

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS**

**Savarankiško projekto ataskaita**

Studijų modulio

P160M132 DIDŽIŲJŲ DUOMENŲ RINKINIŲ TYRYBOS METODAI

|  |  |
| --- | --- |
|  |  |
| **Darbą parengė:**  Anatolijus Nachumovas  Martynas Ragaišis  Ugnė Butkutė | (parašas) (data) |
|  |  |
| **Darbą priėmė:**  lekt. D. Aliulis | (parašas) (data) |
|  |  |

**Vilnius, 2019**

Turinys

[Turinys 2](#_Toc29148104)

[1. Duomenų rinkinio žvalgomoji analizė 4](#_Toc29148105)

[1.1. Duomenų eliminavimas 4](#_Toc29148106)

[2. Atrinktų duomenų klasterizavimas 6](#_Toc29148107)

[2.1. Duomenų apžvalga prieš klasterizavimą 6](#_Toc29148108)

[2.2. Duomenų klasterizavimas 6](#_Toc29148109)

[3. Klasterių analizė ir prognozavimas 8](#_Toc29148110)

[Išvados 10](#_Toc29148111)

[Literatūros sąrašas 11](#_Toc29148112)

DARBO TIKSLAS

Panaudojus išmoktus duomenų tyrybos metodus ir Apache Spark bei Python programines priemones, sudaryti mobilaus ryšio operatoriaus klientų segmentavimo ir klientų praradimo prognozavimo modelius. Atlikti žvalgomosios duomenų analizės, modeliavimo, modelių įvertinimo programinę realizaciją. Parengti duomenų tyrybos rezultatų aiškinamąjį raštą su išvadomis.

DUOMENYS

Duomenų rinkiniai pateikti Moodle sistemoje nurodytu adresu. Duomenų rinkinį kiekvienam studentui sudaro 2 failai:

1. customer\_usage\_**<varianto\_nr>**.csv – agreguoti vartotojų birželio-rugpjūčio mėnesių paslaugų vartojimo duomenys. Kintamasis user\_account\_id nurodo kliento identifikacinį ID.

2. customer\_churners\_**<varianto\_nr>**.csv – klientų praradimo istorija. Kintamojo churn reikšmė 1 nurodo, kad klientas rugsėjo mėnesį paliko kompaniją, 0 reiškmė nurodo, kad klientas toliau naudojosi paslaugomis.

UŽDUOTIS

1. Atlikti duomenų rinkinio žvalgomąją analizę.

2. Panaudojus duomenų kokybės gerinimo ir K–vidurkių arba GMM metodus sudaryti telekomunikacijų kompanijos klientų geriausią segmentavimo modelį ir parinkti optimalų segmentų skaičių pagal pasirinktą klasterizavimo metriką.

3. Interpretuoti gautus segmentus.

4. Panaudojus klasifikavimo metodus sudaryti klientų praradimo prognozavimo modelius visiems segmentams. Įvertinti sudarytų modelių prognozavimo kokybę pasirinktomis metrikomis.

PAPILDOMA UŽDUOTIS

5. Parinkti optimalų segmentų skaičių pagal klientų praradimo prognozavimo kokybę (kiekvienam segmentui sudarius po klasifikavimo modelį ir jį įvertinus pagal pasirinktą klasifikavimo metriką).

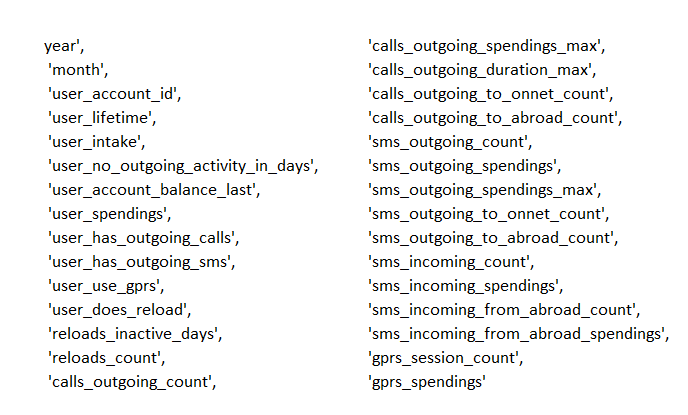
# Duomenų rinkinio žvalgomoji analizė

Mūsų grupės gauto duomenų rinkinio numeris – 4. Pradinėje darbo stadijoje turime du skirtingus .csv tipo failus: customer\_churn\_00004.csv bei customer\_usage\_00004.csv. Duomenų rinkinį sudaro – 49885 skirtingų vartotojų id.

## Duomenų eliminavimas

Atlikus duomenų apžvalgą, buvo nuspręsta pasirinkti 8 skirtingas duomenų grupes, kurių pagalba įgyvendinsime savo projektą. Tam, kad įvertintume, kurie duomenys yra mums naudingi, o kurių eliminavimas nesukeltų didelių kokybės pokyčių projektui, atlikome duomenų koreliacijos tyrimą.

Koreliacijos koeficientus skaičiavome Spearmano bei Pearsono metodais, tačiau buvo priimtas sprendimas atsižvelgti į pastarojo metodo gautus sprendinius. Didesnis duomenų kiekis turėjo aukštesnį nei 0.75 koreliacijos koeficientą. Remiantis gautais rezultatais duomenų rinkinys buvo padalintas į dvi dalis: duomenys, kurių tolimesnėje analizėje nenaudosime ir duomenys, kurie gali būti naudojami tolimesniuose žingsniuose.



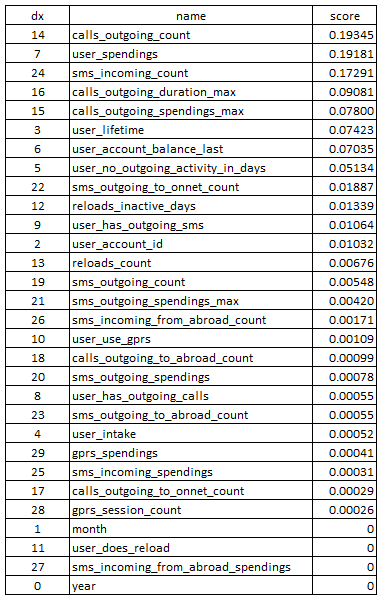
**1 pav.** Duomenų grupė likusi po koreliacijos etapo

Suskirsčius turimą duomenų rinkinį į dvi grupes buvo nuspręsta galutiniam projekto variantui išsirinkti 8 svarbiausius ir daugiausia informacijos galinčius suteikti duomenų stulpelius. Šiame žingsnyje buvo nuspręsta įvertinti mūsų grupės įžvalgas bei atsitiktinių medžių klasifikatoriaus pateiktus rezultatus. Tokie duomenys, kaip „user\_spending“, „user\_no\_outgoing\_activity\_in\_days“, „sms\_incoming count“ bei „calls outgoing\_count“ buvo išskirti tiek mūsų grupės įžvalgose, tiek atsitiktinių medžių klasifikatoriaus rezultatuose.

***RandomForestClassifier(labelCol="churn", featuresCol="initial\_features", seed = 8464,***

***numTrees=10, cacheNodeIds = True, subsamplingRate = 0.7)***

Aukščiau pateikiame sukurtą atsitiktinių medžių klasifikatoriaus apmokymo kodą. Pasinaudojus juo gauname sprendinius, kuriuos palyginame su savo grupės padarytomis įžvalgomis ir galutinai nusprendžiame, kuriuos duomenis įtraukti tarp 8 analizei naudotinų (žr. 2 pav.).



**1 lentelė** Atsitiktinių medžių klasifikatoriaus rezultatai

Atsižvelgiant į gautus sprendinius, matome, kad „calls\_outgoing\_count“, „user\_spendings“ bei „sms\_incoming\_count“ surinko didžiausius balus. Šie trys duomenų rinkiniai taip pat buvo įtraukti per mūsų grupės diskusiją.

Tiesa, nors koreliacijos vertinimo metu buvo atmesti ,,calls\_outgoing\_spending“ duomenys, tačiau vėliau nutarta šiuos duomenis taip pat įtraukti tarp 8 svarbiausių ir tolimesniuose žingsniuose naudotinų duomenų. Pastebėta, kad šie duomenys koreliuoja su duomenų rinkiniais, kurie nebus naudojami sekančiuose analizės žingsniuose.

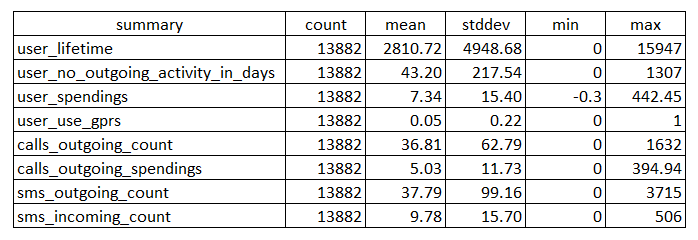
Taigi, sekančiuose skyriuose analizuosime šiuos duomenis: ,,calls\_outgoing\_count“, „user\_spending“, „sms\_incoming\_count“, „user\_use\_gprs“, „sms\_outgoing\_count“, „user\_lifetime“, „user\_no\_outgoing\_activity\_in\_days“ bei „calls\_outgoing\_spending“. Panaudojus atsitiktinių medžių klasifikatoriaus gautus rezultatus ir mūsų išvadas manome, jog šie duomenys yra tinkamiausi gauti išsamias įžvalgas šios telekomunikacijų paslaugas teikiančios įmonės vartotojų įpročių tendencijas.

# Atrinktų duomenų klasterizavimas

Šioje darbo dalyje pateikiame išsirinktų duomenų klasterizavimo metodus ir gautus rezultatus. Klasterizavimas bus atliekamas su 8 skirtingais duomenų rinkiniais, kurių pasirinkimai buvo atlikti po pirminės duomenų apžvalgos.

## Duomenų apžvalga prieš klasterizavimą

Žemiau pateiktoje lentelėje matome išrinktų duomenų imtį (count), vidurkį (mean), standartinį nuokrypį (stddev) bei minimalias (min) ir maksimalias (max) reikšmes (žr. 3 pav.). Matome, kad didesnį nuokrypį turi „user\_no\_outgoing\_activity\_in\_days“ bei „user\_use\_gprs“. Tiesa, išskirčių galima tikėtis atsižvelgiant į maksimalias reikšmes kiekviename iš nagrinėjamų duomenų rinkinių.

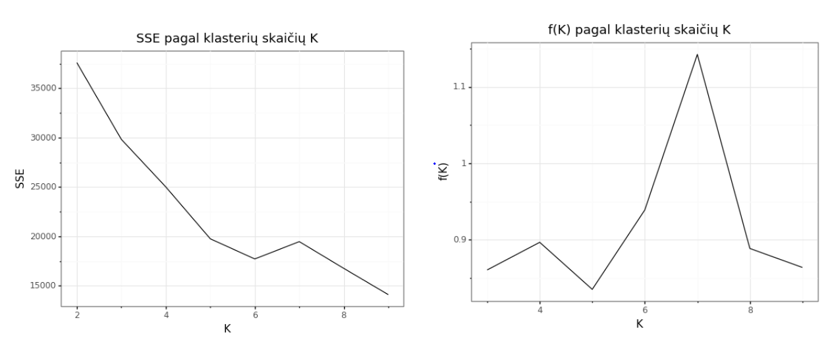


**2 lentelė** Atrinktų duomenų statistinė analizė

Didžiausią „calls\_outgoing\_count“ reikšmę sudaro 44.3 vidurkio reikšmes. Tuo tarpu „sms\_incoming\_count“ maksimali reikšmė vidurkį viršija net 51.74 kartus. Šie duomenys atlikus klasterizavimą padės įvertinti vartotojų įpročius siunčiant ir priimant trumpąsias žinutes (sms), taip pat skambučių įpročius ir su šiomis paslaugomis susijusias išlaidas.

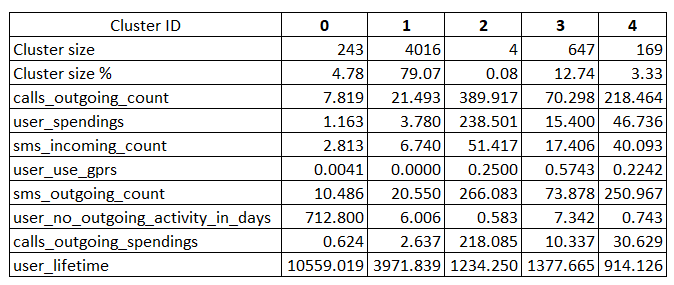
## Duomenų klasterizavimas

Norėdami tiksliai sužinoti, kiek klasterių būtų efektyviausia išskirti su mūsų turimais duomenimis, buvo nuspręsta atlikti SSE bei f(K) metrikų analizę. Pasirinkome abiem atvejam modelius nuo 2 iki 9 klasterių, kad grafikų pagalba būtų lengviau priimti mums labiausiai tinkantį sprendimą. Mūsų atveju abi analizės parodė skirtingus rezultatus (žr. 4 pav.). Pirmuoju atveju, SSE metrikos identifikuoja, kad efektyvu būtų pasirinkti 6 skirtingus klasterius. Visgi, antruoju metodu aiškiai matome, kad 5 klasteriai – efektyviausias sprendimas.



**2 pav**. SSE ir f(K) grafiniai sprendiniai

Turėdami aukščiau pateiktus grafikus pasitarus mūsų grupėje buvo nuspręsta, kad 5 klasteriai visgi būtų efektyviausias ir priimtiniausias sprendimas. Todėl, tolimesniame projekto žingsnyje atliekame vartotojų segmentaciją su 5 klasteriais.



**3 lentelė** Klasteriai ir duomenų centrai

Suskirstyti duomenys yra į 5 klasterius su ID kodais nuo 0 iki 4. Svarbu atkreipti dėmesį į klasterį, kurio ID yra 2. Šis klasteris savyje talpina išskirtis tokias kaip „calls\_outgoing\_spendings“. Matome, kad vidutiniškai kituose klasteriuose išlaidos skambučiams sudaro nuo 0.62 iki 30.63 piniginių vienetų. Visgi, 2-ame klasteryje šių duomenų rinkinio centras yra 218.09. Ta pati situacija pastebima ir su kitu duomenų rinkiniu - „user\_spendings“ (bendros kliento išlaidos). Tuo tarpu klasterio 0 „user\_lifetime“ centras beveik 10 kartų lenkia 2, 3 bei 4 klasterio to pačio duomenų rinkinio centrus.

Šį k-means klasterizavimo modelį išsaugojome Apache Parquet formatu, kad prireikus ateityje, būtų lengvai pasiekiamas visiems grupės nariams. Visi duomenų failai bei aprašai išsaugoti GitHub repozitorijoje.

# Sudarytų klasterių analizė ir prognozavimas

Atlikus duomenų klasterizavimą, buvo pastebėta, kad kelios išimtys buvo priskirtos atskiram klasteriui. Mūsų komanda nusprendė šį klasterį palikti antrosios dalies aprašyme, tačiau tolimesnėje analizėje pastaroji grupė nebus analizuojama. Taigi, toliau nagrinėsime 4 klasterius.

Sukurto failo ,,***Train\_cluster2.py***“ pagalba kiekvieną iš toliau analizuojamų klasterių padaliname į tris imtis: apmokymo, testavimo bei patikrinimo. Šios imtys yra išsaugomos ,,parquet“ tipo formatu (pvz. „sample\_aggregated\_usage\_with\_churn“ kataloge). Labai svarbu paminėti, jog duomenys buvo papildyti keliais kintamaisiais iš pradinių duomenų prieš duomenų imties skaidymo procesą. Galutinai šiai užduočiai naudotų kintamųjų sąrašas yra pateiktas žemiau:

* churn,calls\_outgoing\_spendings,
* calls\_outgoing\_count,user\_spendings,
* sms\_incoming\_count,calls\_outgoing\_duration\_max,
* calls\_outgoing\_spendings\_max,user\_account\_balance\_last,
* sms\_outgoing\_to\_onnet\_count,
* reloads\_inactive\_days,reloads\_count,
* sms\_outgoing\_count,
* sms\_outgoing\_spendings\_max,
* sms\_outgoing\_to\_abroad\_count,
* sms\_outgoing\_spendings,
* gprs\_spendings,
* sms\_incoming\_spendings,
* calls\_outgoing\_to\_onnet\_count,
* gprs\_session\_count,
* user\_no\_outgoing\_activity\_in\_days,
* user\_has\_outgoing\_sms,user\_has\_outgoing\_calls,
* user\_intake,user\_use\_gprs,user\_does\_reload

Kaip ir antroje dalyje, išvardinti kintamieji buvo atrinkti diskusijos metu ir ,,RandomForest“ algoritmo pagalba.

Tolimesniame etape sukuriamas ***FullModel\_TrainSplit.py*** failas. Jis atlieka tuos pačius procesus su duomenimis, kaip ir prieš tai aptartas ***Trian\_cluster2.py***, tačiau šiuo atveju imties padalinimas vykdomas su pilna duomenų imtimi. Šis žingsnis yra ypač svarbus tolimesnės analizės klausimais. Rezultatų palyginimui bei sudaromoms prognozėms interpretuoti pilnos duomenų imties rezultatai padės tiksliau juos įvertinti.

Taigi, analizuojame klasterius 0, 1, 3 bei 4. Primename, kad 2 klasteris yra sudarytas iš išskirčių, kurių nenorėjome trinti, tačiau analizės kontekste jų naudojimas nėra prasmingas, taigi šis klasteris nebus naudojamas tolimesniuose žingsniuose.

Duomenų faile ,,***TrainValidateClusters.py***“ iš Json failo nuskaitome klasterio numerį, tada toliau atitinkamai atliekamas duomenų nuskaitymo procesas iš „parquet“ formatu išsaugotos atitinkamos patikrinimo, mokymosi ir testavimo imties pagal klasterio numerį. Šitame žingsnyje mes išsaugome klasterio duomenų rinkinį po transformavimo operacijos, kad vėliau apjungti su kitais klasteriais ir bendrai visam apjungtam klasterių duomenų rinkiniui paskaičiuoti metrikas. Kaip ir anksčiau, klasifikatorių sudarymui naudojame „RandomForest“ modelį. Analizėje buvo naudoti šie hyperparametrai: numTrees( medžių skaičius) ir maxDepth (gylis). Šių hyperparametrų reikšmės parenkamos cikle.

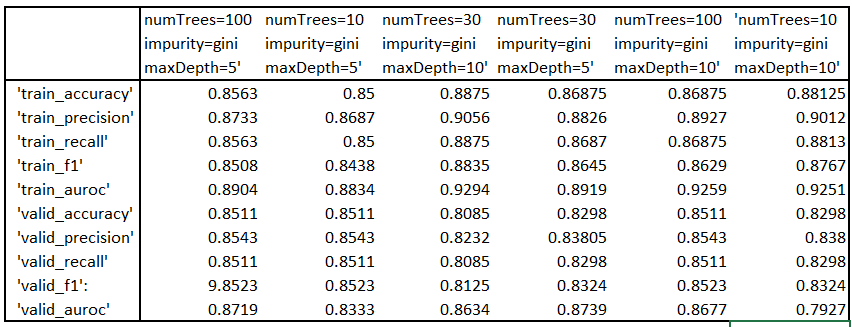
Svarbu atkreipti dėmesį į tai, kad parinkdami parametrus mes apmokome klasifikatorių su mokymosi imtimi, tačiau pačias metrikas tikriname visiems trims duomenų rinkiniams. T.y. testavimo, apmokymo ir patikrinimo imtis.

Duomenų faile ***,,FullModel\_Classification.py“*** atliekami paskutiniai analizės veiksmai. Šio kodo pagalba klasterių duomenų rinkiniai padalinti į mokymo, tikrinimo ir testavimo imtis bei yra įtraukiami į skaičiavimą. Transformuoti klasterių rinkiniai apjungiami į vieną duomenų rinkinį. Pagrindinis tikslas duomenų rinkinių metrikų palyginimas. Įvykdžius tai, apskaičiuojami rezultatai, kurie yra gaunami po galutinio klasifikavimo. Klasteriams buvo paimtas vienas iš variantų kurių parametrai yra tokie : medžių skaičius=30, gylis=10. Atsižvelgus į juos analizuojamu atveju pastebima, kad pradinis klasterizavimas duoda šiek tiek geresnius rezultatus su tais pačiais parametrais. Vietoje mokymosi metrikų vaizduojamos testavimo metrikos, norint pabrėžti kad abiejų metrikų rezultatai panašus.

Duomenų rinkiniai, kuriems atliekamas metrikų skaičiavimas yra išsaugomi atskirai „*parquet“* formato faile. Tai padės tolimesniuose analizės žingsniuose sujungti klasterių rinkinius ir apskaičiuoti klasifikavimo metrikas bendram sujungtam duomenų rinkiniui.

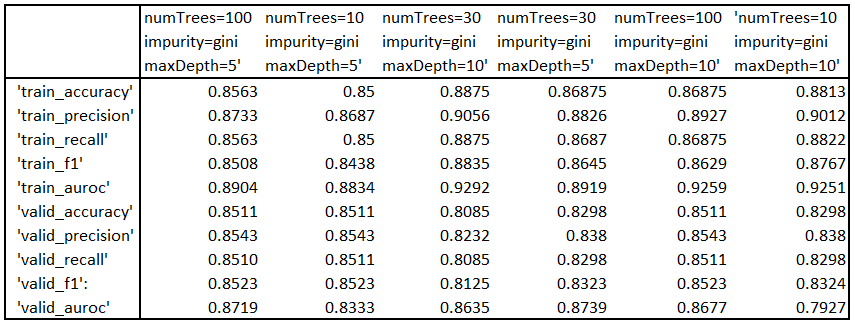
Taigi, atlikus šiuos techninius žingsnius gauname klasifikavimo rezultatus kiekvienam iš keturių analizuojamų duomenų grupių bei bendram duomenų rinkiniui. Šios analizės tikslas - išrinkti geriausius rezultatus turinti modelį, kurio šie parametrai yra aukščiausi:

* “accuracy”
* "precision“
* "recall“
* "f1“
* "auroc“



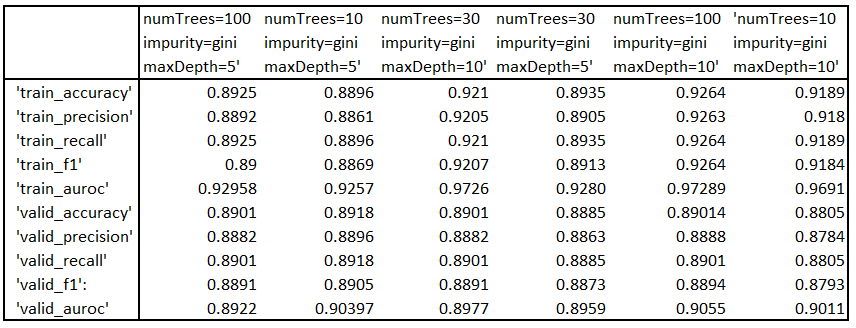
**4 lentelė** Visos pradinės duomenų imties rezultatai

Visu pirma, modelį pritaikome visai duomenų imčiai. Gauti rezultatai yra pateikiami 4 lentelėje. Matome, kad geriausi rezultatai gaunami tuomet, kai medžių skaičius yra lygus 30, o gylis yra lygus 5. Tuomet „valid\_auroc“ parametras yra lygus 0.8739. Tiesa, prasčiausi rezultatai gauti su šiais parametrais: medžių skaičius ir gylis yra lygus 10.



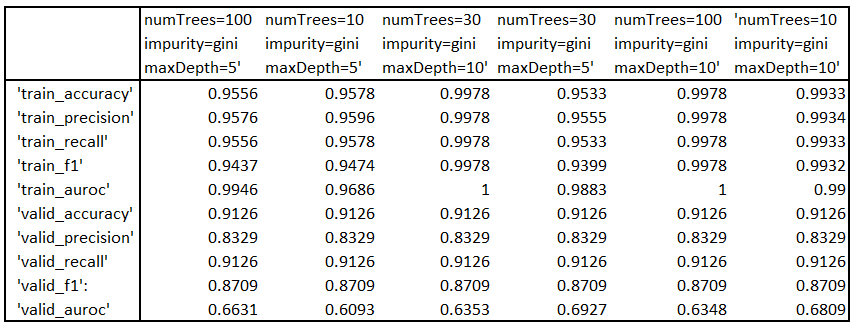
**5 lentelė** Klasterio Nr. 0 rezultatai

Aukščiau pateiktoje lentelėje (žr. 5 lentelė) matome pateiktus modelio rezultatus su 0 klasteriu, keičiant gylio ir medžių parametrus. Pagal ,,valid\_auroc“ rezultatą, matome, kad geriausias modelis yra tas, kuriuo medžių skaičius yra lygus 30, o gylis siekia 5. Šiuo atveju parametras yra lygus 0.8739. Tuo tarpu prasčiausi rezultatai pastebimi, kuomet medžių ir gylio skaičius yra lygus 10. Tuomet parametras yra lygus 0.7927. Vis gi, galima pastebėti, kad ,,valid\_accuracy“, nėra pats aukščiausias, lyginant su kitais bandymais. Tiesa, mūsų komanda nusprendė, kad 0.8298 tikslumas nėra prastas.



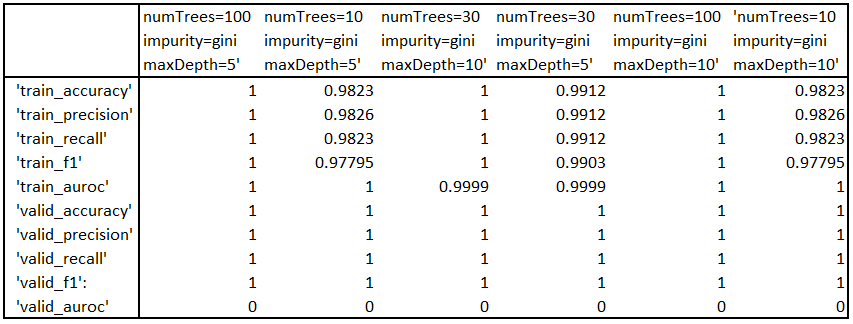
**6 lentelė** Klasterio Nr. 1 rezultatai

Klasterio nr. 1 atveju, pastebime, kad aukščiausias valid\_auroc rodiklis gaunamas su šiais parametrais: medžių skaičius – 100, gylis – 10. Tuomet rezultatas yra lygus 0.9055 (žr. 6 lentelė). Taip pat svarbu pastebėti, kad šiuo atveju, valid\_accuracy parametras yra antras pagal geriausią rezultatą pateikusį. Jis siekia 0.89014. Patys prasčiausi rezultatai užfiksuoti, kuomet medžių skaičius yra lygus 100, o gylis siekia 5. Tuomet valid\_auroc parametras yra lygus 0.8922, o modelio tikslumas siekia 0.8901.



**7 lentelė** Klasterio Nr. 3 rezultatai

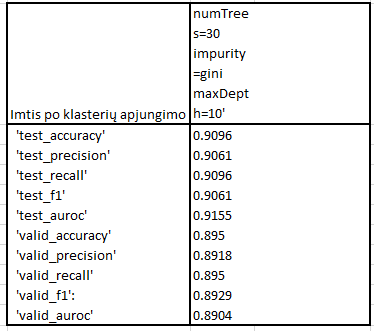
Klasterio nr. 3 rezultatai pateikti 7 lentelėje. Matome, kad train\_auroc siekia keliais atvejais 1. Tuo tarpu valid\_auroc rezultatai gerokai žemesni. Sprendžiant iš šių rezultatų, galima manyti, jog modelį reiktų tobulinti. Visgi, pagal gautus rezultatus, matome, kad geriausias modelis atsižvelgiant į valid\_auroc rezultatą gaunamas tuomet, kai medžių skaičius siekia 30, o gylis 5. Šiuo atveju rezultatas yra lygus 0.6927. Prasčiausi parametrai šiai duomenų imčiai yra šie: medžių skaičius lygus 10, o gylis lygus 5.



**8 lentelė** Klasterio Nr. 4 rezultatai

Aukščiau pateiktoje lentelėje (žr. 8 lentelė) matome pateiktus modelio rezultatus su 4 klasteriu, keičiant gylio ir medžių parametrus. Visgi, matome, kad modelis nebuvo tinkamas šio klasterio naudojimui. Tam įtakos galėjo turėti duomenų imties dydis. Galbūt šį klasterį reikėtų prijungti prie jau esančio ir analizuoti 3 klasterius. Šiuo atveju pasakyti, kurie parametrai buvo geriausi negalime.

Žemiau pavaizduota apjungtų klasterių metrikų lentelė, testavimo ir tikrinimo imčiai.



**9 lentelė**. Rezultatai po klasterių apjungimo

Išvados

1. Atlikus pirminę duomenų analizę bei pasitelkus sprendimų medžio sudarytų duomenų reitingavimu buvo išrinkti 8 mūsų nuomone svarbiausi duomenų rinkiniai, kurie toliau panaudoti klasterių sudarymui;
2. Išanalizavus SSE bei f(K) pateiktus siūlymus buvo priimtas sprendimas duomenų rinkinius klasterizuoti į 5 atskirus klasterius;
3. Atlikus k-means klasterizavimą pastebėta, kad vienas iš klasterių išsiskiria iš kitų dėl ypač mažo duomenų kiekio ir didelių išskirčių. Visgi, daugiau nei 79 proc. duomenų pateko į klasterį pažymėtu 1 ID;
4. Atsižvelgus į galutinius analizės rezultatus pastebime, kad pradinis klasterizavimas duoda šiek tiek geresnius rezultatus prie atitinkamų hiperparametrų. Bet manome, kad būtina peržiūrėti visus variantus, kad išvengti atsitiktinumo;
5. Atlikus atskirų klasterių modelių analizę, pastebėta, kad klasteris Nr. 4 turėtų būti priskirtas prie jau esamų klasterių. Tuo atveju RandomForest metodas galėtų būti pritaikytas šiam klasteriui.

Literatūros sąrašas

1. Han J., Kamber M. Data Mining: Concepts and Techniques (3 ed.). Burlington, MA, USA: Elsevier Science & Technology, 2011.-496 pp. Prieiga per http://site.ebrary.com/lib/ktu/Doc?id=1039930

2. Didžiųjų duomenų rinkinių tyrybos metodai. Studijų medžiaga. Kaunas: KTU, 2019. Prieiga per http://vma.ktu.lt/

3. PySpark programinės įrangos dokumentacija [interaktyvus]. 2019 [žiūrėta 2019-10-11]. Prieiga per: http://spark.apache.org/docs/latest/api/python/index.html

4. Python 3.7 programavimo kalbos dokumentacija [interaktyvus]. 2019 [žiūrėta 2019-10-11]. Prieiga per: https://docs.python.org/3/

5. Exacaster White paper: Essential Guide for Predicting Customer Churn [interaktyvus]. 2019 [žiūrėta 2019-10-11]. Prieiga per: http://www.exacaster.com/whitepaper-essential-guide-predicting-customer-churn/

6. Segmentation for Prepaid: An Exacaster White Paper. 2019 [žiūrėta 2019-10-11]. Prieiga per: http://www.exacaster.com/whitepaper-segmentation-for-prepaid/

7. *Berry M. J., Linoff G. S. Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management (3rd Edition). Hoboken, NJ, USA, Wiley, 2011*. Prieiga per http://site.ebrary.com/lib/ktu/docDetail.action?docID=10513818