Matematički fakultet, BU

Istraživanje podataka 1

Projekat na temu: Drug consumption

Asistent: Stefan Kapunac

Profesor: prof. dr Nenad Mitić

Radio:

David Živković 098/2020

jun 2023. godine

Sadržaj

1. Uvod

Projekat iz predmeta istraživanje podataka, rađen nad skupom podataka „Drug consumption quantified“. Projekat je rađen u programskom jeziku **Python** i u IBM-ovom pomoćnom alatu **SPSS**.

Cilj projekta je prikaz različitih tehnika obrade podataka, prevashodno analize i pretprocesiranja ne bi li se oktrilo nešto više o datom skupu podataka. Naime, skup podataka sadrži informacije o konzumacijama više supstanci za koje treba pronaći nešto više informacija. Nakon toga i primena raznih tehnika mašinskog učenja, konkretno istraživanja podataka, te tehnike se svrstavaju u sledeće grupe:

- **Klasifikacija** – na osnovu zadatih atributa odrediti kojoj klasi pripada svaka instanca podataka (konkretno ovde za svaku supstancu ispitati koliko često je konzumirana)
- **Klasterovanje** – pokušaj da se grupisanjem podataka dođe do dodatnih pravilnosti među podacima
- **Pravila pridruživanja** – pokušaj da se nadju međusobne zavisnosti među klasama, u smislu

Podaci sami po sebi su namenjeni klasifikaciji i shodno tome tu prilagodjeni tehnikama klasifikacije, ali su (sa znatno lošijim rezultatima) primenljivi i za ostale tehnike.

Uopšteno o podacima

Skup podataka se sastoji od 1885 instanci i 32 osobine tih instanci (ukupno 60320 ćelija). Od tih 32 atributa, skup se sastoji od 13 atributa (među kojima je jedan atribut jedinstveni identifikator, konkretno ovde redni broj) koji predstavljaju lične podatke o instancama (osobama), i njihove psihološke osobine. Pored tih atributa se nalazi i 19 klasa koje se klasifikuju svaka ponaosob (među kojima se nalazi kontrolna klasa „Semer“ koja predstavlja supstancu koja ne postoji, tako da se izbacuje).

Atributi su jedinstvene numeričke (realne) vrednosti koje označavaju odgovarajuću kategoričku vrednost, dok su klase kategoričke vrednosti koje imaju raspon CL0 – CL6 shodno tome koliko je neka supstanca bivala konzumirana.

Lični podaci instanci su:

- **ID:** ID ispitanika: 1-1885
- **Age:** Broj godina: 18-65+
- **Gender:** Pol: M, Z
- **Education** Stepen obrazovanja: osnovno – doktorat
- **Country:** Drzavljanstvo: (SAD, UK, NZ, Australija, Irska, Kanada, Ostalo)
- **Ethnicity:** Etnicka pripadnost: (Evropeidno, Negroidno, Mongoloidno, Zambos, Mulat, Melez)

Psihiloške osobine instanci su:

- **Nscore:** Stepen neurotičnosti: odgovarajuća vrednost
- **Escore:** Stepen ekstrovertnosti: odgovarajuća vrednost
- **Oscore:** Stepen otvorenosti prema novim idejama/doživljajima: odgovarajuća vrednost
- **Ascore:** Stepen slaganja sa drugima: odgovarajuća vrednost
- **Cscore:** Stepen savesnosti: odgovarajuća vrednost
- **Impulsiveness:** Stepen impulsivnosti: odgovarajuća vrednost
- **SS:** Doživljavanje senzacija od strane čula: odgovarajuća vrednost

Klase koje se ispituju su:

- **Alchocol**
- **Amphet**
- **Amyl**
- **Benzos**
- **Caff**
- **Cannabis**
- **Choc**
- **Coke**
- **Crack**
- **Ecstasy**
- **Heroin**
- **Ketamine**
- **LegalH**
- **LSD**
- **Meth**
- **Mushrooms**
- **Nicotine**
- **Semer** (lažna supstanca, tako da se odbacuje)
- **VSA**

Svaka od navedenih supstanci ima raspon vrednosti CL0-CL6, što označava:

- **C0** - Nikad korišćeno
- **C1** - Korišćeno pre vise od 10 godina
- **C2** - Korišćeno u zadnjih 10 godina
- **C3** - Korišćeno u zadnjih godinu dana
- **C4** - Korišćeno u zadnjih mesec dana
- **C5** - Korišćeno u zadnjih nedelju dana
- **C6** - Korišćeno u zadnjih 24h

Pretprocesiranje

Od pomoćnih biblioteka je korišćeno:

- numpy (transformacija skupa podataka)
- pandas
- matplotlib.pyplot (vizuelizacija rezulatat i podataka)
- seaborn (vizuelizacija)
- sklearn (algoritmi za klasifikaciju i klasterovanje)

Podaci su sami po sebi vec pretprocesirani i spremni za primenu većine algoritama klasifikacije nad njima. Neophodno je bilo dodati imena atributa i klasa za sve podatke jer se unutar .data fajla (koji je manuelno prebacen u .csv fajl, zbog lakseg formatiranja podataka) ne nalaze te informacije, moglo se i uraditi „ručno“, direktnim zapisivanjem unutar fajla, ali su na kraju imena kolona ubačene uspomoc Python-a.

Jedino što bi trebalo raditi jeste *feature selection* i *feature extraction*, ne bi li se izbacile nepotrebni atributi (poput ID npr), isto treba izbaciti podatke koji prikazuju loš stepen korelacije, kako ne bi smetali, što se ispituje matricom korelacije. Vrednosti van granica su ispitane crtanjem odgovarajućih *boxplot*-ova za svaki atribut ponaosob. Nakon crtanja je izvršena provera *zscore*-a, čime se prebrojalo koliko ima elemenata van granica za svaki atribut. Pokazalo se da ih ima minimalno, usled čega nije bilo potrebe sanirati ih.

Nedostajuće vrednosti ne postoje, što je i naznačeno u opisu podataka, ali je svako izvršena provera da slučajno ne postoje i neki nekonvencijalni načini zapisivanja nedostajućih vrednosti. Kategoričke vrednosti atributa su transformisane u realne konstante, vrednosti čak i deluju normalizovane, ali zbog primena nekih algoritama potrebno ih je doskalirati na raspon [0, 1].

Zarad prikaza balansiranosti klasa I razumevanja datih numeričkih vrednosti izvršena je I transformacija podataka u njima odgovarajuće kategoričke, kako bi bilo lakše uvideti raspon vrednosti po atributima I kategorijama. Isti podaci su kasnije upotrebljeni prilikom rada sa pravilima pridruživanja. Balansiranost je prikazana crtanjem odgovarajućih *barplot*-ova

Pojedini atributi su visoko dizblansirani, ali su to atributi koji I nisu od preteranog značaja, (što se I vidi kod algoritma stabla odlučivanja) poput godine, državljanstvo, itd. Atributi koji imaju znatno veću ulogu pri odgovarajućoj klasifikaciji, poput neurotičnosti, impulsivnosti neke osobe, itd. su u potpunosti balansirani. Pojedine klase su malo lošije balansirane, ali ništa preterano, tako da su I više nego adekvatne za upotrebu.

Elementi van granica nisu primetni, tako da nije bilo potrebe odstanjivati ih, provera se vršila crtanjem odgovarajućih boxplotova.

2. Primene modela

KLASIFIKACIJA

Modeli korišćeni za klasifikaciju su: *DecisionTreeClassifier*, *KNeighborsClassifier*, *MLPClassifier*, *RandomForestClassifier*, *LogisticRegression*. Od pomoćnih funkcija implementirano je *report* za evaluaciju modela, *report_imbalanced* za evaluaciju I prikaz matrice korelacije, kao I druge funkcije za vizuelizaciju rezultata.

DecisionTreeClassifier – stabla odlučivanja su hijerarhijske strukture koje se sastoje iz čvorova I grana. Svaki čvor je upit na osnovu koga se vrednosti dalje prosledjuju odgovarajućim granama. Vrednosti se kreću do listova u kojima se vrši klasifikovanje kako bi se odredilo kojoj klasi data instanca.

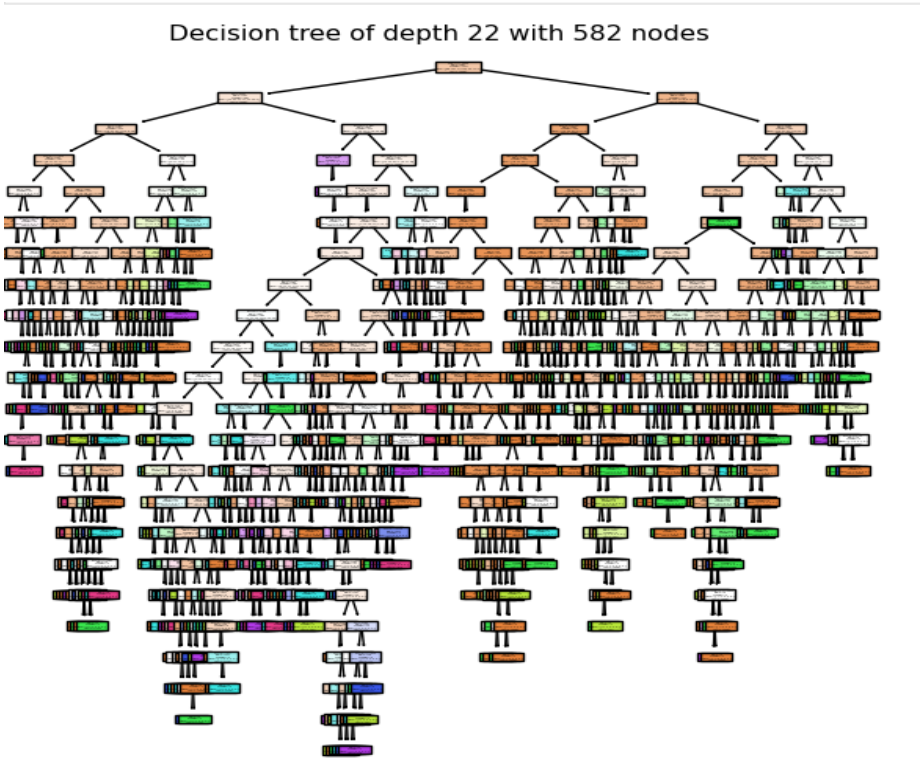
Prednosti su: laka primena, laka razumljivost, vizuelizacija u hijerarhijskoj strukturi, pružaju meru značajnosti atributa, čime pokazuju koje atribute vrednuju više propocijalno njihovoj relevantnosti pri klasifikaciji, što pomaže prilikom odabira atributa.

Za ovaj skup podataka stabla nisu najbolji izbor jer stabla imaju problema sa podacima koji nisu balansirani. Ovde je pokušano raditi klasifikaciju nakon balansiranja podataka, konkretno, korišćenjem nekih tehnika uzorkovanja I postavljanjem atributa *class-weight* na “balanced”, ali su rezultati bili minimalno poboljšani. Pokušano je I skaliranje nakon train-test podele skupova, ali nije urodilo plodom.

Problem sa stablima je još I to što ne barata baš najbolje sa numeričkim podacima, koji su ovde zastupljeni svuda. Ima problem pri odabiru najbolje granice da podelu vrednosti. Pokušano je postavljanje odgovarajućih (najboljih) hiperparametara, poput promene dubine stabla (u rasponu od 3 do 40), promene vrednosti pri evaluaciji upita (gini, entropija), I drugih. Koriscenjem hiperparametara je dobijeno poboljšsanje modela od svega 5-10% u zavisnosti od klase.

Na kraju su postavljeni hiperparametri:

- Criterion: entropy
- Max_depth: 28
- Max_features: 3



Slika 1: prikaz celokupnog stabla odlučivanja

Kako bi se odstranilo višak nepotrebnih (irelevantnih, redudandnih) atributa, korišćena je tehnika *PCA*, koja smanjuje broj atributa pri čemu je gubitak kvaliteta klasifikacije minimalan. Pokazalo se da sa smanjenjem broja atributa kvalitet algoritama opada proporcijalno, tako da se odustalo od te ideje.

KNeighborsClassifier Ovaj algoritam se oslanja na slicnost izmedju instanci kako bi donosio odluke o klasifikaciji (*"ako ide kao patka, kvače kao patka I izgleda kao patka, onda je najverovatnije to patka"*). Klasifikacija se obavlja pomoću bliskosti izmedju instance koja se predvidja i njenih K najblizih suseda, pri čemu je K parameter koji korisnik zadaje. Bliskost se obicno meri uspomoć nekih metrika, poput euklidskog rastojanja.

Klasifikacija se vrsi na osnovu slicnosti izmedju instance koja se predvidja i njenih najblizih suseda. Ovo je posebno korisno u problemima gde su slicne instance sklonije pripadati istoj klasi, odakle I motivacija za upotrebom ovog algoritma, stim što se nije pokazao u najboljem svetlu.

Ovaj algoritam se moze primeniti na razlicite vrste podataka, ali je neophonodo da ti podaci budu skalirani, ne bi li neki neskalirani atribut previse preovladao. Usled toga se podaci nakon podele na trening I test skupove skaliraju kako bi bili istog raspona. Pokušano je podešavanje hiperparametara u vidu određivanja K za broj najbližih suseda uzetih u razmatranje, I postavljanje parametra "weighted" na distance, kako bi bliži susedi imali prednost u odnosu na one dalje. Ali je primećeno da će se samo sa porastom broja K podaci preprilagođavati kategoriji kojih ima najviše I to će davati lažnu sliku o preciznosti. Uprkos tome, algoritam je davao bolje rezultate od stabala odlučivanja. Na kraju su odabrani hiperparametri: Algorithm: auto, n_neighbours: 18, weights: uniform.

MLPClassifier Algoritam se zasniva na neuronskim mrezama. MLPClassifier koristi metodu propagacije unapred, kako bi preneo ulazne podatke kroz mrežu i generisao izlazne vrednosti. Zatim se koristi metoda propagacije unazad kako bi se prilagodili parametri mreze da budu što bolji, na osnovu razlike izmedju predvidjenih izlaza i stvarnih vrednosti ciljnih klasa. Ovaj proces se iterativno ponavlja kako bi se minimizovala greska i optimizovali parametri mreze.

Algoritam uspešno modeluje kompleksne veze između ulaznih osobina i ciljnih klasa. Samim tim je I odličan prilikom rada sa numeričkim vrednostima, ima sposobnost da nauči i predstavi nelinearne veze koje drugi linearni modeli ne mogu. Algoritam je takođe zadovoljavajuće otporan na šumove, ali to ovde svakako ne bi predstavljalo nikakav problem.

Prilikom postavljanja nekih hiperparametara poput funkcija aktivacije, dubinu I širinu neurona, došlo je do znatnih poboljšanja modela, čime se došlo do rezultata boljih nego korišćenjem prethodnih tehnika, što je u neku ruku I očekivano shodno mogućnostima koje neuronske mreže pružaju.

Dosad pomenuti algoritmi su koristili atribute kako bi klasifikovali odgovarajuće klase, ali kao pokušaj da se algoritmi poboljšaju korišćene su čak I klase (supstance) kao vid atributa prilikom klasifikacije. Motivacija za to je da je možda lakše naći koje se supstance koriste ako se zna I istorija korišćenja ostalih supstanci. Klase su diskretizovane, potom I adekvatno skalirane na [0,1]. Rezultati ipak nisu urodili plodom, sa povećanjem atributa sa 12 na 30 je minimalno poboljšan ishod. Odabrani hiperparametri su: activation: tanh, learning_rate: constant, hidden_layer_sizes: (50, 30, 20), max_iter = 20.

RandomForestClassifier Algoritam se zasniva na konceptu ansambl metoda i kombinuje više stabala odlučivanja kako bi donosio odluke o klasifikaciji. Radi tako što kombinovanjem predviđanja svakog stabla dolazi od rezultata. Odluka se obično vrši “većinskim glasanjem”, gde se klasa koja je najčešća među svim stablima smatra konačnim predviđanjem. Svako stablo se trenira na različitom podskupu podataka dobijenih metodom slučajnog izbora, što omogućava da ne dođe do preprilagođavanja datom skupu.

Ovaj algoritam ima nekoliko prednosti, visoku tačnost predviđanja, otpornost na preprilagođavanje. Paralelizacija treninga stabala omogućava brzu obradu velikog broja instanci ili osobina, čime se štedi vreme. Dizbalansirane klase ne predstavljaju problem, što je u značajnoj meri doprinelo prilikom klasifikacije ovih podataka. Rezultati dobijeni su znatno bolji od svih malopređašnje navedenih algoritama.

Logistic Regression – algoritam odbrađivan na drugim kursevima. Logistička regresija radi najbolje sa numeričkim podacima, kakvi već I jesu. Prikazuje zadovoljavajuće rezultate. Radi po principu da uspomoć datih podataka traži odgovarajuću jednačinu kojom bi mogao da što verodostojnije prikaže model. Pri čemu sam unapred skalira podatke na raspon [-1, 1].

KLASTEROVANJE

Podaci sami po sebi nisu napravljeni za klasterovanje, ali će se svakako pokušati da se otkrije nešto više o njima grupisanjem po određenim parametrima u nadi da ćemo dobrim rezultatima dobiti neophodne informacije kako bismo unapredili klasifikaciju. Ako su klasteri jasno definisani i ako bismo klase pravili po klasterima, verovatno je da bismo poboljšali predviđanja modela.

Skup podataka se obrađuje tako što će se *undersemplovati* kako bi se izbalansirale klase, ali i da bi se ubrzao proces treniranja modela, pošto je klasterovanje znatno skuplja operacija od klasifikacije. To ovde nije rađeno jer se uzorkovanjem dobija previse malo instanci, kojih I inače nema previse. Klase će biti iste kao za problem klasifikacije.

Glavno pitanje je, da li je podela na 7 klasa (pošto postoji 7 kategorija konzumacija supstanci) dobra ili možda postoji neka druga konfiguracija tih klasa koja bi dala bolje rezultate. Za reprezentaciju dobijenih rezultata biće korisceni atributi *Ascore* i *Oscore*. Razlog zašto se oni koriste, je taj što jedini daju bilo kakav pristojan grafički prikaz podataka nakon klasterovanja. Do toga se došlo korišćenjem *pairplot* funkcije iz biblioteke *seaborn*.

Modeli korišćeni za klasterovanje su: *KMeans*, *AgglomerativeClustering*, *DBSCAN*

KMeans Algoritam za cilj ima pronalaženje K klastera, pri čemu je K unapred odredjen broj klastera zadat od strane korisnika. Koristi se zbog jednostavnosti implementacije i brzine izvršavanja. Međutim, treba imati na umu da KMeans može biti osetljiv na inicijalnu postavku centroida klastera, što čini odabir parametra K problematičnim. Kako bi se to rešilo, uvodi se pomocna funkcija *plot search* koja ce tražiti idealni odabir parametra za zadati opseg (0, k]. Za meru evaluacije modela koristi se *silhouette score*. Najbolji pronadjen parametar je k = 7, što je nekako i logično obzirom da ima 7 stepena konzumacije supstanci. Iako najbolji, svakako ne prikazuje ocenu koja ističe neku preteranu zanimljivost.

Uvedena je i još jedna funkcija, preuzeta sa scikit-learn sajta, koja crta dijagram atributa senki. Kao što se I moglo očekivati, klasterovanje nije pokazalo zanimljive rezulate, mimo toga koji je odabir vrednosti za parametar K.

AgglomerativeClustering Algoritam pocinje sa svakom instancom podataka kao jednim klasterom. Zatim se iterativno spajaju parovi klastera koji su najbliži jedan drugom na osnovu definisane mere udaljenosti, pri cemu se formira hijerarhijska struktura klastera. Prednosti aglomerativnog klasterovanja uključuju jednostavnost implementacije i mogućnost rukovanja različitim merama udaljenosti, omugućava I odabir broja klastera.

Na kraju procesa, rezultat je predstavljen *dendrogramom* koji vizualizuje hijerarhijsko povezivanje klastera. Od pomoćnih funkcija koriste se, vec opisane, *plot search* i funkcija za crtanje *silhouette diagram*-a. Rezultati nisu mnogo bolji, lose se bira hijerarhija što se I može zaključiti na osnovu crteža dendrograma.

DBSCAN Algoritam je algoritam klasterovanja zasnovan na gustini koji se koristi za grupisanje podataka na osnovu njihove prostorne raspodele. Ovaj algoritam je sposoban da identifikuje klasterovanje u skupu podataka bez potrebe za unapred definisanim brojem

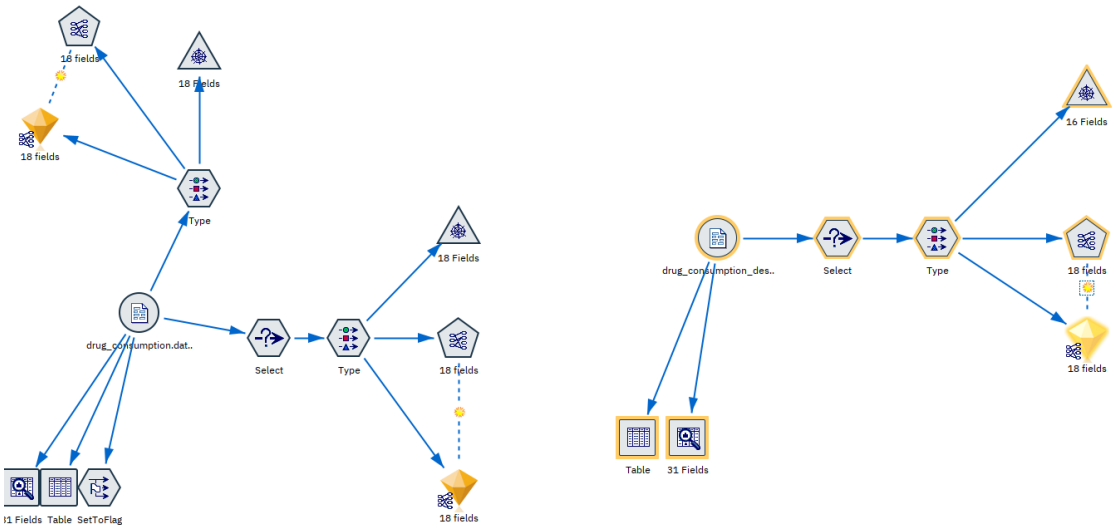
klastera. DBSCAN ima nekoliko prednosti, sposobnost otkrivanja klastera proizvoljnih oblika, otpornost na šum i sposobnost rukovanja promenljivim gustinama klastera. Od pomoćnih funkcija je korišćen samo *plot search*. Rezultati su najlošiji od svih metoda klaterovanja, uzrok tome je taj što algoritmi klasterovanja zasnovani na gustini, veoma loše skaliraju sa povećanjem broja dimenzija (atributa).

PRAVILA PRIDRUŽIVANJA

Apriori Algoritam funkcioniše na principu pronalaženja čestih uzoraka u podacima. Asocijativna pravila su oblika "A sledi B" (A -> B), gde A i B predstavljaju skupove stavki, a strelica predstavlja implikaciju. Ova pravila se zasnivaju na merama podrške i pouzdanosti, koje se koriste za procenu značajnosti i pouzdanosti pravila. Minimalne vrednosti za ove mere se zadaju unapred.

Modelovanje i obrada skupa podatka su rađeni pomoću IBM-ovog alata **SPSS**. Obzirom da ovaj algoritam radi samo sa kategoričkim atributima, prvobitno su odbačeni numerički atributi, tako da se radi samo sa klasama, ili se numerički atributi transormišu u odgovarajuće kategoričke, što je već rađeno prilikom pretprocesiranja podataka, tako da se I taj skup posebno ispitivao ne bi li se dobilo nesto vise ovim algoritmom.

Minimalna podrška je postavljena na 15%, dok je minimalna pouzdanost 70%. Rađeno je nad dva tipa podataka, sa I bez atributa. Maksimalan broj atributa koji se uzima u obzir prilikom generisanja pravila je postavljen na 10. Shema toka izgleda ovako:



Slika 2: shema u SPSS-u

3. Rezultati

REZULTATI KLASIFIKACIJE

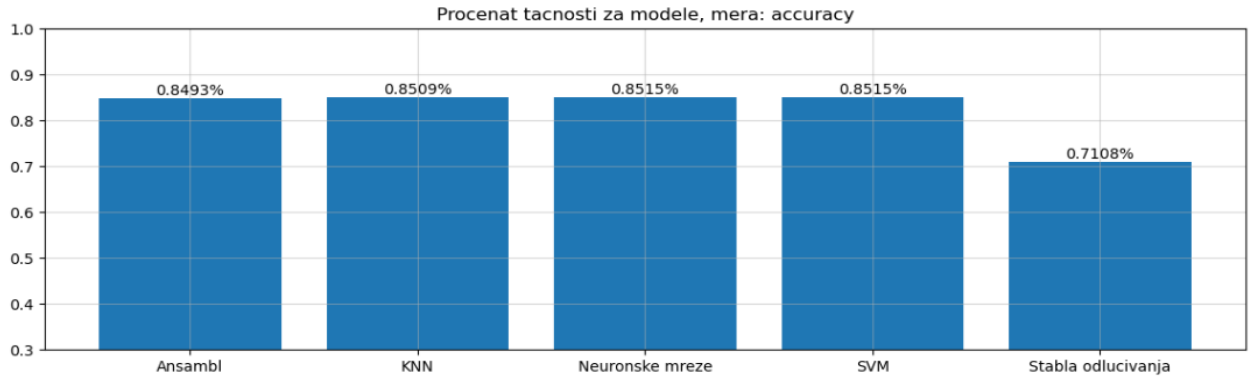
Mere ocena su dobijeni uspomoć tehnike *Cross validation*. Od svih istreniranih modela odabrani su oni koji su imali najviše uspeha prilikom primene nad podacima. Modeli sa adekvatnim hiperparametrima su izdvojeni iz odgovarajućih fajlova, nakon čega su učitani u datoteku prilokom provere rezultata. Algoritmi čiji se modeli upoređuju su:

- DecisionTreeClassifier
- RandomForestClassifier
- KNN
- MLPClassifier
- SVC

Koriscene mere ocene za evaluaciju modela su:

- accuracy
- precision
- recall
- f1-score
- vreme treniranja modela

Accuracy:

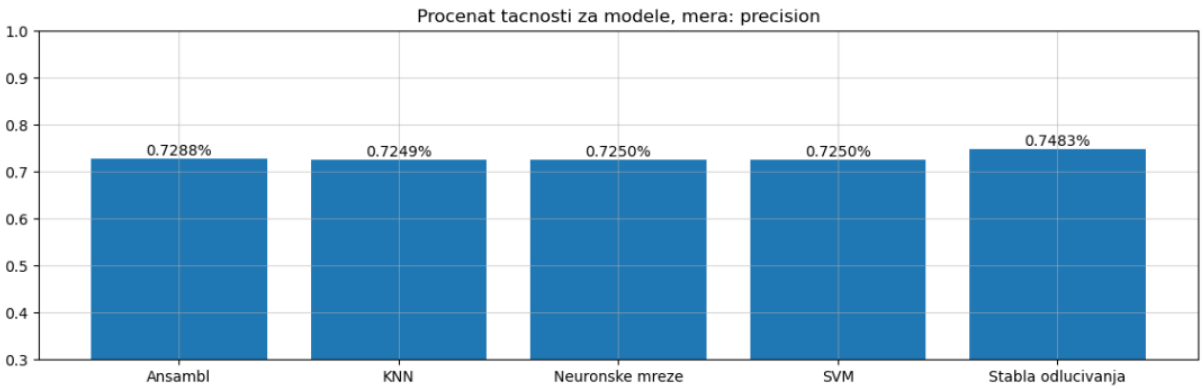


Slika 3: accuracy modela

Svi modeli daju zadovoljavajuće rezultate. Možda jedino stablo odlučivanja daje malo lošije, ali se to I da očekivati, obzirom da loše radi sa numeričkim podacima. Ostali modeli daju manje ili više iste rezultate.

Precision:

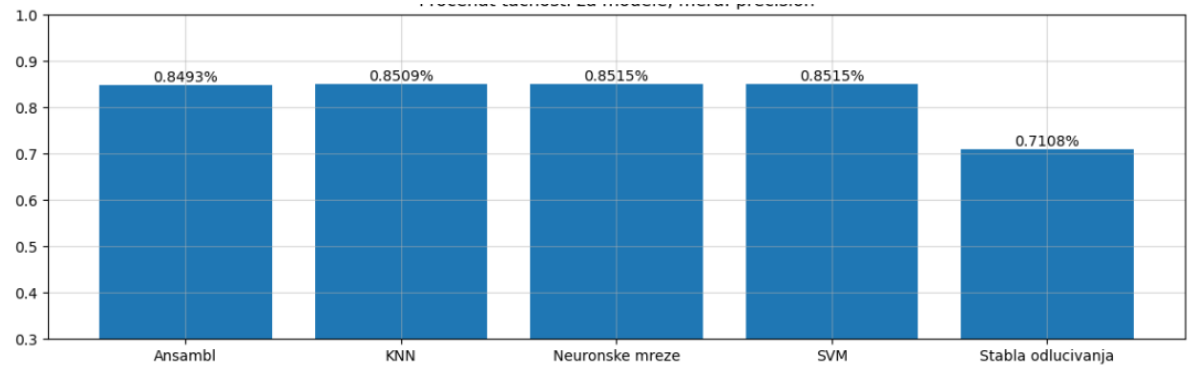
Precision je mera koja predstavlja odnos tačno klasifikovanih pozitivnih instanci prema zbiru tačno klasifikovanih pozitivnih I lažno klasifikovanih pozitivnih instanci.



Slika 4: precision score modela

Recall score:

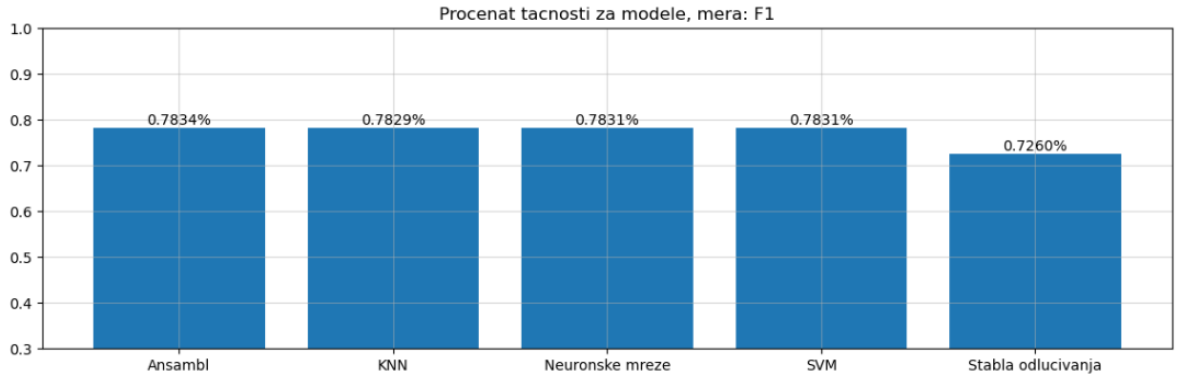
Recall predstavlja odnos tačno pozitivno klasifikovanih instanci I zbira tačno pozitivno klasifikovanih i lažno negativno klasifikovanih instanci. Recall meri sposobnost modela da pravilno identifikuje sve pozitivne instance. Daje slične rezultate kao I accuracy. Ako su rezultati preciznosti i odziva jednaki za model, to znači da je model postigao iste rezultate prilikom tacnog identifikovanja pozitivnih instanci i ukupne tacnosti svojih predviđanja.



Slika 5: recall score modela

F1 score:

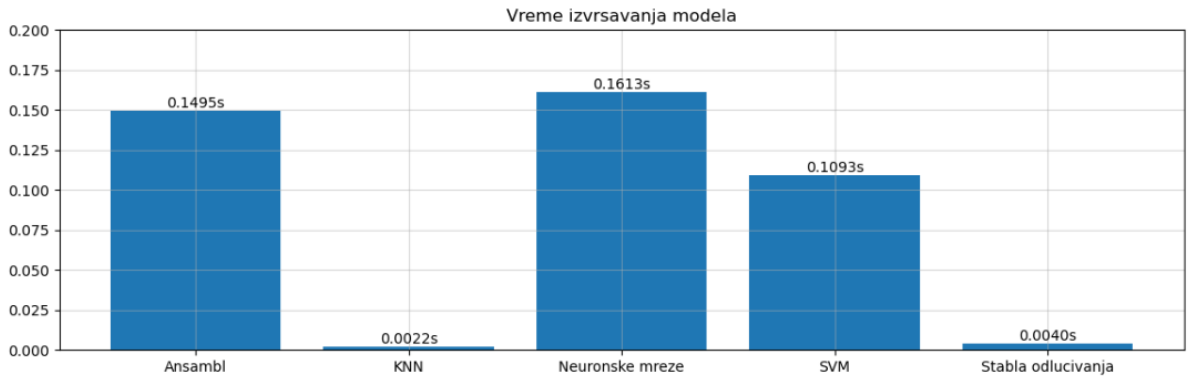
F1-score predstavlja harmoničnu sredinu preciznosti i odziva. Kombinuje informacije o tačnosti modela u klasifikaciji pozitivnih i negativnih instanci.



Slika 6: f1 score modela

Zaključak je da od svih modela RandomForestClassifier pokazuje najbolje rezultate, mada je sličan sa ostalima, osim možda sa stablima odlučivanja, što je I očekivano. Ono po čemu se znatno razlikuju jeste vreme potrebno za treniranje.

Vreme treniranja modela



Slika 7: vreme treniranja modela

Odavde se vidi da su ansambl metoda I metoda neuronske mreže znatno sporije od ostalih, što je I očekivano, obzirom da rade tako da žrtvuju brzinu zarad što bolje tačnosti.

REZULTATI KLASTEROVANJA

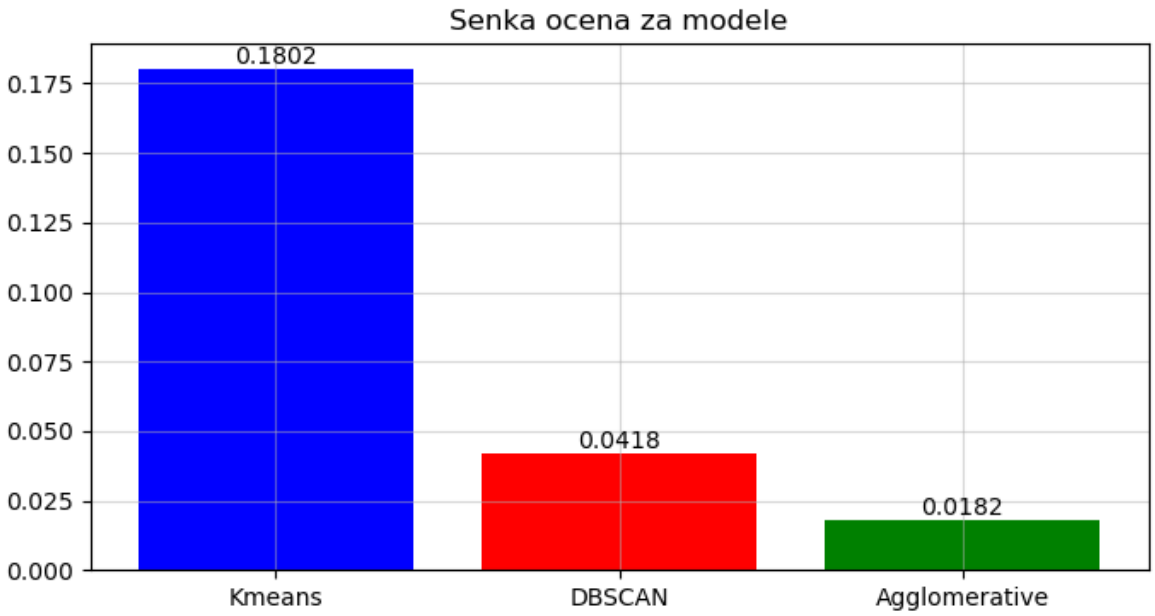
Kao I za klasifikaciju, modeli sa svojim hiperparametrima su učitani u posebnu datoteku za ocenjivanje kvaliteta modela. Korišćene mere za evaluaciju modela klasterovanja su:

- Silhouette score
- Dunn index
- Intra-cluster distances

Razmatrani modeli korišćeni za klasterovanje su:

- KMeans
- AgglomerativeClustering
- DBSCAN

Ocena senki je metrika koja meri koliko dobro je svaki objekat dodeljen odgovarajućem klasteru, uzimajući u obzir gustinu i razdvajanje klastera. Vrednosti ocene senki se krecu u opsegu od -1 do 1. Veće vrednosti ukazuju na bolje klasterovanje, pri cemu vrednost bliska 1 ukazuje na dobro definisane i odvojene klastera, dok vrednost bliska -1 ukazuje na objekte koji su pogrešno dodeljeni klasterima.

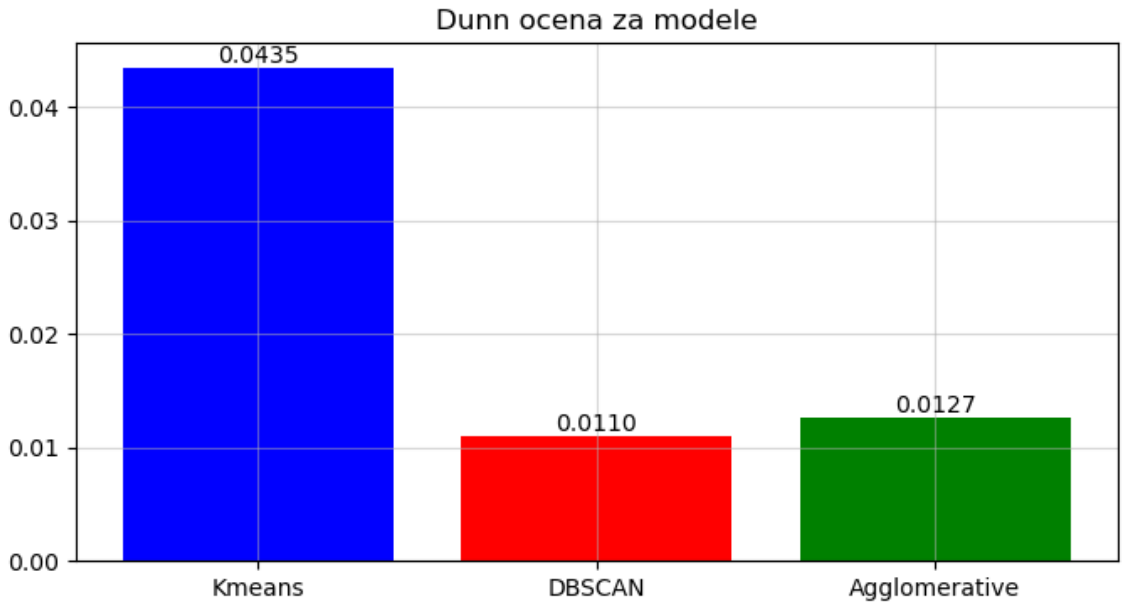


Slika 8: senka ocene modela

Kmeans daje najbolje rezultate, verovatno jer je najbolje birao broj klastera.

DUNN indeks

Dunn index je mera ocene koja meri razdvajanje između klastera i gustinu objekata unutar klastera. Visoke vrednosti Dunn indeksa ukazuju na dobro odvojene klastere i dobru gustinu unutar klastera, niske vrednosti ukazuju na nejasno definisane klastere.

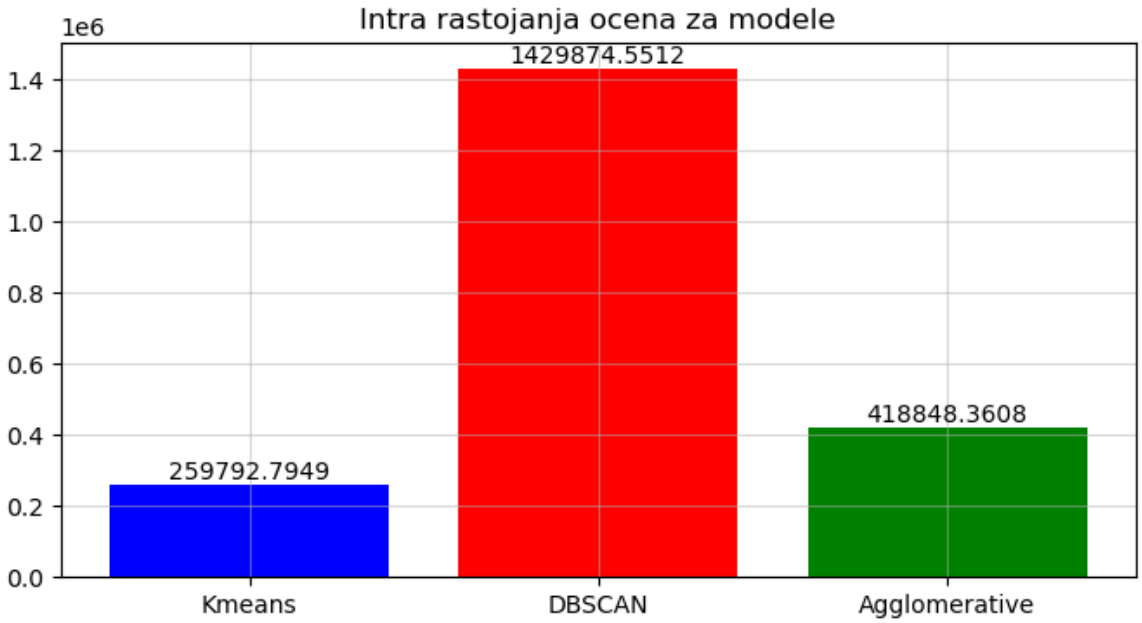


Slika 9: Dunn ocena modela

Ponovo kmeans prikazuje najbolje rezultate, ali su rezultati svakako loši.

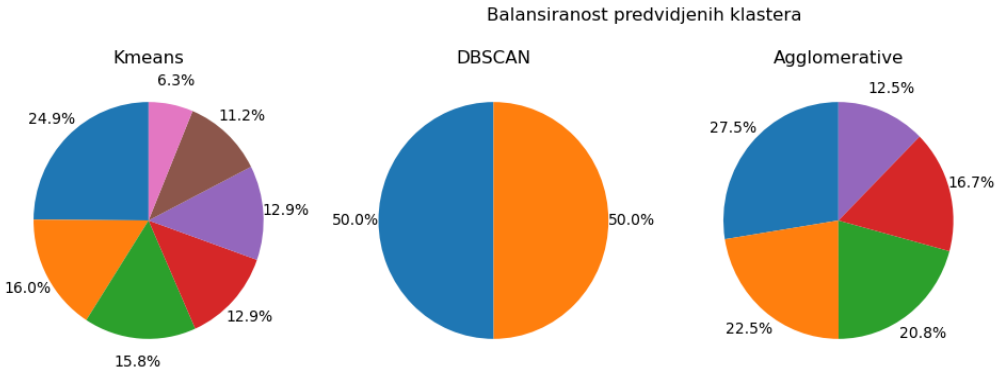
Intra-klaster rastojanje

Intra-klaster rastojanje je mera ocene koja se koristi za merenje prosečne udaljenosti između objekata unutar istog klastera. Ocenjuje kompaktnost i zgusnutost klastera. To radi tako što računa prosečno rastojanje izmedju svih parova objekata unutar istog klastera. Manje vrednosti intra-klaster rastojanja ukazuju na bolje klasterovanje. Naime, ukazuju na manje rastojanje između objekata unutar klastera.



Slika 10: Intra-klaster ocene modela

Ponovo je Kmeans najbolji, stim što je ovog puta I DBSCAN ubedljivo najlošiji. Odakle se može I zaključiti da je Kmeans bio najuspešniji. Pored ovih ocena, rađena je I provera balansiranosti nakon klasterovanja, tj. balansiranost klastera.



Slika 11: Balansiranost klasa ocene modela

Uprkos tome što je DBSCAN najbolje balansiran, usled lose raspodele klastera daje manje zanimljive klastere. Ali se svakako da I videte da je metod hijerahijskog klasterovanja valjano balansirao klase, verovatno stopivši nekoliko stepena konzumacije u jedno.

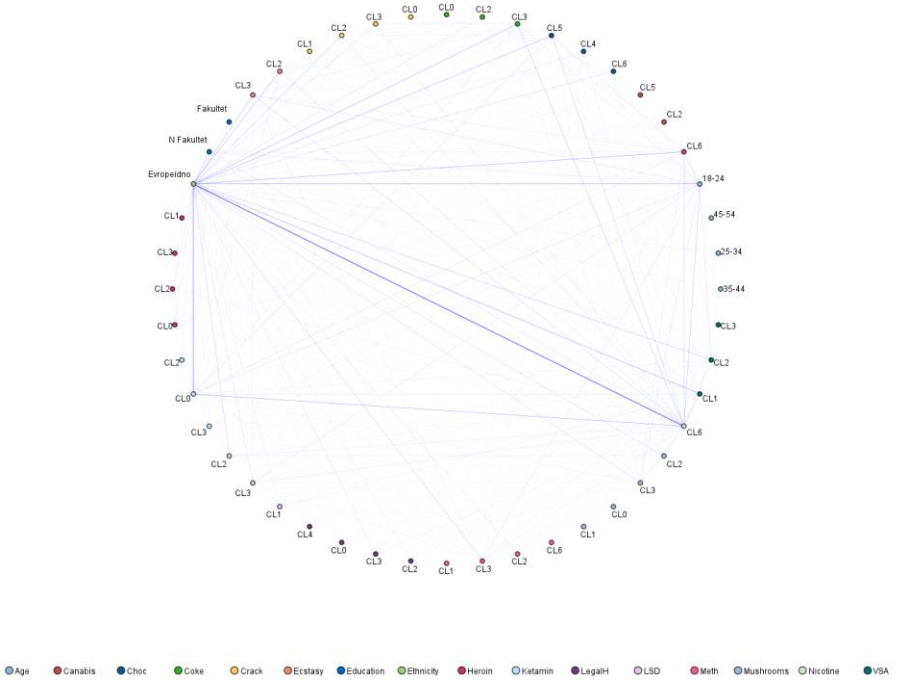
REZULTATI PRAVILA PRIDRUŽIVANJA

Iako podaci nisu namenjeni da se nad njima traže pravila pridruživanja, svakako je nadjeno cetrdesetak pravila. Prikazana su samo ona koja imaju zadovoljavajuć lift factor. Naime, pravila koja imaju lift factor blizu 1 I nisu interesantna, te su izostavljena. Naredna pravila su dobijena korišćenjem svih atributa.

Consequent	Antecedent	Support %	Confidence %	Lift
VSA = CL2	Education = N Fakultet Age = 18-24 Country = USA Ethnicity = Evropeidno	15.663	76.923	2.28
VSA = CL1	Age = 35-44	20.482	94.118	2.232
VSA = CL1	Age = 35-44 Ethnicity = Evropeidno	18.072	93.333	2.213
VSA = CL2	Education = N Fakultet Age = 18-24 Country = USA	18.072	73.333	2.174
VSA = CL2	Education = N Fakultet Age = 18-24 Ethnicity = Evropeidno	20.482	70.588	2.092
VSA = CL1	Country = UK	21.687	77.778	1.844
VSA = CL1	Country = UK Ethnicity = Evropeidno	21.687	77.778	1.844
Canabis = CL6	Age = 18-24 Country = USA Ethnicity = Evropeidno	30.12	72.0	1.66
Amyl = CL0	Age = 18-24 Country = USA	36.145	80.0	1.581
Amyl = CL0	Age = 18-24	45.783	76.316	1.508
Amyl = CL0	Age = 18-24 Country = USA Ethnicity = Evropeidno	30.12	76.0	1.502
Ketamin = CL0	Gender Country = USA	15.663	76.923	1.485
Amyl = CL0	Education = N Fakultet Age = 18-24	22.892	73.684	1.456
Amyl = CL0	Education = N Fakultet Age = 18-24 Country = USA	18.072	73.333	1.449

Slika 12: Pravila pridruživanja

Grafik mreže ovih rezultata izgleda ovako:



Slika 12: Grafik mreže

Postoje neka pravila, jedino koje se znatno ističe jeste da pripadnici evropeidne rase svakodnevno konzumiraju nikotin, što je pravilo koje I nije od nekog značaja.

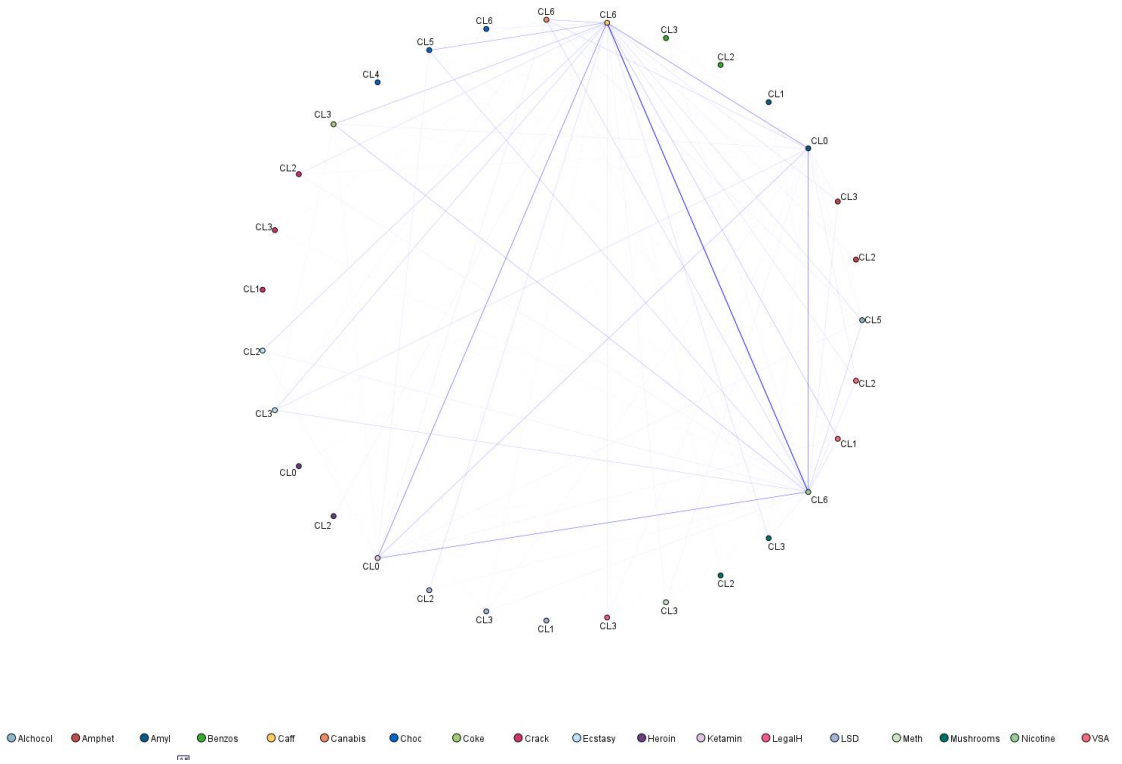
Pravila nastala priliko korišćenja samo klasa:

Consequent	Antecedent	Support %	Confidence %	Lift
Mushrooms = CL1	LSD = CL1 VSA = CL1	18.072	73.333	5.072
Meth = CL1	Crack = CL1 VSA = CL1	18.072	80.0	4.743
Crack = CL1	Meth = CL1 VSA = CL1	15.663	92.308	4.507
Meth = CL1	Crack = CL1	20.482	70.588	4.185
Crack = CL1	Meth = CL1	16.867	85.714	4.185
LSD = CL1	Meth = CL1	16.867	71.429	3.294
Amyl = CL1	Crack = CL1 VSA = CL1	18.072	73.333	3.204
Amyl = CL1	Crack = CL1	20.482	70.588	3.084
Coke = CL2	Canabis = CL2	15.663	76.923	2.902
LSD = CL3	Ketamin = CL3	18.072	73.333	2.536
Ecstasy = CL2	Coke = CL2 Caff = CL6	16.867	78.571	2.329
Coke = CL3	Amphet = CL3 Amyl = CL0 Nicotine = CL6	15.663	84.615	2.266
VSA = CL1	Amyl = CL1 Caff = CL6	18.072	93.333	2.213
VSA = CL1	Meth = CL1	16.867	92.857	2.202
Coke = CL3	Ecstasy = CL3 Amyl = CL0 Nicotine = CL6	18.072	80.0	2.142
VSA = CL1	Amyl = CL1	22.892	89.474	2.122
Ecstasy = CL2	Amphet = CL2 Ketamin = CL0	16.867	71.429	2.117
VSA = CL2	Ecstasy = CL2 Amyl = CL0	16.867	71.429	2.117
Coke = CL3	Heroin = CL3 Caff = CL6	16.867	78.571	2.104
Amphet = CL3	Ketamin = CL3	18.072	73.333	2.099
VSA = CL1	Crack = CL1	20.482	88.235	2.092
Coke = CL3	Canabis = CL5	15.663	76.923	2.06
Coke = CL3	Crack = CL3 Nicotine = CL6 Caff = CL6	15.663	76.923	2.06
Coke = CL3	LSD = CL3 Amyl = CL0 Nicotine = CL6	15.663	76.923	2.06

Slika 13: Pravila pridruživanja

Korišćenjem samo klasa je pronađeno mnogo više pravila (skoro 400), među kojima je znatno visok lift. Što ukazuje na to da ukoliko neko konzumira neku supstancu veće su šanse da će još neku, ili preciznije, postoji tendencija da nikad nije samo jedna.

Grafik mreže ovih rezultata izgleda ovako:



Slika 14: Grafik mreže

I ovde se na grafiku ne uočavaju neka preterano zanimljiva pravila. Najviše je istaknuto da svakodnevni konzumatori nikotina konzumiraju I kofein svakodnevno, što I nije neko novo otkriće.

4. Zaključak

Nakon primene različitih tehnika istraživanja podataka i mašinskog učenja na datom skupu podataka, istrenirano je oko 10 modela i ispitane su karakteristike I ocene svakog od njih. Što se skupa podataka tiče, podaci su možda preopširni brojem klasa, a pritom imaju mali broj instanci tako je teško doći do nekog stoprocentno tačnog zaključka.

Za klasifikaciju je RandomForestClassifier najbolji model za dati skup podataka. Za klasterovanje postoji vise mogućih podela podataka I mogu se dobiti valjane ocene I sa više različitih metoda. Pravila pridruživanja su samo predstavila očigledno.