ISTRAŽIVANJE SAOBRAĆAJNIH NESREĆA U FRANCUSKOJ

Seminarski Rad u okviru kursa
Istraživanje Podataka 1
Matematički fakultet

Aleksa Knežević i Darko Nešković
mi16009@alas.matf.bg.ac.rs
mi16208@alas.matf.bg.ac.rs

13.08.2019

Ukratko o seminarskom

Tema ovog seminarskog jeste istraživanje skupa podataka o saobraćajnim nesrećama u Francuskoj. Analiziranjem ravnih podataka, naš cilj je bio da se uoče zanimljivosti unutar skupa podataka i da se korišćenjem algoritama klasterovanja nađu veze između instanci samog skupa. Atribut kome je posvećena najveća pažnja u ovom seminarskom jeste atribut "praznik", odnosno želeli smo da uradimo istraživanje za dane u godini na koje su bili praznici. To smo radili u kombinaciji sa drugim atributima, koje smo našli zanimljivim ili neophodnim za smislene rezultate. U daljem tekstu, objašnjen je način pripremanja podataka, obrade istih, kao i rezultati primene algoritama obrađenih na kursu na te podatke.

Uvod

Klasterovanje, odnosno klaster analiza predstavlja proces obrade podataka, pri čemu se podaci grupišu u klastere. Grupisanje se vrši na način da se podaci, odnosno instance koje su slični grupišu u isti klaster, dok se podaci koji se više međusobno razlikuju grupišu u razdvojene klastere. Grupisanje se rešava preko rastojanja, koje se može računati na različite načine.

Ovi algoritmi predstavljaju algoritme sa nenadgledanim učenjem, jer ne postoji ciljni atribut kao što smo imali kod klasifikacije. Algoritmi koje smo mi učili na ovom kursu, i iskoristili prilikom izrade ovog seminarskog su:

- K-sredina
- Kohonen
- DBSCAN
- Hijerarhijsko klasterovanje

Kohonen i K-sredina su odrađeni u alatu IBM SPSS Modeler, dok su DBSCAN i Hijerarhijsko klasterovanje odrađeni u programskom jeziku Python primenom biblioteke Scikit-learn.

Nešto o skupu podataka

Korišćeni skup podataka zove se Accidents in France from 2005 to 2016 i može se naći na Kaggle sajtu: https://www.kaggle.com/ahmedlahlou/accidents-in-france-from-2005-to-2016#users.csv
Nakon izdvajanja relevantnih podataka sastoji se od ≈80000 slogova i 19 atributa, od kojih su korišćeni samo oni koji su značajni sa našu temu.

Pretprocesiranje podataka

Za uspešan rad nad podacima koje smo dobili, bilo je potrebno prvo iste i pripremiti. Prilikom učitavanja tabela koje su bile u .csv formatu, primetili smo da tabela characteristics.csv ne može korektno da se učita u SPSS Modeler, jer delimiter "," se pojavljuje i u string atributu "Naziv ulice".
Rešavanje tog problema svelo se na pisanje skripte u Python-u gde smo preko Regex-a našli zareze iz tih atributa i obrisali ih. Taj atribut nam svakako nije bio od značaja u kasnijoj analizi podataka, ali smo morali da ga sredimo svakako, radi učitavanja tabele.
Nakon što smo učitali tabele, filtrirali smo iste. Odnosno, odbacili smo atribute koji nam nisu bili potrebni iz tabela, pri čemu smo uvideli da u tabeli holidays.csv ne postoji atribut id_nesreće po kom smo želeli da spajamo tabele, ali postoji atribut datum. Tu smo naišli na problem, jer atribut datum nismo imali u ostalim tabelama, ali u tabeli characteristics.csv smo imali godinu, mesec i dan, od kojih smo načinili novi atribut datum. Sve tabele osim tabele holidays.csv smo spojili po atributu id_nesreće. Nakon toga smo novodobijenu tabelu spojili sa holidays.csv preko novonapravljenog atributa datum. Iz nje smo izbacili sve podatke koje imaju null vrednosti, to smo uradili jer su činili jako mali procenat podataka (bilo ih je svega 2000 u bazi od 80000 instanci, oko 2,5%).

Takodje, podaci koje smo imali, bili su zapisani tako da su u brojevnom obliku, ali brojevi sami po sebi nose značenje, tako da primenom klaster algoritama ne bi dobili korektne rezultate. Na primer, broj 1 nije bio logički "bliži" broju 2 od broja 4. Zbog toga smo sve korišćene atribute morali da binarizujemo, ne bi li imali korektne rezultate.

Obrada podataka u pythonu:

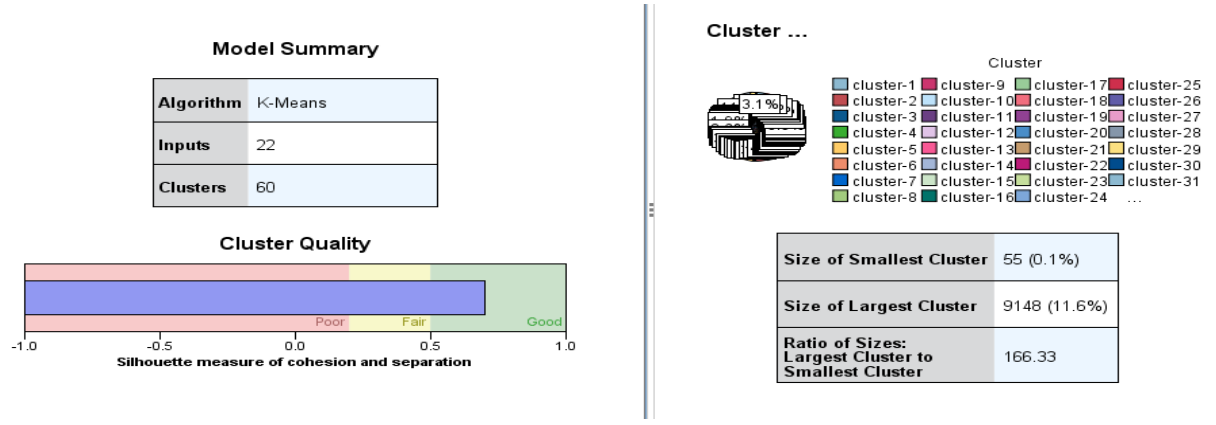
Nakon što smo u SPSS modeleru učitali sve CSV fajlove, I spojili svakih pet tabela u jednu, izveli smo takvu tabelu, te smo je učitali u python-u (jupyter notebook). Uvideli smo da postoje brojni redovi sa NULL vrednostima, i sve redove koji imaju makar jednu
smo uklonili metodom file.dropna() (gde je “file” zapravo ucitan CSV file).
Problem koji smo imali u tom slučaju jeste da python uobičajeno postavlja sve brojevne vrednosti da budu tipa “float”, što nama nije odgovaralo za neke kolone. Taj problem smo rešili metodom astype(int).

Nakon što smo odabrali atribute nad kojima želimo da radimo klasterovanje, morali smo da binarizujemo kolone “osvetljenje” i “stepen_povrede”, a ostale 3 kolone smo skalirali na vrednosti izmedju 0 i 1. Na kraju obrade smo samo spojili binarizovane atribute sa skaliranim, time je naša obrada podataka u python-u završena.

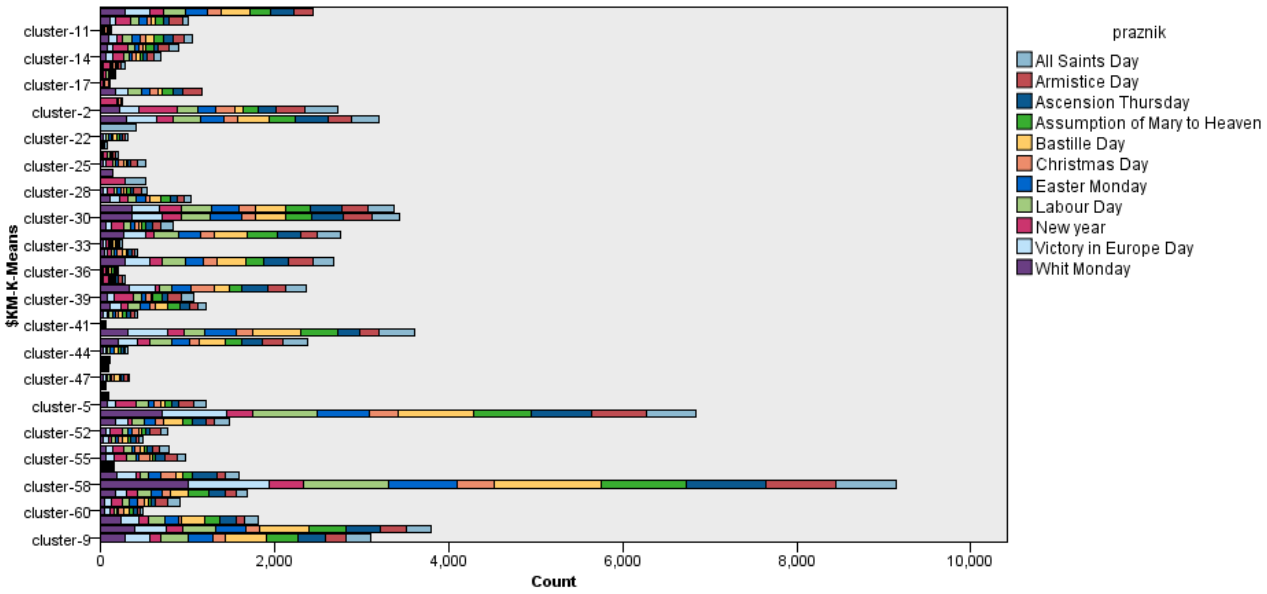
K-sredina

Prvi primenjen algoritam nad podacima bio je algoritam k-sredina, koji je kao parametar zahtevao broj klastera. Za parametar **k** (koji nam je predstavljao broj kalstera) je isprobano više vrednosti, koje nisu dale kao rezultat dobar senka koeficijent. Senka koeficijent je korišćena kao mera kvaliteta klastera.

Nakon isprobavanja više različitih vrednosti za parametar, došli smo do vrednosti 60.



Opšte za k-sredina

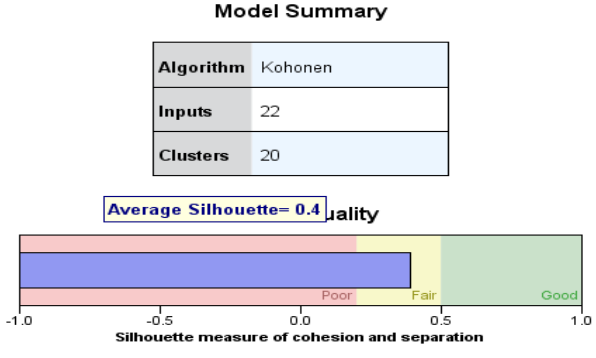


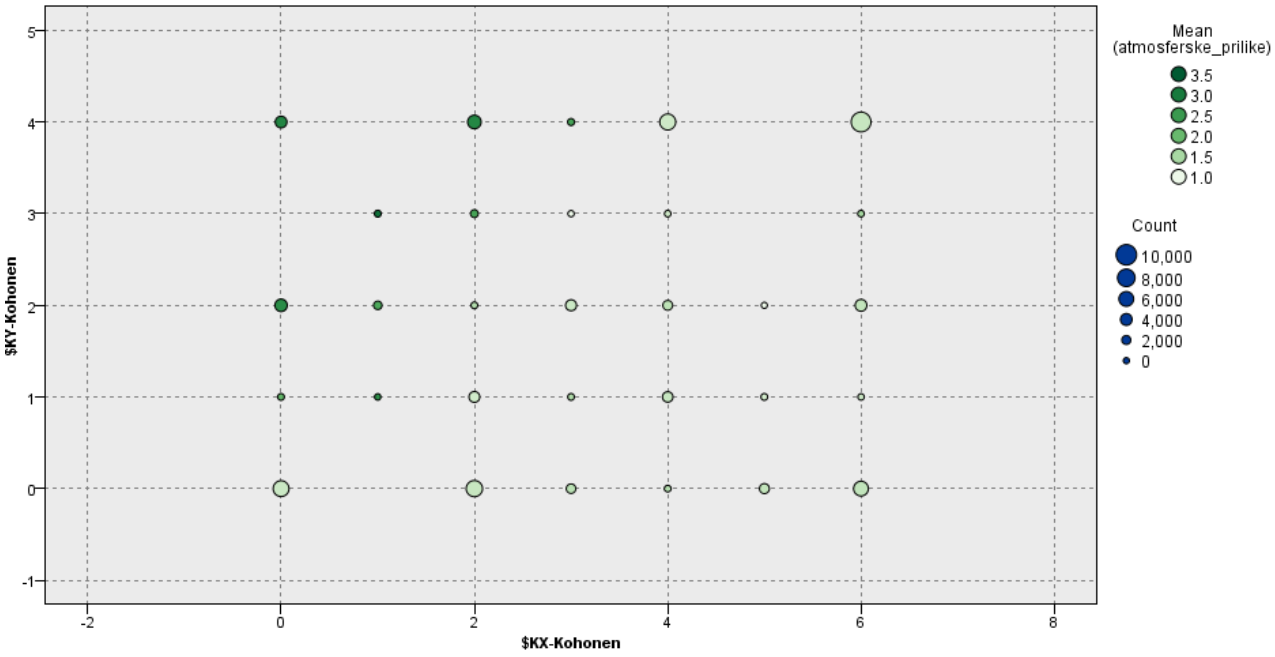
Raspodela klastera prilikom korišćenja algoritma k-sredina

Za **k** = 60 smo dobili 0,7 senku koeficijent, što je bilo zadovoljavajuće.
Zaključak izveden na osnovu isprobavanja mogućnosti K-sredina algoritma je da je potrebno primeniti neku od metoda za pametno odredivanje broja klastera. Jedan od načina bi mogao da bude korišćenjem Pravila lakta (eng: Elbow method), ali taj deo zahteva veliki broj iteracija za
različite vrednosti K. Drugi način je da se posmatra SSE (eng: Sum of Squares Error i da se odabere parametar K za koji je SSE minimalan.

Kohonen

Kohonen algoritam pripada grupi SOM algoritama. SOM stoji za Self Organizing Map. Algoritam je zasnovan na neuralnoj mreži, koja se sama trenira nadgledanim učenjem. Kohonen je zasnovan na modelima koji oponašaju nervni sistem.
Algoritam smo primenili u SPSS modeleru. Algoritam je primenom na atribut "Praznik" i atribute "stepen_povrede", "atmosferski_uslovi" i "tip_sudara" dao senka koeficijent od 0.4 što
nije bilo toliko loše.



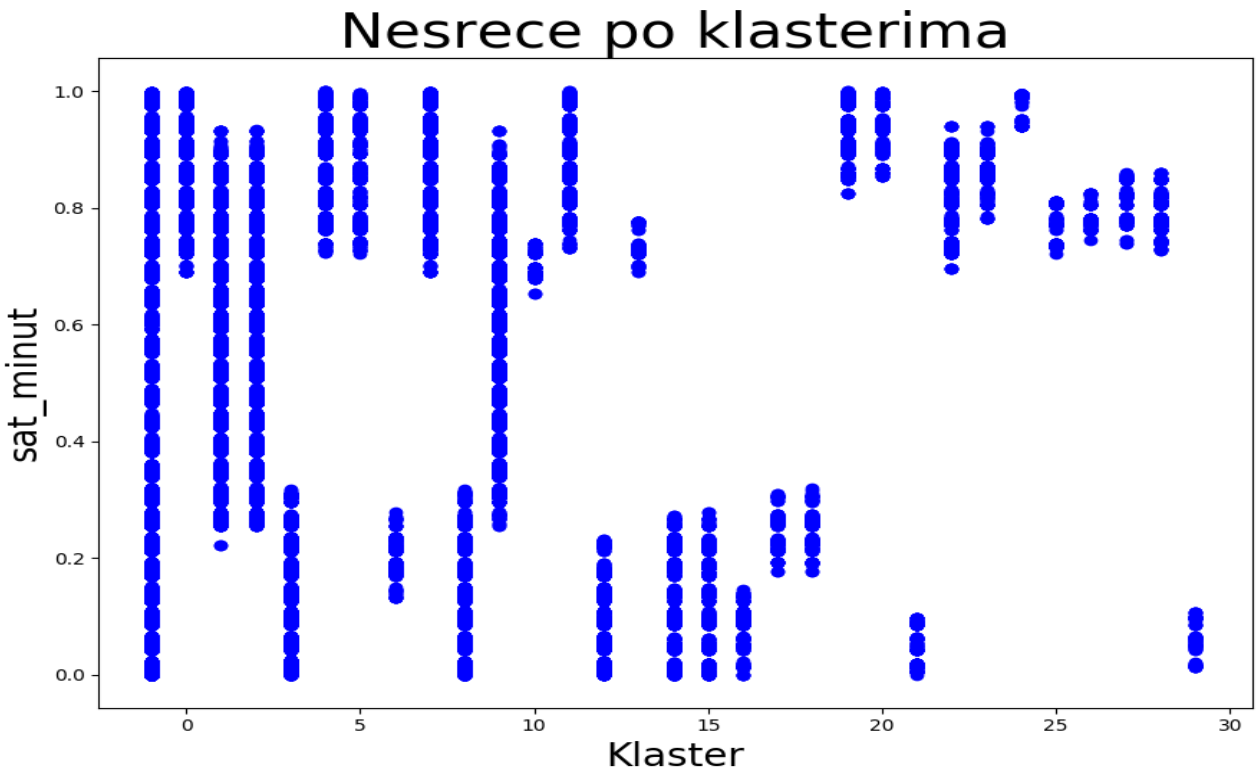


Klasterovanje DBSCAN

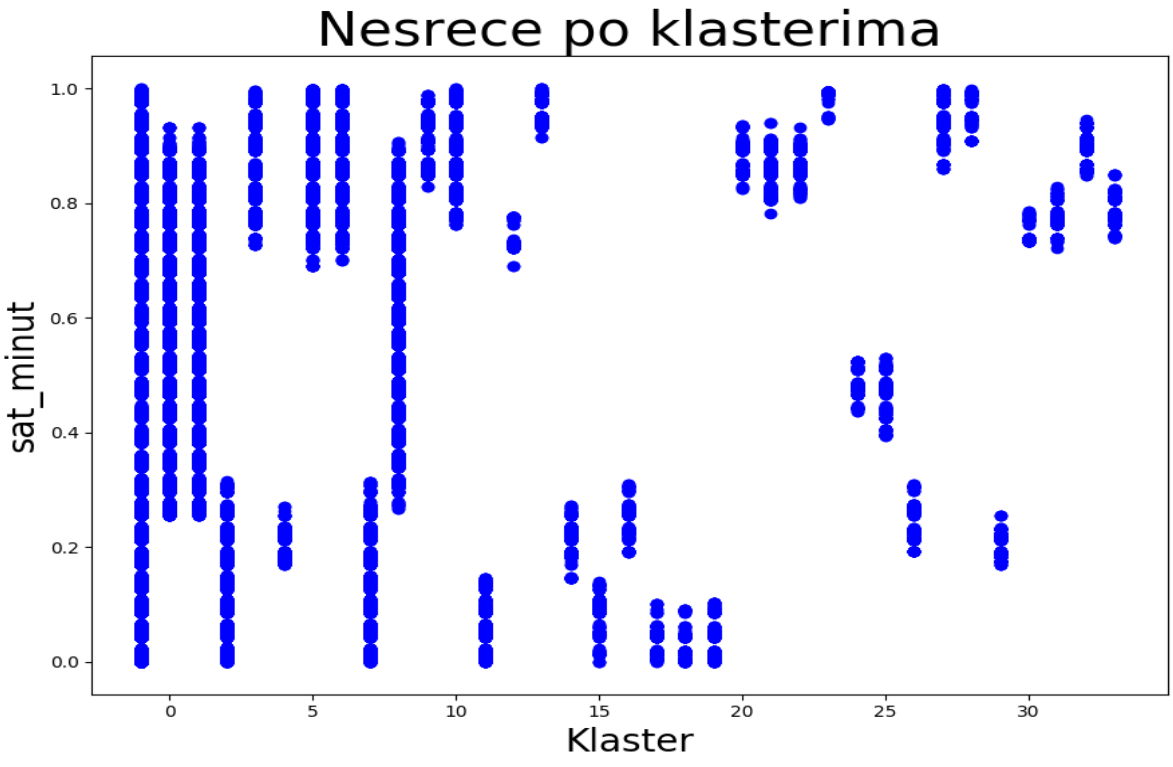
DBscan algoritam za klasterovanje je veoma bitan kada želimo da detektujemo guste klastere, on nije ograničen njihovim oblikom. Ono što je specifično za ovaj algoritam jeste da za razliku od k-najbližih suseda, može odstraniti odudarajuće podatke kao šum. Ovaj algoritam prima dva parametra, prvi je epsilon, drugi T. Pronalaze se sve tačke koje čine jezgro, odnosno u epsilon rastojanju poseduju bar T drugih tačaka.
Kao i u slučaju algoritma K-sredina, ne postoji jasno pravilo kojim možemo da se vodimo prilikom izbora ova dva parametra. Jasno je da u ovom algoritmu ne možemo da biramo broj klastera kao parametar.

U našem istraživanju, uradili smo DBSCAN za 3 različita epsilon, dok nam je broj T uvek bio fiksan (100).

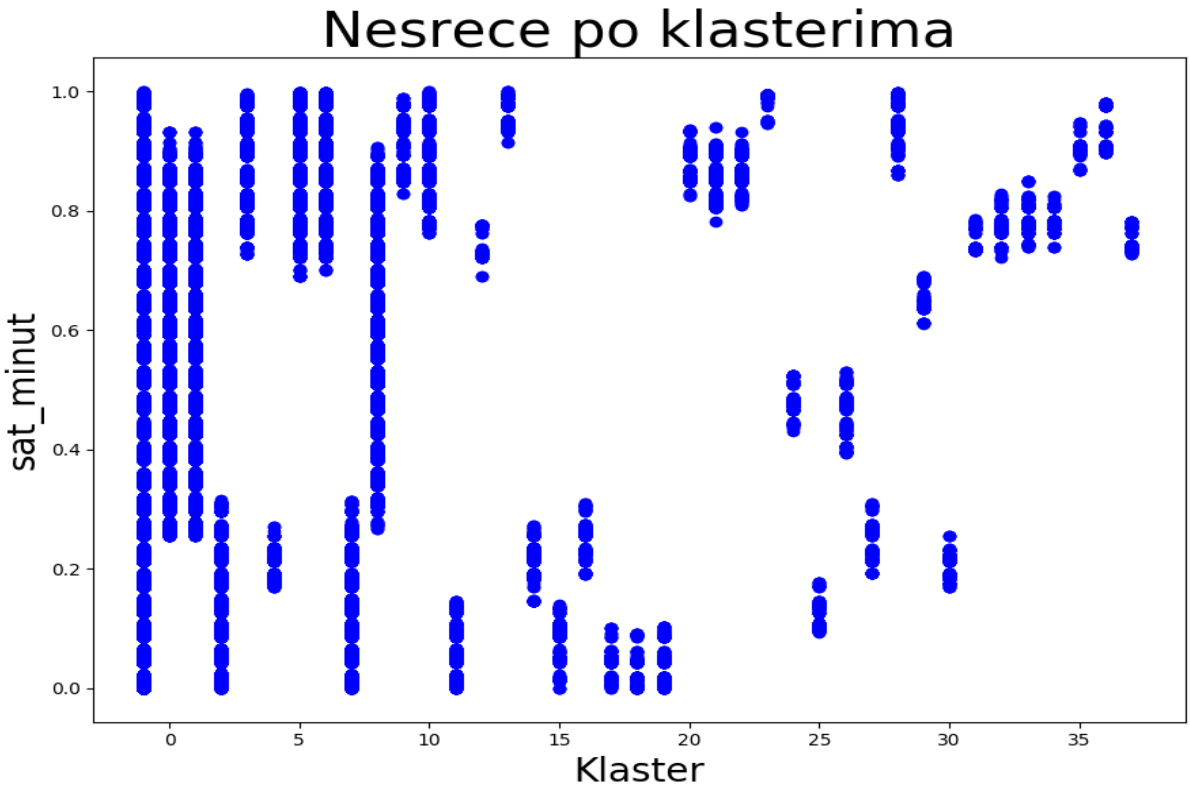
Zbog različitog parametra epsilon uvek smo imali različit broj klastera, kao i različit senka koeficijent, a neki od rezultata su izgledali ovako:



epsilon: 0.048000
Broj klastera: 31
Senka koeficijent: 0.725312



epsilon: 0.046000
Broj klastera: 35
Senka koeficijent: 0.697891



epsilon: 0.044000
Broj klastera: 39
Senka koeficijent: 0.701062

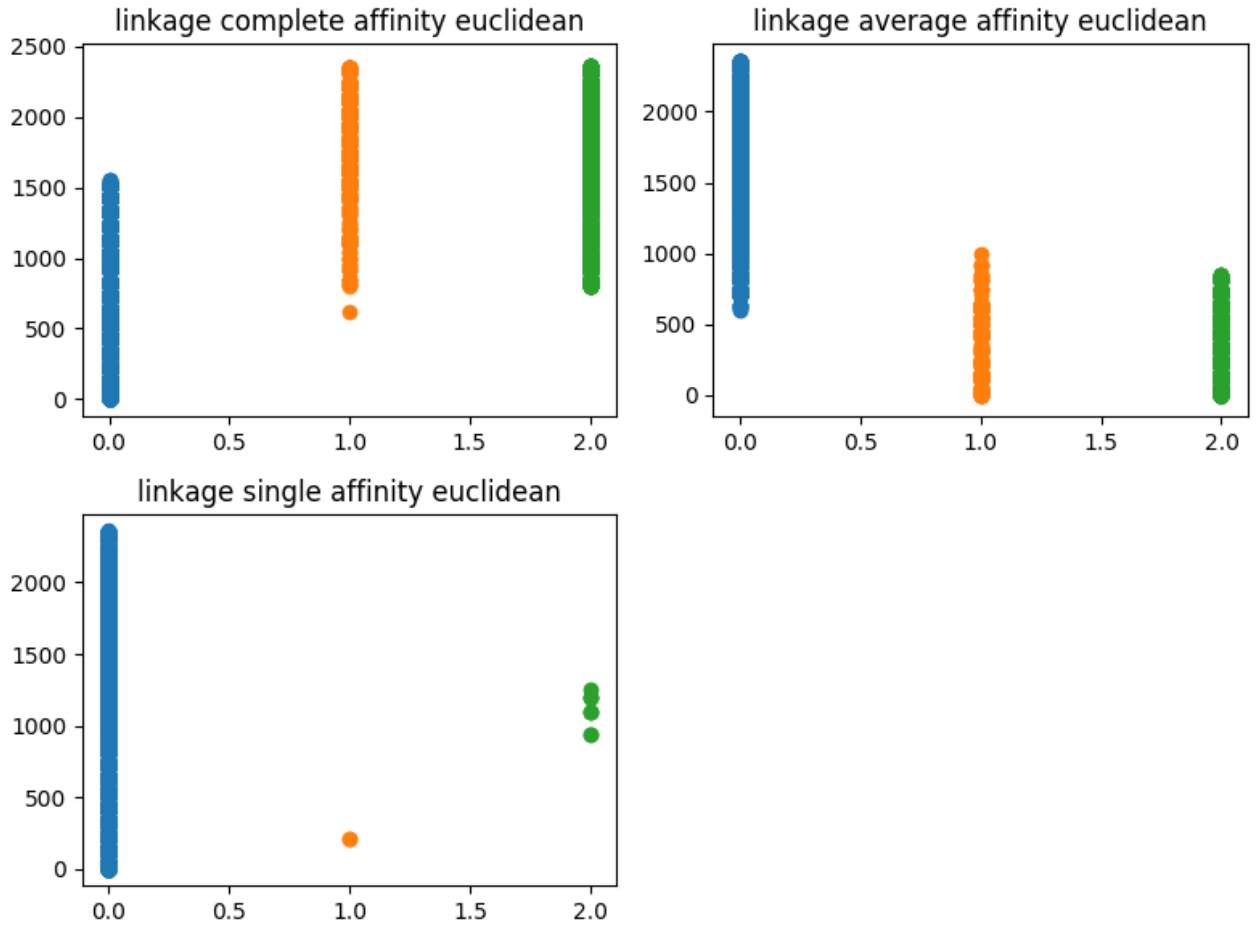
Iz priloženih rezultata, možemo reći da smo dobili sasvim korektan senka koeficijent (senka koeficijent se kreće izmedju -1 i 1. Svaki rezultat koji je veci od 0.5 može se smatrati zadovoljavajućim), a najbolji kada je parametar epsilon iznosio 0.048. Na grafikonu na Y osi stoji “sat_minut” koji je skaliran na vrednost izmedju 0 i 1. Druga stvar koja bi trebala da se vidi iz grafikona jeste da li je važnost prediktora podjednaka (ali nju stvarno ne znam kako da uočim).

Hijerarhijsko Klasterovanje

Postoji više pristupa u rešavanju problema hijerarhijskog klasterovanja. Jedan koji se često razmatra je hijerarhijsko aglomerativno klasterovanje pri kojem se skup klastera

inicijalizuje pojedinačnim instancama, a potom se u svakom koraku, spajaju dva najsličnija klastera u jedan, čime se konstruiše binarno stablo. Pre svega, bitno je napomenuti da za potrebe našeg projekta, nismo bili u mogućnosti da ovaj algoritam izvršimo na svojim računarima zbog očiglednih memorijskih nedostataka. Te smo zbog toga odlučili da ga izvršimo na slučajnom uzorku od 12 hiljada redova. Ovaj algoritam takođe prima više parametara na osnovu kojih se izvršava. Parametri su: broj klastera, mera rastojanja i prosek, minimum ili maksimum rastojanja njihovih elemenata. Prilikom izvršavanja algoritma, nismo menjali parametar broj klastera.

link average affinity manhattan n of clusters 3 silhouette 0.3966656741499993
link single affinity manhattan n of clusters 3 silhouette 0.46655606553822015



link complete affinity euclidean n of clusters 3 silhouette 0.40905797926406867
link average affinity euclidean n of clusters 3 silhouette 0.43032868383294304
link single affinity euclidean n of clusters 3 silhouette 0.4741387987961995

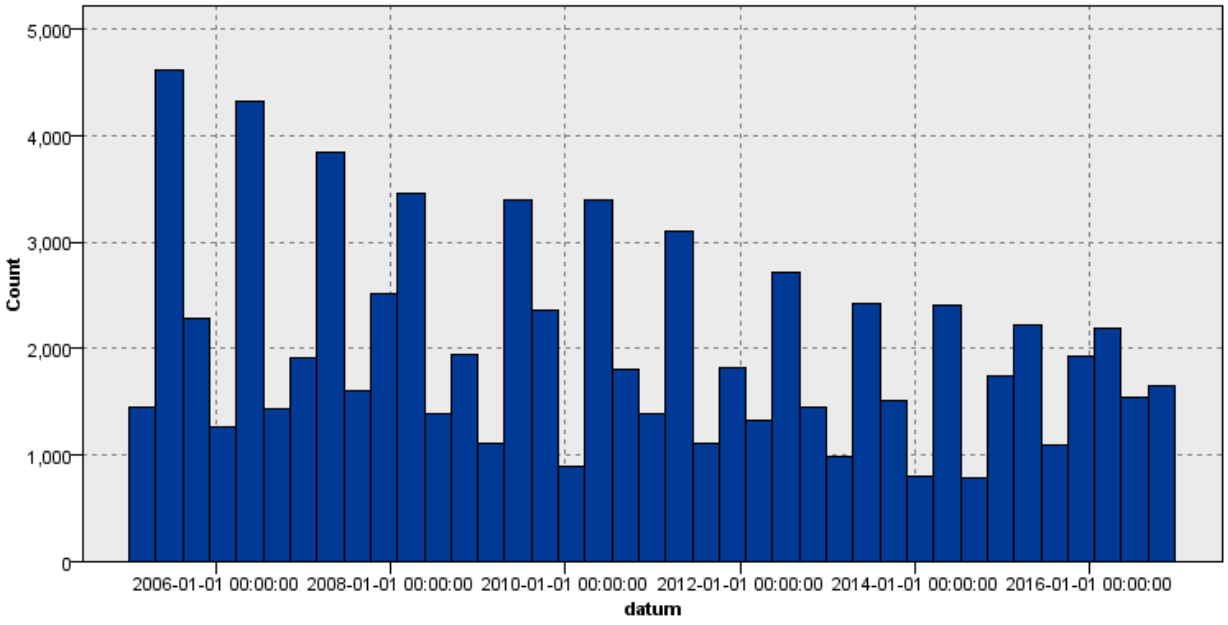
Na osnovu dobijenih rezultata zaključujemo da su se najbolje pokazale verzije “single affinity”, takođe verzija sa euklidskim rastojanjem se pokazala kao neznatno bolja. Ipak, niti jedna verzija nije uspela da pređe broj 0.5, pa ne možemo da budemo

prezadovoljni rezultatima. Naravno, ne možemo da znamo kakvi bi rezultati bili ukoliko bismo pokrenuli klasterovanje nad celim skupom.

Zanimljivosti

Pomoću SPSS modelera izvukli smo nekoliko interesantnih zaključaka iz podataka koje smo dobili.

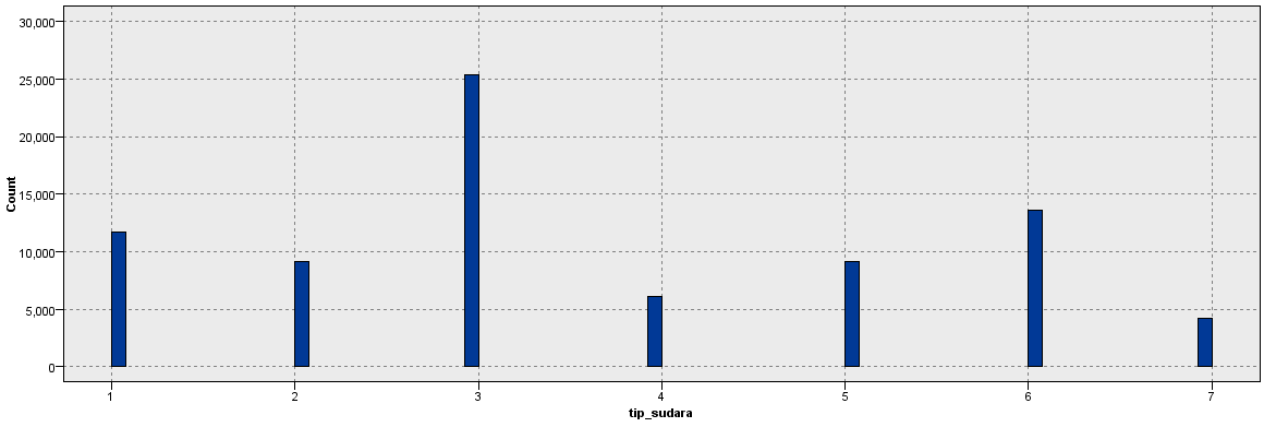
Prvi interesantan je da se broj saobraćajnih nesreća na praznike smanjuje iz godine u godinu, što se može videti na sledećem dijagramu:



Takođe, najmanja stopa saobraćajnih nesreća od svih praznika u godini je za božić. Verovatno jer je običaj da za taj praznik ljudi ne putuju nigde, već sede kod kuće sa svojim porodicama.

Value	Proportion	%	Count
Bastille Day		10.7	8476
All Saints Day		9.94	7869
Victory in Europe Day		9.85	7802
Ascension Thursday		9.71	7686
Labour Day		9.55	7562
Whit Monday		9.54	7555
Armistice Day		9.24	7319
Assumption of Mary to Heaven		9.16	7249
Easter Monday		8.33	6596
New year		7.91	6261
Christmas Day		6.07	4805

Takođe kada gledamo tip sudara, vidimo da je najviše sudara kod kojih se dva vozila sudaraju sa strane (oko 32%), dok je najmanji broj sudara bez kontakta (oko 5%)



Broj saobraćajnih nesreća po tipu sudara.

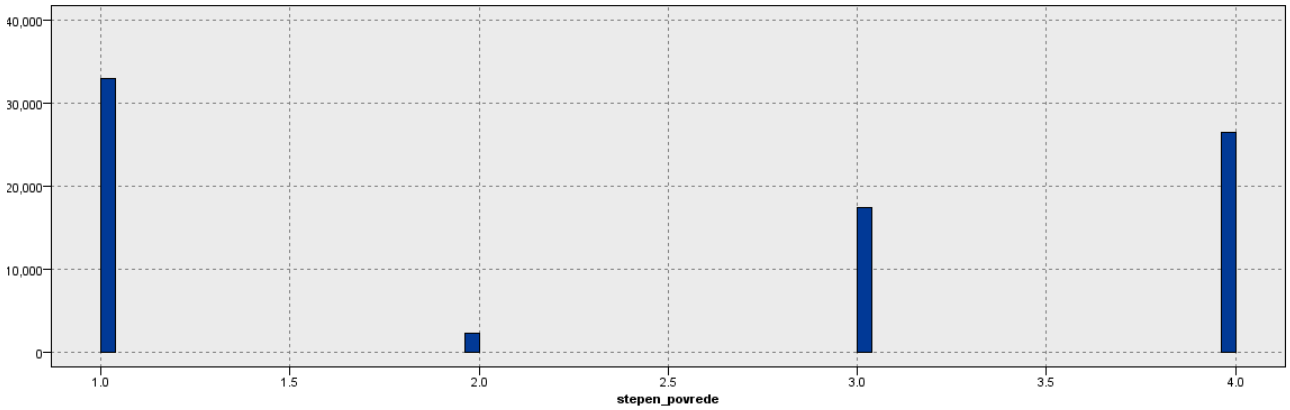
Ako pogledamo broj ljudi po tipu povrede zadobijene prilikom saobraćajne nesreće, ohrabrujuć je podatak da je najveći broj njih nepovređen:

Nepovređenih: 33005 (oko 41%)

Nastradalih: 2295 (oko 3%)

Hospitalizovanih: 17399 (oko 22%)

Lakše povređenih: 26482 (oko 34%)



Broj ljudi po tipu povrede zadobijene prilikom saobraćajne nesreće

Tumačenje brojeva po atributima

tip_sudara : tip sudara:

- 1 - Dva vozila - sa prednje strane
- 2 - Dva vozila - otpozadi
- 3 - Dva vozila - sa strane
- 4 - Dva ili više vozila - lančani
- 5 - Tri ili više vozila - više sudara
- 6 - Ostalo
- 7 - Bez kontakta

atmosferski_uslovi: atmosferski uslovi prilikom saobraćajne nesreće:

- 1 - Normalni
- 2 - Slaba kiša
- 3 - Teška kiša

- 4 - Sneg
- 5 - Magla
- 6 - Jak vetar
- 7 - Hladno
- 8 - Oblačno
- 9 - Ostalo

stepen_povrede: stepen povrede učesnika u saobraćajnoj nesreći:

- 1 - Nepovređen
- 2 - Nastradao
- 3 - Odvezen u bolnicu (Hospitalizovan)
- 4 - Lakše povređen

satminut: vreme udesa zapisano u obliku hhmm

osvetljenje : Osvetljenje na putu prilikom saobraćajne nesreće

- 1 - Dan
- 2 - Svitanje ili zalazak sunca
- 3 - Noć bez javnog osvetljenja
- 4 - Noć sa ugašenim javnim osvetljenjem
- 5 - Noć sa upaljenim javnim osvetljenjem