinklo panaudojimo galimybės vartotojui**
Gretzelis, 2006	Kurti turinį
Acaras ir Polonsky, 2007	Kurti profilius ir tyrinėti kitų profilius
Ryanas in Jonesas, 2009	Ryšio ir santykių kūrimas bei ryšio ir santykių palaikymas
Bussas ir Strausas, 2009	Būrimasis į bendruomenes bei dalinimasis informacija
Asuras ir Hubermanas, 2010	Kurti turinį ir juo dalintis
Kietzmanas, Kristopheras ir Silvestras, 2011	Dalintis informacija ir ją aptarinėti
Kircova ir Enginkaya, 2015	Komunikuoti, kooperuotis ir dalintis informacija

Kaip matoma, dauguma analizuotų autorių prieina prie išvados, kad svarbiausia „Facebook“ socialinio tinklo panaudojimo galimybė yra dalinimasis informacija ir būrimasis į bendruomenes, kurias sieja bendra informacija.

Skleisti ir dalintis informacija „Facebook“ socialiniame tinkle asmeniniams ir verslo profiliams sudarytos skirtingos sąlygos, nes šios dvi paskyrų rūšys turi skirtingą funkcionalumą. Facebook taisyklės draudžia asmeniniame profilyje atstovauti bet ką kitą, išskyrus save, o šios taisyklės nesilaikymas gali privesti prie paskyros praradimo [31]. Todėl nerekomenduotina kurti asmeninėms paskyros su tikslu ja reprezentuoti verslą. Facebook pateikia tokius verslo profilio skirtumus nuo asmeninio [31]:

- Verslo profilio funkcionalumas labiau pritaikytas reprezentuoti verslui nei asmeninis.
- Verslo profiliui suteikiamas priėjimas prie puslapio įžvalgų funkcijos (angl. *page insigthts*). Ši funkcija leidžia matyti įvairius rodiklius, kurie leidžia įvertinti puslapio skelbiamos informacijos pasiekiamumą bei matyti įvairius demografinius duomenis apie vartotojus, kuriuos pasiekė verslo puslapio skelbiami pranešimai. Asmeninių paskyrų naudotojai šios funkcijos neturi.
- Prisijungimą prie verslo profilio gali turėti bet kas, kam savininkas suteikia teises. Priėjimui reikalingas asmeninis „Facebook“ profilis.

- Verslo profiliui leidžiama kurti reklamas bei užsisakyti mokamas paslaugas, kurios pagerina profilio pranešimų pasiekiamumą. Asmeninės paskyros neturi šios galimybės.

Ramsaranas įvardina keturis „Facebook“ verslo paskyros panaudojimo atvejus [32]:

- Verslo paskyros susikūrimas – sukuriamas „Facebook“ puslapis, kuriame galima reprezentuoti verslą.
- Dalinimasis įvykiais ir renginiais – verslas gali reaguoti į bet kurį kitą įrašą ir juo dalintis.
- Reklamos paslaugos – naudojantis mokamomis reklamos paslaugomis verslo profilio įrašai tampa dažniau rodomi vartotojams.
- Asmeninės žinutės – verslai gali sulaukti jiems adresuojamų žinučių iš „Facebook“ vartotojų. Ši funkcija leidžia panaudoti socialinį tinklą kaip klientų aptarnavimo kanalą.

Pagal Threattą pagrindinė „Facebook“ panaudojimo galimybė verslui yra informacijos pateikimas apie produktus ir teikiamas paslaugas [33]. Berkowitzas teigia, kad, verslo paskyra gali atlikti dvi funkcijas – identifikavimo ir skatinimo [34]. Tokios veiklos kaip reklama ir pranešimai atlieka skatinimo funkciją, o tokios kaip grįžtamojo ryšio iš klientų gavimas ir informacijos rinkimas vykdomi verslų atlieka identifikavimo funkciją [34]. Apibendrinus minėtų autorių pateiktas „Facebook“ socialinio tinklo panaudojimo galimybes verslui , matoma, kad pagrindiniai aspektai yra savęs reprezentavimas ir grįžtamojo ryšio iš kliento gavimas, kas iš dalies sutampa su prieš tai 2 lentelėje pateiktomis „Facebook“ vartotojo galimybėmis.

Aktyvus verslo reprezentavimas „Facebook“ socialiniame tinkle vykdomas kuriant įrašus (angl. *posts*). Įrašai yra vienintelė galimybė skleisti turinį ir vystyti socialinės žiniasklaidos rinkodarą „Facebook“ socialiniame tinkle. Įrašas gali būti sudarytas iš įvairių detalių – teksto, nuotraukų, vaizdo įrašų arba iš šių dalių rinkinio. Įrašai pasiekia vartotoją atsirasdami jo naujienų sklaidos kanale (angl. *news feed*) [35]. Socialinio tinklo „Facebook“ vartotojai pasirenka sekti jiems aktualius „Facebook“ puslapius, kad matytų jų pranešimus savo naujienų sklaidos kanale. Tačiau ne visų vartotojo sekamų puslapių įrašai patenka į vartotojo sklaidos kanalą. Taip pat ne visus puslapio sekėjus pasiekia puslapio skelbiami įrašai. Galiausiai vartotojas nebūtinai turi būti puslapio sekėju, kad jį pasiektų puslapio įrašas [35]. Verslo profilio įrašas „Facebook“ socialinio tinklo vartotoją gali pasiekti dviem būdais – mokama reklama arba organiniu pasiekiamumu:

- „Facebook“ mokama reklama padidina puslapio įrašo parodymo vartotojui tikimybę suteikiant šiam įrašui aukštesnį prioritetą patekti vartotojui matomame įrašų sraute [36]. Toks įrašo pasiekiamumo didinimo būdas vadinamas reklamos kampanija. Konfiguruojant tokią reklamos

kampaniją leidžiama pasirinkti, kokią auditoriją norima pasiekti kuriamu įrašu. Auditorija apibrėžiama pagal lokaciją, lytį, amžių ir kitus rodiklius. Tokiai reklamos kampanija yra skiriamas biudžetas bei nustatomas laikas, kada norima, jog klientas matytų įrašą.

- Organinis pasiekiamumas yra įrašo savaiminis plitimas už jį nemokant. Pagrindinis įrašų organinio pasiekiamumo didinimą įtakojantis aspektas yra aktyvi vartotojų reakcija į įrašą [36]. Vartotojų reakciją apibrėžia prieš tai minėtas įsitraukimo rodiklis. Pagal Angelesą, ar įrašo įsitraukimo rodiklis pakankamai aukštas priklauso nuo socialinio tinklo algoritmo, kuris nusprendžia ar šis įrašas aktualus bei vertas būti parodytu [37].

Kadangi „Facebook“ socialiniame tinkle įrašai yra skelbiami nuolatos, vyksta itin didelė konkurencija tarp jų dėl patekimo į vartotojo sklaidos kanalą. „Facebook“ algortimai neparodo visų puslapio įrašų kiekvienam puslapio sekėjui, todėl pagrindinis šių algoritmų veikimo tikslas yra atrinkti kuo aktualesnį turinį vartotojui [38]. „Facebook“ vyriausiojo naujienų sklaidos kanalo reitingavimo inžinieriaus (angl. *Engineering Manager for News Feed Ranking*) Kacholia teigimu, „Facebook“ organinio ir mokamo įrašo paplitimą nusprendžiantys algoritmai yra paremti mašininio mokymo principais ir vertina virš tūkstančio skirtingų parametrų [39]. Tačiau yra keli reikšmingiausi kriterijai. Ballingsas teigimu, kuo didesni komentarų, pasidalinimų ir „patinka“ paspaudimų skaičius, tuo didesnis įrašo organinis pasiekiamumas [40]. Youngo teigimu, organinis pasiekiamumas vedamas stipraus turinio, kai bendruomenė sąveikauja su įrašu naudodami „patinka“ paspaudimus, pasidalinimus ir komentarus [41]. Angeleso teigimu, kiekviena interakcija („patinka“ paspaudimas, pasidalinimas, komentaras) turi skirtingą svorį [37]. Jo teigimu labiausiai turinio plitimui įtaką daro pasidalinimai, po kurių pagal svarbumą seka komentarai, dar mažiau svarbūs “patinka” paspaudimai ir, galiausiai, paspaudimai ant nuorodos esančios įraše. Galiausiai, „Socialbakers“ teigimu, „Facebook” algoritmai reaguoja ne į nominalų komentarų, pasidalinimų ir “patinka” paspaudimų skaičių, o į jų santykį su puslapio sekėjų skaičiumi [42]. „Socialbakers“ teigimu, šiam rodikliui viršijus 0.1 reikšmę, „Facebook“ algoritmai įrašą pateikia daugiau nei 30% verslo puslapio sekėjų [43].

3 lentelė. Faktoriai nulemiantys įrašo patekimą į „Facebook“ vartotojo sklaidos kanalą

Autorius, metai	**Kriterijus**
„Socialbakers“, 2012	Komentarų, pasidalinimų, "patinka" paspaudimų skaičiaus santykis su puslapio sekėjų skaičiumi
Angeles, 2014	Komentarai, pasidalinimai, "patinka" paspaudimai, paspaudimai ant nuorodos - mažėjančia tvarka
Youngas, 2014	Komentarai, pasidalinimai, "patinka" paspaudimai
Ballingsas, 2016	Komentarai, pasidalinimai, "patinka" paspaudimai

3 lentelėje pateiktame įrašų plitimo kriterijų apibendrinime matoma, kad kliento interakcija su turiniu komentarais, pasidalinimais ar „patinka“ paspaudimais yra reikšmingas kriterijus, siekiant, kad įrašas pasiektų vartotoją organiškai. Todėl kuriamas socialinės žiniasklaidos rinkodaros turinys turi būti orientuotas į kuo didesnę vartotojų interakciją su juo.

Atlikus „Facebook” socialinio tinklo panaudojimo socialinėje rinkodaroje galimybių analizę buvo atskleista, kad verslo profilio įrašai gali pasiekti klientą keliais būdais – mokama reklama arba organiškai. Atlikta „Facebook” organinio pasiekiamumo algoritmo analizė leidžia manyti, kad kuriant vartotoją įtraukiančius įrašus galima pasiekti auditoriją be reklamos išlaidų. Todėl kokybiško ir aktualaus turinio kūrimas leis pasiekti siekiamą auditoriją be apmokamos reklamos. Organinis pasiekiamumas yra vedamas stipraus turinio, kai vartotojai sąveikauja su įrašu naudodami „patinka“ paspaudimus, pasidalinimus ir komentarus, todėl akivaizdu, kad įsitraukimas yra itin svarbus rodiklis tokio pobūdžio rinkodaroje. Dėl šios priežasties, išskirtinis dėmesys sutelkiamas į turinio kūrimo proceso tobulinimą, siekiant, kad skelbiami įrašai taptų įtraukiantys vartotoją.

1.3. Vartotojo įsitraukimą į skelbiamus įrašus lemiantys veiksniai

Įsitraukimas padeda prekės ženklams skleisti informaciją globaliai ir greitai. Vartotojų įsitraukimas į socialinės žiniasklaidos turinį prisideda prie turinio skleidimo, kadangi „Facebook“ socialinio tinklo platformoje vartotojo atliktas veiksmas būna matomas kitiems jį sekantiems ar kitaip susijusiems asmenims. Ascendo ir Gerbero atliktos apklausos parodė, kad įsitraukimo rodiklio gerinimo siekimas yra vienas iš svarbiausių socialinės žiniasklaidos tikslų prekės ženklams [44]. Lee, Hosangerio ir Nairo teigimu, socialinės žiniasklaidos rezultatų nebūtina tiesiogiai susieti su pardavimais [45]. Šie autoriai teigia, kad verta vertinti ir siekti pagerinti vien patį įsitraukimo rodiklį, neatsižvelgiant į pardavimus, nes šis rodiklis apibrėžia prekės ženklo stiprumą ir įsitraukimas kuria naudą prekės ženklui [45].

Tačiau Cvijikj ir Michahellesas taip pat pabrėžia, kad vartotojų įsitraukimas taip pat turi įtakos vartotojo psichologijai – vartotojas, įsitraukęs į prekės ženklo socialinės žiniasklaidos rinkodarą, yra labiau linkęs įsigyti prekę ar paslaugą [46]. Kelly apibrėžia įsitraukimo metriką, kaip socialinio tinklo vartotojo reakciją į turinį bei teigia, kad įsitraukimas yra pardavimo prognozės rodiklis, nes klientas, turėjęs kontaktą su prekės ženklo turiniu, yra labiau linkęs įsigyti prekę ar paslaugą [7]. Taip pat, įsitraukimo iš vartotojų susilaukianti socialinė žiniasklaida sukuria pasitikėjimo jausmą kitiems vartotojams, kurie prieš tai neturėjo kontakto su turiniu [47].

Cvijikj atliktas tyrimas, kuriame buvo analizuojama skirtingų industrijų socialinės žiniasklaidos rinkodarą „Facebook“ socialiniame tinkle, atskleidė, kad turinio tipas (pramoginis ar informacinis),

pateikimo tipas (vaido įrašai, nuotrauka, nuoroda) ir įrašo paskelbimo laikas turi įtakos prekės ženklo sekėjų įsitraukimui [46]. Kowk ir Yu, pasirinkę „patinka“ paspaudimų ir komentarų skaičiaus sumą, kaip įrašo populiarumo matą, nustatė, kad įrašai su nuotraukomis sulaukia daugiau populiarumo negu įrašai su nuorodomis ar vaizdo įrašais [48]. Coursaris atliktas tyrimas atskleidė, kad kliento įsitraukimui teigiamos įtakos turi įrašo turiningumas – teksto ilgis, nuotraukos, vaizdo įrašai ar nuorodos į kitus puslapius [49]. Hosanageris teigia, kad turinys su emociniais ar filantropiniais aspektais taip pat daro įtaką vartotojo įsitraukimui, tačiau pabrėžia, kad informacinis turinys, toks kaip produkto kainos ir privalumai, mažina įsitraukimą [45]. Sun, Su ir Reynoldso teigimu, įrašai, kuriuose naudojamas humoras ir tiesioginis skatinimas įsitraukti, sulaukia daugiau sekėjų susidomėjimo. Šie autoriai taip pat teigia, kad klausiamąja forma pateikti įrašai susilaukia daugiau komentarų [50]. Malhotras teigimu, nuotraukos, reagavimas į to meto aktualijas, skatinimo įsitraukti pateikimas pranešimo tekste, edukuojantis turinys, emocijos, humoras, įrašo naujumas yra įsitraukimą įtakojantys aspektai [51]. 4 lentelėje pateikti analizuotų autorių įrašo įtraukiamumo faktoriai.

4 lentelė. Nagrinėtų autorių pateikiami faktoriai, darantys įtaką įrašo įtraukiamumui

Autorius, metai	**Faktoriai, darantys įtaką įrašo įtraukiamumui**
Cvijikj , 2013	Turinio tipas, pateikimo tipas, paskelbimo laikas
Kowk ir Yu, 2013	Nuotraukos
Malhotras ir See, 2013	Nuotraukos, reagavimas į to meto aktualijas, skatinimo įsitraukti pateikimas pranešimo tekste, edukuojantis turinys, emocijos, humoras, įrašo naujumas
Hosangeris, 2014	Emociniai ir filantropiniai aspektai, informacinis turinys
Sun, Su ir Reynoldsas, 2015	Humoras, skatinimas įsitraukti, klausiamojo pobūdžio įrašai
Coursaris, 2016	Teksto ilgis, nuotraukos, vaizdo įrašai, nuorodos

Matoma, kad įrašo įtraukiamumą nulemia įvairios įrašo turinio detalės. Autoriai nėra vieningos nuomonės šiuo klausimu. Tačiau galima teigti, kad įrašo įsitraukimui prognozuoti reikalingas platus kintamųjų spektras.

Socialinė žiniasklaida yra pritaikoma rinkodaroje ir keičia tiesioginę reklamą. Rinkodaroje socialinė žiniasklaida naudojama siekiant vartotoją pritraukti kuriamu turiniu. Įeinanti rinkodara yra efektyvesnė nei išeinanti, nes vartotojus per dieną pasiekia didelis pranešimų kiekis, siūlančių įsigyti prekes. Tuo tarpu įmonės pasiekia vartotojus naudodamos socialinę žiniasklaidą socialiniuose tinkluose. Nors ir ne visada lengvai atsispindinti finansinėse ataskaitose, socialinės žiniasklaidos rinkodaros kuriama nauda ir efektyvumas yra išmatuojami rodikliais, tokiais kaip matomumas, įtaka,

įsitraukimas. "Facebook" buvo identifikuotas kaip populiariausias socialinis tinklas Lietuvoje. Į „Facebook“ socialinio tinklo populiarumą Lietuvoje reaguoja ir įmonės skirdamos reklamos biudžetą šiam socialiniam tinklui. Verslo profilio įrašas „Facebook“ socialinio tinklo vartotoją gali pasiekti dviem būdais – mokama reklama arba organiniu pasiekiamumu. „Facebook" algoritmų analizė atskleidė, kad kliento interakcija su turiniu komentarais, pasidalinimais ar „patinka“ paspaudimais yra reikšmingas kriterijus, siekiant, kad įrašas pasiektų vartotoją organiškai. Todėl kuriamas socialinės žiniasklaidos rinkodaros turinys turi būti orientuotas į kuo didesnę vartotojų interakciją su juo.

2. Tyrimo metodai

Įrašo surenkamas „patinka“ paspaudimų, pasidalinimų ir komentarų skaičius priklauso nuo tą įrašą skelbiančio puslapio sekėjų skaičiaus. Todėl nominalus įsitraukusių vartotojų skaičius nėra tikslus įrašo įsitraukimo matas. „Socialbakers“ siūlomas įrašų įsitraukimo rodiklis, apibrėžtas kaip vartotojų interakcijos ir puslapio sekėjų santykis, yra teisingesnis būdas apibrėžti ar įrašas yra sėkmingas vartotojų įsitraukimo atžvilgiu. Šiai reikšmei pasiekus daugiau nei 0.1 įrašas tampa pastebėtas „Facebook“ algoritmų ir skleidžiamas platesnei klientų auditorijai. Šiame darbe pasirinkta optimizituoti socialinės žiniasklaidos sprendimus, sukuriant modelį, galintį prognozuoti ar įrašas taps pakankamai įtraukiančiu ir pralauš minėtą įsitraukimo rodiklio ribą.

Identifikavus vartotojų įsitraukimo į įrašą socialiniame tinkle svarbą bus siekiama sukurti modelį, kuris leistų numatyti ar įrašas taps įtraukiančiu vartotoją. Įtraukiantis įrašas apibrėžtas kaip įrašas, kurio įsitraukimo rodiklis pasiekia didesnę nei 0.1 reikšmę. Tokio pobūdžio problema yra klasifikavimo uždavinys. Kadangi įrašus skelbiantys puslapiai yra itin skirtingo pobūdžio, prieš juos klasifikuojant bus atliekamas klasterizavimas pagal jų skelbiamos socialinės rinkodaros turinio pobūdį. Taip siekiama patikrinti ar klasifikavimas atskiruose klasteriuose pagerintų modelio tikslumą. Turint klasterius bus atliekamas klasifikavimas atskiruose klasteriuose ir klasifikavimas visam duomenų rinkiniui. Siekiant atlikti šiuos veiksmus reikalingas socialinio tinklo „Facebook“ įrašų duomenų rinkinys. Šiam žingsniui buvo pasitelktas minėto socialinio tinklo siūlomas „Facebook Graph API“

Tyrimo problema

Kaip išvengti kliento susidomėjimo nesusilaukiančių įrašų publikavimo „Facebook“ socialiniame tinkle?

Tyrimo tikslas

Sukurti klasifikavimo modelį, kurį naudojant būtų galima patikrinti, ar įrašas bus sėkmingas vartotojo įsitraukimo atžvilgiu.

Tyrimo uždaviniai

1. Išgauti ir apdoroti „Facebook“ socialinio tinklo įrašų duomenis;
2. Atlikti aprašomąją analizę siekiant atrasti reikšmingus kintamuosius įrašų klasifikavimui;

3. Atlikti puslapių klasterizavimą pagal jų įrašuose skelbiamą socialinės žiniasklaidos rinkodaros pobūdį;

4. Sudaryti klasifikavimo modelį įrašų sėkmingumui nustatyti bei įvertinti gautus modelius;

2.1. Duomenų išgavimas

„Facebook Graph API“ yra pagrindinis būdas programiškai prieiti prie „Facebook“ viešų įrašų duomenų. Tai žemo lygio HTTP pagrįsta aplikacijų programavimo sąsaja, kurią programos gali naudoti, siekiant kurti duomenų užklausas, skelbti naujus įrašus, tvarkyti skelbimus, įkelti nuotraukas ir atlikti įvairias kitas užduotis. "Graph API" yra pavadintas pagal "socialinio grafo" idėją - informacijos „Facebook“ socialiniame tinkle mechanizmą. Šį grafą sudaro:

• Mazgai (angl. *nodes*) - atskiri objektai, pvz., naudotojas, nuotrauka, puslapis arba komentaras;

• Kraštai (angl *edges*)- jungtys tarp objektų kolekcijos ir vieno objekto.

• Laukai (angl *fields*) - duomenys apie objektą

Paprastai naudojant mazgus gaunamos objektų kolekcijos viename objekte, o naudojant laukus, gaunami duomenys apie vieną objektą ar kiekvieną kolekcijos objektą. Laukai naudojami, norint nurodyti, kuriuos duomenis norima įtraukti į atsakymus. Jei norima gauti konkrečius duomenis (vadinamų laukų) apie mazgą, galima įtraukti parametrų laukus ir nurodyti, kuriuos laukus norima gauti kreipimosi atsakyme [52].

Populiariausių socialinio tinklo „Facebook“ puslapių Lietuvoje sąrašas išgaunamas naudojantis „Socialbakers“ platforma [53]. Ši platforma suteikia galimybę matyti visus kiekvienos šalies „Facebook“ puslapius, pagal juos sekančių vartotojų skaičių. Šio darbo tyrimui pasirinkti Lietuvos „Facebook“ puslapiai.

2.2. Klasterizavimo metodai

Siekiant klasterizuoti duomenų rinkinį, buvo atlikta klasterizavimo metodų analizė. 4 lentelėje pateiktas populiariausių klasterizavimo metodų apibendrinimas [54].

5 lentelė. Klasterizavimo algoritmų apžvalga

Metodas	**Parametrai**	**Pritaikomumas**	**Metrika**
K-vidurkių (angl. *k-means*)	Klasterių skaičius	Platus panaudojimas, panašaus dydžio klasteriai, mažas klasterių skaičius.	Atstumas tarp taškų
Santykių dauginimo (angl. *Affinity propagation*)	Slopinimas, imties prioritetas	Daug klasterių, skirtingo dydžio klasteriai	Grafinis atstumas
Vidurkio pokyčio (angl. *mean shift*)	Pralaidumas	Daug klasterių, skirtingo dydžio klasteriai	Atstumas tarp taškų
Spektrinis klasterizavimas (angl. *spectral clustering*)	Klasterių skaičius	Mažas klasterių skaičius, skirtingo dydžio klasteriai	Grafinis atstumas
Wardo hierarchinis klasterizavimas (angl. *Ward hierarchical clustering*)	Klasterių skaičius	Daug klasterių, jungimo apribojimai	Atstumas tarp taškų
Kaupiamasis (angl. *Agglometavive clustering*)	Klasterių skaičius, jungties tipas	Daug klasterių, jungimo apribojimai	Bet koks porinis atstumas
DBSCAN	Kaimynystės dydis	Skirtingo dydžio klasteriai	Atstumas tarp artimiausių taškų
Gauso mišinių metodas (angl. *Gaussian mixtures*)	Didelis skaičius skirtingų parametrų	Patogus tankumo vertinimui	Atstumas iki centrų
Birčo (angl. *Birch*)	Šakojimasis	Dideliems duomenų rinkiniams, išskirčių šalinimas	Euklido atstumas

Pagal klasterizavimo algoritmo pritaikomumą palankiausias ir turimą duomenų kiekį darbe pasirinkta naudoti K-vidurkių klasterizavimo metodą. Šis klasterizavimo K-vidurkių klasterizavimo algoritmas apjungia taškus į klasterius minimizuodamas skirtumus tarp taškų ir maksimizuodamas atstumą tarp klasterių. Šis algoritmas reikalauja parametruose nurodyti klasterių skaičių. Taigi tikslinė funkcija naudojama gerinti padalijimo kokybę, kad objektai, esantys klasteryje, būtų kuo panašesni, tačiau kuo mažiau panašūs su objektais esančiais kituose klasteriuose. Šis klasterizavimo metodas yra centrų padalijimu grįstas metodas. Toks klasterizavimo metodas tinka dideliems duomenų kiekiams klasterizuoti bei yra naudojamas daugelyje įvairių sričių

Algoritmas susideda iš keturių žingsnių:

1. Iš duomenų rinkinio parenkami pirminiai klasterių centrai;

2. Kiekvienas objektas yra perskirstomas į artimiausią klasterį, pagal tame klasteryje esančių objektų vidurkį;

3. Perskaičiuoti klasterių vidurkius

4. Kartoti 2-3 žingsnius, kol taškams priskiriami klasteriai nesikeis.

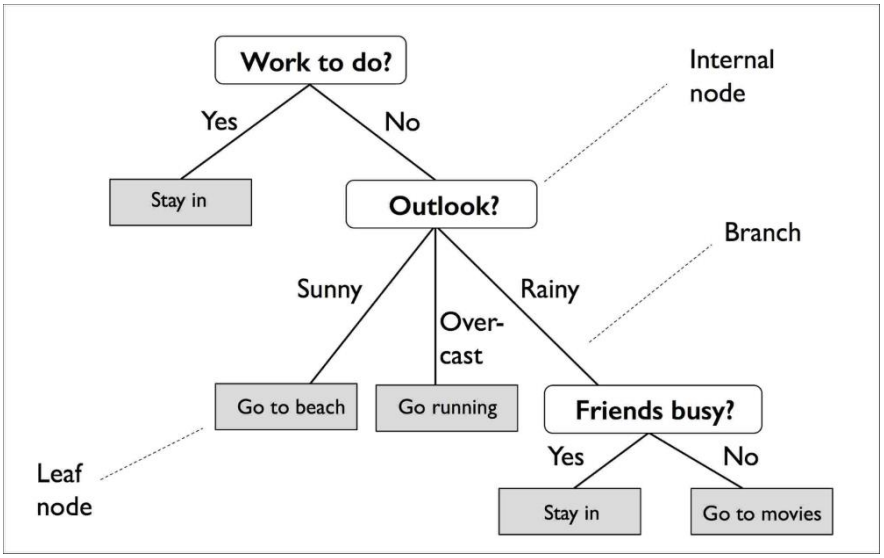
Šis klasterizavimo metodas sudarytas centrų padalijimo principu naudoja klasterio centrą C_i, kuris apibrėžia klasterį. Klasterio centru laikomis jo centrinis taškas. Centroidas gali būti apibrėžiamas įvairiais būdais, tokiais kaip klasteriui priskirtų objektų vidurkis ar mediana. Skirtumas tarp objekto, atstovaujančio klasterį, matuojamas dist(p, c_i), kur dist(x, y) yra euklido atstumas tarp dviejų taškų x ir y. Klasterio C_i kokybę galima išmatuoti pagal klasterio vidaus variaciją - kvadratinės paklaidos sumą tarp visų objektų C_i ir centroidų ci, kurie apibrėžiami kaip [55] :

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, c_i)^2 \qquad (2.2.1)$$

2.3. Klasifikavimo metodai

Sprendimų medis

Sprendimų medžių klasifikatoriai yra patrauklūs modeliai, kai aktualus modelio paaiškinamumas. Sprendimų medžio klasifikavimo metodu sudaroma diagrama apibūdinanti sprendimų seką atsižvelgiant į duomenis. 2 paveikslėlyje pateiktas sprendimų medžio pavyzdys.



2 pav. Sprendimų medžio schema

Remiantis apmokymo duomenų imties atributais, sprendimų medis sudaro seką klausimų, kuriais būtų galima nustatyti duomenų rinkinio klases. Nors ankstesniame paveiksle iliustruojama sprendimų medžio sąvoka, pagrįsta kategoriniais kintamaisiais, tas pats principas taikomas, jei

reikšmės skaitinės. Pavyzdžiui, galima nustatyti ribinę kintamojo reikšmę nuo kurios rinkinys padalijamas į skirtingas klases.

Naudojantis sprendimų medžio algoritmu, pradedama nuo medžio šaknies ir padalijame duomenis tuo kintamuoju, kurio didžiausias informacijos gavimas (IG). Iteraciniu metodu ši procedūra kartojama, kol visos klasės lieka atskirtos. Praktiškai tai gali privesti prie labai didelio medžio sudarymo, todėl dažnai yra norima apkarpyti medį, nustatant didžiausią medžio gylį.

Modelio privalumai:

- Modelis yra lengvai interpretuojamas. Medis gali būti vizualizuojamas;
- Reikalauja mažai duomenų paruošimo;
- Modelis gali dirbti su skaitiniais arba kategoriniais kintamaisiais;
- Modelį galima įvertinti įvairiais statistiniais testais ir taip sužinoti jo patikimumą;

Modelio trūkumai:

- Sprendimų medžio algoritmas gali sukurti per didelis medžius. Tai dažniausiai priveda prie persimokymo (angl. *overfitting*) problemos;
- Sprendimų medis gali tapti nestabiliu esant mažiems pakitimams duomenyse. To rezultate gali būti gaunamas visiškai kitoks medis;
- Sprendimų medis yra mažiau veiksmingas kai klasių skaičius yra nesubalansuotas ir yra dominuojanti klasė

Norint sudarytį medį naudojant daugiausiai informacijos teikiančius kintamuosius, reikalinga tikslo funkcija, kuri apibrėžia informacijos gavimą.

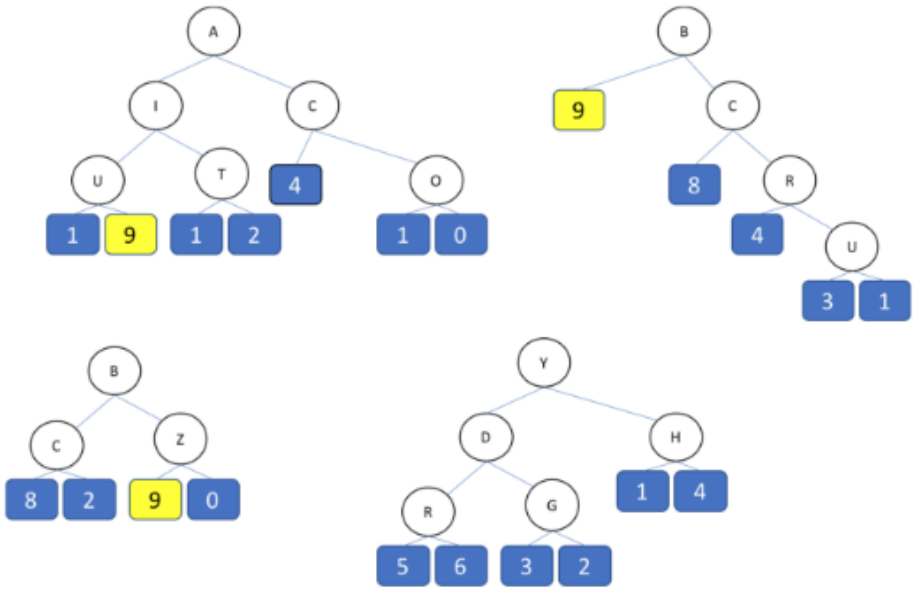
$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j) \qquad (2.3.1)$$

Čia, *f* yra kintamasis, kuriuo siekiama atlikti padalinimą, D_p ir D_j yra duomenų rinkinys ir j-tasis mazgas, *I* yra nešvarumo metrika, N_p yra bendras imties kiekis mazgo lygyje ir N_j yra mėginių kiekis *j* mazgo lygyje. Taigi informacijos gavimas yra skirtumas tarp nešvarumo viršutiniame mazge ir nešvarumo apatiniuose mazguose – kuo žemesnis apatinių mazgų nešvarumas tuo didesnis informacijos gavimas [56].

Atsitiktiniai miškai

Atsitiktinių miškų metodas yra išvestas iš sprendimų medžio metodo. Atsitiktinių miškų modelis sudarytas iš sprendimų medžių rinkinių. Kai turimas tik vienas sprendimų medis, modelis gali būti jautrus triukšmui bei turėti tam tikrą šališkumą. Tačiau, turint didelį skaičių apmokytų sprendimo medžių, galima sumažinti modelio šališkumą.

Sprendimų miško sudarymas grįstas atsitiktiniu kintamųjų ir imties eilučių parinkimu. Tad kiekvienas sprendimų medis atsitiktiniuose miškuose būna apmokomas ant mažos dalies duomenų. Jeigu didelė dalis medžių siūlo tokį patį sprendimą, tada tai nulemia viso modelio galutinį spėjimą. Atsitiktinių miškų diagramos pavyzdys pateiktas 3 paveikslėlyje.

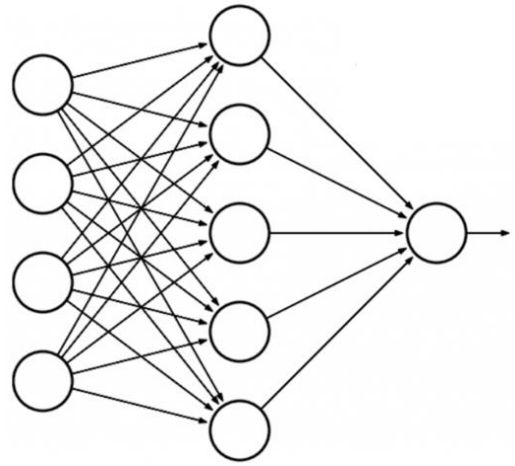


3 pav. Atsitiktinių miškų pavyzdys

Paveikslėlyje pavaizduota, kad bendras modelis siūlo 9, nepaisant jog ketvirtasis medis nesiūlo sprendinio 9. [57]

Neuroniniai tinklai

Panašiai kaip ir biologinėse neuroninėse struktūrose, neuroninių tinklų modeliuose neuronas yra laikomas centriniu apdorojimo vienetu, kuris atlieka matematines operacijas siekdamas sugeneruoti rezultatą iš gautų dedamųjų. Tokio metodo funkcijos rezultatas yra dedamųjų dalių svorinė suma kartu su šališkumo parametru (angl. *bias*). Neuroninio tinklo struktūra pateikta 4 paveikslėlyje.



4 pav. Neuroninio tinklo schema

Viso neuroninio tinklo funkcija yra visų neuronų išeigos suma. Todėl neuroninio tinklo modelis yra matematinių funkcijų aproksimacijų rinkinys. Pagrindiniai terminai, naudojami sudarant neuroninių tinklų modelius [58]:

- įvedimo sluoksnis - Įvedimo sluoksnio neuronai yra atsakingi už duomenų įvedimą į tinklą. Kartais jie taip pat vykdo tam tikrą pradinį įvedimo duomenų apdorojimą. Pavyzdžiui vykdant rašmenų atpažinimą gali tekti suvienodinti visų simbolių aukščius, nes skirtingų žmonių rašysenos yra skirtingos.

- Paslėptasis sluoksnis - Paslėptųjų sluoksnių neuronai remdamiesi įvedimo reikšmėmis ir apmokymo metu nustatytais jungčių svoriais vykdo su uždavinio sprendimu susijusios skaičiavimus.

- Išvedimo sluoksnis - Išvedimo sluoksnio neuronų veikla priklauso nuo paslėptųjų sluoksnių neutronų išvedimo reikšmių ir su jais susijusių ryšių svorių.

- Svoriai – Svoriai jungia kiekvieną sluoksnį.

- Šališkumo parametras – Tai papildomas neuronas, pridedamas kiekviename sluoksnyje.

Logistinė regresija

Sudarant logistinės regresijos modelį siekiama prognozuoti tikimybę, kad klasifikuojamas kintamasis priklausys tam tikrai klasei, modeliuojant tikimybę kaip kintamų dydžių $X_{1,2},\ldots,X_p$ funkciją. Logistinė regresija yra linijinis modelis skirtas klasifikavimui. Regresijos modelio lygtis [59]:

$$logit\ \pi(x_1, x_2, \ldots x_2) = \beta_0 + \beta_1 x_2 + \cdots + \beta_p x_p \qquad (2.3.2)$$

2.4. Klasifikavimo vertinimas

Atlikus klasterizavimą ir sudarius klasifikavimo modelius svarbu įvertinti kaip tiksliai klasifikatorius gali numatyti ar įrašas bus įtraukiantis. Skirtingos našumo metrikos naudojamos vertinant skirtingus mašininio mokymosi algoritmus. Toliau bus sutelkiamas dėmesys į tas metrikas, kurios naudojamos klasifikavimo problemoms.

Sumaišymo matrica yra viena iš intuityviausių ir paprasčiausių metrikų, naudojamų modelio teisingumui ir tikslumui nustatyti. Ši metrika naudojama klasifikavimo problemai, kai išvestis gali būti dviejų ar daugiau klasių tipai. Sumaišymo matrica savaime nėra klasifikavimo modelio metrika, tačiau beveik visos tokių modelių metrikos yra pagrįstos sumaišymo matrica ir jos reikšmėmis. Ši matrica parodo duomenų kiekį, kuriam buvo teisingai prognozuotos teigiama ir neigiamos klasės bei klaidingai prognozuotos teigiama ir neigiama klasės.

6 lentelė. Sumaišymo matrica

		Tikros reikšmės	
		Teigiamos	Neigiamos
Prognozuotos reikšmės	Teigiamos	Teisinga teigiama (TP)	Neteisinga teigiama (FP)
	Neigiamos	Neteisinga neigiama (FP)	Teisinga neigiama (TN)

Iš tokio šios matricos išvedamos šios reikšmės:

- Tikslumas (angl. *Accuracy*) - klasifikavimo tikslumas yra teisingų prognozių, atliktų modelio, skaičius, atsižvelgiant į visų rūšių prognozes. Tikslumas yra gera metrika, kai tikslinės klasės duomenys yra beveik subalansuoti.

$$acc = \frac{tp+tn}{tp+fp+tn+fn} \qquad (2.4.1)$$

- Preciziškumas (angl. *Precision*) – matuoja teisingai prognozuotų teigiamos klasės reikšmių santykį su teigiamos klasės reikšmų skaičiaus suma. Pavyzdžiui, preciziškumas parodo, kokia dalis pacientų, kuriems diagnozuota vėžys, iš tiesų turėjo vėžį.

$$p = \frac{tp}{tp+fp} \qquad (2.4.2)$$

- Atkūrimas (angl. *Recall*) – atsižvelgia į prognozuotų teigiamos klasės reikšmių santykį su teisingai modelio prognozuotomis teigiamomis ir neigiamomis reikšmėmis.

Pavyzdžiui, ši metrika parodo, kokia dalis pacientų, kuriems iš tikrųjų buvo vėžys, buvo algoritmo prognozuotas vėžys.

$$r = \frac{tp}{tp+fn} \qquad (2.4.3)$$

- Specifiškumas (angl. *Specificity*) –Specifiškumas yra metrika, kuri parodo, kokia dalis pacientų, kurie neturėjo vėžio, modelio buvo prognozuojami kaip neturintys vėžio.

$$SPC = \frac{tn}{fp+tn} \qquad (2.4.5)$$

- F1 įvertis (angl.*F1 Score)* – vertina atkūrimo ir preciziškumo metrikų vidurkį. Jo reikšmės yra tarp 0 ir 1.

$$FS = \frac{2*p*r}{p+r} \qquad (2.4.6.)$$

- AUC (angl. *area under the curve)* - AUC nustato visą dvimatę plotą po visa ROC kreive.

$$AUC = \frac{S_p - \frac{(n_p+1)}{2}}{n_p * n_n} \qquad (2.4.7)$$

2.5. Programinė įranga

Python

„Python“ yra interpretuojama, interaktyvi, objektinė programavimo kalba. Ji apima modulius, išimtis, dinaminį įvedimą, labai aukšto lygio dinaminius duomenų tipus ir klases. „Python jungiasi su daugeliu sistemų bibliotekų, taip pat su įvairiomis langų sistemomis ir yra išplečiama „C“ arba „C++“ kalbomis. Ši kalba taip pat gali būti naudojama kaip pratęsimo kalba programoms, kurioms reikia programuojamos sąsajos. Galiausiai „Python" yra lankstus: jis veikia daugelyje „Unix" variantų, „Mac" ir „Windows".

„Python" yra aukšto lygio bendrosios paskirties programavimo kalba, kuri gali būti taikoma daugybei skirtingų problemų klasių. Kalba pateikiama su didele standartine biblioteka, apimančia tokias sritis kaip styginių apdorojimas (įprastinės išraiškos, failų skirtumų skaičiavimas). programinės įrangos inžinerija (vieneto testavimas, registravimas, profiliavimas, analizuojant „Python“ kodą) ir operacinės sistemos sąsajos (sistemos skambučiai, failų sistemos). Taip pat yra įvairių trečiųjų šalių plėtinių, skirtų „Python“ programavimo kalbai [61].

Scikit Learn

„Scikit-learn“ (anksčiau žinoma kaip „scikits.learn“) yra nemokama programinės įrangos mašininio mokymo biblioteka, skirta „Python“ programavimo kalbai. Joje yra įvairūs klasifikavimo, regresijos ir klasterizavimo algoritmai, įskaitant pagalbines vektorines mašinas, atsitiktinius miškus, gradientų didinimo priemones, k-priemones. Šios bibliotekos sukurtos taip, kad būtų suderintos su populiariais "Python" paketais, tokiais kaip "NumPy".

„Scikit-learn“ programinės įrangos paketo galimas pritaikymas:

- Klasifikacija
- Regresija
- Klasterizavimas
- Dimensijų mažinimas
- Modelių įvertinimas
- Duomenų paruošimas

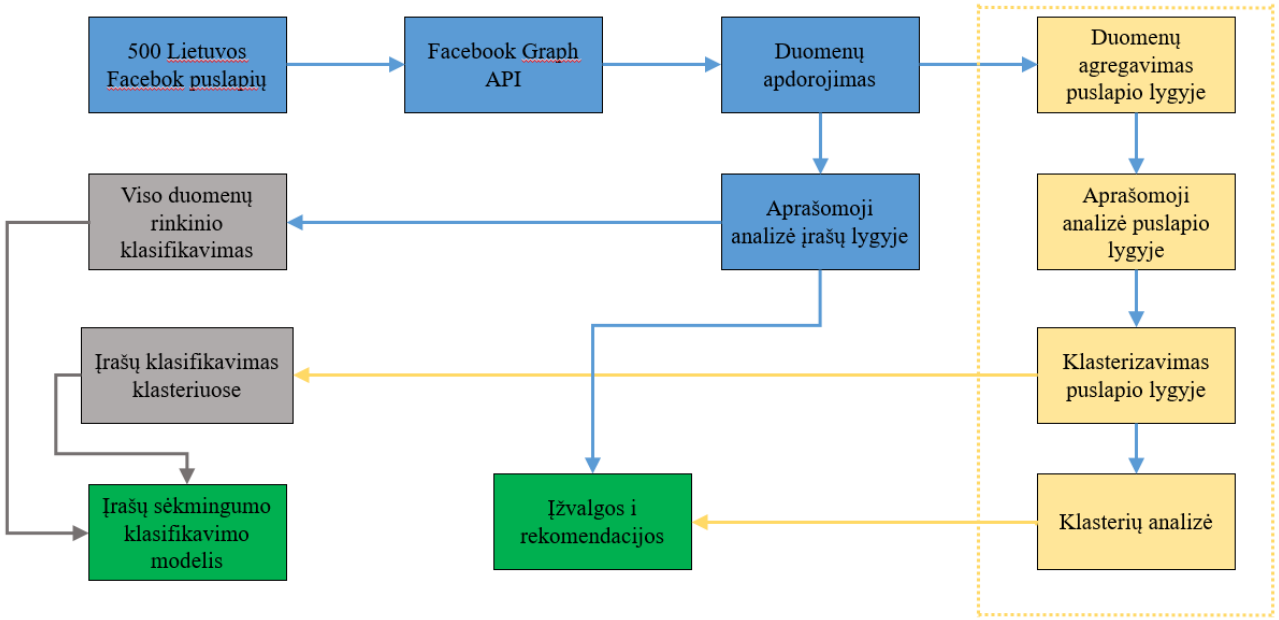
"Scikit-learn" daugiausia parašyta "Python" programavimo kalbai, tačiau taip pat palaikomos vektorinės mašinos "Cython" aplinkoje naudojant „LIBSVM“ [62].

Tableau

„Tableau“ yra galingas verslo žvalgybos ir duomenų vizualizavimo įrankis, turintis intuityvią vartotojo sąsają. Tai labai naudinga, kai atliekami duomenų pjūviai. Tableau dažnai naudojama duomenų vizualizavimo programinė įranga, naudojama duomenų mokslui ir verslo analizei. "Tableau" galima sukurti platų skirtingų vizualizacijų spektrą ir interaktyviai pateiktų duomenis ir įžvalgas. Naudojantis „Tableau" galima prieiti prie beveik bet kurio duomenų šaltinio, įskaitant "Microsoft Excel" ar žiniatinklio duomenis [63].

2.6. Tyrimo metodika

Rasti tyrimo metodai leis sudaryti klasifikavimo modelį, galintį prognozuoti įrašo sėkmingumą vartotojo įsitraukimo atžvilgiu. 5 paveikslėlyje pateikta tyrimo atlikimo schema.



5 pav. Tiriamosios dalies schema

Tiriamąją dalį galima išskaidyti į keturias pagrindines dalis: duomenų išgavimas, klasterizavimas, klasifikavimas ir įžvalgų bei rekomendacijų sudarymas. Pirmiausia bus surenkami 500 populiariausių Lietuvos puslapių „Facebook" socialiniame tinkle (žr. 1 priedą) naudojant „Facebook Graph API". Surinkti įrašų bei jų parametrų duomenys bus apdoroti į struktūruotą duomenų rinkinį. Gautas duomenų rinkinys bus išanalizuotas siekiant išgauti socialinės žiniasklaidos sprendimų tobulinimui naudingas įžvalgas bei geriau suprasi įrašų interakcijos su klientu aspektus. Siekiant atlikti sėkmingos interakcijos su įrašu prognozavimą įrašai bus klasterizuojami pagal juos kuriančių puslapių vykdomą socialinės žiniasklaidos rinkodarą ir kiekvienam klasteriui bus apmokomas klasifikavimo modelis. Šiam tikslui duomenys bus agreguojami iki puslapio lygio išvedant puslapio rinkodarą apibendrinančius rodiklius. Galiausiai bus apmokomi skirtingi modeliai siekiant klasifikuoti įrašų sekmingumą kliento įsitraukimo atžvilgiu.

3. Tyrimo rezultatai ir jų aptarimas

3.1. Duomenų išgavimas ir apdorojimas

Išgaunant duomenis naudojant „Facebook Graph API" buvo pasitelkiami papildomi „Python" paketai. Jie skirti palengvinti duomenų apdorojimą bei išgavimą. Naudojami paketai ir jų paskirtis atliktame darbe pateikti lentelėje.

7 lentelė. Darbe naudojami „Python" paketai

Paketas	**Paskirtis**
requests	Kreipimasis į API
json	JSON apdorojimas
os	Operacinės sistemos valdymas
pandas	Duomenų struktūravimas ir analizė
numpy	Skaičiavimų operacijos
datetime	Manipuliavimas datų reikšmėmis
time	Manipuliavimas laiko duomenimis
re	Manipuliavimas tekstiniais kintamaisiais
calendar	Manipuliavimas datų reikšmėmis

Toliau aprašoma kreipimosi į API funkcija. „Facebook Graph API" suteikia galimybę surinkti vieno puslapio įrašus vienu kreipimusi. Tačiau išgaunami metų laikotarpio įrašai, todėl vienu kreipimųsi į išgaunamų įrašų kiekis buvo per didelės apimties. Dėl šios priežasties surenkant kiekvieno puslapio duomenis buvo 12 kartų kreipiamąsi į API. Taip pat pagal numatymą „Facebook Graph API" gražina tik 25 paskutinius įrašus. Dėl šios priežasties kreipimosi funkcijoje nurodomas limito parametras, kurį naudojant grąžinama tiek įrašų kiek nurodyta limite. Pagrindinis parametras, suteikiantis priėjimą prie API yra žetonas (angl. *token*), kuris gali būti gaunamas naudojant „Facebook" vartotojo profilį. Šiame darbe panaudotas žetonas gautas naudojant darbo autoriaus asmeninę „Facebook" paskyrą. Taigi aprašyta funkcija reikalaus šių parametrų:

- Žetono ID
- Puslapio ID
- Įrašų limito
- Pradžios datos
- Pabaigos datos

Šie parametrai naudojami kreipimosi į API URL. Šiame URL nurodyti šie duomenų laukai, kuriuos siekiama gauti iš API:

- „name“
- „category“
- „fan_count“
- „posts

Kreipimosi rezultatas grąžinamas JSON formatu. Šiuo formatu grąžinami duomenų objektai, sudaryti iš atributų ir reikšmių porų. Šiam JSON skaityti naudojama „Python“ paketo „json“ funkcija „json.loads“, kuri paverčia gautą objektą į python žodyno (angl. *dictionary*) klasę ir leidžia ją patogiai apdoroti. Toliau kiekvieno kintamojo išgavimui naudojama skirtinga funkcija. 7 Lentelėje aprašomas kiekvienas gautas kintamasis. Gauti kintamieji buvo rašomi į .txt tipo failą.

8 lentelė. Duomenų rinkinio aprašymas

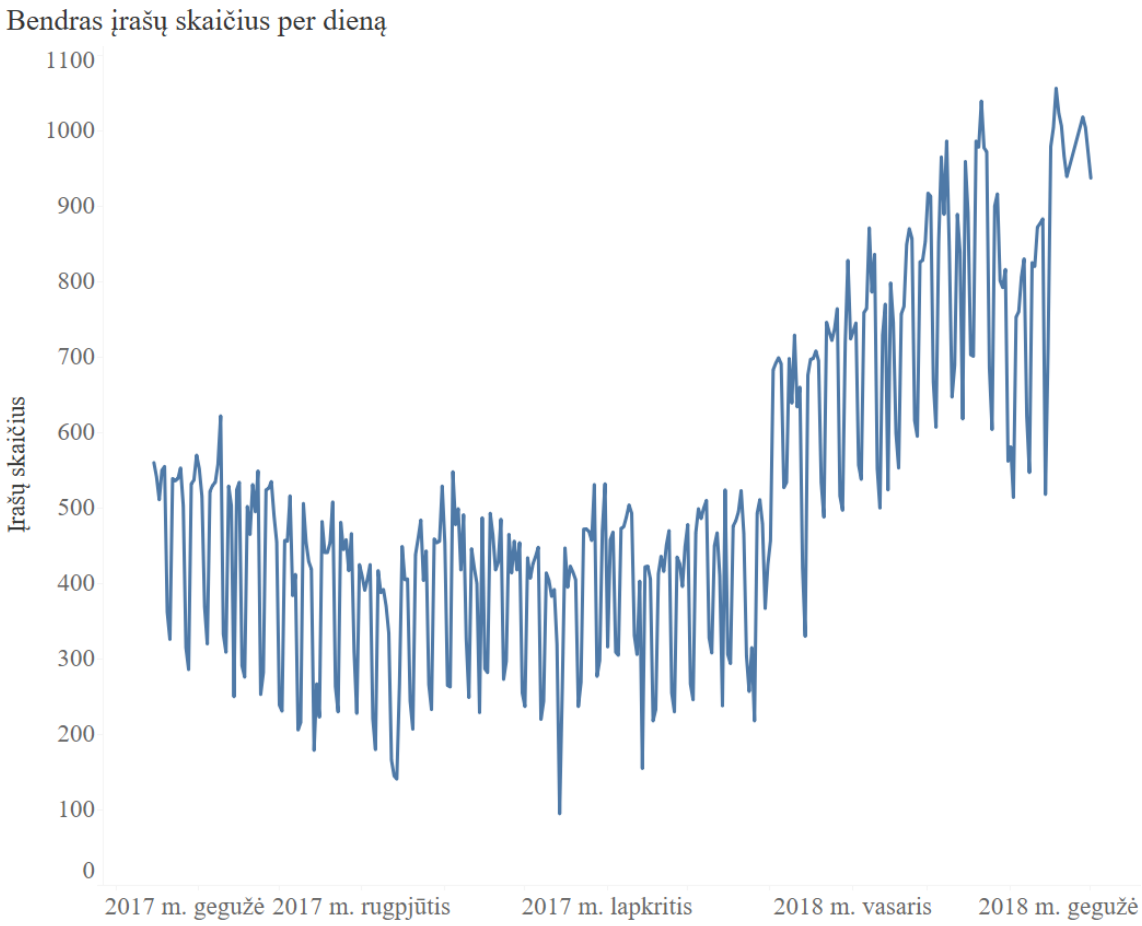
Reikšmės pavadinimas	**Tipas**	**Paaiškinimas**
page_id	Skaičius	Facebook priskiriamas unikalus ID
page_name	Simbolinis	Puslapio pavadinimas
page_category	Simbolinis	Puslapio kategorija, kurią parinko puslapio savininkas iš duotų galimų
fan_count	Skaičius	Sekėjų skaičius
post_id	Simbolinis	Unikalus įrašo ID
message	Simbolinis	Įrašo žinutės tekstas
created_date	Data	Įrašo sukūrimo data
created_time	Laikas	Įrašo sukūrimo laikas
like_count	Skaičius	Įrašo surinktų patinka kiekis
reactions_count	Skaičius	Įrašo surinkų reackijų (angl. *reactions*) skaičius. Po šiuo kintamuoju patenka ir „patinka“ skaičius
comment_count	Skaičius	Įrašo komentarų skaičius
shares_count	Skaičius	Įrašo pasidalinimų skaičius
link	Simbolinis	Prie įrašo prisegto objekto nuordoda
weekday	Simbolinis	Įrašo paskelbimo savaitės diena
time_of_day	Simbolinis	Dienos metas – rytas, diena, vakaras, naktis
has_emoji	Loginis	Ar įrašo tekste yra panaudotas jaustukas (angl *emoji*)
has_hashtag	Loginis	Įrašo tekste naudojamos grotelės

Reikšmės pavadinimas	Tipas	Paaiškinimas
has_exclamation	Loginis	Įrašo tekste naudojamas šauktukas
has_question_mark	Loginis	Įrašo tekste naudojamas klaustukas
has_share_contest	Loginis	Įraše skelbiamas konkursas dovanai laimėti. Šis faktas identifikuojamas pagal raktinius žodžius, kuriuo naudojami šiuos konkursus skelbiantys įrašai
message_symbol_count	Skaičius	Įrašo simbolių skaičius žinutėje
message_word_count	Skaičius	Įrašo žodžių skaičius žinutėje
message_contains_link	Skaičius	Įrašo tekste pateikta nuoroda
has_link_attached	Loginis	Įrašo objektas yra nuoroda į kitą puslapį
has_video_attached	Loginis	Įrašo objektas - nuoroda į „Facebook“ arba “Youtube“ „video“
has_photos_attached	Loginis	Įrašo objektas yra nuoroda į nuotrauka
hours_since_last_post	Loginis	Valandos nuo paskutinio įrašo
did_post_last_24h	Loginis	Ar buvo darytas kitas įrašas per pastarąsias 24 valandas
did_post_last_72h	Loginis	Ar buvo darytas kitas įrašas per pastarąsias 72 valandas
did_post_last_7d	Loginis	Ar buvo darytas kitas įrašas per pastarąsias 7 dienas

Apdorojant duomenis bei ieškant reikšmingų klasifikavimo kintamųjų buvo identifikuojami įrašai, kuriais „Facebook” socialiniame tinkle buvo skelbiami „pasidalinimo” konkursai („has_share_contest” kintamasis). Įrašo pasidalinimai buvo identifikuojami pagal raktinių žodžių derinius, kuriuos dažniausiai turi tokio tipo įrašai „Facebook” socialiniame tinkle. Raktiniai žodžių deriniai parinkti darbo autoriaus empiriniu tyrimu analizuojant tokio pobūdžio įrašus.

Iš viso gauti 25 kintamieji. Turint “patinka” paspaudimų, pasidalinimų ir komentarų skaičių buvo išskaičiuota interakcijos sėkmingumą apibrėžiantis – „įsitraukimas“.

Facebook Graph API suteikiamas priėjimo žetonas galioja tik valandą, todėl nepavyko ištraukti visų duomenų vienu ciklo paleidimu – duomenų išgavimo eigoje žetono galiojimą reikėdavo pratęsti rankiniu būdu. Šie duomenys buvo išgauti dalimis per tris kartus, todėl duomenų gavimas užtruko 3-4 valandas. Išgavimo metu nebuvo pasiektas API kreipimosi limitas. Gautų duomenų dieninis kiekis pateiktas 6 paveikslėlyje.

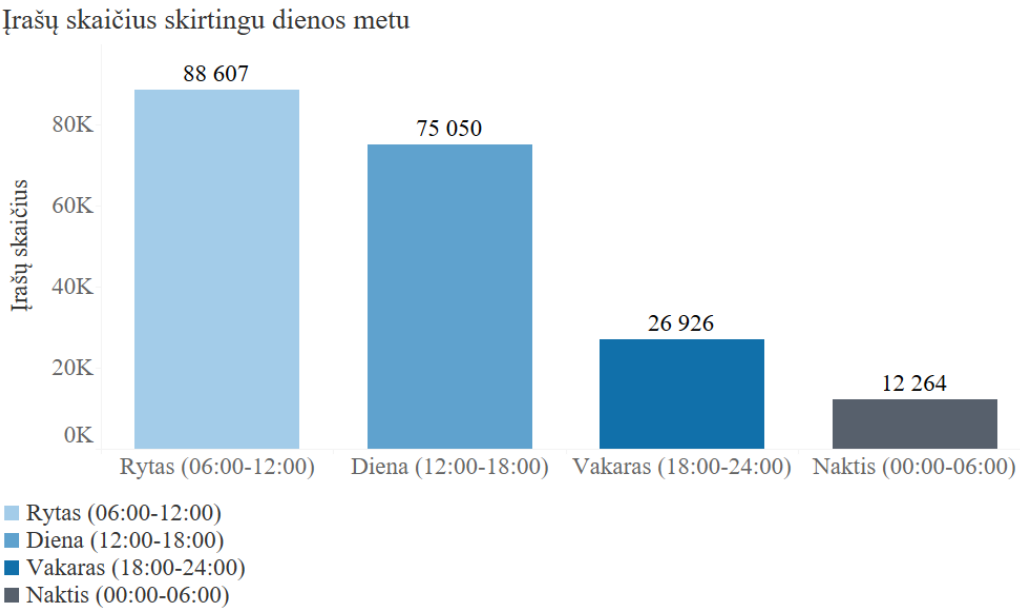


6 pav. Duomenų kiekio kitimas laike

Vizualiai matoma, kad duomenys yra švarūs. Nėra netikėtų didelių iššokimų arba tuščių dienų. Duomenys sudaryti iš 202 847 eilučių. Iš pasirinkto 500 populiariausių puslapių 35 neturėjo įrašų analizuojamu laikotarpiu, todėl pavyko surinkti 465 „Facebook" socialinio tinkle puslapių duomenis.

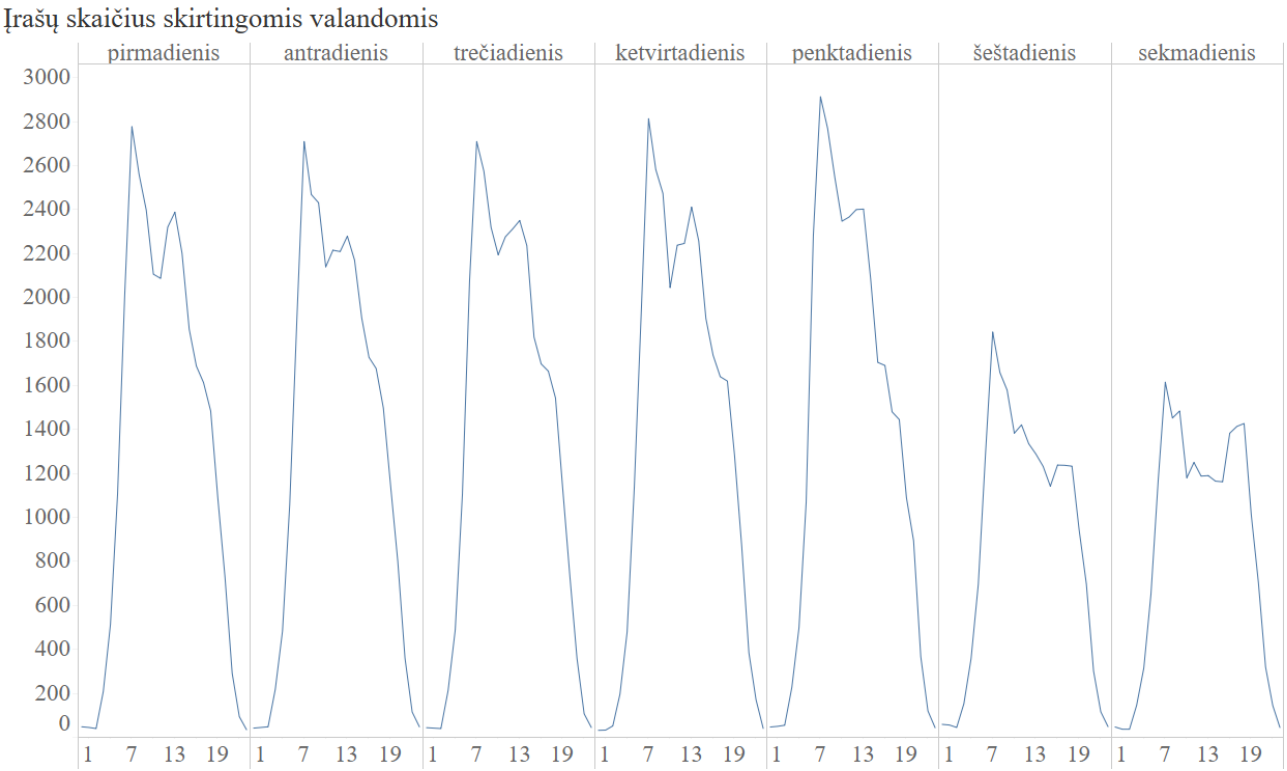
3.1. Duomenų aprašomoji analizė

Atliekama įrašų analizė siekiant atpažinti galiojančius dėsningumus Lietuvos „Facebook" puslapių socialinės žiniasklaidos rinkodaroje. Pirmiausia siekiama įvertinti įrašų kiekį ir laiką dienoje. Skritulinėje diagramoje pateiktas įrašų skaičius skirtingu dienos metu, kas 6 valandas.



7 pav. Įrašų skaičius skirtingu dienos metu

Matoma, kad Lietuvos socialinė žiniasklaida aktyviausia ryte bei stipriai aktyvi dienos metu. Siekiant išanalizuoti šį dėsnį detaliau, įrašų skaičius buvo skaidomas į valandų lygį, pateikiant įrašų skaičių savaitės dienomis (žr pav).

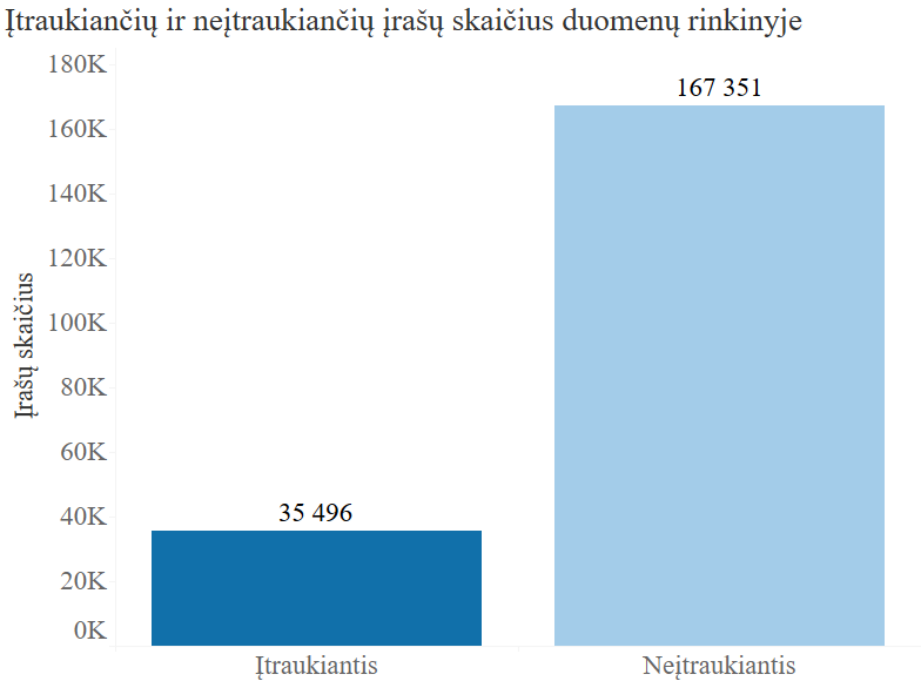


8 pav. Įrašų skaičiaus kitimas valandomis ir savaitės dienomis

Grafike matoma, kad 7 valandą ryto skelbiama daugiausiai įrašų. Dažnai įmonių skelbiamas rinkodarinis turinys būna paruoštas iš anksto ir jo paleidimas būna nustatytas naudojantis „Facebook”

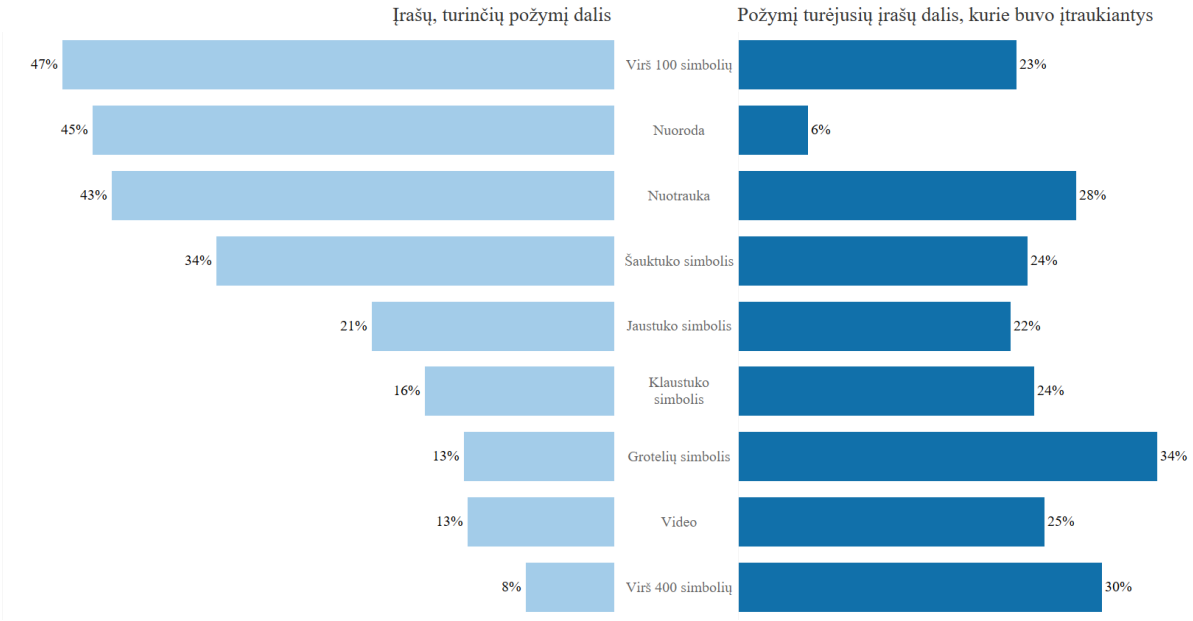
funkcijomis. Šiuo atveju galima daryti išvadą, kad rinkodaros specialistai siekia, kad vartotojas būtų pasiektas vos darbo dienai prasidėjus. Įdomu ir tai, kad ši tendencija galioja ir savaitgaliais. Tikėtina, kad „Facebook" vartotojai yra mažiau aktyvūs tokiu metu ne darbo dienomis, tačiau savaitgalio rytais skelbiamų įrašų kiekis yra mažesnis nei rytais darbo dienomis. Savaitgaliai taip pat skiriasi tuo, kad jų metu socialinė žiniasklaida suaktyvėja vakare tarp 18:00 ir 20:00. Galiausiai matoma, kad nakties metu Lietuvos „Facebook" aktyvumas nuslūgsta ir beveik nebevyksta.

Išanalizavus įrašų kiekio ir laiko specifiką buvo pereita prie pagrindinio klasifikuojamo rodiklio – įrašo įtraukimo analizės. Įtraukiančių ir neįtraukiančių įrašu skaičius pateiktas 9 paveikslėlyje.



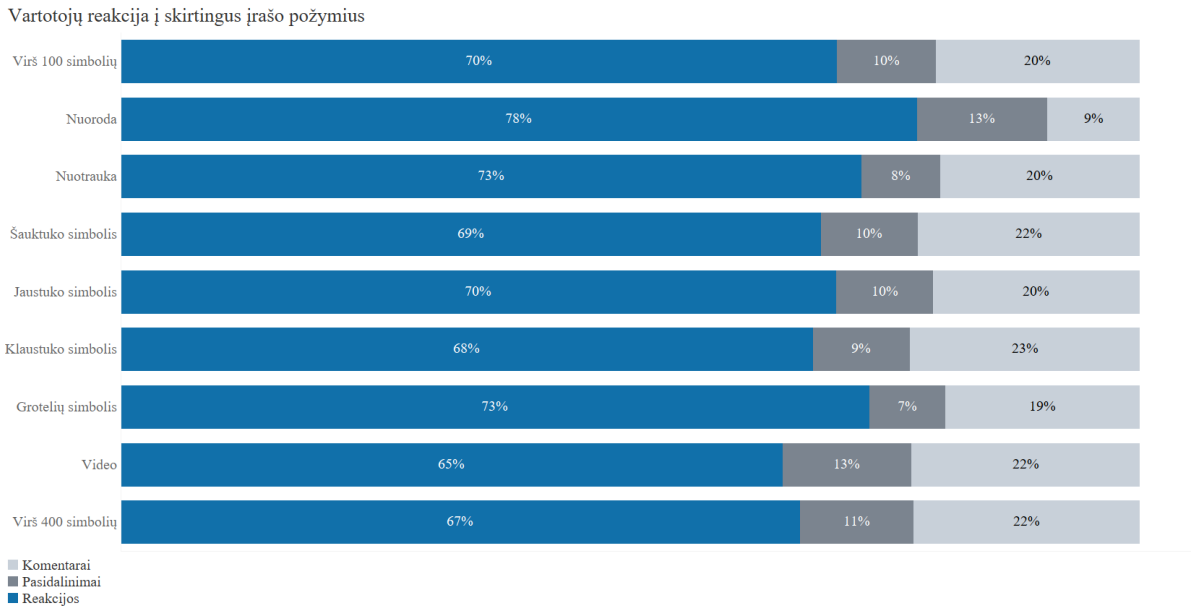
9 pav. Įtraukiančių ir neįtraukiančių įrašu skaičius

Matoma, kad iš visų įrašų tik 35 496 (17%) pasiekia sėkmingo įsitraukimo lygį. Toliau buvo siekiama detaliai išanalizuoti šį rodiklį lemiančius požymius. Analizuojamas atskiros įrašų detalės, kurios buvo išgautos duomenų apdorojimo procese (žr. 10 pav.).



10 pav. įrašų įsitraukimo ryšys su požymiais

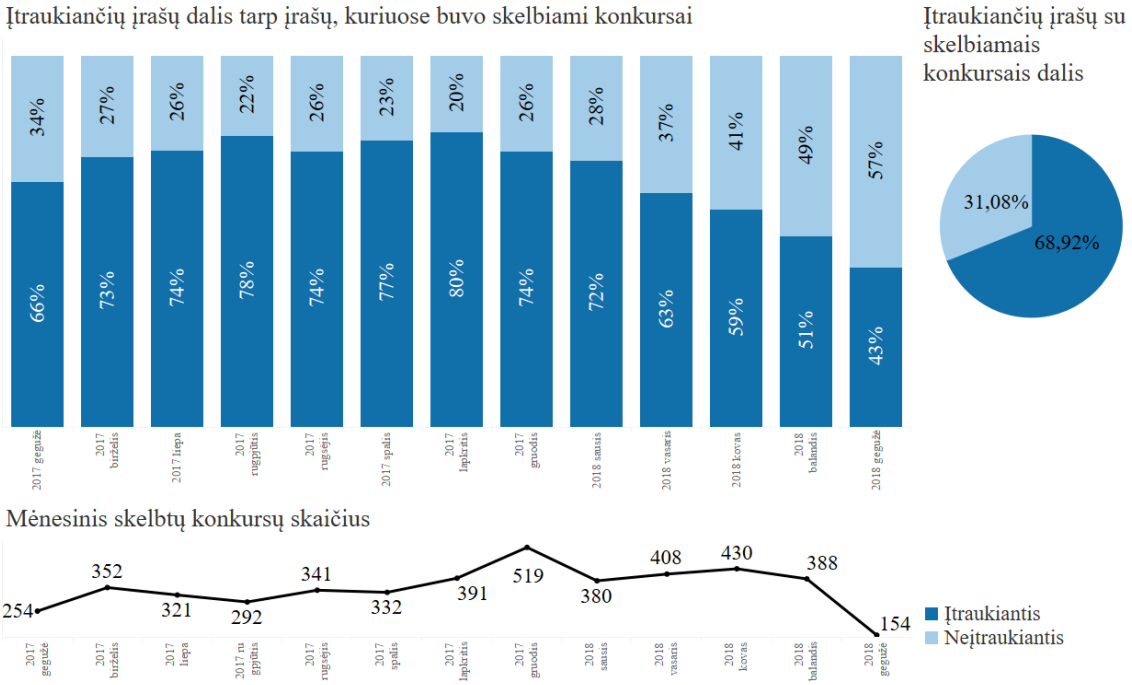
Apačioje matoma, kad daugiau negu 100 simbolių turi beveik pusė analizuojamų įrašų. Galima daryti išvadą, kad skelbiami įrašai yra aprašomi bent sakiniu. Trečdalis įrašų turi bent vieną šauktuko simbolį. Taip pat matoma, kad nuotraukos ir nuorodos yra pridedamos į pusę įrašų. Vaizdo įrašai yra rečiau pasitaikanti turinio forma. Matoma, kad įraše esančios nuorodos požymis daro mažiausią įtaką įsitraukimui. Taip pat buvo analizuojama vartotojų reakcija į įrašus turinčius skirtingus požymius (žr. 11 pav.)



11 pav. Vartotojų reakcija į skirtingus įrašų aspektus

Vartotojo reakcija į skirtingus įrašų požymius dažniausiai pasižymi reakcijos paspaudimu („patinka” ir jaustukai). Tačiau nuorodų ir vaizdo įrašo požymiai išsiskiria dažnesniu dalinimusi nei kitus požymius turintys įrašai.

Atskira analizė atlikta „pasidalinimų“ konkursų įrašų vertinimui. 12 paveikslėlyje pateikti šių konkursų grafikai.



12 pav. Pasidalinimų konkursų analizė

Diagrama apačioje rodo, kad tai itin stipriai įsitraukimo metriką įtakojantis rodiklis. Tačiau šių įrašų nėra daug. Tikėtina, kad tai bus stiprus rodiklis klasifikuojant.

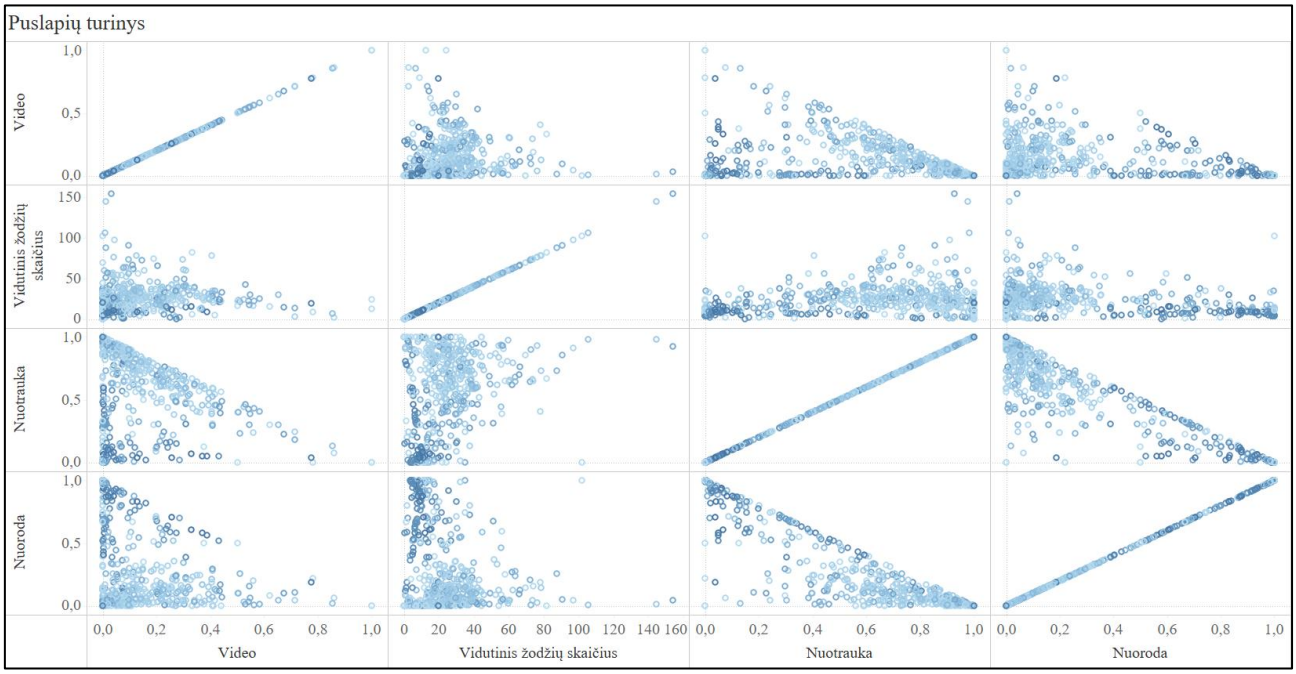
3.2. Klasterizavimas

Turimi duomenys yra įrašo lygyje, todėl, norint turėti duomenis puslapio lygyje, reikalingas jų agregavimas. Duomenų agregavimas atliktas naudojant „Spark“. Siekiant padidinti skaičiavimų efektyvumą turimas duomenų rinkinys buvo particionuojamas mėnesio lygyje. Toks particionavimas leidžia ženkliai padidinti skaičiavimo greitį. Particionuotų duomenų agregavimas aprašytas „SQL“ funkcijomis, kurios buvo įvykdytos „Spark“. Duomenis suagregavus iki puslapio lygio, gautas duomenų failas su metrikomis, apibrėžiančiomis puslapių turinio tipą, turinio kiekį ir dažnumą. Gauto duomenų failo eilučių skaičius lygus puslapio skaičiui – 465. Metrikos pagal kurias bus atliekamas klasterizavimas pateiktos 8 lentelėje.

9 lentelė. Klasterizavimo metrikos

Reikšmės pavadinimas	Tipas	Paaiškinimas
percent_of_posts_with_video	Skaičius	Įrašų su vaizdo įrašais sudaroma dalis
percent_of_posts_with_photo	Skaičius	Įrašų su nuotraukomis sudaroma dalis
percent_of_posts_with_link	Skaičius	Įrašų su nuorodomis sudaroma dalis
percent_of_posts_with_share_contest	Skaičius	Įrašų su konkursais sudaroma dalis
avg_message_word_count	Skaičius	Vidutinis žodžių skaičius įraše
avg_hours_since_last_post	Skaičius	Vidutinis valandų kiekis praėjęs nuo paskutinio įrašo
post_count	Skaičius	Įrašų skaičius

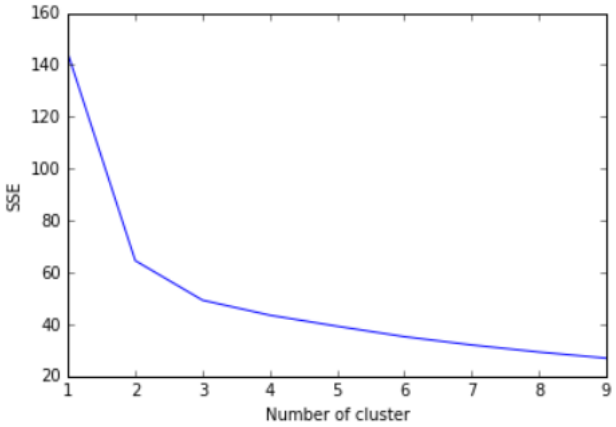
Dalis klasterizavimo metrikų pateikta taškinėse diagramose apačioje. Pirmiausia matoma neigiama tarpusavio koreliacija tarp skaitmeninio turinio dalies įrašuose - nuotraukos, vaizdo įrašai arba nuorodos. Tas parodo, kad kartu šie objektai nėra dažnai skelbiami – pasirenkamas tik vienas iš jų. Matoma, kad dažnai vaizdo įrašo formato įrašus skelbiantys puslapiai vengia juos aprašyti (mažas vidutinis žodžių skaičius). Tačiau nuotraukas dažnai skelbiantys puslapiai dažnai pasirenka aprašyti savo įrašus. Taip pat matoma, kad nuotraukas dažnai skelbiančių puslapių skaičius yra didelis. Daugiau negu 80% savo įrašuose naudojančių vaizdo įrašus skelbiančių puslapių yra vos keli.



13 pav. Puslapių skelbiamo turinio analizė

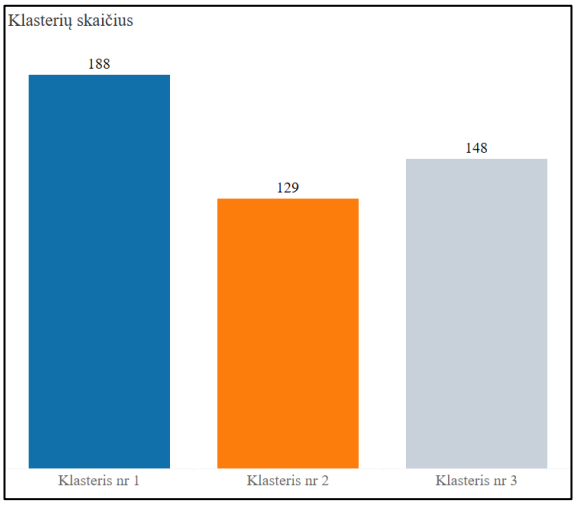
Klasterizavimas atliktas K- vidurkių metodu. Atliekant klasterizavimą buvo naudojamas „scikit learn“ paketas. Klasterizavimo modelis buvo sudaromas 10 kartų, keičiant numatytą klasterių skaičių nuo 1 iki 10. Klasterizavimo rezultatas vertintas pagal SSE rodiklį (žr. pav). Matoma, kad pagal SSE

vizuali alkūnė yra ties 2 ir 3 klasteriais. Kadangi 2 klasteriai būtų per grubus klasterizavimas iš loginės pusės, pasirinkta taikyti 3 klasterių modelį.



14 pav. Klasterių skaičius

Klasterizavimo metodas padalino analizuojamus puslapius pakankamai tolygiai (žr 15 pav). Nei vienas klasteris neturi ženkliai daugiau arba ženkliai mažiau puslapių nei kiti.



15. pav. Klasterių dydžiai

Atlikus klasterizavimą vėl pasitelktos dalies metrikų taškinės diagramos. Matoma, kad puslapių skaičiai išsidėstę pakankamai šalia.

16 pav. Klasterių ryšys su turinio pobūdžiu

Klasterių aprašymas:

- Klasteris nr 1 – Šiame klasteryje esantys puslapiai skelbia įvairaus tipo įrašus, tačiau rečiau renkasi nuorodas.
- Klasteris nr 2 – Šie puslapiai skelbia įrašus su nuotraukomis ir dažnai jas aprašo.
- Klasteris nr 3 – Šiame klasteryje dominuoja nuorodas skelbiantys puslapiai, tačiau šie puslapiai taip pat nevengia skelbti vaizdo įrašo formato įrašų. Šie puslapiai labai retai arba beveik niekada neskelbia nuotraukų.

3.3. Klasifikavimas

Pirmiausia atliekant klasifikavimą buvo susidurta su klasių disbalansu. Klasių disbalansas buvo sprendžiamas mažinant didesnės klasės narių skaičių (angl. *undersampling*).

Buvo atliekamas parametrų derinimas sprendimų medžių bei atsitiktinių miškų metodams. Sprendimų medis buvo sudaromas iteraciniu principu keičiant medžių gylį maksimalų gylį (5, 10, 15, 20, 25, 30), bei minimalų padalijimų skaičių (50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950, 1000). Atsitiktiniai miškai buvo sudaromi keičiant maksimalų gylį (5, 10, 15, 20, 25, 30) ir medžių skaičių (100, 275, 450, 625, 800). Modeliams įvertinti buvo naudojama kryžminė patikra.

Visas duomenų rinkinys

10 lentelė. Viso duomenų rinkinio klasių balansas

Klasifikuojamas kintamasis	Originalus klasių skaičius	Klasių skaičius po subalansavimo
Įtraukiantis	35387	35387
Neįtraukiantis	167300	35387

Geriausio medžio parametrai:

- Maksimalus gylis – 15
- Minimalus padalijimų skaičius - 250

Geriausio miško parametrai:

- Maksimalus gylis – 10
- Medžių skaičius - 625

11 lentelė. Viso duomenų klasifikavimo rezultatai

Įvertis	Sprendimų medis	Atsitiktinis miškas	Neuroninis tinklas	Logistinė regresija
Taiklumas	0.695	0.757	0.739	0.726
AUC	0.695	0.758	0.739	0.726
TP	6029	6299	6157	6214
TN	6268	7104	6921	6637
FP	2571	1735	1918	2202
FN	2826	2556	2698	2641
Preciziškumas	0.701	0.784	0.762	0.738
Atkūrimas	0.681	0.711	0.695	0.702
F1 įvertis	0.691	0.746	0.727	0.72
Jautrumas	0.681	0.711	0.695	0.702
Specifiškumas	0.709	0.804	0.783	0.751

Pirmojo klasterio klasifikavimas

12 lentelė. Pirmo klasterio klasių balansas

Klasifikuojamas kintamasis	Originalus klasių skaičius	Klasių skaičius po subalansavimo
Įtraukiantis	13528	13528
Neįtraukiantis	29804	13528

Geriausio medžio parametrai:

- Maksimalus gylis – 10

- Minimalus padalijimų skaičius – 350

Geriausio miško parametrai:

- Maksimalus gylis – 10
- Medžių skaičius – 100

13 lentelė. Pirmo klasterio klasifikavimo rezultatai

Įvertis	Sprendimų medis	Atsitiktinis miškas	Neuroninis tinklas	Logistinė regresija
Taiklumas	0.657	0.664	0.627	0.624
AUC	0.657	0.665	0.627	0.624
TP	2182	2155	2205	2317
TN	2262	2339	2039	1903
FP	1102	1025	1325	1461
FN	1218	1245	1195	1083
Preciziškumas	0.664	0.678	0.625	0.613
Atkūrimas	0.642	0.634	0.649	0.681
F1 įvertis	0.653	0.655	0.636	0.646
Jautrumas	0.642	0.634	0.649	0.681
Specifiškumas	0.672	0.695	0.606	0.566

Antrojo klasterio klasifikavimas

14 lentelė. Antrojo klasterio klasių balansas

Klasifikuojamas kintamasis	Originalus klasių skaičius	Klasių skaičius po subalansavimo
Įtraukiantis	8267	8267
Neįtraukiantis	37871	8267

Geriausio medžio parametrai:

- Maksimalus gylis – 15
- Minimalus padalijimų skaičius – 50

Geriausio miško parametrai:

- Maksimalus gylis – 10
- Medžių skaičius – 800

15 lentelė. Antrojo klasterio klasifikavimo rezultatai

Įvertis	Sprendimų medis	Atsitiktinis miškas	Neuroninis tinklas	Logistinė regresija
Taiklumas	0.808	0.821	0.793	0.777
AUC	0.808	0.822	0.794	0.777
TP	1648	1684	1572	1607
TN	1691	1711	1707	1604
FP	332	312	316	419
FN	463	427	539	504
Preciziškumas	0.832	0.844	0.833	0.793
Atkūrimas	0.781	0.798	0.745	0.761
F1 įvertis	0.806	0.82	0.786	0.777
Jautrumas	0.781	0.798	0.745	0.761
Specifiškumas	0.836	0.846	0.844	0.793

Trečiojo klasterio klasifikavimas

16 lentelė. Trečiojo klasterio klasių balansas

Klasifikuojamas kintamasis	Originalus klasių skaičius	Klasių skaičius po subalansavimo
Įtraukiantis	13592	13592
Neįtraukiantis	99625	13592

Geriausio medžio parametrai:

- Maksimalus gylis – 15
- Minimalus padalijimų skaičius – 250

Geriausio miško parametrai:

- Maksimalus gylis – 10
- Medžių skaičius – 800

17 lentelė. Trečiojo klasterio klasifikavimo rezultatai

Įvertis	Sprendimų medis	Atsitiktinis miškas	Neuroninis tinklas	Logistinė regresija
Taiklumas	0.751	0.763	0.739	0.741
AUC	0.75	0.762	0.738	0.74
TP	2778	2840	2677	2701
TN	2325	2345	2344	2332
FP	1034	1014	1015	1027
FN	659	597	760	736
Preciziškumas	0.729	0.737	0.725	0.725
Atkūrimas	0.808	0.826	0.779	0.786
F1 įvertis	0.766	0.779	0.751	0.754
Jautrumas	0.808	0.826	0.779	0.786
Specifiškumas	0.692	0.698	0.698	0.694

Rezultatų apibendrinimas

17 lentelėje pateiktas sudarytų modelių apibendrinimas. Matoma, kad visuose duomenų rinkiniuose geriausius rezultatus pasiekė atsitiktiniai miškai.

18 lentelė. Klasifikavimo rezultatų apibendrinimas

Duomenų rinkinys	Geriausiai modelis	Taiklumas	AUC	TP	TN	FP	FN
Pilnas	Atsitiktinis miškas	0.757	0.758	6299	7104	1735	2556
1-asis klasteris	Atsitiktinis miškas	0.664	0.665	2155	2339	1025	1245
2-asis klasteris	Atsitiktinis miškas	0.821	0.822	1684	1711	312	427
3-asis klasteris	Atsitiktinis miškas	0.763	0.762	2840	2345	1014	597

Atliekant klasifikavimą atskiruose klasteriuose nebuvo pastebėtas žymus rezultatų pagerėjimas lyginant su klasifikavimu neklasterizuojant duomenų rinkinio. Tačiau trečiojo klasterio klasifikavimas išsiskiria tuo, kad šiame klasteryje buvo sėkmingai suklasifikuota žymiai daugiau teisingai teigiamų reikšmių (TP).

Atliktus tyrimą galima teigti, kad sudaryti modeliai įrašo sėkmingumą vartotojų įsitraukimo atžvilgiu gali prognozuoti 75,7% taiklumu.

Išvados

Remiantis atlikta literatūros analize, buvo identifikuota, jog įmonės pasiekia vartotojus naudodamos socialinę žiniasklaidą socialiniuose tinkluose. Nors ir ne visada lengvai atsispindinti finansinėse ataskaitose, socialinės žiniasklaidos rinkodaros kuriama nauda ir efektyvumas yra išmatuojami rodikliais, tokiais kaip matomumas, įtaka, įsitraukimas. "Facebook" buvo identifikuotas kaip populiariausias socialinis tinklas Lietuvoje. Ne visi verslo puslapių skelbiami įrašai „Facebook“ socialiniame tinkle pasiekia jų sekėjus. Verslo profilių įrašai „Facebook“ socialinio tinklo vartotoją gali pasiekti dviem būdais – mokama reklama arba organiniu pasiekiamumu. "Facebook" algoritmų analizė atskleidė, kad kliento interakcija su turiniu komentarais, pasidalinimais ar „patinka“ paspaudimais yra reikšmingas kriterijus, siekiant, kad įrašas pasiektų vartotoją organiškai. Todėl kuriamas socialinės žiniasklaidos rinkodaros turinys turi būti orientuotas į kuo didesnę vartotojų interakciją su juo.

Identifikavus vartotojų įsitraukimo į įrašą socialiniame tinkle svarbą buvo siekiama sukurti modelį, kuris leistų numatyti ar įrašas taps įtraukiančiu vartotoją. Įtraukiantis įrašas apibrėžtas kaip įrašas, kurio įsitraukimo rodiklis pasiekė didesnę nei 0.1 reikšmę. Tokio pobūdžio problema buvo apibrėžta kaip klasifikavimo uždavinys. Šiai problemai buvo identifikuoti tinkamiausi klasifikavimo metodai – sprendimų medis, atsitiktiniai miškai, neuroniniai tinklai ir logistinė regresija.

Sudarius klasifikavimo modelius buvo nustatyta, kad atsitiktinių miškų modelis yra tiksliausiai įrašo įsitraukimą prognozuojantis modelis. Sudarytas klasifikavimo modelis leidžia 75.7% tikslumu nuspėti ar skelbiamas įrašas bus įtraukiantis vartotoją. Tokio pobūdžio modelis socialinės žiniasklaidos rinkodaros vadybininkui suteiktų galimybę pasitikrinti skelbiamu įrašo patrauklumą vartotojui. Įrašui sulaukus neigiamo modelio įvertinimo, jis galėtų būti neskelbiamas arba tobulinamas.

Atlikus literatūros ir tyrimo metodų analizę bei įvertinus sudaryto modelio tikslumą, galima daryti išvadą, kad socialinės žiniasklaidos sprendimai socialiniame tinkle „Facebook“ gali būti optimizuoti didžiųjų verslo duomenų analitikos sprendimais.

Tyrimo ribotumai ir tolimesnių tyrimų kryptys

Griežtėjantys duomenų apsaugos standartai verčia „Facebook“ nuolatos peržvėlgti savo privatumo politiką. Viena iš šios tendencijos pasekmių yra tai, kad „Facebook“ tampa vis mažiau prieinamas programiškai. Ribotas prieinamumas siaurina tokio pobūdžio tyrimų galimybes.

Tyrimas galėtų būti tobulinamas pasinaudojant teksto analitikos sprendimais. Tyrimo metu nebuvo naudojami įrašų komentarų duomenys ir jų tekstas, kuris yra prieinamas naudojant „Facebook

Graph API“. Tikėtina, kad siekiant prognozuoti įrašo sėkmingumą vartotojo įsitraukimo atžvilgiu, komentarų detalesnė analizė galėtų pagerinti šį procesą. Galimai reikšmingas rodiklis galėtų būti įrašą skelbiančio puslapio įsitraukimas į to įrašo komentaras. Tikėtina, kad „Facebook“ puslapiai, atsakantys į vartotojų komentarus po jų įrašais, sulaukia daugiau įsitraukimo iš vartotojų.

Tyrimas galėtų būti tobulinamas turint pilną priėjimą prie konkretaus „Facebook“ puslapio, susiaurinant analizę iki to puslapio skelbiamų įrašų. Tokiu atveju būtų prieinamas platesnis spektras duomenų, kuris leistų atlikti gilesnę analizę bei, tikėtina, sudaryti geresnį klasifikavimo modelį.

Literatūros sąrašas

1. Calfee, J. E., & Ringold, D. J. *The 70% majority: Enduring consumer beliefs about advertising*. Journal of public policy & marketing, 228-238. (1994). [žiūrėta 2018-03-14].
2. Hajli, M. N. *A study of the impact of social media on consumers*. International Journal of Market Research, 56(3), 387-404. (2014), [žiūrėta 2018-04-04].
3. Steenburgh, T. J., Avery, J., & Dahod, N.. *HubSpot: Inbound Marketing and Web 2.0.* (2009). [žiūrėta 2018-03-12].
4. Scott, D. M, *The New Rules of Marketing & PR: How to Use Social Media, Online Video, Mobile Applications, Blogs,* News, Releases, & Viral Marketing to Reach Buyers Directly (2013). [žiūrėta 2018-03-11].
5. Luke, K. *Marketing the new-fashioned way: connect with your target market through social networking sites*. Journal of Financial Planning, 2009(November/December), 18-19. (2009). [žiūrėta 2018-03-16].
6. Mize, S. R. *Social network benefits*. (2009). [žiūrėta 2018-05-04]. Prieiga per : http://ezinearticles.com/?Social-Network-Benefits&id=464645
7. Kelly, N. *How to Measure Social Media: A Step-By-Step Guide to Developing and Assessing Social Media ROI* . (2012) [žiūrėta 2018-02-04] Prieiga per: https://www.safaribooksonline.com/library/view/how-to-measure/9780133099812/ch00.html
8. Blanchard, Oliver. *Social Media ROI: Managing and Measuring Social Media Efforts in Your Organization*, (2011) [žiūrėta 2018-02-14]
9. Laskey, H., Day, E. and Crask, M.R. (1989), *Typology of main message strategies for television commercials*, Journal of Advertising, Vol. 18 No. 1, pp. 36-41. (1989) [žiūrėta 2018-03-01]
10. Lee, D., Hosanagar, K., & Nair, H. S. *The effect of social media marketing content on consumer engagement: Evidence from facebook*. Stanford Graduate School of Business. (2014). [žiūrėta 2018-03-13]
11. Mangold, W. G., & Faulds, D. J. *Social media: The new hybrid element of the promotion mix*. Business horizons, 52(4), 357-365. (2009). [žiūrėta 2018-03-22]
12. Goldenberg, J., Libai, B., Muller, E., & Stremersch, S. *The evolving social network of marketing scholars*. Marketing Science, 29(3), 561-567. (2010). [žiūrėta 2018-03-21]
13. Ertell, K. *The key to driving retail success with social media: focus on Facebook.* (2010). [žiūrėta 2018-04-04] Prieiga per: http://www.foreseeresults.com/research-white-papers/the-key-to-driving-retail-su ccess-with-social-media.shtml.

14. Ireson, N. *Over 2 million cars to be sold on social networks this year. (2010).* [žiūrėta 2018-03-17] Prieiga per http://www.thecarconnection.com/marty-blog/1047906_over-2-million-cars-to-be-sold-on-social-networks-this-year/.
15. Peppard, J. Customer relationship management (CRM) in financial services. European Management Journal, 18(3), 312-327., (2000). [žiūrėta 2018-03-11]
16. *Most marketers not profiting from social media*. [žiūrėta 2018-04-04] Prieiga per: http://web.a.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=1&sid=304db46f-ded6-4f30-9854-8a82473e4060%40sessionmgr4009"
17. Cooper, M. J., & Budd, C. S. *Tying the pieces together: A normative framework for integrating sales and project operations*. Industrial Marketing Management, 36(2), 173-182. (2007). [žiūrėta 2018-02-04]
18. Haven, B. *Marketing's new key metric: engagement*. Marketing. (2007). [žiūrėta 2018-03-04]
19. Freberg, K., Graham, K., McGaughey, K., & Freberg, L. A. *Who are the social media influencers? A study of public perceptions of personality.* Public Relations Review, 37(1), 90-92. (2011). [žiūrėta 2018-03-19]
20. Rinkūnas, D. *TNS LT: moterys dažniau naudojasi „Facebook", vyrai – „Youtube",* (2015). [žiūrėta 2018-04-30] prieiga per: http://www.tns.lt/lt/news/tns-lt-moterys-dazniau-naudojasi-facebook-%2C-vyrai-youtube-/
21. Juronienė, G. *"Kantar TNS" tyrimas: šalies reklamos rinka augo kaip tikėtasi – 3,8 proc.*, (2018). [žiūrėta 2018-05-04] prieiga per: http://www.tns.lt/lt/news/-kantar-tns%E2%80%9D-tyrimas-salies-reklamos-rinka-augo-kaip-tiketasi-3%2C8-proc-/
22. Karlson, K. 8 *Ways Intelligent Marketers Use Artificial Intelligence (2013).* [žiūrėta 2018-05-04] prieiga per: *http*://contentmarketinginstitute.com/2017/08/marketers-use-artificial-intelligence/
23. Acar, A. S., & Polonsky, M. *Online social networks and insights into marketing communications*. Journal of Internet Commerce, 6(4), 55-72. (2007). [žiūrėta 2018-04-04]
24. Kırcova, İbrahim Ve Enginkaya, Ebru (2015). Sosyal Medya Pazarlama. İstanbul: Beta.
25. Ryan, D. & Jones, C.. *Understanding digital marketing: marketing strategies for engaging the digital generation*. London and Philadelphia: Kogan Page. (2009) [žiūrėta 2018-04-04]
26. Buss, A. & Strauss, N. *The online communities handbook: building your business and brand on the web*. USA: New Riders (2009).) [žiūrėta 2018-06-04]
27. Eley, B. & Tilley, S. *Online Marketing Inside Out*. Australia: SitePoint Pty (2009). [žiūrėta 2018-03-04]

28. Kıetzman, J.H., Krıstopher, H.M,I.P. Sılvestr, B, , *Social media? Get serious! Understanding the Functional Building Blocks of Social Media*, Business Horizons, 54, 241-251 (2011) [žiūrėta 2018-03-04]
29. Asur, S., & Huberman, B. A. *Predicting the future with social media*. In Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01 (pp. 492-499). IEEE Computer Society. (2010) [žiūrėta 2018-03-04]
30. Gretzel, U. *Consumer generated content – trends and implications for branding e-Review of Tourism Research*, 4 (3), pp. 9-11" (2006) [žiūrėta 2018-03-04]
31. *Why should I convert my personal account to a Facebook Page?* [žiūrėta 2018-03-04] prieiga per: https://www.facebook.com/help/201994686510247
32. Ramsaran-Fowdar, Rooma Roshnee; Fowdar, Sooraj. Contemporary Management Research. Mar2013, Vol. 9 Issue 1, p73-83. 11p. DOI: 10.7903/cmr.9710. *Internet marketing on facebook* [žiūrėta 2018-03-04] prieiga per: http://web.a.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=1&sid=304db46f-ded6-4f30-9854-8a82473e4060%40sessionmgr4009"
33. Threatt, R. S. *Facebook and the Ideal Social Market Place: A Study of The Marketing Benefits of Social Media Practices*. Masters Thesis, University of Southern California, U.S.A. (2009). [žiūrėta 2018-03-04]
34. Berkowitz, D. A. (Ed.). *Social meanings of news: A text-reader*. Sage. (1997). [žiūrėta 2018-03-04]
35. Constine, J. *How Facebook News Feed Works*, (2016). [žiūrėta 2018-03-04] prieiga per: https://techcrunch.com/2016/09/06/ultimate-guide-to-the-news-feed/?guccounter=1
36. *Prepare to Advertise on Facebook* [žiūrėta 2018-02-12] prieiga per: https://www.facebook.com/business/help/714656935225188/?helpref=hc_fnav&bc[0]=AHCv1&bc[1]=Ads%20Help&bc[2]=Advertising%20Basics&bc[3]=About%20Facebook%20Advertising
37. Ángeles Oviedo-García, M., Muñoz-Expósito, M., Castellanos-Verdugo, M., & Sancho-Mejías, M. *Metric proposal for customer engagement in Facebook.* Journal of Research in Interactive Marketing, 8(4), 327-344. (2014). [žiūrėta 2018-03-04]
38. Facebook. *Organic reach on Facebook: your questions answered*, available at: www.facebook.com/business/news/Organic-Reach-on-Facebook (2014). [žiūrėta 2018-05-15]
39. Kacholia, V. *News feed FYI: Showing more high quality content* (2013) [žiūrėta 2018-03-04] Prieiga per:://newsroom.fb.com/news/2013/08/news-feed-fyi-showing-more-high-quality-content).

40. Ballings, M., Van den Poel, D., & Bogaert, M. *Social media optimization: Identifying an optimal strategy for increasing network size on Facebook*. Omega, 59, 15-25. (2016). [žiūrėta 2018-03-04]
41. Young, S. W., Tate, A. M., Rossmann, D., & Hansen, M. A. *The social media toll road: The promise and peril of Facebook advertising*. College & Research Libraries News, 75(8), 427-434. (2014). [žiūrėta 2018-03-04]
42. Socialbakers, *Alcohol Brands Shake Up Engagement, (2012).* [žiūrėta 2018-03-04] prieiga per: https://www.socialbakers.com/blog/1073-alcohol-brands-shake-up-engagement
43. Socialbakers, *Watch It Live: ROE Proven! Engagement Correlates With Reach*, (2012) . [žiūrėta 2018-03-04] prieiga per: https://www.socialbakers.com/blog/979-watch-it-live-roe-proven-engagement-correlates-with-reach/?ref=article
44. Ascend2:, *Marketing Strategy Report: Social Media*. Technical report, Ascend2 and Research Underwriters (2013). [žiūrėta 2018-02-11]
45. Lee, D., Hosanagar, K., & Nair, H. S. *The effect of social media marketing content on consumer engagement: Evidence from facebook.* Stanford Graduate School of Business. (2014). [žiūrėta 2018-03-11]
46. Coursaris, C. K., van Osch, W., & Balogh, B. A*Informing brand messaging strategies via social media analytics*. Online Information Review, 40(1), 6-24. . (2016). [žiūrėta 2018-03-04]
46. Cvijikj, I.P. and Michahelles, F. *Online engagement factors on Facebook brand pages*, Social Network Analysis and Mining , Vol. 3 No. 4, pp. 843-861. (2013). [žiūrėta 2018-05-04]
47. Kang, J. , Tang, L. and Fiore, A.M. *Enhancing consumer–brand relationships on restaurant Facebook fan pages: maximizing consumer benefits and increasing active participation*, International Journal of Hospitality Management (2014)). [žiūrėta 2018-03-04]
48. Kwok, L. and Yu, B. *Spreading social media messages on facebook an analysis of restaurant business-to-consumer communications*, Cornell Hospitality Quarterly , Vol. 54 No. 1, pp. 84-94. (2013) [žiūrėta 2018-03-04]
49. Coursaris, C. K., van Osch, W., & Balogh, B. A. *Informing brand messaging strategies via social media analytics. Online Information Review*, (2016). [žiūrėta 2018-03-04]
50. Su, N., Reynolds, D., & Sun, B. *How to make your Facebook posts attractive: A case study of a leading budget hotel brand fan page*. International Journal of Contemporary Hospitality Management, 27(8), 1772-1790. (2015). [žiūrėta 2018-03-04]

51. Malhotra, A., Malhotra, C. K., & See, A. *How to create brand engagement on Facebook*. MIT Sloan Management Review, 54(2), 18. (2013). [žiūrėta 2018-05-04]
52. Facebook, *Graph API Explorer Guide*, prieiga per: https://developers.facebook.com/docs/graph-api/explorer [žiūrėta 2018-04-13]
53. Socialbakers, *Lithuania Facebook page statistics*, prieiga per: https://www.socialbakers.com/statistics/facebook/pages/total/lithuania/ [žiūrėta 2018-02-05]
54. Scikitlearn, *Overview of clustering methods*, [žiūrėta 2018-03-04] prieiga per: http://scikit-learn.org/stable/modules/clustering.html#k-means
55. Reddy, C.K., Aggarwal, C.C., Chapman and Hall/CRC, ISBN: 9781498785778, *Data Clustering* , (2016)
56. Mirjalili, V. *Python Machine Learning - Second Edition, (2017).* [žiūrėta 2018-02-13] prieiga per: https://www.safaribooksonline.com/library/view/python-machine-learning/9781787125933/ch03s06.html
57. Lea, P. *internet of Things for Architects* (2018) [žiūrėta 2018-05-13] prieiga per: https://www.safaribooksonline.com/library/view/internet-of-things/9781788470599/e3418e8b-ac43-424b-887a-7882f154af7f.xhtml
58. Venkateswaran, B. *Neural Networks with R* (2017) [žiūrėta 2018-02-04] Prieiga per https://www.safaribooksonline.com/library/view/neural-networks-with/9781788397872/de017b88-eb53-4dda-a7c9-218dac637526.xhtml
59. Allison, P. *Logistic Regression Using SAS, 2nd Edition*, (2012). [žiūrėta 2018-02-04] prieiga per: https://www.safaribooksonline.com/library/view/logistic-regression-using/9781607649953/
60. Tax , D. M. J., Riddler, D, Heijden, F., Zoi, Y. , Feng, M., Xu, G., Lei, B., John Wiley & Sons, *Classification, Parameter Estimation and State Estimation, 2nd Edition (2017)*
61. *General Python FAQ*, [žiūrėta 2018-02-04] prieiga per https://docs.python.org/3/faq/general.html
62 *Documentation of scikit-learn 0.19.1*, [žiūrėta 2018-04-11] prieiga per:http://scikit-learn.org/stable/documentation.html
63 Tableau, *We make breakthrough products that change the way people use data* [žiūrėta 2018-04-15] prieiga per: https://www.tableau.com/about

Priedai

1 priedas

Pasirinkti „Facebook“ puslapiai

Žmonės ; LAIMA ; Ji ; ZIP FM ; BC Žalgiris ; www.simka.lt ; Fashion ; Kas vyksta Vilniuje ; diena.lt ; Vichy vandens parkas ; VERO MODA ; VARLE ; Vytautas ; Eurovaistinė ; Utenos Radler Nealkoholinis ; Starkaus & Radzevičiaus kelionės ; TV3 televizija ; Troliai ; Troliai ; Kelionės ; Total fun ; TopTravel ; TOPO CENTRAS ; Antanas Guoga ; Tymbark Lietuva ; Tez Tour Lietuva ; Telia Lietuva ; TELEGRAMA.LT ; Tele2 Lietuva ; Tekinis ; Tavo vaikas ; Sweety.lt ; Swedbank Lietuvoje ; Svarbiausia žmogus ; Stilingai.lt ; Stiliausidėjos.lt ; SPORTLAND ; Skonio reikalai ; Shkertik ; Lietuva senose fotografijose ; SenMergė ; Samsung ; Rūkymui NE ! ; Riešutai ; Radistai ; Internetinė radijos stotis - Radio Ritmas ; Radijo stotis "Radiocentras" ; Radijo stotis M-1 ; Pukuota ; Psichologija ; Protingos Mintys ; IKI ; Radijo stotis M-1 Plius ; Pirk.lt - Spalvingam gyvenimui ; Pigūs drabužiai ; Pigu.lt ; Pažiūrėk ; Simpsonai ; Panelė ; OZAS ; Outlet Park Lietuva ; Ot Dushy ; Online24.lt - geros prekės geromis kainomis ; Obuolys ; Nusišypsok ; Novaturas - tik ypatingi pasiūlymai ; NoriuPramogu.lt ; NoriuNoriuNoriu.lt ; Noriu atostogų! ; Norfa ; Njoy ; Nerūkau ir tuo didžiuojuosi! ; Neptunas ; Gamta ; Neįtikėtini faktai ; Zarasų Jaunimo Muzika ; Music+Music ; Motyvacija ; MOTERIS ; Mokslo Pasaulis ; Mojo Lounge ; Manija ; Milka ; Aš myliu savo mamą ; Milijonieriaus užrašai ; MemberShop ; Megalinkas ; McDonald's ; Maxima LT ; Mano pramogos! ; Mano Namai ; Man patink vairuot! ; Mamos receptai ; Makalius ; Maybelline New York Lietuva ; MADOS GURU ; MadeinVilnius.lt ; Luminor Lietuva ; LRT ; lrytas.lt ; L'Oréal Paris ; LOFTAS ; L'OCCITANE en Provence ; LNK TV ; Linas Kleiza ; Poilsis Palangoje ; Lietuvos balsas ; Lietuviška muzika! ; Lietuva ; Lietuva ; Lidl Lietuva ; Leon Somov & Jazzu ; Langų sistemos ; Laisvalaikio Dovanos ; Laikas - įdomus ir gyvas ; LABAS ; KREPŠINIO ŠIRDIS ; Knygų klubas ; Kitokie pasikalbėjimai ; KIKA ; Kelioniu Klausimai ; Dreamfly ; Kelioniu akademija ; Kauno diena ; Kaunas ; Kas vyksta Kaune ; Kasdien po akciją ; Kalnapilis nealkoholinis ; Juokingi video kasdien ; Jonas Valančiūnas ; horoskopai Community ; Jazzu ; Jevgenijus Cernys ; Jacobs ; kitaip.com ; Intymi Pagunda ; Impuls sveikatingumo klubai ; IKEA ; Humoro FM ; Humans of Trūlai ; HoroscoPosts ; Hesburger Lietuva ; Pure fun ; Happeak ; Grupinis.lt ; grozioguru.lt ; GJan ; Gyvūnų globos organizacija "LESĖ" ; Gyvenimas yra saldus ; Geležinė Lapė ; gaspadine.lt receptai ; GAMTOS GROŽIO FORMULĖ ; Fun for Fun ; Fun for all ; Forum Cinemas ; For-fun.lt ; Filmukai ; Fashion2Get.lt ; EŽYS ; EUROKOS ; Estrella ; Estela.Lt ; Senukai ; Elmenhorster ; DrogasLietuva ; Parfumerija Douglas Lietuva ; Donny Montell ; domoplius.lt ; Dirol Lietuva ; beta.lt ; Demotyvacija.lt ; DELFI.lt ; Deichmann ; Dar pažiūrėsim! ; Danija ; Dalia Grybauskaitė ; Crocs ; Cosmopolitan Lietuva ; Coca-Cola ; Club Music ; Exit Palanga ; Čili Pizza Lietuva ; Cido

*sultys ; Charlie Pizza ; CAN CAN pizza ; Caffeine LT ; Black Clouds ; BITĖ Lietuva ; Bilietai.lt ; bilder-welten.net ; Bernardinai.lt ; Benediktas Vanagas ; Nail and Beauty Vilnius ; Beatos Virtuvė ; BC Žalgiris Kaunas ; Balsas.lt ; balduturgus.lt ; Bacardi ; Autoplius.lt ; automanas ; Autogidas.lt ; Audimas ; Atsikvėpk nuo darbo ; Atrask Paulig kavos pasaulį ; ASUS Lietuva ; Aš Moteris - Žurnalas Moterims ; Aš myliu MADĄ ; Aš myliu muziką ; ArSkaitei.lt ; anekdotai.lt ; Andrius Mamontovas ; Alfa.lt ; AKROPOLIS | Vilnius ; AirGuru ; 5braškės.lt ; Draugai ; Nunu ; Pokštai ; Nemokamos ONE.LT paslaugos ; 15min ; MN ; Pasaulio idomybes ; Linksmos ir idomios naujienos ; Mokslo įdomybės ; Assorti ; Kasdien nauja akcija ; We love Lithuania ; Power Hit Radio Lithuania ; linksmiau.net ; Nusišypsok ; Gyvenk linksmiau; Berry baldai ITALY; Lauris Reiniks LATVIA; LG Electronics Lithuania ; Suaugusiems ; PLC Mega ; Rauginti agurkai ; AVON pasaulis ; Sil ; Galingas.lt Chip Tuning; Radijo stotis „Lietus“ ; Skaniausių šeimos kepinių knyga ; Taffel ; CV-Online LT ; LION ; Krepšinio fanai ; LSP.lt ; Kauno Akropolis ; Nemokami renginiai ; 1a - Your Smartstore ; Grožio chirurgija ; Studentų brolija ir seserija "barake.lt" ; Dienraštis „Klaipėda“ ; Kino siena ; Kablys ; Meilė ; Baldai1.lt ; Vero Cafe ; Celerauto ; PANORAMA ; Blue Sheep Born To Chill ; FISHKI.LT ; Išbandyti receptai ; Žaidimai ; Multikino Ozas ; Lithuania. Real is beautiful ; Lilas ir Innomine ; Moterys meluoja geriau ; G&G Sindikatas ; Knygynų tinklas „Pegasas“ ; ApieVestuves.lt ; BasketNews.lt ; Klasiškas vyras ; Vištiena Kitaip ; Orai.lt ; Ricardas Berankis ; Kauno savivaldybė ; MARSKINELIS.LT ; NESCAFÉ Lietuva ; Cardin ; Puikios nuotaikos puslapis ; RASA ; Tarp mūsų mergaičių ; Lietuvos futbolas ; Vakarų ekspresas ; Vintazine.lt ; Papuosaluguru.lt ; LinkoManija ; 15min Vardai ; Kas madinga šiais metais? ; Liūdni Slibinai ; Kitas Kampas ; *** N-18 *** ; Sushi Express ; Music.lt ITALY; Max Factor ; Dormeo LT ; Vaida Kurpiene - sveika mityba ; NEMOKAMI sąskaitos papildymai ! ; Visi receptai ; Keliones.lt - visos kelionės vienoje vietoje ; Linksmosios pėdutės ; Mano šuo ; Actors Agency Lithuania ; Parfum Express ; Dzordana Butkute ; TOP įdomybės ; Švyturys GO Nealkoholinis ; Ladies Fashion ; Pasirink Sparnus ; Asmeninė trenerė/mitybos specialistė Rūta ; Getstyle ; Istorijos ; Karklė ; Cha.lt; Pinigų karta ; Linas Dambrauskas Photography ; Kosmetikosguru.lt ; Režisierius Emilis Vėlyvis ; Komiksai ; Mykolo Romerio universitetas ; Žalgirio arena ; DELFI Maistas ; GRANATOS LIVE ; Keulė Rūkė ; Neoclubber ; Darau pats ; TheMusiclt.com ; Zuokas ; Lamų slėnis ; Emilio Vėlyvio filmai ; LITEXPO ; Lietuvos draudimas ; La Crepe ; Gyvūnėlių Mylėtojai ; 15min Verslas ; Garnier ; Krepšinio klubas ; „Vilkyškių pieninė“ oficialus puslapis ; SEB Lietuvoje ; Aš sveikas ; Circle K Lietuva ; Pigios kelionės - Greitai.lt ; Moterys vairuoja geriau ; „Karūna" šokoladas ; FĖJOS NAMAI ; Augink atsakingai ; iLietuva ; DJ Café ; Keliauju tik PASKUTINĘ MINUTĘ! ; Kiekvienas Gali Būti Didelis ; Moki-Veži. Sodui ; Neste Lietuva ITALY; Aš myliu savo šeimą! ; DOMUS LUMINA ; KELIONĖS nuo 1 Lt ; Mokslo ir technologijų pasaulis - Technologijos.lt ; Brangioji aš tikrai negirtas ; Gyvūnų gerovės tarnyba PIFAS ; Aprangos galerija ; NameList ; Geriausi*

vaizdeliai ; FC Vilniaus Žalgiris ; DejaVu ; Mano mėgstami dalykai ; C&D Style - dovanų ir interjero aksesuarų salonai ; OTTO.lt ; Europos Komisijos atstovybė Lietuvoje; PrieJuros.LT ; TonyResort ; Lietuvos kariuomenė ; Skittles Lietuva ; ISM Vadybos ir ekonomikos universitetas ; LaSpell.lt ; futbolo vergai ; juokas24.lt ; CAMELIA ; joms.lt ; Grynas ; Flirtas.lt ; TAUTINIS BRANDAS ; Valio ; Sportas.lt ; CVbankas.lt ; Vasaros Terasa ; Padlyzos ; Linksmi prikolai ; Olympic Casino ; BeeWood ; Domus galerija ; Parfumerinė ; Gera dovana - www.geradovana.lt ; SMS'ietis ; Megaturas ; Keistuolių Teatras ; LIETUVOS KREPŠINIS ; Kalba vilniečiai ; Skelbiu.lt ; Ir aš kalbu su savo Mikiu ; Volfas Engelman ; www.neocube.lt ; Kino teatras "Pasaka" ; Nieko Rimto ; Prabanga Grozis ; Gintarinė Vaistinė ; Lengvai iš pirmo karto ; TABOO club ; Manosveikata.lt ; Philips ; Laikas juoktis ; Uoga Uoga ; MAGGI ; General Financing ; DELFI Veidai ; okay.lt - eat today ; Ecodenta ; Ogmios miestas ; Drakono Ženklas ; Express Pizza ; Begalybės fragmentai ; Grožio Draugas ; DFDS Lietuva ; Vilniaus universitetas / Vilnius University ; Drift D1Sport racing team ; SAULĖS MIESTAS ; Pasiūlymų BANKAS ; Harmony Park ; EDITA ; ODA Kosmetika ; 15min Gazas ; AKROPOLIS | Klaipėda ; Friedricho pasažas ; Labai mėgstu kavą ; Moki-Veži. Apdailai ir statybai ; Kvapų namai ; Androidas vėl išsikrovė ; BTV ; ElektroMarkt Lietuva ; Elonas Penikas ; Firsty.lt ; Išskirtiniai namai ; Krekenavos kai norisi mėsos ; VASARA PALANGOJE ; Apie viska. ; X-clubshop ; Filosofija ; Klubas Pantera ; Jovani@club ; www.klase.lt ; Smilga Beauty Lab ; Neįtikėtinas pasaulis ; LIETUVOS TALENTAI ; MOLAS ; Neįtikėtinas pasaulis ; Sveikata.lt ; Annonymus Pabaltijoje ; Trollbeads ; L'Officiel Mada ; CONVERSE ; Žinių radijas ; Maggi 5Min Lithuania ; KitKat ; Hematogenas ; Geri Lęšiai .LT ; Kino Pavasaris | Vilnius International Film Festival ; Nampro Vilnius ; VIASAT Lietuva ; Autobroliai ; Meistro namai ; BARBORA.LT ; Caif cafe ; PriceOn ; Vilniaus duona. Visos Lietuvos duona ; Tadas Vidmantas ; DADU ledai ; Pigios prekės ; JYSK LT ; Sasha Song ; Dovanumanija.lt ; Lays Lietuva ; SuperDovana.Lt ; Cat Cafe Vilnius Kačių Kavinė ; Uogos.lt - akcijos kiekvieną dieną ; BENU Vaistinė ; Moterys meluoja geriau. ; fotogravija.lt ; The Body Shop ; Įdomios KAUNO vietos ; Spice.lt ; Biržų duona ; Mama.lt ; Pistonai ; Julia Janus ; SPORT1 ; Philips Avent ; Cropp ; ACME Lietuva ; Butų nuomos pasiūlymai Vilniuje ; Jurgis ir Drakonas ; CityBee - imu ir važiuoju ; KFC ; BC LIETUVOS RYTAS VILNIUS ; TV3 Play ; Tamsta Club ; Sveika šeima ; Kultūros naktis / Culture night Vilnius ; Avitela ; Žeurei močnas ; Prekybos miestelis Urmas ; VIVI kosmetika ; Flashmob Lietuva ; Vilnieciai ; Parfumerijos ir kosmetikos parduotuvė internete Olialiashop.lt ; Jumex Lietuva ; Filmukai.eu ; Šefo Receptai ; Trip Shop ; TV6 televizija ; JAV Ambasada Vilniuje / U.S. Embassy Vilnius; INŽI mados namai ; IamFashion.lt ; NEMOKAMI arba paskutinės minutės SEMINARAI! ; DPD Lietuva ; Leon Somov ; dyl.lt saviems ; atsikeli ir varai ; Trys seserys ITALY; Pet24.lt ; Pačios gražiausios Jūsų ir Mūsų mintys. ;

Duomenų išgavimo kodas

```
import requests
import json
import os
import pandas as pd
import numpy as np
import datetime
from datetime import date, datetime
from datetime import timedelta
import time
import re
import calendar
import datetime as dtm

#netinkamu simboliu salinimas
#list_of_bad_symbols=('\u26bd','ðŸ€','â›¹ï¸â€','â™‚ï¸','â›¹ï¸â€â™€ï¸','ðŸ‘Œ','ð
Ÿ‡±ðŸ‡¹ï¸','âœˆï¸','â˜€ï¸','â¤ï¸',
#                     'Ð¼Ð°Ñ-Ð°ÐºÑ€Ð¶-Ð¿Ð»Ð°Ð½ÐµÑ‚-Ñ€Ð²ÐµÑ,')
#RE_EMOJI = re.compile('[\U00010000-\U0010ffff]', flags=re.UNICODE)
# def strip_emoji(text):
#     text_lower_case=text.lower()
#     text_without_lt_letters=(text_lower_case.replace('Ä—','e')
#            .replace('Ä…','a').replace('Ä','c').replace('Ä™','e')
#            .replace('Ä¯','i').replace('Å¡','s').replace('Å³','u')
#            .replace('Å«','u').replace('Å¾','z'))
#     for emoji in list_of_bad_symbols:
#         text_without_lt_letters=text_without_lt_letters.replace(emoji,'')
#     text_without_emojis=text_without_lt_letters
#     text_with_encoded_correct_emojis =RE_EMOJI.sub(r'', text_without_emojis)
#     return text_with_encoded_correct_emojis
def remove_lt_letters(text):
    text_lower_case=text.lower()
    text_without_lt_letters=(text_lower_case.replace('Ä—','e')
           .replace('Ä…','a').replace('Ä','c').replace('Ä™','e')
           .replace('Ä¯','i').replace('Å¡','s').replace('Å³','u')
           .replace('Å«','u').replace('Å¾','z'))

    return text_without_lt_letters
def _removeNonAscii(s): return "".join(i for i in s if ord(i)<128)


#facebook token
#pasiemam is cia https://developers.facebook.com/tools/explorer
token =
'EAACEdEose0cBAD8tKiVKQI4VUhUQiZCxnZCLhXTbzsCVaZCCbnZAkrNQJuYlQTx9zvOxdIAvC9N6AGeCkL2yiOe
```

```
YEC2NjQTiCljRVls7absfZCaZBOJRyHALcYgRIt6nZC9l4jZCduTWh7aywvh5jC8iMtY0pC7gn0eWZA1MeRdDZC5q
Um5H2auziUORFxTWc0wKTIT2TgKPwmuQZDZD'


#max nusiurbiamu irasu skaicius per kreipimasi i api
post_limit = 759

#menesiu pradzia ir galas naudojama iteracijai kreipiantis i api
month_start_list = (
'05/15/2017',
'06/01/2017',
'07/01/2017',
'08/01/2017',
'09/01/2017',
'10/01/2017',
'11/01/2017',
'12/01/2017',
'01/01/2018',
'02/01/2018',
'03/01/2018',
'04/01/2018',
'05/01/2018')

month_end_list = (
'05/31/2017',
'06/30/2017',
'07/31/2017',
'08/31/2017',
'09/30/2017',
'10/31/2017',
'11/30/2017',
'12/31/2017',
'01/31/2018',
'02/28/2018',
'03/31/2018',
'04/30/2018'
'05/15/2018')


#api kreipimosi funkcija
def request_to_facebook_graph_api(token,page_name,post_limit,date_since,date_until):
    return
json.loads(requests.get("https://graph.facebook.com/v2.12/{page_name}?fields=category,fan
_count,posts.since({date_since}).until({date_until}).limit({post_limit})%7Bmessage%2C7B,l
ink,shares.limit(1).summary(true),likes.limit(0).summary(true),reactions.limit(0).summary
(true),created_time,comments.limit(0).summary(true)%7D&access_token={token}".format(page_
```

```
name=page_name, post_limit=post_limit,
token=token,date_since=date_since,date_until=date_until)).text)


def get_page_id(request_result):
    try:
        page_id= request_result['id']
    except:
        page_id = 'empty'
    return page_id

def get_page_category(request_result):
    try:
        page_category= request_result['category'].replace(',','')
    except:
        page_category = 'empty'
    return page_category

def get_page_fan_count(request_result):
    try:
        fan_count= request_result['fan_count']
    except:
        fan_count = 'empty'
    return fan_count
def get_page_posts(request_result):
    return request_result['posts']['data']
def get_post_id(post):
    try:
        post_id= post['id']
    except:
        post_id = 'empty'
    return post_id

def get_post_message(post):
    try:
        message = post['message'].replace(',',' ').replace('\n',' ').replace('\r','
').replace("""/""",' ').replace("'",'').replace('"','')
    except:
        message='empty'
    return message
def get_post_created_date(post):
    return post['created_time'][0:10]
def get_post_created_time(post):
    return post['created_time'][-13:-5]
def get_post_like_count(post):
    try:
        like_count=post['likes']['summary']['total_count']
```

```
    except:
        like_count=0
    return like_count

def get_post_reactions_count(post):
    try:
        reactions_count=post['reactions']['summary']['total_count']
    except:
        reactions_count=0
    return reactions_count



def get_post_comment_count(post):
    try:
        comments_count=post['comments']['summary']['total_count']
    except:
        comments_count=0
    return comments_count


def get_post_shares_count(post):
    try:
        shares= post['shares']['count']
    except:
        shares = 0
    return shares


def get_post_link(post):
    try:
        link= post['link'].replace(',','%2C')
    except:
        link='empty'
    return link


def make_date(date_string):
    return datetime.strptime(date_string, '%Y-%m-%d').date()
def get_weekday(post_created_date):
    return calendar.day_name[make_date(post_created_date).weekday()]


def get_time_of_day(post_created_time):
    hour_as_int=int(post_created_time[0:2])
    if hour_as_int >= 0 and hour_as_int <6:
        time_of_day = 'night'
```

```
    elif hour_as_int >=6  and hour_as_int <12:
        time_of_day = 'morning'
    elif hour_as_int >= 12 and hour_as_int <18:
        time_of_day = 'daytime'
    elif hour_as_int >= 18:
        time_of_day = 'evening'
    return time_of_day


def get_has_emoji(post_message):
    RE_EMOJI = re.compile('[\U00010000-\U0010ffff]', flags=re.UNICODE)
    if RE_EMOJI.sub(r'',post_message) != post_message:
        return 1
    else:
        return 0

def get_has_hashtag(post_message):
    hashtag = '#'
    if hashtag in post_message:
        return 1
    else:
        return 0


def get_has_exclamation(post_message):
    exclamation= '!'
    if exclamation in post_message:
        return 1
    else:
        return 0



def get_has_question_mark(post_message):
    question_mark= '?'
    if question_mark in post_message:
        return 1
    else:
        return 0


def get_message_symbol_count(post_message):
    if post_message == 'empty':
        length = 0
    else:
        length =len(post_message)
    return length
```

```
def get_message_word_count(post_message):
    if post_message == 'empty':
        length = 0
    else:
        length =len(post_message.split())
    return length


def get_message_contains_link(post_message):
    http= 'http'
    if http in post_message:
        return 1
    else:
        return 0

def get_has_link_attached(link):
    if 'videos' in link or 'youtube' in link or 'photos' in link:
        return 0
    else:
        return 1

def get_has_video_attached(link):
    if 'videos' in link or 'youtube' in link:
        return 1
    else:
        return 0

def get_has_photo_attached(link):
    if 'photos' in link:
        return 1
    else:
        return 0

def get_hours_since_last_post(date_before, date_now, page_name_before, page_name):
    datetimeFormat = '%Y-%m-%d %H:%M:%S'
    time1 = date_now.replace('T',' ')[0:19]
    time2 = date_before.replace('T',' ')[0:19]
    timediff = dtm.datetime.strptime(time1, datetimeFormat) -
dtm.datetime.strptime(time2,datetimeFormat)
    if page_name_before==page_name:
        hours_diff = round(timediff.seconds/3600,0)+timediff.days*24
    else:
        hours_diff = 999999
    return int(hours_diff)
```

```
def get_did_post_last_24h(date_before, date_now, page_name_before, page_name):
    datetimeFormat = '%Y-%m-%d %H:%M:%S'
    time1 = date_now.replace('T',' ')[0:19]
    time2 = date_before.replace('T',' ')[0:19]
    timediff = dtm.datetime.strptime(time1, datetimeFormat) -
dtm.datetime.strptime(time2,datetimeFormat)
    total_seconds = timediff.days*86400+timediff.seconds
    if total_seconds <= 86400 and page_name_before==page_name:
        return 1
    else:
        return 0

def get_did_post_last_72h(date_before, date_now, page_name_before, page_name):
    datetimeFormat = '%Y-%m-%d %H:%M:%S'
    time1 = date_now.replace('T',' ')[0:19]
    time2 = date_before.replace('T',' ')[0:19]
    timediff = dtm.datetime.strptime(time1, datetimeFormat) -
dtm.datetime.strptime(time2,datetimeFormat)
    total_seconds = timediff.days*86400+timediff.seconds
    if total_seconds <= 86400*3 and page_name_before==page_name:
        return 1
    else:
        return 0
def get_did_post_last_7d(date_before, date_now, page_name_before, page_name):
    datetimeFormat = '%Y-%m-%d %H:%M:%S'
    time1 = date_now.replace('T',' ')[0:19]
    time2 = date_before.replace('T',' ')[0:19]
    timediff = dtm.datetime.strptime(time1, datetimeFormat) -
dtm.datetime.strptime(time2,datetimeFormat)
    total_seconds = timediff.days*86400+timediff.seconds
    if total_seconds <= 86400*7 and page_name_before==page_name:
        return 1
    else:
        return 0



#duomenu surinkimas i viena eilute
def get_all_data_in_one_row(page_id,page_name, page_category,page_fan_count,post,
post_message, link,did_post_last_24h,did_post_last_72h,did_post_last_7d):
    return (str(page_id)+', '+
        str(page_name)+', '+
        str(page_category)+', '+
        str(page_fan_count) +', '+
        str(get_post_id(post))+', '+
        str(get_post_message(post))+', '+
        str(get_post_created_date(post))+', '+
```

```
        str(get_post_created_time(post))+', '+
        str(get_post_like_count(post))+', '+
        str(get_post_reactions_count(post))+', '+
        str(get_post_comment_count(post))+', '+
        str(get_post_shares_count(post))+', '+
        str(get_post_link(post))+', '+
        str(get_weekday(post_created_date))+', '+
        str(get_time_of_day(post_created_time))+', '+
        str(get_has_emoji(post_message))+', '+
        str(get_has_hashtag(post_message))+', '+
        str(get_has_exclamation(post_message))+', '+
        str(get_has_question_mark(post_message))+', '+
        str(get_message_symbol_count(post_message))+', '+
        str(get_message_word_count(post_message))+', '+
        str(get_message_contains_link(post_message))+', '+
        str(get_has_link_attached(link))+', '+
        str(get_has_video_attached(link))+', '+
        str(get_has_photo_attached(link))+', '+
        str(hours_since_last_post)+', '+
        str(did_post_last_24h)+', '+
        str(did_post_last_72h)+', '+
        str(did_post_last_7d)+
        '\n')

#kintamuju pvadinimu irasymas i failo pirma eilute
column_names = """page_id,page_name,
page_category,
fan_count,
post_id,
message,
created_date,
created_time,
like_count,
reactions_count,
comment_count,
shares_count,
link, weekday,
time_of_day,
has_emoji,
has_hashtag,
has_exclamation,
has_question_mark,
message_symbol_count,
message_word_count,
message_contains_link,
has_link_attached,
has_video_attached,
```

```
has_photos_attached,
hours_since_last_post,
did_post_last_24h,
did_post_last_72h,
did_post_last_7d
""").replace('\n','')+'\n'
os.remove('facebook_posts_data.txt')
with open('facebook_posts_data.txt', 'a') as the_file:
    the_file.write(column_names)

# tikrinama kurie puslapiai is saraso neturejo nei vieno iraso analizuojamu periodu


data = pd.read_csv('pagenames.txt', header = 0, sep = ';')

dead_pages = []
for page_name in data.fbid:
    api_result =
request_to_facebook_graph_api(token,page_name,1,month_start_list[0],month_end_list[11])
    try:
        get_page_posts(api_result)
    except:
        dead_pages.append(page_name)
        print(page_name)
len(dead_pages)

already_downloaded_pages = pd.read_csv('facebook_posts_data.txt', header = 0, sep = ',')

already_downloaded_pages.page_name.unique()

listas=('DELFI.Lietuva','DELFI.Lietuva')


page_name_before=''
date_now='1018-04-29T08:16:39+0000'
for page_name in data.fbid:
    #print(page_name)
    if page_name in dead_pages:
        continue
    if page_name.lower() in ' '.join(list(set(already_downloaded_pages.page_name))):
        continue
    for i in range (0,12):
        print(page_name +' '+month_start_list[i]+' '+month_end_list[i])
        api_result =
request_to_facebook_graph_api(token,page_name,post_limit,month_start_list[i],month_end_li
st[i])
        page_id = get_page_id(api_result)
```

```
        page_category = get_page_category(api_result)
        try:
            get_page_posts(api_result)
        except:
            print('fail')
            print(api_result)
            continue
        page_fan_count=get_page_fan_count(api_result)
        for post in get_page_posts(api_result):
            post_message = get_post_message(post)
            link = get_post_link(post)
            post_created_time=get_post_created_time(post)
            post_created_date=get_post_created_date(post)
            date_before=post['created_time']
            did_post_last_24h=get_did_post_last_24h(date_before, date_now,
page_name_before, page_name)
            did_post_last_72h=get_did_post_last_72h(date_before, date_now,
page_name_before, page_name)
            did_post_last_7d=get_did_post_last_7d(date_before, date_now,
page_name_before, page_name)
            hours_since_last_post=get_hours_since_last_post(date_before, date_now,
page_name_before, page_name)
            try:
                with open('facebook_posts_data.txt', 'a') as the_file:

the_file.write(_removeNonAscii(remove_lt_letters(get_all_data_in_one_row(page_id,page_nam
e, page_category,page_fan_count,post, post_message,
link,did_post_last_24h,did_post_last_72h,did_post_last_7d))))
                page_name_before=page_name
                date_now=date_before

            except:
                print(get_all_data_in_one_row(get_all_data_in_one_row(page_id,page_name,
page_category,page_fan_count,post, post_message, link,date_before, date_now,
page_name_before)))
                with open('facebook_posts_data.txt', 'a') as the_file:
the_file.write(_removeNonAscii(remove_lt_letters(get_all_data_in_one_row(page_id,page_nam
e, page_category,page_fan_count,post, post_message, link,date_before, date_now,
page_name_before))))
                break
# In[116]:
request_to_facebook_graph_api(token,'',post_limit,'07/16/2017','07/30/2017')
```

3 priedas

Duomenų paruošimas klasterizavimui

```
import pandas as pd
```

```
import numpy as np

from pyspark.sql import SparkSession
spark = (SparkSession.builder
    .appName('tmp_er')
    .config('hive.exec.dynamic.partition', 'true')
    .config('hive.exec.dynamic.partition.mode', 'nonstrict')
    .config('hive.merge.sparkfiles', 'true')
    .config('parquet.compression', 'snappy')
    .config("spark.sql.broadcastTimeout",  3600)
    .enableHiveSupport()
    .getOrCreate())
df = pd.read_csv('abc.csv')
df=sqlContext.createDataFrame(df)
df.registerTempTable("df")

database_name = 'db_apps_exacaster_dm'
table_name='tmp_er'

spark.sql("""drop table
{database_name}.{table_name}""".format(database_name=database_name, table_name  =
table_name ))
def create_table_tmp_er_key_metrics(spark,  database_name, table_name):
    spark.sql("""
      CREATE TABLE IF NOT EXISTS {database_name}.{table_name} (
        page_id string,
        page_name string,
        page_category string,
        fan_count int,
        post_id string,
        like_count int,
        reactions_count int,
        comment_count int,
        shares_count int,
        created_date string,
        created_time string,
        weekday string,
        time_of_day string,
        has_emoji int,
        has_hashtag int,
        has_exclamation int,
        has_question_mark int,
        message_symbol_count int,
        message_word_count int,
        message_contains_link int,
        has_link_attached int,
        has_video_attached int,
```

```
        has_photos_attached int,
        hours_since_last_post int,
        did_post_last_24h int,
        did_post_last_72h int,
        did_post_last_7d int,
        message string
      ) partitioned BY (yearmonth string)
    """.format(database_name=database_name, table_name  = table_name ))

create_table_tmp_er_key_metrics(spark,  database_name, table_name)
def insert_data_tmp_er(spark, table, table_name, database_names):

    temp_table_name = 'dff'
    table.createOrReplaceTempView(temp_table_name)

    (spark.sql("""
              SELECT
                page_id,
                page_name,
                page_category,
                fan_count,
                post_id,
                like_count,
                reactions_count,
                comment_count,
                shares_count,
                ` created_date`,
                created_time,
                ` weekday`,
                time_of_day,
                has_emoji,
                has_hashtag,
                has_exclamation,
                has_question_mark,
                message_symbol_count,
                message_word_count,
                message_contains_link,
                has_link_attached,
                has_video_attached,
                has_photos_attached,
                hours_since_last_post,
                did_post_last_24h,
                did_post_last_72h,
                did_post_last_7d,
                message,
                concat(year(ltrim(` created_date`)),month(ltrim(` created_date`))) as
year_month
```

```
            FROM dff
            """.format(
                    temp_table=temp_table_name
                )).coalesce(40)
                  .write
                  .option('compression','snappy')
                  .format('parquet')
                  .insertInto('{database_name}.{table_name}
'.format(database_name=database_name, table_name  = table_name ), overwrite = True))

table=spark.table('df')

insert_data_tmp_er(spark, table, table_name, database_name)
spark.table('db_apps_exacaster_dm.tmp_er').printSchema()


correct_names = pd.read_csv('correct_names.txt')
correct_names=sqlContext.createDataFrame(correct_names)
correct_names.registerTempTable("correct_names")

spark.sql("""select

page_id ,
page_name  ,
page_category,
fan_count  ,
post_id ,
like_count ,
reactions_count ,
comment_count,
shares_count  ,
created_date  ,
created_time  ,
weekday ,
time_of_day  ,
has_emoji  ,
has_hashtag  ,
has_exclamation ,
has_question_mark  ,
message_symbol_count ,
message_word_count ,
message_contains_link,
has_link_attached  ,
has_video_attached ,
has_photos_attached  ,
hours_since_last_post,
did_post_last_24h  ,
```

```
did_post_last_72h  ,
did_post_last_7d   ,
message ,
yearmonth  ,


case when
((message like  '%nugale%' or message like '%laime%' or message like '%laiming%')
and (message like '%skelbsi%' or message like '%rinksi%')
)
or (message like '%pamegt%' and message like  '%pasidalint%')
or (message like '%jei norite laimeti%')
or (message like '%#konkursas%')
or (message like '%dovanosim%')
or (message like 'zymekit')
or (message like 'komentaruose parasyk')
then 1 else 0 end as has_share_contest ,
row_number () over (partition BY post_id ORDER BY post_id desc ) = 1 as dedublicator
from db_apps_exacaster_dm.tmp_er
where hours_since_last_post >= 0 and hours_since_last_post <> 999999
and lower(page_name) not in ('neitiketinaspasaulisofficial','troliai','lietuva')
and  page_id in (select page_id from correct_names )
group by 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30

 """).createOrReplaceTempView("data")
spark.sql("""select * from data where dedublicator = 1
""").toPandas().to_csv('final_m_df_20180516_semi.csv',sep=';')
spark.sql("""select count(*),count(distinct post_id), count(distinct page_id) from data
where dedublicator = 1  """).toPandas()

spark.sql("""select count(*),count(distinct post_id), count(distinct page_id) from data
where dedublicator = 1  """).toPandas()

# # data for clusters
# - percent_Of_posts_with_video
# - percnet_of_posts_with_photo
# - percnet_of_posts_with_link
# - percent_of_posts_with_contest
# - avg_message_word_count
# - avg_hours_since_last_post
# - avg_message_word_count
# - post_count

# In[27]:

spark.sql("""
select page_name,page_id,
```

```
sum(has_video_attached)/count(*) as percent_of_posts_with_video,
sum(has_photos_attached)/count(*) as percent_of_posts_with_photo,
sum(has_link_attached)/count(*) as percent_of_posts_with_link,
sum(has_share_contest)/count(*) as percent_of_posts_with_share_contest,
avg(message_word_count) as avg_message_word_count,
avg(hours_since_last_post) as avg_hours_since_last_post,
count(*) as post_count
from data
where hours_since_last_post >= 0 and hours_since_last_post <> 999999
group by page_name,page_id
order by page_name
""").createOrReplaceTempView("data_for_clustering")


spark.table('data_for_clustering').toPandas().to_csv('ccc.txt')

# # PREPARING DATA FOR CLA


p_df = pd.read_csv('puslapiupavadinimai.txt',sep=';')
p_df=sqlContext.createDataFrame(p_df)
p_df.registerTempTable("p_df")

ddd = pd.read_csv('final_m_df_20180516_semi.csv',sep=';')
ddd=sqlContext.createDataFrame(ddd)
ddd.registerTempTable("ddd")

# post on that day count
spark.sql("""select a.post_id, count(*) as posts_made_on_given_day
--from data as a
from ddd as a
left join data as b on a.page_id = b.page_id and a.created_date = b.created_date
group by 1 """).registerTempTable("p_count")


# In[25]:

spark.sql("""
select page_id,
post_id,
case when has_emoji =1 then 'TRUE' ELSE 'FALSE' END AS  has_emoji,
case when has_hashtag =1 then 'TRUE' ELSE 'FALSE' END AS  has_hashtag,
case when has_exclamation =1 then 'TRUE' ELSE 'FALSE' END AS  has_exclamation,
case when has_question_mark =1 then 'TRUE' ELSE 'FALSE' END AS  has_question_mark,
message_word_count,
case when message_contains_link =1 then 'TRUE' ELSE 'FALSE' END AS
message_contains_link,
```

```
CASE WHEN has_link_attached = 1 THEN 'TRUE' ELSE 'FALSE' END AS  has_link_attached,
CASE WHEN has_video_attached = 1 THEN 'TRUE' ELSE 'FALSE' END AS  has_video_attached,
CASE WHEN has_photos_attached = 1 THEN 'TRUE' ELSE 'FALSE' END AS  has_photos_attached,
hours_since_last_post,
weekday,
time_of_day,

CASE WHEN has_share_contest = 1 THEN 'TRUE' ELSE 'FALSE' END AS  has_share_contest,

engaged
from(

select
a.post_id,
a.page_id,
a.has_emoji,
a.has_hashtag,
a.has_exclamation,
a.has_question_mark,
a.message_word_count,
a.message_contains_link,
a.has_link_attached,
a.has_video_attached,
a.has_photos_attached,
a.hours_since_last_post,
a.weekday,
a.has_share_contest,
a.time_of_day,
case when(((a.reactions_count + a.comment_count +
a.shares_count)/p_count.posts_made_on_given_day)/p_df.total_fans)*100
  > 0.1 then 'engaged' else 'not_engaged' end as engaged,
  row_number () over (partition BY a.post_id ORDER BY a.post_id desc ) = 1 as
dedublicator
--from db_apps_exacaster_dm.tmp_er as a
from ddd as a
--inner join data_for_clustering as c on c.page_id = a.page_id
inner join p_df on ltrim(lower(p_df.fbid)) = ltrim(lower(a.page_name)) --and
ltrim(lower(p_df.fbid))
inner join p_count on ltrim(lower(p_count.post_id)) = ltrim(lower(a.post_id))) a
where dedublicator = 1
""").toPandas().to_csv('data_for_classification.csv')

num_lines = sum(1 for line in open('data_for_classification.csv'))
num_lines

spark.sql("""
select count(distinct a.page_id),count(*)
```

```
from(
select a.page_id, a.post_id, row_number () over (partition BY a.post_id ORDER BY
a.post_id desc ) = 1 as dedublicator
from db_apps_exacaster_dm.tmp_er as a
--inner join c_df on ltrim(lower(c_df.page_name)) = ltrim(lower(a.page_name))
inner join p_df on ltrim(lower(p_df.fbid)) = ltrim(lower(a.page_name)) --and
ltrim(lower(p_df.fbid))
--not in ('neitiketinaspasaulisofficial','troliai','lietuva')
inner join p_count on ltrim(lower(p_count.post_id)) = ltrim(lower(a.post_id))
inner join data_for_clustering as c on c.page_id = a.page_id)
where dedublicator = 1

""").toPandas()

sum(1 for line in open('data_for_classification.csv'))
sum(1 for line in open('ccc.txt'))
sum(1 for line in open('data_for_classification.csv'))
#clean page ids
spark.sql("""
select distinct a.page_id
from db_apps_exacaster_dm.tmp_er as a
inner join p_df on ltrim(lower(p_df.fbid)) = ltrim(lower(a.page_name)) --and
ltrim(lower(p_df.fbid))
inner join p_count on ltrim(lower(p_count.post_id)) = ltrim(lower(a.post_id))
inner join data_for_clustering as c on c.page_id = a.page_id
""").toPandas().to_csv('correct_names.txt')

correct_names = pd.read_csv('correct_names.txt')
correct_names=sqlContext.createDataFrame(correct_names)
correct_names.registerTempTable("correct_names")
```

4 priedas

Klasterizvavimo kodas

```
import pandas as pd
from sklearn import preprocessing
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

df = pd.read_csv('data_for_clustering.txt', sep = ",").drop("Unnamed:
0",1).drop("""page_name""", 1)
```

```
data = df[['percent_of_posts_with_video', 'percent_of_posts_with_photo',
'percent_of_posts_with_link',
          'percent_of_posts_with_share_contest', 'avg_message_word_count',
'avg_hours_since_last_post','post_count']]


min_max_scaler = preprocessing.MinMaxScaler()
np_scaled = min_max_scaler.fit_transform(data)
df_normalized = pd.DataFrame(np_scaled)


data = df_normalized


sse = {}
for k in range(1, 10):
    kmeans = KMeans(n_clusters=k ).fit(data)
    data["clusters"] = kmeans.labels_
    #print(data["clusters"])
    sse[k] = kmeans.inertia_ # Inertia: Sum of distances of samples to their closest
cluster center
plt.figure()
plt.plot(list(sse.keys()), list(sse.values()))
plt.xlabel("Number of cluster")
plt.ylabel("SSE")
plt.show()

kmeans = KMeans(n_clusters=3, max_iter=1000).fit(data)
kmeans.labels_



print(kmeans.labels_.tolist().count(0))
print(kmeans.labels_.tolist().count(1))
print(kmeans.labels_.tolist().count(2))


df[['page_id','post_count']].join(pd.Series(kmeans.labels_.tolist(), name
='cluster')).drop('post_count',1).to_csv('page_id_and_its_cluster.csv')
```

5 priedas

Klasifikavimo kodas

```
from __future__ import print_function
import os
import subprocess
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier, export_graphviz
import io
import pydotplus
import graphviz
import sys
from __future__ import print_function
import os
import subprocess
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.tree import DecisionTreeClassifier, export_graphviz
import io
import pydotplus
import graphviz
from scipy import misc
from sklearn import metrics
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification
from sklearn.metrics import roc_auc_score
from sklearn.metrics import confusion_matrix
from sklearn import svm
from sklearn.neural_network import MLPClassifier
from sklearn import linear_model
from sklearn.metrics import roc_curve, auc
from sklearn.model_selection import RandomizedSearchCV

#file = open('data_for_classification.csv', 'r')
#a=file.read()
#text_file = open("data_for_classification_clean.csv", "w")
#text_file.write(a.replace(' ',''))
#text_file.close()

#data_for_classification =
pd.read_csv('data_for_classification_clean.csv').drop("""Unnamed: 0""", 1)
data_for_classification =
pd.read_csv('data_for_classification_with_dummies.csv').drop("""Unnamed: 0""", 1)
```

```
data_of_clusters = pd.read_csv('page_id_and_its_cluster.csv').drop("""Unnamed: 0""", 1)
df = pd.merge(data_for_classification, data_of_clusters, on=['page_id'])
def encode_target(df, target_column):
    """Add column to df with integers for the target.

    Args
    ----
    df -- pandas DataFrame.
    target_column -- column to map to int, producing
                     new Target column.

    Returns
    -------
    df_mod -- modified DataFrame.
    targets -- list of target names.
    """
    df_mod = df.copy()
    targets = df_mod[target_column].unique()
    map_to_int = {name: n for n, name in enumerate(targets)}
    df_mod["Target"] = df_mod[target_column].replace(map_to_int)

    return (df_mod, targets)
def show_tree(tree, features, path):
    f = io.StringIO()
    export_graphviz(tree, out_file=f,feature_names=features)
    pydotplus.graph_from_dot_data(f.getvalue()).write_png(path)
    img = misc.imread(path)
    plt.rcParams["figure.figsize"] = (20,20)
    plt.imshow(img)
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
```

```
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')



# raw variantas
def get_model_evaluation(model_name, y_test,y_pred):
    print()
    print(model_name)
    print('AUC '+ str(roc_auc_score(y_test, y_pred)))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
    print('tp '+str(tp))
    print('tn '+str(tn))
    print('fp '+str(fp))
    print('fn '+str(fn))
    print('accuracy '+str(accuracy_score(y_test, y_pred)))
    print('preciziskumas '+str(precision_score(y_test, y_pred)))
    print('atkurimas ' +str(recall_score(y_test, y_pred)))
    print('f1 ivertis ' + str(f1_score(y_test, y_pred)))
    print('jautrumas ' + str(tp/(tp+fn)))
    print('specifiskumas '+str(tn/(fp+tn)))
    print('')
    print('confusion matrica')
    print(str(tp)+' '+str(fn))
    print(str(fp)+' '+str(tn))


#variantas dedant tiesiai i worda
def get_model_evaluation(model_name, y_test,y_pred):
    print()
    print(model_name + ',Taiklumas,'+ str(accuracy_score(y_test, y_pred)))
    print(model_name +',AUC,'+ str(roc_auc_score(y_test, y_pred)))
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
    print(model_name +',TP,'+str(tp))
```

```
    print(model_name +',TN,'+str(tn))
    print(model_name +',FP,'+str(fp))
    print(model_name +',FN,'+str(fn))
    print(model_name +',Preciziskumas,'+str(precision_score(y_test, y_pred)))
    print(model_name +',Atkūrimas,' +str(recall_score(y_test, y_pred)))
    print(model_name +',F1 įvertis,' + str(f1_score(y_test, y_pred)))
    print(model_name +',Jautrumas,' + str(tp/(tp+fn)))
    print(model_name +',Specifiskumas,'+str(tn/(fp+tn)))




def use_models(df):
    df2, targets = encode_target(df, "engaged")
    features = list(df2.columns[:22])
    X = df2[features]
    y = df2["Target"]
    X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
    #decision tree
    dt = DecisionTreeClassifier(max_depth=20, random_state=1) #,min_samples_split=5000
    dt_y_pred = dt.fit(X_train, y_train).predict(X_test)
    get_model_evaluation('Sprendimų medis', y_test,dt_y_pred)
    #random forest
    rf = RandomForestClassifier(n_estimators = 100, max_depth=20, random_state=1)
    rf_y_pred = rf.fit(X_train, y_train).predict(X_test)
    get_model_evaluation('Atsitiktiniai miškai', y_test,rf_y_pred)
    #neural net
    nnet = MLPClassifier(solver='lbfgs', alpha=1e-5,
    hidden_layer_sizes=(5, 2), random_state=1)
    nnet_y_pred = nnet.fit(X_train, y_train).predict(X_test)
    get_model_evaluation('Neuroninis tinklas', y_test,nnet_y_pred)
    #log reg
    logreg = linear_model.LogisticRegression(C=1e5)
    logreg_y_pred = logreg.fit(X_train, y_train).predict(X_test)
    get_model_evaluation('Logistinė regresija', y_test,logreg_y_pred)


print(df['engaged'].value_counts())
df_engaged = df.loc[df['engaged'] == 'not_engaged']
count_of_values_to_remove = df['engaged'].value_counts()[0]-
df['engaged'].value_counts()[1]
print(count_of_values_to_remove)
import pandas as pd
import numpy as np
np.random.seed(10)

remove_n = count_of_values_to_remove
```

```
drop_indices = np.random.choice(df_engaged.index, remove_n, replace=False)
balanced_df = df.drop(drop_indices)
print(balanced_df['engaged'].value_counts())



df_unclustered = balanced_df.drop('cluster',1).drop('page_id',1)#.replace(' ','')
use_models(df_unclustered)
df_unclustered = balanced_df.drop('cluster',1).drop('page_id',1)#.replace(' ','')
df_unclustered=balanced_df

df2, targets = encode_target(df_unclustered, "engaged")
features = list(df2.columns[:22])
X = df2[features]
y = df2["Target"]


X.count()




X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)



dt = DecisionTreeClassifier(max_depth=20, random_state=99) #,min_samples_split=5000
dt_y_pred = dt.fit(X_train, y_train).predict(X_test)
get_model_evaluation('decision tree', y_test,dt_y_pred)


#https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-
scikit-learn-28d2aa77dd74
max_depth = [int(x) for x in np.linspace(start = 5, stop = 30, num = 6)]
min_samples_split=[int(x) for x in np.linspace(start = 50, stop = 1000, num = 20)]
random_grid = {'max_depth': max_depth,
               'min_samples_split': min_samples_split}
dt_random = RandomizedSearchCV(estimator = dt, param_distributions = random_grid,  cv =
10, verbose=2, random_state=42, n_jobs = -1)
# Fit the random search model

dt_y_pred = dt_random.fit(X_train, y_train).predict(X_test)
get_model_evaluation('decision tree', y_test,dt_y_pred)

dt_random.best_params_

rf = RandomForestClassifier(n_estimators = 400, max_depth=70, random_state=0)
```

```
rf_y_pred = rf.fit(X_train, y_train).predict(X_test)
get_model_evaluation('random forest', y_test,rf_y_pred)



#https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 800, num = 5)]
max_depth = [int(x) for x in np.linspace(start = 5, stop = 30, num = 5)]
random_grid = {'n_estimators': n_estimators,
               'max_depth': max_depth}
rf_random = RandomizedSearchCV(estimator = rf, param_distributions = random_grid,  cv = 10, verbose=2, random_state=42, n_jobs = -1)
# Fit the random search model

rf_y_pred = rf_random.fit(X_train, y_train).predict(X_test)
get_model_evaluation('random forest', y_test,rf_y_pred)



rf_random.best_params_

nnet = MLPClassifier(solver='lbfgs', alpha=1e-5,
                     hidden_layer_sizes=(5, 2), random_state=1)
nnet_y_pred = nnet.fit(X_train, y_train).predict(X_test)
get_model_evaluation('neural net', y_test,nnet_y_pred)


#log reg
logreg = linear_model.LogisticRegression(C=1e5)
logreg_y_pred = logreg.fit(X_train, y_train).predict(X_test)
get_model_evaluation('logisine regresija', y_test,logreg_y_pred)
df_cluster_1 = df.loc[df['cluster'] == 2].drop('cluster',1).drop('page_id',1)
```