Traffic violations

Projekat za kurs Istraživanje podataka 1

Autor: Stefan Jevtić mi20187
Asistent: Marija Erić
Profesor: prof. dr Nenad Mitić

Sadržaj

Uvod

Svrha ovog projekta je demonstriranje algoritama:

- Klasifikacije (KNN - K Next Neighbours, Algoritmi stabala odlučivanja)
- Klasterovanja (Algoritam K-sredina, Algoritam DBSCAN,
  Algoritam Sakupljajućeg Klasterovanja)
- Pravila Pridruživanja u SPSS-u (Apriori Algoritam)

Skup podataka sa kojim je rađeno je "Traffic violations", koji se može naći
na sledećem linku:
OpenML

Algoritmi klasifikacije i klasterovanja, kao i pretprocesiranje skupa podataka rađeni su u okruženju Jupyter Notebook, dok je Apriori algoritam (Pravila pridruživanja) rađen u IBM SPSS Modeleru.

Kompletan kod ovog projekta može se naći u repozitorijumu predmeta Istraživnje Podataka 1 za 2023. godinu:
MATF-istrazivanje-podataka-1/2023_Data_Mining_Traffic_violations_Dataset (github.com)

Skup Podataka

Kao što je već pomenuto, ovaj projekat je rađen nad skupom podataka "Traffic violations" koji se sastoji iz datoteke traffic_violations.csv.

Ciljni atribut je Violation.Type koji govori o tipu prekršaja.

U nastavku ovog poglavlja ću podeliti detalje o transformacijama nad skupom podataka, kako bi bilo pogodnije za primenu algoritama klasifikacije, klasterovanja i pravila pridruživanja.

Eksplorativna analiza

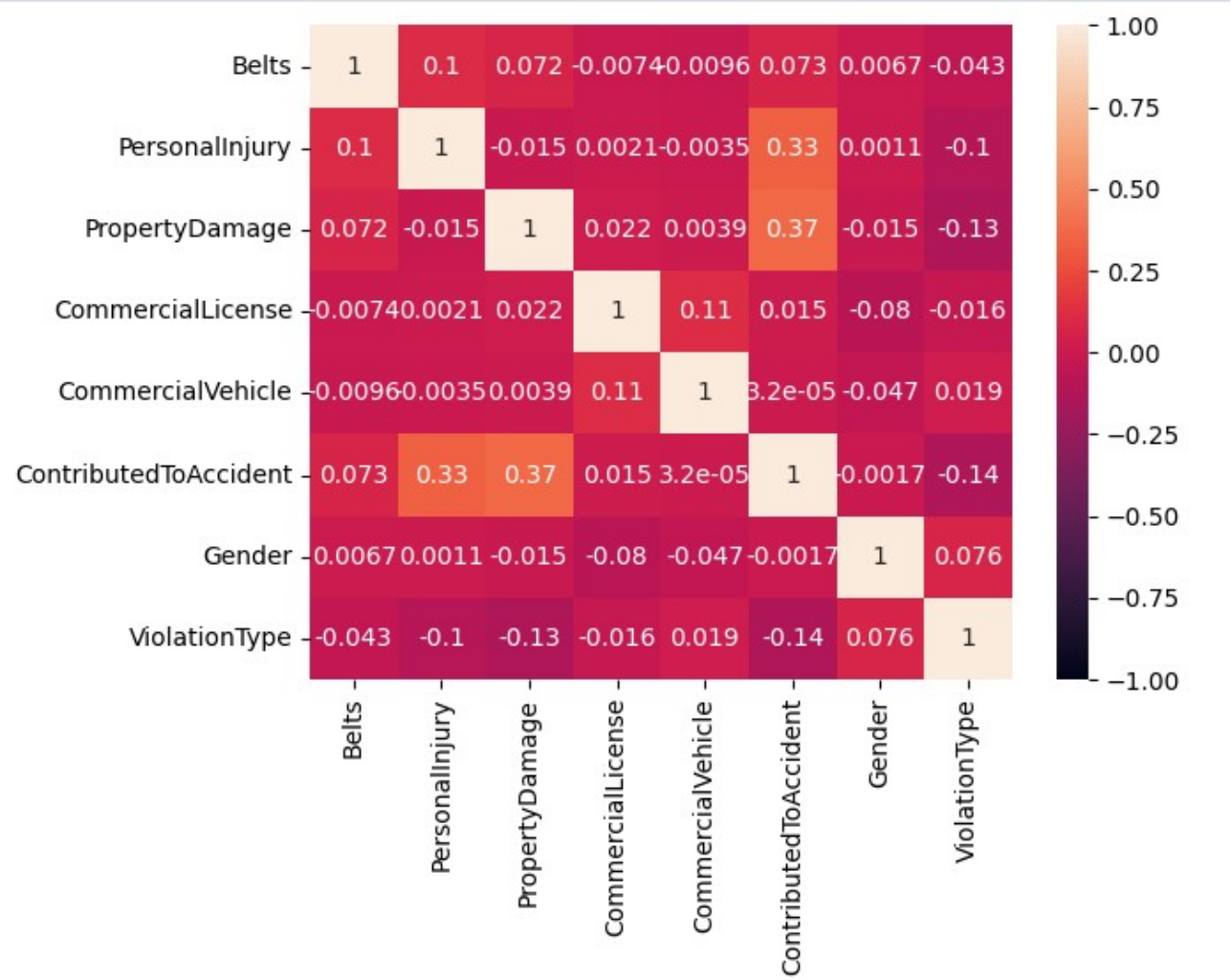
Ovaj skup podataka sadrži 9625 jedinstvenih vrednosti atributa, kao i 20 kolona (atributa) koje nam ga bolje opisuju:

- Description
- Belts
- Personal.Injury
- Property.Damage
- Commercial.License
- Commercial.Vehicle
- State
- Vehicle.Type
- Year
- Make
- Model
- Color
- Charge
- Contributed.To.Accident
- Race
- Gender
- Driver.City
- Driver.State
- DL.State
- Arrest.Type
- Violation.Type

Pogledajmo prethodno pomenute jedinstvene vrednosti unutar skupa podataka:

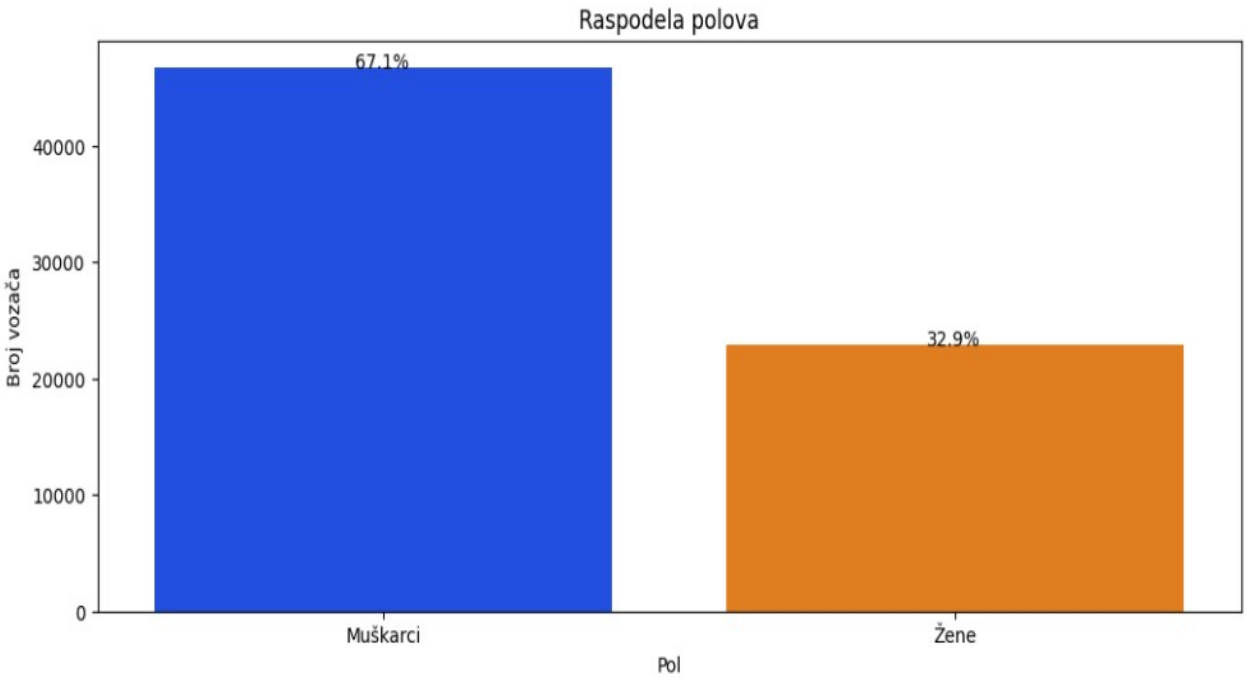
```
1. Description sadrži 2130 jedinstvenih vrednosti: ["'DISPLAYING EXPIRED REGISTRATION PLATE ISSUED BY ANY STATE'"
 "'DRIVER FAIL TO STOP AT RED TRAFFIC SIGNAL BEFORE RIGHT TURN'"
 "'DRIVING UNDER THE INFLUENCE OF ALCOHOL PER SE'" ...
 "'PARTS NOT SPECIFICALLY PROVIDED FORNOT IN SAFE OPERATING CONDITIONHEADER BOARD CRACKED'"
 "'fail to return card'"
 "'FAILURE TO MAINTAIN LEGIBLE REGISTRATION PLATE FREE FROM FOREIGN MATERIALSREAR TAG COVERED IN DIRT'"]
2. Belts sadrži 2 jedinstvenih vrednosti: ['No' 'Yes']
3. Personal.Injury sadrži 2 jedinstvenih vrednosti: ['No' 'Yes']
4. Property.Damage sadrži 2 jedinstvenih vrednosti: ['No' 'Yes']
5. Commercial.License sadrži 2 jedinstvenih vrednosti: ['No' 'Yes']
6. Commercial.Vehicle sadrži 2 jedinstvenih vrednosti: ['No' 'Yes']
7. State sadrži 58 jedinstvenih vrednosti: ['NC' 'MD' 'IL' 'PA' 'VA' 'DC' 'VT' 'LA' 'XX' 'WV' 'TX' 'MI' 'TN' 'CA'
 'DE' 'AL' 'WA' 'NJ' 'GA' 'US' 'MT' 'FL' 'KY' 'MN' 'OK' 'AK' 'IN' 'OH'
 'WI' 'NY' 'MA' 'AZ' 'IA' 'CT' 'SC' 'CO' 'NV' 'MO' 'KS' 'NE' 'NH' 'RI'
 'ME' 'AR' 'NS' '?' 'NM' 'MS' 'ID' 'HI' 'PR' 'UT' 'OR' 'ND' 'MB' 'SD' 'ON'
 'VI']
8. VehicleType sadrži 22 jedinstvenih vrednosti: ["'02 - Automobile'" "'05 - Light Duty Truck'" "'28 - Other'"
 "'01 - Motorcycle'" "'08 - Recreational Vehicle'" "'03 - Station Wagon'"
 "'06 - Heavy Duty Truck'" "'29 - Unknown'" "'19 - Moped'"
 "'07 - Truck/Road Tractor'" "'25 - Utility Trailer'" "'10 - Transit Bus'"
 "'20 - Commercial Rig'" "'12 - School Bus'" "'04 - Limousine'"
 "'27 - Farm Equipment'" "'24 - Camper'" "'14 - Ambulance(Non-Emerg)'"
 "'09 - Farm Vehicle'" "'26 - Boat Trailer'" "'21 - Tandem Trailer'"
 "'11 - Cross Country Bus'"]
9. Year sadrži 97 jedinstvenih vrednosti: ['2013' '2015' '2000' '2012' '2010' '2007' '1998' '2011' '2004' '2016'
 '1994' '2006' '1999' '2002' '2003' '2001' '1996' '2014' '2008' '1993'


 '7705b4' '22206c' '113911C' '2112051b1' '22412a' '2221511' '113961']
14. Contributed.To.Accident sadrži 2 jedinstvenih vrednosti: ['No' 'Yes']
15. Race sadrži 6 jedinstvenih vrednosti: ['WHITE' 'OTHER' 'BLACK' 'HISPANIC' "'NATIVE AMERICAN'" 'ASIAN']
16. Gender sadrži 3 jedinstvenih vrednosti: ['F' 'M' 'U']
17. Driver.City sadrži 1890 jedinstvenih vrednosti: ['ASHEVILLE' "'SILVER SPRING'" 'COLUMBIA' ... 'NEBO' 'ALOHA' 'MEQUON']
18. Driver.State sadrži 57 jedinstvenih vrednosti: ['NC' 'MD' 'PA' 'DC' 'LA' 'VA' 'MI' 'ND' 'TN' 'CA' 'FL' 'DE' 'AL' 'GA'
 'NY' 'IL' 'XX' 'OH' 'NJ' 'WV' 'MA' 'TX' 'HI' 'SC' 'KY' 'CO' 'NV' 'MT'
 'NE' 'CT' 'PR' 'IN' 'NM' 'MO' 'NH' 'IA' 'OR' 'WA' 'AZ' 'UT' 'MS' 'MN'
 'RI' 'ON' 'SD' 'AK' 'WI' 'KS' 'OK' 'AR' 'ID' 'ME' 'BC' 'VT' 'QC' 'MB'
 'AB']
19. DL.State sadrži 64 jedinstvenih vrednosti: ['NC' 'MD' 'PA' 'VA' 'DC' 'XX' 'LA' '?' 'WV' 'MI' 'TN' 'NV' 'FL' 'CO' 'CA'
 'DE' 'AL' 'NJ' 'GA' 'NY' 'IL' 'RI' 'MT' 'IN' 'KY' 'TX' 'SC' 'OH' 'WI'
 'OK' 'MA' 'IA' 'CT' 'VT' 'HI' 'US' 'MO' 'ND' 'NE' 'WA' 'PR' 'ME' 'MB'
 'NM' 'NH' 'VI' 'AZ' 'SD' 'UT' 'MN' 'ON' 'OR' 'MS' 'AR' 'KS' 'ID' 'IT'
 'AK' 'AB' 'BC' 'SK' 'WY' 'QC' 'NS']
20. Arrest.Type sadrži 19 jedinstvenih vrednosti: ["'A - Marked Patrol'" "'B - Unmarked Patrol'"
 "'S - License Plate Recognition'" "'Q - Marked Laser'" "'L - Motorcycle'"
 "'O - Foot Patrol'" "'R - Unmarked Laser'" "'M - Marked (Off-Duty)'"
 "'E - Marked Stationary Radar'" "'G - Marked Moving Radar (Stationary)'"
 "'I - Marked Moving Radar (Moving)'"
 "'H - Unmarked Moving Radar (Stationary)'" "'P - Mounted Patrol'"
 "'N - Unmarked (Off-Duty)'" "'D - Unmarked VASCAR'"
 "'J - Unmarked Moving Radar (Moving)'" "'F - Unmarked Stationary Radar'"
 "'C - Marked VASCAR'" "'K - Aircraft Assist'"]
21. Violation.Type sadrži 3 jedinstvenih vrednosti: ['Citation' 'SERO' 'Warning']
```

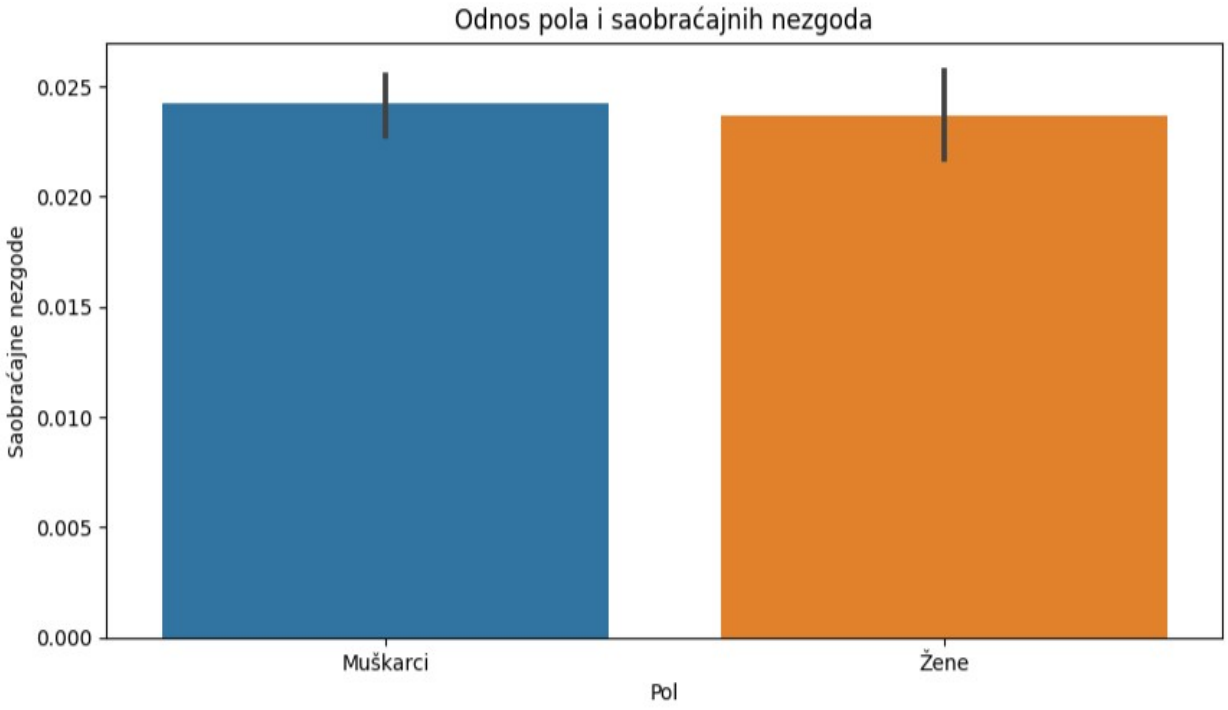
Nakon pretprocesiranja, o kojem će biti reči kasnije, pogledajmo korelacije između ulaznih atributa:

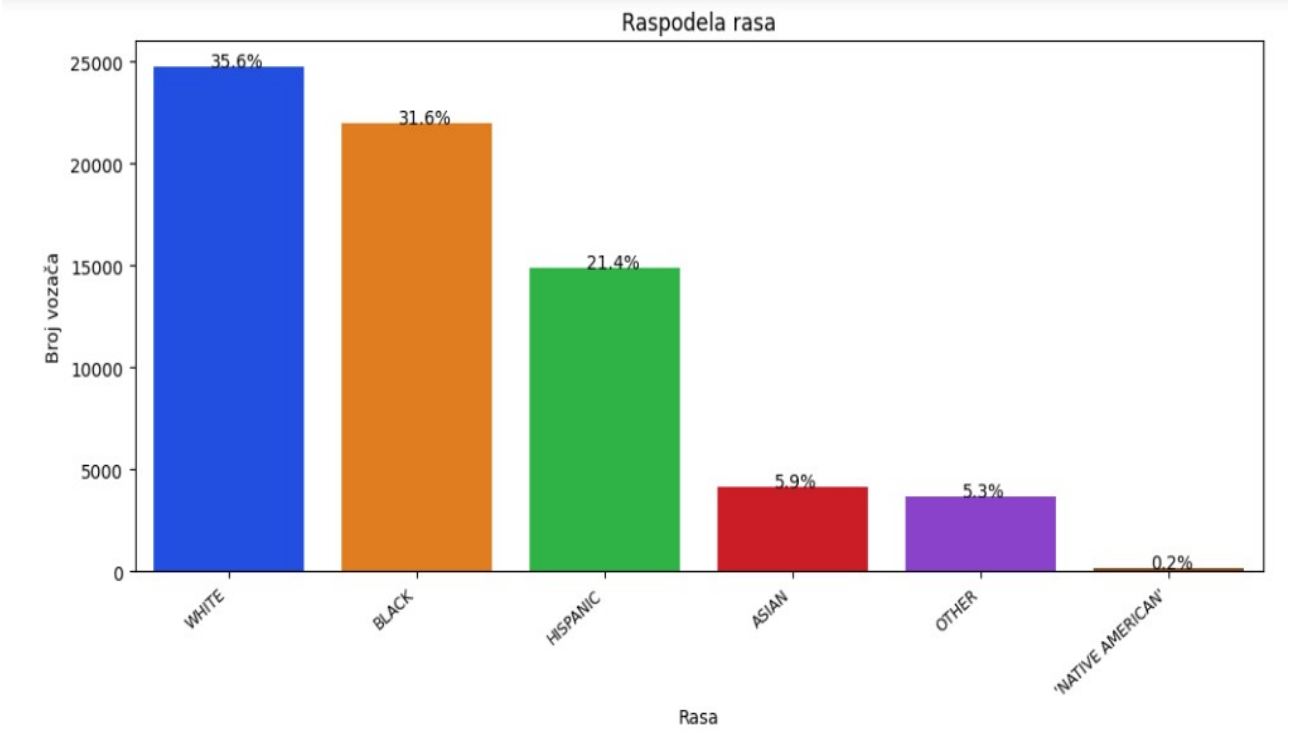


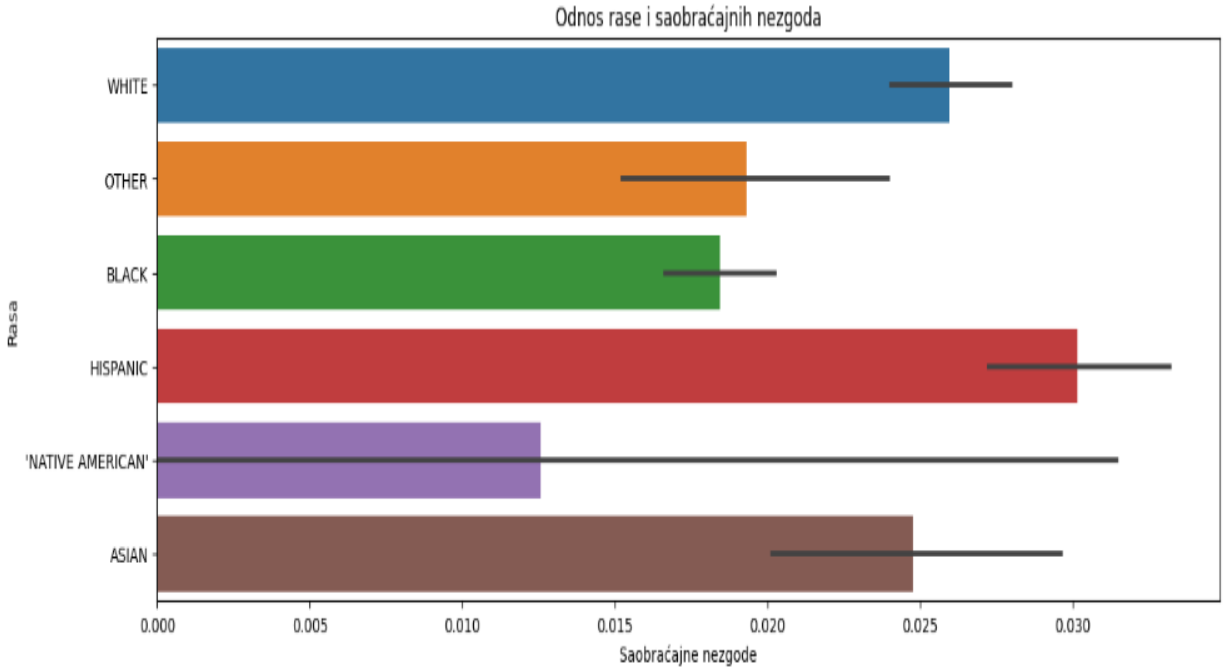
Primećujemo veoma blagu korelaciju između atributa ContributedToAccident, PersonalInjury i PropertyDamage.
Ove činjenice su veoma logične i nisu iznenađujuće.

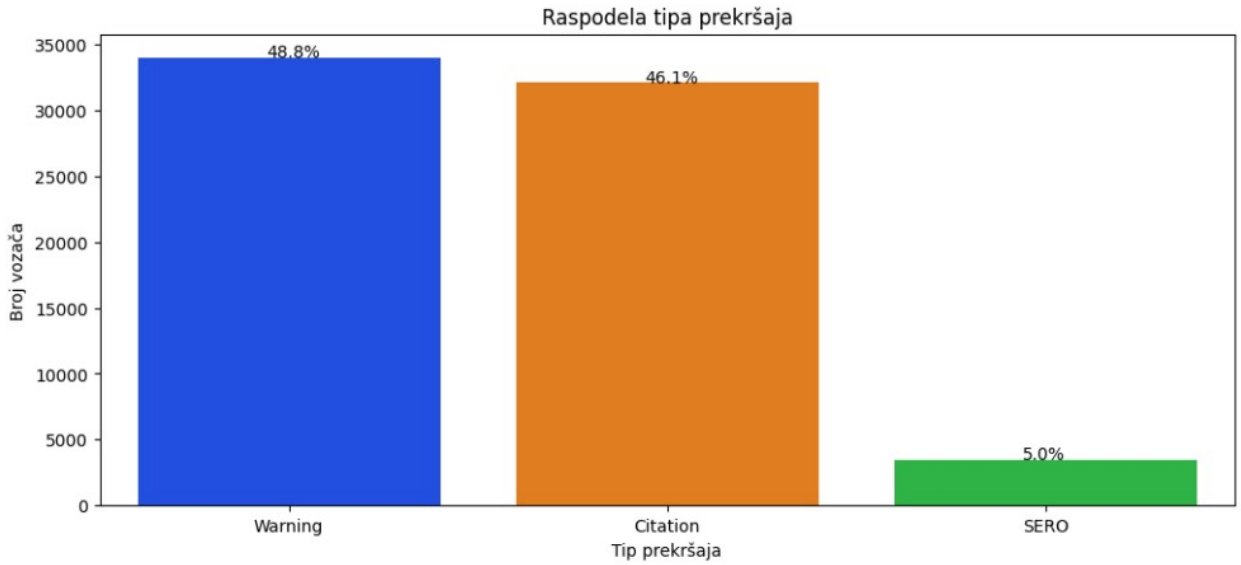
Pogledajmo neke ostale zavisnosti između atributa:

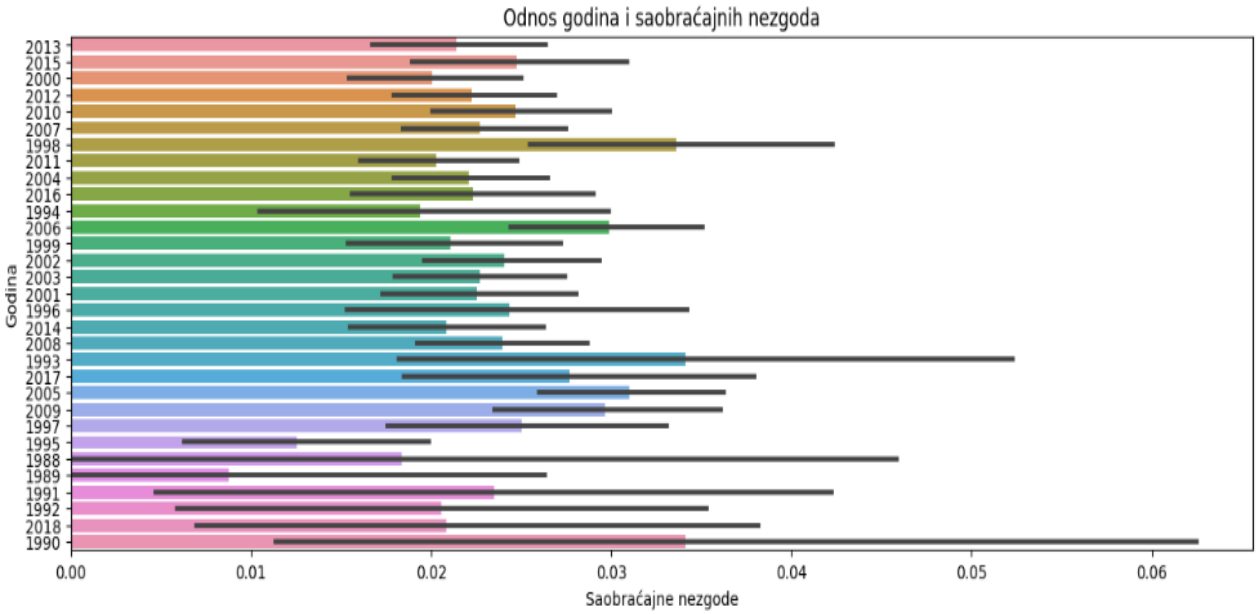


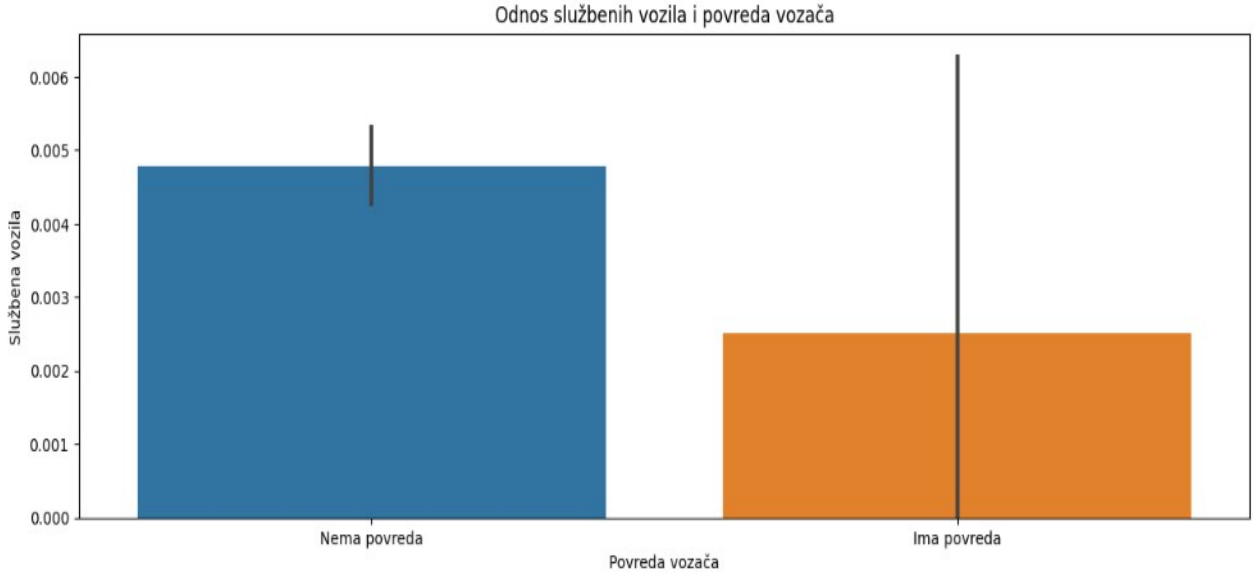



Raspodela rasa
Broj vozača
25000
20000
15000
10000
5000
0
35.6%
31.6%
21.4%
5.9%
5.3%
0.2%
WHITE
BLACK
HISPANIC
ASIAN
OTHER
'NATIVE AMERICAN'
Rasa


Odnos rase i saobraćajnih nezgoda
Rasa
WHITE
OTHER
BLACK
HISPANIC
'NATIVE AMERICAN'
ASIAN
0.000
0.005
0.010
0.015
0.020
0.025
0.030
Saobraćajne nezgode


Raspodela tipa prekršaja
48.8%
46.1%
5.0%
Broj vozača
Warning
Citation
SERO
Tip prekršaja


Odnos godina i saobraćajnih nezgoda
Godina
Saobraćajne nezgode


Odnos službenih vozila i povreda vozača
Službena vozila
Nema povreda
Ima povreda
Povreda vozača

Pretprocesiranje

Pretprocesiranje podataka podrazumeva:

- Čišćenje podataka
- Prenošenje jednog tipa podatka u drugi
- Redukciju i transformaciju podataka

Ovi koraci nisu univerzalni, i u nastavku ćemo raditi pojedinačna, specifična pretprocesiranja za svaki algoritam koji budemo implementirali.

Međutim, postoje neki koraci koje bismo ponavljali u svakom pojedinačnom pretprocesiranju, tako da su oni odrađeni unapred, odmah nakon eksplorativne analize.

Čišćenje podataka

Na prvi pogled nije bilo nedostajućih vrednosti.
Međutim, postoje instance čije su vrednosti nepoznate u kolonama State, Year, Make, Model, Color, Driver.City i DL.State.

Njih sam pretvorio u numpy.nan i potom obrisao iz razloga što zauzimaju manje od 1 % svih instanci iz navedenih atributa.

Prenošenje jednog tipa podatka u drugi

Atribute sa dva ili tri jedinstvena elementa sam kodirao kao 0, 1 i 2.

Redukcija i transformacija podataka

Kolone Charge, Color, Driver.City, Driver.State, DL.State, State i Description nam ne igraju bitnu ulogu u predviđanju rezultujuće klase Violation.Type.
Na samom kraju pretprocesiranja sam kodirao nominalne vrednosti atributa u binarne (0 i 1) koji označavaju pripadnost instance datom atributu, pri čemu sam jedinstvene vrednosti svih kolona izdvojio kao posebne kolone.

Klasifikacija

Klasifikacija je problem određivanja ciljne funkcije *f* koja preslikava skup atributa *X* u neku od unapred određenih oznaka klasa *y*.

Ciljna funkcija se obicno naziva klasifikacionim modelom.

Prilikom klasifikacije, podaci se dele na trening skup, skup za proveru i test skup. Nekada se skup za proveru (validacioni skup) izostavlja.

Trening skup služi za obucavanje modela koji ćemo koristiti za klasifikaciju.
Test skup služi da proverimo koliko je naš model dobar na podacima koje nikad pre nije video.

Skup za proveru služi za odabir različitih parametara modela ili modela uopšte.
U ovom projektu, skup za proveru nije korišćen.

KNN Algoritam

KNN - K-Nearest Neighbors (K Najbližih Suseda) Algoritam je jedan od najpoznatijih algoritama klasifikacije.

Za ovaj algoritam neophodni su nam:

- Skup sačuvanih slogova (trening skup)
- Metrika kojom merimo rastojanje izmedu instanci (obično se koristi Euklidsko rastojanje)
- Broj suseda – k

Opis algoritma:

1. Izračunati rastojanje test instance do svih instanci iz trening skupa
2. Odrediti k najbližih suseda
3. Glasanjem utvrditi klasu test instance

Pretprocesiranje

Kao što je već napomenuto, iako smo izvršili “uopšteno” pretprocesiranje, neophodno je i uraditi “specifično” pretprocesiranje za svaki algoritam.

Za ovaj algoritam, želimo da uključimo više atributa. Međutim, potrebno nam je da ti atrubuti budu predstavljeni brojevima. Zbog toga ćemo enkodirati kategoričke vrednosti u 0 i 1 u zavisnosti od toga da li određena instanca pripada datom atributi ili ne.

Atributi koje smo izabrali za treniranje modela su:

- 'Belts'
- 'PersonalInjury'
- 'PropertyDamage'
- 'CommericalLicense'
- 'CommercialVehicle'
- 'VehicleType'
- 'Year'
- 'Make'
- 'Model'
- 'ContributedToAccident'
- 'Race'
- 'Gender'
- 'ArrestType'

Ove atribute smo skalirali korišćenjem MinMax metode.

Rezultati algoritma

Nakon pretprocesiranja, KNN model smo istrenirali na trening skupu, i zatim razmotrili koliko precizne rezultate dobijamo.
Koristio sam tačnost i matricu konfuzije kao metrike.
Tačnost nekog modela računa se kao procenat korektno klasifikovanih instanci.

```
Classification report for model KNeighborsClassifier on test data
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.50      0.44      0.47      9637
           1       0.06      0.32      0.10      1049
           2       0.53      0.36      0.43     10205

    accuracy                           0.40     20891
   macro avg       0.37      0.38      0.33     20891
weighted avg       0.49      0.40      0.43     20891

--------------------------------------------------------------------------------
Confusion matrix for model KNeighborsClassifier on test data
--------------------------------------------------------------------------------
   Kazna  Oprema  Upozorenje
0   4225    2477        2935
1    383     340         326
2   3804    2695        3706
--------------------------------------------------------------------------------
```

Optimizacija KNN Algoritma

GridSearchCV

GridSearchCV je tehnika za podešavanje hiperparametara u mašinskom učenju, koja uključuje traženje najboljeg skupa hiperparametara koji rezultuju optimalnom performansom modela.

U GridSearchCV-u, skup hiperparametara i njihove odgovarajuće vrednosti su unapred definisani, a algoritam procenjuje performansu modela za svaku kombinaciju ovih hiperparametara.

Procena se obično vrši pomoću unakrsne validacije, gde se skup podataka deli na nekoliko preklopljenih delova, a model se trenira na jednom delu podataka i testira na preostalom delu.

Ovo pomaže u izbegavanju preprilagođavanja i pruža pouzdaniju procenu performanse modela.
GridSearchCV izvršava iscrpnu pretragu svih mogućih kombinacija hiperparametara i vraća kombinaciju koja daje najbolju performansu na validacionim podacima.

Metrika performanse koja se koristi za evaluaciju može biti specificirana od strane korisnika i može se razlikovati u zavisnosti od konkretnog problema.
Kreirao sam dva modela, jedan koristi parametar *uniform*, a drugi *distance*.

Pre pozivanja GridSearchCV-a, definisali smo jednu mapu:

```
params = {
    'n_neighbors': [3, 4, 5, 6, 7, 8, 9, 10],
    'weights' : ['uniform'],
    'p': [1, 2]
}
```

Ova mapa nam daje opcije za sledeće hiperparametre za KNN klasifikator:

- 'n_neighbors' sa opsegom od 3 do 10
  (Ovo predstavlja koji broj K ćemo pokušavati da uzmemo)
- 'weights' sa opcijama:
  - "uniform" znači da će svaki sused imati podjednak uticaj na klasifikaciju novog uzorka.
  - "distance" znači da će bliži susedi imati veću težinu, dok će udaljeni susedi imati manju težinu. Ova opcija uzima u obzir udaljenost suseda prilikom klasifikacije i može biti korisna kada su bliži susedi relevantniji za klasifikaciju.
- 'p' sa opcijama:
  - 1 - za rastojanje između suseda koristiće se Menhetn rastojanje
  - 2 - za rastojanje između suseda koristiće se Euklidsko rastojanje

Prvi model – weights = uniform

Nakon pokretanja GridSearchCV-a, dobijamo sledeće “idealne” hiperparametre:
{'n_neighbors': 3, 'p': 1, 'weights': 'uniform'}

kao i procenu za tačnost: 0.5769144617107658

Rezultati KNN algoritma uz GridSearchCV optimizaciju hiperparametara:

Classification report for model KNeighborsClassifier on test data

	precision	recall	f1-score	support
0	0.50	0.50	0.50	9637
1	0.07	0.16	0.10	1049
2	0.53	0.45	0.49	10205
accuracy			0.46	20891
macro avg	0.37	0.37	0.36	20891
weighted avg	0.49	0.46	0.47	20891

Confusion matrix for model KNeighborsClassifier on test data

	Kazna	Oprema	Upozorenje
0	4861	1085	3691
1	457	172	420
2	4451	1170	4584

Drugi model – weights = distance

Nakon pokretanja GridSearchCV-a, dobijamo sledeće "idealne" hiperparametre:
{'n_neighbors': 5, 'p': 1, 'weights': 'distance'}

kao i procenu za tačnost: 0.588632227355453

Rezultati KNN algoritma uz GridSearchCV optimizaciju hiperparametara:

```
Classification report for model KNeighborsClassifier on test data
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.50      0.49      0.49      9637
           1       0.07      0.21      0.10      1049
           2       0.53      0.43      0.47     10205

    accuracy                           0.44     20891
   macro avg       0.37      0.37      0.36     20891
weighted avg       0.49      0.44      0.46     20891

--------------------------------------------------------------------------------
Confusion matrix for model KNeighborsClassifier on test data
--------------------------------------------------------------------------------
   Kazna  Oprema  Upozorenje
0   4687    1509        3441
1    448     216         385
2   4246    1598        4361
```

Primećujemo da nema bitne razlike između ova dva modela.

Bagging Classifier

Bagging klasifikator, takođe poznat kao bootstrap agregacija, je vrsta algoritma za učenje ansambla koji kombinuje više modela drveta odlučivanja radi poboljšanja ukupne performanse zadatka klasifikacije.

Tokom faze predviđanja, svaki model drveta odlučivanja daje sopstveni rezultat klasifikacije, a konačna klasifikacija se određuje agregiranjem rezultata svih pojedinačnih modela.

Ovo agregiranje se može obaviti tako što se uzima većinsko glasanje, gde se klasifikacija sa najviše glasova bira kao konačno predviđanje.

Kao parametre za ovaj klasifikator smo takođe koristili optimizovane parametre dobijene GridSearchCV-em.
Rezultati:

```
--------------------------------------------------------------------------------
Classification report for model BaggingClassifier on test data
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.52      0.26      0.34      9637
           1       0.06      0.61      0.10      1049
           2       0.54      0.25      0.34     10205

    accuracy                           0.27     20891
   macro avg       0.37      0.37      0.26     20891
weighted avg       0.51      0.27      0.33     20891

--------------------------------------------------------------------------------
Confusion matrix for model BaggingClassifier on test data
--------------------------------------------------------------------------------
   Kazna  Oprema  Upozorenje
0   2463    5186        1988
1    193     641         215
2   2065    5565        2575
--------------------------------------------------------------------------------
```

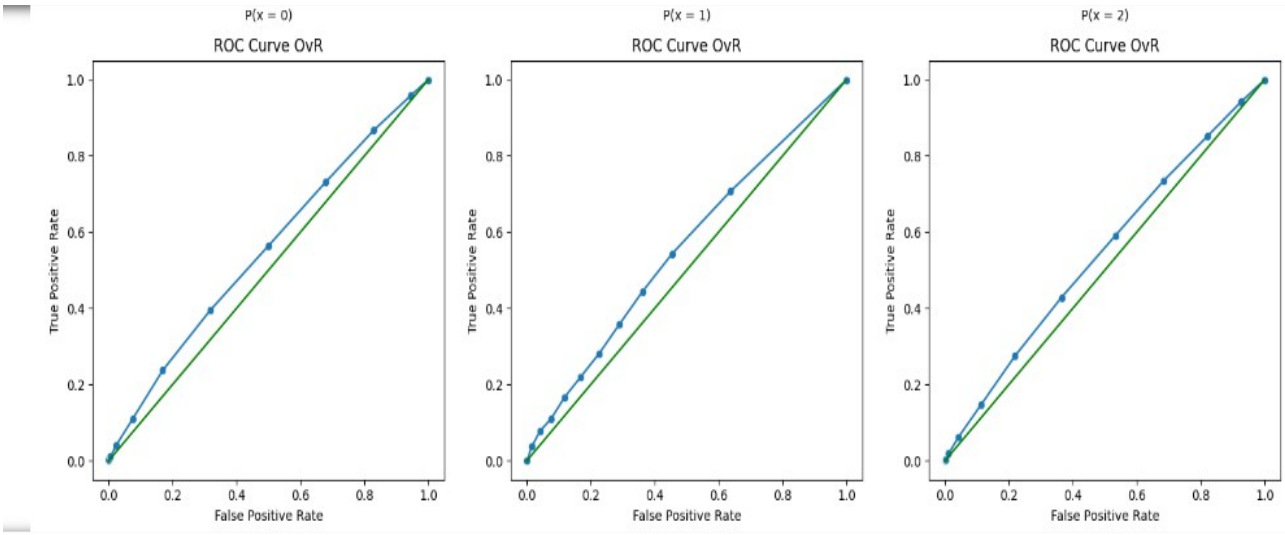
Primećujemo da je tačnost drastično opala u odnosu na KNN GridSearchCV. Međutim, pogledajmo poređenje na osnovu grafika ispod.

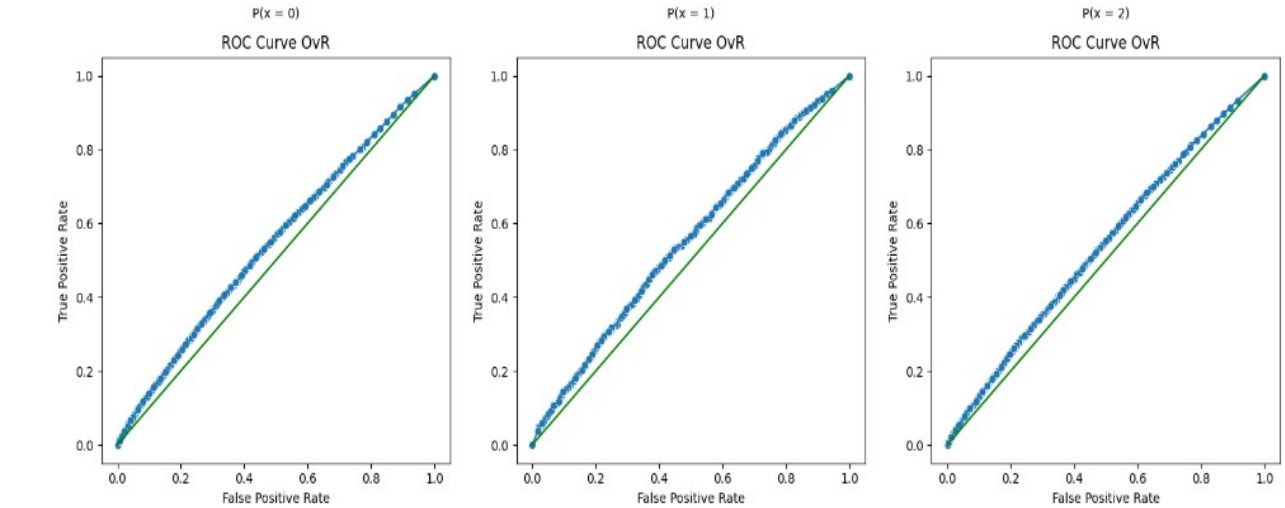
Poređenje modela

Koristićemo ROC krivu.
Receiver Operating Characteristic - ROC kriva je grafički prikaz odnosa između procenta lažno pozitivnih (False Positive Rate - FPR) i procenta tačno pozitivnih (True Positive Rate -TPR) instanci.
Ona prikazuje TPR u odnosu na FPR za različite pragove klasifikacije.

Moramo izmeniti standardnu metodu koja se koristi za binarnu klasifikaciju.

Koristićemo OvR (One vs Rest - Jedan naspram Ostalih) metodu za evaluaciju modela. Tj. merićemo razdvojenost svake klase zasebno od svih ostalih klasa





Stabla odlučivanja

Stabla odlučivanja su popularna metoda za rešavanje problema klasifikacije. U pitanju je vrsta algoritma nadgledanog učenja koji uči da donosi odluke konstruišući model odluka u obliku drveta i njihovih mogućih posledica.

U stablu odlučivanja, svaki čvor predstavlja odluku, a svaka grana predstavlja mogući ishod te odluke. Drvo se konstruiše rekurzivno particionišući podatke u podskupove na osnovu vrednosti ulaznih atributa, sa ciljem minimiziranja neke mere nečistoće ili entropije.

Pri svakom čvoru, algoritam bira karakteristiku koja najbolje razdvaja podatke u različite klase i deli podatke na odgovarajući način.

Pretprocesiranje skupa podataka za ovaj model izvršeno je identično kao pretprocesiranje za KNN model.

Napravio sam dva modela, prvi ne koristi nikakve parametre, dok drugi ima fiksiranu dubinu i entropiju kao meru.

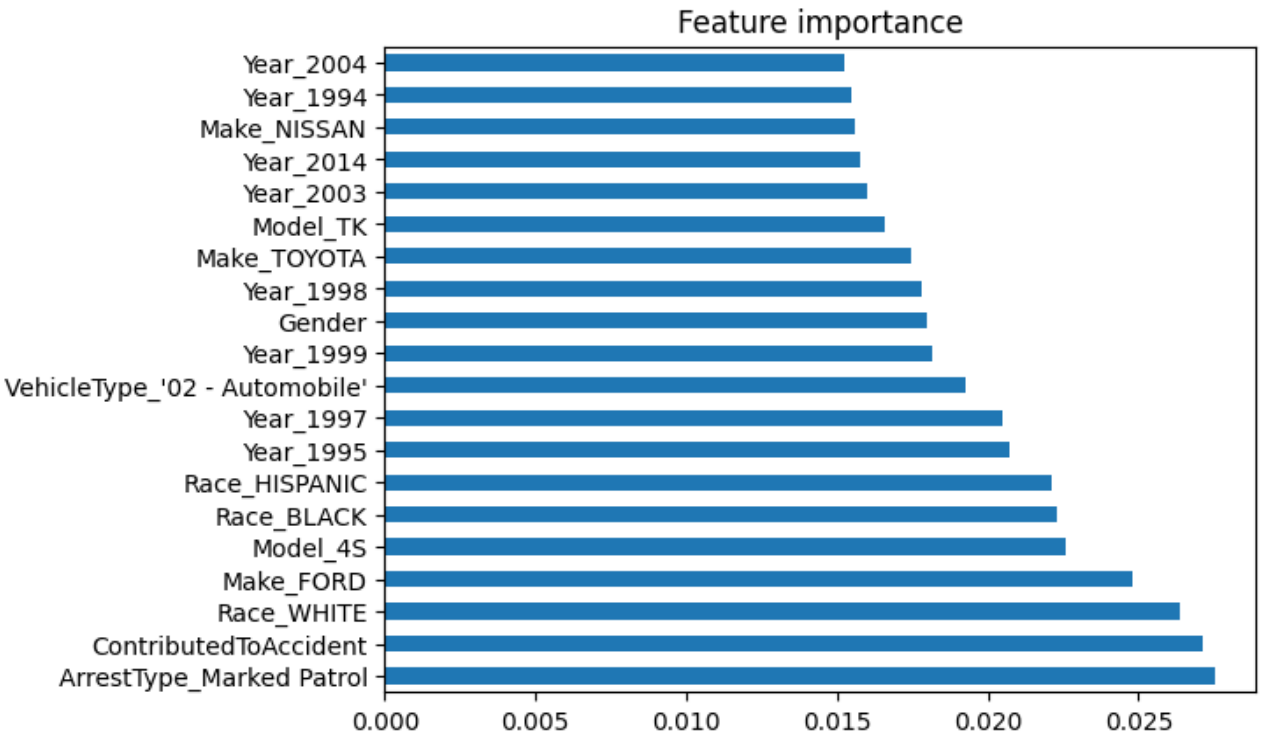
Rezultati klasifikacije ovog algoritma za model bez parametara:

```
Classification report for model DecisionTreeClassifier on test data
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.51      0.52      0.51      9637
           1       0.07      0.19      0.10      1049
           2       0.54      0.44      0.48     10205

    accuracy                           0.46     20891
   macro avg       0.37      0.38      0.37     20891
weighted avg       0.50      0.46      0.48     20891

--------------------------------------------------------------------------------
Confusion matrix for model DecisionTreeClassifier on test data
--------------------------------------------------------------------------------
   Kazna  Oprema  Upozorenje
0   4965    1181        3491
1    435     195         419
2   4307    1395        4503
```

Rezultati modela sa parametrima:

```
Classification report for model DecisionTreeClassifier on test data
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.52      0.45      0.48      9637
           1       0.07      0.32      0.12      1049
           2       0.55      0.42      0.47     10205

    accuracy                           0.43     20891
   macro avg       0.38      0.40      0.36     20891
weighted avg       0.51      0.43      0.46     20891

--------------------------------------------------------------------------------
Confusion matrix for model DecisionTreeClassifier on test data
--------------------------------------------------------------------------------
   Kazna  Oprema  Upozorenje
0   4310    2148        3179
1    350     337         362
2   3616    2310        4279
--------------------------------------------------------------------------------
```

Pogledajmo koji su atributi bili od najvećeg značaja prilikom formiranja stabla odlučivanja:



Optimizacija Stabla Odlučivanja uz GridSearchCV

Baš kao i za KNN, koristićemo GridSearchCV da podešavamo hiperparametre koji dovode do optimalnog rezultata ovog modela.

Pre pozivanja GridSearchCV-a, definisali smo mapu, kao i u KNN-u:

```
params = {'criterion': ['gini', 'entropy'],
          'max_depth': [10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70]
         }
```

Ova mapa nam daje opcije za sledeće hiperparametre za KNN klasifikator:

- 'criterion' sa opcijama:
  - gini - mere nečistoće se izračunavaju računanjem verovatnoće netačne klasifikacije nasumično izabranog podatka u tom čvoru.
  - entropy - mere nečistoće se računaju na osnovu verovatnoća pojavljivanja različitih klasa u čvoru.
- 'max_depth' sa nizom brojeva od 10 do 70 - određuje dubinu drveta

Rezultati algoritma stabla odlučivanja uz GridSearchCV optimizaciju hiperparametara:

```
Classification report for model DecisionTreeClassifier on test data
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.51      0.51      0.51      9637
           1       0.06      0.18      0.09      1049
           2       0.54      0.44      0.49     10205

    accuracy                           0.46     20891
   macro avg       0.37      0.38      0.36     20891
weighted avg       0.50      0.46      0.48     20891

--------------------------------------------------------------------------------
Confusion matrix for model DecisionTreeClassifier on test data
--------------------------------------------------------------------------------
   Kazna  Oprema  Upozorenje
0   4901    1272        3464
1    461     187         401
2   4220    1451        4534
--------------------------------------------------------------------------------
```

Random Forest Classifier

Random Forest Classifier je algoritam ansambla. On gradi više drveća odlučivanja na nasumičnim podskupovima skupa podataka, a zatim kombinuje njihova predviđanja kako bi napravio konačno predviđanje.

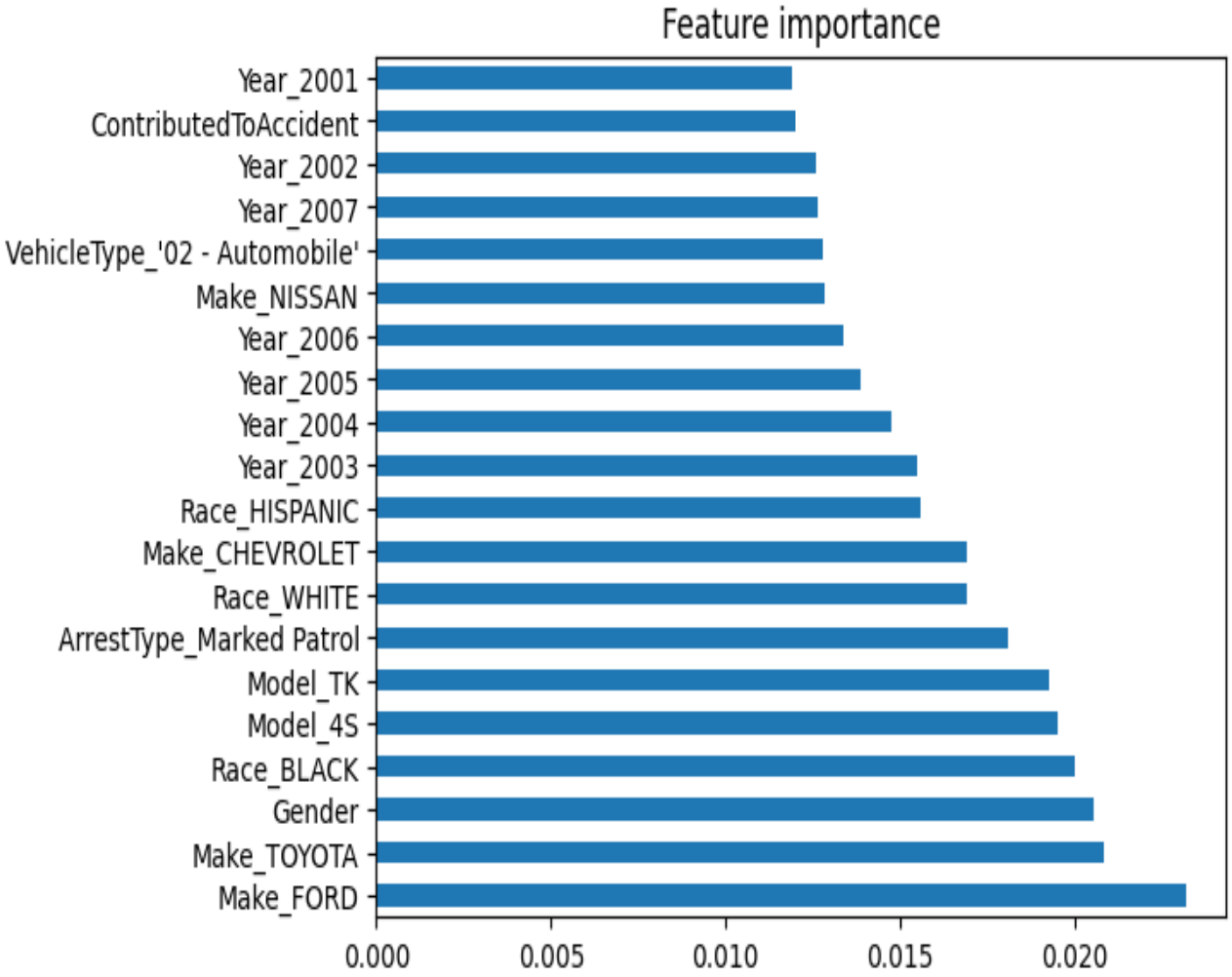
Algoritam nasumično bira podskup karakteristika za svako drvo odlučivanja kako bi smanjio preprilagođavanje trening podacima. Takođe koristi bootstrap uzorkovanje za stvaranje više verzija skupa podataka, što mu omogućava da uhvati više informacija iz skupa podataka i smanji varijansu modela.

Rezultati algoritma:

```
--------------------------------------------------------------------------------
Classification report for model RandomForestClassifier on test data
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.54      0.51      0.52      9637
           1       0.08      0.20      0.11      1049
           2       0.55      0.49      0.52     10205

    accuracy                           0.48     20891
   macro avg       0.39      0.40      0.38     20891
weighted avg       0.52      0.48      0.50     20891

--------------------------------------------------------------------------------
Confusion matrix for model RandomForestClassifier on test data
--------------------------------------------------------------------------------
   Kazna  Oprema  Upozorenje
0   4895    1180        3562
1    388     208         453
2   3858    1370        4977
--------------------------------------------------------------------------------
```

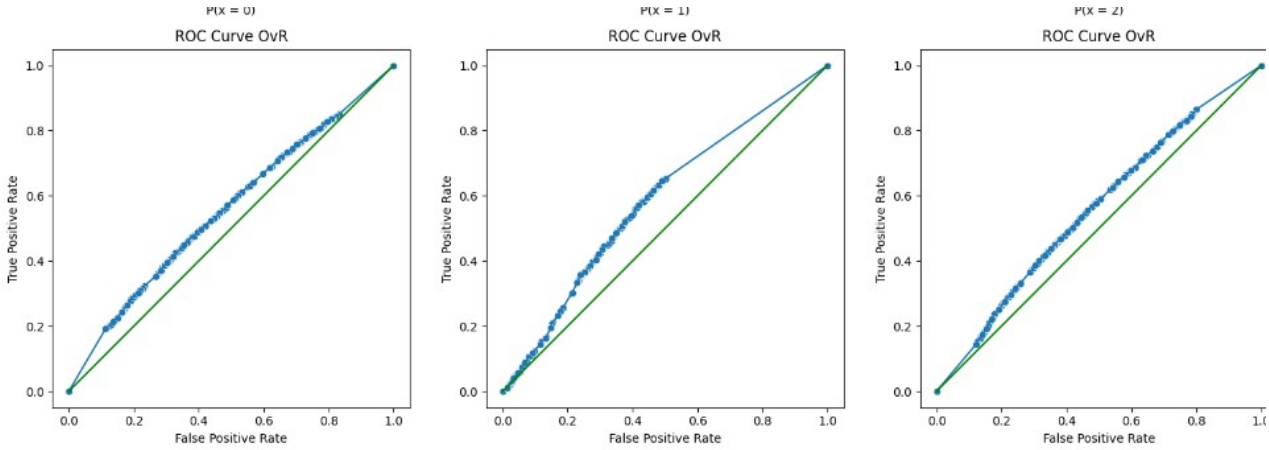
Iako nije velika razlika, možemo primetiti da najbolje performanse daje algoritam Random Forest Classifier, tj slučajne šume.

Pogledajmo koji su atributi bili od najvećeg značaja prilikom formiranja modela ansambla:

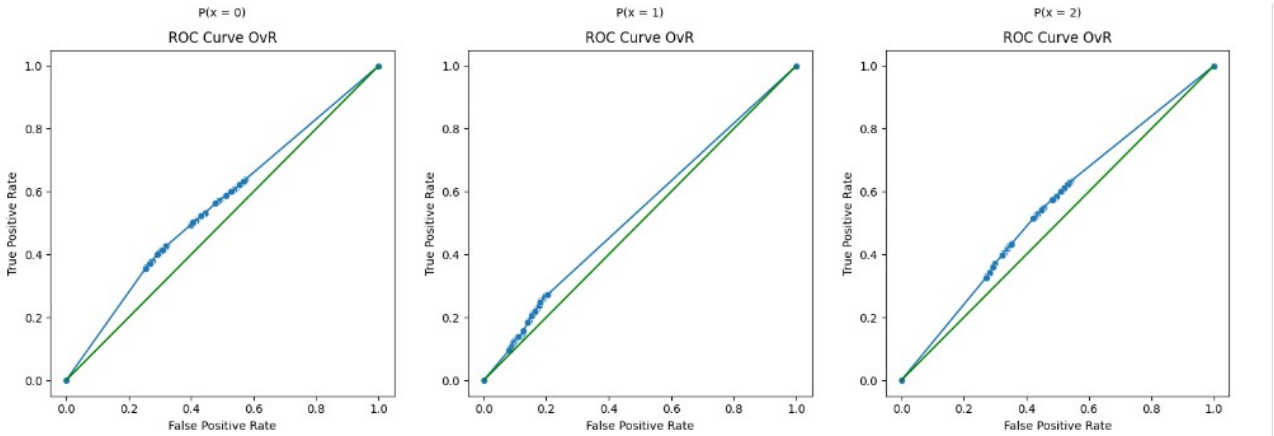


Poređenje modela

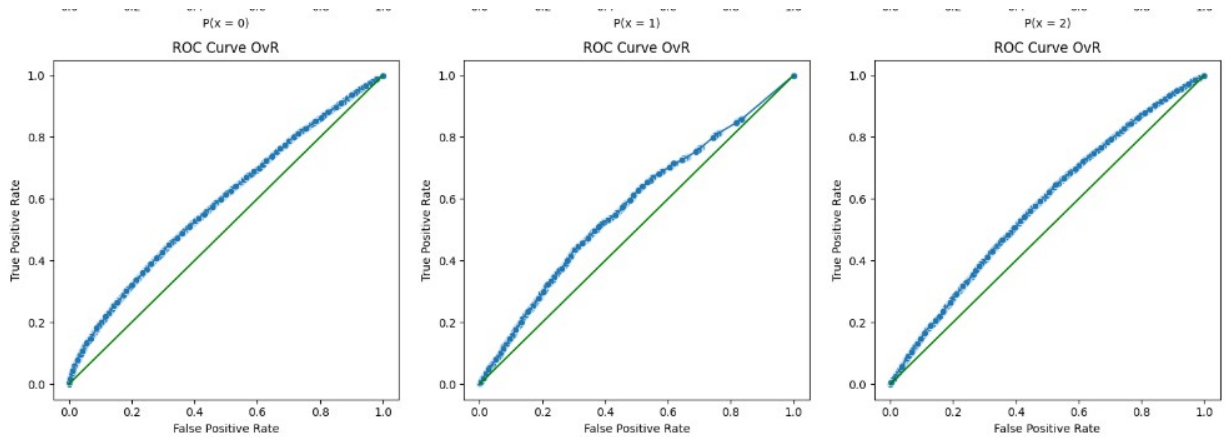
Kao kod algoritma KNN, iskoristićemo OvR metodu za evaluaciju modela.



Slika 1: Stablo odlučivanja – AUC kriva



Slika 2: Stablo odlučivanja GridSearchCV – AUC kriva



Slika 3: Stablo odučivanja RandomForest – AUC kriva

SVM algoritam

Ideja je pronaći pravu (ili u više dimenzija hiperravan) koja razdvaja podatke iz dve klase. Takvih razdvajajućih hiperravni ima više, treba izabrati najbolju - onu koja se nalazi na bezbednom rastojanju i od jedne, i od druge klase.

Ako podaci nisu baš linearno razdvojivi, koristimo meku marginu - dopuštamo neke instance sa pogrešne strane granice.

Ako podaci nisu uopšte linearno razdvojivi, koristimo kernele.

GridSearch CV

Matrica sa parametrima:

```
params = [{
                'kernel': ['linear'],
                'C': [0.01, 0.1, 1, 10],
            },
            {
                'kernel': ['rbf'],
                'C': [0.01, 0.1, 1, 10],
                'gamma': [0.01, 0.1, 1, 10],
            },
          ]
```

Najbolji parametri: {'C': 1, 'gamma': 0.1, 'kernel': 'rbf'}

Najbolja ocena: 0.668752624947501

```
Classification report for model SVC on test data
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.58      0.48      0.52      9637
           1       0.06      0.04      0.05      1049
           2       0.56      0.67      0.61     10205

    accuracy                           0.55     20891
   macro avg       0.40      0.40      0.39     20891
weighted avg       0.54      0.55      0.54     20891

--------------------------------------------------------------------------------
Confusion matrix for model SVC on test data
--------------------------------------------------------------------------------
   Kazna  Oprema  Upozorenje
0   4625     290        4722
1    316      39         694
2   3074     314        6817
--------------------------------------------------------------------------------
```

Primećujemo da je algoritam SVM uz GridSearchCV optimizaciju dao najbolje rezultate klasifikacije.

Klasterovanje

Klasterovanje je tehnika koja se koristi kako bi se grupisali slični objekti ili podaci na osnovu njihovih karakteristika. Cilj klasterovanja je identifikovanje prirodnih grupa ili obrazaca unutar skupa podataka, pri čemu su objekti unutar iste grupe sličniji jedni drugima nego objektima u drugim grupama.

U klasterovanju, algoritmu nisu pružene unapred definisane oznake ili kategorije. Umesto toga, algoritam analizira podatke i dodeljuje objekte klasterima na osnovu njihove sličnosti. Sličnost između objekata određuje se uzimajući u obzir karakteristike ili atributa.

Često korišćeni algoritmi klasterovanja uključuju K-sredina, hijerarhijsko klasterovanje i DBSCAN (klasterovanje na osnovu gustine prostornih podataka sa šumom).

Pretprocesiranje

Klase za klasterovanje smo napravili na isti način kao i klase za klasifikaciju.
Ove atribute smo skalirali korišćenjem Standard Scalera.

Algoritam K sredina

K-sredina (K-means) je jedan od najčešće korišćenih algoritama klasterovanja u mašinskom učenju i analizi podataka. Cilj mu je da podeli dati skup podataka na k klastera, pri čemu svaka tačka podataka pripada klasteru sa najbližim srednjim vrednostima (centrom).

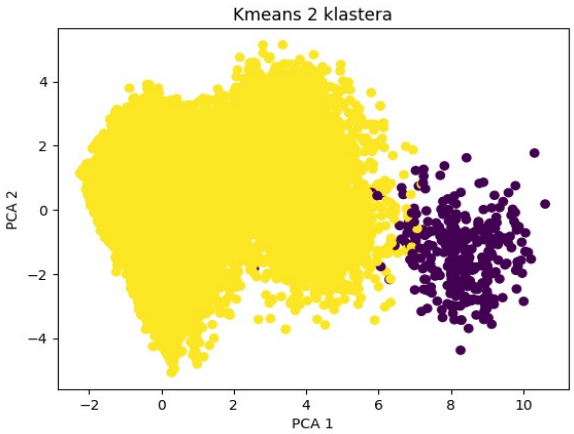
Takođe, cilj K-sredina algoritma je da minimizuje zbir kvadratnih udaljenosti između tačaka podataka i njihovih odgovarajućih klaster centara.

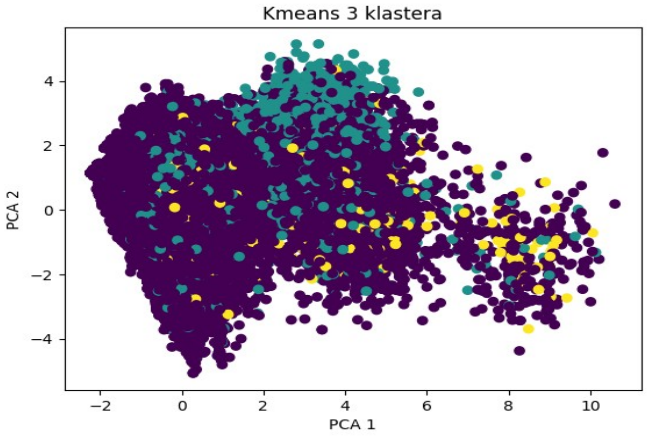
Koristio sam tri tipa ovog algoritma:

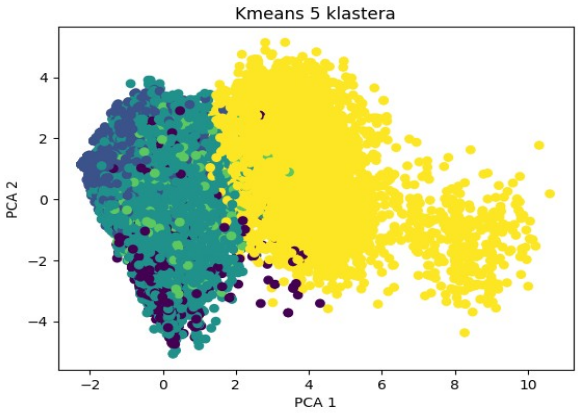
- K means
- Bisecting K means - Prvo podeliti instance u 2 klastera, zatim izabrati jedan od postojećih i podeliti ga na 2 klastera.
  Proces se ponavlja dok se ne formira k klastera
- Fuzzy C means - Fuzzy C Means algoritam je predstavnik soft clustering algoritama koji dozvoljavaju da tačke pripadaju istovremeno većem broju klastera sa različitim stepenom pripadnosti.
  C-means u nazivu označava C centroida (identično kao kod K-means).

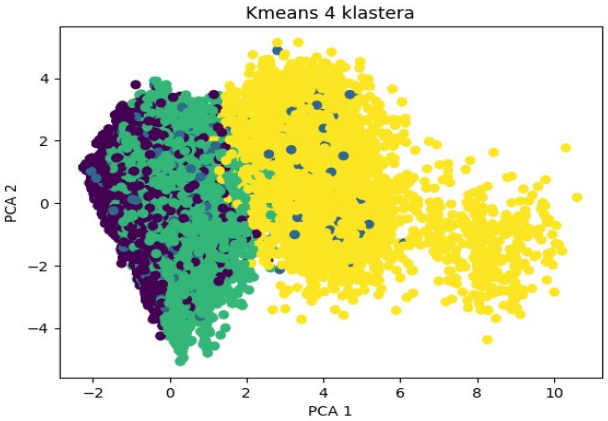
K means i Bisecting K Means algoritmi pripadaju hard clustering algoritmima, gde jedna tačka pripada najviše jednom klasteru.

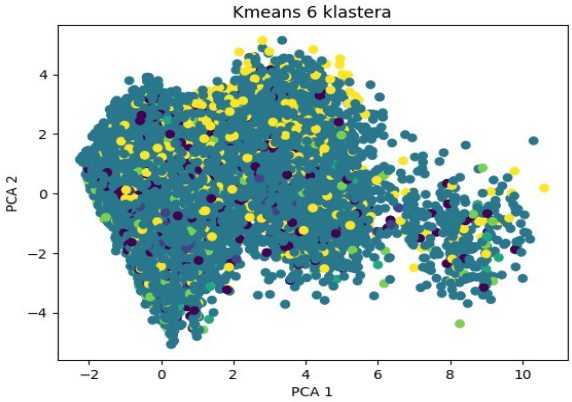
K means

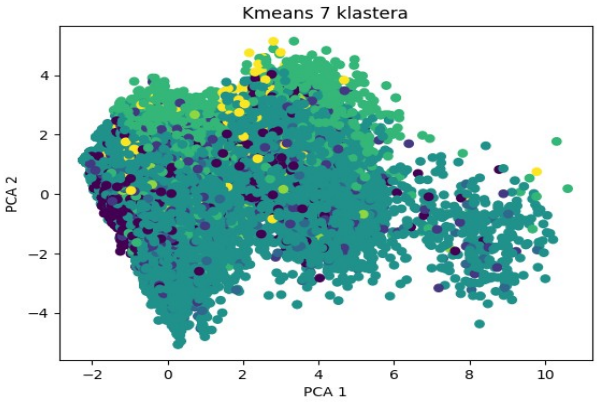

Kmeans 2 klastera
PCA 2
PCA 1

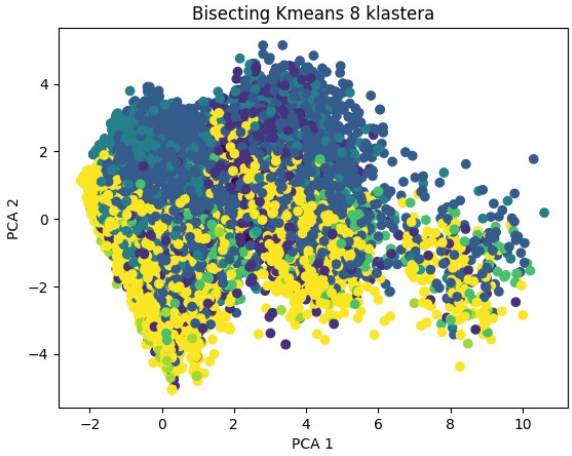

Kmeans 3 klastera
PCA 2
PCA 1

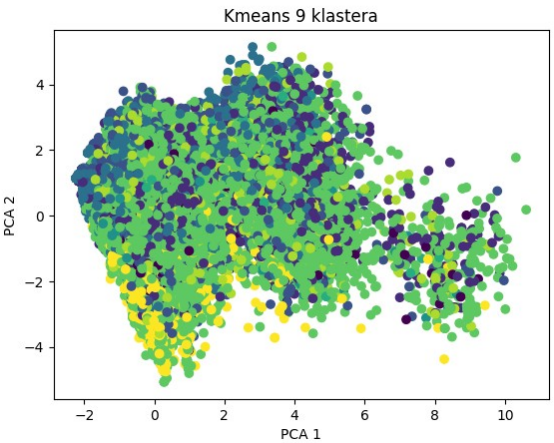

Kmeans 5 klastera
PCA 2
PCA 1

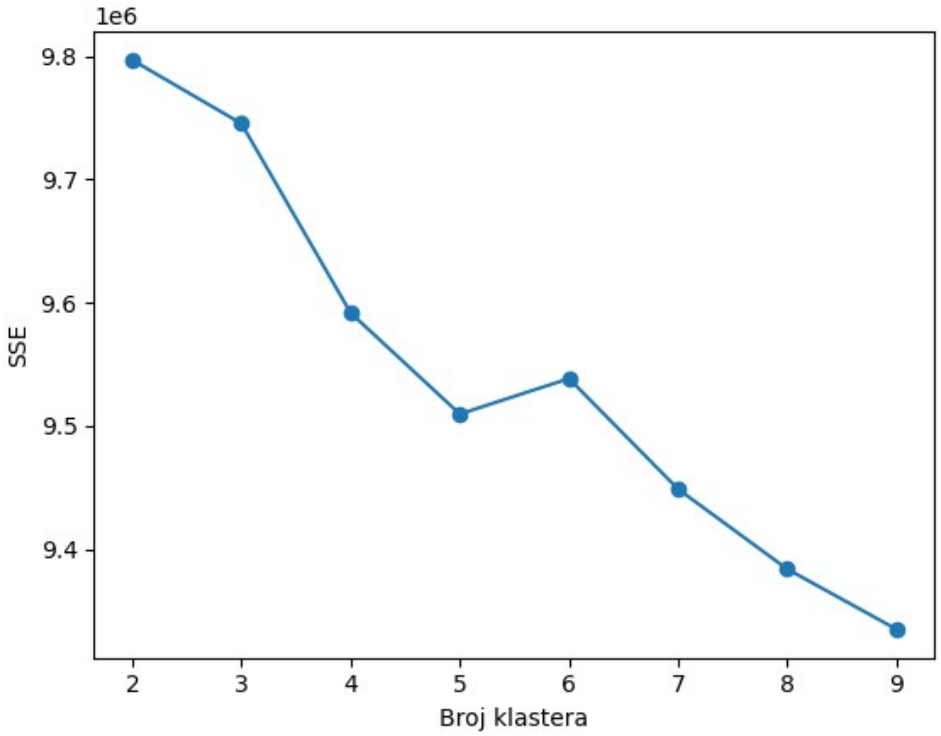

Kmeans 4 klastera
PCA 2
PCA 1


Kmeans 6 klastera
PCA 2
PCA 1

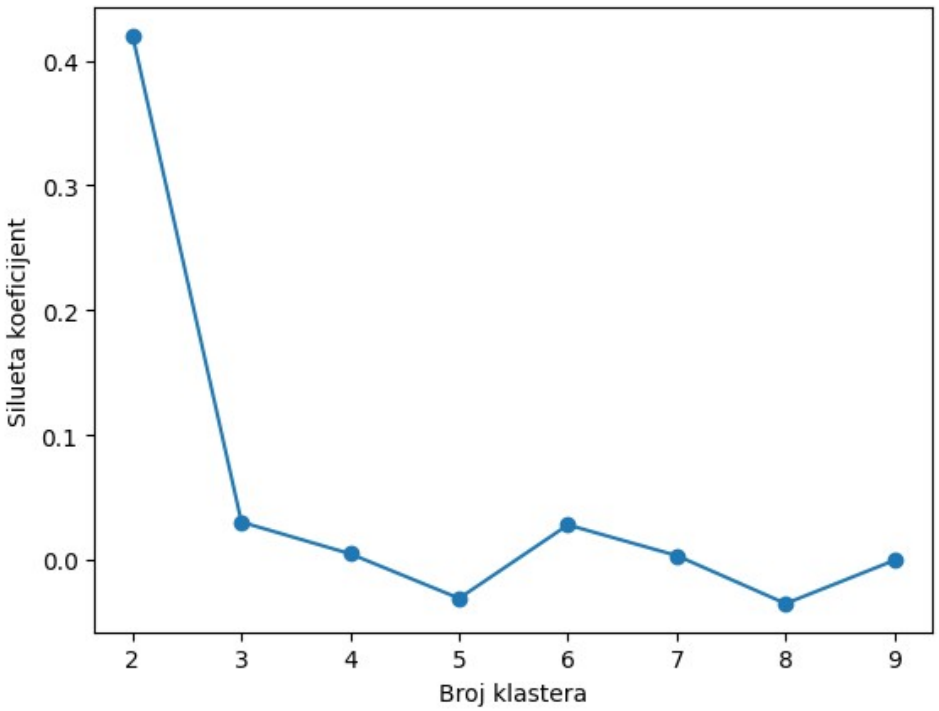

Kmeans 7 klastera
PCA 2
PCA 1


Bisecting Kmeans 8 klastera
PCA 2
PCA 1


Kmeans 9 klastera
PCA 2
PCA 1

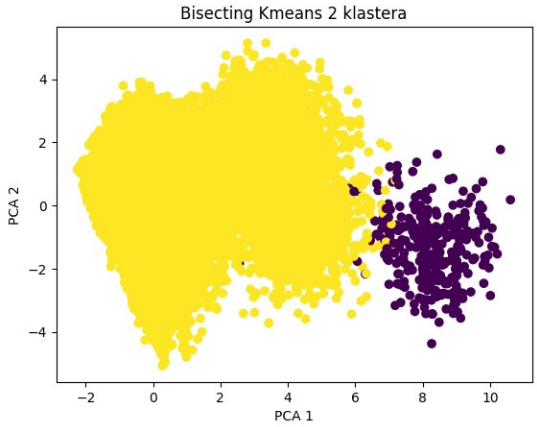


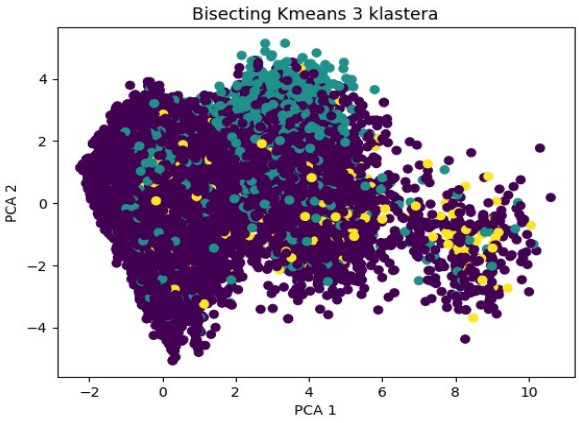
Očekivano da SSE opada sa porastom broja klastera.

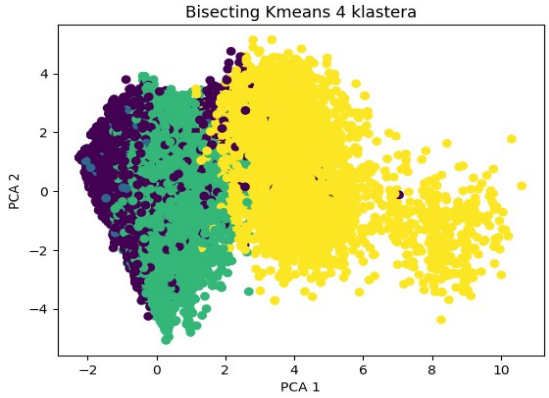


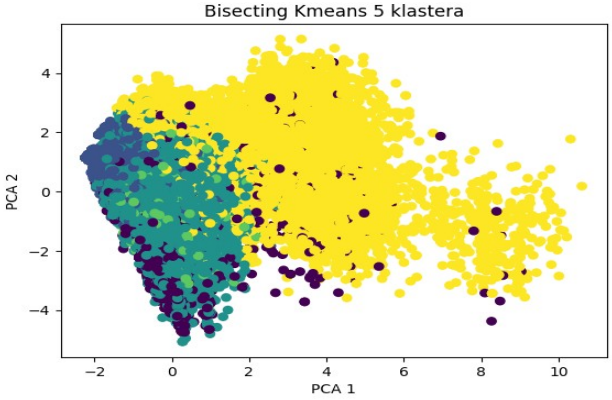
Koristeći pravilo lakta, možemo zaključiti da je optimalan broj klastera 7.

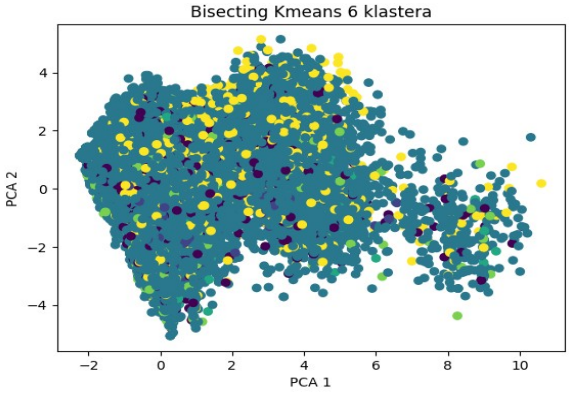
Bisecting K means

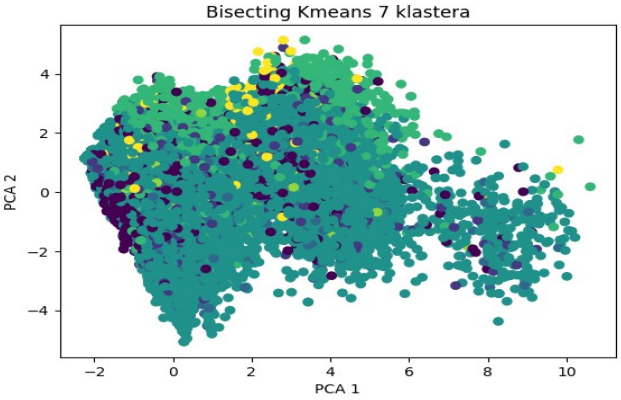


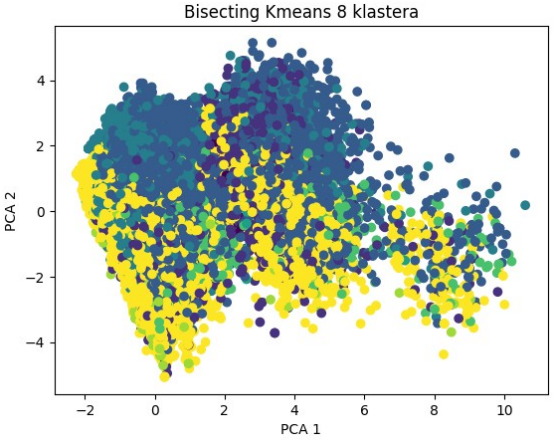


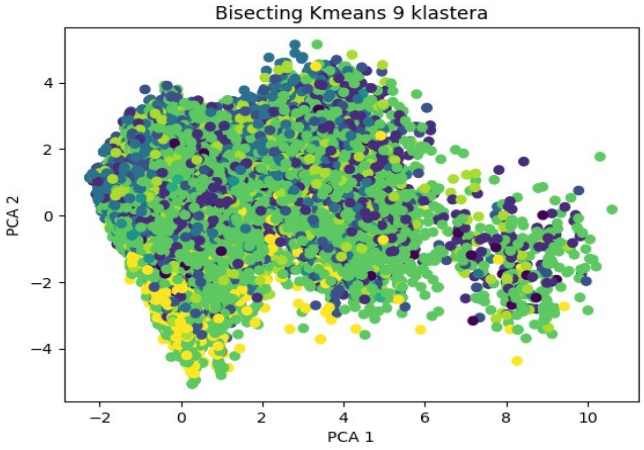


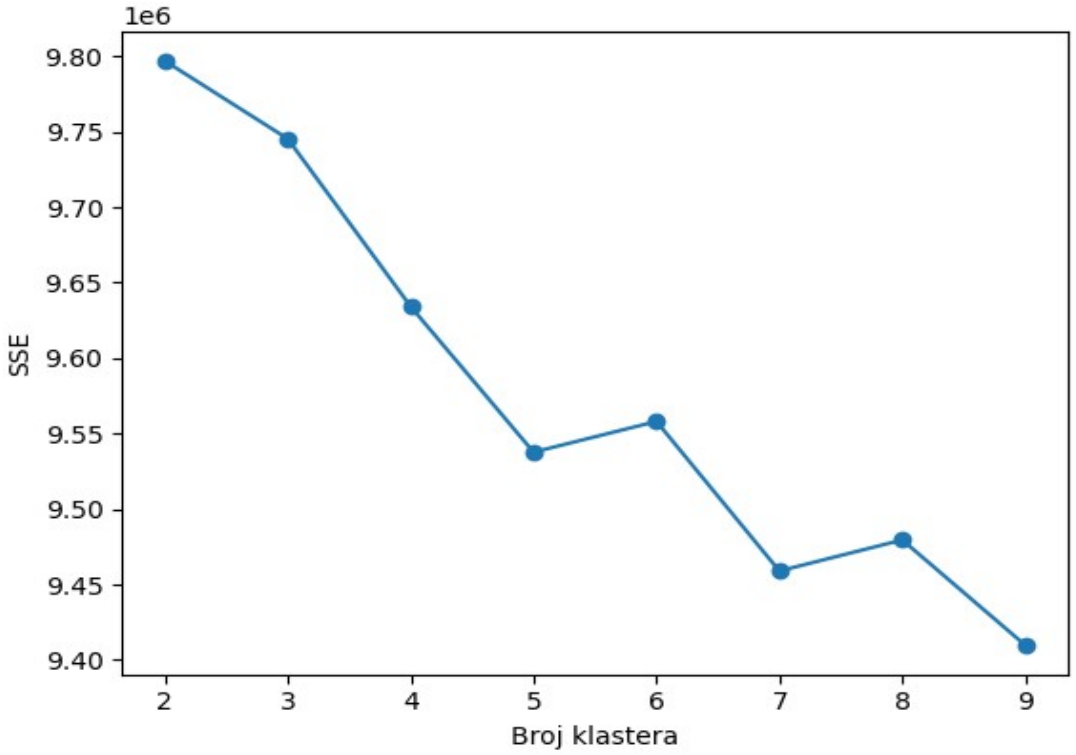


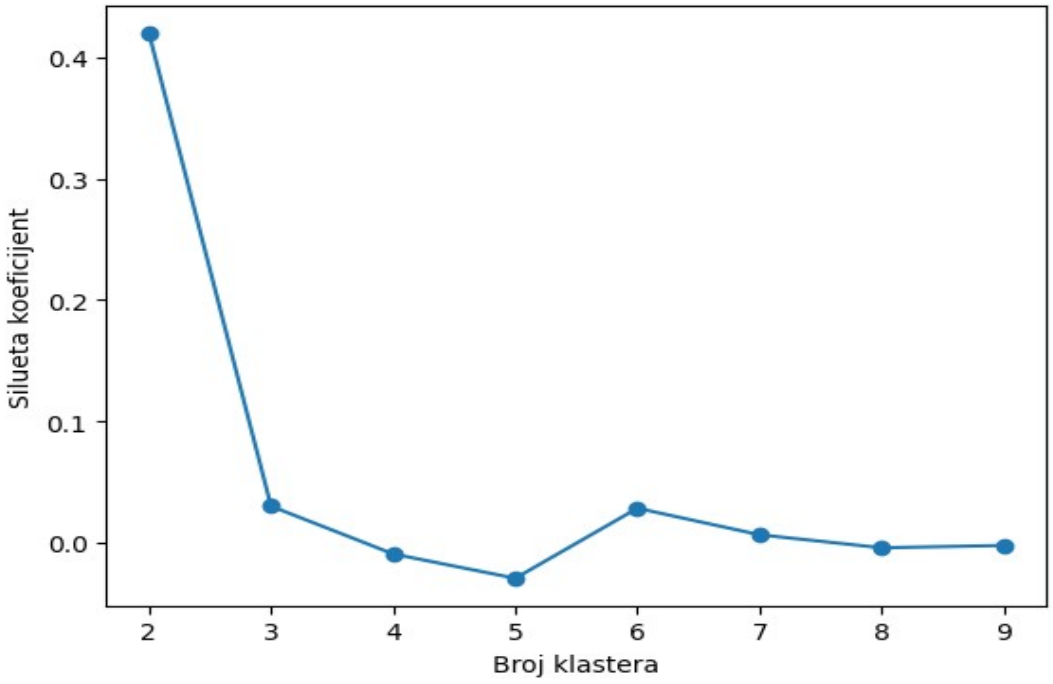






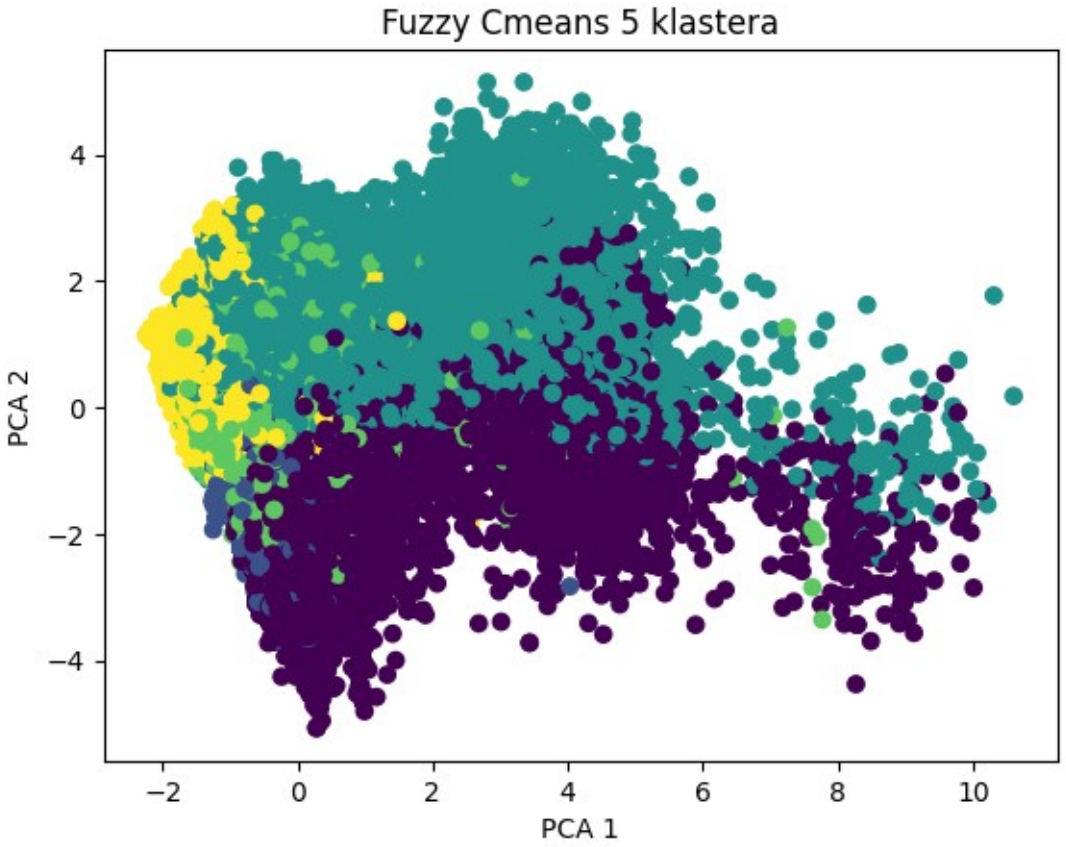

Bisecting Kmeans 9 klastera
PCA 2
PCA 1
4
2
0
−2
−4
−2
0
2
4
6
8
10

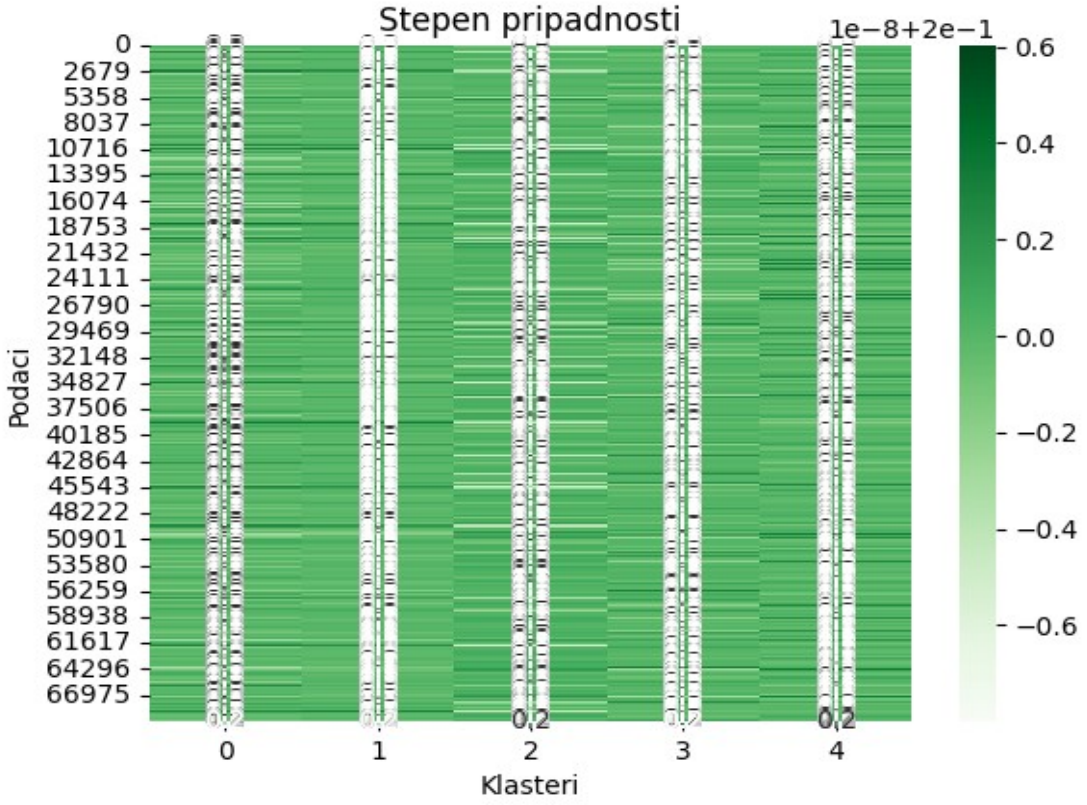




Koristeći pravilo lakta, zaključujemo da je optimalan broj klastera 7.

Fuzzy C means





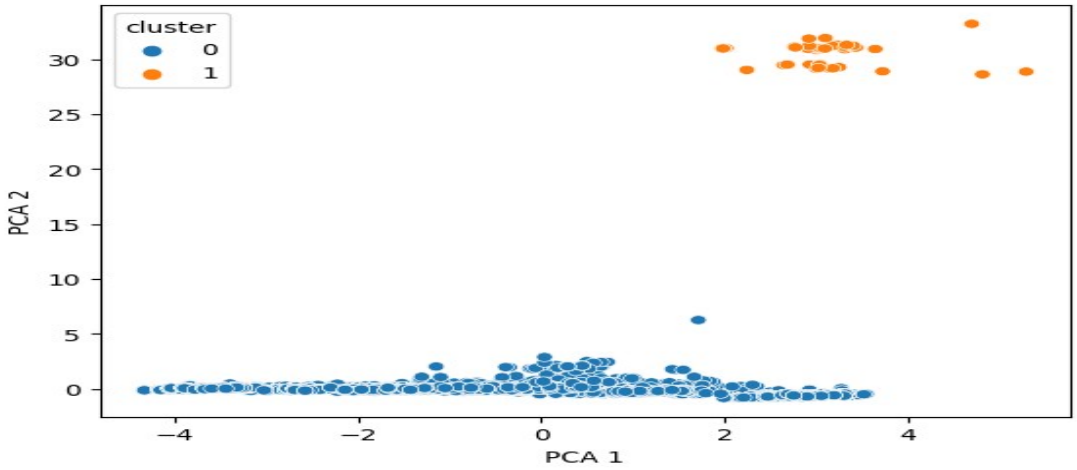
Algoritam sakupljajućeg klasterovanja

Sakupljajuće klasterovanje (Agglomerative Clustering) je algoritam hijerarhijskog klasterovanja koji se koristi za grupisanje sličnih tačaka podataka u klasterima.

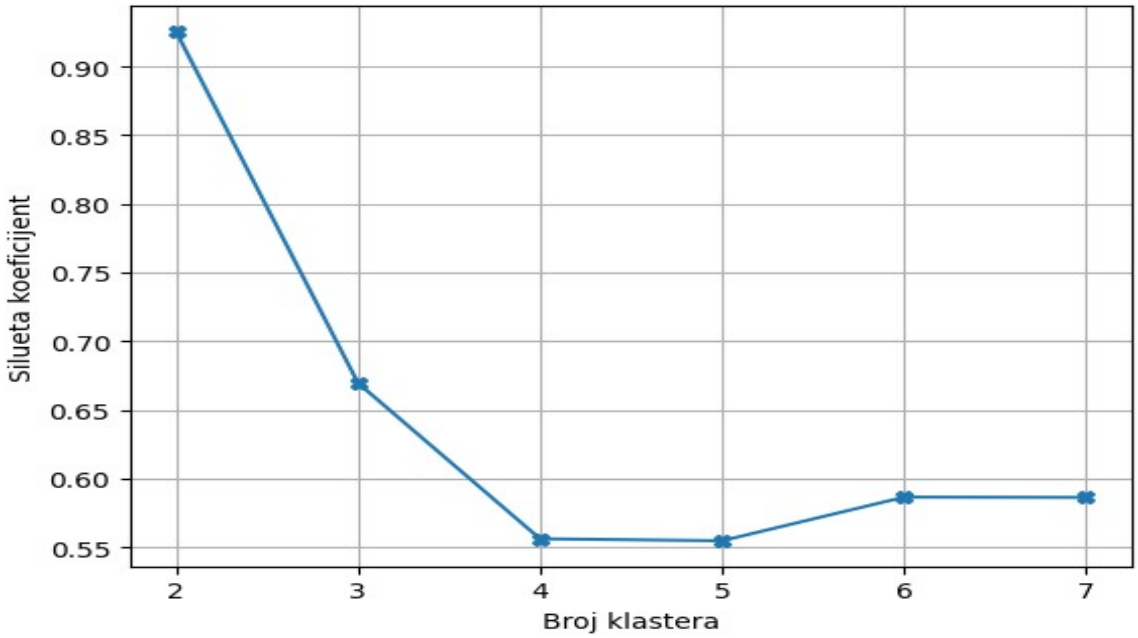
Ovo je pristup odozdo prema gore (bottom-up), gde svaka tačka podataka počinje kao sopstveni klaster, a zatim se iterativno spajaju klasteri na osnovu njihove sličnosti sve dok se ne dostigne željeni broj klastera.

Pretprocesiranje skupa podataka i priprema za ovaj algoritam je identična kao za algoritam K sredina.

Napravićemo najbolji model u odnosu na silueta koeficijent.

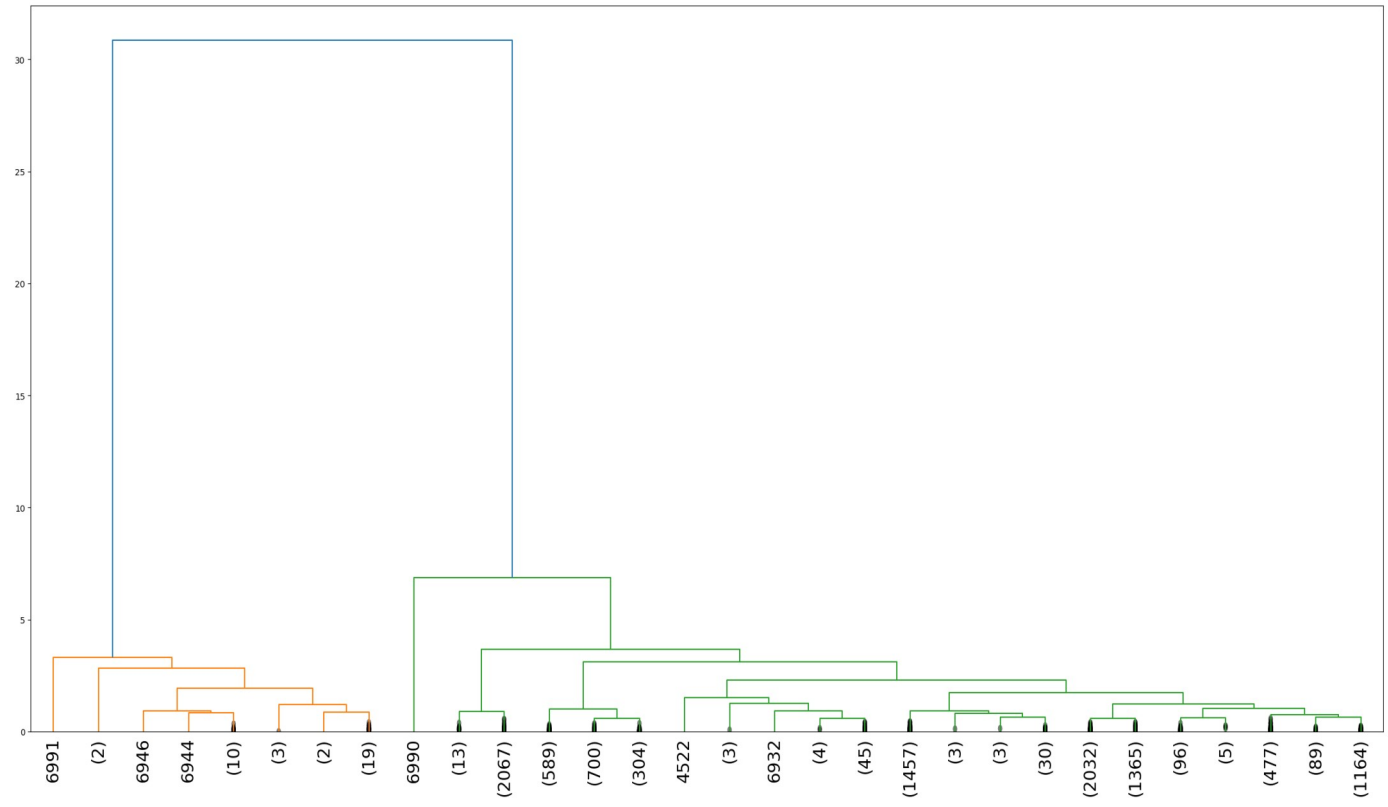


Vidimo da sadrži dva klastera.
Njegova ocena je: 0.9256169698570447

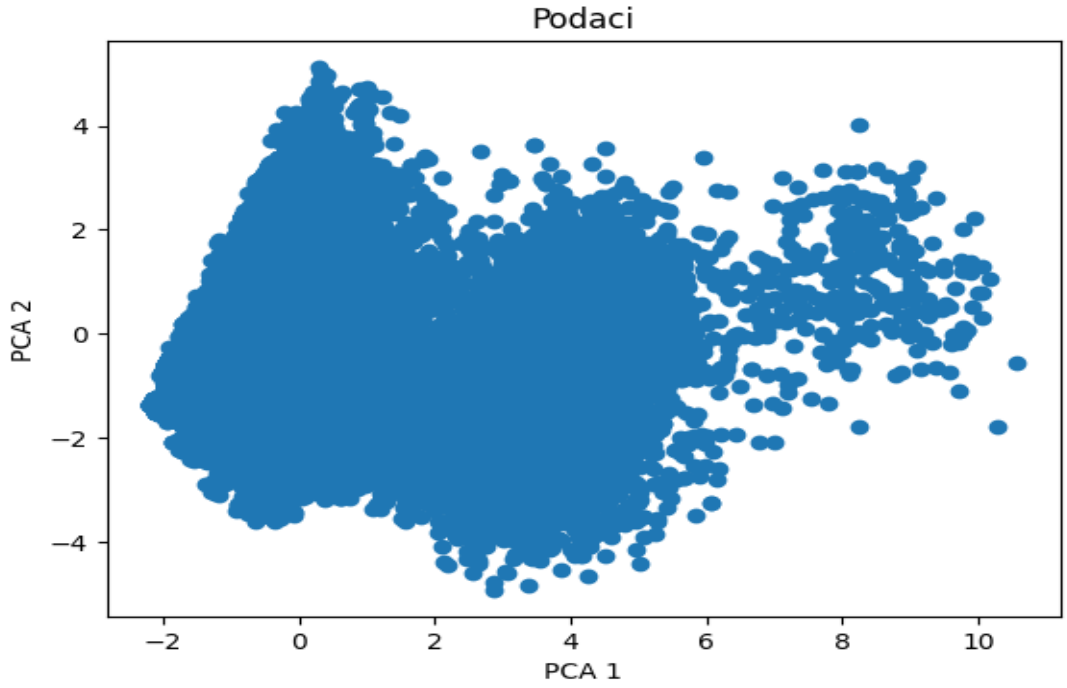


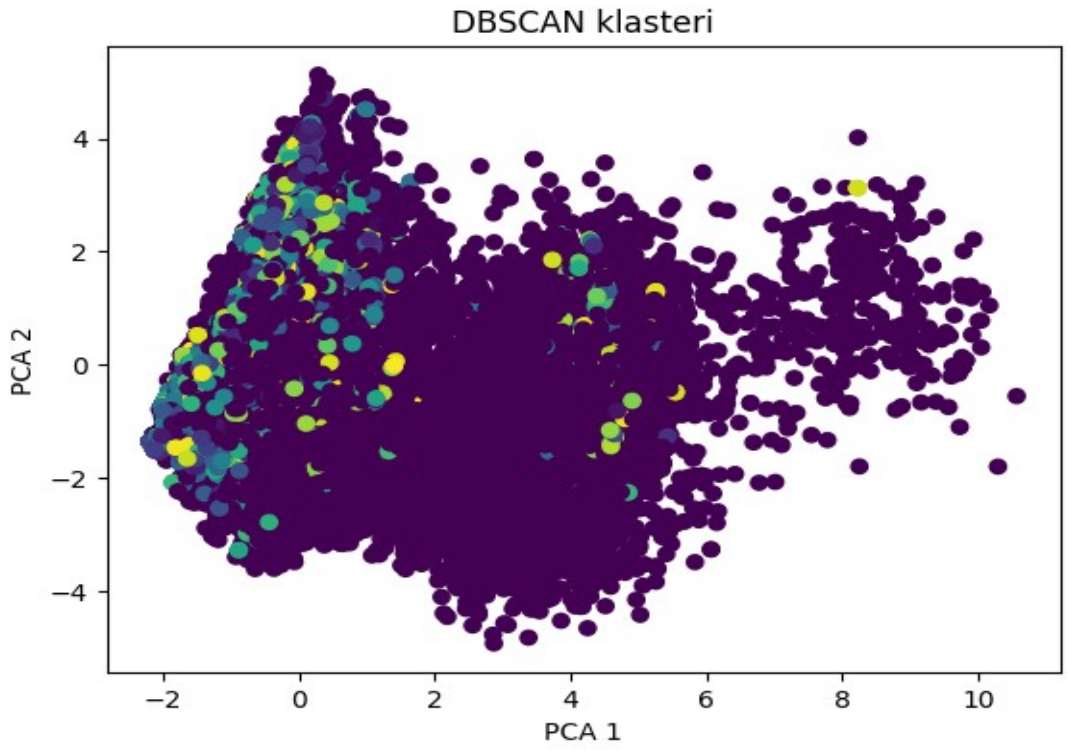
Na osnovu pravila lakta zaključujemo da je optimalan broj klastera 4.

Pogledajmo dendrogram najboljeg modela.



Algortiam DBSCAN





Silueta koeficijent modela iznosi 0.2124265816275008

Pravila Pridruživanja – SPSS

Koristimo ovu metodu kada imamo veliki broj atributa i želimo da odredimo koji od tih atributa su medusobno povezani.
Pravila pridruživanja nam upravo služe da nađemo veze između podataka.
Ove veze su nam znacajne jer time dobijamo bolje razumevanje podataka

Apriori Algoritam

Jedan od najpoznatijih algoritama za izdvajanje pravila pridruzivanja je Apriori algoritam.

Apriori algoritam u fazi generisanja čestih skupova stavki koristi osobine podrške kako bi se smanjio broj skupova stavki za koje je potrebno izračunati podršku da bi se odredilo da li je skup stavki čest.
IBM SPSS Modeler nam omogućava laku implementaciju ovog algoritma

Napravio sam dva modela.
Kod prvog je parametar *support* postavljen na 1 i sadrži 34 instance.

Sort by: Confidence % 34 of 34

Consequent	Antecedent	Support %	Confidence %
Violation.Type = Citation	Personal.Injury Arrest.Type = A - Marked Patrol	1,079	94,993
Violation.Type = Citation	Property.Damage Gender = M Arrest.Type = A - Marked Patrol VehicleType = 02 - Automobile	1,163	94,377
Violation.Type = Citation	Personal.Injury	1,146	94,293
Violation.Type = Citation	Property.Damage Gender = M Arrest.Type = A - Marked Patrol	1,339	94,268
Violation.Type = Citation	Property.Damage Gender = M VehicleType = 02 - Automobile	1,197	94,181
Violation.Type = Citation	Property.Damage Gender = M	1,389	93,859
Violation.Type = Citation	Contributed.To.Accident Gender = M Arrest.Type = A - Marked Patrol	1,55	93,578
Violation.Type = Citation	Contributed.To.Accident Gender = M Arrest.Type = A - Marked Patrol VehicleType = 02 - Automobile	1,349	93,572
Violation.Type = Citation	Contributed.To.Accident Gender = M	1,392	93,565

Dok drugi model ima parametar *support* postavljen na 2 i sadrži samo 5 vrednosti

Consequent	Antecedent	Support %	Confidence %
Violation.Type = Citation	Contributed.To.Accident Arrest.Type = A - Marked...	2,295	92,379
Violation.Type = Citation	Contributed.To.Accident Arrest.Type = A - Marked... VehicleType = 02 - Auto...	2,017	92,319
Violation.Type = Citation	Contributed.To.Accident VehicleType = 02 - Auto...	2,088	91,831
Violation.Type = Citation	Contributed.To.Accident	2,396	91,691
Violation.Type = Citation	Belts Gender = M	2,219	60,858

Drugi model ima bolju podršku, ali je skup poprilično raznovrstan.

Kada spustimo granicu dobijamo više pravila (Model 1).

Zaključak

Kada je u pitanju klasifikacija, algoritam koji je pokazao najbolje rezultate na ovom skupu je SVM algoritam uz GridSearchCV optimizaciju sa tačnošću 0.68 %

Najbolje rezultate algoritama klasterovanja dao je algoritam sakupljajućeg klasterovanja sa ocenom 0.93.

Zaključak Apriori algoritma u IBM SPSS modeleru je da su svi vozači bili muškarci, kao i da su zaustavljeni od strane policajca na dužnosti, tokom vožnje.

Link do projekta:
MATF-istrazivanje-podataka-1/2023_Data_Mining_Traffic_violations_Dataset_(github.com)