VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

Duomenų tyryba. Klasifikavimas.

Savarankiško darbo ataskaita.

Atliko: Justinas Rimavičius, Edvardas Ražanskas

VU el. p.: edvardas.razanskas@mif.stud.vu.lt, justinas.rimavicius@mif.stud.vu.lt

Vertino: dr. Jolita Bernatavičienė

Vilnius
2024

TURINYS

1. ĮVADAS

1.1. Darbo tikslas ir uždaviniai

Šio darbo tikslas – SDSS (Sloan‘o skaitmeninio dangaus tyrimo DR17) duomenų imčiai su pilnu ir atrinktų požymių rinkiniu, bei tos paties duomenų imties sumažintos dimensijos duomenų rinkiniui pritaikyti „kNN“ ir „RandomForest“ klasifikavimo algoritmus, palyginti jų rezultatus, apibrėžti susidariusių klasterių specifiką.

Darbo uždaviniai:

1. Trumpai aprašyti tiriamą duomenų aibę, jos požymius, pagrindines savybes.
2. Pasirinkti ir pagrįsti pagal kokius požymius bus atliekamas klasterizavimas.
3. Sumažinti duomenų dimensijas iki 2 naudojant „PCA“ metodą.
4. Klasifikuoti duomenis naudojant „kNN“ ir „RandomForest“ klasifikavimo algoritmus.
5. Patikrinti kiekvieno algoritmo tikslumą ir gebėjimą atskirti klases naudojant „K-Fold“, „ROC“ metodus ir „AUC“ metriką.
6. Palyginti klasifikavimo algoritmus, apibendrinti rezultatus, pateikti jų interpretaciją.

1.2. Darbo įrankiai

Duomenų apdorojimas, transformacija, analizė, dimensijų mažinimo metodai ir klasterizavimo metodai buvo pritaikyti naudojant „Python 3.12.0” programavimo kalbą ir jos bibliotekas (daugiau žiūrėti skyrių Kodas).

2. DUOMENŲ ANALIZĖ

2.1. Tiriamos duomenų aibės ir jos požymių aprašymas

Pateiktoje žvaigždžių klasifikacijos duomenų aibėje („Stellar Classification Dataset“) yra 100000 eilučių, 18 požymių stulpelių. Jutiklių matavimai yra „float“ tipo (t.y. priklauso realiųjų skaičių aibei) , „class“ požymis yra „object“ tipo (t.y. simboliai), likę požymiai yra „int“ tipo (t.y. priklauso sveikųjų skaičių aibei).

```
Data columns (total 18 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   obj_ID       100000 non-null  float64
 1   alpha        100000 non-null  float64
 2   delta        100000 non-null  float64
 3   u            100000 non-null  float64
 4   g            100000 non-null  float64
 5   r            100000 non-null  float64
 6   i            100000 non-null  float64
 7   z            100000 non-null  float64
 8   run_ID       100000 non-null  int64
 9   rerun_ID     100000 non-null  int64
 10  cam_col      100000 non-null  int64
 11  field_ID     100000 non-null  int64
 12  spec_obj_ID  100000 non-null  float64
 13  class        100000 non-null  object
 14  redshift     100000 non-null  float64
 15  plate        100000 non-null  int64
 16  MJD          100000 non-null  int64
 17  fiber_ID     100000 non-null  int64
dtypes: float64(10), int64(7), object(1)
```

2.1 pav. pradinė duomenų aibė

Duomenų aibės požymių aprašymai:

- obj_ID = objekto identifikatorius, unikali dangaus kūno vertė, identifikuojanti objektą CAS naudojamame vaizdų kataloge.
- alpha = dešiniojo pakilimo kampas (pagal J2000 epochą)
- delta = deklinacijos kampas (pagal J2000 epochą)
- u = ultravioletinis astrofotometrinės sistemos filtras
- g = žaliasis astrofotometrinės sistemos filtras
- r = raudonasis astrofotometrinės sistemos filtras
- i = artimųjų infraraudonųjų spindulių filtras astrofotometrinė sistemoje

- z = infraraudonųjų spindulių filtras astrofotometrinė sistemoje
- run_ID = serijos numeris, naudojamas konkrečiam nuskaitymui identifikuoti
- rereun_ID = pakartotinio paleidimo numeris, nurodantis, kaip vaizdas buvo apdorotas
- cam_col = kameros stulpelis, skirtas skenavimo linijai nustatyti
- field_ID = lauko numeris kiekvienam laukui identifikuoti
- spec_obj_ID = unikalus optinių spektroskopinių objektų ID (tai reiškia, kad 2 skirtingi stebėjimai su tuo pačiu spec_obj_ID turi turėti bendrą išvesties klasę)
- class = objekto klasė (galaktika, žvaigždė arba kvazaras)
- redshift (raudonasis poslinkis) = raudonojo poslinkio vertė, pagrįsta bangos ilgio padidėjimu
- plate = plokštės ID, identifikuojantis kiekvieną SDSS plokštę
- MJD = modifikuota Julijaus data, naudojama nurodyti, kada buvo paimta tam tikra SDSS duomenų dalis
- fiber_ID = pluošto ID, identifikuojantis pluoštą, kuris nukreipė šviesą į židinio plokštumą kiekvieno stebėjimo metu

2.2. Požymių ir objektų apdorojimas

Pašalinti šie požymiai, nedarantys įtakos kosminio kūno klasifikavimui:

- „obj_ID“ požymis, nes tai identifikacinis numeris nedarantis įtakos duomenims;
- „alpha“ ir „delta“ požymiai nusako kosminio objekto poziciją, o jos nėra susijusios su skirtingų objektų (galaktikų, žvaigždžių, kvazarų) fizinėmis savybėmis;
- „spec_obj_ID“ požymis, nes 2 skirtingi stebėjimai su tuo pačiu spec_obj_ID turi turėti bendrą išvesties klasę, o visos šio požymio reikšmės yra skirtingos;
- „rerun_ID“ požymis, nes yra tik viena unikali reikšmė;
- „MJD“ požymis, nes ji simbolizuoja datą, kada užfiksuotas stebėjimas

Duomenų aibė neturėjo praleistų reikšmių. Tolimesniems uždaviniams pasirinkome „redshift“, „u“, „g“, „r“, „i“ ir „z“ požymius.

Duomenų aibė turėjo vieną eilutę, kurioje „u“, „g“ ir „z“ reikšmės buvo -9999, tad šią triukšmo eilutę panaikinome.

„Class“ požymis yra kategorinis požymis, kuris turi tris unikalias reikšmes duomenų aibėje: GALAXY – galaktika, QSO – kvazaras(ypač šviesus objektas galaktikos centre), STAR – žvaigždė. Kiekviena šių reikšmių buvo pakeista atitinkamai į skaičius 0, 1, 2.

2.3. Objektų atrinkimas

Tolimesniam duomenų analizavimui, apdorojimui ir vizualizavimui buvo atsitiktinai atrinkti 3750 objektų, 0 ir 1 klasėms po 1875 objektų(2.2 pav.):

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3750 entries, 0 to 3749
Data columns (total 12 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   u         3750 non-null   float64
 1   g         3750 non-null   float64
 2   r         3750 non-null   float64
 3   i         3750 non-null   float64
 4   z         3750 non-null   float64
 5   run_ID    3750 non-null   int64
 6   cam_col   3750 non-null   int64
 7   field_ID  3750 non-null   int64
 8   class     3750 non-null   int64
 9   redshift  3750 non-null   float64
 10  plate     3750 non-null   int64
 11  fiber_ID  3750 non-null   int64
dtypes: float64(6), int64(6)
memory usage: 351.7 KB
```

2.2 pav. duomenų aibė su pasirinktais objektais

Pasirinkome tik 2 klases, kad ka as zina utu paaiskink blet.

Vėliau šie duomenys bus paskirstyti į mokymo(80%) ir validavimo(20%). Tada nuo atrinktų mokymo(80%) duomenų kiekvienam klasifikavimo algoritmui duomenis toliau skirstėme į mokymo(80%) ir validavimo(20%). Paskutiniame skyriuje palyginsime KNN ir RandomForest klasifikavimą, duosime jiems suklasifikuoti pradinius validavimo objektus, ir playginsiome rezultatus.

2.4. Duomenų aibės normavimas

Duomenų normavimui buvo parinkti šie požymiai: „redshift“, „u“, „g“, „r“, „i“ ir „z“. Kiti požymiai buvo nenormuoti, nes jie yra arba identifikaciniai („run_ID“, „field_ID“, „cam_col“, „plate“, ir „fiber_ID“) arba kategoriniai („class“). Duomenys buvo normuoti naudojant „min-max“ metodą (2.3 pav.)

	u	g	r	i	z	run_ID	cam_col	field_ID	class	redshift
count	3750.000000	3750.000000	3750.000000	3750.000000	3750.000000	3750.000000	3750.000000	3750.000000	3750.000000	3750.000000
mean	0.553554	0.567326	0.600287	0.575135	0.626007	4538.986667	3.458667	187.930667	0.500000	0.150682
std	0.153779	0.130514	0.137701	0.135934	0.154527	2006.732319	1.581020	145.382743	0.500067	0.130637
min	0.000000	0.000000	0.000000	0.000000	0.000000	109.000000	1.000000	11.000000	0.000000	0.000000
25%	0.445102	0.499826	0.535350	0.508909	0.546452	3185.000000	2.000000	84.000000	0.000000	0.060935
50%	0.545191	0.592381	0.630793	0.595148	0.644221	4187.000000	3.000000	148.000000	0.500000	0.104413
75%	0.653307	0.651715	0.699891	0.675557	0.741953	5640.000000	5.000000	251.000000	1.000000	0.222368
max	1.000000	1.000000	1.000000	1.000000	1.000000	8162.000000	6.000000	837.000000	1.000000	1.000000

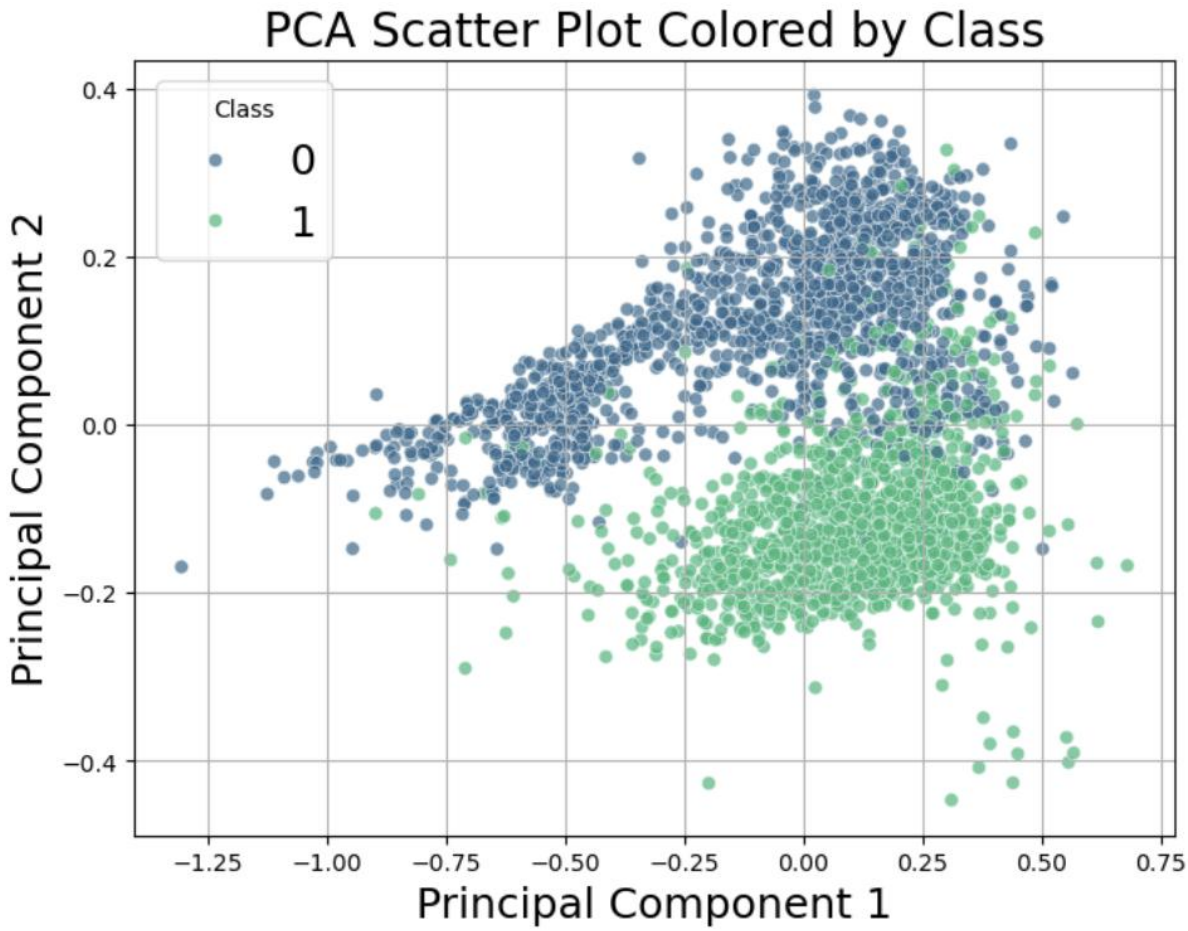
2.3 pav. min-max metodu normuotos duomenų aibės statistika

2.5. PCA algoritmu sumažintos dimensijos duomenys

Žemiau esančiame grafike (2.4 pav.) pateikiamas pagrindinių komponentų analizės („PCA“) grafikas, pritaikytas atrinktiems požymiams. Analizės metu buvo naudojami „u“, „g“, „r“, „i“, „z“ ir „redshift“ požymiai, kurie prieš tai buvo normuoti, siekiant užtikrinti teisingą

„PCA“ metodo veikimą. Šis metodas buvo pritaikytas atsitiktinai atrinktiems 3570 objektų, siekiant sumažinti duomenų dimensiją iki 2 pagrindinių komponenčių. Gauti rezultatai naudojami tolimesniuose klasifikavimo algoritmuose.

Pagrindinių komponenčių analizės rezultatai parodė, kad pirmoji komponentė (PC1) paaiškina 71.7 %, o antroji komponentė (PC2) – 18,8 % duomenų variacijos, tai reiškia, kad šios dvi komponentės kartu apima didžiąją dalį duomenų informacijos (90,5 %).



2.4 pav. "PCA " metodu sumažintos dimensijos duomenys.

2.6. Duomenų aibės dalinimas į mokymo ir validavimo

Iš pradinių atrinktų 3875 objektų, 80% jų buvo atrinkti KNN ir RandomForrest mokymui, ir 20% - jų galutiniam validavimui. Toliau paimti 80% buvo vėl padalinti į validavimo(20%) ir mokymo(80%) – šį antro laipsnio dalinimą naudojo atitinkamai KNN ir RandomForrest klasifikatoriai, kad būtų surasti optimalūs parametrai mūsų duomenų aibei. Tuomet suradus optimalius KNN ir RandomForrest parametrus, apmokintus algoritmus lyginsime su pradiniais 20% validavimo objektais. Visuose dalyse buvo išlaikyta klasių pusiausvyra.

3. KNN ALGORITMAS

3.1.1. Aprašymas

KNN klasifikavimo algoritmas yra pagrįstas atstumų matavimu tarp duomenų taškų, priskiriant klasę pagal artimiausių kaimynų daugumą. Šis algoritmas ypač tinka mažiems ir vidutinio dydžio duomenų rinkiniams, kur atstumai tarp taškų gerai reprezentuoja jų tarpusavio panašumą, pvz. PCA algoritmas. KNN klasifikavimo procesas susideda iš šių pagrindinių žingsnių:

1. Apskaičiuojami atstumai tarp klasifikuojamo taško ir visų mokymo duomenų taškų.
2. Surandami „k“ artimiausi kaimynai.
3. Klasė priskiriama pagal daugumos kaimynų klasę (su galimybe naudoti skirtingus svorius, pvz., pagal atstumą).

Svarbiausi KNN algoritmo parametrai:

- n_neighbors: artimiausių specifinės klasės kaimynų skaičius, reikalingas norint priskirti objektą tai klasei. Mažesnis skaičius gali būti jautrus triukšmui, o didesnis gali sulieti klasterių ribas.
- weights: svėrimo metodas. Galimos reikšmės:
    - uniform: visi kaimynai turi vienodą įtaką klasifikacijai.
    - distance: kaimynai, esantys arčiau, turi didesnę įtaką.
- p: metrika atstumo matavimui. Dažniausiai naudojamos:
    - p=1: Manheteno (L1) metrika.
    - p=2: Euklido (L2) metrika.

Tolimesniuose žingsniuose šie parametrai ir jų kombinacijos bus ištestuoti, bandant surasti optimaliausią kombinaciją pagal F1 metriką.

3.1.2. KNN algoritmas sumažintos dimensijos duomenims

Žemiau esančiame grafike(x.x pav) pavaizduotas KNN algoritmo pritaikymo sumažintų dimensijų duomenims F1-rezultato kitimas priklausomai nuo „n_neighbors“, „p“ ir „weight“ parametrų.

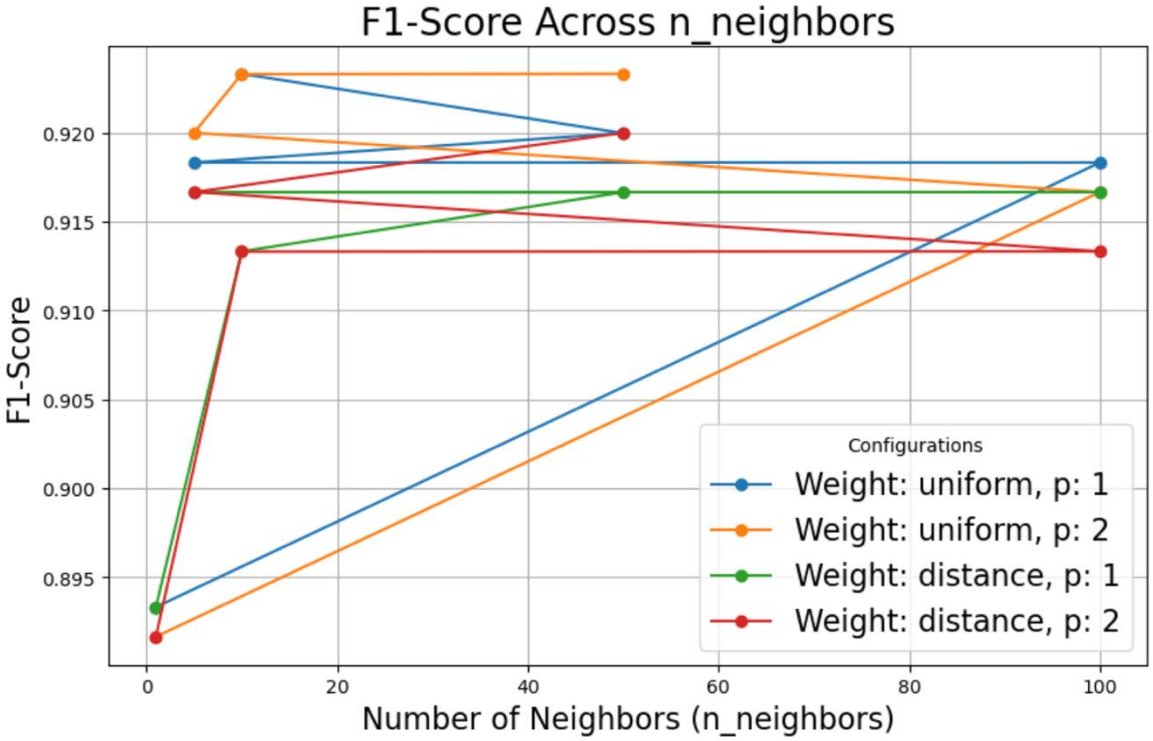
F1-rezultatas, pateiktas vertikalioje ašyje, rodo klasifikacijos balanso tarp tikslumo (precision) ir atgaminimo (recall) efektyvumą. Horizontalioje ašyje matomas kaimynų skaičius, naudojamas klasifikacijai.

Iš grafiko matyti, kad „weight“ ir „p“ parametrai nesuteikia tokio didelio F1 reikšmės skirtumo, kokį suteikia „n_neighbors“. Lyginant „n_neighbors“ reikšmes su skirtingomis „weight“ ir „p“ reikšmėmis negalima pasakyti, kad tam tikros jų reikšmės nulemia tikslesnį klasifikavimą.

Tiksliausias klasifikavimas pagal F1 metriką gautas 2 parametrų rinkiniuose:

- kai „n_neighbors“ lygus 10, “weight” lygus “uniform”, “p” lygus 1 arba 2.
- Kai “n_neighbors” lygus 50, “weight” lygus “uniform”, “p” lygus 2.

Toliau naudosime „n_neighbors“ = 10, „weight“ = „uniform“ ir „p“ = 1 parametrus.



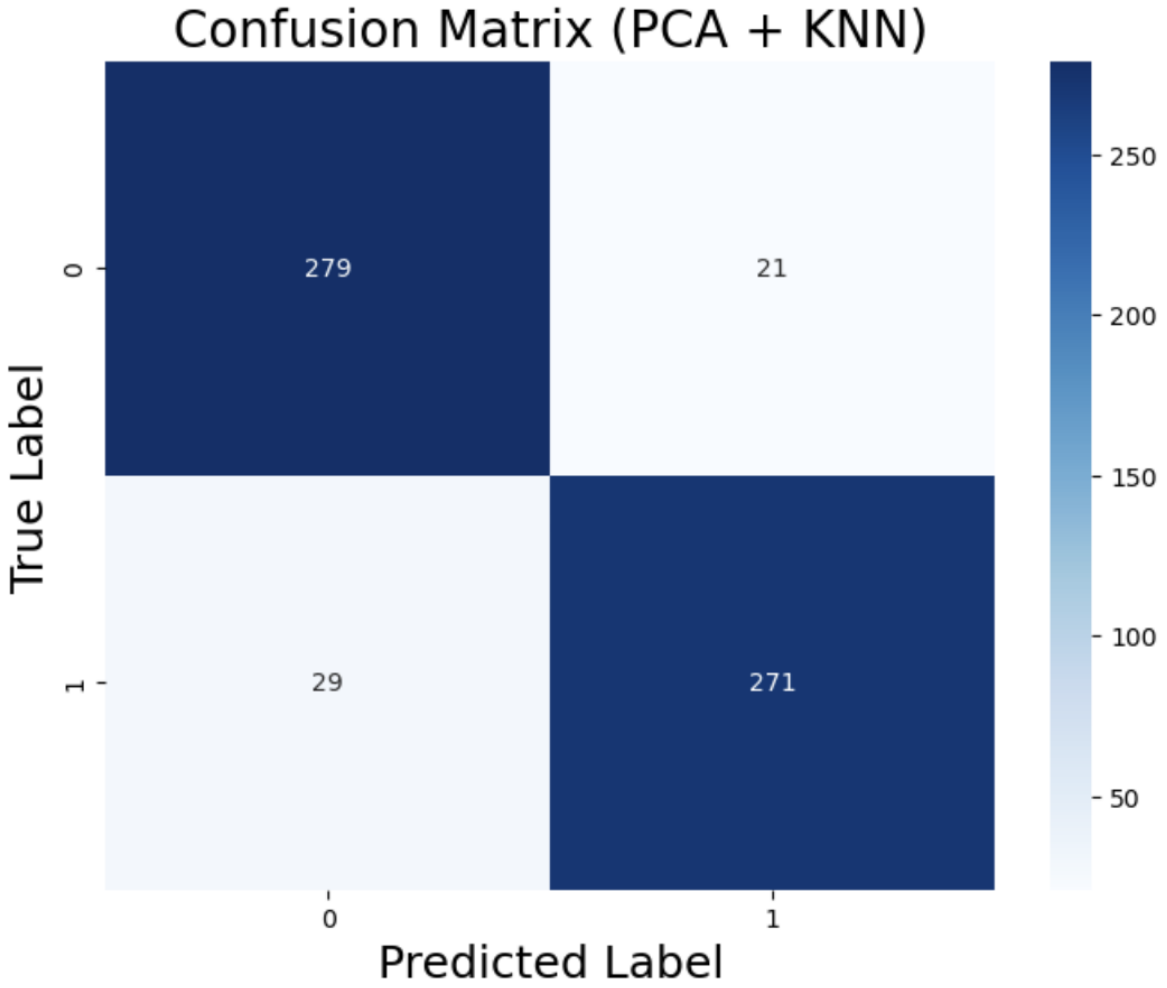
	Precision (tikslumas)	Recall (Atkūrimas)	F1	Objektų kiekis
0 klasė	0.91	0.93	0.92	300
1 klasė	0.93	0.90	0.92	300
Accuracy(tikslumo rodiklis)			0.92	600
Macro avg(makro vidurkis)	0.92	0.92	0.92	600
Weighted avg(svertinis vidurkis)	0.92	0.92	0.92	600

Naudojant optimalius KNN parametrus (n_neighbors=10, weights=uniform, p=1), buvo pasiekti šie klasifikavimo rezultatai(aukščiau lentelė):

Bendra klasifikavimo tikslumo reikšmė siekia 92%.

Abiejų klasių F1 metrika yra identiška - 92%. Tai reiškia, kad modelis taip pat gerai atpažįsta abi klases, klasifikacija yra subalansuota.

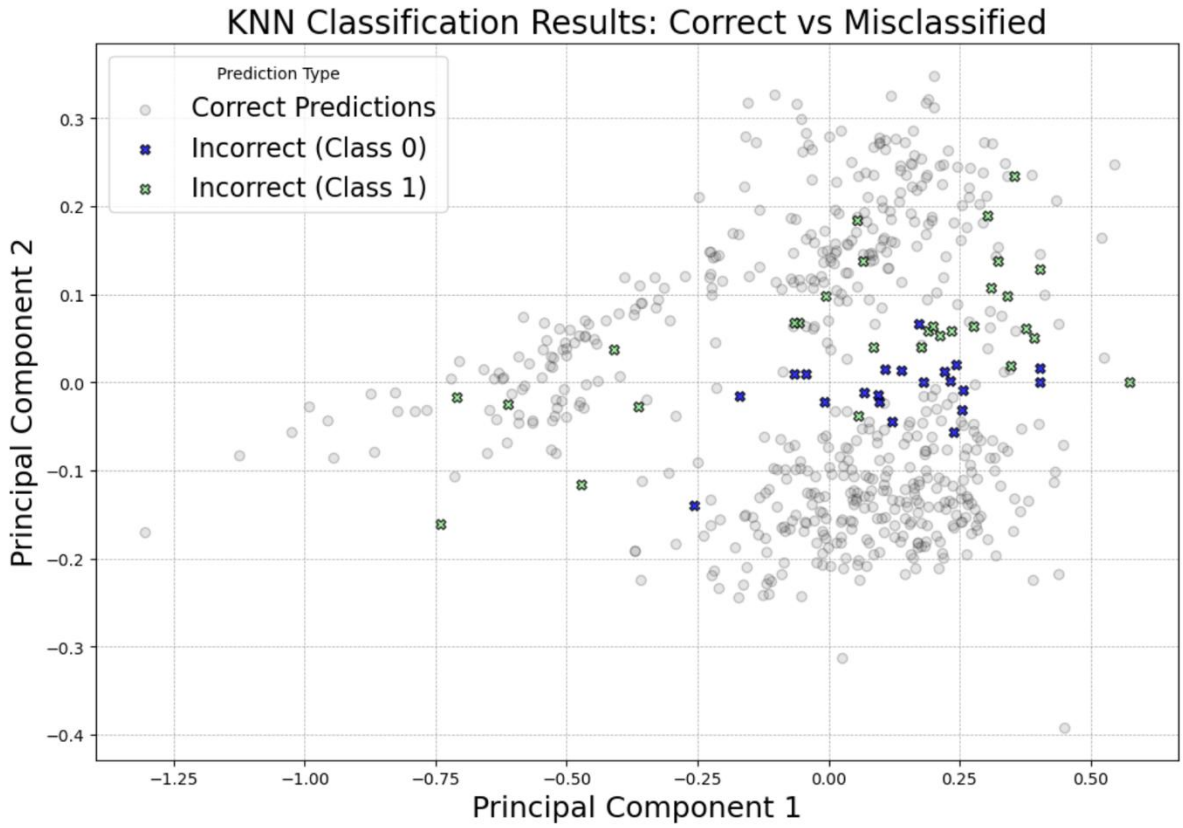
Makro vidurkis („macro avg“) ir svorinis vidurkis („weighted avg“) patvirtina, kad modelis dirba subalansuotai, atsižvelgiant į visas klases.



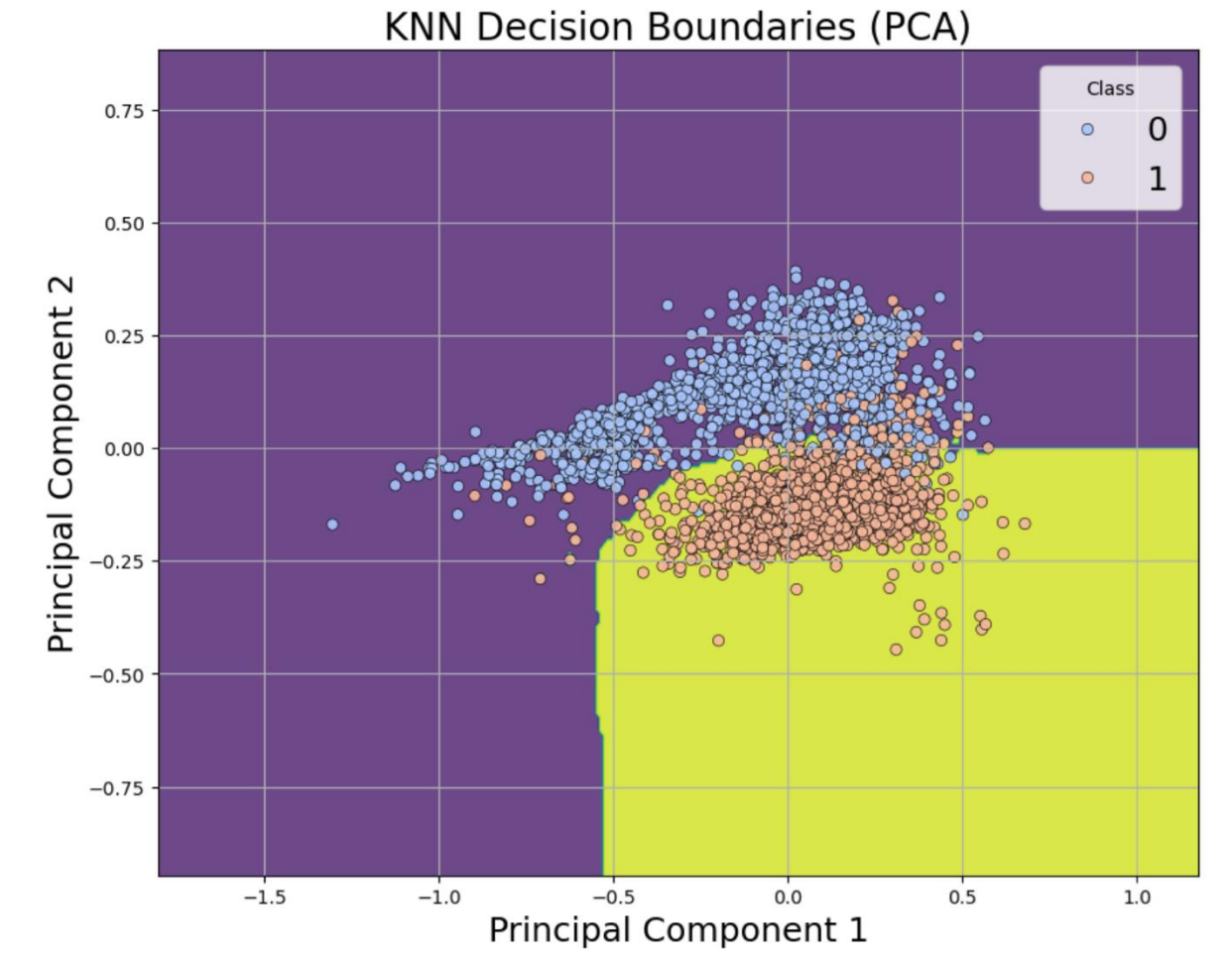
Žvelgiant į sumaišymo matricą(aukšt. pav.) galima matyti, kuriuos objektus algoritmas priskyrė 0 klasei, 1 klasei, ir ar tai buvo teisingas ar neteisingas priskyrimas.

Priskiriant objektus 1 klasei, KNN klasifikatorius suklydo maždaug **9.7%** kartų.

Priskiriant objektus 0 klasei, KNN klasifikatorius suklydo maždaug **6.9%** kartų.



Aukšt. Pav. galima matyti, kuriuose vietose KNN klasifikatorius blogai klasifikavo objektus. Pagrinde blogai klasifikuoti objektai buvo 0 ir 1 klasių sankirtoje. Šioje vietoje 0 klasės objektas gali būti arčiau daugiau objektų, priklausančių 1 klasei, ir atvirkščiai su 1 klasės objektu apsuptu 0 klasės objektų, dėl ko klasifikatorius negali tiksliai nustatyti klasės. Iš šio grafiko ir sprendimų ribų tinklelio(žem. Pav.) galima daryti spėjimus, kuriai klasei objektas priklausys, nežinant jo tikros klasės.



Išvada: Modelio klasifikavimo rezultatai rodo, kad KNN modelis, naudojant PCA metodą sumažinti duomenų dimensiją, 0 ir 1 klases dėl jų mažo persidengimo atpažysta gan gerai – klasifikuodamas klysta tik apytiksliai **8%** kartų.

3.1.3. KNN algoritmas originaliai duomenų aibei

Žemiau esančiame grafike(x.x pav) pavaizduotas KNN algoritmo pritaikymo originalių duomenų F1-rezultato kitimas priklausomai nuo „n_neighbors“, „p“ ir „weight“ parametrų.

F1-rezultatas, pateiktas vertikalioje ašyje, rodo klasifikacijos balanso tarp tikslumo (precision) ir atgaminimo (recall) efektyvumą. Horizontalioje ašyje matomas kaimynų skaičius, naudojamas klasifikacijai.

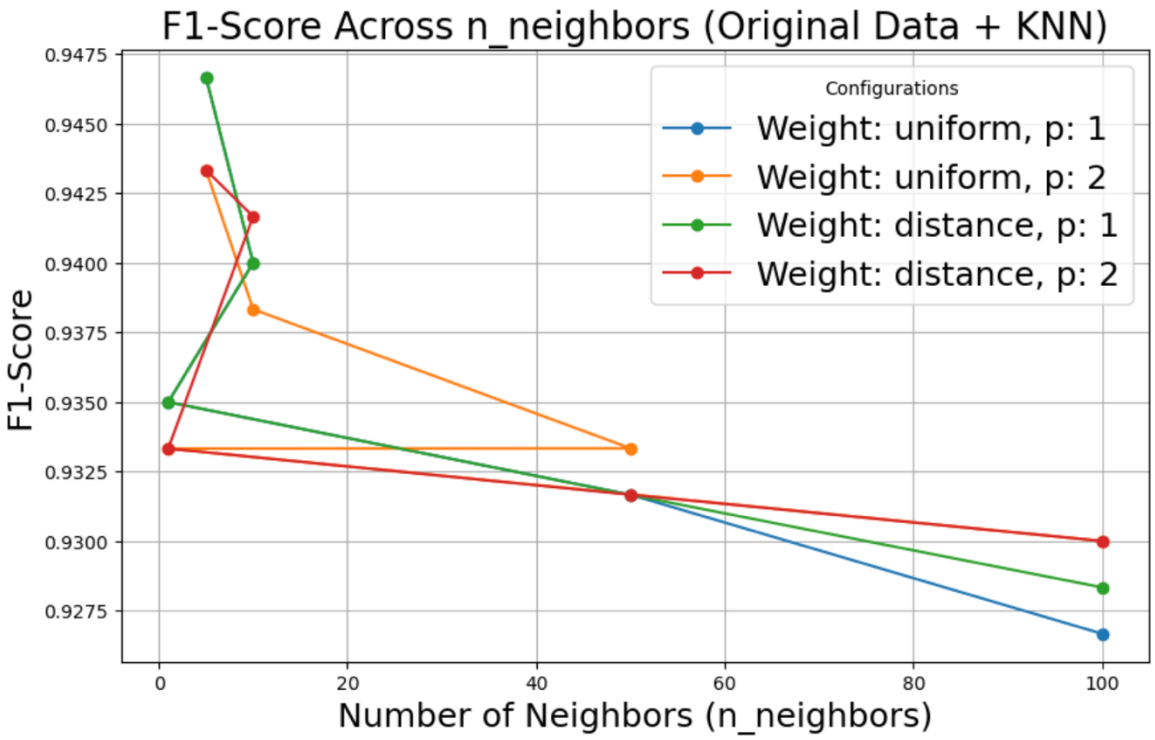
F1 metrika naudojant originalius duomenis yra labiau nenuspėjama parametrų keitimui, lyginant su KNN klasifikavimu PCA metodu sumažintos dimensijos duomenims.

Iš grafiko matyti, kad „weight“ ir „p“ parametrai F1 reikšmei nedaro, nuspėjamos, didelės įtakos – stebint tą pačią „n_neighbors“ reikšmę ir lyginant „weight“ ir „p“ parametrų pokyčius matoma, jog F1 pokytis niekada nebus didesnis nei **0.1%**.

Tiksliausias klasifikavimas pagal F1 metriką gautas naudojant šiuos parametrus:

- kai „n_neighbors“ lygus 5, “weight” lygus “uniform”, “p” lygus 1.

Toliau naudosime šiuos parametrus.



	Precision (tikslumas)	Recall (Atkūrimas)	F1	Objektų kiekis
0 klasė	0.93	0.95	0.94	300
1 klasė	0.95	0.93	0.94	300
Accuracy(tikslumo rodiklis)			0.94	600
Macro avg(makro vidurkis)	0.94	0.94	0.94	600
Weighted avg(svertinis vidurkis)	0.94	0.94	0.94	600

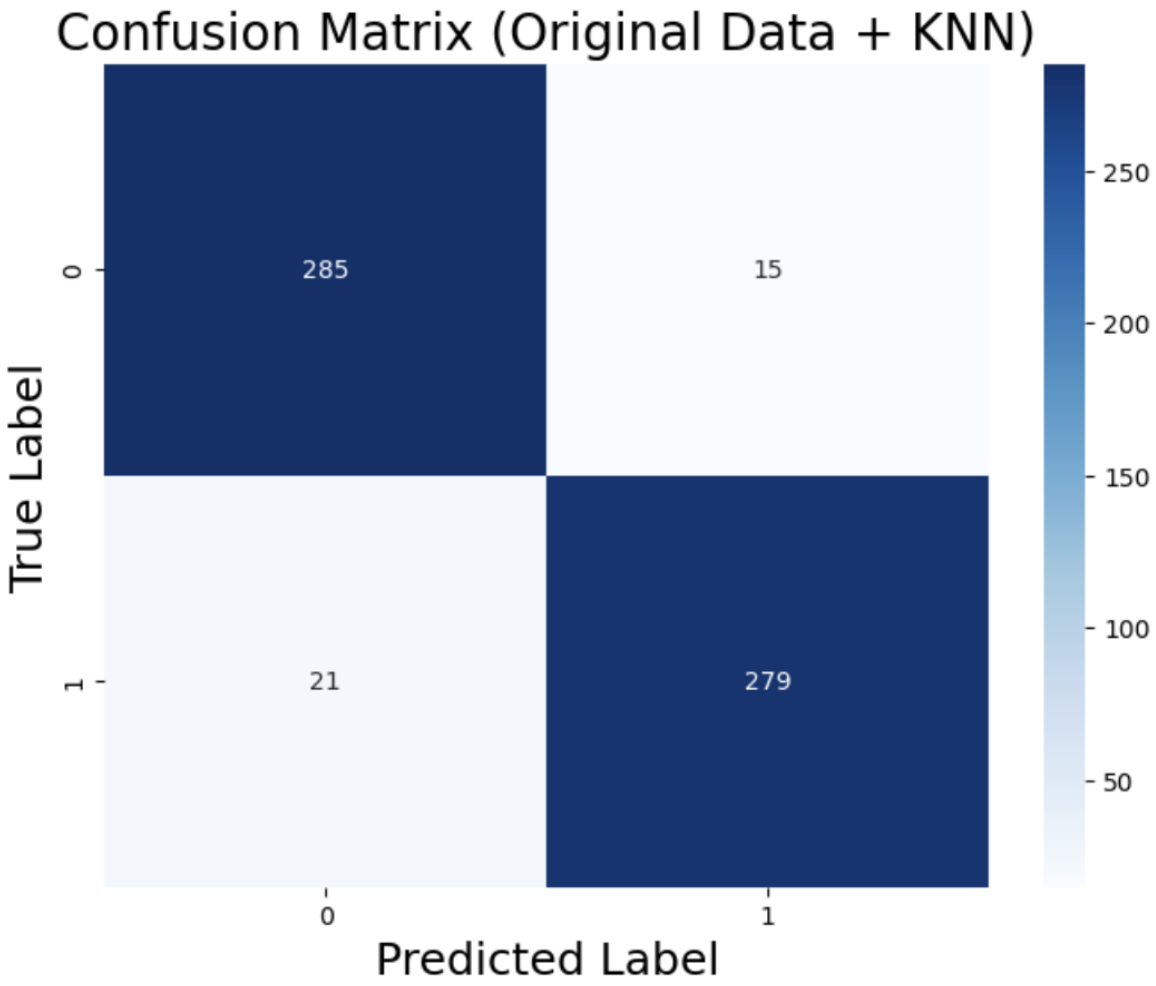
Naudojant optimalius KNN parametrus (n_neighbors=5, weights=uniform, p=1), buvo pasiekti šie klasifikavimo rezultatai(aukščiau lentelė):

Bendra klasifikavimo tikslumo reikšmė siekia 94%.

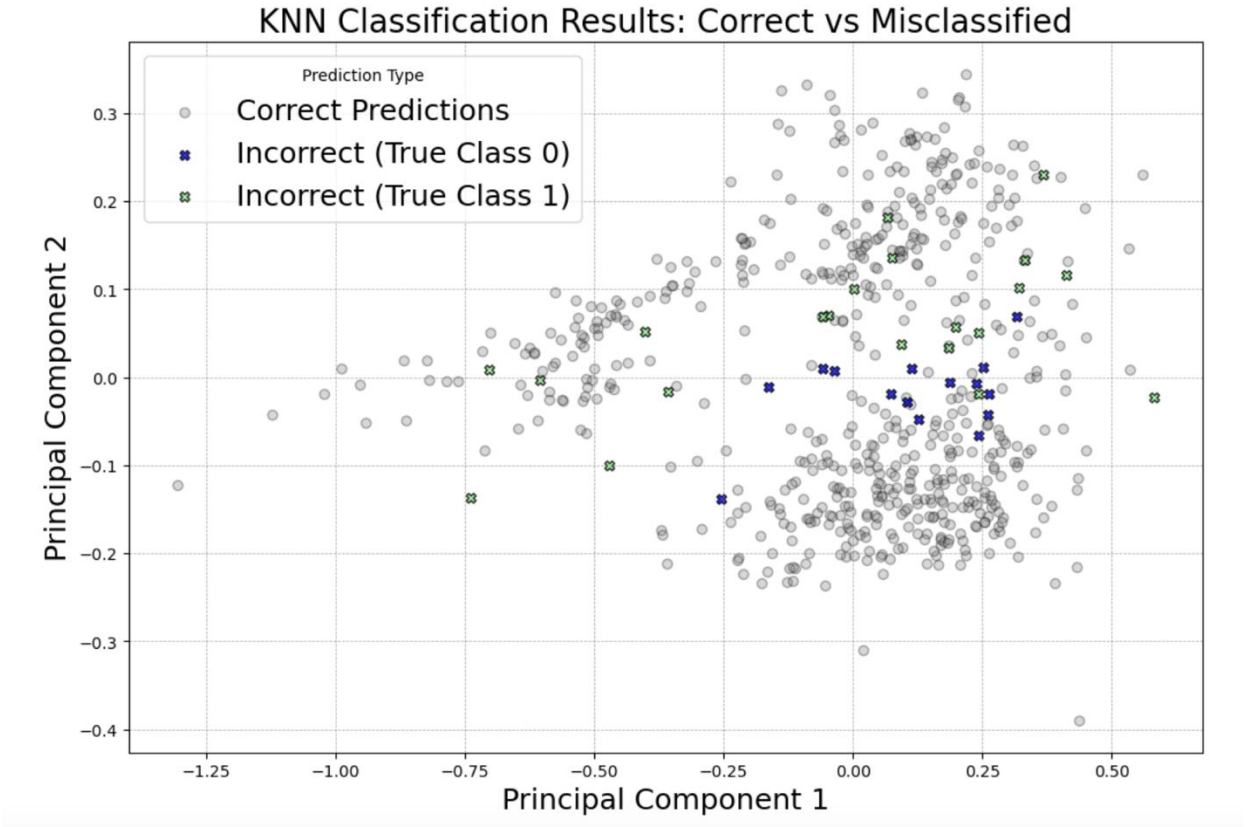
Abiejų klasių F1 metrika yra identiška - 94%. Tai reiškia, kad modelis taip pat gerai atpažįsta abi klases, klasifikacija yra subalansuota.

Makro vidurkis („macro avg“) ir svorinis vidurkis („weighted avg“) patvirtina, kad modelis dirba subalansuotai, atsižvelgiant į abi klases.

Naudojant originalius duomenis, o ne sumažinus dimensiją su PCA metodu gaunami 2% tikslesni rezultatai. Taip yra dėl to, nes PCA metodas negali tobulai atvaizduoti 6 atributus turinčių objektų dvimatėje erdvėje – PC1 ir PC2 komponentės vaizduoja tik maždaug 90% duomenų variacijos.



Sumaišymo matricoje(aukšt. pav.) galima matyti, kad KNN klasifikatorius su originalių duomenų mokymu ir testavimo priskiriant 0 klasei klydo **5%** kartų, o priskiriant 1 klasei – **6.9%** kartų. Objektų klasifikavimo neatitikimus ir atitikimus atvaizdavome naudodami PCA dimensijų mažinimą po KNN algoritmo – šis grafikas yra beveik identiškas aukščiau esančiam, kai pirma pritaikomas PCA, tada klasifikuojama su KNN. Iš to galima daryti išvadą, jog ir sprendimų ribų tinklelis atrodys identiškai.

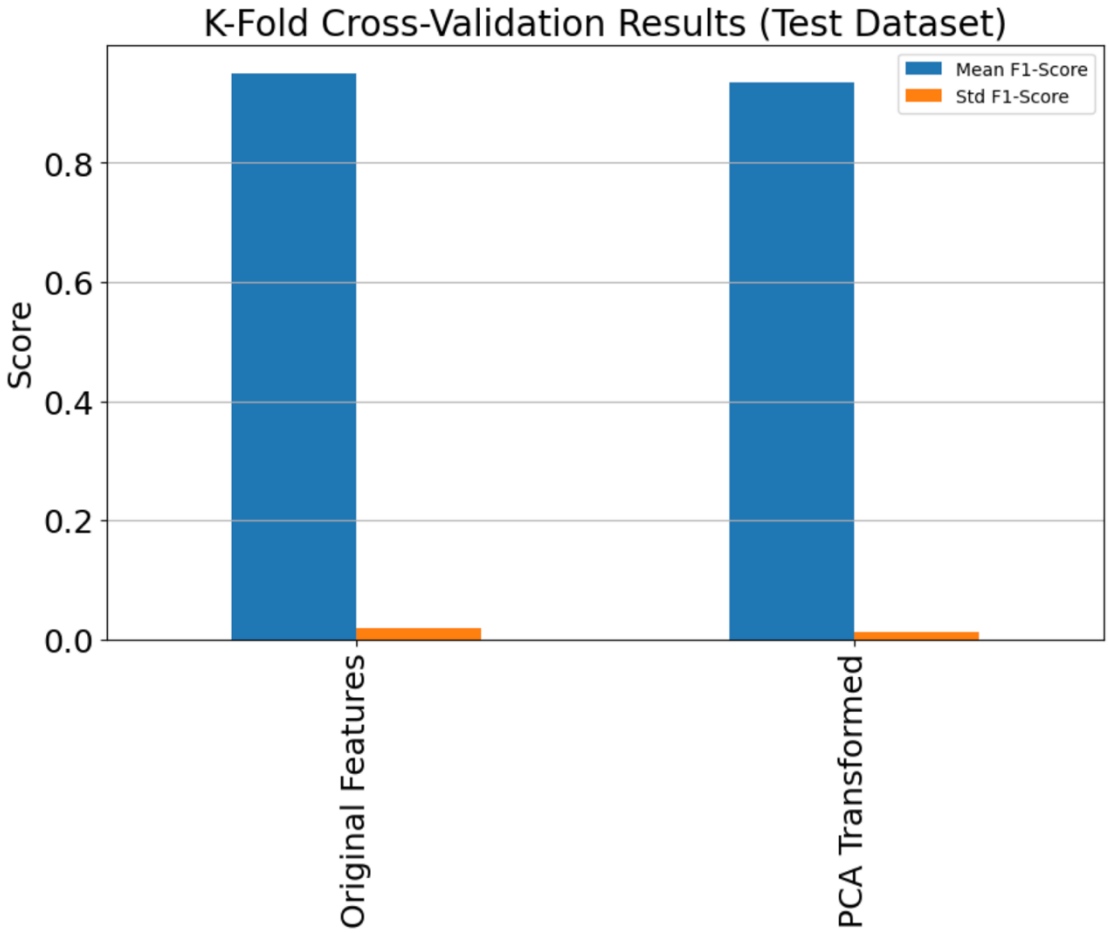


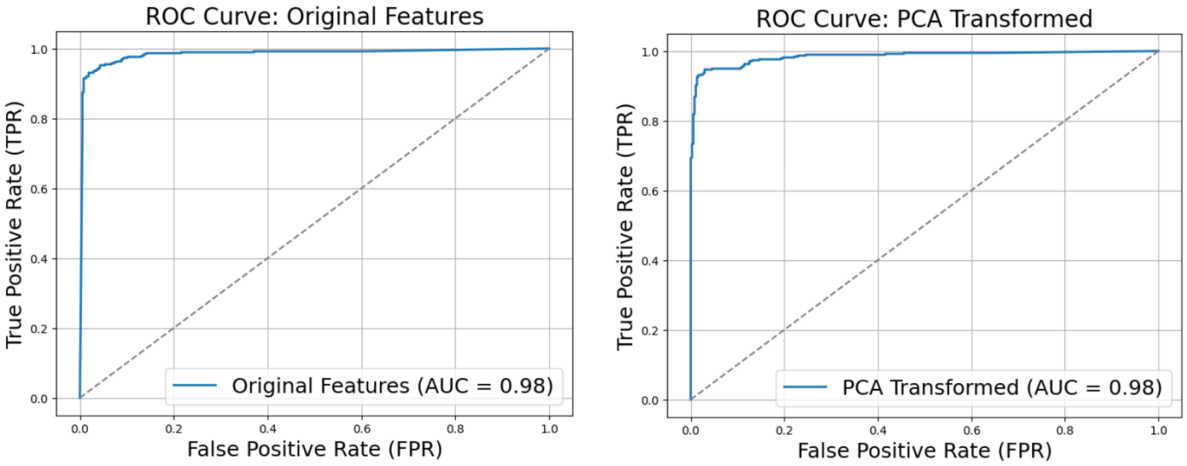
Išvada: Modelio klasifikavimo rezultatai rodo, kad KNN modelis, naudojant originalius duomenis, 0 ir 1 klases klasifikuoja 2% geriau nei naudojant duomenis PCA algoritmu sumažinta dimensija.

3.1.4. Klasifikatoriaus tikslumas

Žemiau esančiame(xx. Pav.) grafike yra matomi k-fold cross-validation testavimo rezultatai. Jam naudojome algoritmui dar nežinomus duomenis – pradinius atskirtus 20% testavimo objektų. Lyginant originalių duomenų ir PCA metodu sumažintos dimensijos duomenų F1 reikšmę ir standartinį nuokrypį, galima pastebėti, kad originalių duomenų našumas yra šiek tiek aukštesnis nei sumažintos dimensijos duomenų. Originalių duomenų vidutinė F1-metrika siekia **0.9507**, tuo tarpu PCA-redukuotų duomenų – **0.9347**. Tai rodo, kad originalių požymių rinkinys geriau išlaiko klasifikavimo tikslumą.

Tačiau sumažintos dimensijos duomenys turi mažesnį standartinį nuokrypį (**0.0115**) lyginant su originaliais duomenimis (**0.0187**). Tai reiškia, kad sumažintos dimensijos duomenų rezultatai yra stabilesni per skirtingus fold'us, nepaisant mažesnės vidutinės F1 metrikos.





Paskutiniai 2 grafikai susiję su KNN klasifikatoriumi yra ROC kreivės ir AUC reikšmės(xx. Pav., xx + 1. Pav.) grafikuose pateikiamos ROC kreivės sudarytos naudojant KNN algoritmą originaliems duomenims ir PCA metodu sumažintos dimensijos duomenims. ROC kreivė parodo algoritmo gebėjimą atskirti klases, vertinant tikslumą esant skirtingiems slenksčiams.

Pirmajame grafike (xx. Pav.) ROC kreivė parodo, kad originalių požymių rinkinys pasiekia labai aukštą tikslumą su AUC mato reikšme 0.9846. Tai rodo, kad originalūs duomenys suteikia puikų klasifikavimo tikslumą ir yra efektyvūs atskiriant klases.

Antrajame grafike (xx. pav.) yra sumažintos dimensijos ROC kreivė ir AUC matas. Abu grafikai yra beveik identiški, tačiau AUC matas antrame grafike yra minimaliai didesnis – 0.9848. Tai reiškia, kad sumažinus dimensijas PCA metodu, klasifikavimo tikslumas iš esmės nenukentėjo, o modelis išlaikė gebėjimą tiksliai atskirti klases.

Šie rezultatai rodo, kad tiek originalūs, tiek sumažintos dimensijos duomenys pasiekia beveik identišką klasifikavimo tikslumą. PCA transformacija išlaiko labai aukštą AUC reikšmę, šiek tiek lenkdama originalius duomenis, nors skirtumas yra beveik nereikšmingas.

3.1.5. Išvados

KNN algoritmas buvo įvertintas naudojant originalius duomenis („u“, „g“, „r“, „i“, „z“, „redshift“) ir PCA metodu sumažintos dimensijos duomenis. Pagrindinės išvados:

- Originalūs duomenys pasiekė aukštesnę vidutinę F1 metriką (0.9507) nei sunažintų dimensijų (0.9347).
- Sunažintų dimensijų duomenys turėjo mažesnį standartinį nuokrypį (0.0115), kas rodo stabilesnius rezultatus.
- AUC reikšmės buvo beveik identiškos: originalūs duomenys (AUC = 0.9846), sumažintos dimensijos (AUC = 0.9848). Tai rodo, kad PCA dimensijų sumažinimas reikšmingai nepaveikė klasifikavimo tikslumo.
- Geriausi rezultatai pasiekti su n_neighbors = 5, weights = 'uniform', p = 1.

KNN algoritmas pasirodė esantis efektyvus, stabilus ir tinkamas tiek originalių, tiek PCA-redukuotų duomenų klasifikacijai, tačiau naudojant originalius duomenis jo tikslumas buvo minimaliai geresnis.

3.2. „RandomForest“ klasifikatorius

3.2.1. Aprašymas

Atsitiktinis miškas (angl. „Random Forest“) yra ansamblio tipo klasifikavimo algoritmas, t.y. jis sudarytas iš daugelio tarpusavyje nepriklausomai augintų sprendimų medžių, kad pagerintų bendrą tikslumą ir stabilumą. Kiekvienas sprendimo medis yra apmokomas iš atsitiktinai parinktos originalių duomenų aibės imties (su pasikartojimais) bei atsitiktinai atrinkto požymių pogrupio. Tokiu būdu gaunamas skirtingų, tarpusavyje ne itin koreliuotų medžių rinkinys, kurio bendras sprendimas priimamas balsavimo principu. Atsitiktiniai miškai paprastai pasižymi dideliu tikslumu, yra ganėtinai atsparūs permokymui („overfitting“) ir gali apdoroti didelį duomenų kiekį bei įvairių tipų požymius.

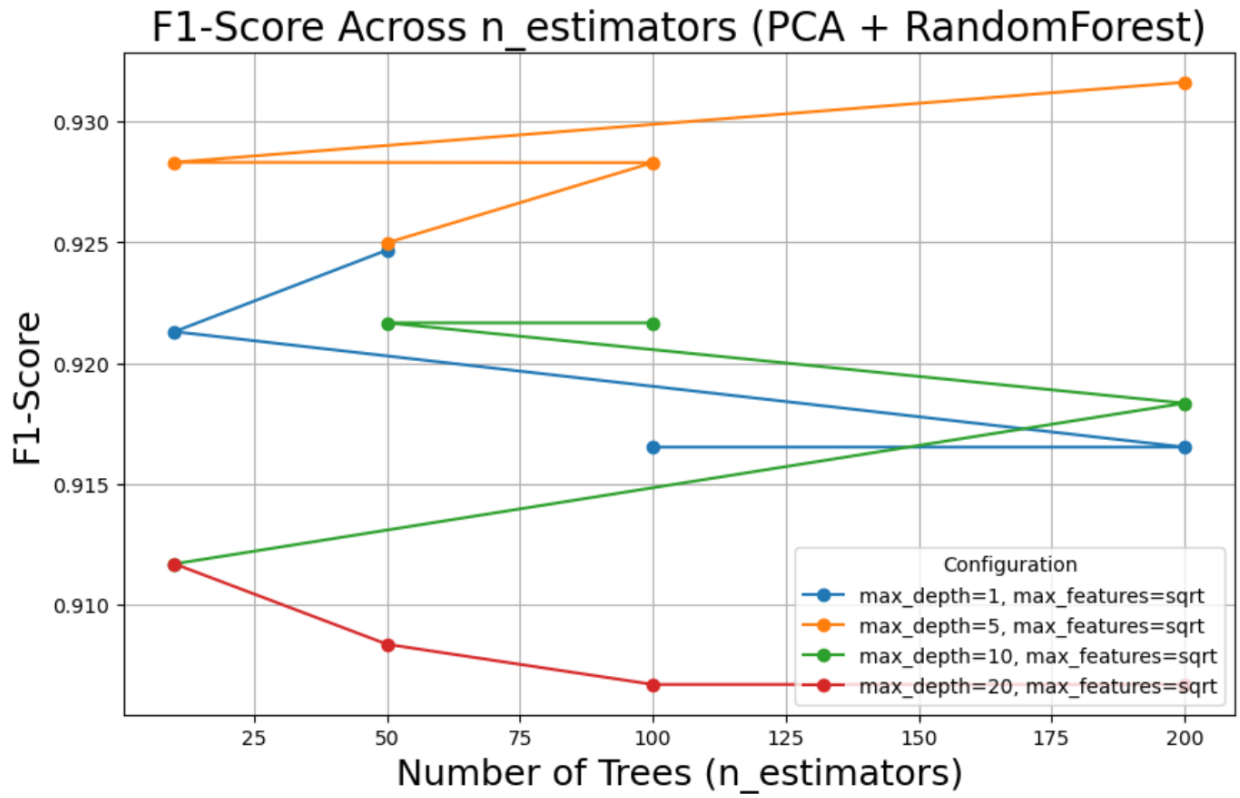
Pagrindiniai „Random Forest“ parametrai yra:

- **n_estimators**: sprendimų medžių skaičius miške. Didesnis medžių skaičius dažniausiai užtikrina didesnį modelio stabilumą ir tikslumą, tačiau didina skaičiavimo sąnaudas.
- **max_features**: didžiausias požymių kiekis, naudojamas kiekvienam medžio padalinimui. Šis parametras kontroliuoja funkcijų atsitiktinumą ir padeda sumažinti koreliaciją tarp medžių.
- **max_depth**: apriboja medžio gylį, taigi kontroliuoja modelio kompleksiškumą ir sumažina per didelio pritaikymo riziką.

Šiuo atveju „Random Forest“ klasifikatorius buvo taikytas:

- originaliai duomenų aibei, normalizuotai „min-max“ metodu ir turinčiai požymį „class“;
- tai pačiai duomenų aibei, bet su matmenų mažinimu (pvz., pritaikius PCA ar kitas dimensijos mažinimo technikas), siekiant įvertinti, ar sumažėjus požymių skaičiui keičiasi klasifikavimo kokybė.

3.2.2. Klasifikatoriaus taikymas sumažintos dimensijos duomenims



Šis grafikas (pav. xx) vaizduoja „F1-Score“ priklausomybę nuo „n_estimators“ parametro reikšmės (sprendimų medžių skaičiaus) naudojant „RandomForest“ klasifikatorių „PCA“ metodu sumažintų dimensijų duomenų aibei. Skirtingų spalvų kreivės atspindi įvairias „max_depth“ reikšmes, o „max_features“ reikšmė išlaikoma pastovi - „sqrt“.

Didžiausias „F1-Score“ (0.933) pasiektas, kai parametrų „max_depth“ reikšmė lygi 5, „n_estimators“ reikšmė lygi 200. Tai rodo, kad vidutinio gylio medžiai („max_depth“ = 5) kartu su didesniu sprendimų medžių skaičiumi užtikrina geriausią modelio našumą ir stabilumą.

Kai parametro „max_depth“ reikšmė yra 1, „F1-Score“ reikšmė yra žemesnė (tarp 0.917 ir 0.925), tačiau ne žemiausia. Tai tikėtina dėl to, kad pernelyg mažas medžio gylis riboja gebėjimą modeliuoti sudėtingus duomenų ryšius, bet nemažas medžių kiekis („n_estimators“) yra pakankamas, kad „F1-Score“ būtų aukštas. Pastebimas sąlyginai didelis svyravimas tarp mažesnių medžių skaičių (25–50) taip pat rodo, kad modelio stabilumas čia yra mažesnis.

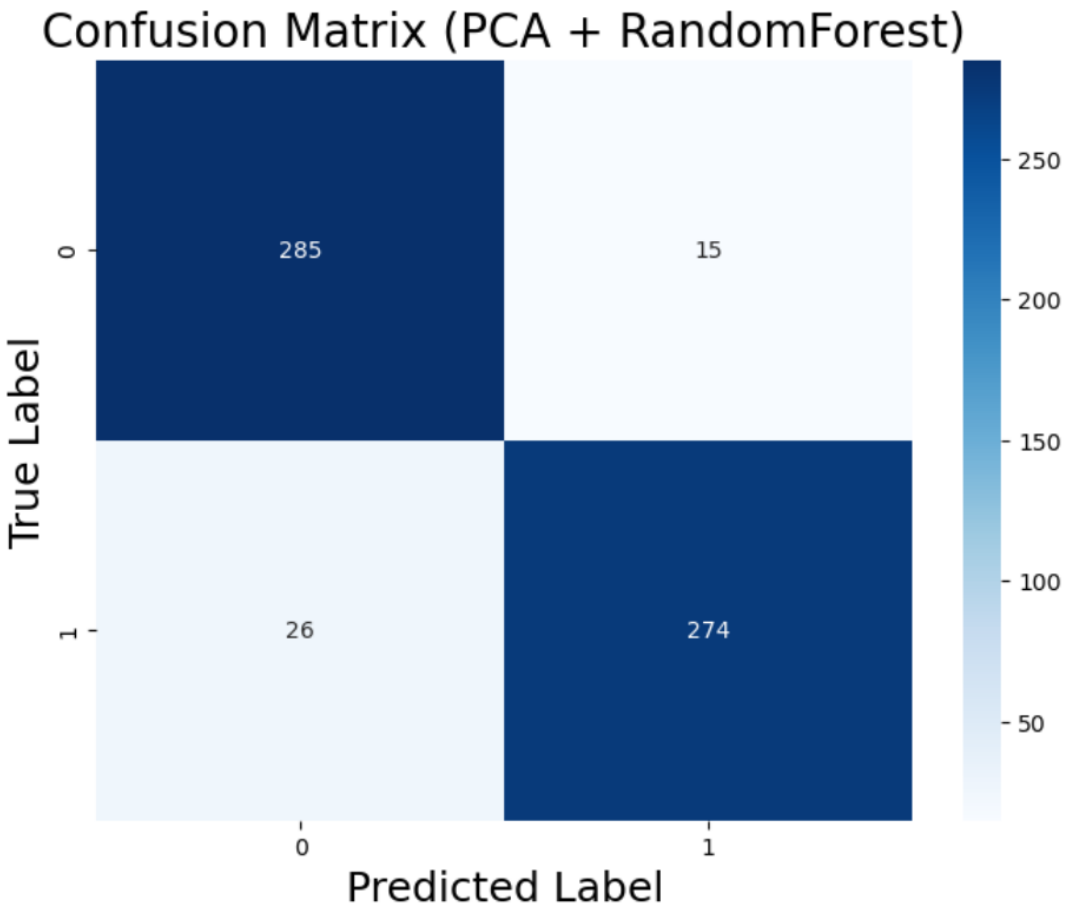
Kai parametro „max_depth“ reikšmė lygi 20, pastebima pastovus „F1-Score“ reikšmių mažėjimas didėjant medžių skaičiui. Tai gali rodyti, kad per didelis medžio gylis lemia permokymą („overfitting“), o didesnis „n_estimators“ kiekis – silpnina modelio našumą.

Pakeitus parametro „max_features“ reikšmę iš „sqrt“ į „log2“, „F1-Score“ reikšmės su visomis parametrų konfigūracijomis liko identiškas.

	Precision	Recall	F1-Score	Support
Class 0	0.92	0.94	0.93	300
Class 1	0.94	0.91	0.93	300
Accuracy			0.93	600
Macro Avg	0.93	0.93	0.93	600
Weighted Avg	0.93	0.93	0.93	600

Lentelė rodo "RandomForest" algoritmo klasifikavimo rezultatus sumažintos dimensijos duomenims, naudojant optimalius parametrus. Klasifikacijos tikslumas (accuracy) siekia 93%, o abiejų klasių F1 metrika yra identiška – 0.93 – tai parodo subalansuotą modelio veikimą.

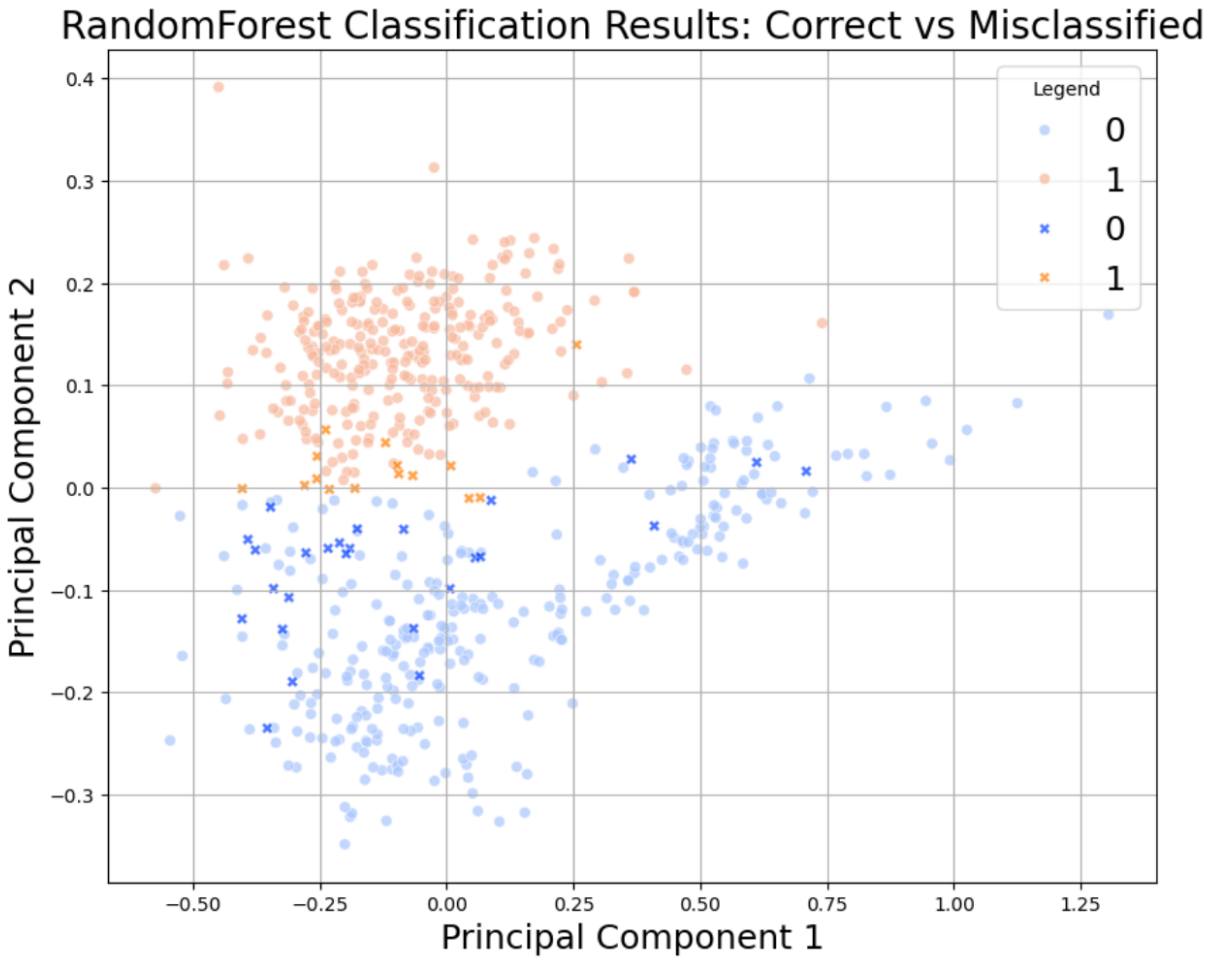
„Macro avg" ir „weighted avg" metrikos rodo, kad klasifikatorius vienodai gerai veikia visoms klasėms. Šie rezultatai rodo, kad sumažinta duomenų dimensija (naudojant „PCA" metod1) išlaikė aukštą klasifikavimo tikslumą, tačiau modelio veikimas tarp klasių yra šiek tiek asimetriškas.



Grafikas (pav XX) yra sumaišymo matrica (angl. Confusion Matrix), kuri vizualizuoja „RandomForest" klasifikatoriaus pritaikymo rezultatus sumažintos dimensijos duomenims, naudojant PCA metodą. Matrica aiškiai parodo teisingai ir neteisingai priskirtų objektų kiekius pagal tikrąsias ir numatytas klases.

- Tikslumas („accuracy“): Bendras tikslumo vertinimas yra aukštas, nes didžioji dalis objektų buvo teisingai priskirti savo tikrosioms klasėms.
- 0 klasės klaidos („false positives“): Nedidelis kiekis, tik 15, 0 klasės objektų klaidingai pateko į 1 klasę, kas rodo persidengimą tarp šių klasių požymių PCA erdvėje.
- 1 klasės klaidos („false negatives“): Šiek tiek daugiau klaidų fiksuota 1 klasėje – 26 1 klasės objektams buvo neteisingai priskirta 0 klasė. Tai rodo, kad 1 klasės požymiai PCA transformacijoje nėra visiškai atskiriami ir labiau linkę būti priskirti 0 klasei.

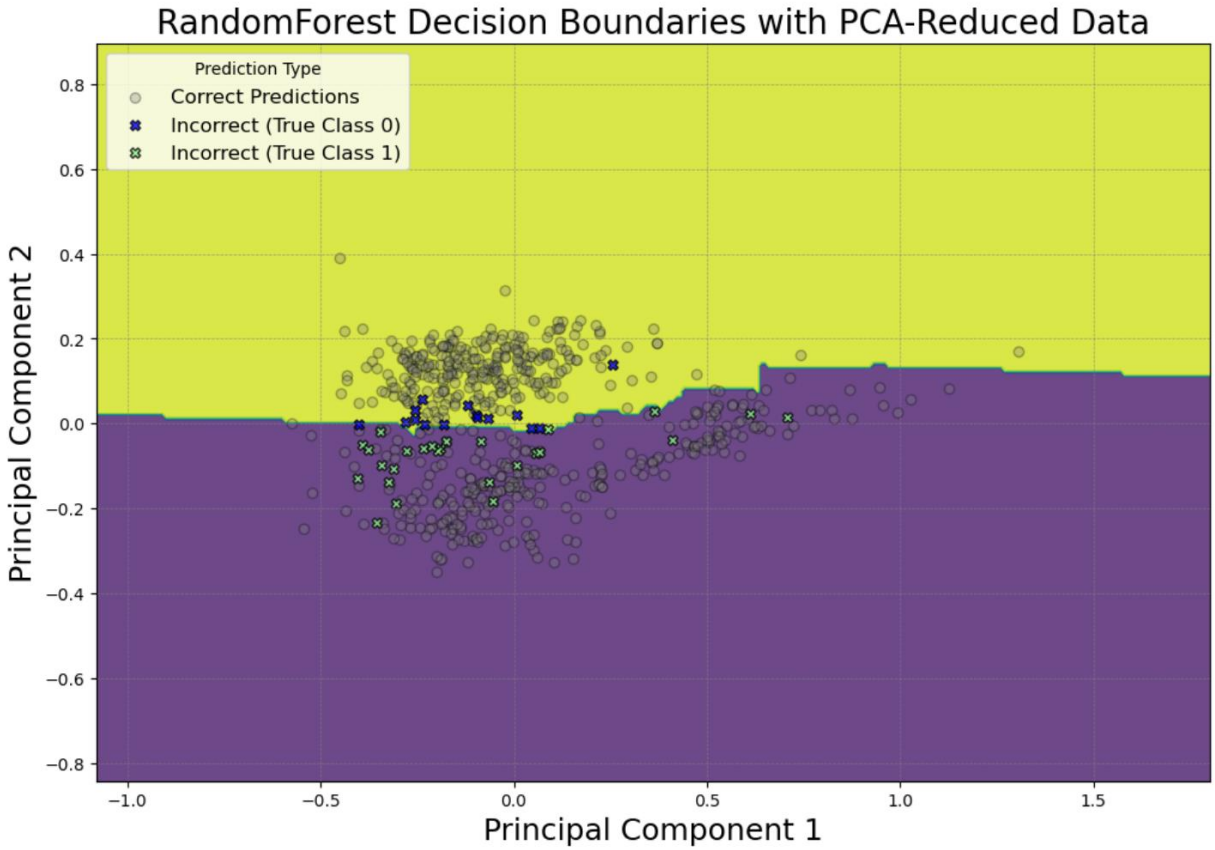
Ši sumaišymo matrica rodo, kad „RandomForest“ klasifikatorius su PCA sumažintais duomenimis veikia efektyviai, tačiau kai kurios klaidos tarp klasių rodo požymių persidengimą. Modelis ypač gerai atpažįsta 0 klasę, o 1 klasės klaidos reikalauja tolimesnės analizės arba parametrų optimizavimo. Bendras tikslumas išlieka aukštas, o rezultatai yra subalansuoti.



Grafikas (pav. xx) vizualizuoja „RandomForest“ klasifikatoriaus rezultatus dvimatėje erdvėje, kurioje požymiai sumažinti naudojant „PCA“ metodą. Horizontalioje ašyje žymima Pirmoji pagrindinė komponentė (PC1), o vertikalioje ašyje – Antroji pagrindinė komponentė (PC2). Skirtingos spalvos ir žymekliai parodo teisingai ir neteisingai klasifikuotus objektus.

Šiame grafike vizualizuojamos „RandomForest“ klasifikatoriaus nustatytos sprendimų ribos, kai duomenys yra sumažinti naudojant „PCA“. Apačioje esantys mėlyni skrituliai rodo teisingai suklasifikuotus 0 klasės objektus, o viršuje oranžiniai skirtuliai rodo teisingai suklasifikuotus 1 klasės objektus, jų yra didžioji dauguma. Tačiau yra ir netikslių priskyrimų

(kryžiukai), kurie randami pereinamojoje zonoje tarp dviejų klasių. Tai rodo, kad klasifikatorius sunkiai atskiria objektus, kai jų savybės artimos abiejų klasių požymiams.



XX2 pav. esanti geltona sritis vaizduoja 1 klasės prognozuojamą zoną, o violetinė – 0 klasės prognozuojamą zoną. Pilki skrituliai rodo teisingai suklasifikuotus objektus, žali kvadratai neteisingai klasifikuotus objektai, kurių tikroji klasė yra 1 („False Negatives“). Pastarieji yra pasiskirstę netoli sprendimų ribos, bet daugiausia violetinėje srityje. Tai rodo, kad modelis sunkiai atskiria kai kuriuos 1 klasės objektus, kurių savybės sutampa su 0 klasės. Mėlyni kvadratai neteisingai klasifikuotus objektai, kurių tikroji klasė yra 0 („False Positives“) yra dažniausiai randami pereinamojoje zonoje, netoli sprendimų ribos. Tai rodo, kad modelis kartais „pernelyg prisitaiko“ prie klasės 1 požymių.

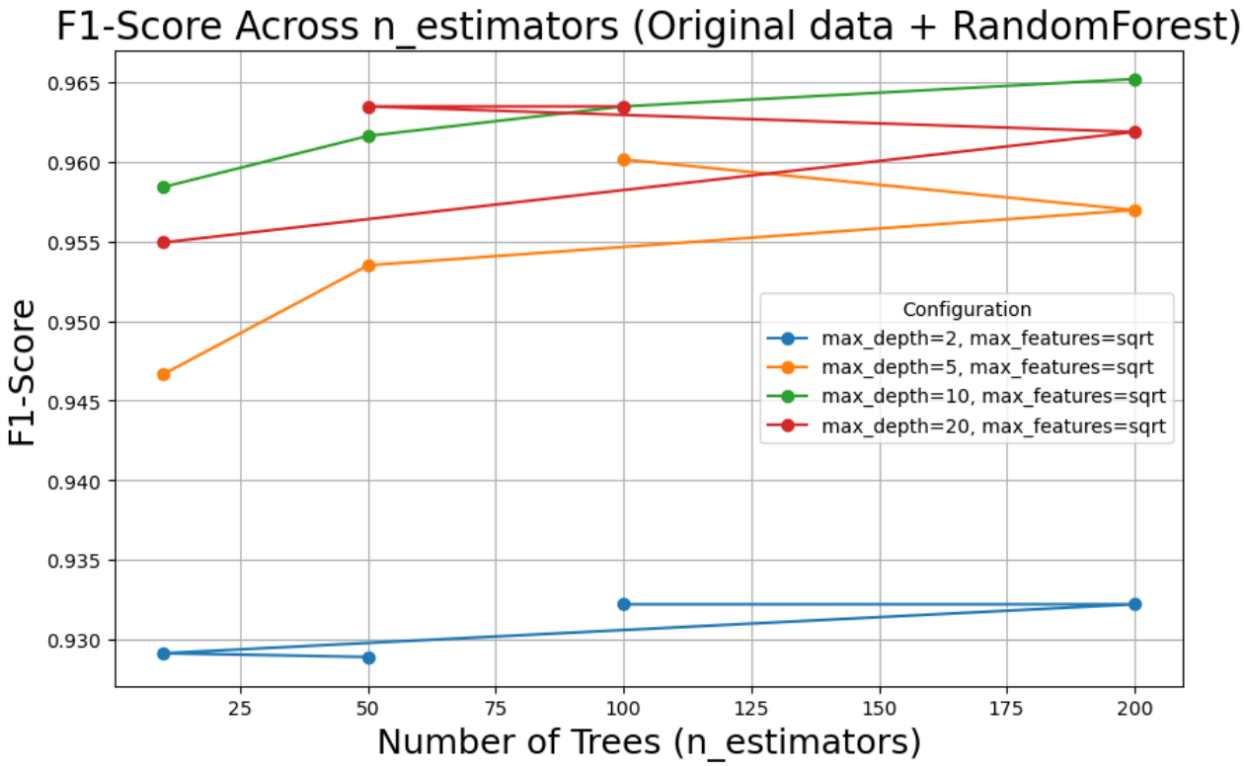
Tačiau klaidų koncentracija esanti abiejuose grafikuose pereinamojoje zonoje tarp dviejų klasių rodo modelio ribotumus dėl duomenų požymių persidengimo. PCA sumažinimas efektyviai atskleidė pagrindines tendencijas, tačiau aiškiai matome, kad buvo prarastas tikslumas ir atskyrimas nėra labai efektyvus.

Sprendimų riba yra nevienalytė ir sudėtinga. Tai rodo, kad „RandomForest“ modelis geba efektyviai modeliuoti netiesinį duomenų paskirstymą. Pereinamojoje zonoje (PC1 reikšmės tarp -0.5 ir 0.5) sprendimų ribos yra „nelygios“, kas rodo duomenų persidengimą.

3.2.3. Klasifikatoriaus taikymas originaliai normuotai duomenų aibei

Šis grafikas (pav. xx) vaizduoja „F1-Score“ priklausomybę nuo „n_estimators“ parametro reikšmės (sprendimų medžių skaičiaus) naudojant „RandomForest“ klasifikatorių su originaliais,

normalizuotais duomenimis. Skirtingų spalvų kreivės atspindi įvairias „max_depth“ reikšmes, o „max_features“ reikšmė išlaikoma pastovi - „sqrt“.



Didžiausias F1-score (0.966) pasiektas su max_depth=10 ir max_depth=20, kai n_estimators yra nuo 100 iki 200. Tai rodo, kad didesnis sprendimų medžių skaičius stabilizuoja modelio našumą. Kai parametro „max_depth“ reikšmė yra lygi 2, „F1-Score” yra mažiausias (0.928). Tai tikėtina dėl to, kad per mažas medžio gylis neleidžia modeliui tinkamai atskleisti sudėtingų duomenų ryšių.

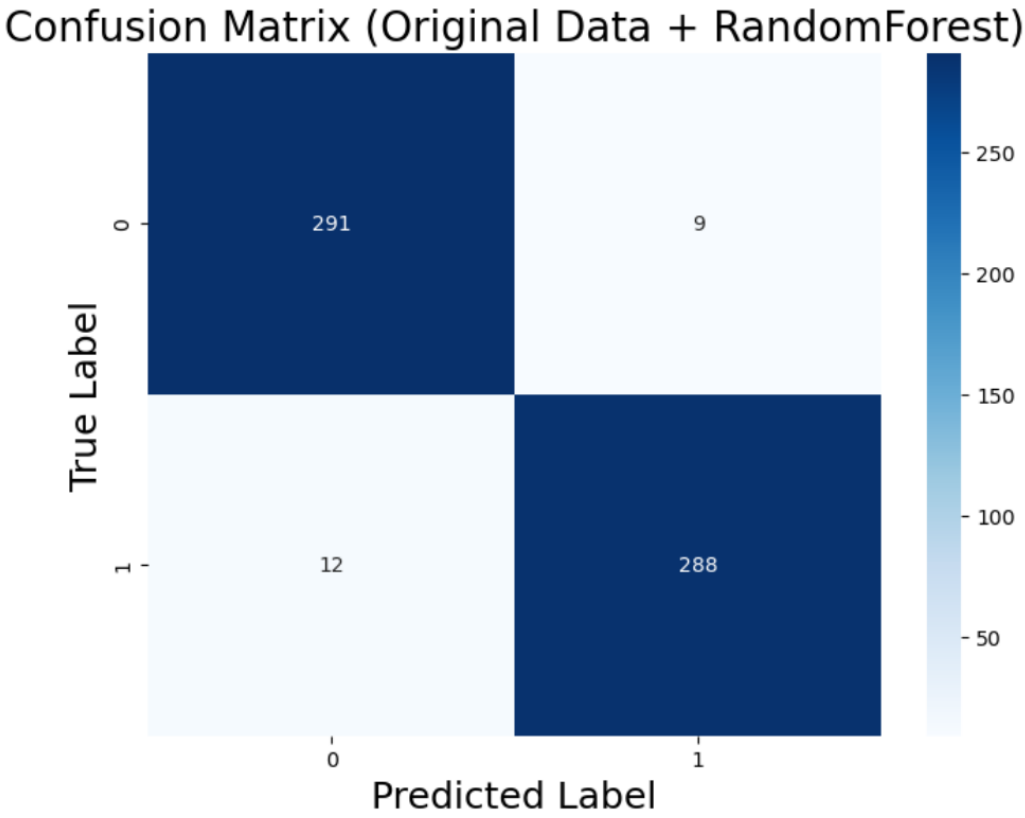
Mažesnės parametro „n_estimators“ reikšmės (25–50) lemia mažesnį stabilumą. Tai labiausiai atsikleidžia vidutinio gylio medžiams („max_depth“ = 5). Tačiau, per didelės „n_estimators“ reikšmės, gali sumažinti „F1-Score” reikšmę, tai rodo, kad didesnės medžių kiekis nebūtinai duos geresnius rezultatus. Medžių gylis (parametras „max_depth“) daro dar didesnę įtaką rezultatams negu medžių kiekis (parametras „n_estimators“). Optimalūs „F1-Score” rezultatai pasiekiami, kai „max_depth“ yra lygus 10, ir nežymiai geresnis kai „max_depth“ yra lygus 20. Verta pastebėti, kad, kai medžių gylis yra pakankamai aukštas, medžių kiekis daro vis mažesnę įtaką „F1-Score“ reikšmei.

Pakeitus parametro „max_features“ reikšmę iš „sqrt“ į „log2“, „F1-Score“ reikšmės su visomis parametrų konfigūracijomis liko identiškas.

	Precision	Recall	F1-Score	Support
Class 0	0.96	0.97	0.97	300
Class 1	0.97	0.96	0.96	300
Accuracy			0.96	600
Macro Avg	0.97	0.96	0.96	600
Weighted Avg	0.97	0.96	0.96	600

Lentelė pateikia klasifikacijos rezultatus, taikant “RandomForest” algoritmą originaliai duomenų aibei, su optimizuotais parametrais. Klasifikacijos tikslumas (accuracy) siekia 96%, o abiejų klasių F1 metrika yra labai artima – 0.96–0.97, tai parodo dar geresnį veikimą nei sumažintos dimensijos atveju.

„Macro avg“ ir „weighted avg“ metrikos yra beveik identiškos – jos patvirtina subalansuotą modelio veikimą tarp visų klasių. Šie rezultatai rodo, kad naudojant originalius duomenis, modelis pasiekia aukštesnį klasifikavimo tikslumą nei sumažintos dimensijos duomenys. Tai gali būti dėl išsamesnės informacijos, kurią prarado „PCA“ metodu transformuota duomenų aibė.



Šis grafikas rodo sumaišymo matricą („Confusion Matrix“), kuri įvertina „RandomForest“ klasifikatoriaus veikimą, taikant modelį originaliems, normalizuotiems duomenims.

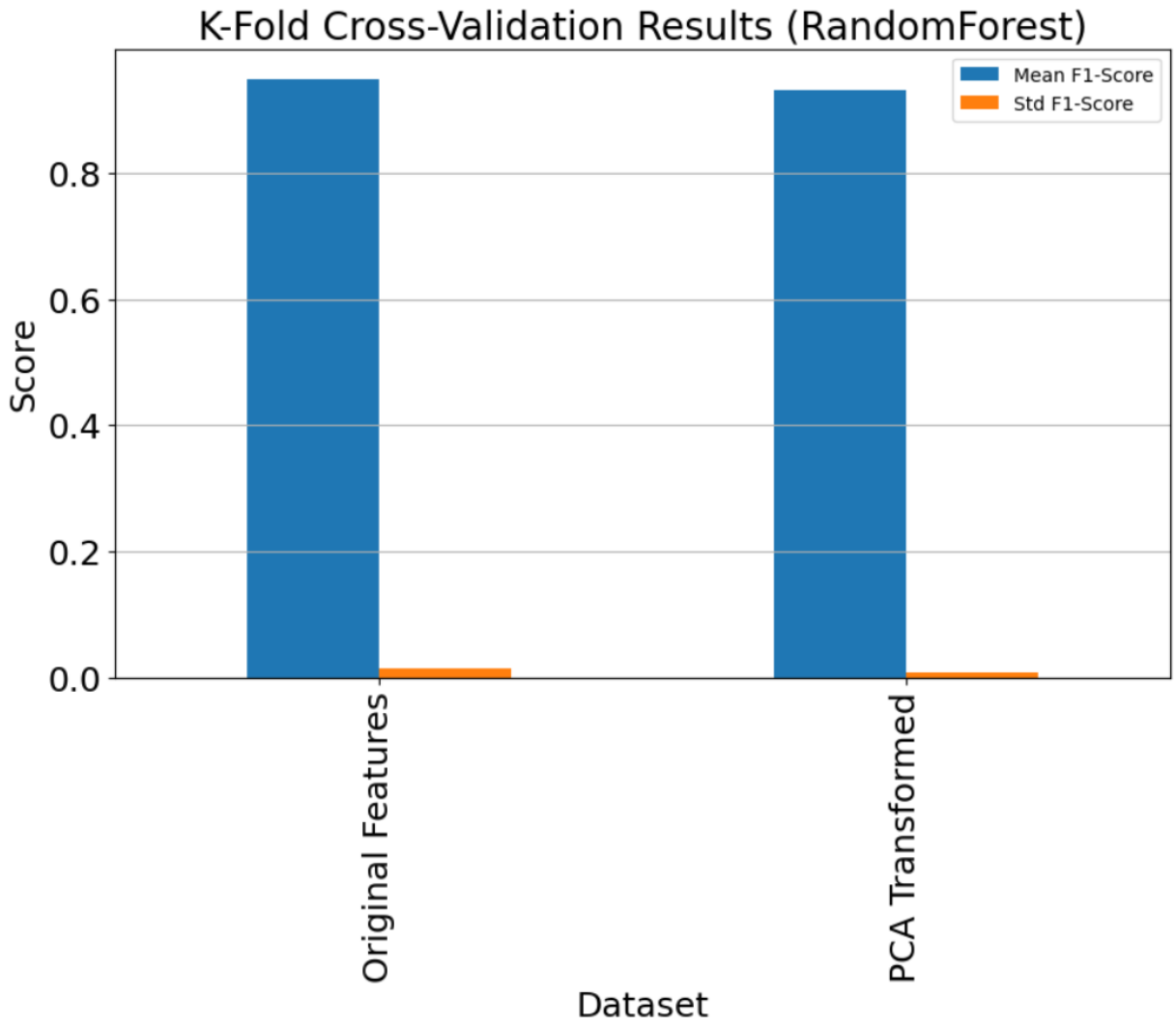
- Tikslumas („accuracy“): Bendras tikslumo vertinimas yra labai aukštas, nes didžioji dalis - 579 iš 600 - objektų buvo teisingai priskirti savo tikrosioms klasėms.
- 0 klasės klaidos („false positives“): Nedidelis kiekis, tik 9, 0 klasės objektų klaidingai pateko į 1 klasę.
- 1 klasės klaidos („false negatives“): Beveik tiek pat, tik 12, 1 klasės objektų klaidingai pateko į 0 klasę.

Ši sumaišymo matrica rodo, kad netikslumų skaičius yra labai mažas (tik 21 neteisingai priskirtas objektas iš 600), o klaidos tarp klasių yra pakankamai tolygiai paskirstytos. Tai rodo, kad modelis yra patikimas ir efektyviai atskiria abi klases originalioje duomenų erdvėje.

3.2.4. Klasifikatoriaus tikslumas

Šiame skyriuje buvo naudojami visai ištreniruotiems modeliam nematyti duomenys. Šie duomenys nebuvo naudojami nei „RandomForest“ modelių („PCA“ metodu mažintai duomenų aibei ir modelio originaliai, normuotai duomenų aibei) apmokymui, nei įvertinimui, kurie buvo atlikti prieš tai esančiuose skyreliuose.

Šis grafikas lygina „RandomForest“ klasifikatoriaus veikimą dviejuose skirtinguose duomenų rinkiniuose: originalių požymių ir „PCA“ metodu transformuotų požymių. Rodomi vidutiniai F1-rezultatai („Mean F1-Score“) ir jų standartiniai nuokrypiai („Std F1-Score“) taikant „K-fold Cross-Validation“ metodą.



Vidutinis F1-rezultatas:

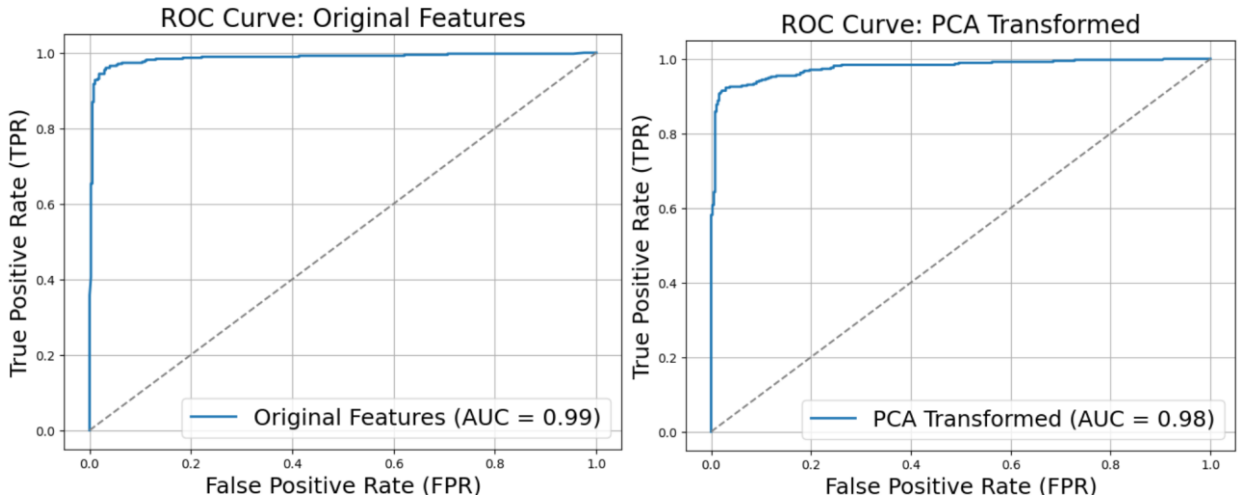
- Originalūs požymiai: Vidutinis F1-rezultatas yra labai aukštas (~0.96), kas rodo puikų klasifikatoriaus gebėjimą apdoroti originalią duomenų aibę.
- PCA transformuoti požymiai: Vidutinis F1-rezultatas yra panašus (~0.95), tačiau šiek tiek mažesnis nei naudojant originalius požymius.
- Tai rodo, kad PCA sumažinus duomenų dimensiją, klasifikatoriaus tikslumas praktiškai nenukenčia.

Standartinis nuokrypis:

- Originalūs požymiai: Standartinis nuokrypis yra labai mažas, rodantis, kad modelio rezultatai yra stabilūs tarp skirtingų k-fold iteracijų.
- PCA transformuoti požymiai: Standartinis nuokrypis taip pat yra mažas, bet dar šiek tiek mažesnis nei naudojant originalius požymius.
- Tai rodo, kad PCA sumažinti duomenys yra stabilesni, nes mažesnis dimensijų skaičius gali sumažinti atsitiktinius svyravimus.

Taigi, šis grafikas rodo, kad „RandomForest“ klasifikatorius efektyviai veikia tiek su originaliais, tiek su „PCA“ metodu transformuotais duomenimis. Nors vidutinis „F1-Score“ yra šiek tiek aukštesnis originaliuose duomenyse, „PCA“ transformuoti duomenys pasižymi mažesniu standartiniu nuokrypiu, o tai reiškia stabilesnius rezultatus. Tai rodo, kad PCA dimensijų

mažinimas gali būti naudingas siekiant optimizuoti modelio stabilumą be reikšmingo tikslumo praradimo.



Šie grafikai vaizduoja ROC kreives („Receiver Operating Characteristic“) dviejų skirtingų duomenų rinkinių – originalių požymių ir „PCA“ metodu transformuotų požymių, kai klasifikavimui naudojamas „RandomForest“ algoritmas. ROC kreivė parodo klasifikatoriaus gebėjimą atskirti klases esant įvairiems sprendimų slenksčiams.

Duomenų rinkinio su originaliais, normalizuotais požymiais ROC kreivė:

- Kreivė yra arti viršutinio kairiojo kampo, o tai reiškia puikų modelio gebėjimą atskirti klases.
- AUC = 0.99 reiškia, kad modelio tikslumas yra beveik idealus, aukštas klasifikavimo našumas.

„PCA“ metodu transformuoto duomenų rinkinio ROC kreivė:

- Kreivė yra labai panaši į originalių duomenų kreivę, tačiau šiek tiek mažiau arti viršutinio kairiojo kampo.
- AUC = 0.98 rodo, kad klasifikatorius vis dar efektyviai atskiria klases, tačiau šiek tiek mažesniu tikslumu nei su originaliais požymiais.

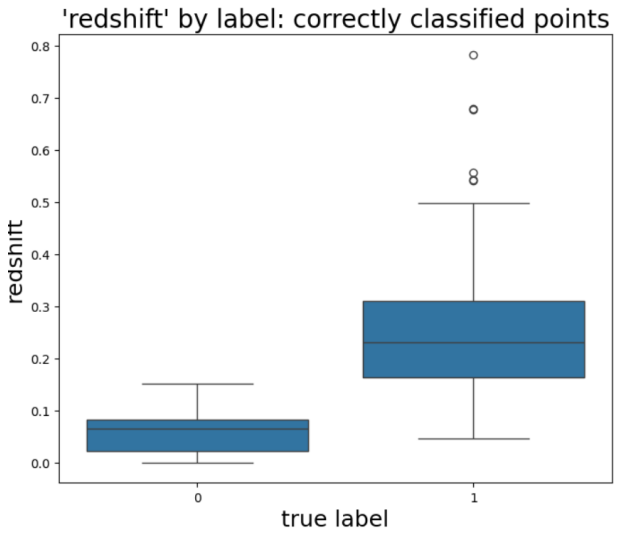
Šie grafikai rodo, kad „RandomForest“ klasifikatorius yra labai efektyvus tiek su originaliais, tiek su „PCA“ metodu transformuotais duomenimis. Originalūs požymiai suteikia maksimalų AUC rezultatą (0.99), tačiau „PCA“ metodu transformuoti duomenys taip pat pasiekia puikų rezultatą (AUC = 0.98), išlaikydami beveik identišką klasifikavimo gebėjimą su mažesniu duomenų sudėtingumu.

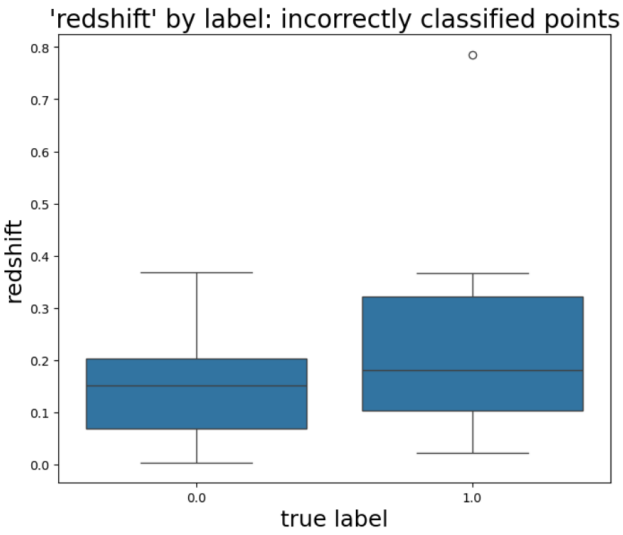
Bendras tikslumas: „RandomForest“ klasifikatorius originaliai duomenų aibei buvo testuotas su visiškai nematytais duomenimis. Tikslumo vertė apskaičiuota pagal šią formulę: teisingai klasifikuotų objektų dalis padalinta iš visų testavimo objektų padauginti iš šimto. Gautas bendras tikslumas yra 95.87%, tai yra šiek tiek mažiau, negu su apmokymo ir testavimo duomenimis apskaičiuotas modelio tikslumas, kuris buvo 96.5%, tačiau pakankamai aukštas.

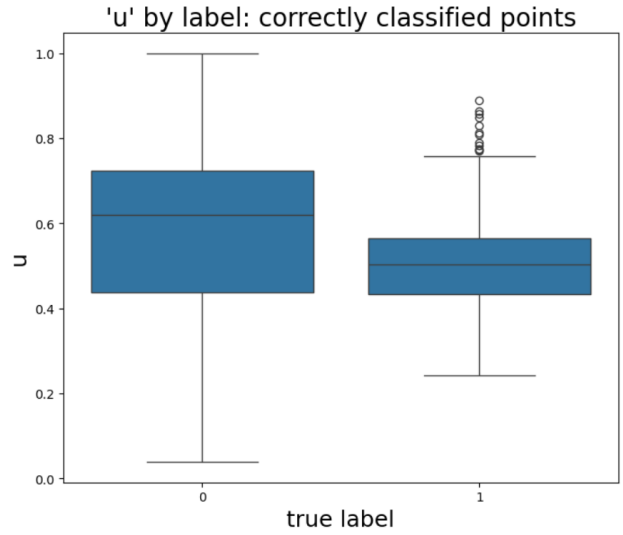
3.2.5. Neteisingai suklasifikuotų objektų analizė

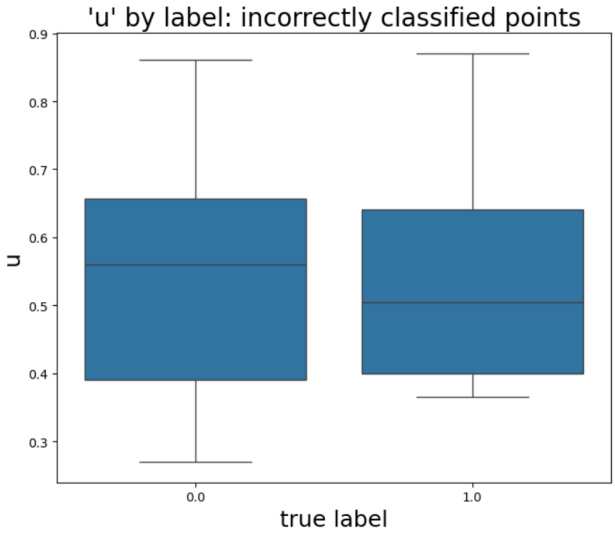
Šioje dalyje analizuojami „RandomForest“ optimalaus modelio (originaliai duomenų aibei) teisingai ir neteisingai suklasifikuoti objektai pagal šešis požymius (redshift, u, g, r, i, z) naudojant

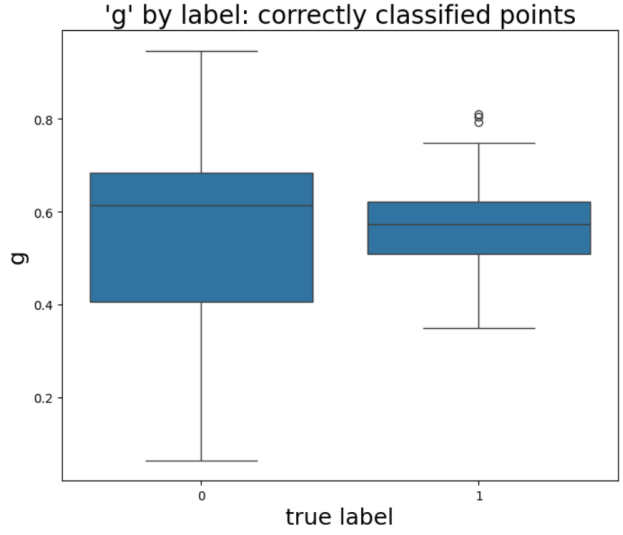
stačiakampes diagramas. Rezultatai rodo, kaip požymių pasiskirstymas skiriasi tarp klasių (0 ir 1) ir tarp teisingai bei neteisingai suklasifikuotų objektų.

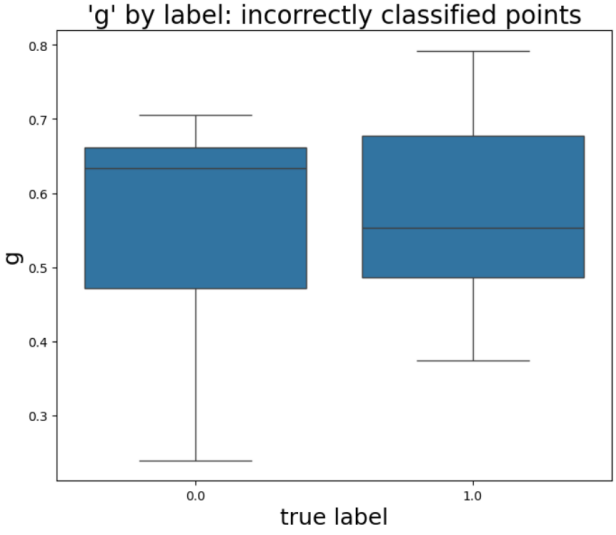


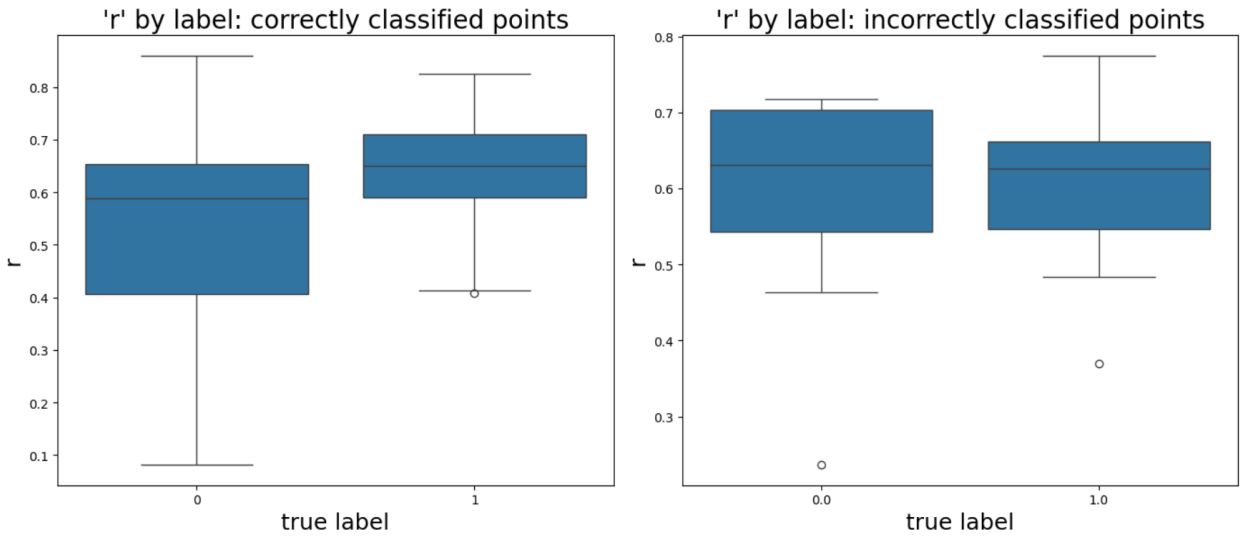


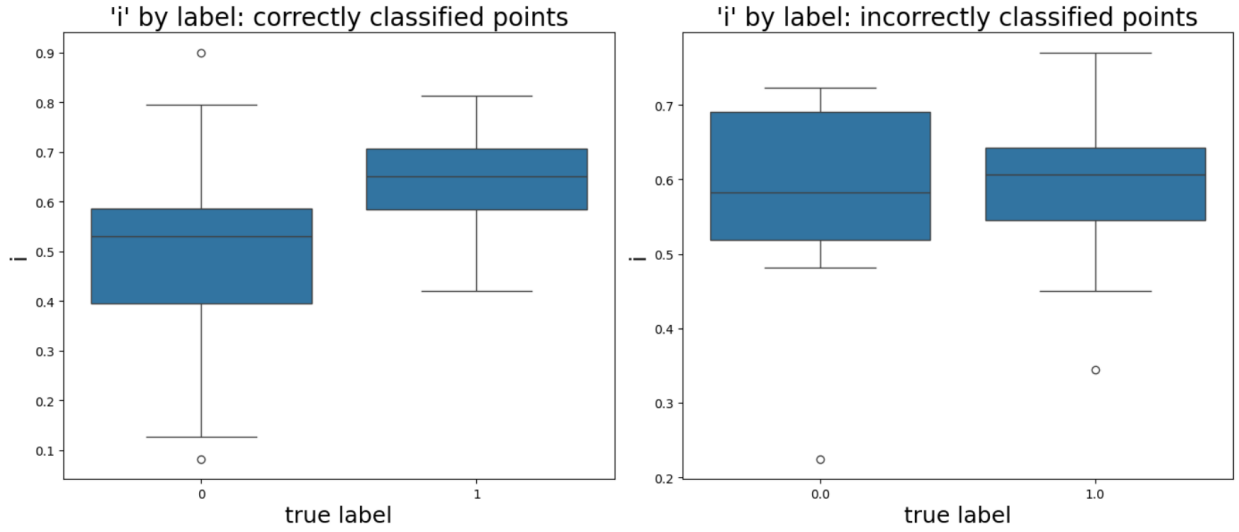


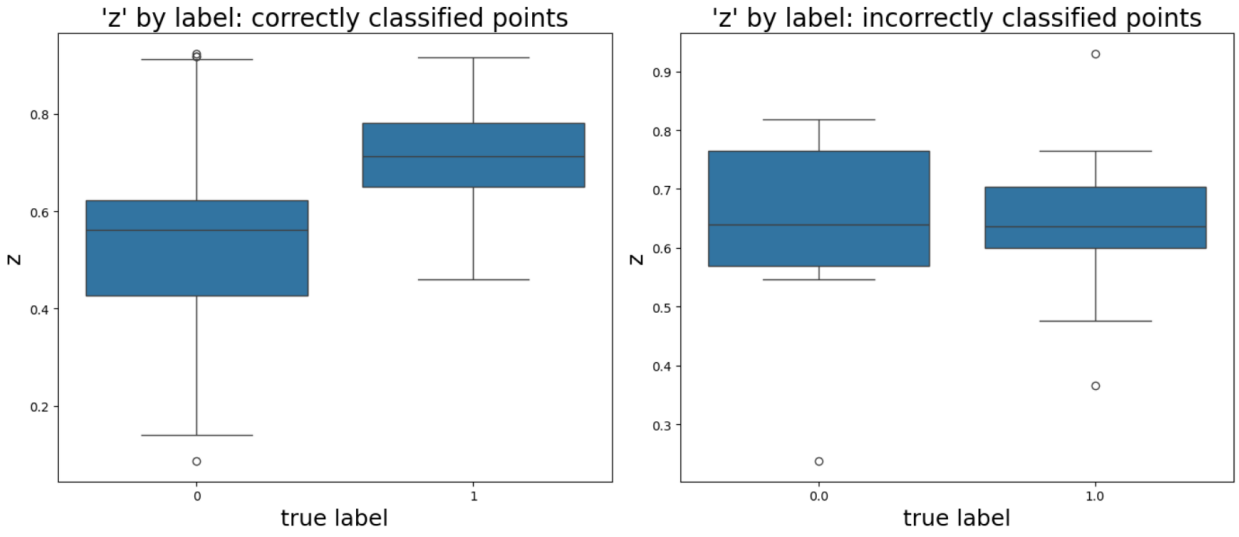












Iš pateiktų stačiakampių diagramų, galima daryti šias išvadas:

- Teisingai klasifikuoti objektai:
    - Skirtumai tarp 0 ir 1 klasės pastebimi redshift, i, ir z požymiuose, kur 1 klasė dažniausiai turi didesnes vertes.
- Neteisingai klasifikuoti objektai:
    - Dauguma požymių (u, g, r, i, z) rodo reikšmingą persidengimą tarp klasių, o tai apsunkina modelio gebėjimą atskirti objektus.

- Reikšmingiausias požymis:
    - Redshift ir z filtro vertės labiausiai prisideda prie klasifikacijos, tačiau dėl persidengimų modelis vis dar daro klaidų.

Šie rezultatai rodo, kad neteisingų klasifikacijų priežastis yra požymių pasiskirstymo panašumas tarp klasių, ypač pereinamojoje zonoje.

3.2.6. Išvados

Nustatant „RandomForest“ klasifikatoriaus parametrus buvo pastebėta, kad didėjant „n_estimators“ reikšmėms, klasifikavimo rezultatai stabilizuojasi, o optimalus medžio gylis („max_depth“) užtikrina aukštą „F1-Score“. Taip pat buvo pastebėta, kad sumažinus duomenų dimensiją naudojant „PCA“ metodą, klasifikatoriaus tikslumas nežymiai sumažėjo, tačiau modelio rezultatai tapo stabilesni dėl mažesnio standartinio nuokrypio.

Geriausias klasifikavimo rezultatas buvo gautas naudojant originalius duomenis – šiuo atveju AUC reikšmė siekė 0.99, o „F1-Score“ buvo didžiausias. Naudojant „PCA“ metodu transformuotus duomenis, AUC reikšmė buvo 0.98, tai rodo minimalų tikslumo praradimą, tačiau didesnį stabilumą.

4. IŠVADOS

Šiame darbe buvo atlikta duomenų klasifikacija naudojant „kNN“ ir „RandomForest“ algoritmus su originaliais ir sumažintos dimensijos duomenimis. Pateikiamos pagrindinės išvados:

- kNN klasifikatorius:

Naudojant originalius duomenis, „kNN“ algoritmas pasiekė aukštesnį tikslumą (94 %), palyginus su sumažintos dimensijos duomenimis (92 %). Originalūs duomenys užtikrina tikslesnį klasių atpažinimą, tačiau sumažintos dimensijos duomenys leido sumažinti klasifikavimo rezultatų svyravimus tarp iteracijų.

Optimalūs parametrai buvo nustatyti kaip: n_neighbors = 5, weights = uniform, p = 1.

- RandomForest klasifikatorius:

„RandomForest“ algoritmas pasiekė aukštesnį tikslumą (96 %) su originaliais duomenimis, palyginus su sumažintos dimensijos duomenimis (93 %). Tai rodo, kad originali duomenų aibė suteikia išsamesnę informaciją, kuri pagerina klasifikavimo tikslumą.

PCA sumažintos dimensijos duomenys buvo efektyvūs ir leido pasiekti panašų tikslumą, tačiau stabilumas padidėjo dėl mažesnio rezultatų nuokrypio.

Optimalūs parametrai buvo nustatyti kaip: n_estimators = 200, max_depth = 10, max_features = sqrt.

- Palyginimas:

„kNN“ algoritmas labiau tinka nedidelėms ar vidutinio dydžio duomenų aibėms, kuriose klasių ribos yra aiškiai apibrėžtos. Jo tikslumas labiau priklauso nuo atstumų tarp duomenų taškų. „RandomForest“ algoritmas pasižymi didesniu atsparumu triukšmui ir lankstumu, leidžiančiu efektyviai apdoroti didesnės apimties duomenis. PCA dimensijų mažinimas sumažino skaičiavimo sąnaudas ir pagerino stabilumą, tačiau gali sumažinti klasifikavimo tikslumą.

- Rekomendacijos:

Jei tikslas yra maksimalus klasifikavimo tikslumas, rekomenduojama naudoti originalius duomenis kartu su „RandomForest“ algoritmu. Jei svarbiau sumažinti duomenų sudėtingumą ir padidinti rezultatų stabilumą, verta naudoti PCA metodą ir „kNN“ algoritmą. Gauti rezultatai rodo, kad teisingai parinktas algoritmas ir duomenų apdorojimo metodas gali reikšmingai pagerinti klasifikavimo efektyvumą bei tikslumą.

ŠALTINIAI

- https://scikit-learn.org/stable/modules/clustering.html
- https://machinelearningmastery.com/clustering-algorithms-with-python/
- https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html
- https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.KMeans.html
- https://pandas.pydata.org/docs/
- https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss1

KODAS