Matematički fakultet

Seminarski rad iz Istraživanja podataka

Dijagnostika motornog pogona bez senzora

Autor:
Mihailo Dedić

Profesor:
Prof. Dr. Nenad Mitić

Asistenti:
Marija Erić
Stefan Kapunac

June 22, 2023

Contents

1 Uvod

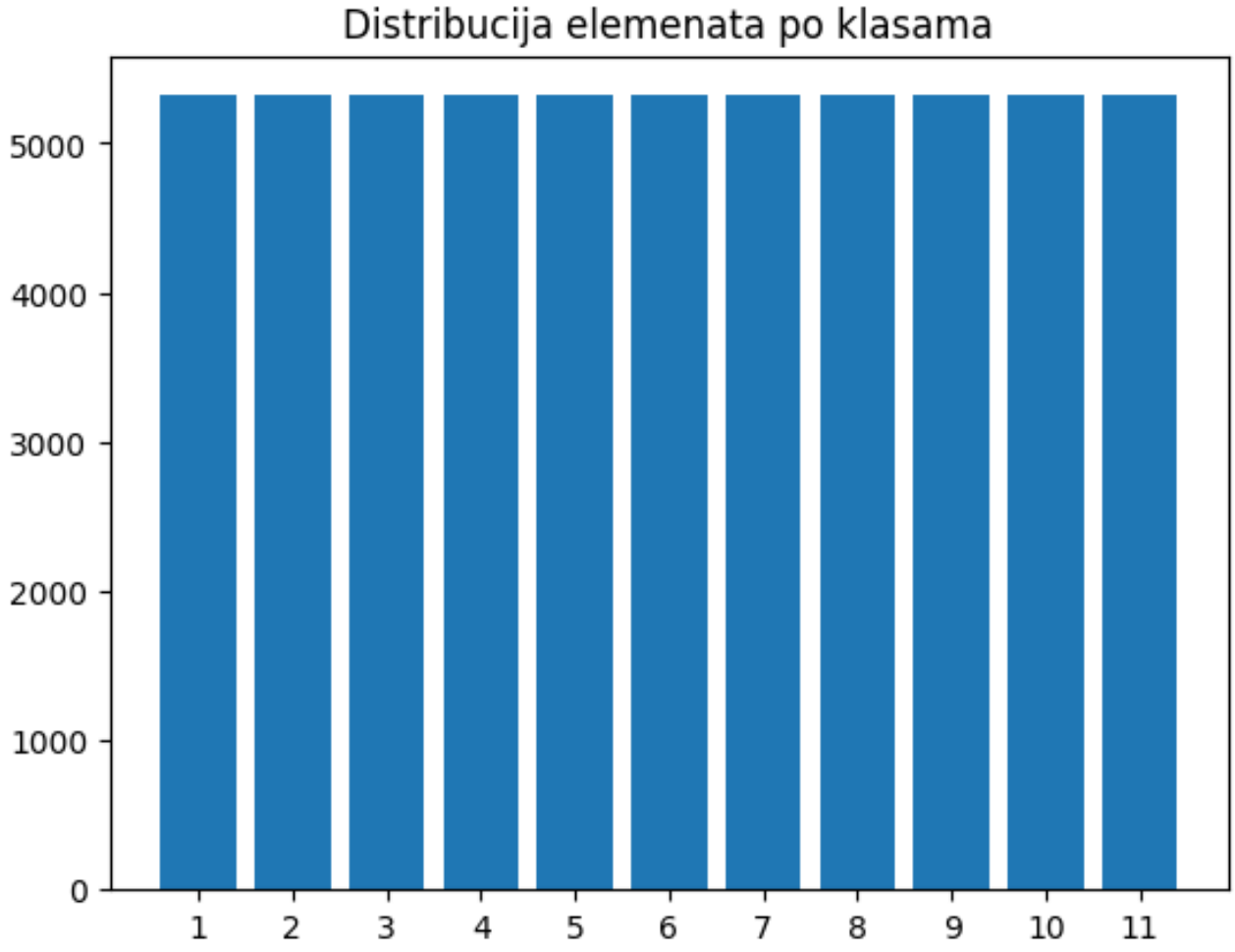
U elektromehaničkim pogonima za vožnju stanja grešaka sinhronih motora mogu se detektovati merenjem faze struje motora, i koristiti za njihovo unapreenje.
Skup podataka prikupljen je merenjem fazne struje motora u okviru projekta AutASS. Sastoji se od 58509 instanci sa 48 neprekidnih atributa i jednim klasnim kategoričkim koji može imati jednu od 11 vrednosti tj. stanja - ispravno, ili 10 različitih neispravnih. Simulirane i evaluirane su različite greške u komponentama motora, pod različitim brzinama silama i opterećenjima. Razlog njihovog pojavljivanja mogu biti loše sastavljanje, istrošenost ili preopterećenje delova. Simulacije su vršene na različitim kombinacijama ispravnih i neispravnih komponenti i na osnovu njih zabeleženi su različiti signali struje motora. Ti signali su opisani pomoću četiri vrednosti redom: sredina, std devijacija, iskrivljenost, kurtoza; i to čini naših 48 atributa.

2 Ekspolrativna analiza

Naš skup podataka smo učitali u Pandas DataFrame df, i podelili ga na skup ulaznih atributa X i ciljni(klasni) atribut y.

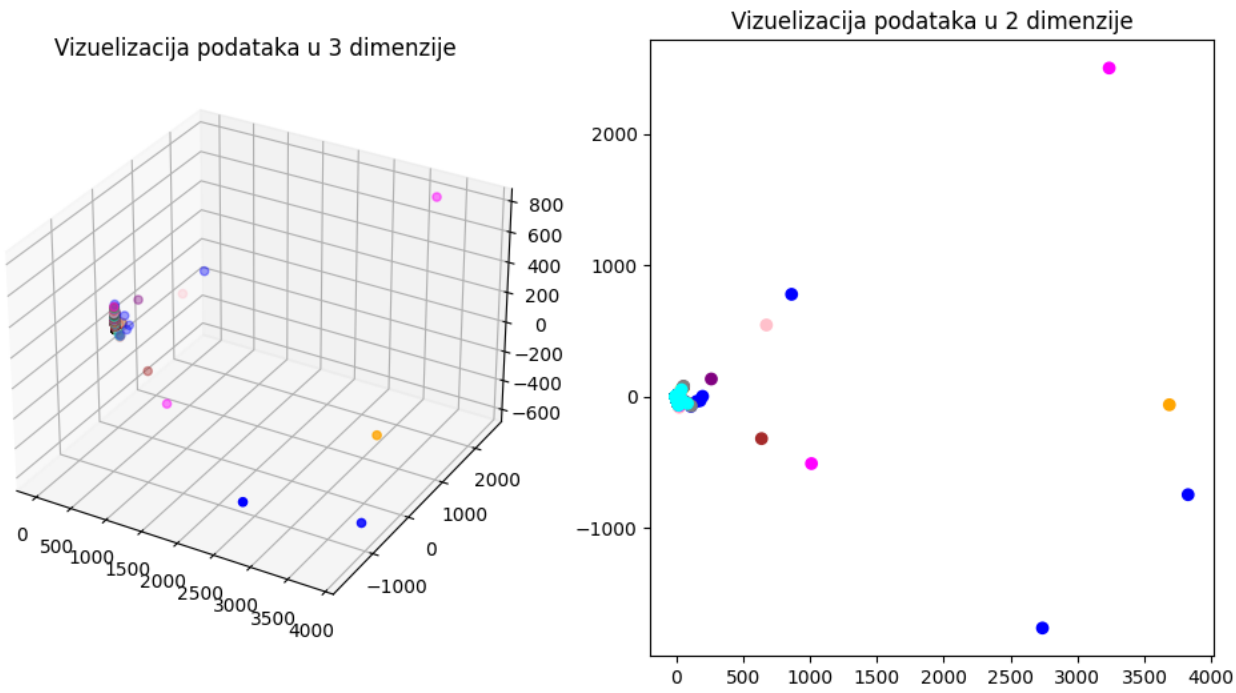
Prva provera koju smo izvršili bila je da li postoje nedostajuće vrednosti - ispostavlja se da nemamo nedostajuću vrednost ni u jednoj od kolona.

Sledeće što proveravamo je balansiranost klasa. Kao što se moze videti iz slike 2.1 klase su potpuno balansirane - svakoj od klasa pripada tačno po 5319 instanci.



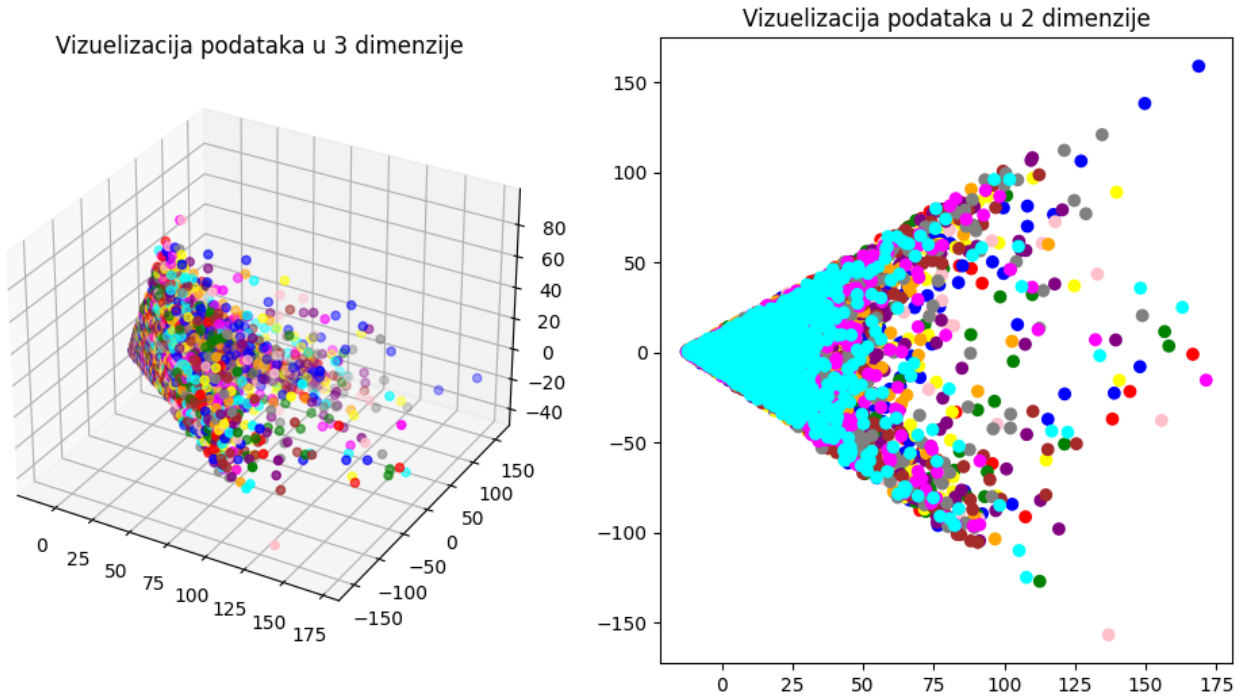
slika 2.1

Sada ćemo pokušati da napravimo vizualizaciju naših podataka. Koristimo PCA tehniku (koja će detaljnije biti opisana u poglavlju 3.) da bismo vizuelizovali nase podatke. Na slici 2.2 prikazani su redom podaci u 3 i 2 dimenzije.



slika 2.2

Može biti problematično opisati šta su tačno elementi van granica, a pritom ne izgubiti neke važne elemente. Na osnovu slike 2.2 jasno se vidi postojanje bar 9 njih. Pošto bi uklanjanje metodom (k = 3) standardne devijacije izbacilo oko 30% elemenata iz skupa, podići ćemo parametar na k = 116 kako bi uklonili samo 9 najekstremnijih elemenata van granica. Nakon ove transformacije prikazujemo redom podatke u 3 i 2 dimenzije na slici 2.3.



slika 2.3

Sada nam je ostao skup sa 58500 instanci sa kojim nastavljamo dalju obradu.

3 Pretprocesiranje u klasifikaciji

Vršimo podelu na trening(X_train) i test(X_test) skup u odnosu 70:30 u korist trening podataka, sa stratifikacijom po klasnom atributu y.

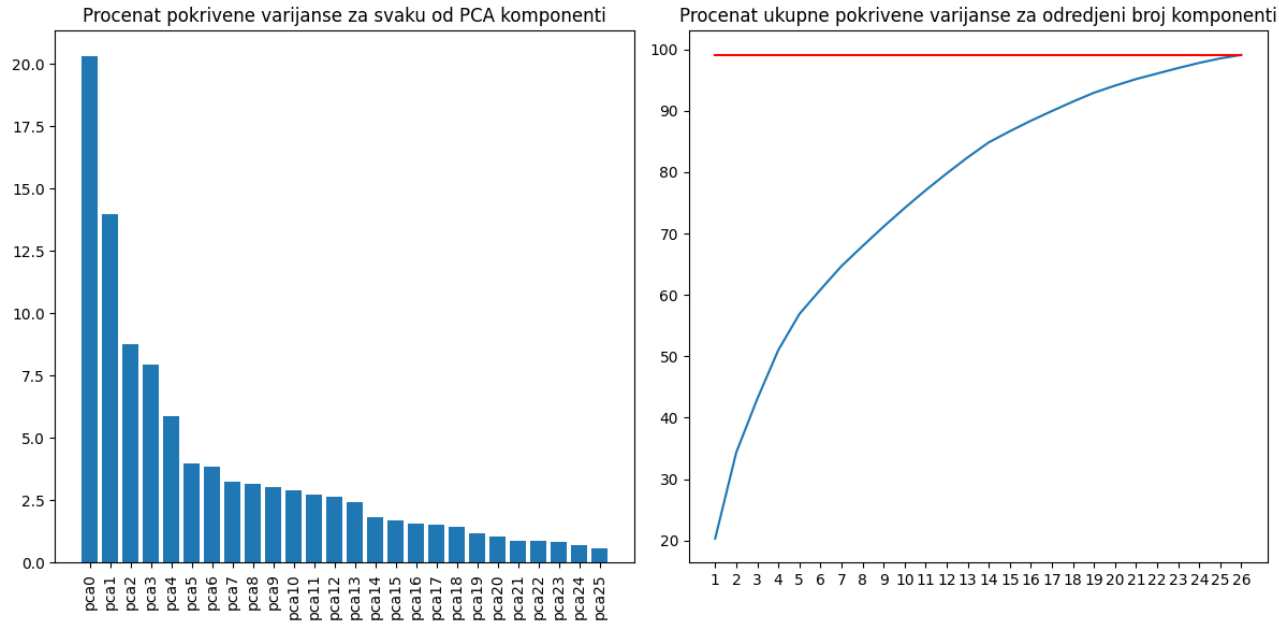
Naši atributi mere različite opsege, zbog čega ćemo za neke od modela morati da koristimo skaliranje. Napravićemo dva skupa:
1. X_train_standardized - standardizovani trening podaci
2. X_train_normalized - normalizovani trening podaci

Pošto je naš skup podataka visoko dimenzionalan primenićemo i 2 tehnike za smanjenje dimenzionalnosti - PCA i RFE.

3.1 Analiza glavnih komponenti(PCA analiza)

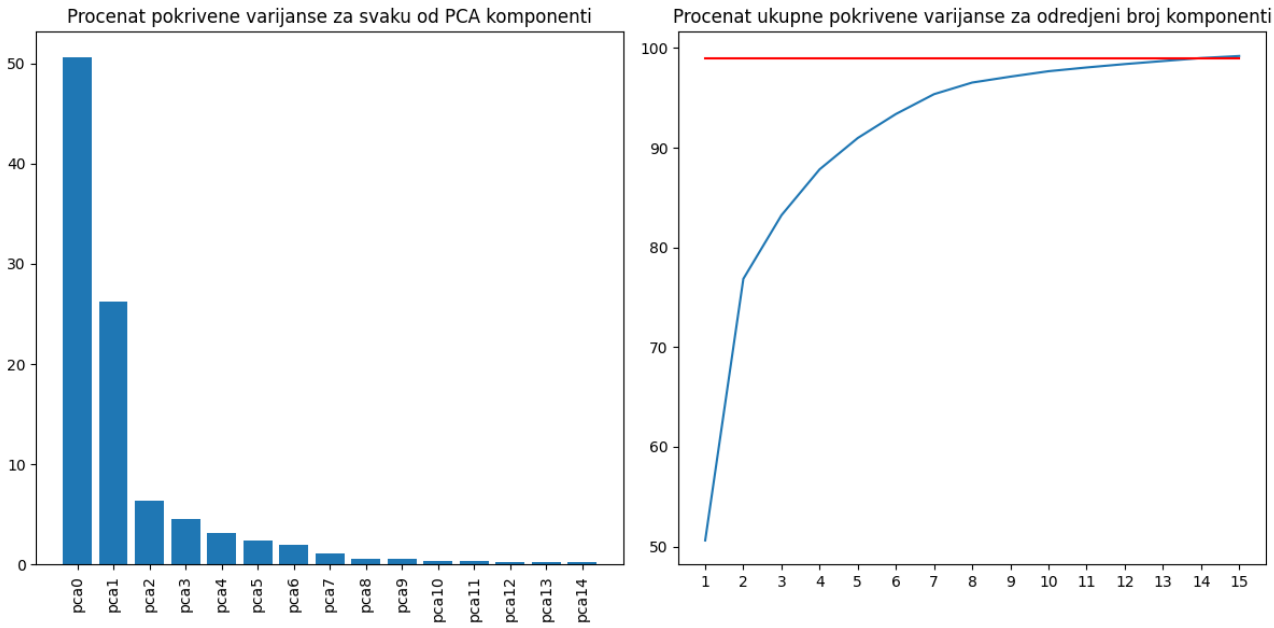
Analiza glavnih komponenti (en. Principal Component Analysis(PCA)) predstavlja tehniku smanjenja dimenzionalnosti korišćenjem linearnih (PCA) vektora dobijenih takvom linearnom kombinacijom atributa za koju je varijansa (pokrivenost) podataka najveća. PCA tehniku je neophodno primeniti nad skaliranim podacima kako bismo mogli da dobijemo maksimalnu varijansu i iz atributa manje veličine. Primenićemo PCA i za standardizovane i za normalizovane podatke i zatim uporediti rezultate. Pri tome uzećemo da je željena pokrivenost varijanse 99%.

Na slici 3.1 prikazani su procenat pokrivene varijanse za svaku od PCA komponenti, kao i procenat ukupne pokrivene varijanse za odredjeni broj komponenti nastalih primenom PCA nad standardizovanim podacima.



slika 3.1

Na slici 3.2 prikazani su procenat pokrivene varijanse za svaku od PCA komponenti, kao i procenat ukupne pokrivene varijanse za odredjeni broj komponenti nastalih primenom PCA nad normalizovanim podacima.



slika 3.2

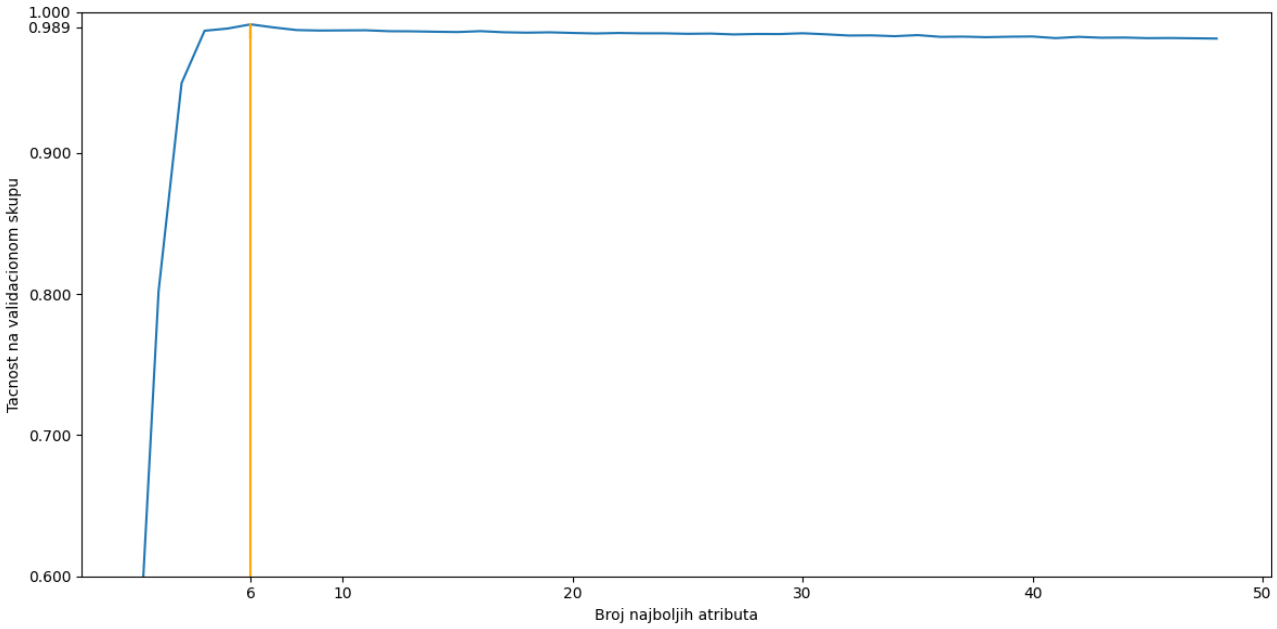
Na ovom mestu primećujemo zanimljivo ponašanje: PCA koji bi trebalo da se najbolje ponaša nad standardizovanim podacima najgore radi sa njima - ovo je iz razloga sto svi naši atributi mere istu stvar(istu meru) a s obzirom da npr. atribut 0 ima vrednosti oko 10^6 a atribut 47 vrednosti oko 10^0 jasno je da će doći do gubitaka u varijansi. Zato PCA nad normalizovanim podacima radi osetno bolje u našem primeru.

Na posletku, pravimo PCA skup sa 15 komponenti nad normalizovanim podacima - X_train_normalized.

3.2 Rekurzivna eliminacija komponenti sa kros-validacijom(RFECV)

Druga tehnika za smanjenje dimenzionalnosti je Rekurzivna eliminacija komponenti (en. Recursive Feature Elimination with Cross Validation (RFECV)). Ovom tehnikom se izbacuju, jedan po jedan, najnebitniji atributi. Za svaki manji broj atributa se zatim vrši testiranje na validacionom skupu za model koji mi zadamo - ovde je odlučeno da to bude Drvo odlučivanja. RFECV zatim sam formira model sa optimalnim brojem atributa.

Na slici 3.3 prikazane su tačnosti na validacionom skupu za svaki mogući broj najvažnijih atibuta.



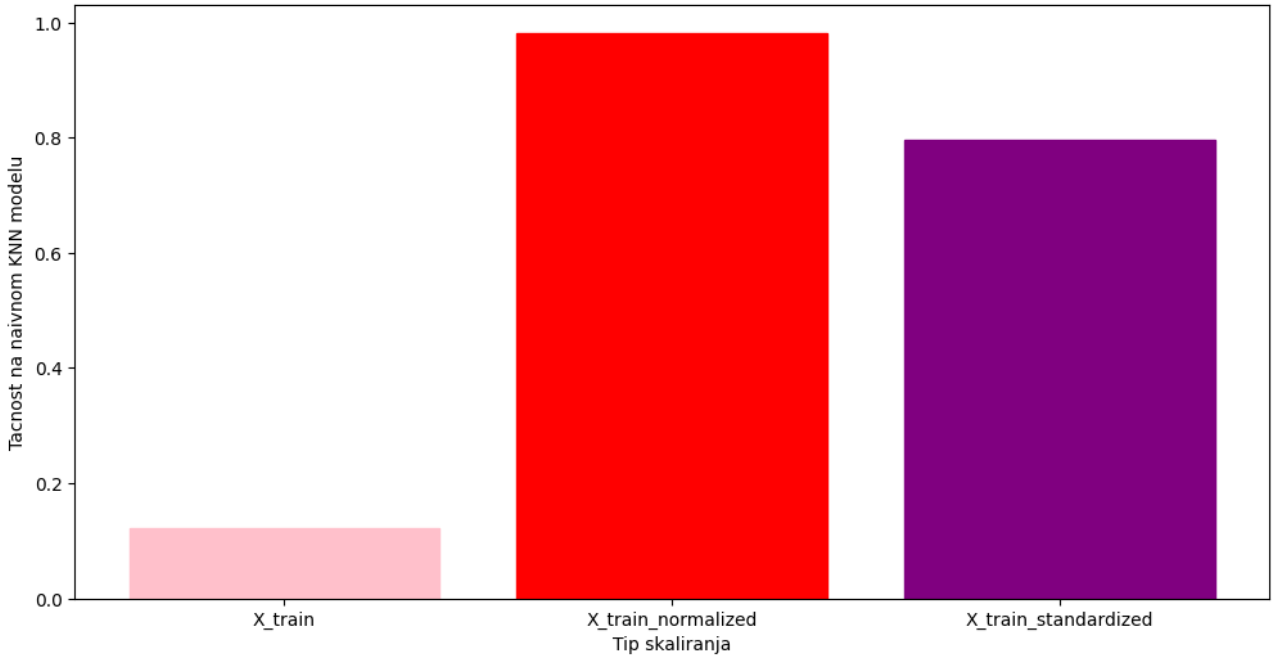
slika 3.3

Vidimo sa slike 3.3 da se optimalni model postiže za 6 najboljih atributa pa ćemo od njih formirati skup X_train_RFE.

3.3 Dodatno pretprocesiranje

U klasifikaciji ćemo koristiti tri modela: DTC, RFC i KNN(detaljnije o njima u poglavlju 4). Ono što je na ovom mestu važno da znamo je da nam za prva dva ne treba dodatno procesiranje, ali nam treba dodatno skaliranje za KNN. To je zato sto je ovaj algoritam zasnovan na udaljenostima (en. distance based), te će razlike u atributima poprilično uticati na njegovo odlučivanje.

Na slici 3.4 prikazana je tačnost algoritma na validacionom skupu za obične, normalizovane i standardizovane podatke.



slika 3.4

Na osnovu slike zaključujemo da je i ovde najbolje raditi sa normalizovanim podacima, te formiramo sledeće skupove:
1. X_train_basic_KNN
2. X_train_PCA_KNN
3. X_train_RFE_KNN

3.4 Zaključak procesiranja u klasifikaciji

Za svaki od navedinih trening skupova iz prethodnih poglavlja formiramo i analogni test skup. Na kraju ih čuvamo u 4 mape skupova:
1. train_sets, test_sets
2. train_sets_KNN, test_sets_KNN

4 Klasifikacija

U okviru klasifikacije iskoristićemo podatke dobijene iz poglavlja 3. Radićemo sa sledećim modelima - DTC, RFC i KNN.

Svaki od njih ćemo ubaciti u Grid Search(GS) model. GS na osnovu matrice parametara i odgovarajućeg estimatora koji predstavlja jedan od naših modela vraća najbolji model na osnovu nekog kriterijuma odredjivanja najboljeg za odgovarajući validacioni skup.

Mi ćemo uvek koristiti sledeće parametre:
1. cv=5 - primeni validaciju za 5 različitih validacionih skupova
2. scoring=accuracy - kriterijum tačnosti
3. verbose=4 - ispiši najviše moguće informacija

Model i odgovarajuća matrica parametara biće naknadno odredjeni za svaki model. Takodje, vršićemo ovaj postupak za svaki od trening skupova, i zatim testirati rezultate na tim trening skupovima i svim odgovarajućim test skupovima.

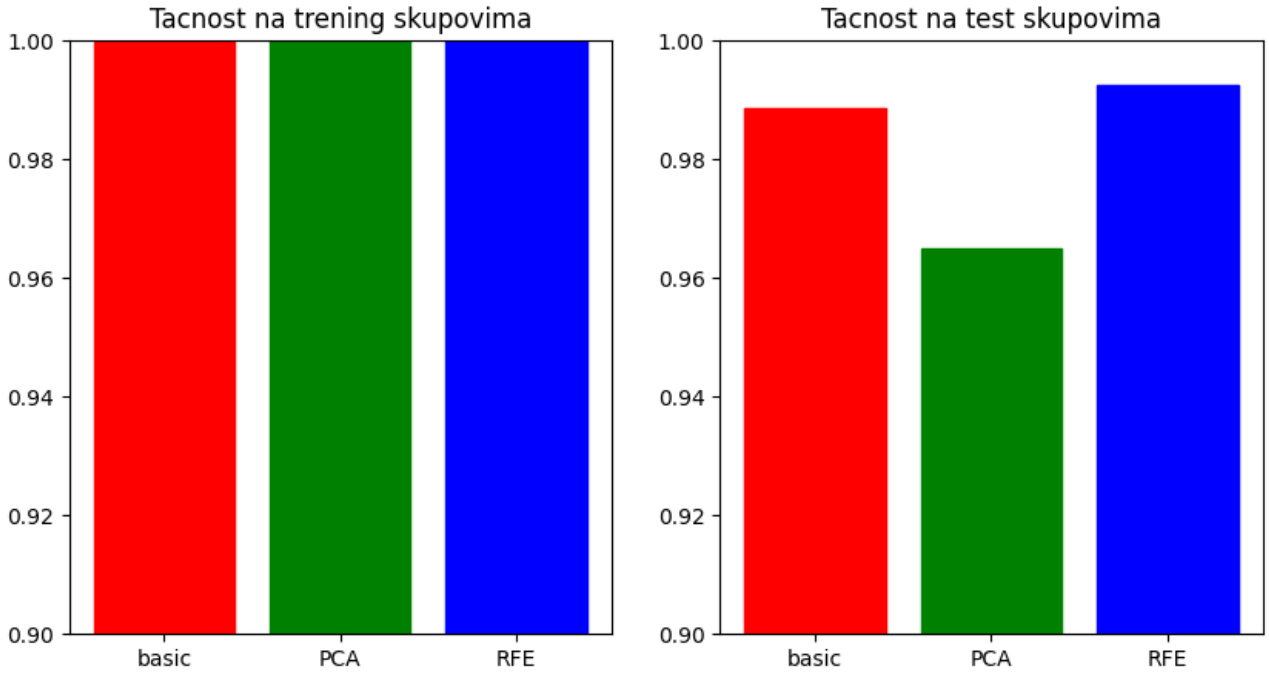
4.1 Drvo odlučivanja(DTC)

Prvi model koji ćemo koristiti je drvo odlučivanja (en. Decision Tree Classifier (DTC)). Odlučili smo se za ovaj model zbog njegove jednostavnosti - obično se pokazuje kao dobar prvi model sa kojim treba da radimo. DTC formira drvo odlučivanja u kome su čvorovi uslovi za neki od atributa, a listovi instance. Kroz drvo se krećemo prateći navedene uslove i tako dolazimo do listova za koje se nadamo da su sto "čistiji" - da skoro ceo list budu instance jedne klase. Tada možemo zaključiti na osnovu kojih uslova stižemo do kojih klasa, i to primenjivati za nepoznate podatke.

Koristimo sledeću matricu parametara:
1. criterion=[gini, entropy] - parametar koji opisuje odredjivanje najboljeg atributa za podelu
2. min_samples_split=[2,3,4,5,6] - parametar koji odredjuje minimalan broj instanci za koji smemo da delimo cvor

Važno je napomenuti da postoje mnogi parametri koji su suštinski povezani sa parametrom min_samples_split, tako da ih nećemo razmatrati ovde.

Na slici 4.1 vidimo rezultate na trening i test skupovima.



slika 4.1

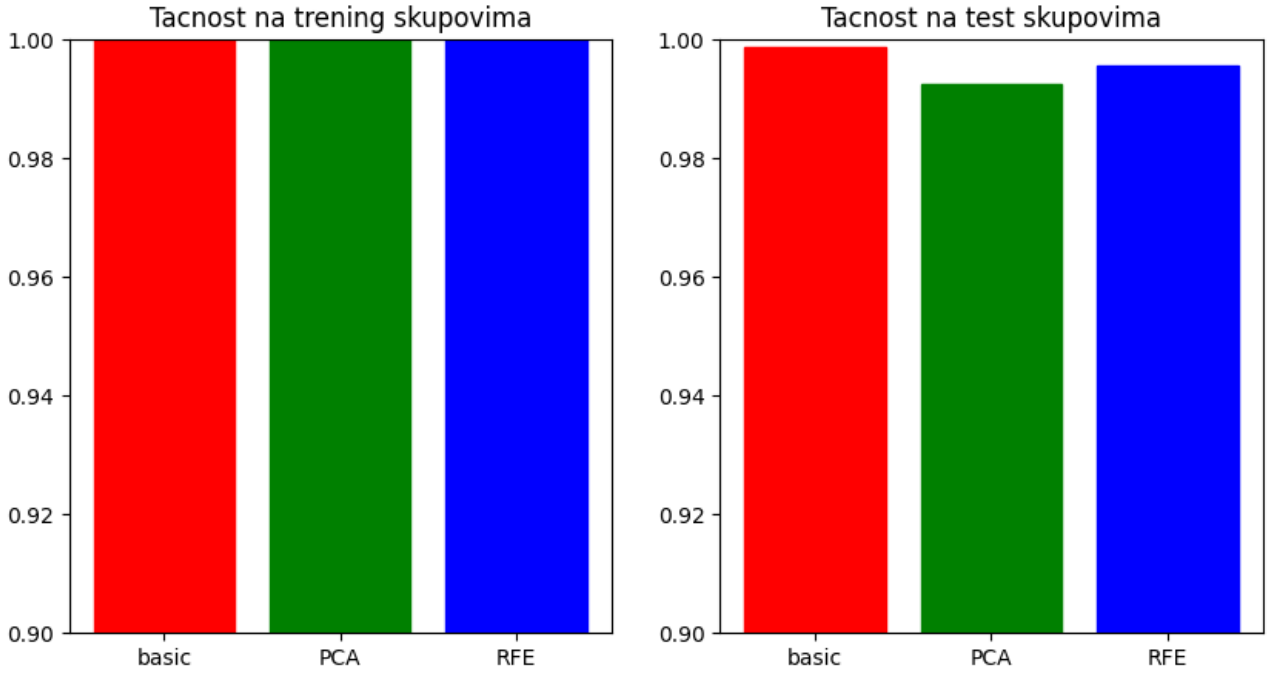
Zaključujemo da se najbolji rezultat postiže na RFE skupu - 99.24% tačnosti na test skupu, što je zanimljivo jer on ima najmanji broj atributa - samo 6. Malo slabiji rezultati se postižu na običnom skupu - 98.96%, a najslabiji (mada i dalje prilično dobri) na PCA skupu - 96.5%. Svi najbolji rezultati su za parametar criterion=entropy, dok je min_samples_split = 2 za PCA i RFE skupove, a 3 za običan skup.

4.2 Nasumična šuma(RFC)

Sledeći model koji ćemo koristiti je Nasumična šuma (en. Random Forest Classifier (RFC)). Ovo je vrlo zanimljiv model koji počiva na Condorcet teoremi žirija, koja kaže da je porota(žiri) pametnija od pojedinca (tj. ako imamo verovatnocu malo veću od 0.5 da je nešto tačno, onda će za veliki broj ljudi koji glasa, verovatnoća da je većina napravila korektan izbor težiti 1). Isto važi ako uzmemo sledeću analogiju:
1. žiri - drvo odlučivanja
2. glasovi - klasa koja se predvidja

Koristimo sledeću matricu parametara:
1. criterion=[gini, entropy] - parametar koji opisuje odredjivanje najboljeg atributa za podelu
2. n_estimators=[10, 50, 100] - parametar koji govori broj drveta odlučivanja koja ce se koristiti u glasanju

Na slici 4.2 vidimo rezultate na trening i test skupovima.



slika 4.2

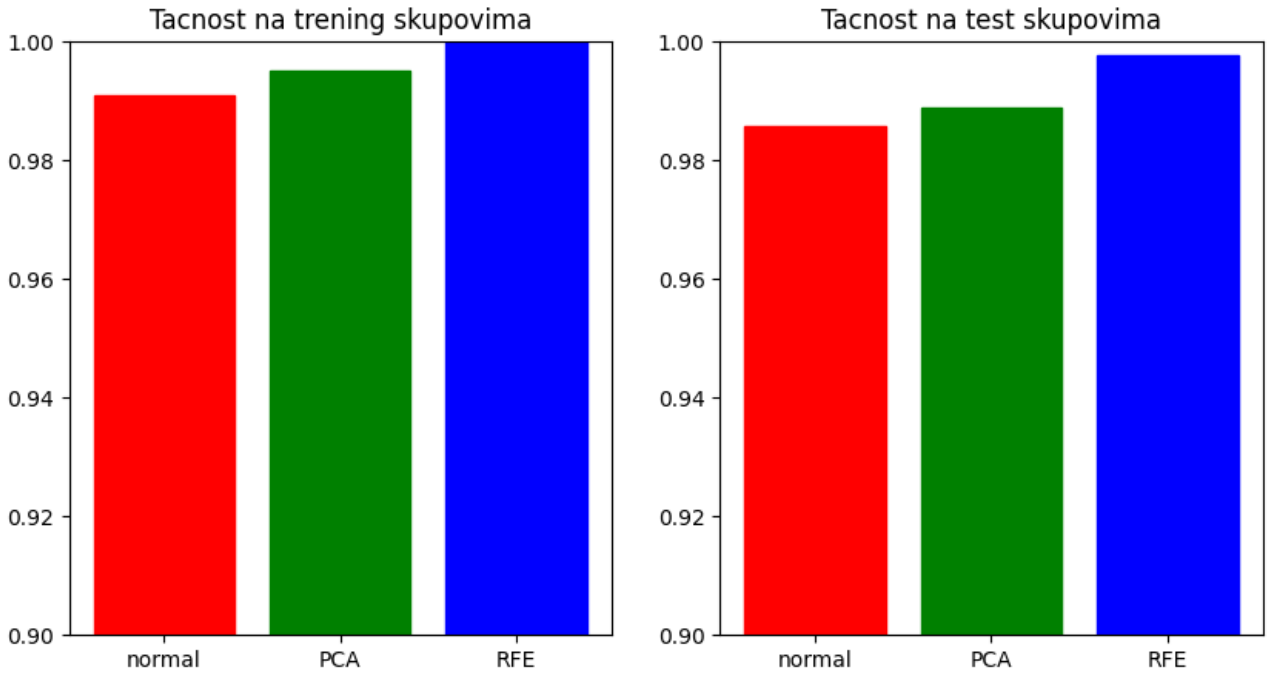
Zaključujemo da se najbolji rezultat postiže na običnom skupu - 99.88% tačnosti na test skupu. Malo slabiji rezultati se postižu na RFE skupu - 99.57%, i na PCA skupu - 99.26%. Ipak treba uzeti u obzir i vreme potrebno za treniranje modela - samo 126s za RFE skup, u poredjenju sa 308 za PCA skup, i 363 za običan skup. Svi najbolji rezultati su za parametre(criterion=entropy, n_estimators=100).

4.3 K najbližih suseda(KNN)

Sledeći model koji ćemo koristiti je K najbližih suseda (en. K Neighbours Classifer (KNN)). Ovaj model je zasnovan na rastojanjima (en. distance based) pa očekujemo da će dobro raditi nad našim normalizovanim numeričkim podacima.

U matrici menjamo samo jedan parametar:
1. n_neighbours=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10] - broj nabližih suseda koje će tražiti model

Na slici 4.3 vidimo rezultate na trening i test skupovima.



slika 4.3

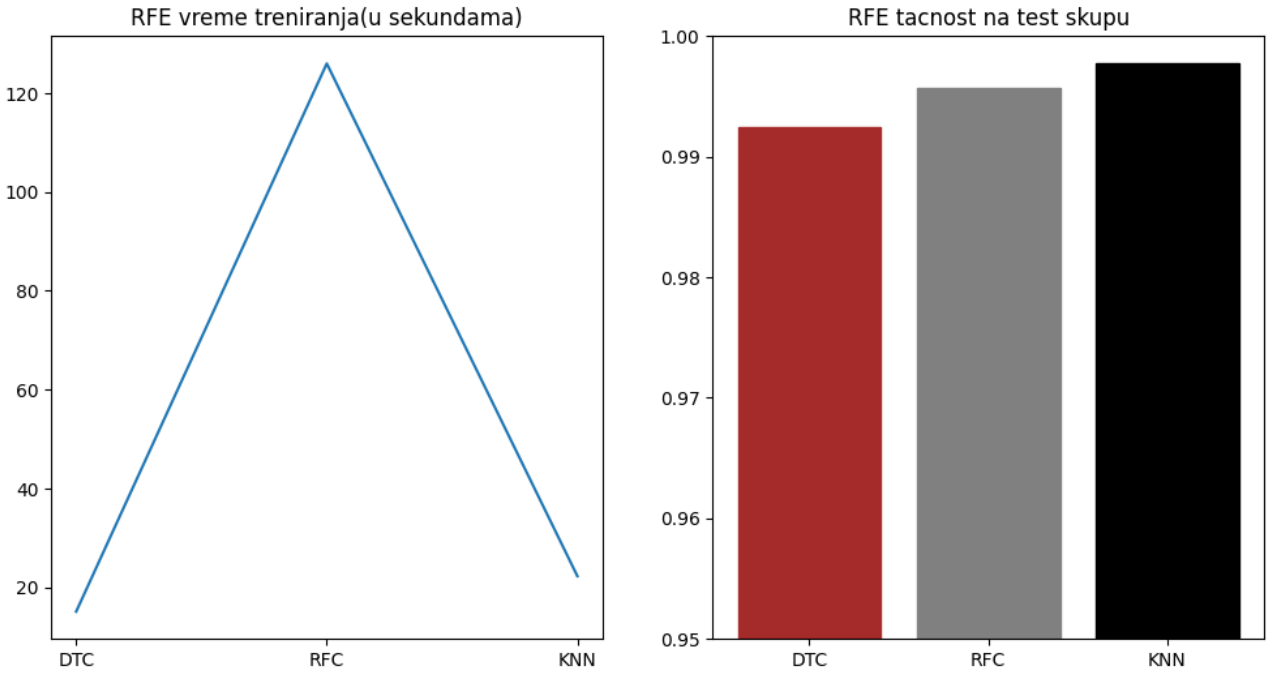
Zaključujemo da se najbolji rezultat postiže na RFE skupu - 99.77% tačnosti na test skupu, što je zanimljivo jer on ima najmanji broj atributa - samo 6. Malo slabiji rezultati se postižu na PCA skupu - 98.88%, i na običnom skupu - 98.59%. Najbolji rezultati su za prilično različit parametar n_neighbours (7 - obican, 3 - PCA, 1 - RFE).

4.4 Poredjenje rezultata klasifikacije

Iz prethodnih poglavlja vidimo da se najbolje pokazao RFE skup - daje najbolje rezultate za DTC i KNN modele, a za RFC daje samo malo lošiji model ali za znatno kraće vreme treniranja. Zbog toga ćemo uporedjivati modele nad RFE skupovima u nastavku.

Ostale mere preciznosti su veoma slične tačnosti, a matrica konfuzije ne otkriva puno informacija, tako da ih nećemo ovde razmatrati.

Ono što ćemo razmatrati su vremena treniranja i tašnosti na test skupu koje vidimo na slici 4.4.



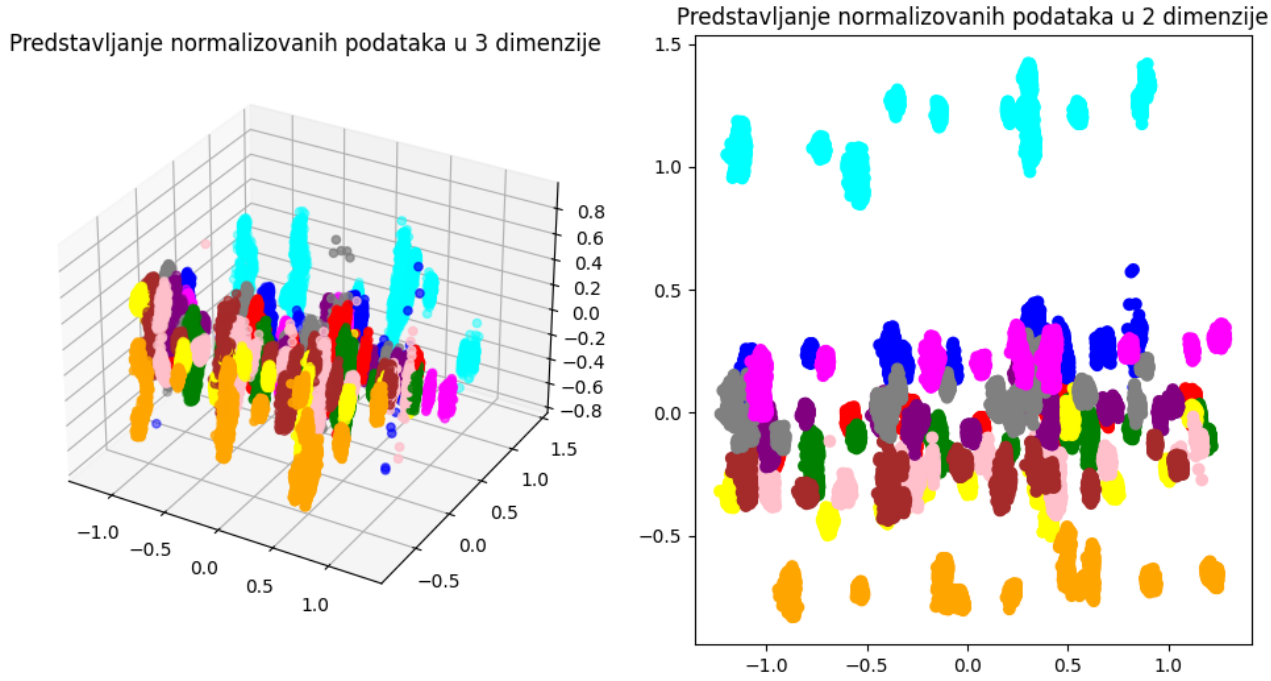
slika 4.4

Izvodimo sledece zakljucke:
1. DTC - iako najlošija, i dalje odlična tačnost uz najkraće vreme treniranja
2. RFC - srednji po tačnosti, uz ubedljivo najduže vreme treniranja
3. KNN - najtačniji, uz veoma dobro vreme treniranja (uz malu napomenu da za KNN treba zbog prirode algoritma nešto duže vremena da se izvrši od ostala 2)

5 Pretprocesiranje u klasterovanju

U ovom poglavlju ćemo primeniti PCA transformaciju nad normalizovanim skupom. Ovo radimo pošto oba modela (Kmeans i Agglomerative) koja ćemo primeniti u sledećem poglavlju rade sa distancama.

Na slici 5.1 prikazani su redom podaci nakon PCA transformacije normalizovanih podataka u 3 i 2 dimenzije.



slika 5.1

U okviru klasterovanja klase će poslužiti isključivo za uporedjivanje sa našim modelima klasterovanja. Sa slike 5.1 vidimo da će biti teško naći slične modele klasterovanja - vidimo klase koji imaju lep oblik ali su "rascepkane" po prostoru tako da je jasno da nijedan model klasterovanja nece moci da uoci takve pravilnosti. Ipak, pokusaćemo da najbolje što umemo ostvarimo klasterovanje.

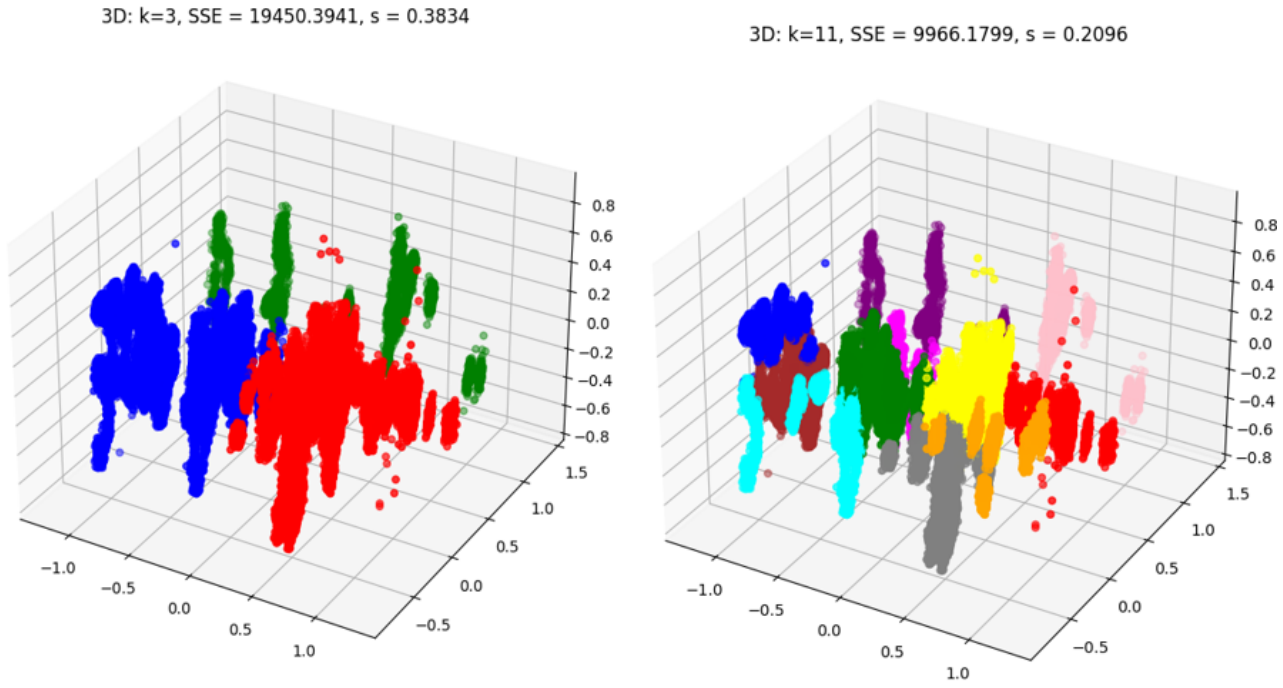
Skupove dobijene navedenim transformacijama koristićemo u poglavlju 6.

6 Klasterovanje

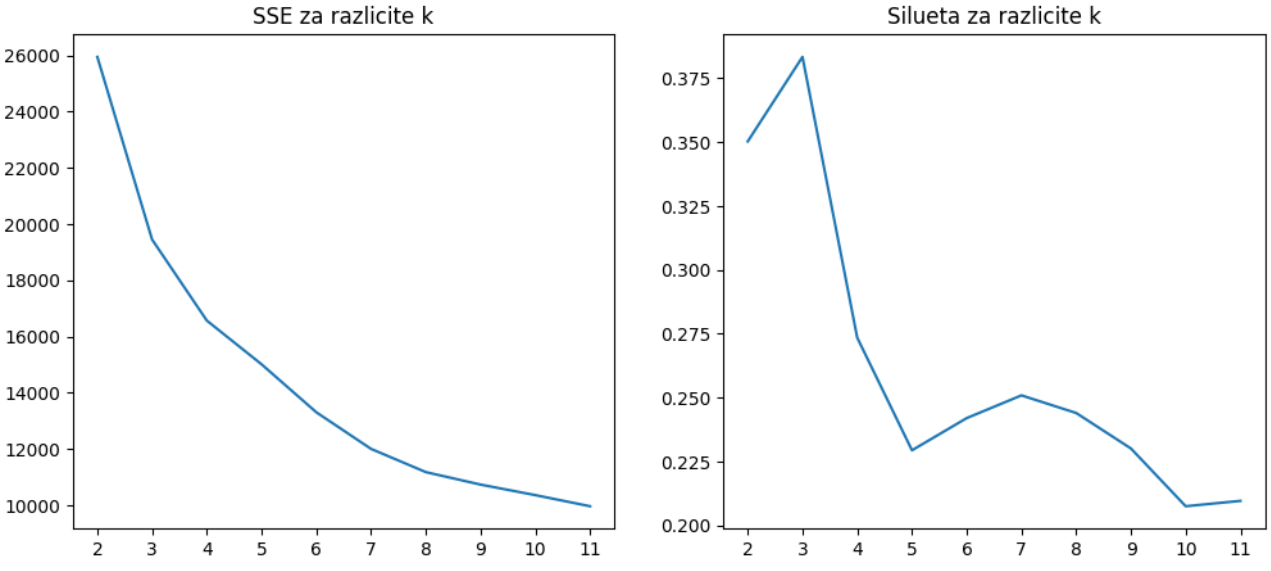
U okviru klasterovanja iskoristićemo podatke dobijene iz poglavlja 5. Radićemo sa sledećim modelima - K-sredina i Hijerarhisko klasterovanje(agglomerative).

6.1 K-sredina

K-sredina (en. k-means) je najpoznatiji model klasterovanja. Prvo što nameštamo je traženi broj klastera. Zatim nameštamo početnu konfiguraciju centroida (u nasim primerima radićemo samo sa kmeans++ inicijalizacijom - koja predstavlja napredniji nacin biranja centroida od obicnog slučajnog izbora). Model radi tako što u svakoj iteraciji pridružujemo instance najbližem centroidu cime formiramo klastere, a zatim za svaki od tih klastera ponovo odredjujemo centroid. Postupak se ponavlja do dostizanja željenog kriterijuma zaustavljanja (odredjeni broj iteracija, ili broj instanci koje su promenile klaster). Primenićemo dva načina rada:
1. primenjujemo model K-sredina i zatim PCA transformaciju nad podacima od tog modela. Na slici 6.1 prikazani su rezultati klasterovanja za 3 i 11 klastera (za ostale klastere videti svesku). Na slici 6.2 prikazani su SSE i silueta za svaki broj klastera izmedju 2 i 11.

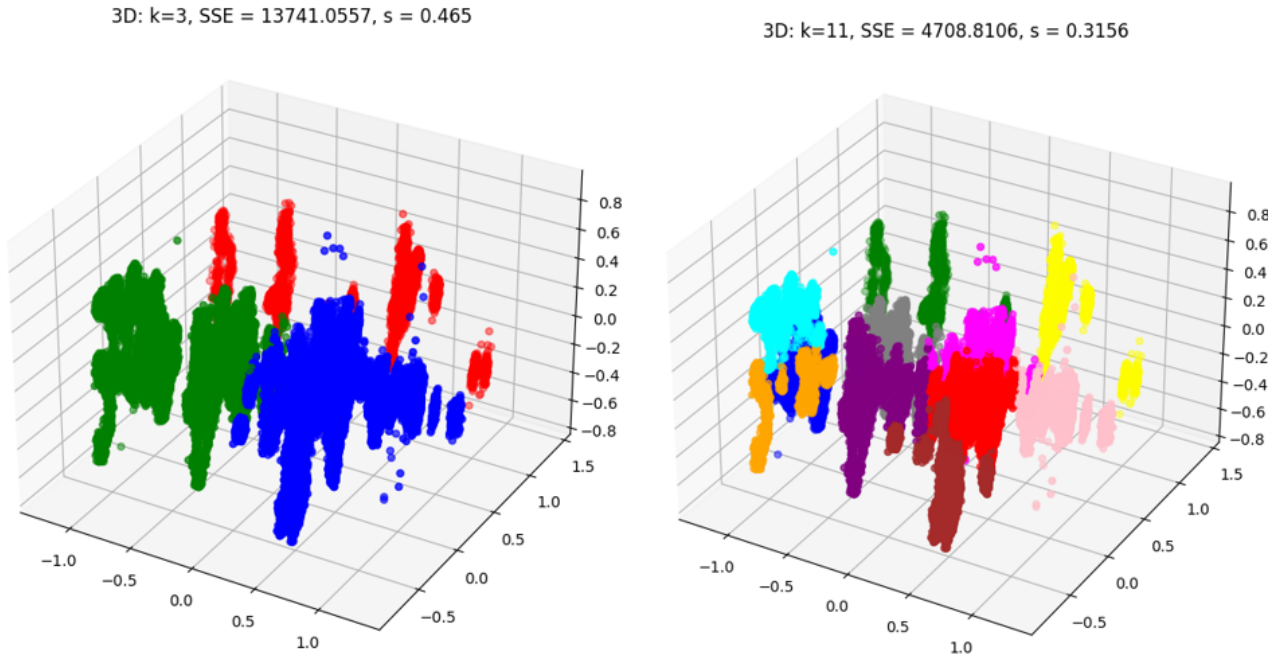


slika 6.1

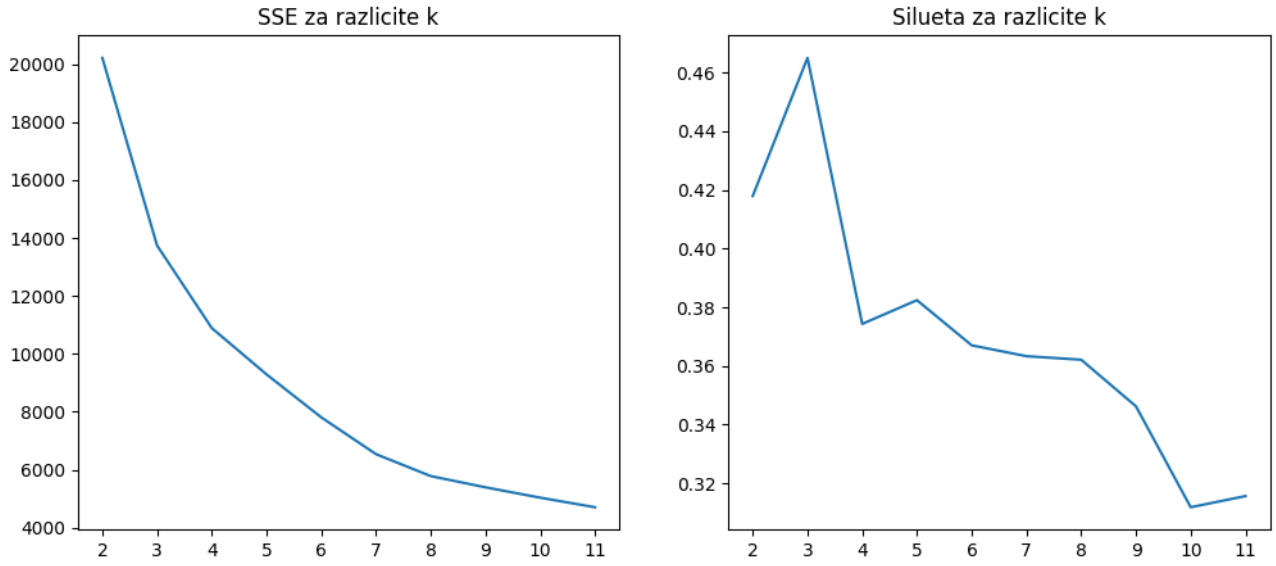


slika 6.2

2. primenjujemo PCA transformaciju i zatim model K-sredina. Na slici 6.3 prikazani su rezultati klasterovanja za 3 i 11 klastera (za ostale klastere videti svesku). Na slici 6.4 prikazani su SSE i silueta za svaki broj klastera izmedju 2 i 11.



slika 6.3



slika 6.4

Primećujemo sličan oblik funkcija SSE i siluete za oba pristupa. Medjutim, drugi pristup daje znatno bolje rezultate, zbog čega ćemo njega odabrati u ovom modelu. Takodje to nam je i motivacija da u sledećem modelu koristimo isključivo ovaj pristup.

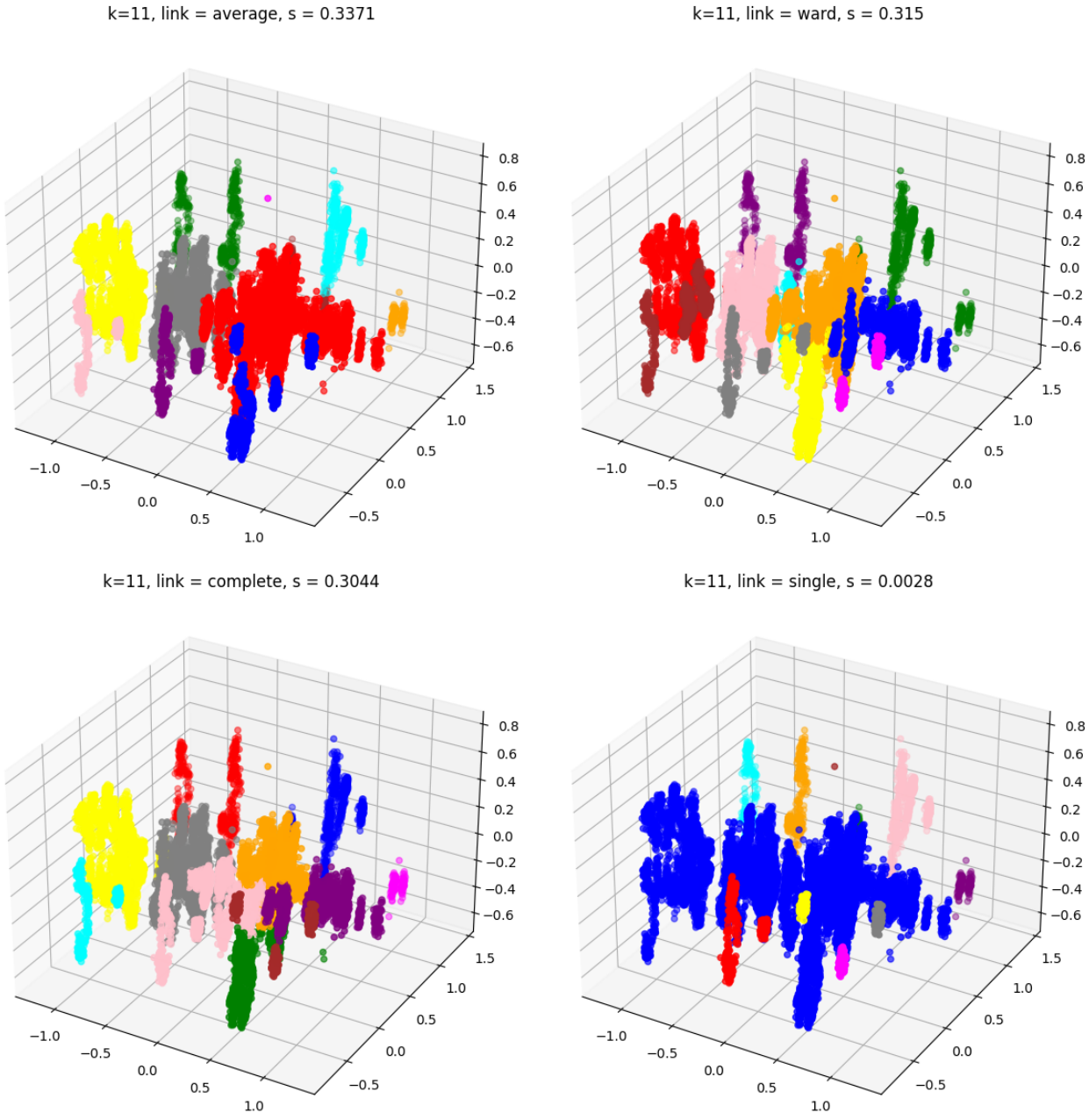
6.2 Hijerarhijsko klasterovanje(agglomerative)

Agglomerative hijerarhijsko klasterovanje je drugi model koji ćemo primeniti. Zasniva se na tome da na početku sve instance posmatra kao klastere, i zatim u svakoj iteraciji odredjuje najbliže rastojanje izmedju dva klastera i njih spaja u jedan. U ovom algoritmu takodje navodimo željeni broj klastera. U daljem radu sa agglomerativom fokusiraćemo se na slučaj od 11 klastera. Mi ćemo ovde pokazati kako algoritam radi za različite vrste rastojanja:

1. single - rastojanje predstavlja rastojanje izmedju 2 najbliže tačke iz svakog od klastera
2. average - rastojanje predstavlja prosečno rastojanje izmedju svake 2 tacke iz svakog od klastera
3. complete - rastojanje predstavlja rastojanje izmedju 2 najdalje tačke iz svakog od klastera

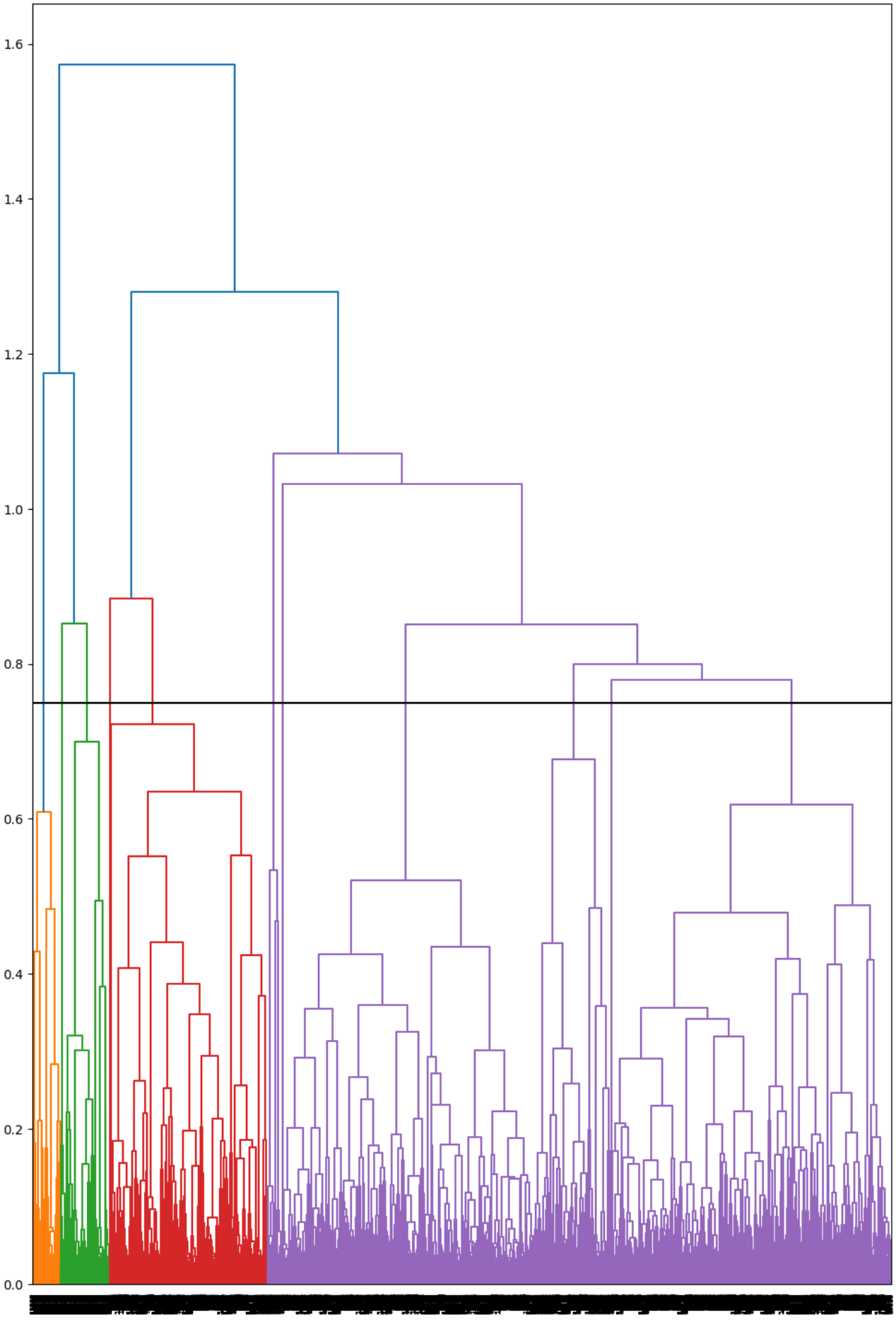
4. ward - rastojanje predstavlja potencijalnu kvadratnu grešku novodobijenog klastera

Na slici 6.5 prikazan je rad modela za 11 klastera za svako od ovih rastojanja.



slika 6.5

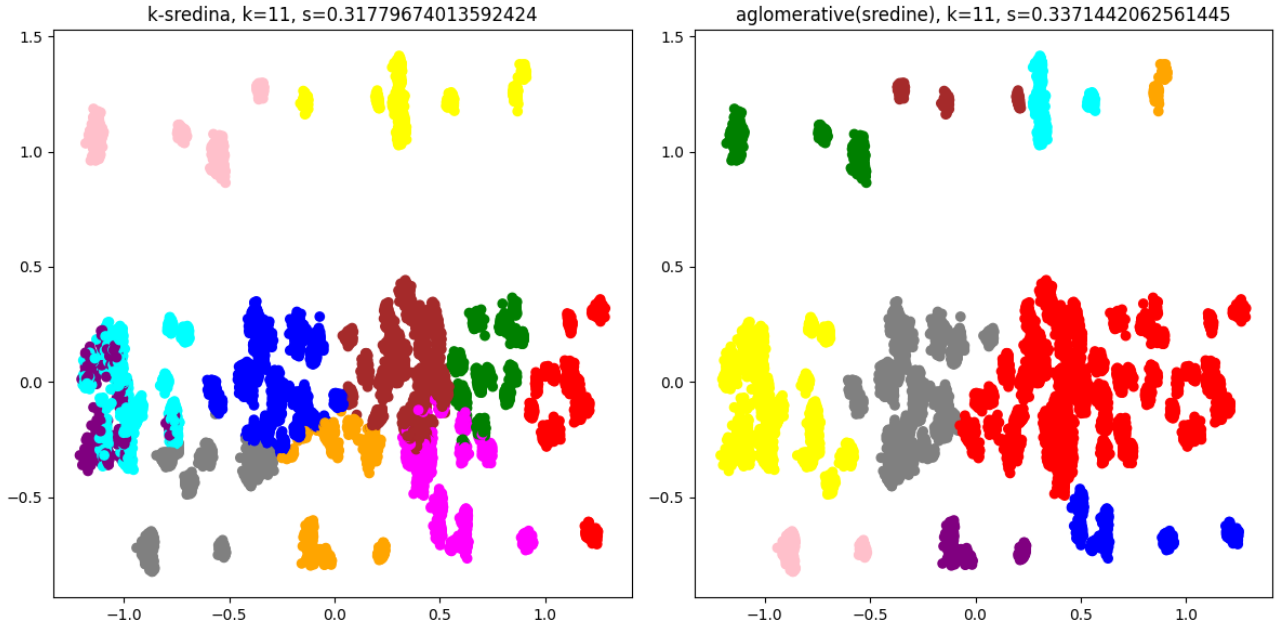
Primenom analognog modela iz scipy biblioteke (sa average parametrom za distancu) dobijamo stablo (dendogram) koje će na kraju doći do korena - samo jedan klaster koji obuhvata sve instance. Na slici 6.6 vidimo dendogram koji smo presekli horizontalnom linijom y = 0.75 što znači da za tu distancu izmedju sledeća dva najbliza klastera, imamo tacno 11 klastera.



slika 6.6

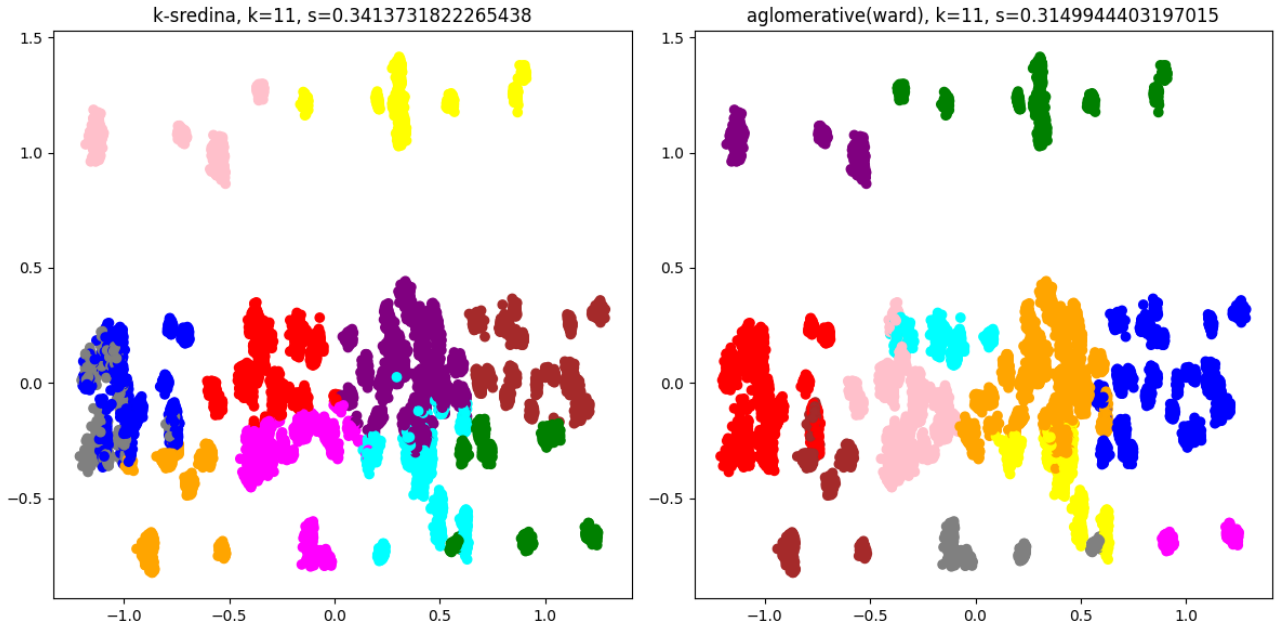
6.3 Poredjenje modela klasterovanja

Nad dobijenim modelima poredjenja ćemo vršiti u 2 dimenzije - čisto da bismo mogli da utvrdimo neke karakteristike rada naših algoritama.
Na slici 6.7 prikazano je poredjenje izmedju k-sredina i agglomerative (average distanca) modela. Na njoj se jasno vidi globularnost klastera kod k-sredina modela, kao i dobijanje za nijansu "izduženijih" klastera (dobijenih spajanjem više globula) kod agglomerative modela.



slika 6.7

Na slici 6.8 prikazano je poredjenje izmedju k-sredina i agglomerative (ward distanca) modela. Na njoj se jasno vidi zanimljivo svojstvo ward distance - da se ponaša slično kao k-means algoritam.



slika 6.8

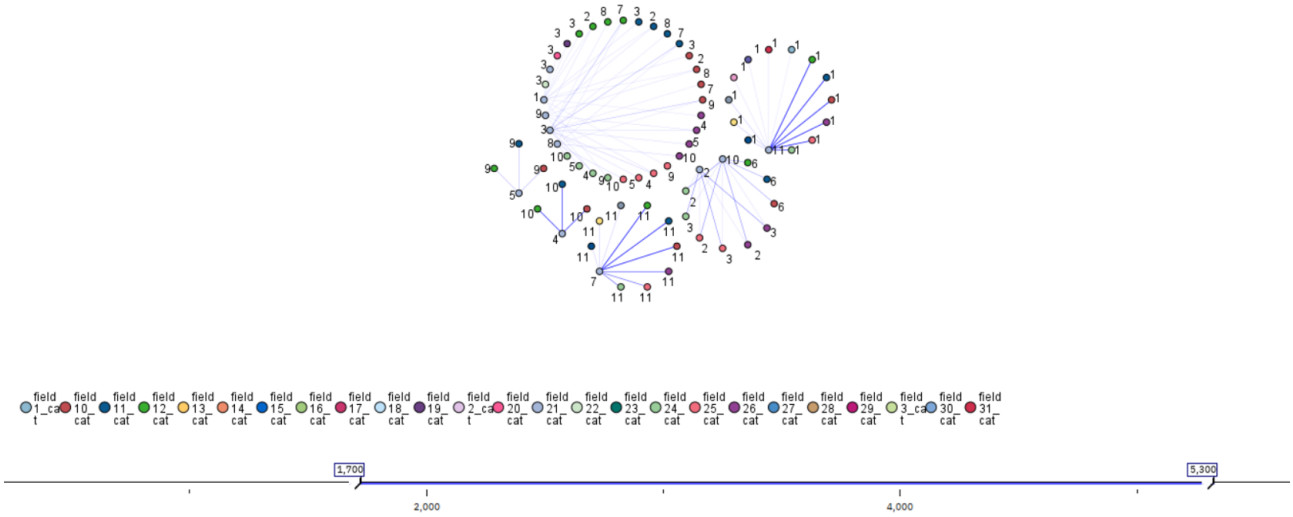
7 Pravila pridruzivanja

Pravila pridruživanja predstavljaju neka zanimljiva pravila oblika: telo - glava, koja govore da ako se u transakciji pojave stavke iz tela, verovatno će se pojaviti i stavke iz glave.

U našem slučaju pravila će biti sledećeg oblika: ako imamo neke vrednosti odredjenih atributa za neku instancu, iz toga zakljucujemo da atribut pripada odredjenoj klasi

Uslov za primenu našeg modela je izvršenje diskretizacije atributa koje postižemo pomoću čvora Bining. Svaki atribut će imati vrednosti od 1 do 11 podeljene tako da imamo jednako instanci koje imaju te vrednosti za svaki od atributa.

Pre same primene modela pokrenućemo usmerenu mrezu (en Directed Web) koja će nam pomoći u razumevanju potencijalnih pravila pridruživanja. Njen rad prikazan je na slici 7.1.



slika 7.1

Zaključujemo da ce u skoro svim pravilima pridruživanja učestvovati atributi field10_cat, field11_cat i field12_cat tako da ćemo naš model Association Rules primeniti prvo nad njima. Pri tome za algoritam u modelu uzimamo Apriori koji se zasniva na sledećem principu:
1. ako je skup čest, onda su i svi njegovi podkupovi cesti
2. ako je skup redak, onda su i svi njegovi nadskupovi retki

Na slici 7.2 prikazano je 10 pravila sa najboljom pouzdanoscu (primetimo da bi potpuno isto sortiranje bilo i po liftu).

Most Interesting Rules by Confidence

					Other Evaluation Statistics		
Rank	Rule ID	Condition	Prediction	Sorted By Confidence(%)	Condition Support (%)	Rule Support (%)	Lift	Deployability (%)
1	1	field12_cat = 1	field49 = 11	99,96	9,09	9,09	11,00	0,00
2	2	field10_cat = 1 field12_cat = 1	field49 = 11	99,96	9,09	9,09	11,00	0,00
3	3	field11_cat = 1 field12_cat = 1	field49 = 11	99,96	9,09	9,09	11,00	0,00
4	4	field10_cat = 1 field11_cat = 1 field12_cat = 1	field49 = 11	99,96	9,09	9,09	11,00	0,00
5	5	field10_cat = 1	field49 = 11	99,94	9,09	9,09	10,99	0,01
6	6	field11_cat = 1	field49 = 11	99,94	9,09	9,09	10,99	0,01
7	7	field10_cat = 1 field11_cat = 1	field49 = 11	99,94	9,09	9,09	10,99	0,01
8	8	field12_cat = 11	field49 = 7	99,15	9,09	9,01	10,91	0,08
9	9	field10_cat = 11 field12_cat = 11	field49 = 7	99,15	9,08	9,00	10,91	0,08
10	10	field11_cat = 11 field12_cat = 11	field49 = 7	99,15	9,08	9,00	10,91	0,08

slika 7.2

Na kraju prikažimo Association Rules model sa Apriori algoritmom i na celokupnom skupu. Na slici 7.3 prikazano je 10 pravila sa najboljom pouzdanošću (primetimo da bi potpuno isto sortiranje bilo i po liftu).

Most Interesting Rules by Confidence

						Other Evaluation Statistics		
Rank	Rule ID	Condition	Prediction	Sorted By Confidence(%)	Condition Support (%)	Rule Support (%)	Lift	Deployability (%)
1	1	field7_cat = 11 field12_cat = 11	field49 = 7	100,00	6,75	6,75	11,00	0,00
2	2	field8_cat = 11 field12_cat = 11	field49 = 7	100,00	6,75	6,75	11,00	0,00
3	3	field7_cat = 11 field10_cat = 11	field49 = 7	100,00	6,75	6,75	11,00	0,00
4	4	field7_cat = 11 field11_cat = 11	field49 = 7	100,00	6,75	6,75	11,00	0,00
5	5	field8_cat = 11 field10_cat = 11	field49 = 7	100,00	6,75	6,75	11,00	0,00
6	6	field8_cat = 11 field11_cat = 11	field49 = 7	100,00	6,75	6,75	11,00	0,00
7	7	field7_cat = 11 field10_cat = 11 field11_cat = 11	field49 = 7	100,00	6,75	6,75	11,00	0,00
8	8	field8_cat = 11 field10_cat = 11 field11_cat = 11	field49 = 7	100,00	6,75	6,75	11,00	0,00
9	9	field9_cat = 11 field12_cat = 11	field49 = 7	100,00	6,75	6,75	11,00	0,00
10	10	field9_cat = 11 field10_cat = 11	field49 = 7	100,00	6,74	6,74	11,00	0,00

slika 7.3

Napomenimo još da pokusaj formiranja Association Rules modela za FPGrowth algoritam nije uspeo niti za jedne parametre, pa ga nećemo ovde dalje obradjivati.

8 Zaključak

Naš skup podataka imao je puno instanci, zbog čega su se neki modeli izvršavali sporije ili uopšte nisu ni mogli da se izvrše. Sa druge strane nepostojanje nedostajućih vrednosti, mali broj ekstremnih vrednosti i balansiranost klasa olakšali su u dobroj meri rad sa podacima.

Modeli klasifikacije ostvaruju veoma dobre rezultate na našem skupu, iz gore pomenutih razloga, i iz razloga što imamo sve atribute koje izražavaju istu meru.

Modeli klasterovanja ipak ne mogu da postignu dobre rezultate (ako želimo da dobijemo iste klase kao u klasifikaciji) jer su klase (iako relativno pravilnih oblika) razbacane po prostoru pa ih je moguće spojiti samo nekom od metoda nadgledanog ucenja.

Model pravila pridruzivanja uspeo je da pronadje odredjena pravila sa poprilicno vekikom pouzdanošću i liftom, tako da iako nije mogao da se izvrši za neke modele, zadovoljni smo rezultatima.

9 Literatura

1. https://www.sciencedirect.com/science/article/pii/S1877050920320706