**Projekat iz Istrazivanja podataka 1**

profesor:Nenad Mitic

astitent:Stefan Kapunac

student: Jelena Mitrovic 357/2020

**Sadrzaj**

[**1.Uvod 3**](#_9ejw2fhcalxn)

[**2.Analiza atibuta 3**](#_bzpk6ye7obge)

[**3.Klasifikacija 7**](#_dgq5oio9srfh)

[3.1. Stabla odlucivanja 7](#_i6xgwpptxui2)

[3.1.2. Slucajna suma 8](#_2dghmjw1shi1)

[3.1.3. Poredjenje modela 9](#_4h12o54uwebm)

[3.2.1 KNN 10](#_our4pvjce48k)

[3.2.2.BaggingClassifier 10](#_tar56g31aa4i)

[3.2.3 Poredjenje modela 11](#_uny5mw32njwk)

[3.2.4 PCA - KNN 11](#_8v02xzj5ci44)

[3.3.1. SVM 13](#_e4ugp4llh9v8)

[**4.Klasterovanje 14**](#_f5lkqxs0h4l4)

[4.1. KMeans 14](#_y6uoslg5zx3g)

[4.2. Agglomerative 15](#_dqfw8jxtrvq1)

[4.2.DBSCAN 15](#_tlx5xz2edceb)

[**5.Pravila pridruzivanja 15**](#_d7d5l42r6t9y)

[5.1.Apriori 16](#_s3i1ed843a30)

[**6. Literatura 18**](#_410dqdvv5dzn)

# 1.Uvod

Podaci koji se nalaze u bazi podataka sa kojom radim su prikupljeni od učesnika u eksperimentalnim događajima brzih sastanaka od 2002.-2004. godine. Prisutne osobe bi imale četvorominutni "prvi sastanak" sa svakim drugim učesnikom suprotnog pola. Na kraju njihova četiri minuta, učesnici su upitani da li bi želeli da ponovo izadju na sastanak . Od njih je takođe zatraženo da ocenjuju svoj sastanak na osnovu šest atributa: privlačnost, iskrenost, inteligencija, zabava, ambicija i zajednički interesi. Skup podataka takođe uključuje podatke iz upitnika prikupljene od učesnika u različitim tačkama procesa. Ova polja uključuju: demografiju, navike za sastanke, uverenja o tome šta drugi smatraju dragocenim kod partnera i informacije o životnom stilu.

# 2.Analiza atibuta

Baza podataka se sastoji od 8378 instanci i 123 atributa.

Atributi baze podatka:

gender: Pol osobe attractive: Ocena osobe za samu sebe - atraktivnost

age: Broj godina osobe sincere: Ocena osobe za samu sebe - iskrenost

age\_o:Broj godina partnera intelligence: Ocena osobe za samu sebe -inteligencija

d\_age: Razlika u godinama funny:Ocena osobe za samu sebe - duhovitost

race: Rasa osobe ambition: Ocena osobe za samu sebe- ambicija

race\_o:Rasa partnera attractive\_partner:Ocena osobe za partnera-atraktivnost

samerace: Da li je vazno da osobe imaju istu rasu(1/0) sincere\_partner:Ocena osobe za partnera-iskrenost

importance\_same\_race: Koliko je vazno da je partner iste rase intelligence\_partner: Ocena osobe za partnera-inteligencija

importance\_same\_religion:Koliko je vazno da je partner iste religije funny\_partner: Ocena osobe za partnera-duhovitost

field: Obrazovanje osobe ambition\_partner: Ocena osobe za patnera-ambicija

pref\_o\_attractive:Koliko je partneru vazna privlacnost shared\_interests\_partner: Ocena osobe za partnera-zaj.inter.

pref\_o\_sinsere: Koliko je partneru vazna iskrenost sports: Interesovanje osobe za sport

pref\_o\_intelligence: Koliko je partneru vazna inteligencija tv sports: Interesovanje osobe za tv sport

pref\_o\_funny: Koliko je partneru vazno da je osoba duhovita exercise: Interesovanje osobe za vezbanje

pref\_o\_ambitious:Koliko je partneru vazna ambicija dining: Interesovanje osobe za izlazak na veceru

pref\_o\_shared\_interests: Koliko je partneru vazno da imaju zaj. inter. museum: Interesovanje osobe za obilazak muzeja

attractive\_o:Ocena osobe od strane partnera posle sastanka o atraktivnosti art:Interesovanje osobe za umetnost

sincere\_o: Ocena osobe od strane partnera posle sastanka o iskrenosti hiking:Interesovanje osobe za planinarenje

intelligence\_o:Ocena osobe od strane partnera posle sastanka o inteligenciji gaming: Interesovanje osobe za igranje igrica

funny\_o:Ocena osobe od strane partnera posle sastanka o duhovitosti clubbing: Interesovanje osobe za nocne izlaske

ambitous\_o: Ocena osobe od strane partnera posle sastanka o ambicioznost reading:Interesovanje osobe za citanje

shared\_interests\_o:Ocena osobe od strane partnera posle sastanka o zaj.inter. tv:Interesovanje osobe za gledanje televizije

attractive\_important:Kakvog partnera trazite - atraktivnost theater: Interesovanje osobe za odlazak u pozoriste

sincere\_important: Kakvog partnera trazite - iskrenost movies: Interesovanje osobe za gledanje filmova

intellicence\_important: Kakvog partnera trazite - inteligencija concerts: Interesovanje osobe za koncerte

funny\_important: Kakvog partnera trazite - duhovitost music: Interesovanje osobe za muziku

ambtition\_important: Kakvog partnera trazite - ambicija shopping: Interesovanje osobe za shopping

shared\_interests\_important:Kakvog partnera trazite -zaj.inter. yoga: Interesovanje osobe za jogu

interests\_correlate: Korelacija interesovanja osobe i partnera

expected\_happy\_with\_sd\_people: Koliko ocekujete da cete biti srecni sa osobama koje upoznate na brzom sastanku

expected\_num\_interested\_in\_me: Od 20 osoba koje cete upoznati, koliko ocekujete da ce biti zainteresovano za zabavljanje sa vama

expected\_num\_matches: Koliko ocekujete pozitivnih odgovora za sledeci sastanak

like: Da li vam se svidja partner

guess\_prob\_liked: Kolika je verovatnoca da se dopadnete vasem partneru

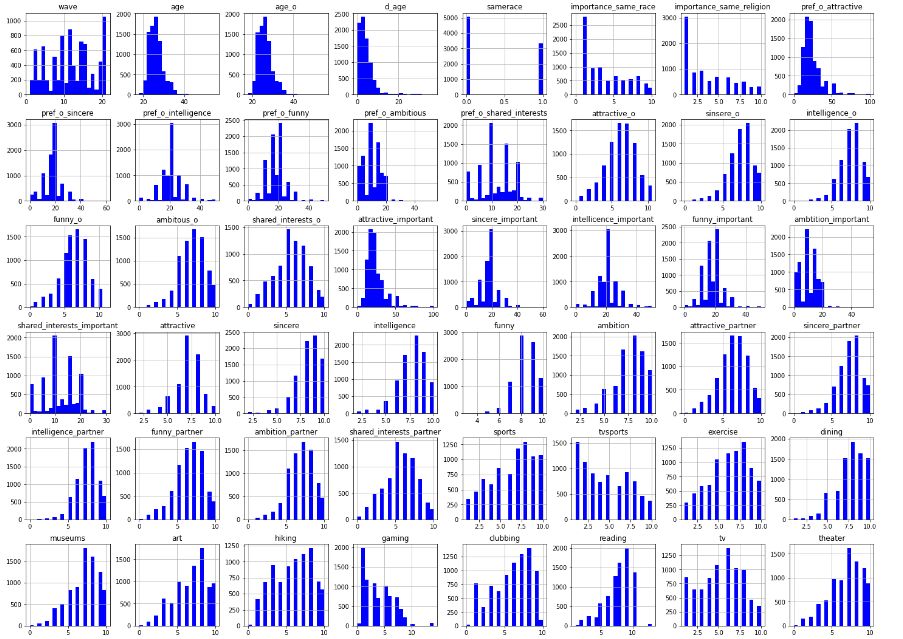
met: Da li ste sreli partera pre ovog sastanka

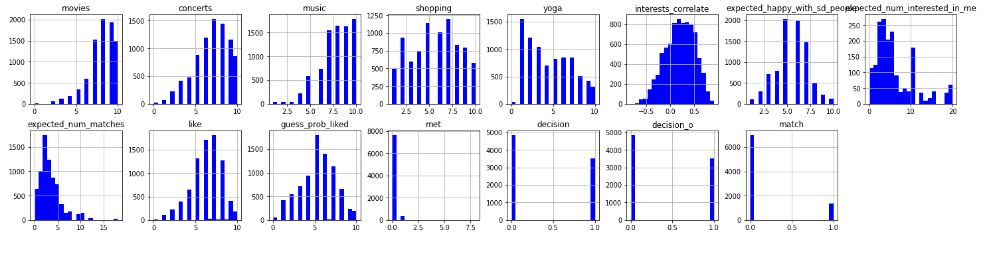
decision: Odluka osobe posle sastanka

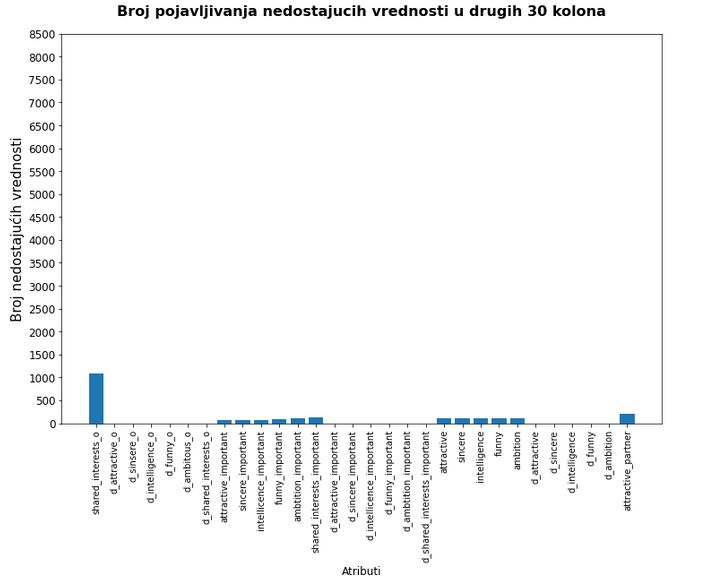
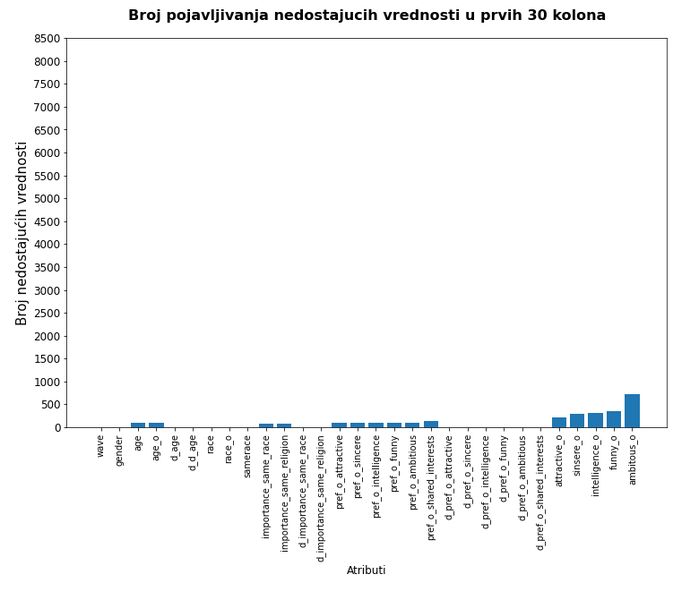
decision\_o: Odluka partnera posle sastanka

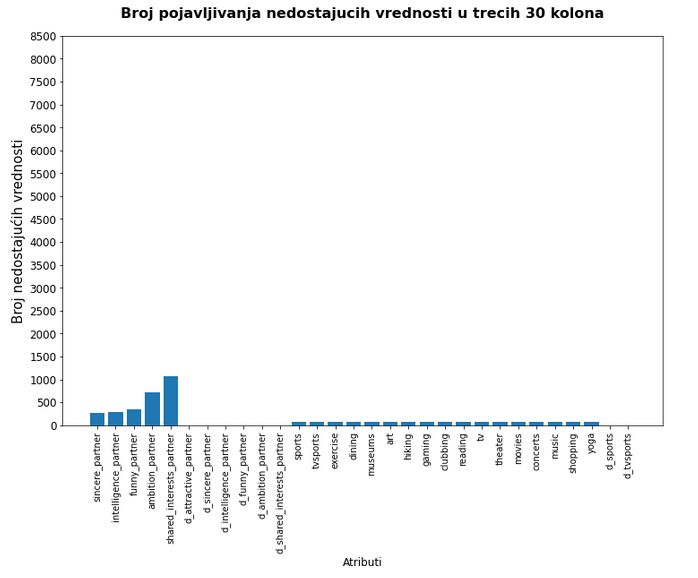
match: Da li ce se osobe opet sresti

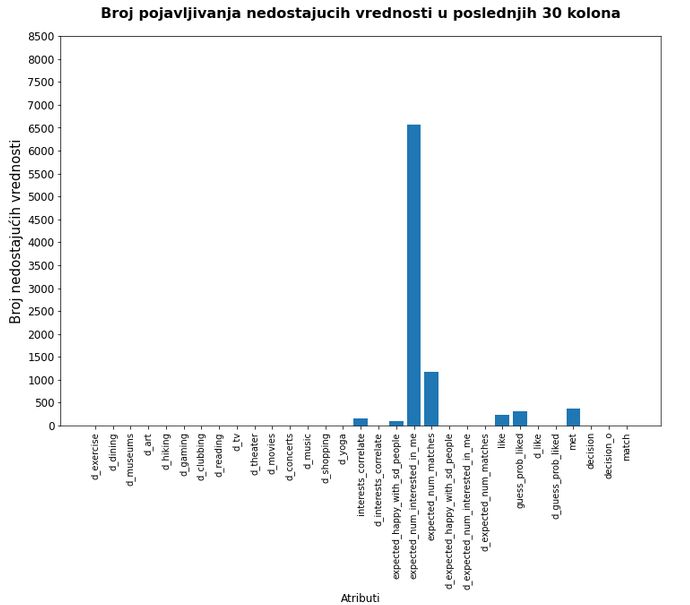
Na slikama ispod prikazan je histogram raspodele podatka nekih od atributa





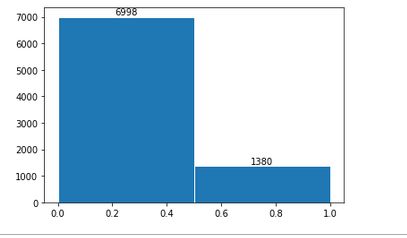
Baza podatka sa kojom radim, ima veliki broj nedostajucih vrednosti.Procenat instanci koje sadrze nedostajuce vrednosti je priblizno 87.5 %. Graficki prikaz nedostajucih vrednosti:





U koraku pretprocesiranja sam sve nedostajuce vrednosti zamenila medijanom za tu kolonu, a sve kategoricke (imenske) atribute sam prevela u numericke koriscenjem LabelEncoder i na taj nacin sam pripremila podatke za klasifikaciju.

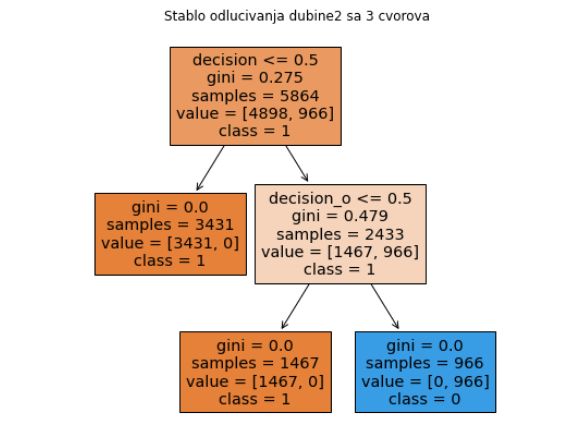
# 3.Klasifikacija

Zadatak klasifikacije u slucaju mog skupa podataka, jeste predvidjanje ciljne promenljive (‘match’), na osnovu ostalih atributa. Dakle, zadatak klasifikacije je predvidjanje da li ce se osobe ponovo vidjeti. Ciljna promenljiva ‘match’ ima 2 vrednosti (0 i 1), pa zakljucujem da se radi o binarnoj klasifikaciji. Na slici ispod je prikazana balansiranost izmedju klasa. Na osnovu slike zakljucujemo da postoji nebalansiranost izmedju klasa, medjutim, mislim da i u jednoj i u drugoj klasi ima dovoljno instanci, pa stoga necu koristiti neke od metoda za smanjenje razlike u dimenziji klasa, medjutim, zbog ovoga necu koristiti tacnost kao kriterijum provere modela, nego f1-meru.

Obradjeni algoritmi klasifikacije: Stabla odlucivanja,Slucajna suma,K najblizih suseda,Bagging Classifier i SVM.

## 3.1. Stabla odlucivanja

Primenom DecisionTreeClassifier() sa podrazumevanim parametrima, dobila sam sledece stablo odlucivanja:



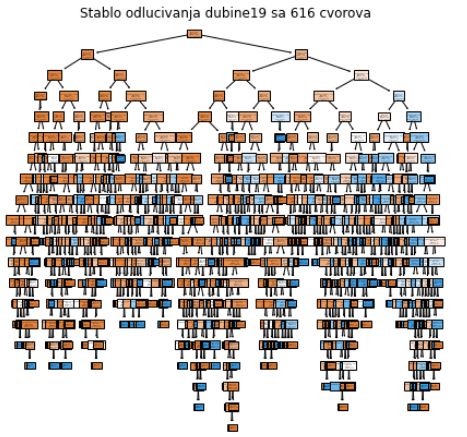
Na slici primecujemo, da se prlikom formiranja stabla koriste samo 2 atributa : ‘decision’ i ‘decision\_o’. Uticaji ovih atributa na formiranje stabla su redom: 0.27812303 i 0.72187697, a uticaj svih ostalih jednak je nula. Koriscenjem ovog modela dobijen je f1-score od 1 i na trening i na test skupu, a odgovarajuce matrice konfuzije za trening i test skup redom su:

1 0 1 0

1 4898 0 1 2100 0

0 0 966 0 0 414

Posto sam videla da na formiranje stabla uticu samo 2 atributa, htela sam da vidim kako bi izgledalo stablo odlucivanja kada bih izbacila te atribute.. Dobila sam stablo dubine 19 sa 616 cvorova.



Ovaj model ne daje dobre rezultete, zato sto se previse prilagodio trening podacima i izgubio moc generalizacije. Matrice konfuzije na trening i test skupu:

1 0 1 0

1 5248 0 1 1506 244

0 0 1035 0 201 143

Model je komplikovaniji nego sto bi trebalo da bude.

## 3.1.2. Slucajna suma

PozivanjemGridSearchCV() za model RandomForestClassifer, dobila sam da su optimalni parametri ovog modela 'max\_depth' =15, 'min\_samples\_split'= 2, 'n\_estimators': 300. Proverom matrice konfuzije, vidimo da model ne pravi greske na trening skupu, ali pravi 10 gresaka na test skupu.

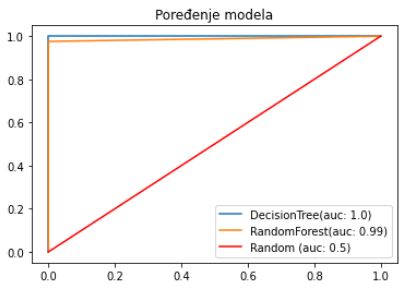
1 0 1 0

1 4898 0 1 2100 0

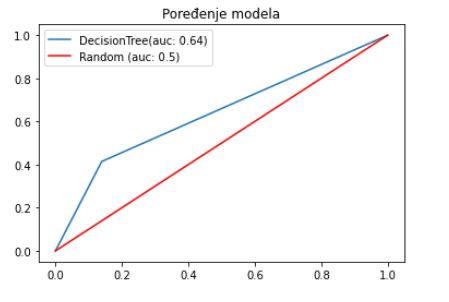
0 0 966 0 10 404

## 3.1.3. Poredjenje modela

Modele poredimo koristeci ROC krivu. Sa slike ispod primecujemo da su navedeni algoritmi dali jako dobre rezultate,zato sto je povrsina ispod krive blizu jedinice.



Poredim model koji ne sadrzi dva glava atributa sa random modelom. Ovaj model je malo bolji od random modela, ali je losiji u odnosu na gore navedene.



## 3.2.1 KNN

Pre primene KNN algoritma moramo da normalizujemo podatke. Za normalizaciju sam korstila StandardScaler. Na normalizovane podatke sam primenila KNeighborsClassifier() sa podrazumevanim parametrima. Model pravi jako veliki broj gresaka gde instance klase 1, klasifikuje kao instance klase 0. Odgovarajuce matrice konfuzije na trening i test skupu su:

1 0 1 0

1 4780 118 1 2024 76

0 516 450 0 307 107

Model je znacajno losiji u odnosu na stabla odlucivanja, pa sam htela da proverim da li postoje elementi van granica koji mozda kvare tacnost ovog modela. Za proveru elemenata van granica koristila sam IQR algoritam i videla da postoji veliki broj elemena koji se tretiraju kao outlieri(u jednoj koloni je procenat takvih elemenata je cak 49%). Zbog velikog broja outiliera nisam mogla da obrisem te podatke, zato sto bih izgubila mnogo podataka, pa sam odlucila da outliere zamenim medijanom i na taj nacin smanjim njihov broj. Ovim postupkom nisam u potpunosti izgubila elemente van granica, ali se procenat istih znacajno smanjio. Primenom KNeighborsClassifier() sa podrazumevanim parametrima nad ovim podacima dobila sam matrice konfuzije koje su skoro identicne. Zatim, probala sam da podesim hiper parametre. Zbog velike nebalansiranosti, za score za koristila f1 meru. Primenom GridSearchCV algoritma, dobila sam da su optimalni parametri KNeighborsClassifier() algoritma: 'n\_neighbors': 10, 'p': 2, 'weights': 'distance', a najveca vrednost f1 scora je piblizno 0.33.Podesavanjem hiper parametara,dobili smo model koji se preprilagodio trening podacima.

1 0 1 0

1 4898 0 1 2044 56

0 0 966 0 331 83

## 3.2.2.BaggingClassifier

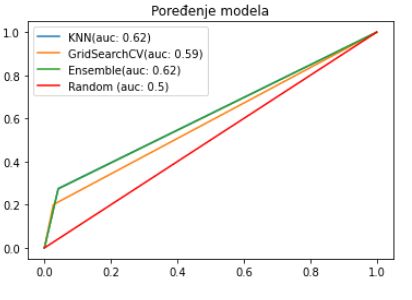
Primenom ove ansambl metode dobila sam priblizno iste rezultate kao i primenom KNN algoritma.Matrice konfuzije na trening i test skupu:

1 0 1 0

1 4812 86 1 2014 86

0 537 429 0 301 113

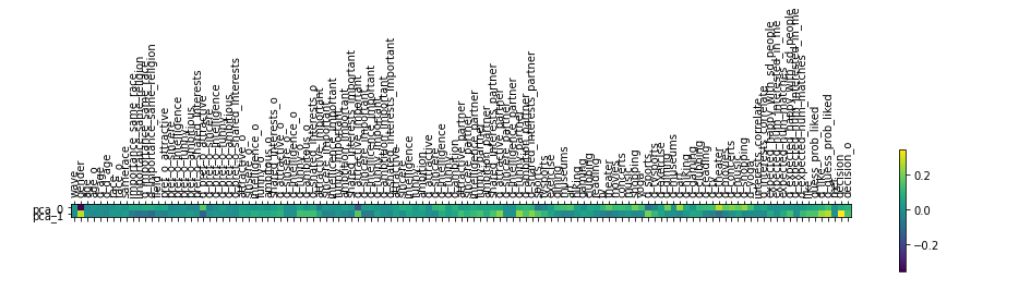
## 3.2.3 Poredjenje modela

****

Modeli su znatno losiji u odnosu na stabla odlucivanja. Zbog velike razlike izmedju broja instaci u klasama, mislim da K najblizih suseda nije najbolji izbor za klasifikaciju mog skupa podataka. Upravo zbog toga sto imamo mnogo manje istanci klase 1, vecina suseda ce pripadati klasi 0 i zbog toga model jako veliki broj FN gresaka.

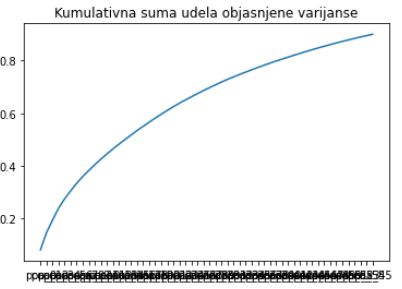
## 3.2.4 PCA - KNN

Moja baza podataka se sastoji od 121. atributa, sto je poprilicno puno. Htela sam da proverim da li primenom PCA algoritma mogu da smanjim broj atributa. Prvo sam probala da broj atributa smanjim na dva. Pokusala sam da vizuelno prikazem koliko je koji atribut uticao na pravljenje novih atributa, medjutim zbog velikog broja atributa vizualizacija najjasnija.

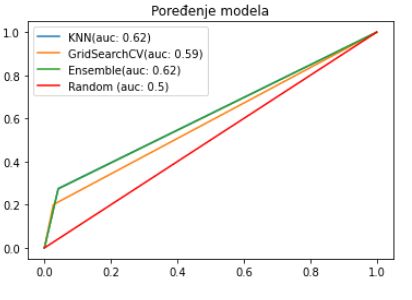
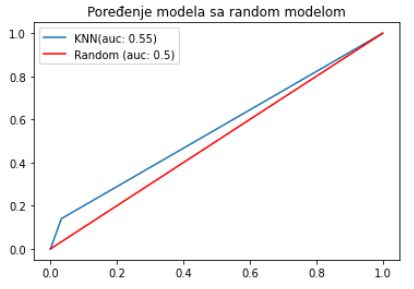




Sa slike vidimo da smo sa 2 glavne komponente uspeli da objasnimo samo 15% ukupne varijanse, pa zakljucujem da nije moguce smanjenje broja atributa na dva. Zatim sam probala da smanjim broj atributa, ali da udeo objasnjenje varijanse bude bar 90%. Uz ovoj uslov broj glavnih atirbuta je porastao na 56.Na slici ispod prikazana je kumulativna suma objasnjene varijanse:



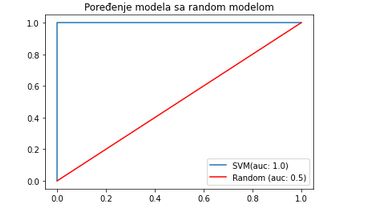
Na transformirane podatke primenila sam KNN algoritam. Na levoj slici ispod prikazano je porednje ovog modela sa random modelom. Dobijeni model je losiji u odnosu na model koji sam dobila primenom KNN algoritma na originalnim podacima (slika desno), ali ne puno losiji.



## 

## 3.3.1. SVM

Poslednji algoritam klasifikacije koji sam obradila jeste SVM. Ideja je pronaci hiperravan koja razdvaja podatke iz 2 klase. Ako su podaci linearno razdvojeni koristimo marginu, odnosno imacemo uslove da se hipperavan nalazi na bezbednom rastojanju od podataka i da su podaci iz jedne ravni sa jedne strane hiperrvani, a drugi sa druge. Ukoliko podaci nisu linearno razdvojeni koristimo meku marginu, odnosno dopustamo da imamo neke pogresne instance. Primenom GridSearchCV arlogitma za model SVM dobila sam da su optimalmi parametri SVM modela {'C= 0.001, 'kernel'= 'linear'}.GridSearch algoritmu nisam naglasila da kao score koristi f1-meru, medjutim, mislim da nije problem sto je koristio accuracy, zato sto pogledom na ROC krivu i AUC vidimo da je model dobar i sa ovom merom.



# 

# 

# 

# 

# 

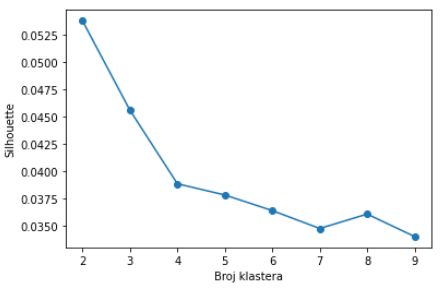
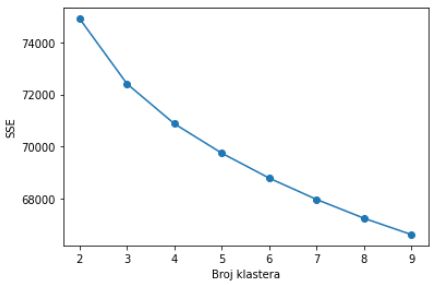
# 

# 4.Klasterovanje

Za razliku od klasifikacije, gde smo imali ciljnu promenljivu koju je trebalo predvideti, klasterovanje spada u nenadledano ucenje, odnosno, bez ciljne promenljive potrebno je naci neke slicnosti izmedju podataka i grupisati podatke. Obradjeni algoritmi klasterovanja: K sredina, hijararhijsko klasterovanje i DBSCAN. Primenom ovih algoritama zaključila sam da moj skup podataka nije pogodan za klasterovanje.

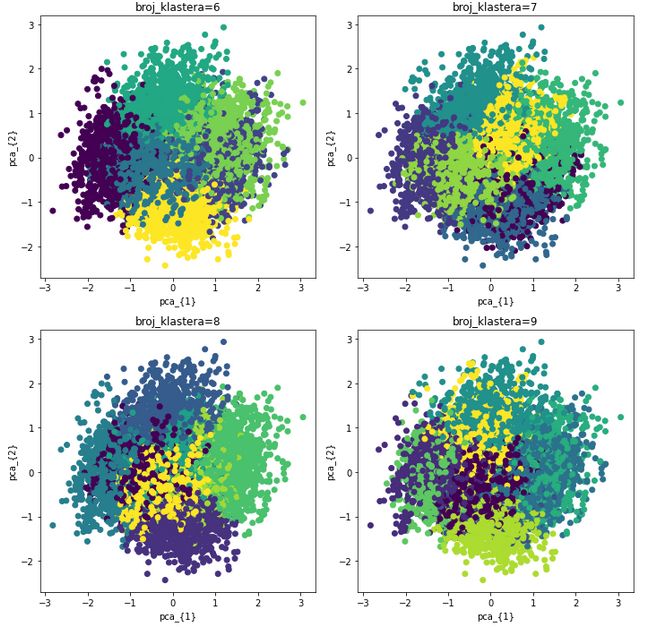
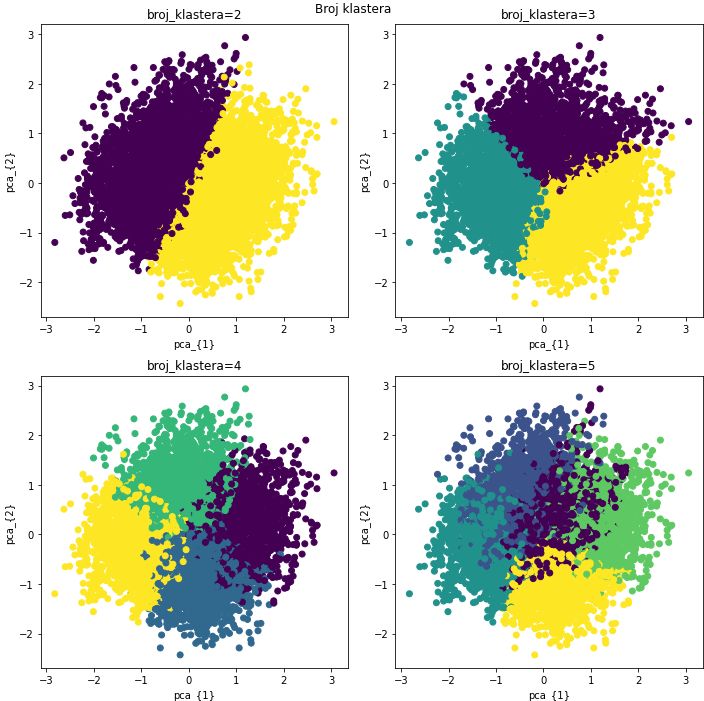
## 4.1. KMeans

KMeans predstavlja iterativni algoritam koji deli podatke u K klastera.Svaka tacka pripada tacno jednom klasteru. Ovaj algoritam zasniva se na reprezentativnim predstavnicima. Prvi korak prilikom primene ovog algoritma jeste odredjivanje optimalnog broja klastera.Na slici ispod prikazana je promena SSE tokom povecanja broja klastera. Promena SSE nije najbolja mera za izbor otimalnog broja klastera, zbog toga sto se povecanjem broja klastera SSE smanjuje, pa zato moramo da posmatramo i promenu Silhouette.Na slici ispod prikazana je ta promena.



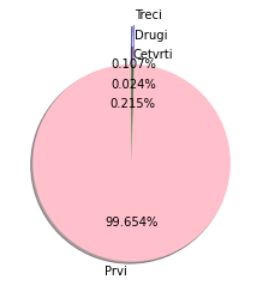
Primenom pravila lakta zakljucujemo da je optimalan broj klastera jednak 4(ili 7?). Silhouette scor za k=4 je priblizno jednak 0.0375, sto je jako lose.

Vec smo videli da primenom PCA algoritma i smanjem broja komponenti na 2 mozemo da objasnimo samo 15% ukupne varijanse podatka, medjutim, uradila sa smanjenje broja komponenti na 2 samo da bih probala da vizualizujem klastere.Na slikama ispod prikazana je promena sa povecanjem broja klastera:



## 4.2. Agglomerative

Primenom algoritma Agglomerative sa parametrima {n\_clusters=4, linkage='average', compute\_distances=True}, dobila sam da najveci broj instanci pripada prvom klasteru, cak 8349, drugom klasteru pripada samo 2 instance, trecem 9, a cetvrtom 18.



Vrednost silhouette scora za ove parametre jeste 0.076253, sto je i dalje jako malo. Proverila sam da li se bolji score dobija ako koristimo single ili complete vezu’, ali u oba slucaja dobijamo manji silhouette score.

## **4.2.DBSCAN**

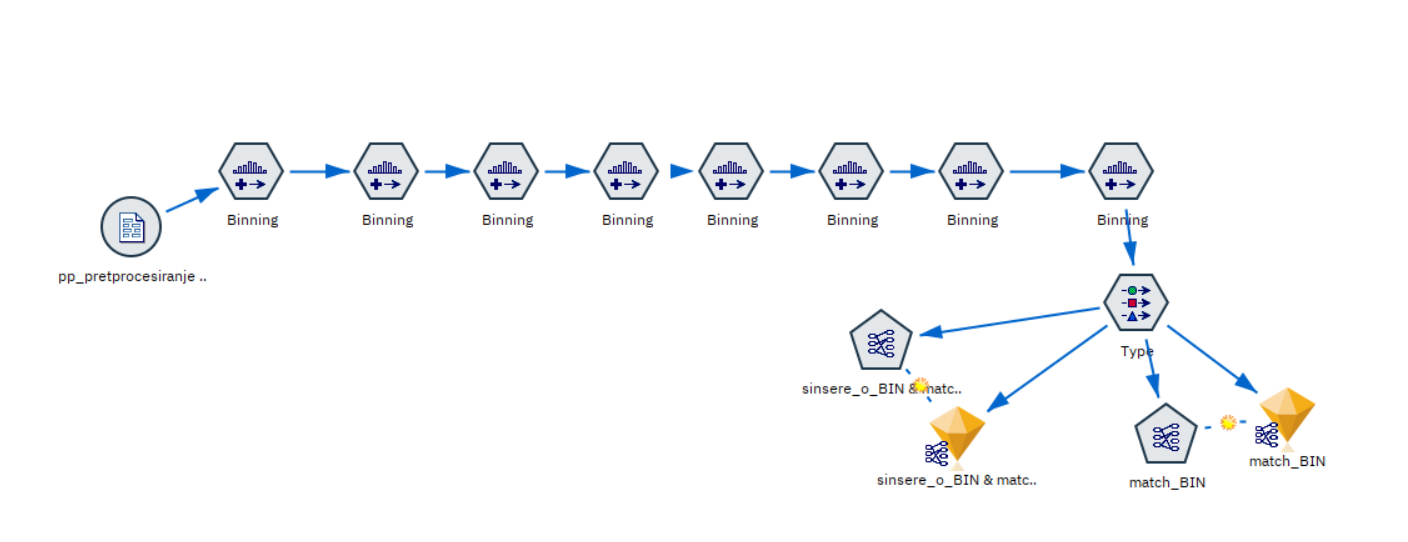
U petlji sam pokusala da odredim koji su dobri parametri ovog modela tako da imamo sto manje suma, odnosno, instanci koje ne pripadaju nijednom klasteru, medjutim, bilo kojom kombinacijom (min\_samples\_value **=** range(2,50) eps\_values **=** [0.1, 0.15, 0.2, 0.25, 0.3]) sam dobila da su sve instance šumovi, odnosno ovaj algoritam nije uspeo da pronadje nijedan klaster. Na osnovu svega ovoga zakljucjujem da moji podaci nisu pogodni za klasterovanje.

# 5.Pravila pridruzivanja

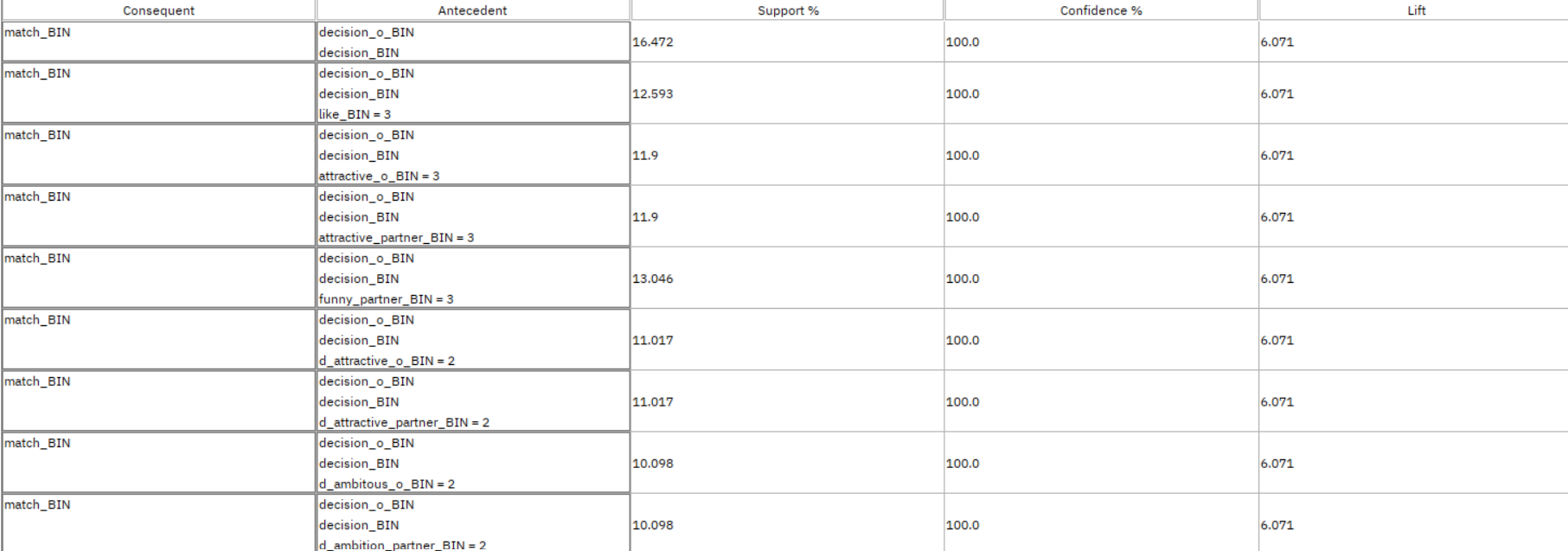
Skup koji sam koristila za pravila pridruzivanja jeste onaj koji sam dobila pretprocesiranjem podataka na pocetku, odnosno zamenom null vrednosti medijanom i koriscenjem LabelEncodera. Zbog jako velikog broja atributa, iskoristila sam Feature Selection da izdvojim glavne atribute. Izdvojila sam 35 najvaznijih atributa, odnosno 36 sa atributom ‘match’. Odlucila sam da cilj pravila pridruzivanja u slucaju mog skupa podataka bude nalazenje skupa osobina koje povlace vrednost atibuta ‘match = 1’, zato sto mi je to imalo najvise smisla.. Obradila sam algoritam Apriori.

## 5.1.Apriori

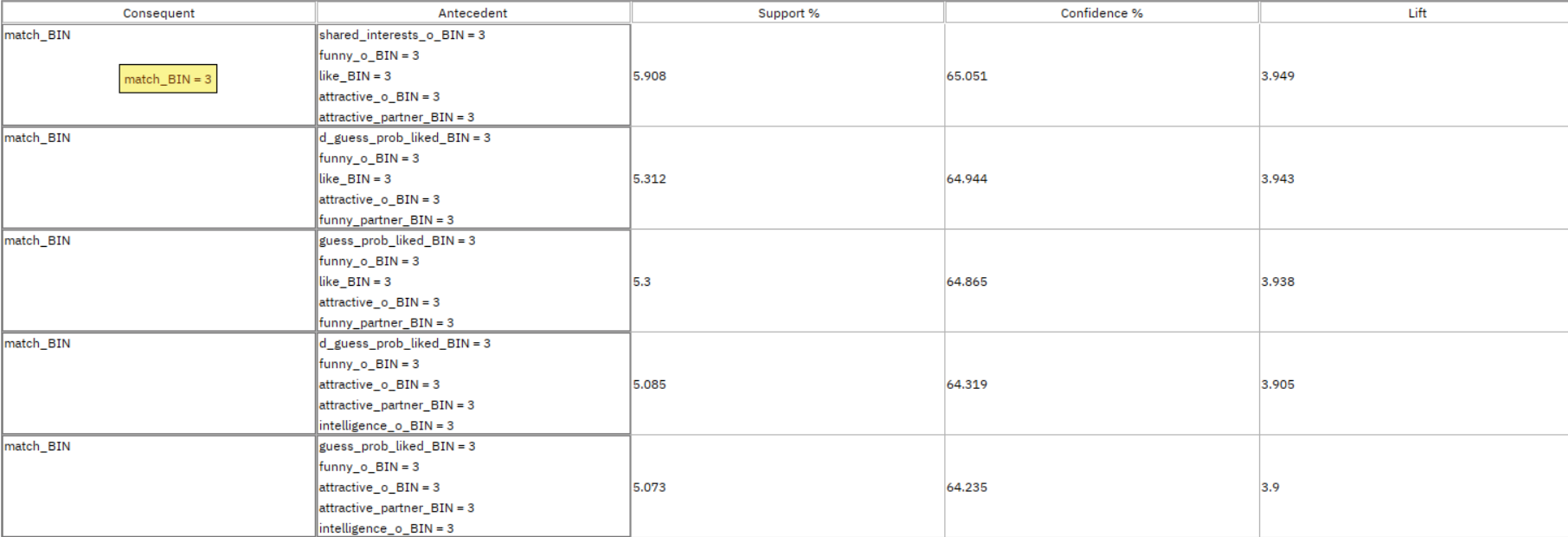
Svih 35 atributa koje sam dobila kao najvaznije za predvidjanje cilja bila su neprekidna, pa sam iskoristila cvor Binning..Nad ovakvim tipovima mogla sam da primenim Apriori algoritam.



Prvo sam primenila algoritam sa uslovima da se u glavi moze naci samo atribut ‘match’, a u telu se mogu naci svi ostali atributi. Pravila koja sam dobila nisu preterano zanimljiva, iako imaju visoku ocenu Lift mere, zato sto je dobijeni zakljucak isti kao i onaj koji sam dobila primenom stabla odlucivanja. Ono sto mora da vazi da bi odgovor za ponovni susret bio pozitivan, tj. da je vrednost promenljive ‘match’= 1, jeste da atributi ‘decision’ i ‘decision\_o’ takodje moraju da budu True. Dakle, dobila sam da ukoliko je vrednost flag atributa ‘decision’ i ‘decision\_o’ true, onda ce se osobe ponovo izadji, bez obzira na vrednost ostalih atributa. Neka od pravila:



Probala sam da izbacim atribute ‘decision’ i ‘decision\_o’ i dobila sam sledeca pravila.Pravila imaju manju pouzdanost i lift meru od gore navedenih, ali nam daju neke nove informacije u odnosu na stabla odlucivanja.



# 6. Literatura

1. Materijali sa predavanja profesora Nenada Mitića

2. Materijali sa vezbi iz kursa Istrazivanje podataka: <https://github.com/MATF-istrazivanje-podataka-1/materijali_2022-2023>

3.Scikit-Learn Machine Learning in Python: <https://scikit-learn.org/stable/>