Vilniaus Universitetas
Matematikos ir informatikos fakultetas

Fausta Burtyliūtė, Evelina Voleišo, Danielius Lesun

Duomenų mokslas 3 kursas

4-as laboratorinis darbas
Klasifikavimas

Darbo aprašas

Turinys

KNN klasifikavimo modelio taikymas 6
KNN algoritmas – kas tai? 6
KNN algoritmo hyper-parametrai: 6
KNN algoritmo atlikimo etapai: 6
KNN algoritmo privalumai: 7
KNN algoritmo trūkumai: 7
KNN klasifikavimo modelio taikymas 7
Klasifikavimas naudojant sprendimų medžio algoritmą 12
Sprendimų medžio algoritmas (angl. Decision Tree) - kas tai? 12
Pagrindiniai hyper-parametrai 12
Algoritmo etapai 12
Privalumai ir trukūmai 13
Klasifikavimas naudojant atsitiktinių miškų algoritmą 17
Atsitiktinių miškų algoritmas (angl. Random forest) – kas tai? 17
Etapai: 17
Privalumai: 17
Trūkumai: 17
Hiperparametrai 18
Atsitiktinių miškų klasifikavimo modelio taikymas 18
Atsitiktinių miškų klasifikatorius sumažintos dimensijos duomenims 22
Išvados 25
Literatūra 25

Tyrime naudojami duomenys

Failas – data.csv
„realSum“ – bendra „Airbnb“ skelbimo kaina (skaitinis)
„room_type“ – siūlomo kambario tipas (privatus kambarys, bendras, visas kambarys/butas/apartamentai) (kategorinis)
„person_capacity“ – maksimalus galimas apsistojančių gyventojų skaičius (skaitinis)
„host_is_superhost“ – ar šeiminankas yra superšeimininkas (ranginis 0 arba 1)
„multi“ – ar į sąrašą įtraukti keli kambariai (ranginis 0-1)
„biz“ – ar sąraše kambarys yra skirtas verslo tikslams (ranginis 0-1)
„cleanliness_rating“ – švaros įvertinimas (skaitinis)
„guest_satisfaction_overall“ – bendras svečių pasitenkinimo skelbimu įvertinimas (skaitinis)
„bedrooms“ – kambarių skaičius (skaitinis)
„dist“ – atstumas iki centro (skaitinis)
„metro_dist“ – atstumas iki metro (skaitinis)
„attr_index“ – patrauklumo indeksas (skaitinis)
„attr_index_norm“ – sunormuoti patrauklumo indeksai
„rest_index“ – restoranų indeksas (skaitinis)
„rest_index_norm“ – sunormuotas restoranų indeksas (skaitinis)
„lng“ – ilguma/ ilgumos koordinatė (skaitinis)
„lat“ – platuma / platumos koordinatė (skaitinis)

Tyrimo tikslas ir uždaviniai

Tyrimo tikslas – Klasifikuoti turimus originalius duomenis ir sumažintos dimensijos duomenis.
Uždaviniai:

- Pasirinkti klasifikavimui naudojamą kintamąjį su dvejomis klasėmis
- Patikrinti aprašomąją statistiką duomenims ir priklausomo kintamojo klasėms
- Atlikti klasifikavimą su trimis algoritmais
- Pavaizduoti rezultatus ROC kreive
- Apskaičiuoti modelio statistikas

Aprašomoji statistika

Patikriname pradinių duomenų aprašomąją statistiką skaitiniams kintamiesiems.

	realSum	person_capacity	Cleanliness_rating	Guest_satisfaction_overall
Min	69,59	2,00	2,00	20,00
Q1	161,99	2,00	9,00	88,00
Mediana	208,53	2,00	10,00	93,00
Vidurkis	288,39	2,756	9,286	90,93
Q3	335,37	3,00	10,00	97,00
Max	6943,70	6,00	10,00	100,00

Praleistos reikšmės	0	0	0	0

1 lentelė . Aprašomoji statistika.

	Bedrooms	Dist	Metro_dist	Attr_indexx	Atrr_index_norm
Min	0,00	0,1199	0,0130	93,82	3,198
Q1	1,00	1,0906	0,2521	282,77	9,637
Mediana	1,00	1,7518	0,3705	389,20	13,265
Vidurkis	1,217	2,1173	0,4349	464,37	15,827
Q3	1,00	2,9492	0,5542	591,59	20,162
Max	6,00	8,4440	2,4028	2934,13	100,00
Praleistos reikšmės	0	0	0	0	0

2 lentelė . Aprašomoji statistika

	Rest_index	Rest_index_norm	Lng	Lat
Min	159,8	3,518	2,105	41,35
Q1	494,4	10,883	2,156	41,38
Mediana	801,8	17,650	2,171	41,39
Vidurkis	877,7	19,320	2,169	41,39
Q3	1211,3	26,663	2,179	41,40
Max	4542,8	100,00	2,226	41,46
Praleistos reikšmės	0	0	0	0

3 lentelė . Aprašomoji statistika

Taip pat patikriname kategorinius kintamuosius.

	Room_type	Host_is_superhost	multi	Biz
Skalė	Nominalinė	Ranginė	Ranginė	Ranginė
Tipas	Nominalusis	Dvinariai	Dvinariai	Dvinariai
Praleistos reikšmės	0	0	0	0

4 lentelė. Kategorinių duomenų informacija

Pažvelgę į aprašomąją statistiką (žr. 1-4 lentelė), galime pamatyti, kad praleistų reikšmių duomenyse nėra. Taip pat matome, kad duomenyse yra požymių, kurie mūsų tyrimo nepagerintų, tokių kaip sunormuoti patrauklumo ir restoranų indeksai (žr. 2-3 lentelė).Duomenys buvo sunormuoti ir pritaikyti t-SNE, MDS ir PCA algoritmai. Todėl toliau atliksime klasifikavimą su originalia sunormuota duomenų aibe ir su sumažintos dimensijos duomenimis.

Atliksime klasifikavimą lankytojo įvertinimo lygiui pagal kitus duomenų aibės požymius. Lankytojo įvertinimo požymis turi dvi klases. Viena iš jų apima duomenis, kurių įvertinimas siekia iki 90 (klasė – 0) ir virš 90 balų (klasė – 1). Tokiu būdu pasirinkę klasifikatorių, mes nesusiduriame su duomenų išbalansavimo problema.

	realSum	person_ capacity	cleanline ss_rating	bedroo ms	dist	Metro _dist	Attr _index	Rest _index	lng	lat
Kiekis	598,00	598,00	598,00	598,00	598,00	598,00	598,00	598,00	598, 00	598,00
Vidurkis	311,49	2,95	8,60	1,26	2,07	0,42	469,58	883,97	2,17	41,39
Standartinis nuokrypis	464,61	1,40	1,21	0,65	1,29	0,26	264,67	454,44	0,02	0,02
Minimumas	69,59	2,00	2,00	0,00	0,14	0,01	93,82	159,84	2,12	41,35
Mediana	196,90	2,00	9,00	1,00	1,72	0,36	381,59	816,56	2,17	41,39
Maximumas	6943,70	6,00	10,00	6,00	8,44	1,68	2065,07	2608,33	2,23	41,46

5 lentelė. Klasifikatoriaus aprašomoji statistika klasės 0

	realSum	Person _capacity	Cleanliness _rating	bedrooms	dist	Metro _dist	Attr _index	Rest _index	lng	lat
Kiekis	957,00	957,00	957,00	957,00	957,00	957,00	957,00	957,00	957,00	957,00
Vidurkis	273,96	2,63	9,72	1,19	2,15	0,45	461,12	873,72	2,17	41,39
Standartinis nuokrypis	179,83	1,18	0,50	0,51	1,39	0,28	270,66	465,72	0,02	0,02
Minimumas	69,59	2,00	6,00	0,00	0,12	0,01	98,66	167,93	2,11	41,35
Mediana	215,28	2,00	10,00	1,00	1,78	0,38	390,76	792,92	2,17	41,39
Maximumas	1770,66	6,00	10,00	3,00	8,06	2,40	2934,13	4542,75	2,22	41,46

6 lentelė. Klasifikatoriaus aprašomoji statistika klasės 1

Pažvelgę į šias paskutines dvi lenteles matome, kad klasifikatoriaus klasės daugiau ar mažiau viena nuo kitos skiriasi tiek kiekiu, vidurkiu, standartiniu nuokrypiu, minimumu, mediana bei maximumu. Kadangi skirtumas tarp šių klasių yra, galime taikyti įvairius klasifikavimo algoritmus.

Duomenis padalinome į testavimo, mokymo aibes (20%, 80%) Bei atlikome klasifikacijas su KNN, “Random forest” ir “Decision tree” algoritmais. Modelių tinkamumo įvertinimui naudojamos yra šios metrikos:

- **Accuracy:** Tai bendro modelio prognozių teisingumo matas. Jis apskaičiuoja teisingai suklasifikuotų pavyzdžių ir bendro pavyzdžių skaičiaus santykį. Vien tik tikslumo gali nepakakti tais atvejais, kai yra klasių disbalansas arba kai klaidingai teigiamų ir klaidingai neigiamų rezultatų kaina skiriasi.
- **Precision:** Tai metrika, kuria kiekybiškai įvertinamas modelio gebėjimas teisingai nustatyti teigiamus pavyzdžius. Ji apskaičiuoja teisingų teigiamų rezultatų ir teisingų teigiamų rezultatų bei klaidingai teigiamų rezultatų sumos santykį.

- **Recall (Sensitivity or True Positive Rate):** Tai metrika, kuria matuojamas modelio gebėjimas rasti visus teigiamus pavyzdžius. Ji apskaičiuoja teisingų teigiamų pavyzdžių ir teisingų teigiamų bei klaidingai neigiamų pavyzdžių sumos santykį.
- **F1 Score:** F1 balas sujungia precision metriką ir recall į vieną rodiklį, kuris subalansuoja abu rodiklius. Tai yra tikslumo ir atšaukimo harmoninis vidurkis, todėl gaunama viena vertė, rodanti bendrą modelio našumą.

KNN klasifikavimo modelio taikymas

KNN algoritmas – kas tai?

"k-Nearest Neighbors" (kNN) algoritmas yra paprastas, tačiau galingas mašininio mokymosi algoritmas, naudojamas ir klasifikavimo, ir regresijos užduotims atlikti. Tai neparametrinis algoritmas, t. y. jis nedaro jokių prielaidų apie pagrindinį duomenų pasiskirstymą.

Pagrindinė kNN algoritmo idėja - klasifikuoti arba prognozuoti naujo duomenų taško vertę remiantis "k" artimiausių mokymo aibės duomenų taškais.

KNN algoritmo hyper-parametrai:

- **n_neighbors:** nustato kaimynų, į kuriuos reikia atsižvelgti atliekant prognozes, skaičių. Jis atitinka "k" reikšmę kNN išraiškoje. Didesnė "k" vertė atsižvelgia į daugiau kaimynų, o tai gali išlyginti sprendimo ribą, tačiau taip pat gali sukelti daugiau šališkumo. Ir atvirkščiai, esant mažesnei "k" vertei, modelis tampa jautresnis atskiriems duomenų taškams, tačiau taip pat gali padidėti triukšmas.
- **weights:** nustato kiekvienam kaimynui priskiriamą svorį prognozavimo metu.
- **algorithm:** apibrėžia algoritmą, naudojamą artimiausiems kaimynams apskaičiuoti.
- **p**: yra Minkovskio atstumo metrikos galios parametras. Jis naudojamas, kai hiperparametrui "svoriai" nustatyta "atstumo" metrika. Minkovskio atstumas yra kitų atstumo metrikų, pavyzdžiui, Euklido ir Manheteno atstumų, apibendrinimas. Kai "p=1", jis atitinka Manheteno atstumą, o kai "p=2" - Euklido atstumą. Galima naudoti ir kitas "p" reikšmes.

KNN algoritmo atlikimo etapai:

1. Nustatome kaimynų skaičių (k), į kuriuos bus atsižvelgiama atliekant klasifikavimą arba regresiją.
2. Pasirinkame atstumo metriką, pavyzdžiui, Euklido atstumą arba Manheteno atstumą, kad įvertintumėme duomenų taškų panašumą arba nepanašumą. Atstumo metrika nustato, kaip arti ar toli vienas nuo kito yra du duomenų taškai.
3. Apskaičiuojamas atstumas tarp tam tikro nepažymėto duomenų taško (testo atvejo) ir visų pažymėtų mokymo duomenų rinkinio duomenų taškų.
4. Nustatomi k duomenų taškai iš mokymo duomenų rinkinio, kurių atstumai iki testuojamo pavyzdžio yra mažiausi. Šie k artimiausi kaimynai bus naudojami prognozėms atlikti.

KNN algoritmo privalumai:

- **Paprasta ir lengva įgyvendinti:** kNN algoritmą paprasta suprasti ir įgyvendinti. Dėl savo paprastumo jis yra geras pasirinkimas pradedantiesiems mašininio mokymosi specialistams.
- **Jokių prielaidų apie duomenų pasiskirstymą**: kNN yra neparametrinis algoritmas, t. y. jis nedaro jokių prielaidų apie pagrindinį duomenų pasiskirstymą. Jis gali gerai veikti su duomenimis, kurie turi sudėtingus modelius ar pasiskirstymus.
- **Lankstumas apdorojant įvairių tipų duomenis:** kNN gali apdoroti ir skaitinius, ir kategorinius duomenis. Jį galima naudoti klasifikavimo užduotims su diskrečiomis klasių etiketėmis ir regresijos užduotims su tolydžiais tiksliniais kintamaisiais.
- **Prisitaikymas prie kintančių duomenų:** kNN yra egzemplioriais pagrįstas algoritmas, t. y. mokymo metu nesukuriamas aiškus modelis. Vietoj to jis naudoja visą mokymo duomenų rinkinį kaip žinių bazę. Dėl šio pritaikomumo kNN gali lengvai įtraukti naujus duomenų taškus neperkvalifikuodamas modelio.
- **Interpretacijos galimybės:** kNN algoritmo prognozes galima lengvai interpretuoti. Išvestis grindžiama daugumos klasės etikete arba vidutine artimiausių kaimynų verte, todėl aiškiai paaiškinama, kaip gauta prognozė.

KNN algoritmo trūkumai:

- **Skaičiavimo sudėtingumas:** Pagrindinis kNN trūkumas - skaičiavimo sudėtingumas, ypač didelių duomenų rinkinių atveju. Didėjant mokymo aibės dydžiui, algoritmui reikia apskaičiuoti atstumus tarp bandomojo pavyzdžio ir visų mokymo pavyzdžių, o tai gali užimti daug laiko ir atminties.
- **Reikalavimai saugojimui:** Kadangi kNN saugo visą mokymo duomenų rinkinį kaip modelio dalį, jam reikia daug atminties visiems duomenų taškams saugoti. Tai gali būti apribojimas dirbant su didelės dimensijos duomenimis arba duomenų rinkiniais, kuriuose yra daug egzempliorių.
- **Jautrus nereikšmingiems požymiams:** apskaičiuojant atstumus kNN vienodai atsižvelgia į visus požymius. Jei duomenų rinkinyje yra nereikšmingų arba išsiskiriančių požymių, tai gali turėti neigiamos įtakos algoritmo veikimui. Algoritmo tikslumui pagerinti gali prireikti požymių atrankos arba matmenų mažinimo metodų.

KNN klasifikavimo modelio taikymas

Pradiniai duomenys

Visų pirma pritaikėme KNN algoritmą sunormuotiems duomenims su default hyper-parametrais:

- **n_neighbors**: 5
- **weights**: 'uniform'
- **algorithm**: 'auto'
- **p**: 2

Modelio tikslumas „Hold-out“ metodu ~ 72%. Modelio rezultatai su šiais parametrais:

	precision	recall	f1-score	accuracy
0	0.67	0.57	0.62	0.72
1	0.77	0.84	0.80	

7 lentelė. Modelio, su default parametrais, rezultatai.

Tuomet išrinkome geriausius šio algoritmo hyper-parametrus naudojant GridSearchCV. Po šios procedūros gauname:

Geriausi parametrai: {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1, 'weights': 'distance'}

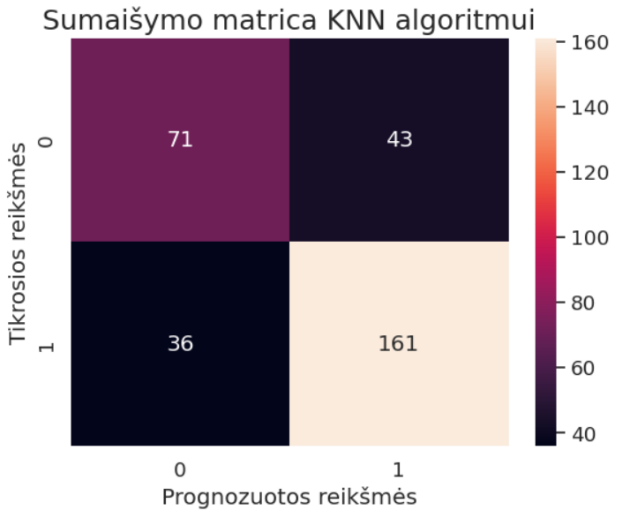
Po šių parametrų pakeitimo gauname tikslumą ~75% su tais pačiais modelio rezultatais:

	precision	recall	f1-score	accuracy
0	0.67	0.62	0.64	0.75
1	0.79	0.82	0.80	0.75

8 lentelė. Modelio, su geriausiais parametrais, rezultatai.

Vidutinis kryžminės validacijos rezultatas naudojant modelį su originalia sunormuota duomenų aibe yra 0,71, kas rodo, kad KNN klasifikatorius su pasirinktais parametrais vidutiniškai teisingai klasifikavo apie 71% testo rinkinio duomenų.

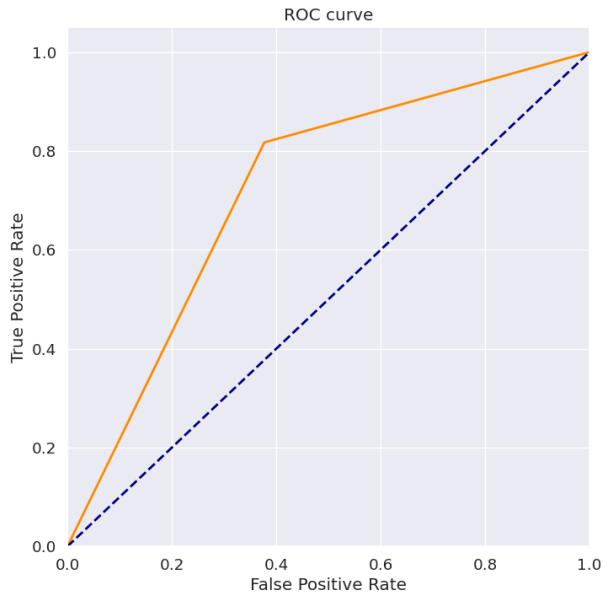
Nubraižę sumaišymo matricą galime dar geriau matyti modelio teisingai ir neteisingai atspėtas reikšmes:



1 pav. Sumaišymo matrica algoritmo prognozėms su geriausiais parametrais

Galime vizualiai matyti, kad modelis gana neblogai suprognozavo reikšmes. Žinoma, tą galėjome pasakyti ir vos sužinoję modelio tikslumą, kuris buvo 71+161/71+36+43+161 = ~0.75

Taip pat nusibraižėme ir ROC tiesę:



2 pav. ROC kreivė algoritmo su geriausiais parametrais įvertinimui

AUC yra vertinimo metrika, kuri matuoja, kaip gerai klasifikatorius gali atskirti teigiamas ir neigiamas reikšmes. Mūsų AUC rezultatas šiuo atveju yra 0.72. Plotas po ROC kreive (AUC-ROC) taip pat paprastai skaičiuojamas kaip apibendrinamoji metrika. Jo reikšmė svyruoja nuo 0,5 (rodo atsitiktinį spėjimą) iki 1,0 (rodo tobulą klasifikatorių). Galime daryti išvadą, kad mūsų klasifiktarius yra vidutinio gerumo.

Sumažintos dimensijos duomenys

Dimensiją mažinome naudodamiesi T-SNE algoritmu, kadangi jis pasirodė geriausiai visuose praeituose mūsų darbuose. Su default KNN algoritmo parametrais gavome ~68% tikslumą. Modelio rezultatai:

	precision	**recall**	**f1-score**	**accuracy**
0	0.56	0.61	0.58	0.68
1	0.7	0.72	0.74	0.68

9 lentelė. Modelio, su default parametrais, rezultatai.

Pasinaudojus GridSearchCV funkcijos pagalba gauname geriausius šiam atvejui hyper-parametrus:

{'algorithm': 'auto', 'n_neighbors': 10, 'p': 2, 'weights': 'distance'}.

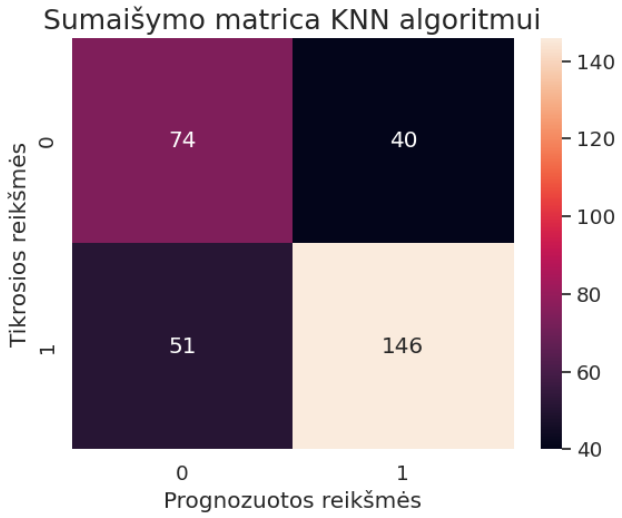
Pakeitę hyper-parametrus gauname jau šiek tiek aukštesnį modelio tikslumą, kuris yra lygus ~71%. Atnaujinto modelio rezultatai:

	precision	**recall**	**f1-score**	**accuracy**
0	0.59	0.65	0.62	0.71
1	0.78	0.74	0.76	0.71

10 lentelė. Modelio, su geriausiais parametrais, rezultatai.

Vidutinis kryžminės validacijos rezultatas naudojant modelį su originalia sunormuota duomenų aibe yra 0,71, kas rodo, kad KNN klasifikatorius su pasirinktais parametrais vidutiniškai teisingai klasifikavo apie 71% testo rinkinio duomenų. Atkreipkime dėmesį, kad šis kryžminės validacijos rezultatas buvo toks pats tik su normuotais duomenimis nemažinant dimensijos.

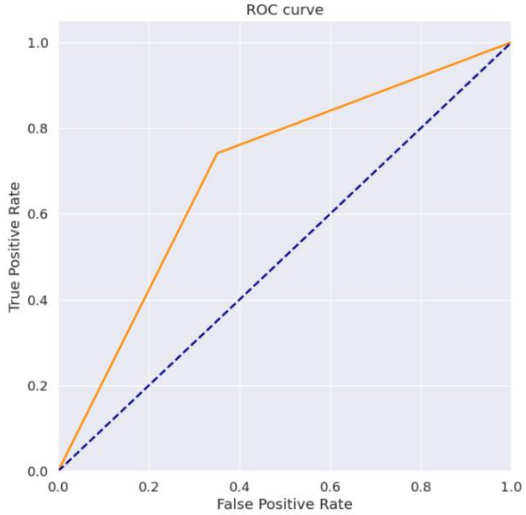
Nubraižę sumaišymo matricą galime dar geriau matyti modelio teisingai ir neteisingai atspėtas reikšmes:



3 pav. Sumaišymo matrica algoritmo prognozėms su geriausiais parametrais

Na, vėl gi iš sumaišymo matricos galime matyti kaip susidaro tas prieš tai minėtas 71% modelio tikslumas. Jį gauname sudėję teisingai atspėtas reikšmes ir padalinus iš visų reikšmių: 220/311 = ~0.71

Taip pat ROC kreivė:



4 pav. ROC kreivė algoritmo su geriausiais parametrais įvertinimui

AUC su sumažintos dimensijos duomenimis yra ~0.69. Pastebėkime, kad ir pati ROC kreivė yra šiek tiek žemiau, negu su sunormuotais duomenimis. Todėl galime sakyti, jog šis klasifikatorius pasirodė prasčiau sumažinus turimų duomenų dimensiją.

Klasifikavimas naudojant sprendimų medžio algoritmą

Sprendimų medžio algoritmas (angl. Decision Tree) - kas tai?

Sprendimų medis yra prižiūrimo mokymosi metodas, kurį galima naudoti ir klasifikavimo, ir regresijos uždaviniams spręsti, tačiau dažniausiai jis pasirenkamas klasifikavimo uždaviniams spręsti. Tai medžio struktūros klasifikatorius, kurio vidiniai mazgai reiškia duomenų rinkinio požymius, šakos - sprendimų priėmimo taisykles, o kiekvienas lapų mazgas - rezultatą.

Pagrindiniai hyper-parametrai

- **Maksimalus gylis (angl. max depth):** Tai nurodo maksimalų galimą sprendimų medžio gylį. Didelis maksimalus gylis gali sukelti perpratimą (angl. overfitting) ir sumažinti sprendimų medžio generalizavimo galimybes, o per mažas gylis gali sukelti nepakankamą sudėtingumą ir sumažinti sprendimų medžio sprendimo tikslumą.
- **Minimalus įrašų skaičius lape (angl. min samples per leaf):** Tai nurodo minimalų galimą įrašų skaičių, kuris turi būti priskirtas lapui. Didelis minimalus skaičius gali sumažinti sprendimų medžio sudėtingumą ir sumažinti perpratimą, o per mažas skaičius atvirkščiai.
- **Minimalus įrašų skaičius mazge (angl. min samples per split):** Tai nurodo minimalų galimą įrašų skaičių, kuris turi būti turimas mazge, kad galėtų būti padalintas į naujus mazgus.
  Kriterijus (angl. criterion): Tai nurodo funkciją, kurią sprendimų medis naudoja, kad nuspręstų, kada nutraukti medžio kūrimą.
- **Maksimalus požymių skaičius (angl. max features) -** nustato, kiek požymių yra galima naudoti. Didesnis skaičius gali leisti modeliui gauti daugiau informacijos.

Algoritmo etapai

Sprendimų medyje, norint nuspėti duoto duomenų rinkinio klasę, algoritmas pradedamas nuo medžio šakninio mazgo. Šis algoritmas palygina šaknies atributo reikšmes su tikrojo duomenų rinkinio atributo reikšmėmis ir, remdamasis palyginimu, seka šaką ir pereina į kitą mazgą.

Kitame mazge algoritmas vėl palygina atributo reikšmę su kitais mazgais ir juda toliau. Procesas tęsiamas tol, kol pasiekiamas medžio lapinis mazgas. Visą procesą galima geriau suprasti naudojant toliau pateiktą algoritmą:

1. Medį pradėkite nuo šakninio mazgo S, kuriame yra visas duomenų rinkinys.

2. Raskite geriausią duomenų rinkinio atributą naudodami atributų atrankos priemonę (ASM).
3. Padalykite S į poaibius, kuriuose yra galimos geriausių atributų reikšmės.
4. Sukurkite sprendimų medžio mazgą, kuriame yra geriausias požymis.
5. 3 žingsnyje sukurtus duomenų rinkinio poaibius naudodami rekursiškai sukurkite naujus sprendimų medžius. Tęskite šį procesą tol, kol pasieksite etapą, kuriame negalėsite toliau klasifikuoti reikšmių ir galutinis mazgas bus lapiniu mazgu.

Privalumai ir trukūmai

Privalumai:

- Sprendimų medžio klasifikatorius yra gana greitas ir efektyvus, nes jis gali lengvai tvarkyti didelius duomenų rinkinius ir reikalauja nedaug resursų.
- Sprendimų medžio klasifikatorius yra gana atsparus triūkšmui ir geba atpažinti net neaiškius ir netiesinius ryšius tarp požymių.

Trukūmai:

- Sprendimų medžio klasifikatorius gali būti linkęs į pernelyg sudėtingus medžius, kurie per daug gerai atitinka mokymo duomenis, tačiau blogai veikia su naujais duomenimis.
- Sprendimų medžio klasifikatorius gali būti jautrus mažiems duomenų rinkinio pakitimams, kurie gali reikalauti naujo medžio apmokymo, kad būtų atnaujinta jo prognozavimo gebėjimai.
- Sprendimų medžio klasifikatorius gali turėti problemų su požymių skalavimu, todėl gali reikėti atlikti kelių kartų bandymų ir sukurti keletą skirtingų medžių, kad būtų pasiektas geriausias rezultatas.

Algoritmo panaudojimas praktikoje

Pradinis modelis

Pradžioje sukūrėme du modelius naudodami originalių ir sumažintos dimensijos duomenų rinkinio mokymosi aibes, tačiau nekeitėme klasifikatoriaus hyper-parametrų. Apskaičiavome modelių tikslumą pasinaudojus ‘sklearn’ bibliotekos tikslumo apskaičiavimo funkcija, kuri parodė, jog mūsų pradiniai modeliai klasifikuoja 72% ir 0.67% teisingų reikšmių.

Optimaliausių hyper-parametrų paieška

Optimaliausių parametrų paieškai pasirinkome kone viena iš populiariausių algoritmų - GridSearchCV.

Kadangi sprendimų medžio algoritmas leidžia modifikuoti daug hyper-parametrų, mes parinkome penkis pagrindinius parametrus, kuriuos aprašėme ankstesniame skyriuje.

Hyper-Parametrai, kurie buvo pateikti GridSearchCV algoritmui:

- Maksimalus gylis (angl. max depth)
- Minimalus įrašų skaičius lape (angl. min samples per leaf)
- Maksimalus funkcijų skaičius (angl. max features)
- Kriterijus (angl. criterion)
- Minimalus įrašų skaičius mazge (angl. min samples per split)

Gavus rezultatus, geriausios hyper-parametrų reikšmės yra:

max depth	min samples per leaf	max features	criterion	min samples per split
5	4	log2	gini	8

11 lentelė. Optimaliausi modelio hyper-parametrai naudojant originalius duomenis.

max depth	min samples per leaf	max features	criterion	min samples per split
5	4	log2	gini	8

12 lentelė. Optimaliausi modelio hyper-parametrai naudojant sumažintos dimensijos duomenis.

Su šiais hyper-parametrais modeliai pagerėjo per 8% ir 1% nuo pradinių modelių, kuriuose buvo naudojami parametrai pagal nutylėjimą. Klasifikavimo kokybės rezultatus aptarsime kartu su kokybės vertinimo metodais.

Klasifikavimo kokybės vertinimo metodai

Kryžminės validacijos metodas

Kad būtų įvertinta šio modelio kokybė, buvo naudojamas kryžminės validacijos metodas, kuris padalijo duomenų rinkinį į penkias lygias dalis ir kiekvieną dalį naudojo kaip testavimo rinkinį, o kitas dalis naudojo kaip apmokymo rinkinį.

Mūsų vidutinis kryžminės validacijos rezultatas naudojant modelį su originalia duomenų aibe yra 0,712, kas rodo, kad sprendimų medžio klasifikatorius su pasirinktais parametrais vidutiniškai teisingai klasifikavo apie 71,2% testo rinkinio duomenų. Tuo tarpu, antrasis modelis su sumažintos dimensijos duomenimis pasirodė prasčiau - 0.67%.

Išlaikymo validacijos metodas

Išlaikymo validacijos metodas yra kitas būdas įvertinti sprendimų medžio klasifikatoriaus kokybę, pagal kurį pirmasis klasifikatorius su pasirinktais parametrais tiksliai klasifikavo apie 69% testavimo rinkinio duomenų. Šis rezultatas skiriasi nuo kryžminės validacijos rezultatų, nes kryžminės validacijos metodas leidžia įvertinti sprendimų medžio kokybę kiekvienam rinkinio dalinimui, o išlaikymo metodas gali būti jautrus tam, kaip duomenys yra padalinti tarp mokymosi ir testavimo rinkinių.

Sumažintos dimensijos aibės klasifikatoriaus rezultatai šiuo metodu yra tokie patys, kaip ir kryžminės validacijos metodu – 0.67% teisingai suklasifikuotų reikšmių.

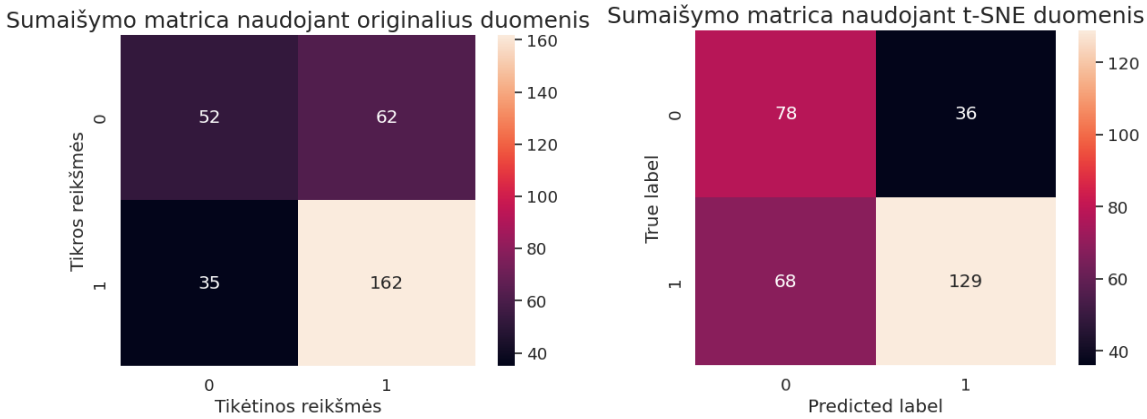
Sumaišymo matrica

Šie rezultatai yra sumaišymo matricos gauti atsakymai Šios matricos tikslas yra įvertinti klasifikatoriaus veikimą, palyginant tikrosųjų reikšmių ir prognozuojamų reikšmių derinius.

Rezultatai rodo, kad pirmasis klasifikatorius, kai naudojami originalus duomenys, turėjo sunkumų atpažįstant nulius: iš 87 tikrųjų nulių reikšmių, tik 52 buvo teisingai suklasifikuoti kaip nuliai. Kitos 35 tikrosios nulių reikšmės buvo klaidingai suklasifikuotos kaip vienetai. Taip pat matome, kad klasifikatorius darė daug klaidų atpažįstant vienetus: iš 224 tikrosios vienetų reikšmės atvejų, tik 162 buvo teisingai suklasifikuoti kaip vienetai, o likę 62 buvo klaidingai suklasifikuoti kaip nuliai.

Tačiau lyginant pirmojo modelio sumaišymo matricos rezultatus su antrojo modelio matricos rezultatais, pirmasis veikė geriau, kai buvo klasifikuojami vienetai, bet blogiau, kai turėjo spėti nulius, tačiau vietoj to prognozavo vienetus.

Taigi, pagal šiuos rezultatus, klasifikatorių veikimas nėra labai geras - yra daug klaidų.



5 pav. Abiejų modelio sumaišymo matricos

Klasifikavimo metrikos

Šie rezultatai yra klasifikacijos modelio kokybės metrikos, skirtos įvertinti modelio tikslumą ir kokybę, naudojant originalius ir sumažintos dimensijos duomenis.

Pagal "Precision" - 60% atvejų, kai modelis spėjo, kad įvykis yra neigiamas (reikšmė 0), tai buvo teisinga prognozė, o 40% atvejų buvo klaidingos prognozės. Dar mažiau teisingų nulių buvo prognozuota naudojant sumažintos dimensijos duomenis. Tačiau buvo prognozuota daugiau teisingų vienetų nei su originaliais duomenimis.

Panaudojus originalų duomenų rinkinį, "Recall" vertė rodo, kad 46% atvejų, kai reikšmė iš tikrųjų yra 0, modelis teisingai ją aptiko, o 54% atvejų buvo klaidingos prognozės. Originalios ir sumažintos dimensijos aibių rezultatai šioje dalyje gerokai skiriasi.

Taip pat, "F1-score" svyruoja nuo 52% iki 77% , kas reiškia, kad modelio rezultatai yra vidutiniškai patenkinami.

"Accuracy" yra bendra modelio kokybės metrika, kuri rodo, kiek procentų įvykių buvo teisingai klasifikuoti. Šiuo atveju tikslumas yra 69%, kas reiškia, kad pirmasis modelis klasifikuoja teisingai tik 69% visų įvykių. Antrasis veikia nežymiai prasčiau, jo tikslumas yra 0.67%

Taigi, bendrai vertinant, modelių kokybė nėra gera, nes yra daug klaidingų prognozių ir tikslumas yra vidutinis.

	precision	recall	f1-score	accuracy
0(114)	0.60	0.46	0.52	0.69
1(197)	0.72	0.82	0.77	

13 lentelė. Modelio rezultatai naudojant originalius duomenis.

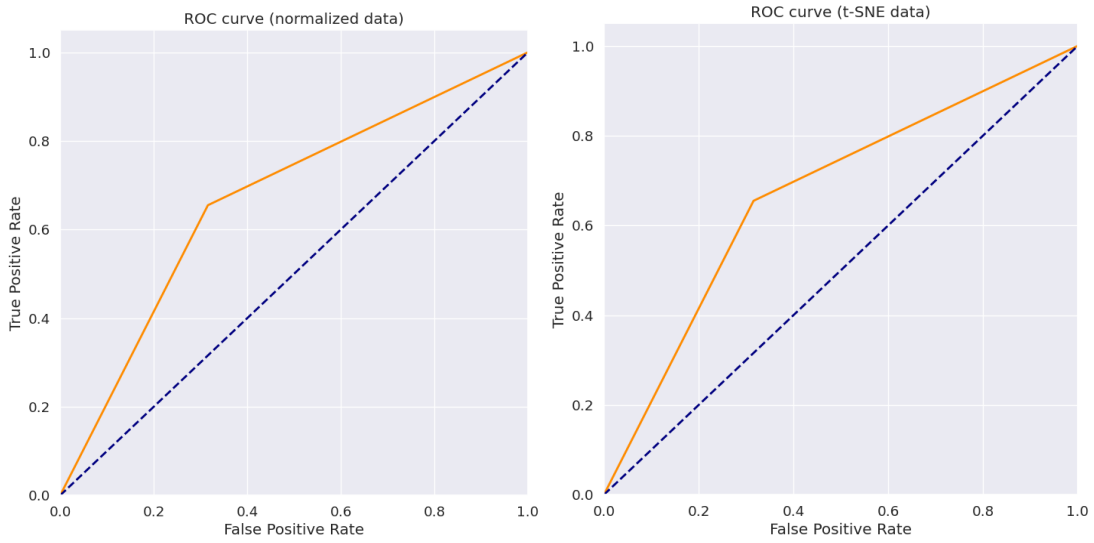
	precision	recall	f1-score	accuracy
0(114)	0.53	0.68	0.60	0.67
1(197)	0.78	0.65	0.71	

14 lentelė. Modelio rezultatai naudojant sumažintos dimensijos duomenis.

ROC kreivė

Pagal grafinį atvaizdavimą, tai yra ROC kreivė, kuri leidžia vizualiai įvertinti klasifikatorius, galime spręsti kaip klasifikatoriaus prognozės kinta priklausomai nuo sprendimo ribos.

AUC yra vertinimo metrika, kuri matuoja, kaip gerai klasifikatorius gali atskirti teigiamas neigiamas reikšmes. Mūsų AUC rezultatas yra 0,6695 abiems atvejais, kai naudojame klasifikatorių normuotiems duomenims bei sumažintos dimensijos duomenims. Reiškia, kad Klasifikatorius turi vidutinį atskyrimo gebėjimą tarp dviejų klasių. Rezultatai patenkinami, tačiau nepakankamai aukšti, kad laikyti modelius pasisekusiais.



6 pav. Modelių ROC kreivių grafikai

Apibendrinant gautus klasifikavimo algoritmo modelius, galime teigti, jog sprendimų medžio algoritmas neparodė labai prastų rezultatų, tačiau gauti modelių tikslumai buvo ne iš geriausių. Taip pat, prasčiau veikė su sumažintos dimensijos duomenimis.

Klasifikavimas naudojant atsitiktinių miškų algoritmą

Atsitiktinių miškų algoritmas (angl. Random forest) – kas tai?

Atsitiktinis miškas - tai klasifikatorius, kuriame yra keletas sprendimų medžių, naudojamų įvairiems duoto duomenų rinkinio poaibiams, ir imamas vidurkis, siekiant pagerinti to duomenų rinkinio prognozavimo tikslumą. Užuot rėmęsis vienu sprendimų medžiu, atsitiktinių sprendimų miškas ima kiekvieno medžio prognozę ir, remdamasis prognozių balsų dauguma, prognozuoja galutinį rezultatą.

Paprasčiau tariant, atsitiktinių sprendimų miškas sukuria kelis sprendimų medžius ir juos sujungia, kad gautų tikslesnę ir stabilesnę prognozę.

Etapai:

Atsitiktinis miškas veikia dviem etapais: pirmiausia sukuriamas atsitiktinis miškas, sujungiant N sprendimų medžių, o antra - atliekamas kiekvieno pirmajame etape sukurto medžio prognozavimas.
1) iš mokymo aibės atsitiktinai atrenkama K duomenų taškų.
2) sukuriami sprendimų medžiai, susiję su atrinktais duomenų taškais (poaibiais).
3) pasirenkamas sprendimų medžių skaičius N.
4) pakartojami 1 ir 2 etapai.
5) naujiems duomenų taškams surandamas kiekviena sprendimų medžio prognozė ir priskiriami nauji duomenų taškai kategorijai, kuri laimėjo daugumą balsų.

Privalumai:

- Padidina modelio tikslumą ir užkerta kelią perteklinio prisitaikymo (persimokymo) problemai, o tai reiškia, kad jis gali gerai apibendrinti naujus duomenis.
- Atsitiktinis miškas yra patikimas algoritmas, galintis apdoroti triukšmingus duomenis ir nukrypimus.
- Vienas tiksliausių mašininio mokymosi algoritmų. Jis gali spręsti ir klasifikavimo, ir regresijos uždavinius ir gali gerai dirbti tiek su kategoriniais, tiek su tęstiniais kintamaisiais.
- Nors atsitiktinis miškas yra sudėtingas algoritmas, tačiau jis yra greitas ir gali apdoroti didelius duomenų rinkinius. Jį taip pat galima lengvai lygia gretinti, kad pagreitėtų mokymas.

- Atsitiktinio miško" metodas: "Random Forest" pateikia požymių svarbos matą, kuri gali padėti atrinkti požymius ir suprasti duomenis.

Trūkumai:

- Didesniam tikslumui reikia daugiau medžių, o didelis medžių skaičius lėtina modelį;
- Nors atsitiktinis miškas yra mažiau linkęs į perteklinį pritaikymą nei pavienis sprendimų medis, jis vis tiek gali per daug pritaikyti duomenis, jei medžių skaičius miške yra per didelis arba jei medžiai yra per gilūs.
- Atsitiktinis miškas gali būti mažiau aiškus nei vienas sprendimų medis, nes jame naudojami keli medžiai. Gali būti sunku suprasti, kaip algoritmas nustatė konkrečią prognozę.

- Mokymo trukmė gali būti ilgesnė nei kitų algoritmų. Ypač jei medžių skaičius ir jų gylis yra dideli.
- Atsitiktiniam miškui reikia daugiau atminties nei kitiems algoritmams, nes jame saugomi keli medžiai. Tai gali būti problema, jei duomenų rinkinys yra didelis.

Hiperparametrai

- Medžių skaičius (angl. n_estimators). Didesnis medžių skaičius miške lemia didesnį tikslumą ir padeda išvengti perteklinio pritaikymo problemos, tačiau didesnis medžių skaičius apima didesnį apmokymo laiką ir išnaudojimą.
- Maksimalus gylis (angl. max depth). Didesnis gylis gali leisti modeliui geriau pritaikyti duomenis klasifikacijai, tačiau per didelis gylis gali sukelti persimokymo problemų.
- Minimalus įrašų skaičius mazge (angl. min samples per split)
- Minimalus įrašų skaičius lape (angl. min samples per leaf)
- Maksimalus funkcijų skaičius (angl. max features)
- Atrankos metodas (angl.bootstrap) kuriu pasirenkama ar algoritmas bus atliktas su pakeitimu arba be pakeitimo.

Atsitiktinių miškų klasifikavimo modelio taikymas

Visų pirma pritaikėme atsitiktinių miškų klasifikavimo modelį sunormuotai originaliai duomenų aibei, su numatytaisiais parametrais:

n_estimator: 100
min_samples_split: 2
min_samples_leaf: 1
max_features: „sqrt“
max_depth: None
bootstrap: True

Modelio tikslumą „Hold-out“ metodu gavome ~76.85 %.

Pradinio modelio rezultatai su numatytaisiais parametrais:

Accuracy	Precision	Recall	F1
76.85%	76.69%	76.85%	76.78%

15 lentelė. Pradinio modelio rezultatai

Atlikome optimalių parametrų paiešką su atsitiktine paieška (angl. Random search), naudojant 5 kartus kryžminę validaciją (cross validation = 5).

Atsitiktinė paieška - tai metodas, kai atsitiktinai parenkami hiperparametrų deriniai ir naudojami modeliui apmokyti. Naudojami geriausi atsitiktiniai hiperparametrų deriniai. Atsitiktinė paieška šiek tiek

panaši į paiešką tinklelyje (angl. grid search), tačiau pagrindinis skirtumas yra tas, kad nenurodome kiekvieno hiperparametro galimų reikšmių rinkinio. Vietoj to imame kiekvieno hiperparametro reikšmes iš statistinio pasiskirstymo.

Šis metodas leidžia kontroliuoti bandomų hiperparametrų derinių skaičių. Skirtingai nuo tinklelio paieškos, kai bandomi visi įmanomi deriniai, atsitiktinė paieška leidžia mums nurodyti treniruojamų modelių skaičių. Paieškos iteracijas galime pagrįsti savo skaičiavimo ištekliais arba vienos iteracijos trukme.

Po šio metodo pritaikymo gauname optimalius parametrus :

n_estimator: 1600
min_samples_split: 10
min_samples_leaf: 1
max_features: „sqrt“
max_depth: 20
bootstrap: True

Pritaikę atsitiktinių miškų modelį su optimaliais parametrais remiantis atsitiktine paieška modelio tikslumą „Hold-out“ metodu gauname tikslumą ~78.14%.

Accuracy	Precision	Recall	F1
78.14%	77.81%	78.14%	77.85%

16 lentelė. Po atsitiktinės paieškos modelio rezultatai su optimaliais parametrais

Pritaikius penkis kartus kryžminę validaciją tikslumo vidurkis gaunasi ~80.04%

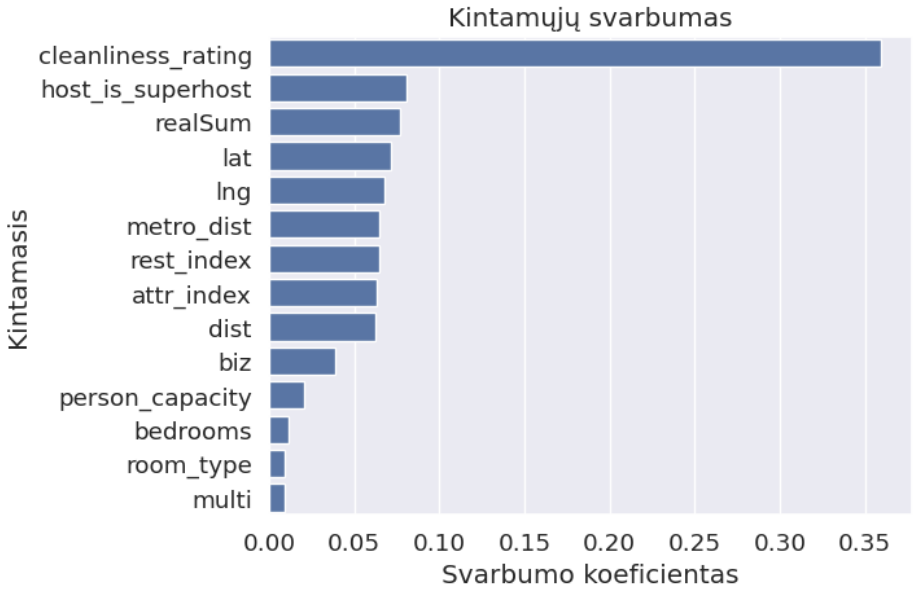
Apskaičiavome modelio statistikas priklausomai nuo klasifikatoriaus reikšmės:

	Accuracy	Precision	Recall	F1
0	0,77	0,69	0,66	0,68
1	0,77	0,81	0,83	0,82

17 lentelė. Po atsitiktinės paieškos modelio rezultatai su optimaliais parametrais klasėms atskirai

Galime pamatyti, kad geriau atpažįsta ir suklasifikuoja duomenis kurie patenka į “1” klasifikatoriaus reikšmę. Kitaip tarant geriau suklasifikuoja duomenis, kurie rodo, kad lankytojo įvertinimas yra didesnis nei 90%.

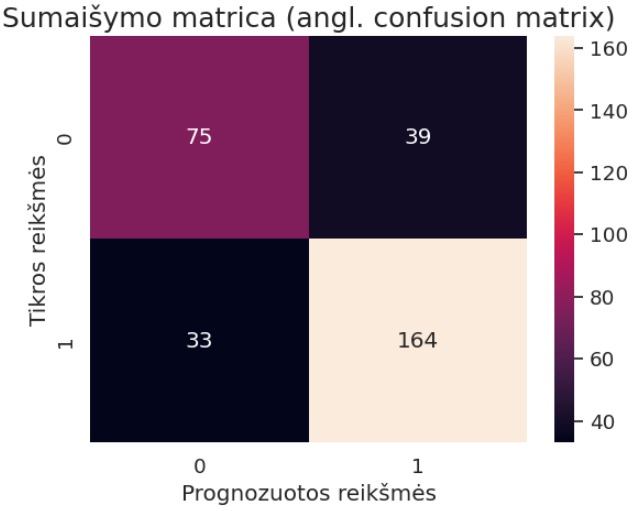
Galime pažiūrėti, kurie kintamieji klasifikavimui turėjo didžiausią įtaką ir kokie požymiai atliekant klasifikavimą yra svarbiausi mūsų duomenų aibėje.



7 pav. Kintamųjų reikšmingumas modelyje

Matome, kad švaros įvertinimas yra svarbiausias požymis klasifikuojant duomenis. O mažiausiai svarbus ir lemiantis klasifikatoriaus rezultatą yra požymis, kuris pasako ar į "Airbnb" sąrašą pateikti keli kambariai (1) ar ne (0).

Sumaišymo matrica parodo, kaip klasifikatorius klasifikuoja duomenis, palyginti su jų tikromis klasėmis. Matrica yra sudaryta iš keturių skaičių, kurie atitinka skirtingas klasifikavimo situacijas: true positives (TP), false positives (FP), true negatives (TN) ir false negatives (FN).

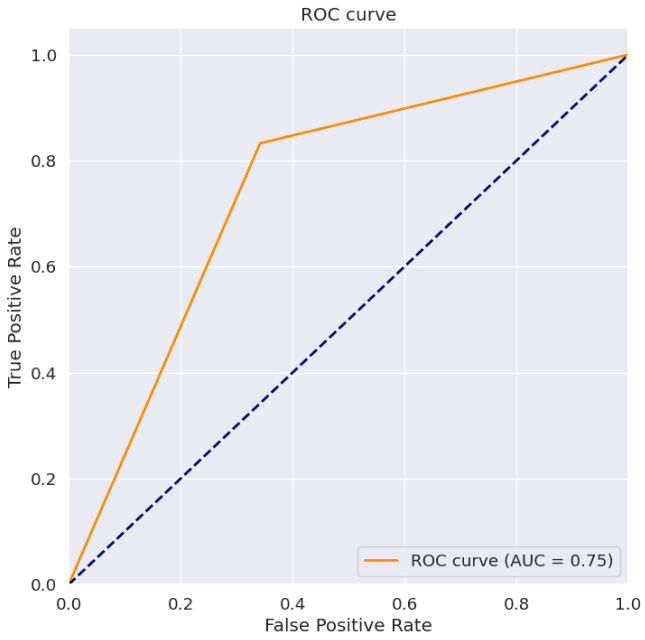


8 pav. Sumaišymo matrica

Matome, kad daugiau klasifikuoja teisingas reikšmes, tačiau taip pat pastebime, kad ne mažas skaičius ir klaidingai klasifikuotų reikšmių. Geriausiai tikrosios reikšmės atspindi prognozes, kai lankytojo

įvertinimas yra virš 90%, o daugiausia suklydimų yra kai tikroji reikšmė sako, kad lankytojas “Airbnb” kambarį įvertino mažiau nei 90%, tačiau prognozuoja kitaip.

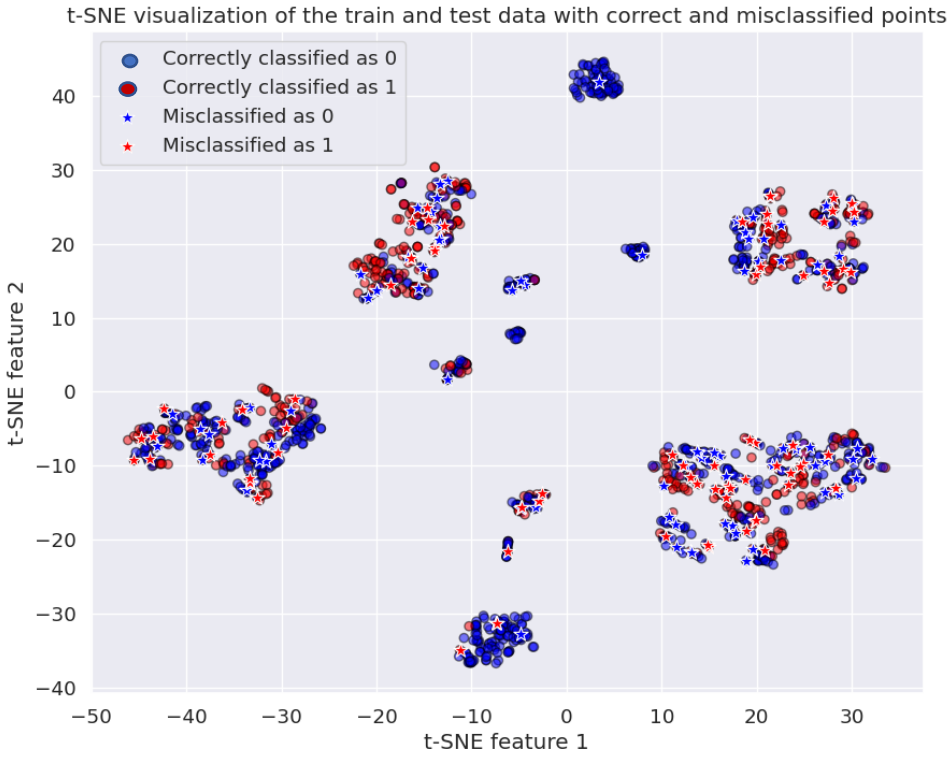
Rezultatams įvertinti, yra braižoma ROC kreivė. Ji rodo, kaip klasifikatorius kinta priklausomai nuo sprendimo ribos, nusakančios, kada duomenys yra priskiriami teigiamai arba neigiamai klasei. AUC yra plotas po kreive, kuris parodo, kiek klasifikatorius tiksliai klasifikuoja.



9 pav. Modelių ROC kreivė

Šiuo atveju AUC yra lygi 0,75. Tai atitinka klasifikavimo tikslumą. Žinome, kad klasifikatorius neprognozuoja reikšmių idealiai, tą parodo ir AUC skaičius.

Taip pat sumažinome dimensiją su šiais klasifikuotais duomenimis, kad galėtume atvaizduoti kaip pasiskirstė klasifikavimo rezultatai. (žr. 10 pav.)



10 pav. Klasifikuotų reikšmių plokštumoje vizualizacija

Atsitiktinių miškų klasifikatorius sumažintos dimensijos duomenims

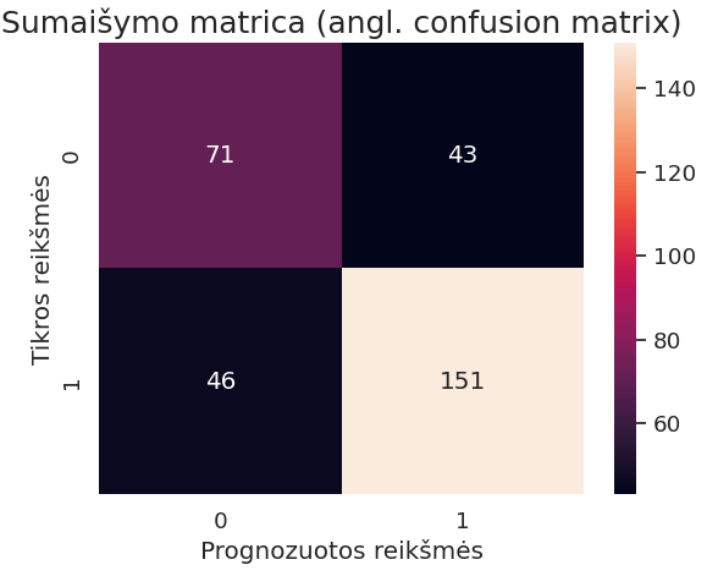
Dabar atliekame tą patį klasifikavimo modelį tik su duomenimis, kurie yra sumažintos dimensijos iki 2.

Su numatytaisiais parametrais „Hold-out“ metodu gauname tikslumą ~69.13 %.

Pritaikę optimalių parametrų paiešką, gavome tokius parametrus:

N_estimators: 1600
min_samples_split: 5
min_samples_leaf: 1
max_features: ‚auto‘
max_depth: 10
bootstrap: True

pritaikę modelį su tokiais parametrais, gavome tiksluma „hold-out“ metodu ~71.38%, matome, kad tikslumas nepakito lyginant su pradiniais parametrais. Tačiau rezultatai yra blogesni lyginant su originalios duomenų aibės klasifikavimo rezultatais.



11 pav. Sumaišymo matrica

Iš sumaišymo matricos, galime pastebėti, kad pamažėjo gerai suklasifikuotų reikšmių ir padaugėjo blogai klasifikuotų reikšmių.

Klasifikavimo statistika taip pat truputį pasikeitė į blogąją pusę.

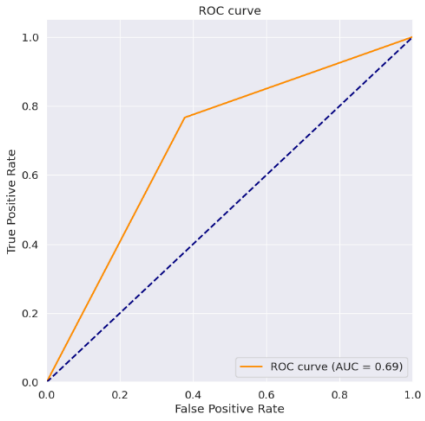
Accuracy	Precision	Recall	F1
71.38%	71.55%	71.38%	71.46%

18 lentelė. Po atsitiktinės paieškos modelio rezultatai su optimaliais parametrais sumažinus dimensiją

Penkių kryžminės validacijos būdu tikslumo vidurkis ~70.74%. Geresnio rezultato nerodo.

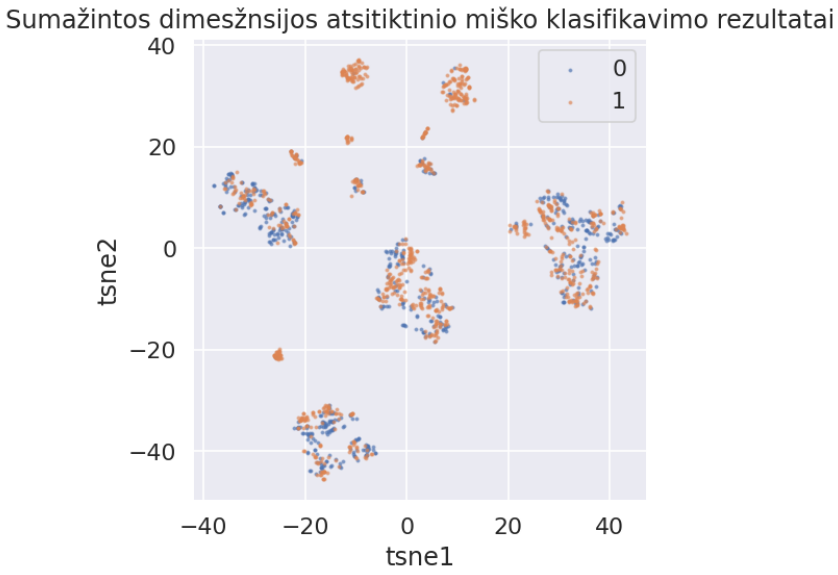
Svarbumo koeficientai tsne1 dimensijai lygus 0,44, o tsne2 dimensijai lygus 0,56. Tai reiškia, kad abi dimensijos yra panašiai svarbios klasifikuojant duomenis, tačiau antra t.y. tsne2 duoda daugiau informacijos klasifikavimui.

ROC kreivė atrodo panašiai, tačiau iš AUC reikšmės matome, kaip ir jau įsitikinome, kad atsitiktinių miškų klasifikatorius sumažintos dimensijos duomenis klasifikuoja prasčiau.



12 pav. Modelių ROC kreivių grafikai

Taip pat nusibraižėme Atsitiktinio miško klasifikuotas reikšmes sumažinus dimensiją su tSNE algoritmu.(žr. 13 pav.)



13 pav. Klasifikuotos reikšmės

Apibendrinant galima teigti, kad atsitiktinio miško algoritmas yra galingas, galintis pateikti tikslesnes klasifikavimo prognozes. Tačiau jis turi tam tikrų apribojimų, susijusių su interpretavimo galimybėmis, pertekliniu pritaikymu, mokymo trukme ir atminties naudojimu.

Išvados

Sėkmingai pasirinkome priklausomą kintamąjį su dvejomis klasėmis. Aprašomojoje statistikoje apie klases pastebėjome, kad jos yra pakankamai panašių dydžių, skirtingų požymių vidurkiai klasėse skiriasi. Pavyzdžiui pirmoji klasė turi aukštesnį švaros įvertinimą, o nulinė klasė turi žemesnį. Atlikome klasifikaciją su trimis skirtingais algoritmais. Rezultatus gavome panašius, tačiau geriausias, bet tuo pačiu metu daugiausiai laiko ir atminties užimantis algoritmas yra atsitiktinio miško. Klasifikatoriaus tikslumas siekė ~78.14%. Visi klasifikatoriai labiausiai klydo prognozuojant 0 klasę.

Literatūra

Hassanat, Ahmad Basheer, et al. "Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach." *arXiv preprint arXiv:1409.0919* (2014).

Liao, Yihua, and V. Rao Vemuri. "Use of k-nearest neighbor classifier for intrusion detection." *Computers & security* 21.5 (2002): 439-448.

Swain, Philip H., and Hans Hauska. "The decision tree classifier: Design and potential." *IEEE Transactions on Geoscience Electronics* 15.3 (1977): 142-147.

Priyanka, and Dharmender Kumar. "Decision tree classifier: a detailed survey." *International Journal of Information and Decision Sciences* 12.3 (2020): 246-269.

Chaudhary, Archana, Savita Kolhe, and Raj Kamal. "An improved random forest classifier for multi-class classification." *Information Processing in Agriculture* 3.4 (2016): 215-222.

Paul, Angshuman, et al. "Improved random forest for classification." *IEEE Transactions on Image Processing* 27.8 (2018): 4012-4024.