HIGGS Dataset

Filip Ogrenjac

2023

Samostalni projekat za kurs *Istraživanje podataka 1* na
Matematičkom fakultetu Univerziteta u Beogradu

Sadržaj

1 Uvod

Skup podataka "HIGGS dataset" je kolekcija od 11 miliona uzoraka dobijenih Monte Karlo simulacijama. Uzorci se koriste za klasifikaciju procesa koji nastaju sudarom čestica u akceleratoru čestica u dve kategorije: signal, proces koji proizvodi Higsov bozon, i pozadina, proces koji ne proizvodi Higsov bozon. Ovaj rad obuhvata analizu i pretprocesiranje podataka. Pokazane su različite tehnike poput klasifikacije pomoću algoritama kao što su stabla odlučivanja i K najbližih suseda, klasterovanja pomoću algoritma K-sredina i algoritma Gausove mešavine, i pronalaženja pravila pridruživanja Apriori algorimom.

1.1 Analiza skupa podataka

Svaki uzorak skupa ima 28 osobina, od kojih 21 predstavlja izmerene kinematičke osobine, a preostalih 7 funkcije prvih 21 osobina.

- target (kategorija)
- lepton_pT
- lepton_eta
- lepton_phi
- missing_energy_magnitude
- missing_energy_phi
- jet_1_pt
- jet_1_eta
- jet_1_phi
- jet_1_b-tag
- jet_2_pt
- jet_2_eta
- jet_2_phi
- jet_2_b-tag
- jet_3_pt
- jet_3_eta
- jet_3_phi
- jet_3_b-tag
- jet_4_pt
- jet_4_eta
- jet_4_phi
- jet_4_b-tag
- m_jj
- m_jjj
- m_lv
- m_jlv
- m_bb
- m_wbb
- m_wwbb

1.1.1 Balansiranost klasa

Proverava se balansiranost klasa, tj. procenat pojavljivanja prve i druge klase u skupu. Velika nebalansiranost može dovesti do pristrasnosti modela prema brojnijoj klasi.

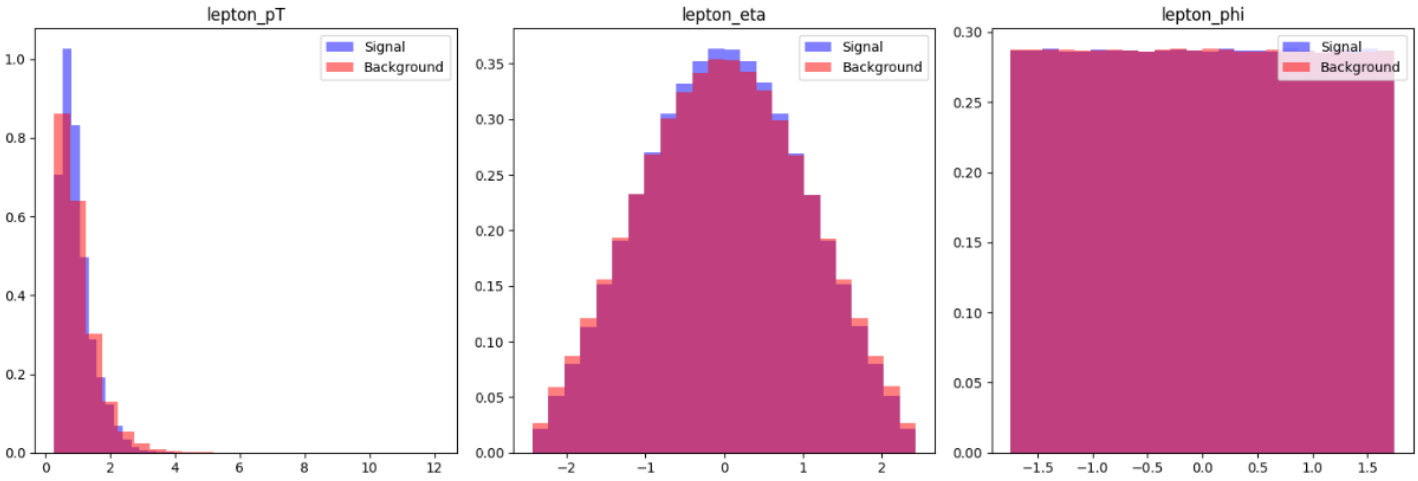
target	count	percentage
1.0	5829123	53.0%
0.0	5170877	47.0%

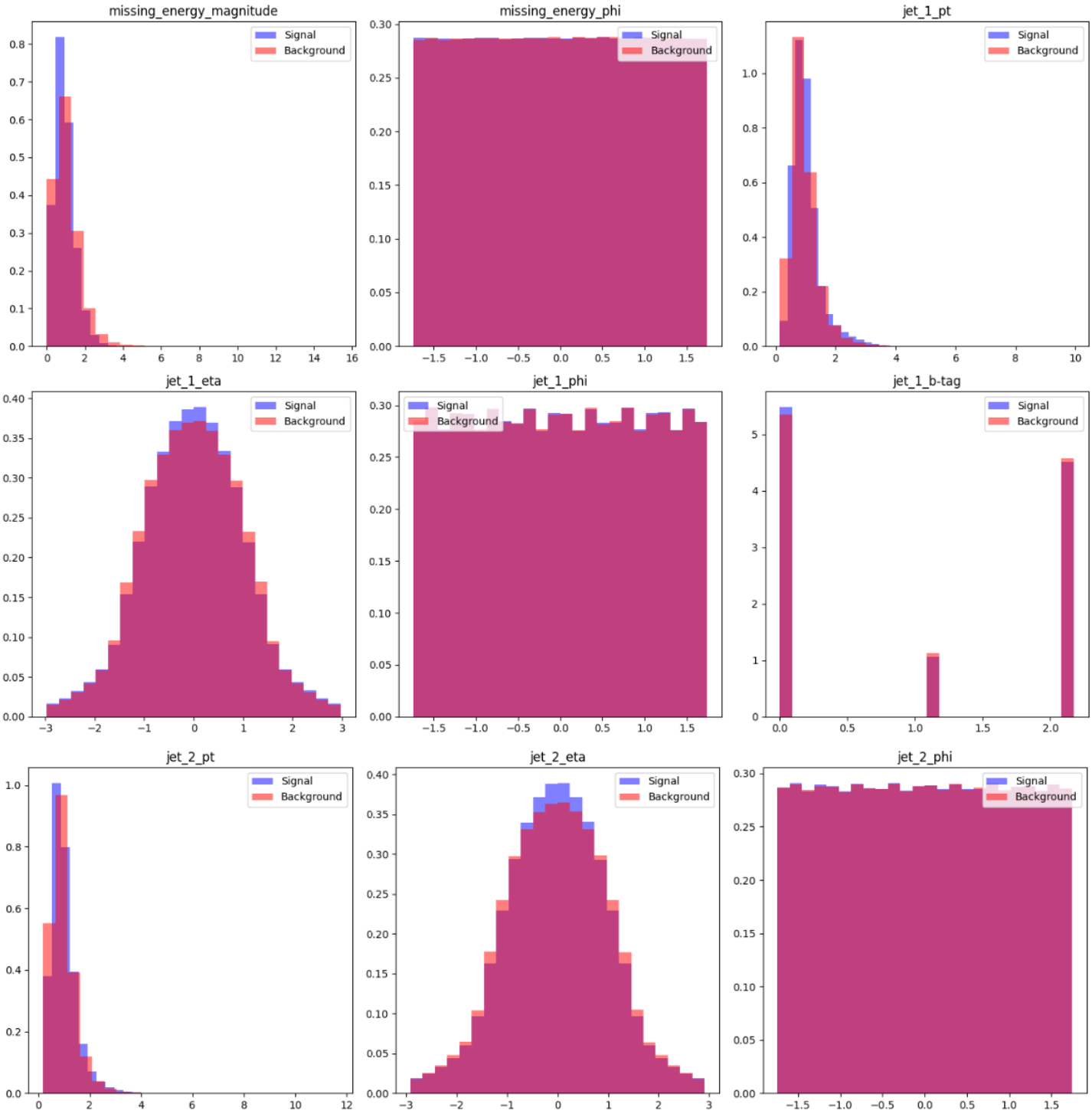
Slika 1.1: Balansiranost klasa.

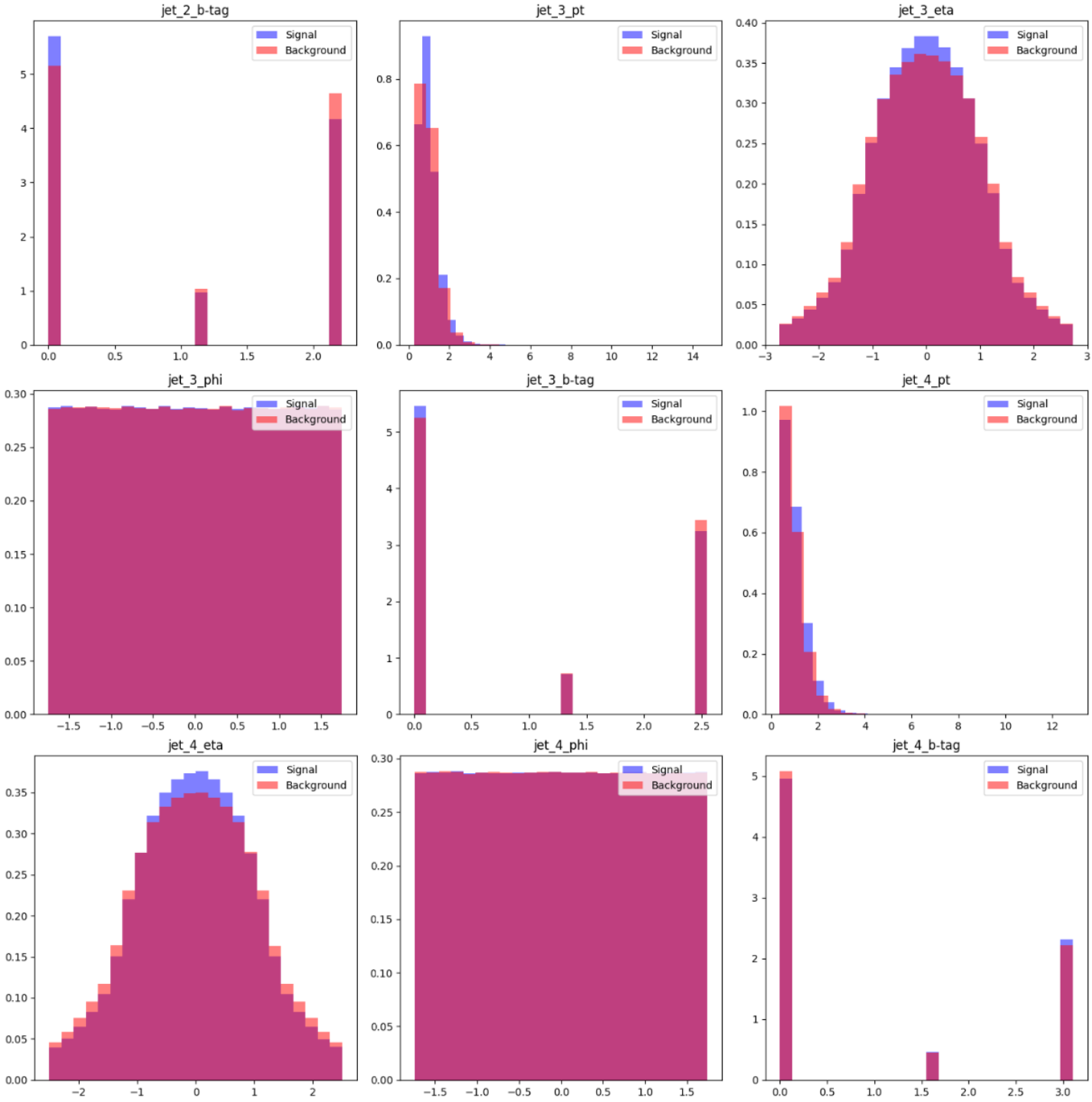
Može videti da je nebalansiranost veoma blaga sa odnosom 1.13:1 i neće praviti problem.

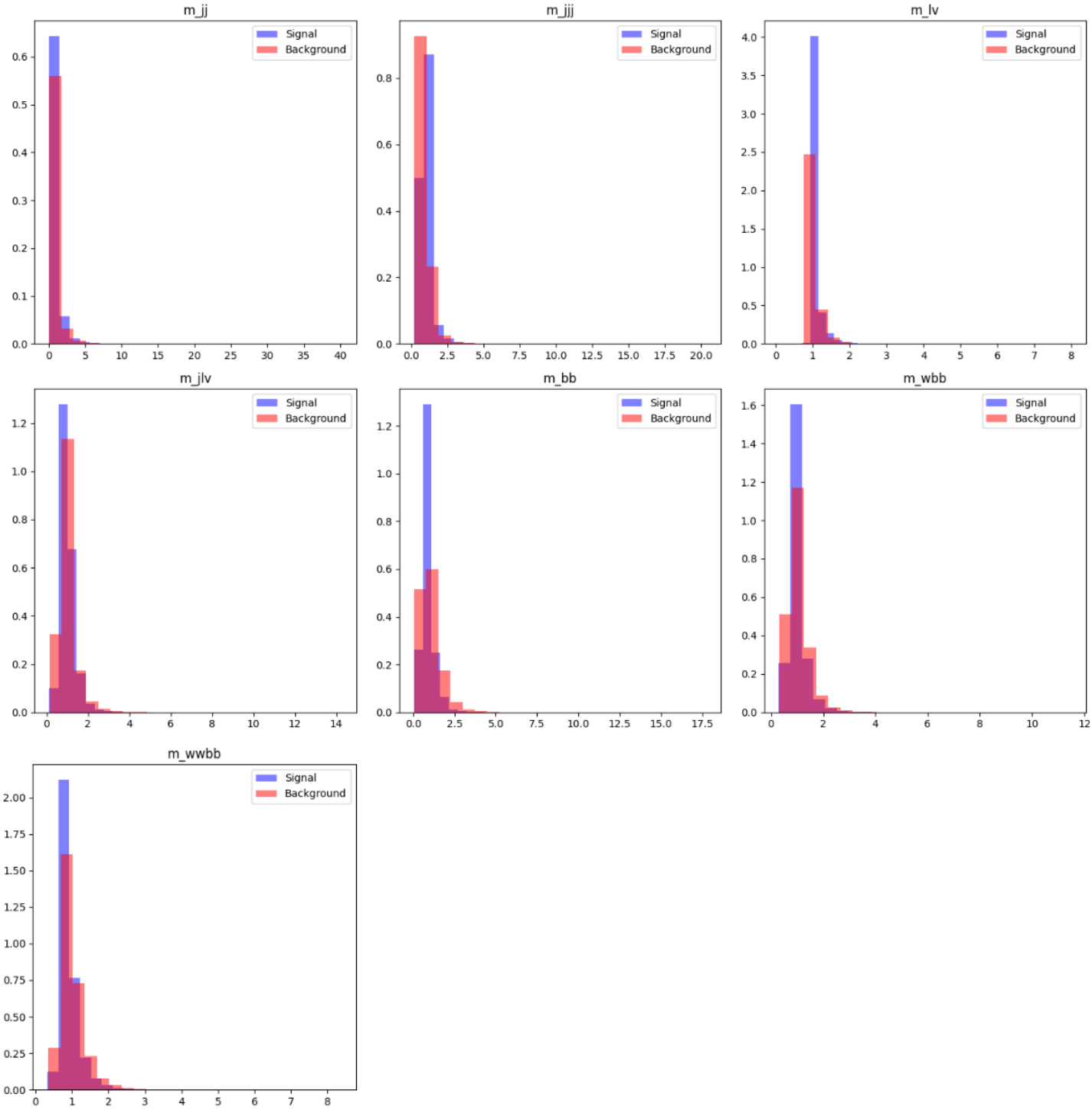
1.1.2 Histogrami atributa

Na narednim slikama prikazani su normalizovani histogrami atributa za dve klase. Normalizovani histogrami omogućavaju upoređivanje oblika raspodela direktno kao da je broj elemenata obe klase jednak i mogu nam ukazati na važnost atributa pri klasifikaciji.




missing_energy_magnitude
missing_energy_phi
jet_1_pt
jet_1_eta
jet_1_phi
jet_1_b-tag
jet_2_pt
jet_2_eta
jet_2_phi
Signal
Background

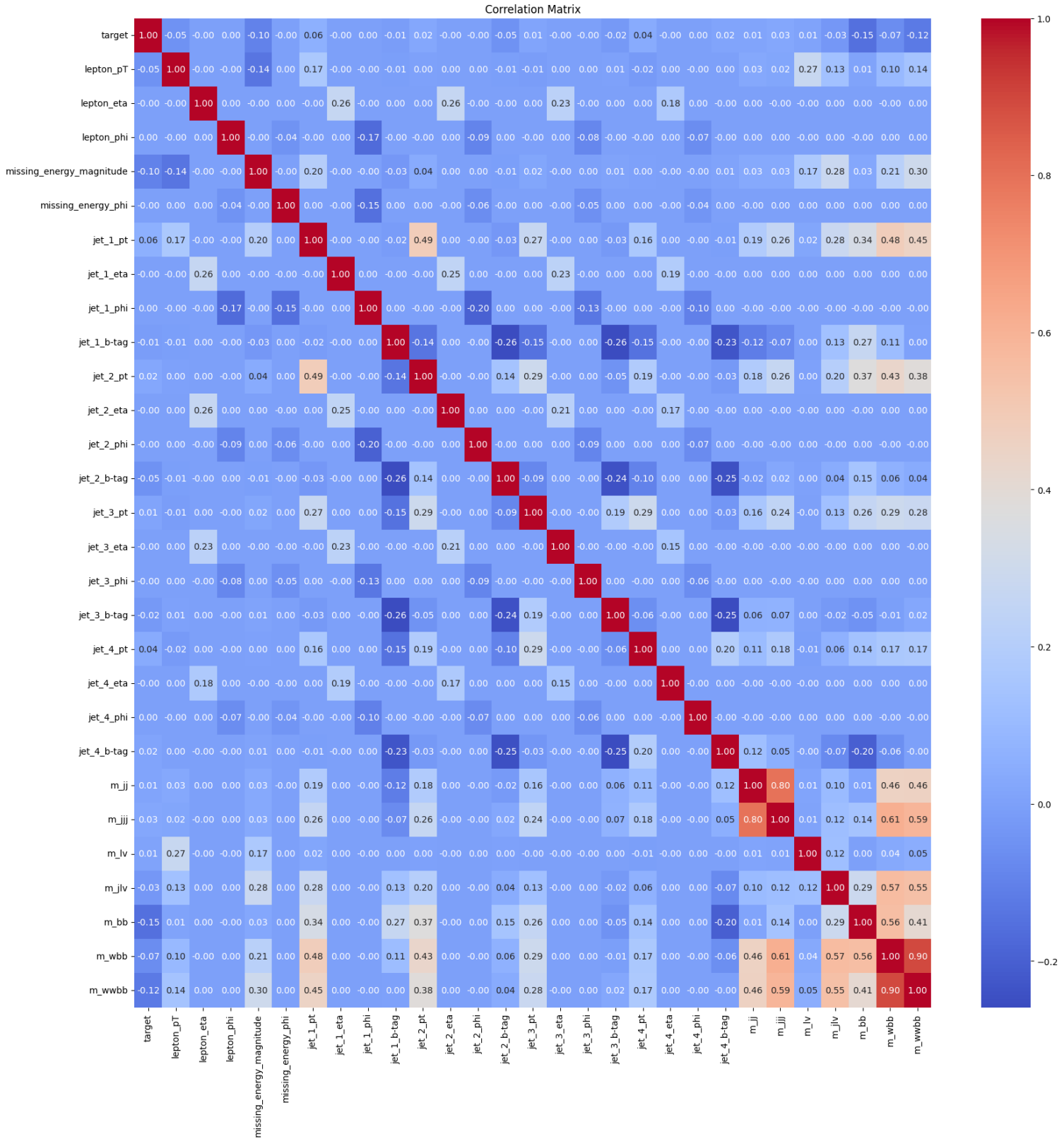

jet_2_b-tag
jet_3_pt
jet_3_eta
jet_3_phi
jet_3_b-tag
jet_4_pt
jet_4_eta
jet_4_phi
jet_4_b-tag
Signal
Background



Slika 1.2: Normalizovani histogrami atributa.

1.1.3 Matrica korelacije

Na sledećoj stranici može se pogledati matrica korelacije koja nam ukazuje na linearni odnos između atributa uzoraka. Vrednosti se kreću od -1 do 1. Pozitivna povezanost između dva atributa postoji kada porast vrednosti jednog atributa dovodi do veće vrednosti drugog atributa. Obrnuto važi za negativnu povezanost. Nula sugeriše da linearna povezanost ne postoji. Na primer, ako se pogleda matrica korelacije može se uočiti visoka korelacija između atributa m_jj i m_jjj kao i između atributa m_wbb i m_wwbb.



Slika 1.3: Matrica korelacije.

1.2 Pretprocesiranje

Pre bilo kakvog procesiranja podataka vrlo je važno podeliti skup podataka na skup za treniranje i skup za testiranje jer se podaci na kojima se testiraju modeli ne smeju dirati. Odvojeno je 30% podataka za test skup.

1.2.1 Nedostajuće vrednosti

Na pocetku pretprocesiranja proverava se prisutnost nedostajućih, tj. "null" vrednosti jer neki algoritmi ne podržavaju nedostajuće vrednosti.

```
1 print('Number of null values:\n')
2 df.isna().sum()
```

```
Number of null values:


target                      0
lepton_pT                   0
lepton_eta                  0
lepton_phi                  0
missing_energy_magnitude    0
missing_energy_phi          0
jet_1_pt                    0
jet_1_eta                   0
jet_1_phi                   0
jet_1_b-tag                 0
jet_2_pt                    0
jet_2_eta                   0
jet_2_phi                   0
jet_2_b-tag                 0
jet_3_pt                    0
jet_3_eta                   0
jet_3_phi                   0
jet_3_b-tag                 0
jet_4_pt                    0
jet_4_eta                   0
jet_4_phi                   0
jet_4_b-tag                 0
m_jj                        0
m_jjj                       0
m_lv                        0
m_jlv                       0
m_bb                        0
m_wbb                       0
m_wwbb                      0
```

Slika 1.4: Broj nedostajućih vrednosti za svaki atribut.

Na slici možemo videti da nema "null" vrednosti prisutnih u našim podacima.

1.2.2 Autlajeri i skaliranje

Neki algoritmi poput algoritama K najbližih suseda i K-sredina osetljivi su na prisustvo autlajera i oni se moraju ukloniti ili zameniti nekim drugim vrednostima. Takođe atributi moraju biti na sličnim skalama kako neki atribut ne bi "vukao" više tokom treniranja modela. Za identifikaciju autlajera korišćen je IQR (Interquartile Range), koji predstavlja raspon između prvog (Q1) i trećeg (Q3) kvartila, efikasno hvatajući središnjih 50% podataka. Vrednosti ispod $Q1 - 1.5 \times \text{IQR}$ i iznad $Q3 + 1.5 \times \text{IQR}$ smatraju se autlajerima. U narednoj tabeli mogu se videti granice i broj elemanata ispod i iznad granica, kao i minimumi, maksimumi i procenti autlajera svakog od atributa.

	lower	min	num_lower	upper	max	num_upper	percentage
lepton_pT	-0.377456	0.274697	0	2.204436	12.098914	301057	3.91%
lepton_eta	-2.955563	-2.434976	0	2.954480	2.434868	0	0.00%
lepton_phi	-3.485763	-1.742508	0	3.485936	1.743236	0	0.00%
missing_energy_magnitude	-0.497921	0.000394	0	2.367797	12.843856	227758	2.96%
missing_energy_phi	-3.485751	-1.743944	0	3.485697	1.743257	0	0.00%
jet_1_pt	-0.058903	0.137502	0	1.908820	9.940391	381233	4.95%
jet_1_eta	-2.750389	-2.969725	24164	2.751328	2.969674	24157	0.63%
jet_1_phi	-3.472709	-1.741237	0	3.472926	1.741454	0	0.00%
jet_1_b-tag	-3.259614	0.000000	0	5.432690	2.173076	0	0.00%
jet_2_pt	-0.161345	0.188981	0	2.019807	10.860058	308401	4.01%
jet_2_eta	-2.780497	-2.913090	17098	2.779646	2.913210	17198	0.45%
jet_2_phi	-3.480257	-1.742372	0	3.479950	1.743175	0	0.00%
jet_2_b-tag	-3.322308	0.000000	0	5.537180	2.214872	0	0.00%
jet_3_pt	-0.205565	0.263608	0	2.078215	11.155643	247327	3.21%
jet_3_eta	-2.799752	-2.729663	0	2.800098	2.730009	0	0.00%
jet_3_phi	-3.484928	-1.742069	0	3.485188	1.742884	0	0.00%
jet_3_b-tag	-3.822337	0.000000	0	6.370561	2.548224	0	0.00%
jet_4_pt	-0.286977	0.365354	0	2.125675	12.882567	259522	3.37%
jet_4_eta	-2.854546	-2.497265	0	2.855290	2.498009	0	0.00%
jet_4_phi	-3.486105	-1.742691	0	3.486232	1.743372	0	0.00%
jet_4_b-tag	-4.652942	0.000000	0	7.754903	3.101961	0	0.00%
m_jj	0.439426	0.075070	198187	1.375836	40.192368	876420	13.96%
m_jjj	0.490419	0.234753	13670	1.439329	20.372782	562017	7.48%
m_lv	0.933704	0.083049	20800	1.072493	7.992739	1510177	19.88%
m_jlv	0.205605	0.157473	5	1.704215	14.262439	393746	5.11%
m_bb	-0.023144	0.047862	0	1.835426	17.762852	466826	6.06%
m_wbb	0.337827	0.295112	16	1.622123	11.496522	482902	6.27%
m_wwbb	0.337072	0.347443	0	1.492587	8.374498	461756	6.00%

Slika 1.5: Pregled informacija o autlajerima.

Pored originalnog skupa za treniranje biće sačuvani i skalirani podaci u kojima su autlajeri uklonjeni i skalirani podaci u kojima su autlajeri zamenjeni granicama. Korišćen je StandardScaler koji standardizuje podatke tako da imaju srednju vrednost jednaku 0 i standardnu devijaciju jednaku 1. Za testiranje modela koji su trenirani na standardizovanim podacima koriste se test skupovi koji su skalirani tako što su se koristile vrednosti iz odgovarajućih skupova za treniranje (bez autlajera i sa zamenjenim autlajerima).

2 Klasifikacija

Zadatak klasifikacije predstavlja dodeljivanje određenih kategorija, tj. klasa ulaznim podacima. U ovom poglavlju fokus je na algoritmima: stabla odlučivanja i K najbližih suseda.

2.1 Stabla odlučivanja

Stabla odlučivanja su algoritmi koji dele prostor ulaznih podataka tako što postavljaju niz pitanja i koriste dobijene odgovore kako bi došli do zaključka o klasi podataka. Kod klasifikacije, stabla odlučivanja su robustna na autlajere i takođe ne mora se vršiti normalizacija podataka prilikom njihovog korišćenja.

2.1.1 GridSearch

Za treniranje modela korišćen je GridSearch koji koristi kros validaciju i vrši iscrpnu pretragu kombinacija prosleđenih vrednosti hiper-parametara. Hiper-parametri su parametri koji se ne uče tokom treniranja i prosleđuju se modelu.

```
1 param_grid_dt = {"max_depth": [4, 6, 8, 12],
2                  "criterion": ["gini", "entropy"]}
3 param_grid_rf = {"n_estimators": [100, 150],
4                  "max_depth": [4, 6, 8, 12],
5                  "criterion": ["gini", "entropy"]}
```

Slika 2.1: Parametri koji se prosleđuju GridSearch-u.

Na slici iznad mogu se videti svi GridSearch parametri za obična stabla odlučivanja i algoritam slučajne šume. Slučajne šume se sastoje od više stabala, treniranih na različitim podskupovima skupa za obučavanje, koji zajednički donose odluke.

- **max_depth**: Maksimalna dubina stabla. Predstavlja broj čvorova od korena do najdubljeg čvora. Moguće vrednosti su: 4, 6, 8, 12.
- **criterion**: Kriterijum koji se koristi za evaluaciju kvaliteta podele. Moguće vrednosti su:
  - *gini*: Gini nečistoća. Meri verovatnoću pogrešne klasifikacije.
  - *entropy*: Entropija. Pokazuje stepen nesigurnosti u skupu podataka.
- **n_estimators**: Broj stabala u šumi. Moguće vrednosti su: 100, 150.

Prvobitno treniranje modela sa GridSearch-om rađeno je na jednoj desetini skupa za treniranje. Na narednim slikama prikazani su najbolji parametri i izveštaji o klasifikaciji na train i test skupu za najbolji model određenim GridSearch-om za obična stabla odlučivanja i slučajne šume.

```
print("Best parameters for Decision Tree: ", grid_search_dt.best_params_)
print("Best score for Decision Tree: ", grid_search_dt.best_score_)
```

```
Best parameters for Decision Tree:  {'criterion': 'entropy', 'max_depth': 12}
Best score for Decision Tree:  0.7026714285714286
```

Slika 2.2: Najbolji parametri za DecisionTree.

```
Train set
----------------------------------------------------------------------
Classification report:
----------------------------------------------------------------------
              precision    recall  f1-score   support

         0.0       0.71      0.70      0.70    361961
         1.0       0.74      0.75      0.74    408039

    accuracy                           0.73    770000
   macro avg       0.72      0.72      0.72    770000
weighted avg       0.73      0.73      0.73    770000

----------------------------------------------------------------------
Confusion matrix:
----------------------------------------------------------------------
[[251796 110165]
 [101038 307001]]
```

Slika 2.3: Izveštaj o klasifikaciji na train skupu za DecisionTree.

```
Test set
----------------------------------------------------------------------
Classification report:
----------------------------------------------------------------------
              precision    recall  f1-score   support

         0.0       0.69      0.67      0.68   1551263
         1.0       0.72      0.73      0.72   1748737

    accuracy                           0.70   3300000
   macro avg       0.70      0.70      0.70   3300000
weighted avg       0.70      0.70      0.70   3300000

----------------------------------------------------------------------
Confusion matrix:
----------------------------------------------------------------------
[[1044445  506818]
 [ 468398 1280339]]
```

Slika 2.4: Izveštaj o klasifikaciji na test skupu za DecisionTree.

```
print("Best parameters for Random Forest: ", grid_search_rf.best_params_)
print("Best score for Random Forest: ", grid_search_rf.best_score_)
```

```
Best parameters for Random Forest:  {'criterion': 'gini', 'max_depth': 12, 'n_estimators': 150}
Best score for Random Forest:  0.7185207792207792
```

Slika 2.5: Najbolji parametri za RandomForest.

```
Train set
----------------------------------------------------------------------
Classification report:
----------------------------------------------------------------------
              precision    recall  f1-score   support

         0.0       0.74      0.70      0.72    361961
         1.0       0.74      0.78      0.76    408039

    accuracy                           0.74    770000
   macro avg       0.74      0.74      0.74    770000
weighted avg       0.74      0.74      0.74    770000

----------------------------------------------------------------------
Confusion matrix:
----------------------------------------------------------------------
[[252279 109682]
 [ 89192 318847]]
```

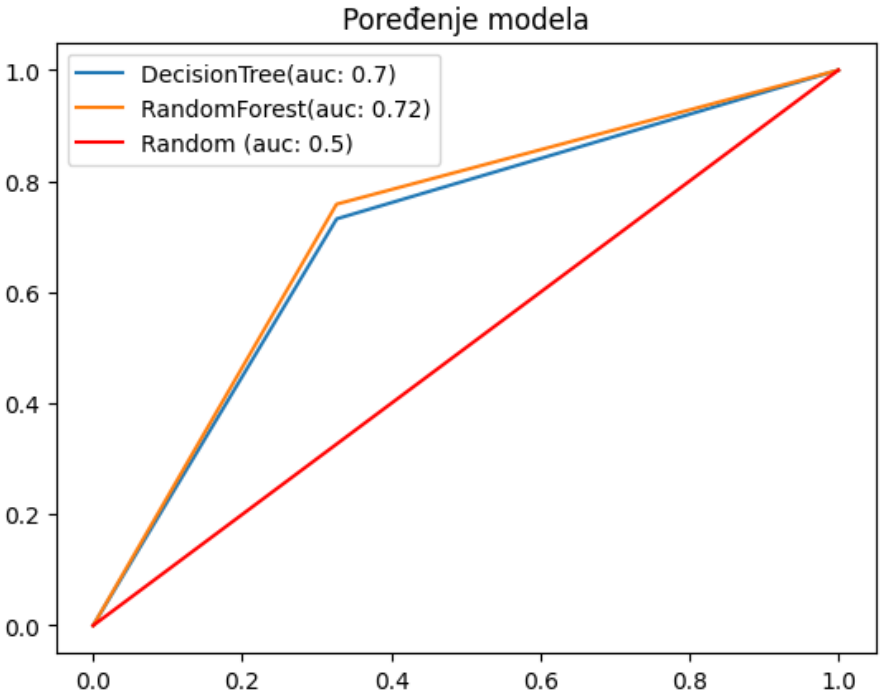
Slika 2.6: Izveštaj o klasifikaciji na train skupu za RandomForest.

```
Test set
----------------------------------------------------------------------
Classification report:
----------------------------------------------------------------------
              precision    recall  f1-score   support

         0.0       0.71      0.67      0.69   1551263
         1.0       0.72      0.76      0.74   1748737

    accuracy                           0.72   3300000
   macro avg       0.72      0.72      0.72   3300000
weighted avg       0.72      0.72      0.72   3300000

----------------------------------------------------------------------
Confusion matrix:
----------------------------------------------------------------------
[[1044794  506469]
 [ 422190 1326547]]
```

Slika 2.7: Izveštaj o klasifikaciji na test skupu za RandomForest.

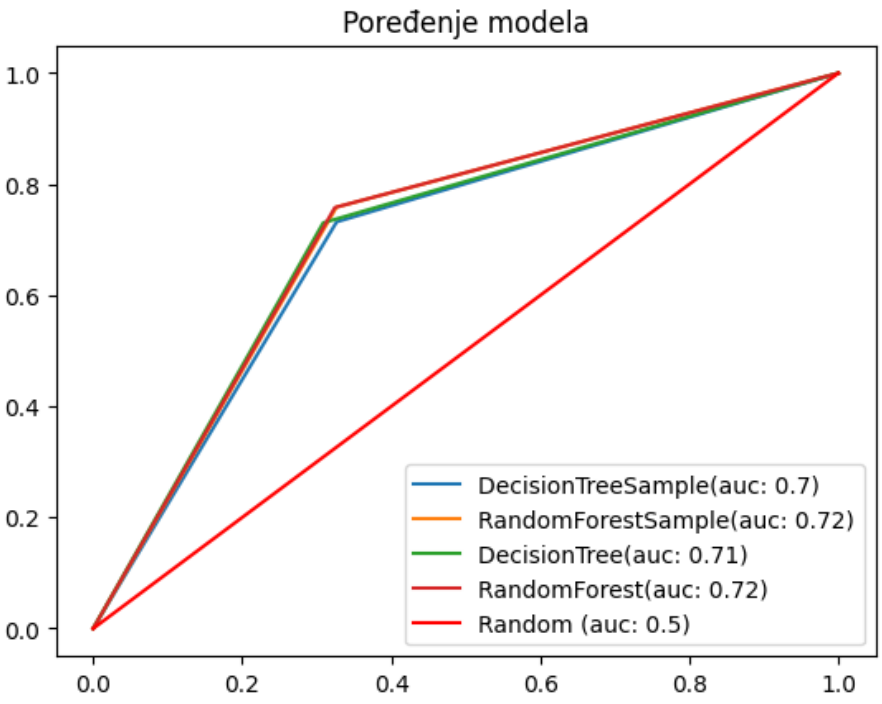
2.1.2 Poređenje DecisionTree i RandomForest modela

Ako se uporede izveštaji o klasifikaciji na test skupu za DecisionTree i RandomForest može se zaključiti da RandomForest nadmašuje DecisionTree. To je i očekivano jer slučajne šume, kao metoda ansambla koja se sastoji od više stabala odlučivanja, često pružaju bolju generalizaciju i otpornost na preprilagođavanje. Poboljšanja u tačnosti, F1-skoru i smanjeni broj lažno negativnih ukazuju da je model RandomForest bolji izbor. To nam potvrđuje i ROC kriva na sledećoj slici gde biramo model sa većom AUC vrednošću.



Slika 2.8: ROC kriva - uzorak 10%.

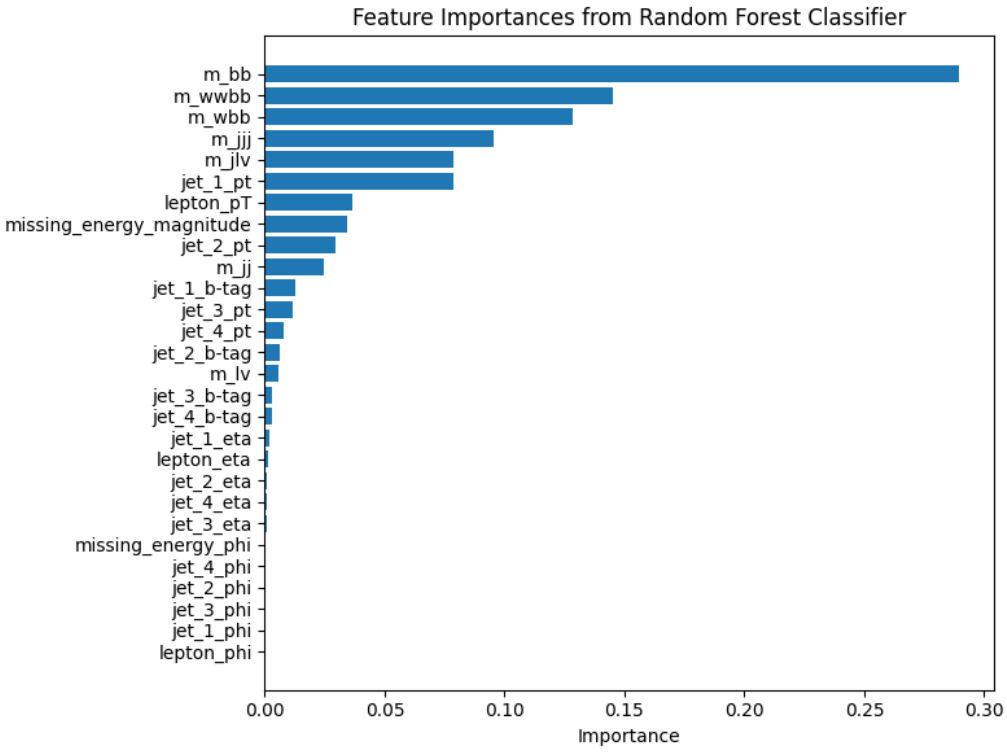
Najbolji rezultati klasifikacije dobili bi se za modele trenirane sa GridSearch-om na celom trening skupu. Prethodni modeli trenirani su na jednoj desetini trening skupa. Na narednoj slici prikazana je ROC kriva na kojoj su dodati modeli koji su trenirani na celom trening skupu koristeći najbolje parametre za podskup koji su dobijeni ranije. To nije idealno ali se može probati. Kod RandomForest-a nema nekog poboljšanja dok se kod DecisionTree-a može uočiti malo poboljšanje.



Slika 2.9: ROC kriva - uzorak 10% i ceo skup.

2.1.3 Značaj osobina

Na grafikonu ispod može se pogledati značaj svih osobina za RandomForest model. Kada se ovaj grafikon uporedi sa normalizovanim histogramima od ranije može se uočiti saglasnost o tome koji atributi igraju najveću ulogu u razlikovanju klasa.



Slika 2.10: Važnost osobina kod RandomForest modela.

2.2 K najbližih suseda

Algoritam K najbližih suseda (kNN) je algoritam zasnovan na principu sličnosti. Kada novi, nepoznati podatak treba da bude klasifikovan, algoritam pretražuje bazu podataka kako bi pronašao k najbližih tačaka (suseda) i na osnovu njihovih klasa donosi odluku o klasi novog podatka. KNN nije robustan na prisustvo autlajera, tj. ekstremnih vrednosti. Takve vrednosti mogu značajno uticati na rezultate KNN jer ovaj algoritam zavisi od računanja rastojanja između instanci. Takođe, kNN je osetljiv na neskalirane podatke. Atributi sa skalama većeg numeričkog opsega dominirali bi pri izračunavanju rastojanja, čineći druge karakteristike manje relevantnim, i tako algoritam ne bi ispravno radio. U nastavku biće upoređivani modeli koji su učeni na trening skupu bez autlajera i trening skupu sa zamenjenim autlajerima (sve skalirano), sa i bez GridSearch-a. Za to će biti korišćen 1% od trening skupova jer je kNN znatno sporiji za predviđanje klasa kad su u pitanju veliki skupovi podataka, u odnosu na stabla odlučivanja.

```
1 knn_params = {
2     'n_neighbors': [3, 5, 7],
3     'weights': ['uniform', 'distance'],
4     'p': [1, 2]
5 }
```

Slika 2.11: Parametri koji se prosleđuju GridSearch-u.

- **n_neighbors**: Broj suseda koji se razmatraju. Moguće vrednosti su: 3, 5, 7.
- **weights**: Moguće vrednosti su:
  - *uniform*: Svi susedi imaju jednaku težinu.
  - *distance*: Težine suseda su obrnuto proporcionalne njihovim udaljenostima. Susedi koji su bliži ciljnom uzorku imaju veći uticaj.
- **p**: Metrika udaljenosti. Moguće vrednosti su:
  - *1*: Menhetn udaljenost.
  - *2*: Euklidska udaljenost.

2.2.1 KNN na skupu bez autlajera

Slede izveštaji o klasifikaciji na train i test skupu za pristup bez autlajera.

Bez GridSearch-a

```
Train set
---------------------------------------------------------------------
Classification report:
---------------------------------------------------------------------
              precision    recall  f1-score   support

         0.0       0.76      0.70      0.73     15558
         1.0       0.77      0.82      0.79     19371

    accuracy                           0.76     34929
   macro avg       0.76      0.76      0.76     34929
weighted avg       0.76      0.76      0.76     34929

---------------------------------------------------------------------
Confusion matrix:
---------------------------------------------------------------------
[[10865  4693]
 [ 3517 15854]]
```

Slika 2.12: Izveštaj o klasifikaciji na train skupu za kNN bez autlajera.

```
Test set
---------------------------------------------------------------------
Classification report:
---------------------------------------------------------------------
              precision    recall  f1-score   support

         0.0       0.61      0.52      0.56   1551263
         1.0       0.62      0.70      0.66   1748737

    accuracy                           0.62   3300000
   macro avg       0.62      0.61      0.61   3300000
weighted avg       0.62      0.62      0.61   3300000

---------------------------------------------------------------------
Confusion matrix:
---------------------------------------------------------------------
[[ 810444  740819]
 [ 520021 1228716]]
```

Slika 2.13: Izveštaj o klasifikaciji na test skupu za kNN bez autlajera.

Sa GridSearch-om

```
1 knn_grid_no_outliers.best_estimator_
```

```
KNeighborsClassifier(n_neighbors=7, p=1, weights='distance')
```

```
1 knn_grid_no_outliers.best_score_
```

```
0.6509490681095937
```

Slika 2.14: Najbolji parametri i skor za GridSearch bez autlajera.

```
Train set
---------------------------------------------------------------
Classification report:
---------------------------------------------------------------
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     15558
         1.0       1.00      1.00      1.00     19371

    accuracy                           1.00     34929
   macro avg       1.00      1.00      1.00     34929
weighted avg       1.00      1.00      1.00     34929

---------------------------------------------------------------
Confusion matrix:
---------------------------------------------------------------
[[15558     0]
 [    0 19371]]
```

Slika 2.15: Izveštaj o klasifikaciji na train skupu za kNN (GridSearch best estimator) bez autlajera.

```
Test set
---------------------------------------------------------------
Classification report:
---------------------------------------------------------------
              precision    recall  f1-score   support

         0.0       0.64      0.54      0.59   1551263
         1.0       0.64      0.73      0.68   1748737

    accuracy                           0.64   3300000
   macro avg       0.64      0.63      0.63   3300000
weighted avg       0.64      0.64      0.64   3300000

---------------------------------------------------------------
Confusion matrix:
---------------------------------------------------------------
[[ 842954  708309]
 [ 478460 1270277]]
```

Slika 2.16: Izveštaj o klasifikaciji na test skupu za kNN (GridSearch best estimator) bez autlajera.

2.2.2 KNN na skupu sa zamenjenim autlajerima

Slede izveštaji o klasifikaciji na train i test skupu za pristup sa zamenjenim autlajerima.

Bez GridSearch-a

```
Train set
-----------------------------------------------------------------------
Classification report:
-----------------------------------------------------------------------
              precision    recall  f1-score   support

         0.0       0.77      0.71      0.74     36196
         1.0       0.76      0.81      0.78     40804

    accuracy                           0.76     77000
   macro avg       0.77      0.76      0.76     77000
weighted avg       0.76      0.76      0.76     77000

-----------------------------------------------------------------------
Confusion matrix:
-----------------------------------------------------------------------
[[25726 10470]
 [ 7677 33127]]
```

Slika 2.17: Izveštaj o klasifikaciji na train skupu za kNN sa zamenjenim autlajerima.

```
Test set
-----------------------------------------------------------------------
Classification report:
-----------------------------------------------------------------------
              precision    recall  f1-score   support

         0.0       0.62      0.56      0.59   1551263
         1.0       0.64      0.70      0.67   1748737

    accuracy                           0.63   3300000
   macro avg       0.63      0.63      0.63   3300000
weighted avg       0.63      0.63      0.63   3300000

-----------------------------------------------------------------------
Confusion matrix:
-----------------------------------------------------------------------
[[ 871538  679725]
 [ 530914 1217823]]
```

Slika 2.18: Izveštaj o klasifikaciji na test skupu za kNN sa zamenjenim autlajerima.

Sa GridSearch-om

```
1 knn_grid_replaced.best_estimator_
```

```
KNeighborsClassifier(n_neighbors=7, p=1, weights='distance')
```

```
1 knn_grid_replaced.best_score_
```

```
0.6481818231230805
```

Slika 2.19: Najbolji parametri i skor za GridSearch sa zamenjenim autlajerima.

```
Train set
---------------------------------------------------------------
Classification report:
---------------------------------------------------------------
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     36196
         1.0       1.00      1.00      1.00     40804

    accuracy                           1.00     77000
   macro avg       1.00      1.00      1.00     77000
weighted avg       1.00      1.00      1.00     77000

---------------------------------------------------------------
Confusion matrix:
---------------------------------------------------------------
[[36196     0]
 [    0 40804]]
```

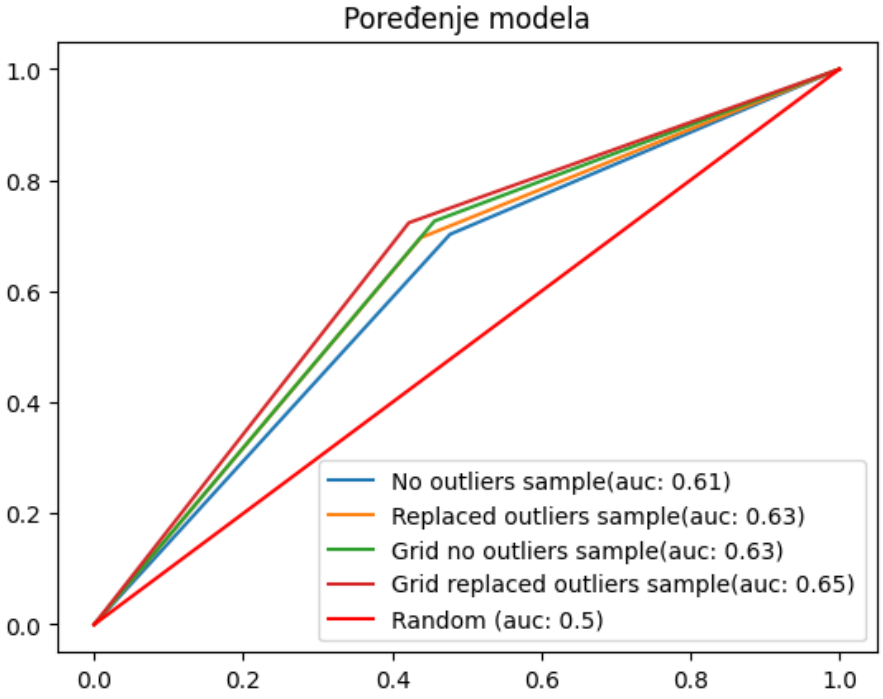
Slika 2.20: Izveštaj o klasifikaciji na train skupu za kNN (GridSearch best estimator) sa zamenjenim autlajerima.

```
Test set
---------------------------------------------------------------
Classification report:
---------------------------------------------------------------
              precision    recall  f1-score   support

         0.0       0.65      0.58      0.61   1551263
         1.0       0.66      0.72      0.69   1748737

    accuracy                           0.65   3300000
   macro avg       0.65      0.65      0.65   3300000
weighted avg       0.65      0.65      0.65   3300000

---------------------------------------------------------------
Confusion matrix:
---------------------------------------------------------------
[[ 896265  654998]
 [ 483690 1265047]]
```

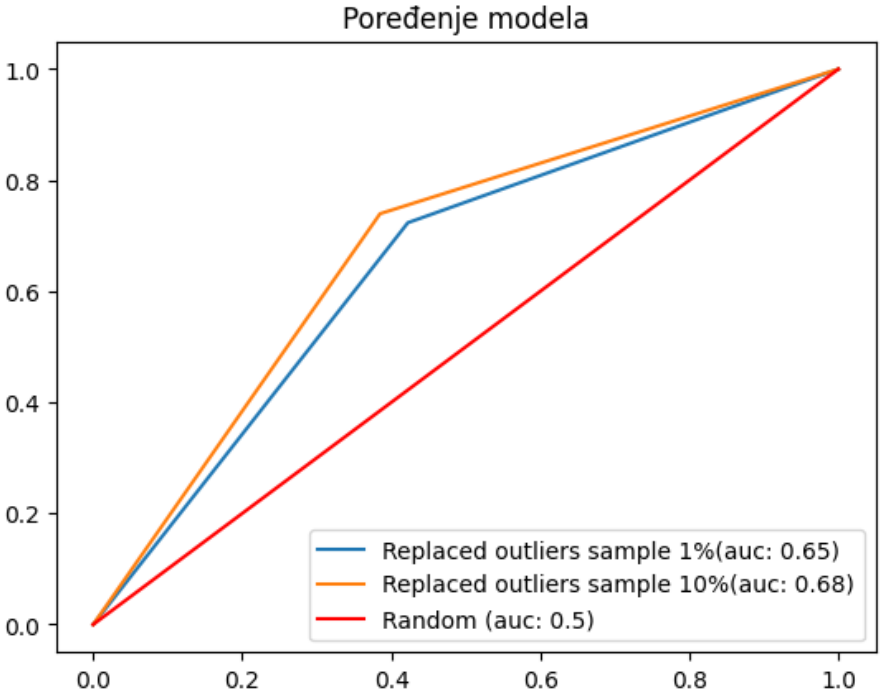
Slika 2.21: Izveštaj o klasifikaciji na test skupu za kNN (GridSearch best estimator) sa zamenjenim autlajerima.

2.2.3 Poređenje KNeighbors modela

Sa naredne slike može se zaključiti da je model koji se najbolje pokazao: kNN model koji je treniran na skaliranom trening skupu sa zamenjenim autlajerima sa parametrima iz GridSearch-a. Takođe, upoređen je model koji je treniran na 1% od trening skupa sa zamenjenim autlajerima sa onim koji je treniran na 10% (sa istim parametrima).



Slika 2.22: ROC kriva.



Slika 2.23: ROC kriva - uzorak 1% i 10%.

3 Klasterovanje

Klasterovanje predstavlja jednu od osnovnih metoda nenadgledanog učenja. Kod klasterovanja cilj je grupisati slične objekte u iste klastere, dok se objekti iz različitih klastera međusobno značajno razlikuju. Postoji više algoritama za klasterizaciju. U nastavku prikazani su algoritam K-sredina i algoritam Gausove mešavine. Pre korišćenja svakog od algoritma rađena je analiza glavnih komponenti (PCA). Kako ni jedan od narednih algoritama nije robustan na autlajere i neskalirane podatke korišćen je skaliran skup podataka sa zamenjenim autlajerima.

3.1 Analiza glavnih komponenti (PCA)

PCA je statistička metoda i tehnika smanjenja dimenzionalnosti koja transformiše originalne promenljive skupa podataka u novi skup nekorelisanih promenljivih nazvanih glavne komponente. PCA smanjuje dimenzionalnost velikih skupova podataka i olakšava vizuelizaciju podataka i poboljšava performanse algoritama. Varijansa, u kontekstu PCA, odnosi se na količinu zahvaćenih informacija (ili varijabilnosti) iz originalnih podataka. Cilj je zahvatiti bar 90% varijanse kako bi zadržali većinu bitnih informacija. Rađenjem PCA na 3 dimenzije zahvata se samo oko 27% varijacije.

```
explained_variance_ratio = pca.explained_variance_ratio_
explained_variance_ratio
```

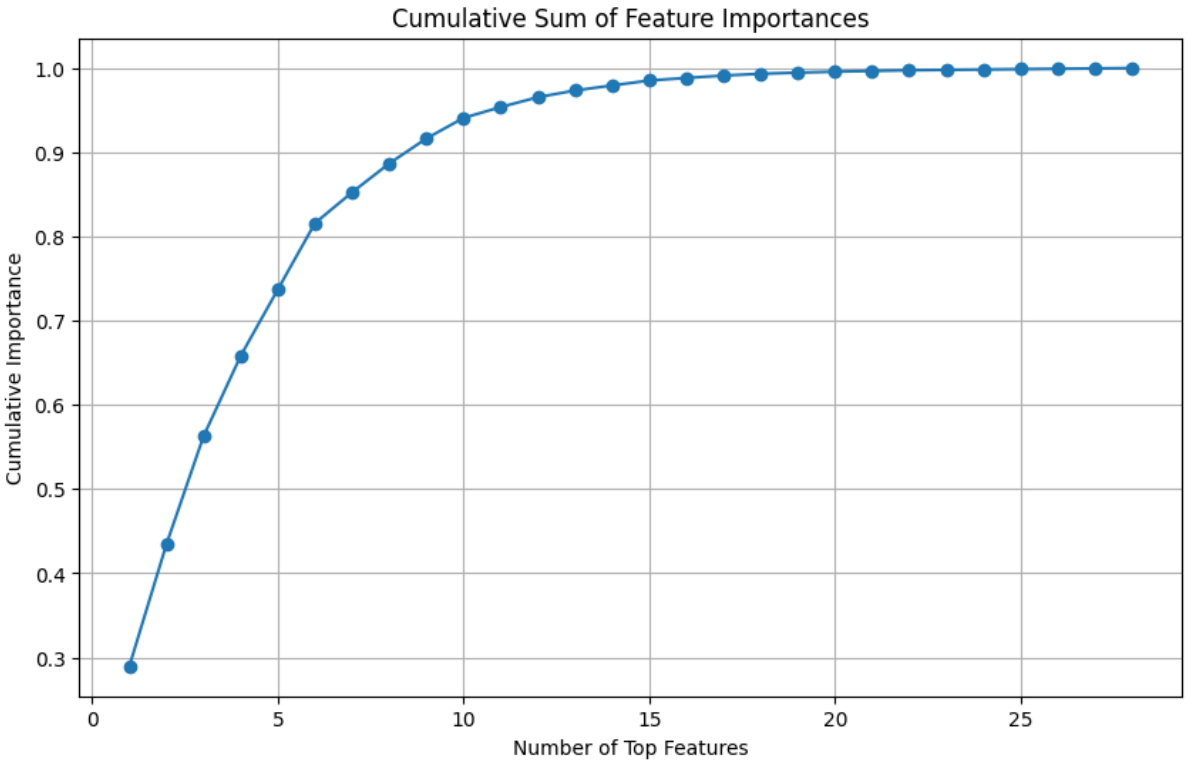
```
array([0.13467162, 0.06706966, 0.06485308])
```

```
sum(explained_variance_ratio)
```

```
0.26659435448558816
```

Slika 3.1: Ukupna varijansa korišćenjem svih atributa za PCA na 3 dimenzije.

Dobijeni rezultat nije zadovoljavajući. Naredna ideja je da se umesto svih atributa koriste samo oni koji su najvažniji za klasifikaciju. Mogu se izabrati pomoću normalizovanih histograma ili uz pomoć već rađenih stabala odlučivanja. Biće izabrani najvažnijih 5 atributa za algoritam RandomForest.



Slika 3.2: Kumulativna suma značaja atributa za RandomForest model.

```
explained_variance_ratio = pca.explained_variance_ratio_
explained_variance_ratio
```

```
array([0.5698167 , 0.19029378, 0.14908575])
```

```
sum(explained_variance_ratio)
```

```
0.9091962345452064
```

Slika 3.3: Ukupna varijansa korišćenjem najvažnijih 5 atributa za PCA na 3 dimenzije.

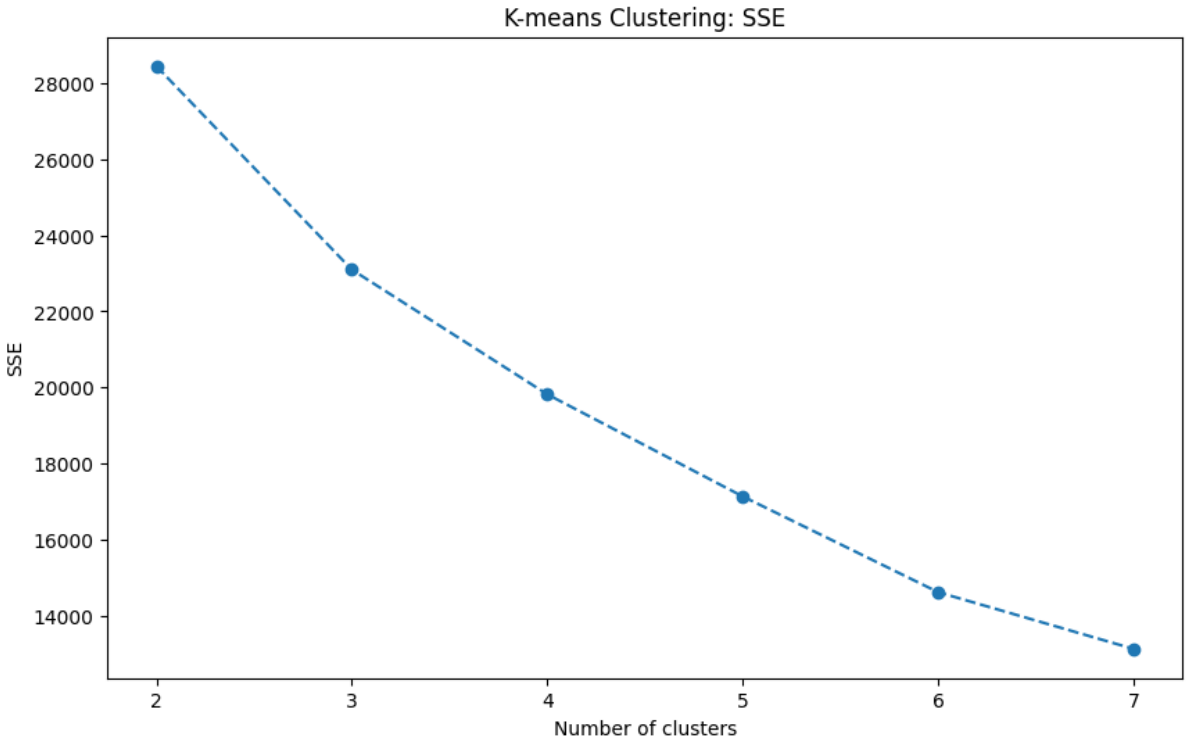
Dobijeni rezultati su zadovoljavajući za naše potrebe ali nisu savršeni. Prvih 5 atributa imaju kumulativnu sumu važnosti oko 75% i dobijena ukupna varijansa iznosi oko 91%. Može se nastaviti sa klasterovanjem.

3.2 K-sredina

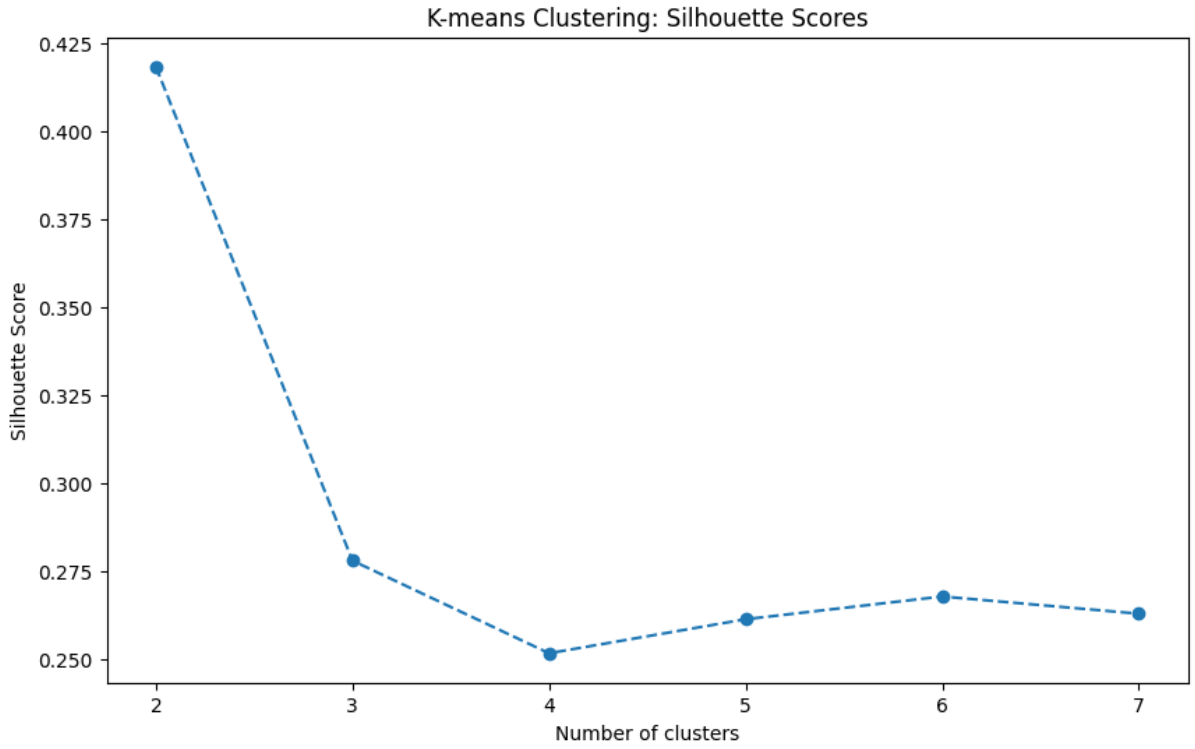
Algoritam K-sredina (K-means) je algoritam klasterovanja koji deli skup podataka na k različitih podskupova (klastera) koji se ne preklapaju. Ukratko, K-means dodeljuje svaku tačku podataka najbližem centroidu, a zatim ponovo izračunava centroid kao srednju vrednost svih tačaka dodeljenih njemu. Proces se ponavlja sve dok se dodeljivanje klastera više ne menja ili dok se ne dostigne maksimalan broj iteracija.

3.2.1 Biranje optimalnog broja klastera za K-sredina

U okviru algoritma K-means vrši se minimizacija zbira kvadratnih grešaka:
$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, c_i)^2$. Optimalan broj klastera bira se uz pomoć SSE i silhouette koeficijenata. Za raspon brojeva klastera plotuje se SSE i silhouette koeficijenti i traži se vrednost za koju se stopa smanjenja SSE naglo menja (podsećajući na lakat). Ako nije baš najjasnije koja je to tačka uzima se ona sa većim silhouette skorom koji predstavlja razdvojenost klastera. Naš algoritam rađen je za broj klatera u rasponu od 2 do 7. U nastavku prikazani su sse i silhouette score za različito k.

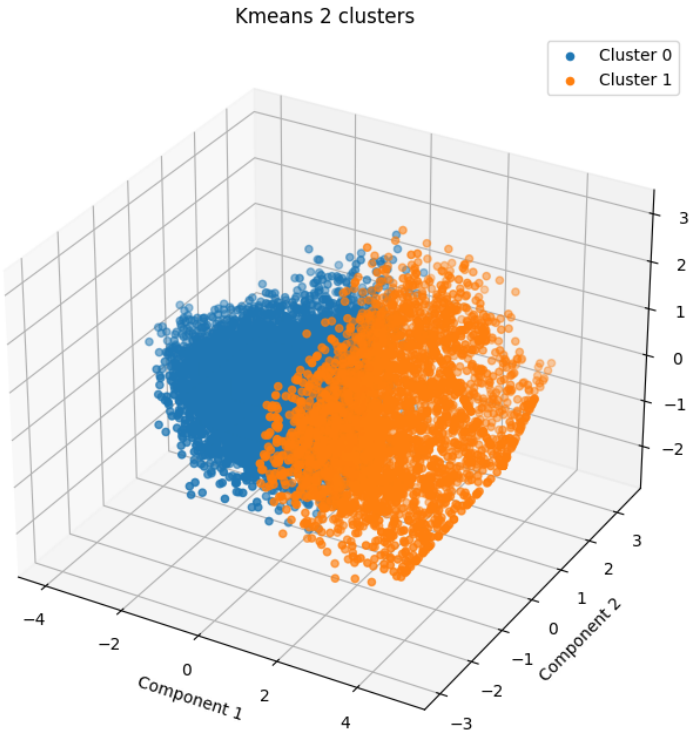


Slika 3.4: SSE u zavisnosti od broja k.

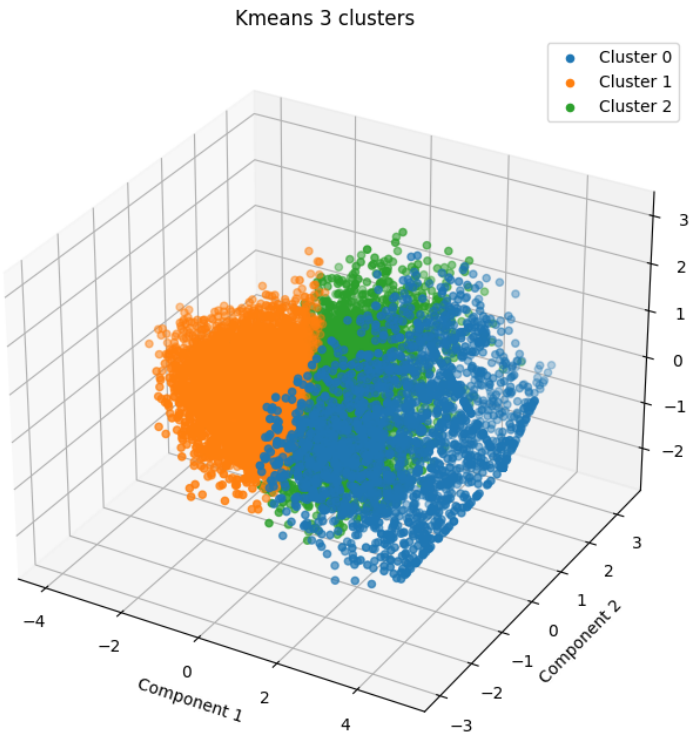


Slika 3.5: Silhouette score u zavisnosti od broja k.

Kada pogledamo grafik sse ne vidimo izraženi lakat dok nam grafik silhouette koeficijenata govori da je za broj klastera najbolje uzeti 2. U nastavku prikazano je klasterovanje na dva i tri klastera.



Slika 3.6: K-means: $k = 2$.



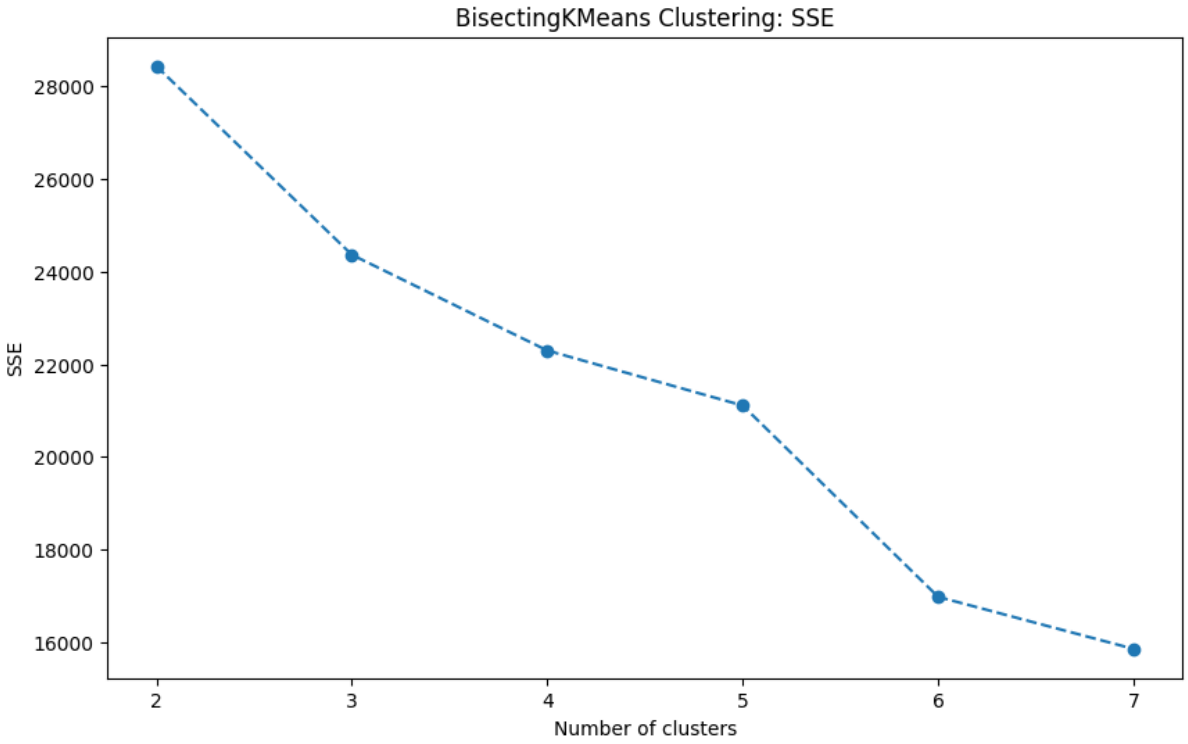
Slika 3.7: K-means: $k = 3$.

3.3 K-sredina sa bisekcijom

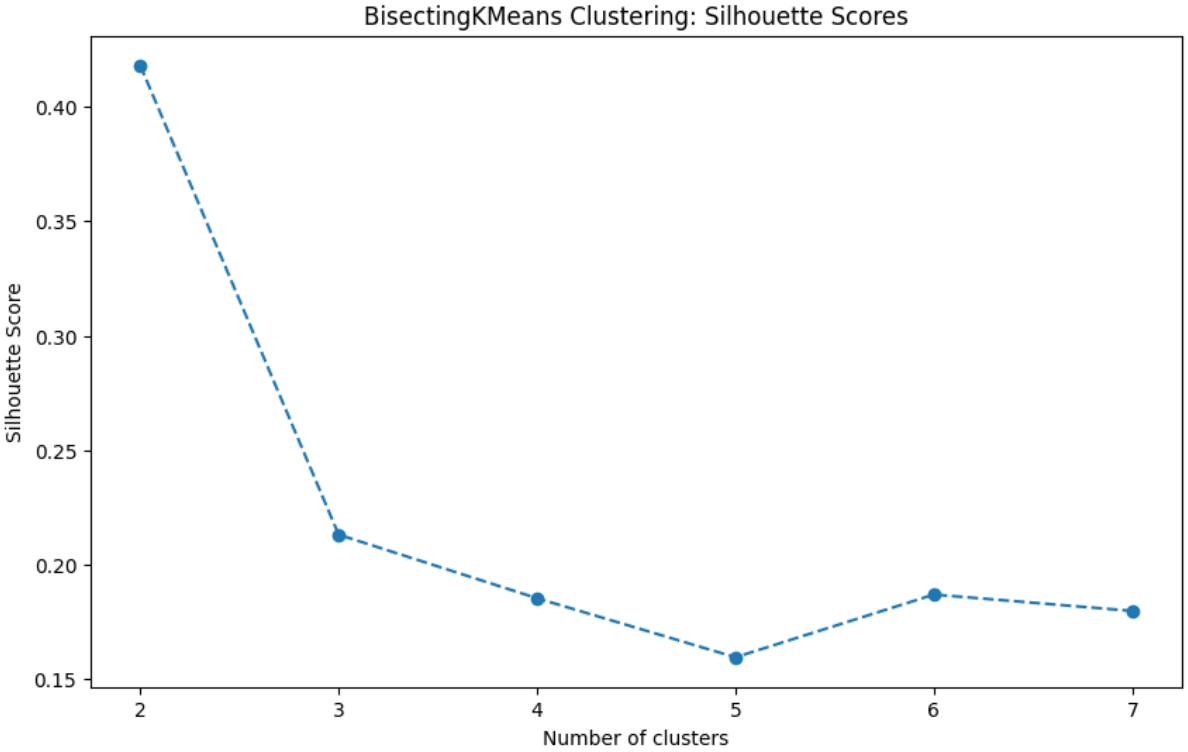
Algoritam Bisecting K-means je varijacija tradicionalnog K-means algoritma. Zasniva se na ideji deljenja instanci u dva klastera, zatim biranjem jednog od postojećih klastera i njegovim deljenjem na dva. Postupak se ponavlja sve dok se ne formira k klastera. Za klaster koji se deli može se izabrati najveći klaster, klaster sa najvećom SSE ili klaster izabran metodom zasnivanim sa SSE i veličini. Kod nas korišćena je strategija najvećeg klastera. Centri klastera dobijenim algoritmom Bisecting K-means često se koriste kao početni centri za algoritam K-means, kako bi se brže konvergiralo ka rešenju.

3.3.1 Biranje optimalnog broja klastera za K-sredina sa bisekcijom

Algoritam je takođe rađen za broj klastera u rasponu od 2 do 7. U nastavku su prikazani sse i silhouette koeficijenti za različito k.

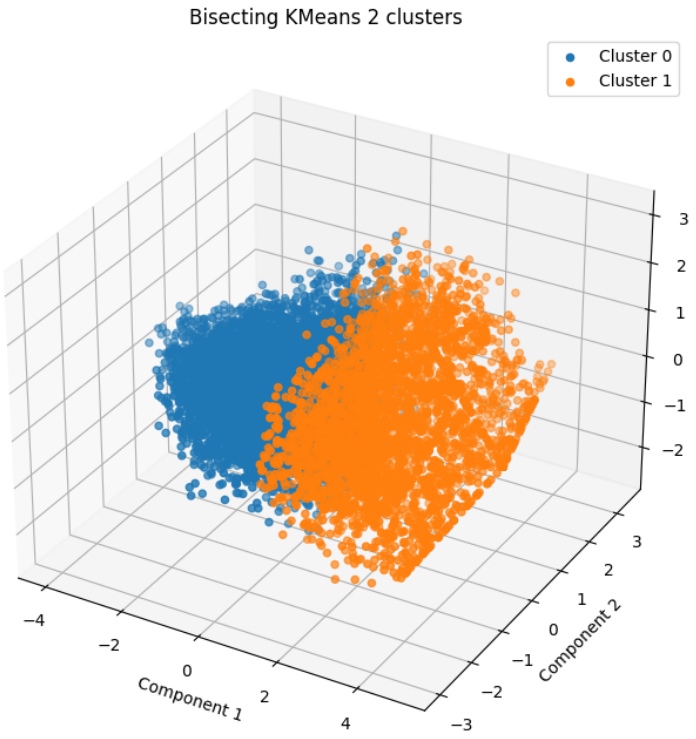


Slika 3.8: SSE u zavisnosti od broja k.

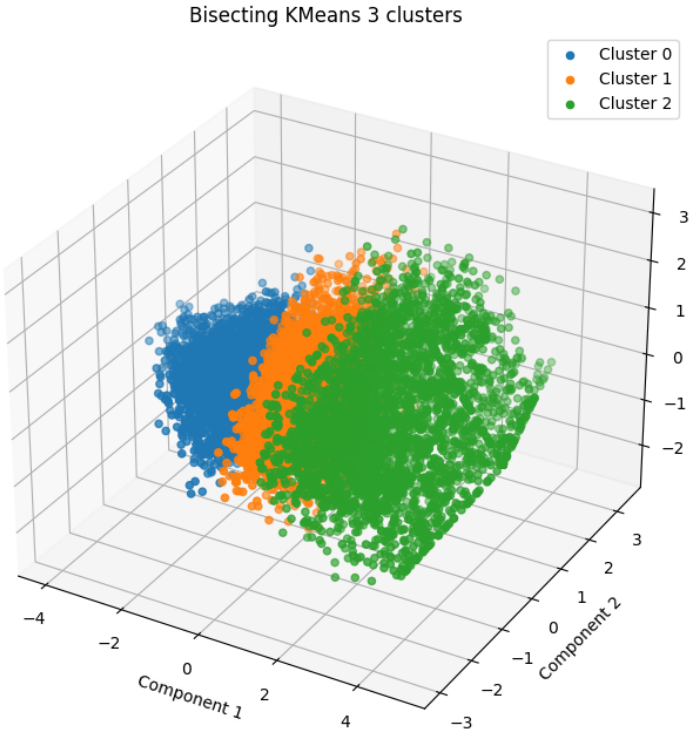


Slika 3.9: Silhouette score u zavisnosti od broja k.

Grafik sse nam blago ukazuje da je za broj klastera možda najbolje izabrati broj 3 ili 6 ali nam grafik silhouette koeficijenata govori da su klasteri najbolje razdvojeni za $k = 2$. U nastavku prikazano je klasterovanje na dva i tri klastera.



Slika 3.10: Bisecting K-means: $k = 2$.



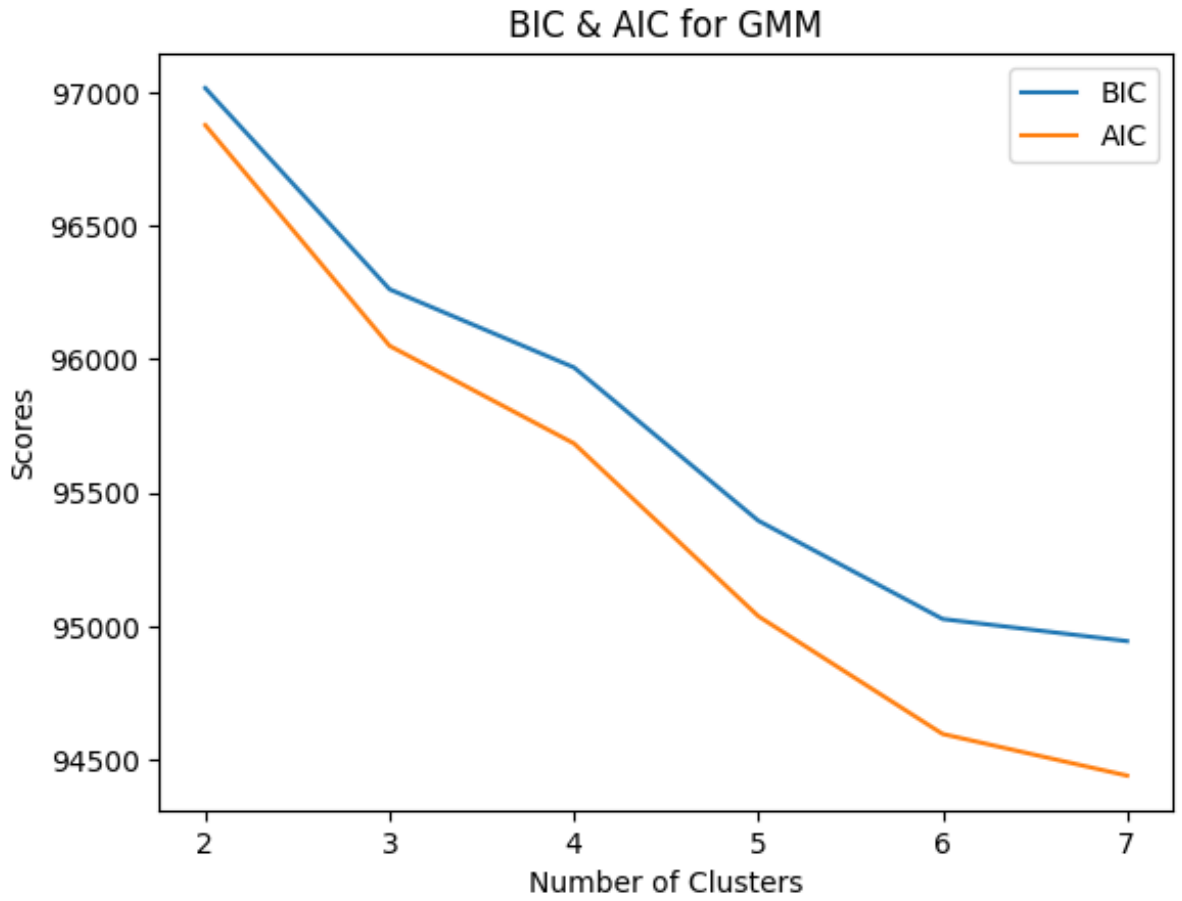
Slika 3.11: Bisecting K-means: $k = 3$.

3.4 Gausov model mešavine

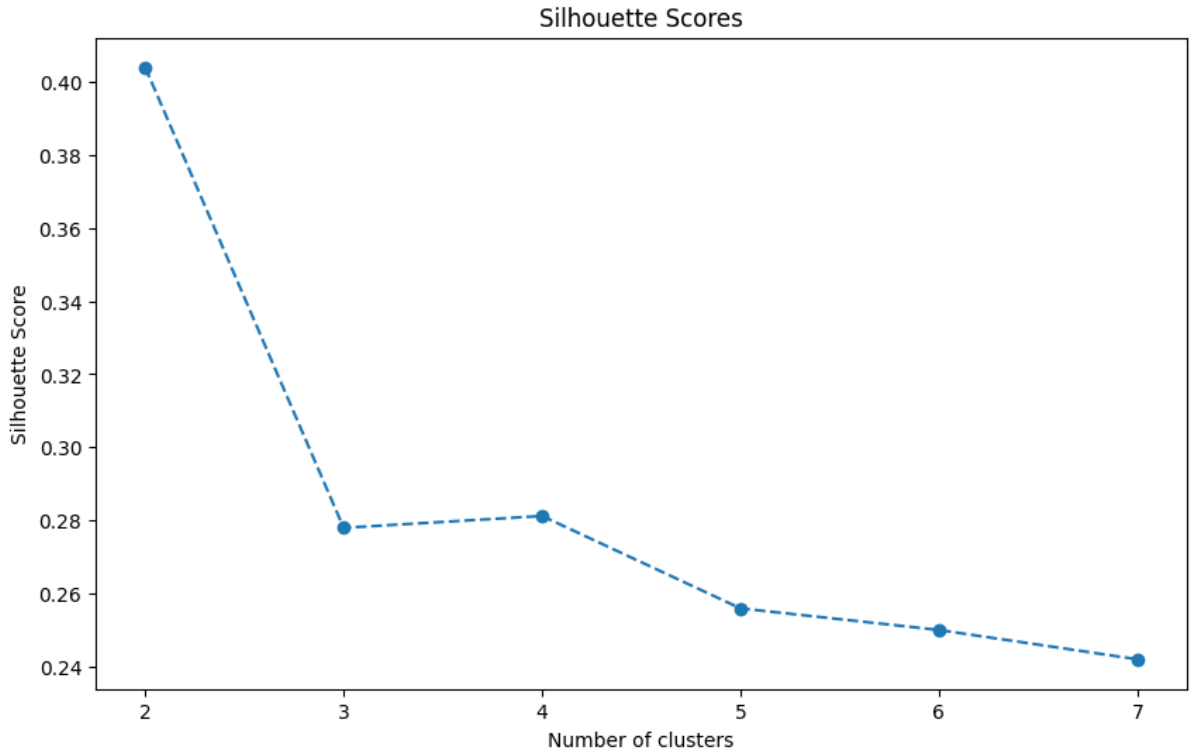
Gausov model mešavine (Gaussian Mixture Model - GMM) je probabilistički model koji pretpostavlja da su podaci generisani iz više Gausovih distribucija. Svaka Gausova distribucija predstavlja klaster. Model pronalazi parametre tih distribucija (srednju vrednost i varijansu) koji najbolje odgovaraju podacima. Za razliku od K-means algoritma koji svaku tačku dodeljuje jednom klasteru, GMM dodeljuje verovatnoću pripadanja svakom klasteru.

3.4.1 Biranje optimalnog broja klastera za Gausov model mešavine

Kao i za prethodne algoritme, GMM je rađen za broj klastera u rasponu od 2 do 7. Pored silhouette skora, koriste se metrike BIC i AIC koje se minimizuju slično kao i sse kod algoritma K-sredina. U nastavku se mogu pogledati silhouette koeficijenti i BIC i AIC za različite vrednosti broja klastera.

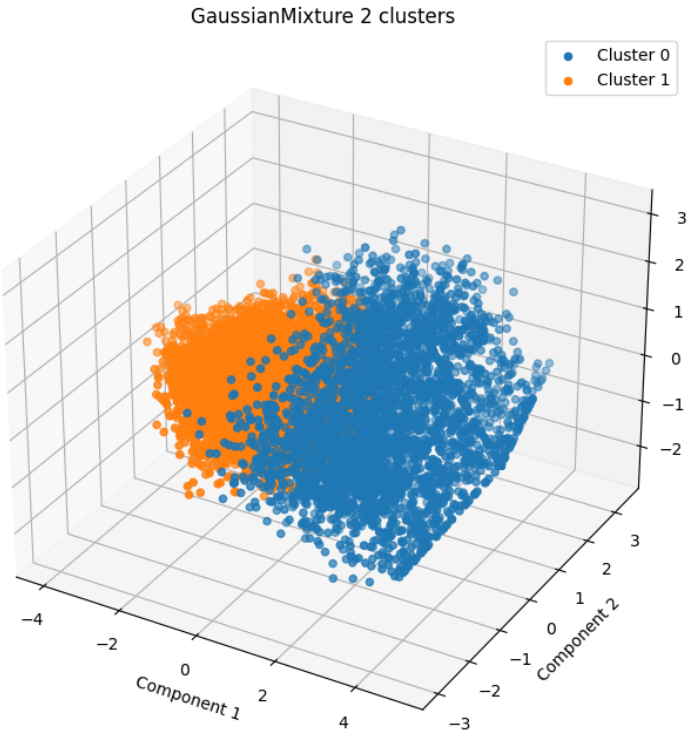


Slika 3.12: BIC i AIC score u zavisnosti od broja k.

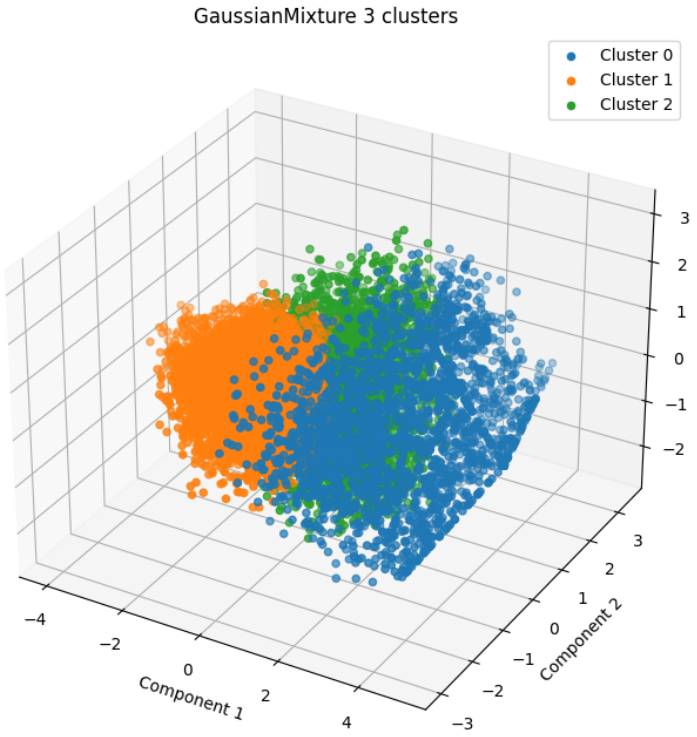


Slika 3.13: Silhouette score u zavisnosti od broja k.

BIC i AIC nam ne govore mnogo o optimalnom broju klastera dok nam silhouette skor preporučuje 2 klastera. Na sledećim slikama može se videti klasterovanje na dva i tri klastera.



Slika 3.14: GMM: $k = 2$.



Slika 3.15: GMM: $k = 3$.

3.5 Poređenje modela za klasterovanje

Za procenu uspešnosti klasterovanja potrebno je široko znanje iz domena, tj. oblasti fizike i vrlo teško je govoriti o detaljima. Ako se pogledaju mere uspeha naših modela nije isključivo da klasterovanje nije baš bilo uspešno. Za računanje bliskosti našeg klasterovanja i prave klasifikacije korišćene su mere ARI i NMI. Bitno je naglasiti da iako klasterovanje nije uspešno za klasifikaciju, ne znači da nije uspešno inače, jer je možda pronašlo neke nove interesantne stvari koje važe za naše podatke. Što se tiče ARI mere, ona poredi grupisanje uzoraka u stvarne klase i grupisanje uzoraka u klastere. Ima vrednost od -1 do 1. NMI meri količinu informacija deljenu između klastera i stvarnih klasa i ima vrednost od 0 do 1. Veće vrednosti su bolje. U nastavku prikazane su ARI i NMI mere za naše modele.

```
Adjusted Rand Index: 0.017671976200457913
Normalized Mutual Information: 0.011367572951140239
```

Slika 3.16: K-means: $k = 2$ - ARI i NMI.

```
Adjusted Rand Index: 0.017623281588269646
Normalized Mutual Information: 0.011330090937800304
```

Slika 3.17: Bisecting K-means: $k = 2$ - ARI i NMI.

```
Adjusted Rand Index: 0.015533695769814688
Normalized Mutual Information: 0.009679067376613715
```

Slika 3.18: GMM: $k = 2$ - ARI i NMI.

Kako su ARI i NMI veoma blizu nule, to označava da naši modeli nisu dobri za klasifikaciju što je i bilo za očekivati zbog kompleksnosti podataka i toga što je "HIGGS dataset" skup podataka za klasifikaciju. Sledeće poglavlje bavi se pravilima pridruživanja i Apriori algoritmom.

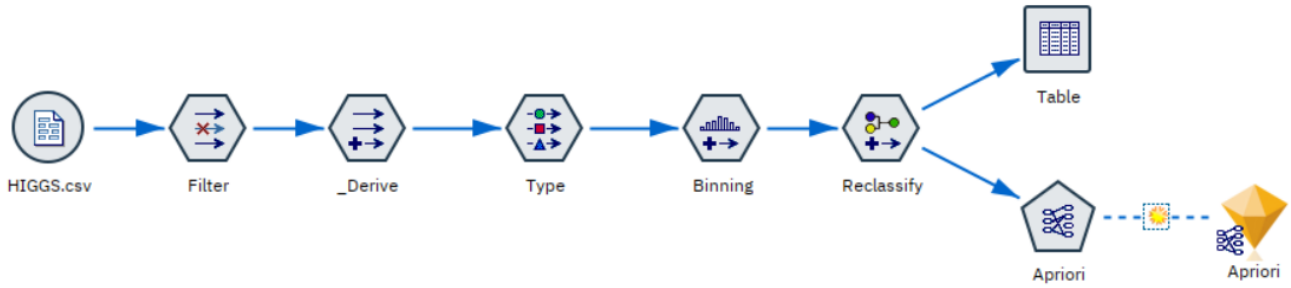
4 Pravila pridruživanja

Pravila pridruživanja predstavljaju jako bitnu komponentu istraživanja podataka, posebno u kontekstu analize potrošačke korpe. Vrše identifikaciju veza između stavki ili skupova stavki u skupu podataka.

- **Skupovi Stavki**: Skupovi jedne ili više stavki. K-skup stavki sadrži k stavki.
- **Support (Podrška)**: Predstavlja odnos pojavljivanja skupa stavki i ukupnog broja zapisa (transakcija).
$$\text{Support}(X) = \frac{\text{Pojavljivanje X}}{\text{Ukupni Zapisi}}$$
- **Confidence (Poverenje)**: Ukazuje na verovatnoću da je stavka Y povezana sa stavkom X. Definiše se kao odnos pojavljivanja X i Y, i pojavljivanja X.
$$\text{Confidence}(X \to Y) = \frac{\text{Support}(X, Y)}{\text{Support}(X)}$$
- **Lift**: Predstavlja verovatnoću pojavljivanja X i Y zajedno u odnosu na individualne verovatnoće.
$$\text{Lift}(X \to Y) = \frac{\text{Confidence}(X \to Y)}{\text{Support}(Y)}$$

4.1 Apriori algoritam

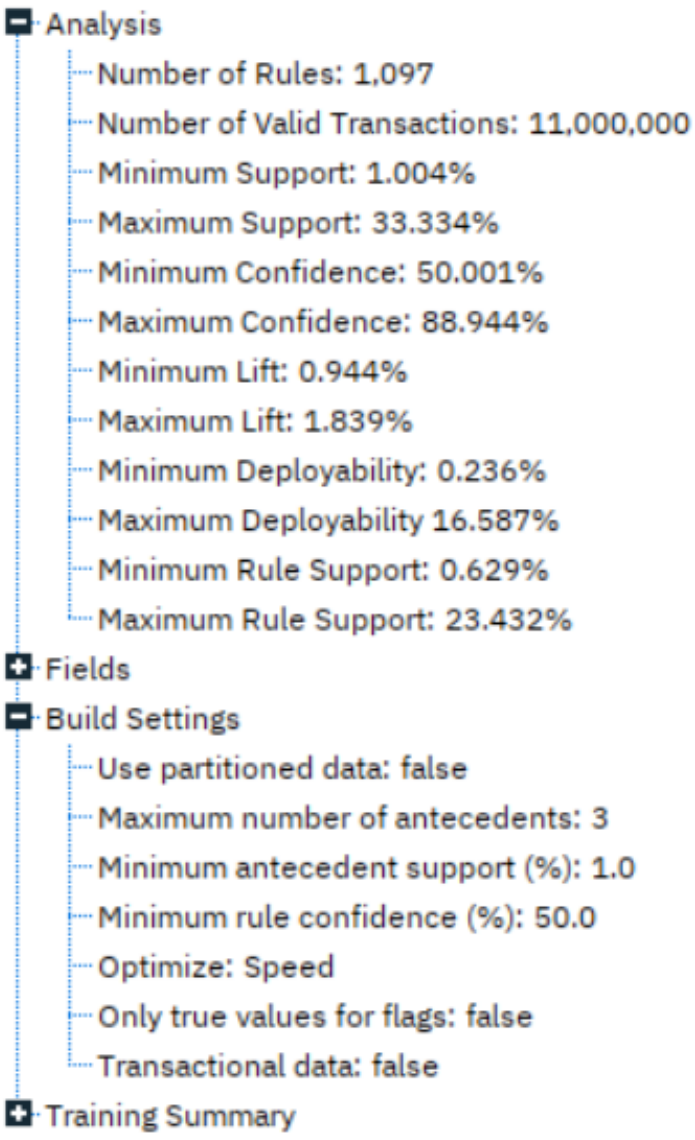
Apriori algoritam je popularan algoritam koji se koristi u analizi potrošačke korpe za generisanje pravila pridruživanja. Oslanja se na sledeće pravilo koje kaže: "Ako je skup stavki čest, onda svi njegovi podskupovi takođe moraju biti česti". Iako apriori algoritam nije prikladan za naš skup podataka (zbog neprekidnih vrednosti), može se iskoristiti uz manje pretprocesiranje. Ideja je da se apriori algoritam radi za target atribut kao posledicu (consequent) i poslednjih 7 atributa (funkcije ostalih atributa) kao prethodnike (antecedents). Vrednosti atributa target pretvorene su iz 0 i 1 u "Not Boson" i "Boson". Takođe, kako bi algoritam mogao da se primeni, potrebno je prebaciti numeričke vrednosti atributa u kategorije low, medium i high. U nastavku prikazani su tok operacija, prvih nekoliko pravila sortiranih po lift-u i kratak pregled informacija o algoritmu.



Slika 4.1: Tok operacija.

Consequent	Antecedent	Support %	Confidence %	Lift
target_Derive = Not b...	m_jlv_LMH = Medium m_wwbb_LMH = High m_jjj_LMH = Medium	2.822	86.435	1.839
target_Derive = Not b...	m_jlv_LMH = Medium m_wwbb_LMH = High m_jjj_LMH = Low	1.963	83.641	1.779
target_Derive = Not b...	m_wwbb_LMH = High m_jjj_LMH = Medium m_jlv_LMH = Low	1.41	82.628	1.758
target_Derive = Not b...	m_jlv_LMH = Medium m_wbb_LMH = High m_jjj_LMH = Medium	2.698	81.694	1.738
target_Derive = Not b...	m_jlv_LMH = Medium m_bb_LMH = High m_wwbb_LMH = High	5.02	80.145	1.705
target_Derive = Not b...	m_bb_LMH = High m_wwbb_LMH = High m_jjj_LMH = Medium	5.103	79.616	1.694
target_Derive = Boson	m_wbb_LMH = Mediu... m_wwbb_LMH = Low m_bb_LMH = Medium	3.683	88.944	1.678
target_Derive = Not b...	m_jlv_LMH = Medium m_wwbb_LMH = High m_jj_LMH = Low	2.298	78.797	1.676
target_Derive = Not b...	m_jlv_LMH = Medium m_bb_LMH = High m_jjj_LMH = Medium	4.583	77.935	1.658

Slika 4.2: Prvih nekoliko pravila sortiranih po liftu.



Slika 4.3: Informacije o algoritmu.

5 Zaključak

"HIGGS dataset" predstavlja jedan kompleksan skup podataka, što odražava složenost povezanu sa fizikom elementarnih čestica. Ovaj projekat prošao je kroz niz tehnika, pokrivajući sve od pretprocesiranja podataka do naprednih analitičkih algoritama. Iako su metode pružile uvid u podatke, proces je takođe istakao potrebu za strpljenjem u istraživanju podataka. To je disciplina koja zahteva upornost i stalno usavršavanje metoda kroz težak niz pokušaja i učenja na greškama.