CONNECT-4

Projekat iz istraživanja podataka

SEPTEMBAR, 2023

MARIJANA ČUPOVIĆ
Asistent: Stefan Kapunac

Contents

Uvod

Ovaj rad je napisan kao prateći document projektu iz predmeta *Istraživanje podataka 1* na Matematičkom fakultetu Univerziteta u Beogradu. Rad sadrži opise tehnika klasifikacije, klasterovanja i pravilia pridruživanja koje su korišćene, kao i analizu rezultata tih tehnika. Rađeno je nad podacima igre connect-4 koji su dostupni na linku, dok se više o samoj igri može naći na sledećem linku.

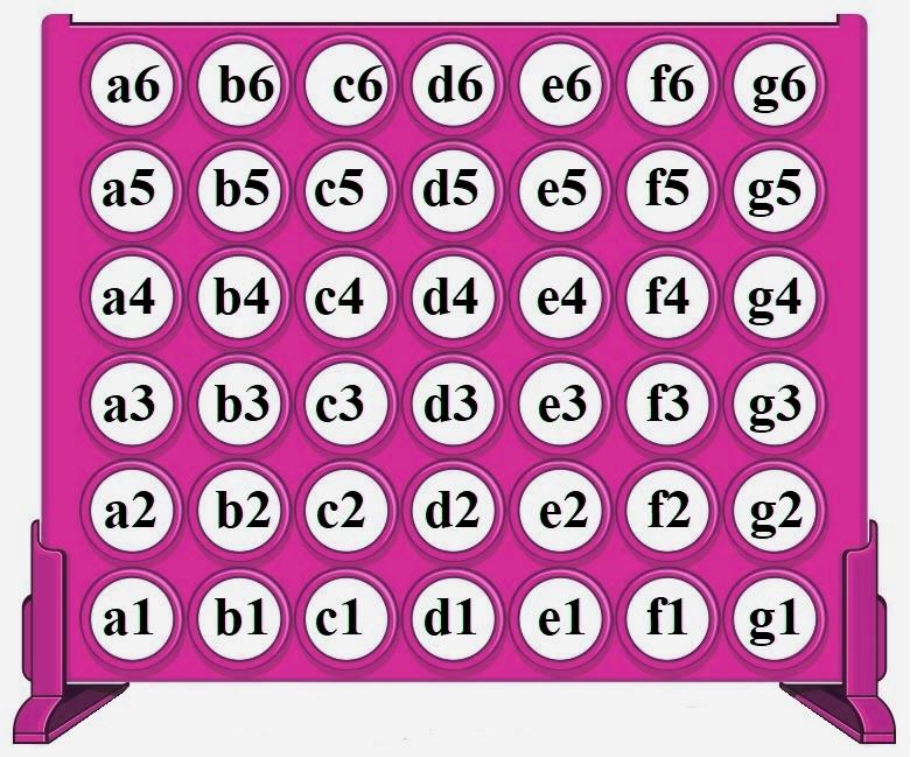
Cilj projekta je bio primena različitih algoritama klasifikacije, klasterovanja i pravila pridruživanja i njihovo poređenje. Za klasifikaciju podataka su korišćeni algoritmi k najbližih suseda, neuronske mreže i drveta odlučivanja. Klasterovanje je rađeno algoritmima k sredina i hijerarjiskim klasterovanjem, dok je za pravila pridruživanja korišćen apriori algoritam u programu SPSS. Skup podataka je namenjen za klasifikaciju ali radi istraživanja svih oblasti koje su rađene na kursu, primenjeni su i ostali algoritmi na iste podatke.

Analiza i priprema podataka

Baza podataka nad kojom je rađeno se sastoji od 67557 instanci koje čuvaju sva legalna stanja igre na ploči 8x8 igre connect-4 gde ni jedan potez nije iznuđen i ni jedan igrač još uvek nije pobedio. Postoje 42 atributa, svaki za po jedno polje table, koji čuvaju informaciju da li je i

	a1	a2	a3	a4	a5	a6	b1	b2	b3	b4	...	f4	f5	f6	g1	g2	g3	g4	g5	g6	Class
0	b	b	b	b	b	b	b	b	b	b	...	b	b	b	b	b	b	b	b	b	win
1	b	b	b	b	b	b	b	b	b	b	...	b	b	b	b	b	b	b	b	b	win
2	b	b	b	b	b	b	o	b	b	b	...	b	b	b	b	b	b	b	b	b	win
3	b	b	b	b	b	b	b	b	b	b	...	b	b	b	b	b	b	b	b	b	win
4	o	b	b	b	b	b	b	b	b	b	...	b	b	b	b	b	b	b	b	b	win

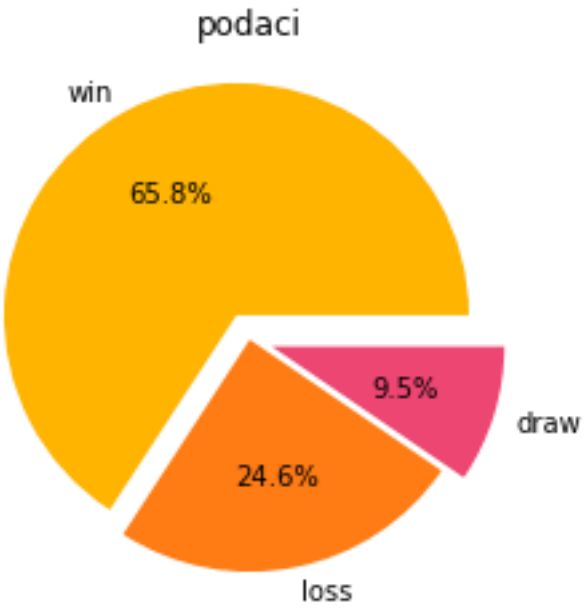
od strane koga je polje popunjeno. Ukolko je prvi igrač popunio neko polje, odgovarajući atribut će sadržati *x*, ako je drugi igrač zauzeo polje, onda će atribut sadržati vrednost *o*, a ako je plje prazno odgovarajući atribut u instanci će sadržati vrednost *b*. Postoji još jedan kategorički atribut koji može da sadrži jednu od tri vrednosti *win, loss, draw* koji čuva podatak o ishodu igre za to



stanje. Imena aributa su dvočlana, sastoje se od slova i cifre i kodiraju pozicije na table kao što je prikazano na slici. U bazi nema nedostajućih vrednosti, pa celokupan dalji rad nastavljen nad nepromenjenom bazom u tom pogledu, jedino što je urađeno, a ne menja informativnost baze je preimenovanje kategoričkih vrednosti.

Oznake za zauzetost polja su prevedene u *0* za prazno polje, *1* za prvog igrača, a *-1* za drugog igrača, dok je klasa, tj ishod igre zamenjena sa *0* ako je bilo nerešeno, *1* ako je pobedio prvi igrač i sa *-1* ako je pobedio drugi igrač.

Klase u bazi podataka nisu balansirane ali zbog prirode podataka i zadataka koje želimo da odradimo nema smisla balansirati klase. Želimo da modeli imaju pravilnu sliku o raspodeli ishoda u svim stanjima igre i ne možemo ni jedno od njih zanemariti.

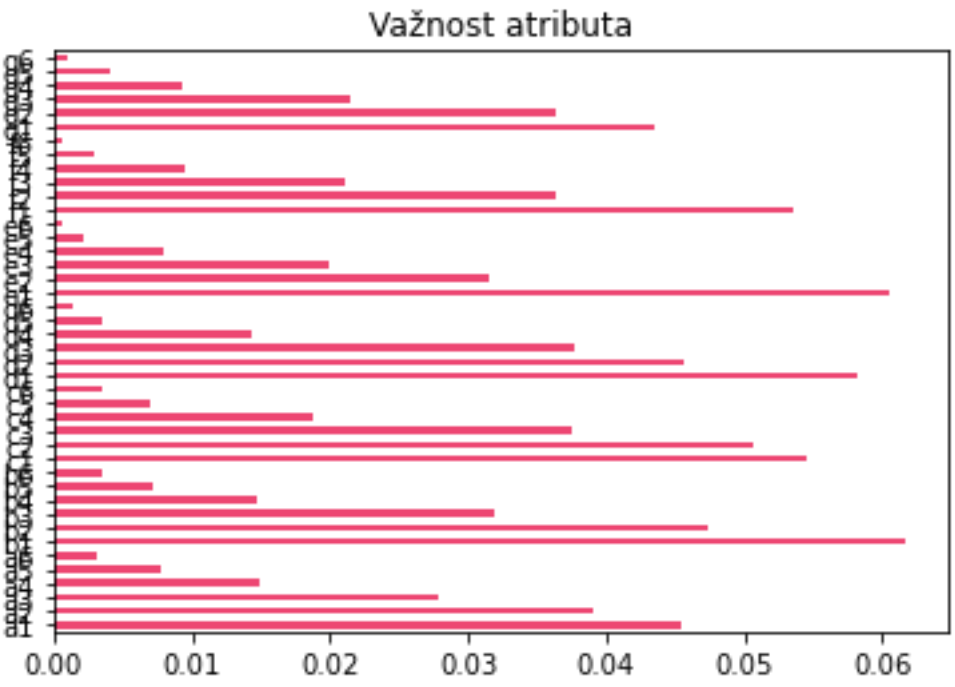


Nakon analize koorelacije atributa i klase vidi se da nema ni jednog atributa koji ima značajnu vezu i može se izdvojiti kao bitna komponenta u daljem radu. Jedino što je ostalo je da se podaci pripreme za dalje korake. Za klasifikaciju su podaci podeljeni na trening i test podatke, dok su za klasterovanje i pravila pridruživanja podaci ostali nepodeljeni.

Klasifikacija

Drveta odlučivanja

Kreirajanjem najosnovnijeg drveta odlučivanja možemo da pogledamo koji atributi su bili najznačajniji prilikom kreiranja drveta i da to prikažemo grafički. Vidimo da su najveći značaj imali atributi koji čuvaju informaciju o prvoj koloni odozdo, a onda drugoj, a najmanje oni koji su na vrhu. Ovo je u saglasnosti s pravilima igre koja naglašavaju da se polja popunjavaju odozdo nagore, što znači da će u velikom broju instanci, tj. stanja igre polja na samom vrhu table ostati prazna, pa samim tim i manje ili u potpunosti nebitna za određivanje pobetnika igre.



Iako je kreirano drvo bez izmene ikakvih parametara, ono je u potpunosti uspelo da opiše trenindatke, tj da preciznost za sve klase, tj. ishode igre, bude 1. Međtim, mogućnost ovog drveta da predvidi klase test podataka je znatno manja, accuracy je 0.77, što nije previse loše ali nam se drvo preprilagodilo. Možemo primetiti da je drvo, ipak, dosta precizno kad je dodela pobede prvom igraču u pitanju, što može da bude korisno ako želimo da obučimo nekog agenta da pobeđuje i potrebna nam je ocean njegovog trenutnog stanja. Dakle drvo, iako najprostije moguće, nije beskorisno.

Grid search

Kako bi se odredilo bolje drvo odlučivanja korišćen je grid-search za ekstenzivnu pretragu kombinacija hiperparametara koji će dati najbolje rezultate. Zadate su maksimalne dubine do kojih

drvo može doći (2, 4, 6, i 8), kao i mere nečistoće (Gini i entropija). Kao najbolji par hiperparametara su se pokazali Gini i dubina 8 i dali accuracy 0.74 na trening i 0.73 na test podacima, što je lošije od najprostijeg drveta, ali barem znamo da se drvo nije previse prilagodilo podacima. Čak se i iz matrica konfuzije može videte da ovja model greši više od onog koji je inicijalno kreiran, ali da mu je data na raspolaganje veća dubina možda bi se bolje pokazao. Mađutim, produbljivanjem drveta ćemo pre naići na problem preprilagođavanja i uvek je bolje uzeti model koji je jednostavniji.

	0	1	-1
0	2252	382	707
1	372	383	540
-1	643	525	7708

Figure 1: matrica konfuzije za prosto drvo na test podacima

	0	1	-1
0	1570	26	1745
1	267	22	1006
-1	550	39	8287

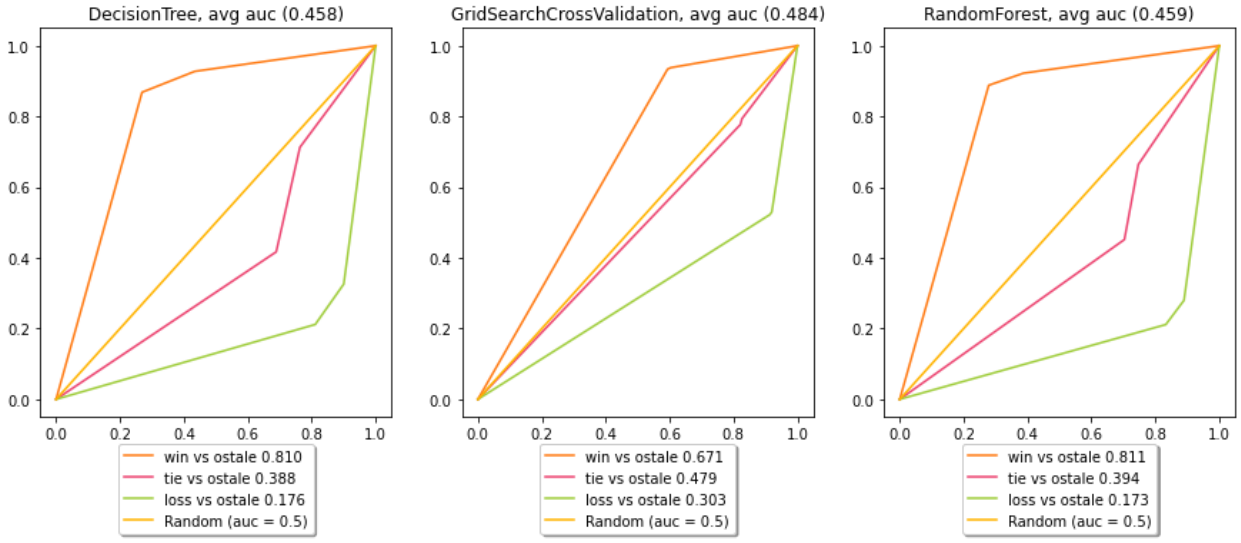
Figure 2: matrica konfuzije za grid-search drvo na test podacima

Random forest

Postoji mogućnost da se poboljšaju ishodi drveta kombinovanjem više stabala u šumu. Ideja je da se više, manje moćnih drveta, udruži i zajedno da model veće moći. Kreiranjem šume sa mamo 5 drveta je već povećalo accuracy na test podacima na 0.78 što je bolje od svega što je do sad viđeno. Ako želimo da poboljšamo i ovaj rezultat možemo da dodamo još veći broj drveta, odabrano je da to bude 500. Šuma sa 500 drveta u potpunosti opisuje trening podatke, što nije ni čudno s obzirom da je veoma veliki broj drveta na raspolaganju. Logično je da se model dobro pokazao I na test podacima postignuvši accuracy od 0.83, što je dosta dobro, ali ako uzmemo u obzir da je sa 0.79 accuracy došao na 0.84 kada smo 100 puta povećali broj drveta možemo reći da je onaj sa manje drveta ipak bolji. Veliki broj resursa koji su uloženi na kreiranje šume od 500 drveta nam je doneo malo poboljšanje u odnosu na skromnih 5 drveta, pa je smatrati ovaj prostiji model moćnijim sasvim opravdano. Dodatno moežemo primetiti da šuma od 5 drveta može da se analizira od strane čoveka i da da neki uvid u podatke koji će čoveku biti jasni, dok šumu sa 500 drveta može da koristi samo računar ili neki veoma entuzijastični istraživač.

Poređenje modela

Pošto radimo sa 3 klase nećemo moći da nacrtamo ROC krivu klasičnim metodama, ali možemo da pogledamo kako modeli prepoznaju jednu klasu naspram svih ostalih i da to uporedimo. Vidimo da modeli dosta dobro određuju pobedu prvog igrača ali da su nerešeno i gubitak dota loši, čak gori od nasumične dodele. Ovakvo poređenje modela jeste dosta deskriptivno ali daje dosta pesimističnu sliku o njima jer smo navikli na ROC krive za binarnu klasifikaciju. Čak i kada se pogleda prosečni AUC modeli izgledaju neimpresivno i skoro neupotrebljivo jer je AUC kod svih manji od 0.5 što je vrednost AUC-a nasumičnog modela. Mađutim, ako se pogleda accuracy za ove modele oni mnogo bolje izgledaju i kao najbolji se izdvaja šuma od 5 drveta.



K najbližih useda

Zanimljiva metoda za klasifikaciju je k najbližih suseda koja može da da lepe rezultate kada su hiperparametri zadati pogodno. Pošto je teško odrediti koji su to tačno hiperparametri, najbolje je iskoristiti grid-search kao kod drveta i kreirati nekoliko modela sa rasličitim hiperparametrima i odabrati najbolji. Hiperparametar koji je ispitivan je naravno k i za njega su uzimane vrednosti iz intervala [3, 10], a za procenu kvaliteta modela je korišćen accuracy. Vidimo da se za accuracy svih modela kreće od 0.75 do 0.804 što zanči da bi svako ka koje je iz intervala [3, 10] dalo pristojene rezultate, ali najbolji je ipak model koji razmatra 9 suseda postižući accuracy od 0.804. Na test podacima je model skoro podjednako moćan jer je accuracy 0.802 što znači da je model mogao lepo da uopšti znanje o podacima i da se ne preprilagodi.

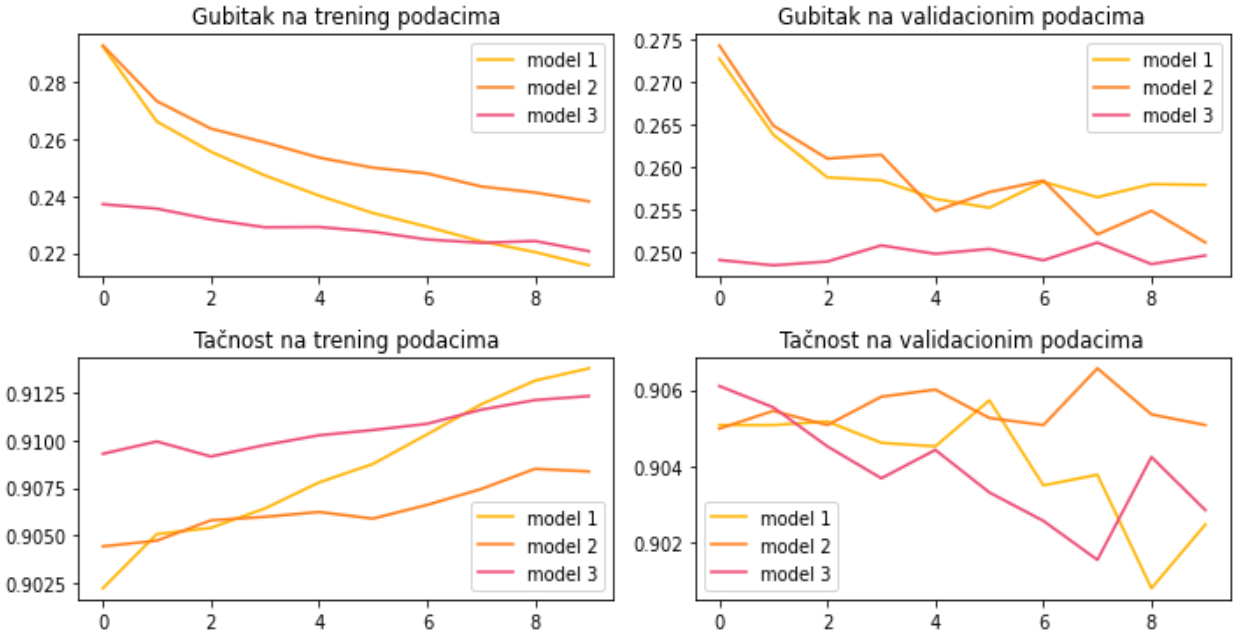
```
[2357,  113,  871]
[ 367,  187,  741]
[ 479,   99, 8298]
```

Figure 3: matrica konfuzije knn na test podacima

Neuronske mreže

Neuronske mreže su veoma jak alata kod istraživanja podataka jer koriste veliki broj parametara u radu i mogu da uoče bitnost atributa koju čovek ne bi mogao. Kod podataka igd+re connect 4 su se pokazale veoma dobro. Model sa samo 3 sloja i treniranja u trajanju od 10 epoha je postigao acurracy od 0.904 što je bolje od svih modela koje smo do sada videli. Dodavanjem Dropout sloja je malo poboljšalo accuracy, podigavši ga na 0.907. Dodavanjem još jednog sloja nismo dobili pomak jer je accuracy sada 0.904, što nije loše ali je gore od prethodne verzije.

Možemo da pogledamo kako se gubitak i tačnost mejaju kroz epohe na sva tri modela, na trening i na validacionim podacima. Vidimo da se svi modeli ponašaju lepo da gubitci opadaju, a da tačnost raste ali i da svi konvergiraju dosta bliskim vrednostima što znači su modeli skoro podjednako jaki. Ako želimo da odaberemo najbolji model, možemo da pogledamo sve podatke zajedno i da kažemo da je to model 2 kod kog smo videli najveću tačnost. Njemu u prednost ide i to što nema preterano komplikovanu arhitekturu, a daje veoma dobre rezultate.

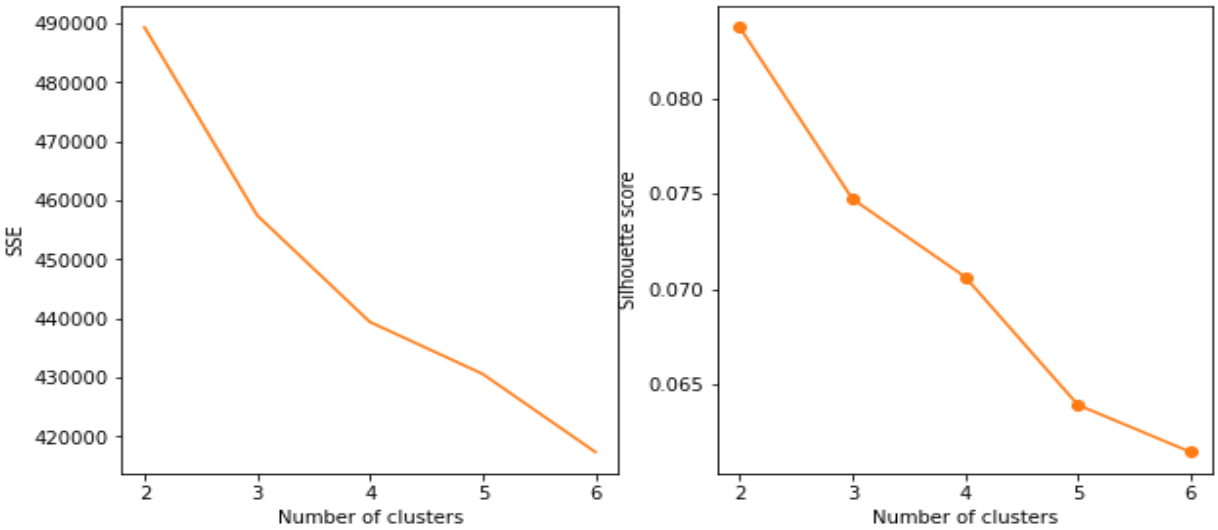

Gubitak na trening podacima
model 1
model 2
model 3
0.28
0.26
0.24
0.22
0
2
4
6
8
Gubitak na validacionim podacima
model 1
model 2
model 3
0.275
0.270
0.265
0.260
0.255
0.250
0
2
4
6
8
Tačnost na trening podacima
model 1
model 2
model 3
0.9125
0.9100
0.9075
0.9050
0.9025
0
2
4
6
8
Tačnost na validacionim podacima
model 1
model 2
model 3
0.906
0.904
0.902
0
2
4
6
8

Klasterovanje

Baza podataka connect 4 ima 3 prirodna klastera koje uočavamo na osnovu klase kojoj instanca pripada i tu informaciju možemo da koristimo tokom analize rezultata ali da bi projekat bio u duhu istraživanja, pravićemo se da to ne znamo kada budemo kreirali modele.

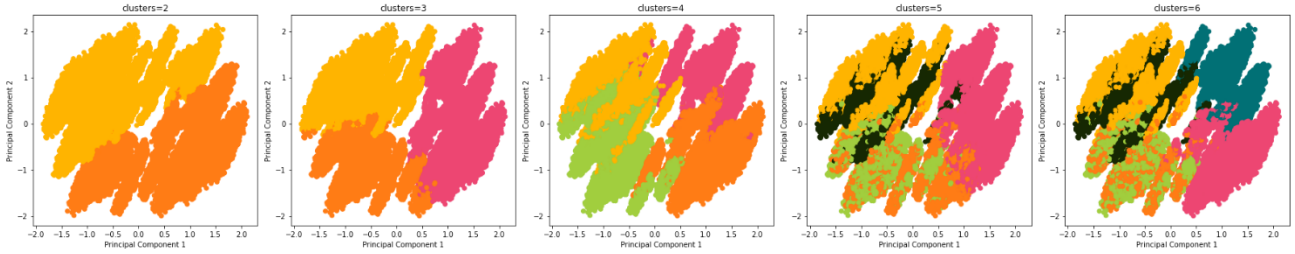
K sredina

Algoritam K sedina zahteva da mu se kaže koliko klastera treba da traži, pa će taj hiperparametar biti uzet iz intervala [2, 6], pri čemu ćemo pratiri srednje kvadratnu grešku i siluetu kako bismo odredili za koju vrednost hiperparametra k je klasterovanje najbolje. Srednje kvadratna



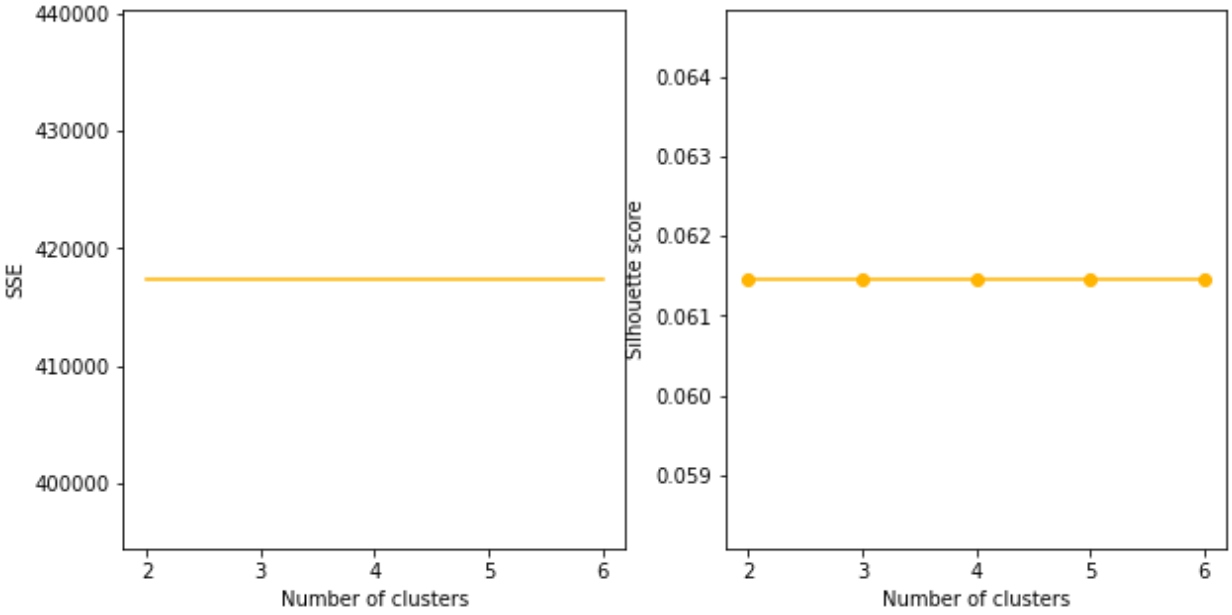
greška, naravno opada s porastom broja klastera, ali i silueta, koja ostaje u okolini nule. Kako mi znamo da postoje 3 prirodna klastera možemo videti da algoritam veoma loše procenjuje tu vrednost jer ni pravilo lakta ni analiza siluete ne ukazuju na to su 3 klastera dobar izbor, štaviše, ako pogledamo siluetu, najbolje bi bilo da odaberemo 2 klastera. Medjutim ovo je veoma loše klasterovanje.

Da bismo prikazali podatke na čoveku lako razumljiv način, moramo smanjiti dimenziju podataka. Za to smo koristili PCA (Principal Component Analysis) da izvojimo 2 ključne

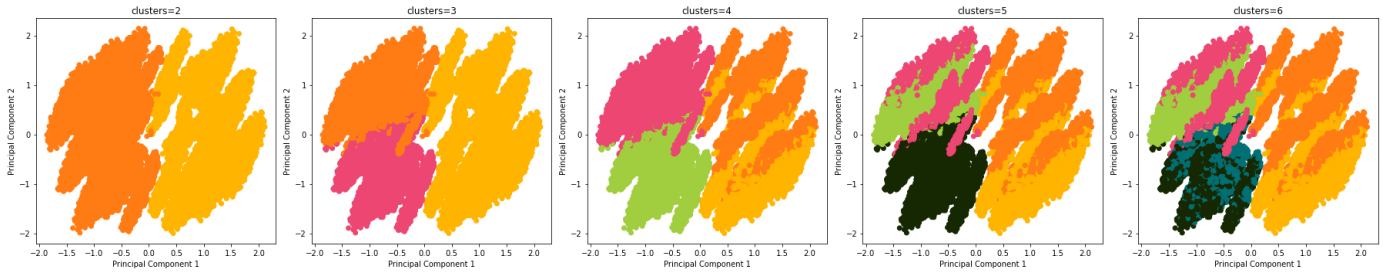


komponente. Zbog ovoga, klasteri izgledaju mnogo prostije nego što zapravo jesu, ali nama je dovoljno dobro da vidimo kako od prilike izgledaju klasteri. Vidmo da su dosta tesni kao ih je 2 ili 3, međutim za ostale vrednosti klasteri su mnogo raštrkaniji.

Pošto je ovaj pristup dao neimpresivne rezultate, možemo da probamo Bisecting k means sa istim skupom koji prosleđujemo za k kao za običan k sredina. Za ocenu kvaliteta klasterovanja ćemo opet posmatrati srednje kvadratnu grešku i siluetu. Ovaj algoritam ne daje nikakvo

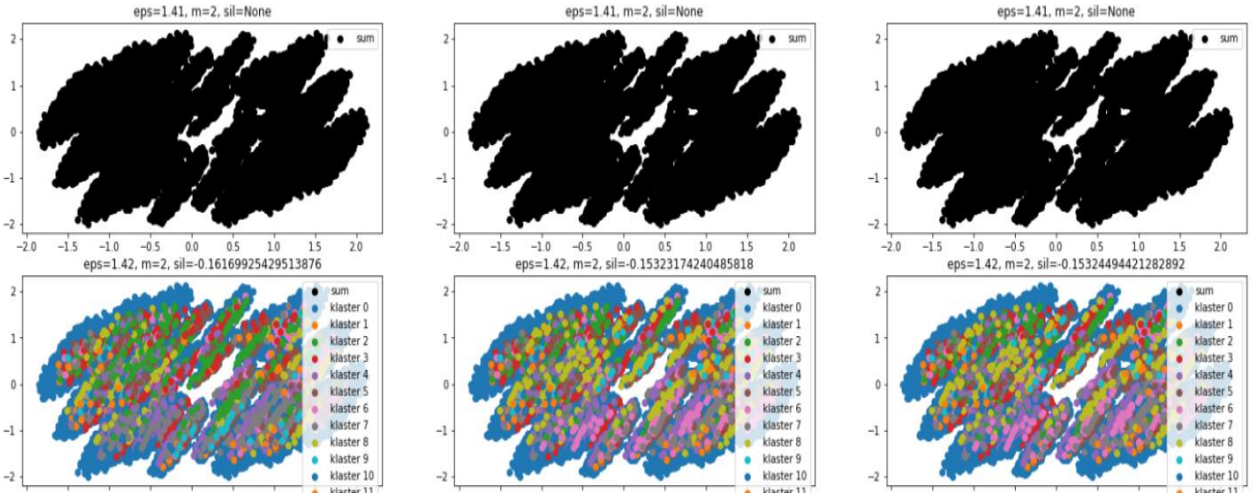


poboljšanje u odnosu na predhodni jer nam kaže da je sve jedno koliko klastera odaberemo jer je i greška isa i silueta, koja je veoma mala. Ako pogledamo klastere vidimo da se za k=2 i k =3 malo razlikuju od onih koje smo dobili predhodnim algoritmom.

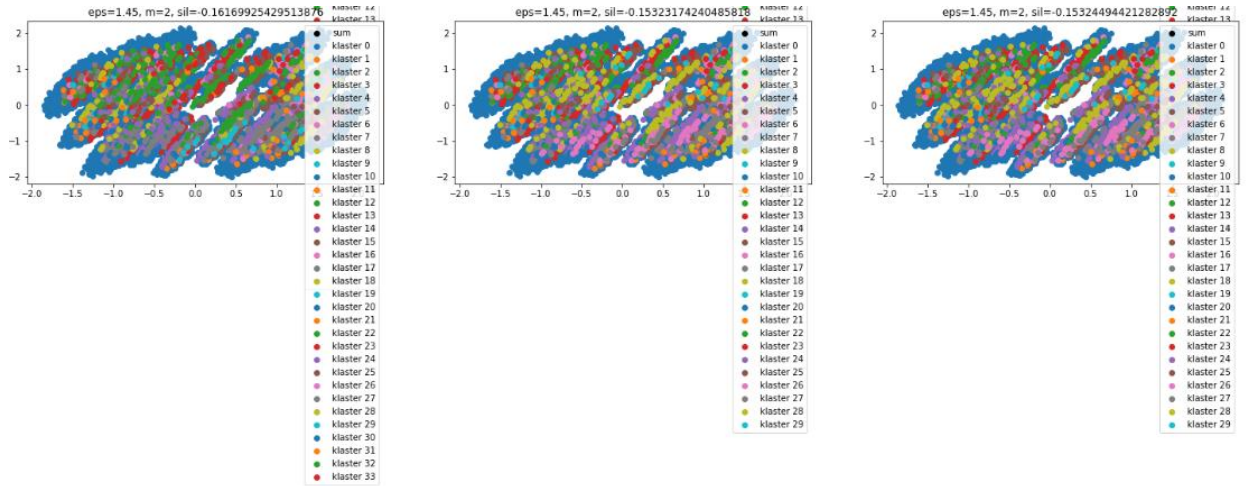


DBSCAN

Za rad sa DBSCAN algoritmom moramo zadati broj suseda koji su potrebni za prihvatanje instance u klaster, kao i maksimalnu udaljenost između dve instance da bi se one smatrale bliskim. Ovo je veoma teško odrediti za podatke koji imaju raspodelu koju imaju podaci iz baze connect 4. Nakon velikog broja pokušaja da se intervali za ova dva parametra kreiraju tako da daju bilo kakve



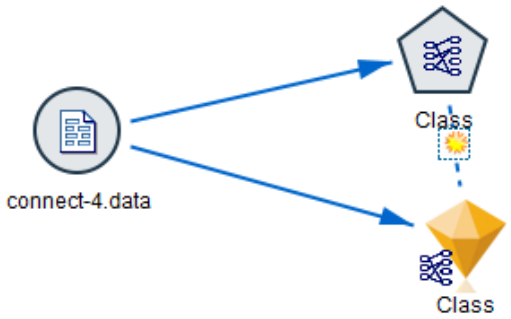
rezultate, odlučeno je da se epsilon uzima iz intervala [1.42, 1.45] sa korakom 0.01, a minimalni broj instanci iz [5, 7]. Tada vidimo da za prvu vrednost epsilona nemamo nikakve klastere, a već za narednu imamo eksploziju klastera i malu vrednost siluete koja ukazuje da su klasteri veoma raštrkani i slabo povezani. Daljim iteriranjem kroz hiperparametre ne dobijao bolje rezuletate već se može videti da je silueta ostala nepromenjena, tj. zadržala se na se -0.15. Broj klastera je išao i preko 30 što znači da je algoritam vrlo loše određivao klastere jer mi znamo da ih je zapravo 3.



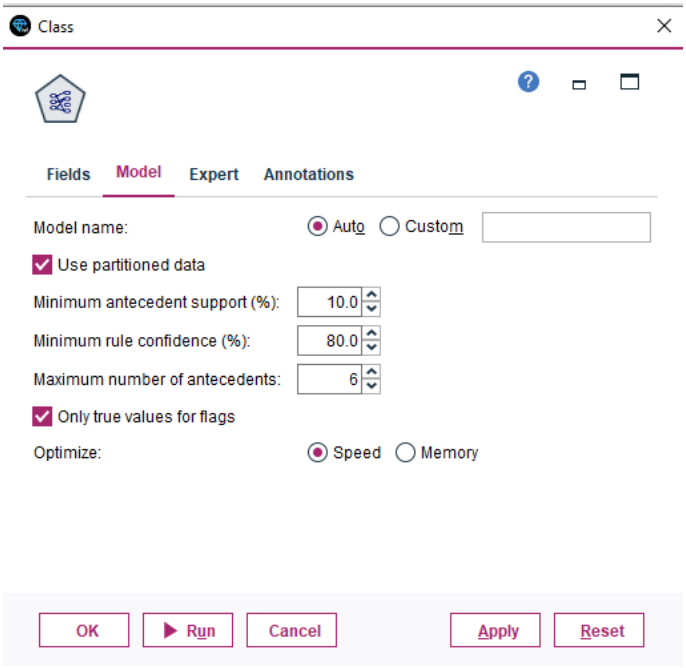
Pravila pridruživanja

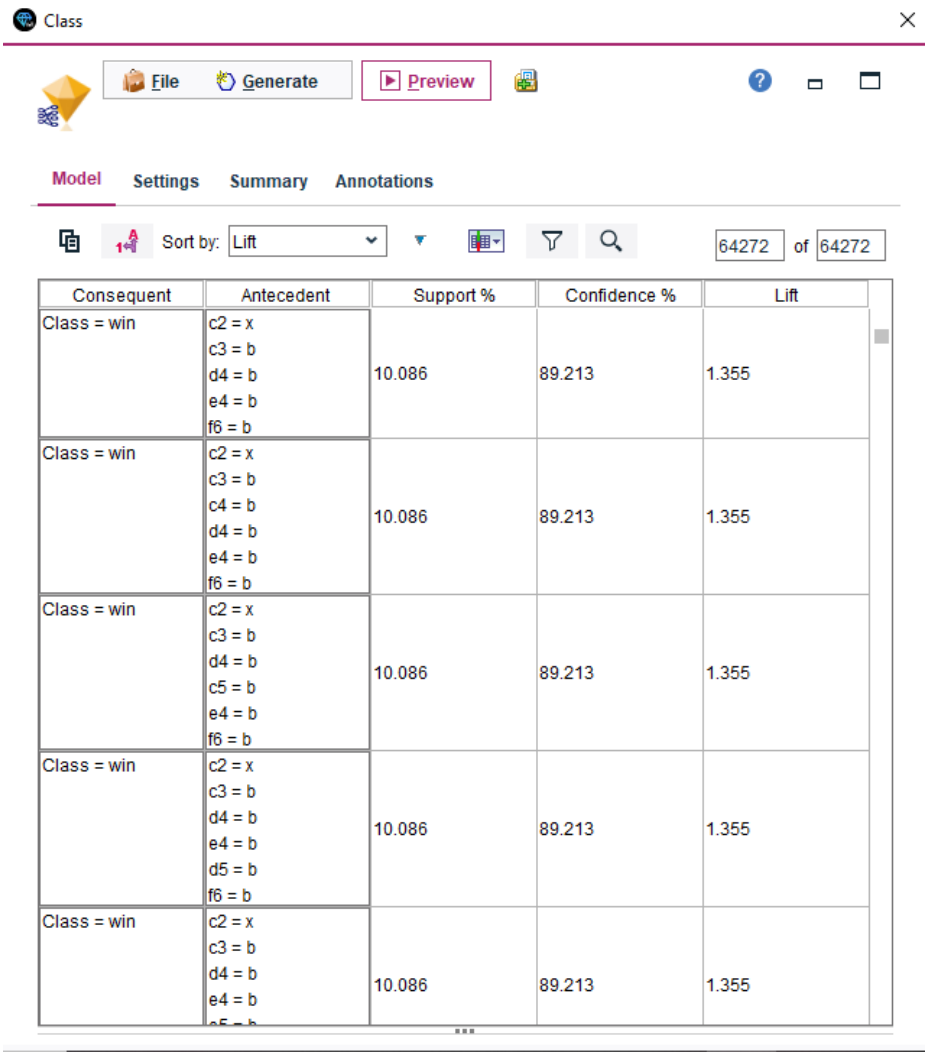
Apriori

Apriori algoritam služi za uočavanje obrazaca u podacima koji značajni, međutim, naši podaci nisu dovoljno deskriptivni da bi se našao neki obrazac, ali možemo da pokušamo da nađemo vezu između klase i ostalih atributa u bazi podataka. Za rad apriori algoritma nije bila potrebna nikakva obrada podataka, tkd je korišćen skup podataka onakav kakv je isporučen.



Za kreiranje modela korišćen je SPSS Modeler u kojem su komponente povezane kao na slici, dok je minimalna podrška postavljena na 10%, a minimalna pouzdanost je 80%.





Consequent	Antecedent	Support %	Confidence %	Lift
Class = win	c2 = x c3 = b d4 = b e4 = b f6 = b	10.086	89.213	1.355
Class = win	c2 = x c3 = b c4 = b d4 = b e4 = b f6 = b	10.086	89.213	1.355
Class = win	c2 = x c3 = b d4 = b c5 = b e4 = b f6 = b	10.086	89.213	1.355
Class = win	c2 = x c3 = b d4 = b e4 = b d5 = b f6 = b	10.086	89.213	1.355
Class = win	c2 = x c3 = b d4 = b e4 = b	10.086	89.213	1.355

Model je našao 64272 pravila i sva se smatraju zanimljivim jer im je lift veći od 1, ali za vrlo malo. Ako malo razmotrimo značenje pravila koja je apriori našao videćemo da je on uočio da će do pobede doći , na primer, ako je igrač zauzeo polje c2, a iznad njega kao i desno-gore su polja prazna. Ovo ima smisla kada se uzme u obzir kako igra funkcioniše, ali nam pravilo ne daje nekei dodatni uvid u to kakvi obrasci vode pobedi.

Zaključak

Prilikom istraživanja podataka moramo obratiti pažnju na to kakvi su naši podaci i kakvu obradu zahtevaju i kakva analiza je nad njima moguća. Dati skup podataka je veoma pogodan za klasifikaciju, što se i videlo kroz modele koji su veoma uspešno uspeli da predviđaju klase test podataka. Međutim, znatno lošije se pokazao pri klasterovanju i pravilima pridruživanja jer su podaci bili takvi da algoritmi koji su primenjeni teško uočavaju potrebne veze.