Seminarski rad iz predmeta Istraživanje podataka 1
Podaci: FIFA19

Nikola Janković
e-mail: nikola_jankovic@tuta.io

Matematički fakultet, Univerzitet u Beogradu

15. avgust 2019

Sažetak

Tema ovog seminarskog rada je detaljnija analiza, sa klasterovanjem, skupa podataka dobijenog iz poslednje verzije igrice FIFA19 (u toku pisanja rada) preuzetog sa adrese: `https://www.kaggle.com/karangadiya/fifa19`. U radu će biti prikazani neki od algoritama za klasterovanje i rezultati dobijeni primenom na ovaj skup. Uz pokušaj da budu dobijeni klasteri to približniji nekim podelama koje postoje u fudbalu.

1 Uvod

Klaster analiza je grupisanje objekata koje se oslanja samo na informacije koje se nalaze u podacima koji opisuju te objekte i veze među njima. Cilj klaster analize je da objekti u grupi budu slični(povezani) međusobno i drugačiji od objekata u drugim grupama. Što je veća sličnost u grupi i različitost među grupama klaster analiza je izrazitija.
U ovom seminarskom radu biće prikazani rezultati klaster analize pomoću algoritama koji su viđeni na kursu Istraživanje podataka 1:

- K-means
- DBSCAN
- Self Organizing Maps (*Kohonen*)
- Hijerahijsko klasterovanje

kao i dva dodatna algoritma:

- Mean-Shift
- BIRCH

Svi algoritmi su primenjeni uz pomoć biblioteka jezika Python uz korišenje softvera IBM Spss Modeler zbog loše dokumentacije vezane za SOM[1] u modulu *minisom*[2].

1.1 Skup podataka korišćen u radu

Skup podataka sastoji se od $\approx$ 18000 igrača(*slogova*) i 89 ocena(*atributa*). Blagi uvid u tabelu je moguć na slici 1.

ID	Name	Age	Photo	Nationality	Flag	Overall	Po
158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentina	https://cdn.sofifa.org/flags/52.png	94	94
20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portugal	https://cdn.sofifa.org/flags/38.png	94	94
190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Brazil	https://cdn.sofifa.org/flags/54.png	92	93
193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spain	https://cdn.sofifa.org/flags/45.png	91	93
192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgium	https://cdn.sofifa.org/flags/7.png	91	92

Slika 1: data.csv

[1] Self Organizing Maps
[2] `https://github.com/JustGlowing/minisom`

Podaci iz skupa se koriste kao parametri koje koristi kompanija *EA Sports* pri kreiranju simulacije fudbalera iz realnog sveta kako bi napravili distinkciju među njima.
Nazivi kolona uglavnom nedvosmisleno ukazuju na njihovo značenje, ali će ipak biti data objašnjenja za neke od atributa, koje korisnik smatra da nisu poznati većini.

- *Value* - Predstavlja procenjenu trenutnu vrednost igrača u dolarima, potrebno je praviti razliku u odnosu na atribut *Release Clause*
- *International Reputation* - Broj između 0 i 1 koji govori koliko je uspeha imao u igrama za reprezentaciju svoje zemlje.
- *Loaned from* - Pojedini igrači mogu biti posuđeni timu X od strane tima Y. Do kraja posudbe tim X je u obavezi da plaća igrača u istom iznosu kao što je to radio tim Y. Na kraju posudbe tim X ima prednost (u nekim slučajevima i pravo) da otkupi u potpunosti prava na igrača od tima Y.
- *LS, ST, RS, ..., RB* - Atributi koji predstavljaju koliko je projektovana Overall ocena igrača u slučaju da ga osoba koja igra igricu postavi na poziciju sa tim nazivom kolone.
- *Release Clause* - Procenjena cena koju je potrebno da tim Y plati timu X kako bi otkupio prava na igrača, često je vrednost ovog atributa veća u odnosu na atribut *Value*, pogotovo kod mlađih igrača.

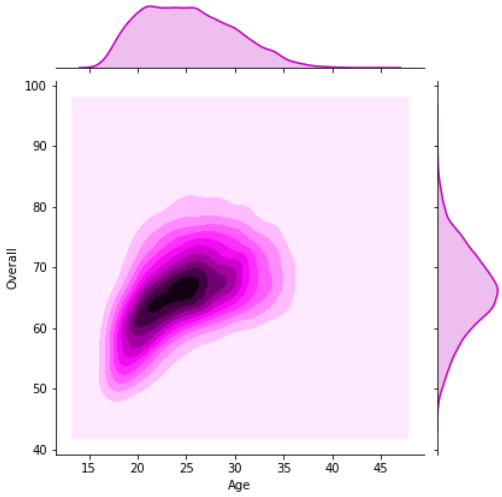
2 Analiza podataka

U ovom delu biće izložene neke zanimljive statistike iz skupa i prikazano kako je izvršeno pretprocesiranje podataka

2.1 Statistike

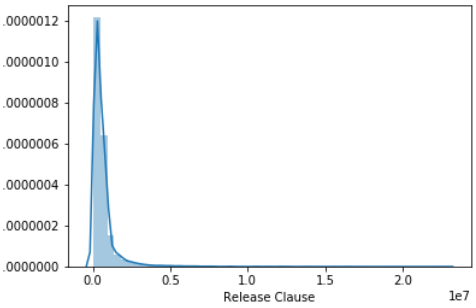
Na slici 2. možemo videti dijagaram zajedničke gustine raspodele za ocene *Overall* i *Age*

Vidimo da je raspodela za *Overall* normalna, dok *Age* podseća na neku $\tilde{\chi}^2$ raspodelu. Kao i da najveći broj fudablera ima između 23 i 27 godina sa *Overall* od 60 do 70 (potpuno očekivano).



Slika 2: Dijagram odnosa Age $\sim$ Overall

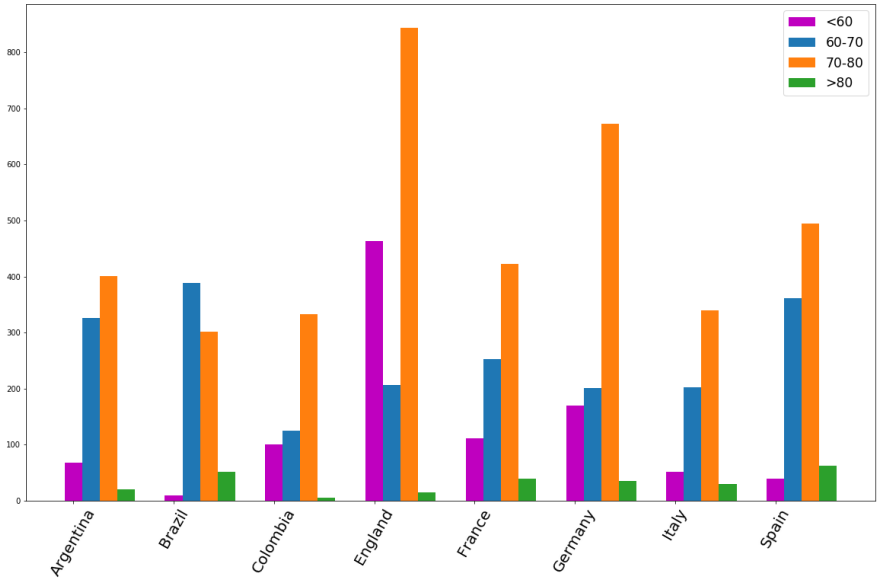
Druga zanimljiva statistika nam pokazuje raspodelu za atribut *Release Clause* i lako se uočava da najveća količina novca figurira u malom procentu igrača. Dok je kod igrača koji nisu vrhunske klase to značajno manje.



Slika 3: Raspodela izlazne klauze

I treći dijagram nam pokazuje koliko igrača nam dolazi iz koje države (razmatrane samo države koje imaju više od 500 predstavnika)

Primetno je da je broj igrača iz Engleske najveći kao i da dominiraju u broju igrača sa ocenom manjom od 60. Razlog ovome je to što u igrici postoje timovi iz čak 4 engleska ligaška takmičenja u kojima većinu čine igrači iz Engleske, a kako su timovi iz 3. i 4. ranga polu-profesionalni, ocene za igrače su očekivano niske.



Slika 4:

2.2 Pretprocesiranje

U početku skup se sastoji od 89 atributa, od kojih $\approx$ 55 su numerički dok su ostali što ordinalni što nominalni.

Prvi korak je bio eliminacija jednog broja kvalitativnih atributa koji nisu mogli nikako biti korišćeni u ovoj klaster analizi. Kao na primer: *ID, Name, Photo, Club, Club Logo, Flag, Jersey Number, Loaned From, Work Rate, Real Face, Joined, Body Type.*

- Atribut *Body Type* je ordinalni atrbiut u vidu stringa sa nekoliko vrednosti: C.Ronaldo, L.Messi, X.Shaqiri kao i vrednstima visok, nizak, zdepast. Pritom atributi *Height i Weight* objašnjavaju sličnu stvar na mnogo egzaktniji način.

- Takođe atribut *Club* bi se mogao mapirati u numeričke vrednosti koje predstavljaju snagu kluba u svetskim okvirima. Na žalost takvu bazu vodi samo evropska fudbalska asocijacija za klubove iz Evrope, dok u skupu podataka ovog seminarskog rada se nalaze podaci o igračima koji igraju širom sveta.

Sledeći korak u ovom delu je bio izbacivanje atributa *LS, ST, RS, ..., RB* jer su oni zapravo nastali iz ostalih numeričkih atributa koji predstavljaju

kvalitet igračeve igre. Pa su izbačeni kako bi ubrzali rad algoritama.

Neki od numeričkih atributa su dati u vidu stringa, pa je njih bilo neophodno parsirati i pretvoriti u Python numerički tip. To su uglavnom atributi koji predstavljaju novčane vrednosti kao na primer *Wage, Value, Release Clause*. U nastavku je fragment koda kojim je rešen ovaj problem.

```
df['Release Clause'] = df['Release Clause'].replace({
    '    ': '',
    'M': '00000',
    'K': '000',
}, regex=True).apply(fix_value).convert_objects(convert_numeric
    =True)
```

Vrednosti atributa *Height i Weight* su konvertovani iz američkog sistema jedinica u evropski.
Atribut *Position* je iz naziva pozicije mapiran u numeričke vrednosti od 0 do 4.2 koje predstavljaju odaljenost pozicije od sopstvenog gola.

```
position_to_num = {
    'GK': 0.0,
    'CB': 1.0,
    'LCB': 1.2,
    'RCB': 1.6,
    'LB': 2.7,
    'RB': 3.2,
    'LWB': 4.5,
    'RWB': 4.6,
    'CM': 6,
    'LCM': 6.2,
    'RCM': 6.4,
    'CDM': 5,
    'LDM': 5.1,
    'RDM': 5.3,
    'LM': 6.5,
    'RM': 6.7,
    'RAM': 7.3,
    'CAM': 7,
    'LAM': 7.1,
    'LW': 8.2,
    'RW': 8.4,
    'CF': 9.1,
    'LF': 9.2,
    'RF': 9.4,
    'LS': 9.5,
    'RS': 9.7,
    'ST': 10
}
df['Position'].replace(position_to_num, inplace=True)
```

Podatak o nazivu države iz koje dolazi igrač je transformisan u dva atributa koji predstavljaju geografsku širinu i visinu te države uz pomoć modula *geopandas*. Za neke od država kao na primer Velika Britanija, Severna Makedonija, Severna Koreja... nema poklapanja naziva u skupu podataka i

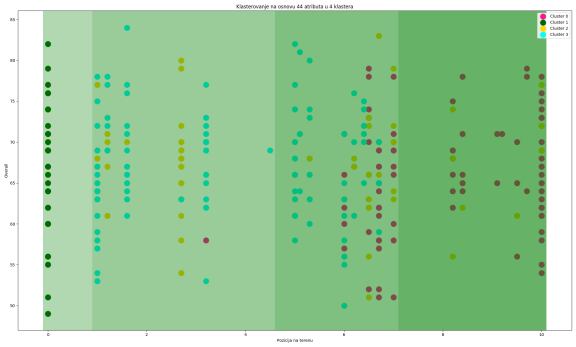
skupu iz *geopandas* modula. Pa je bilo potrebno ručno zameniti. Engleska, Škotska, Vels i Severna Irska su sve preslikane u istu vrednost. Lihtenštajn u vrednost Švajcarske, a Farska Ostrva u vrednost Islanda.
Za neke manje države iz kojih postoji određen broj igrača je zaista bilo potrebno ručno uneti podatke pronađene na webu.
I na kraju su za države sa minornim brojem igrača (Kurakao, Zelenortska ostrva, Antigva i Barbuda...) ostavljene nepoznate vrednosti.

Na kraju u skupu su ostali samo numerički podaci, pa je bilo moguće izvršiti interpolaciju polja sa nedostajućim vrednostima jer mnogo slogova je sadržavalo barem u jednoj koloni nedostajuću vrednost jer je veliki broj atributa i bilo bi loše da smo sve te atribute izbacili. Na kraju je ceo skup podataka skaliran na interval $[0, 1]$. Uz pomoč *sklearn.preprocessing.MinMaxScaler* funkcije iz jezika Python.

3 Primena algoritama

3.1 K-means

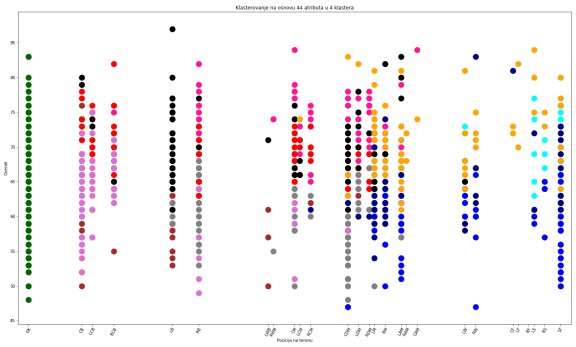
Algoritam K-means je u početku primenjen za vrednost parametra $k = 4$, u nadi da će se izdvojiti četiri klastera igrača za svaku zonu terena po jedan (golman, odbrana, sredina terena i napad).



Slika 5: 4-means Position ∼ Overall

Jasno se vidi da je klasterovanje relativno uspešno, golmani su perfektno izdvojeni jer u skupu postoji 6 atributa karakterističnih za golmane. I oni prave jasnu distinkciju između golmana i igrača u polju. U zajednički klaster su izdvojene i pozicije u napadu sa ofanzivnim veznim pozicijama. Kao i pozicije štopera sa desnim bekovima i zadnjim veznim. Što je opet očekivano jer posao zadnjih veznih i štopera je da "kvare"igru protivnika pa su ti atributi izraženiji, dok je za desne bekove pomalo iznenađujuće.

Po sličnoj intuiciji (11 igrača na teren) isproban je algoritam za parametre $k = 11$,$tol = 1e - 5$

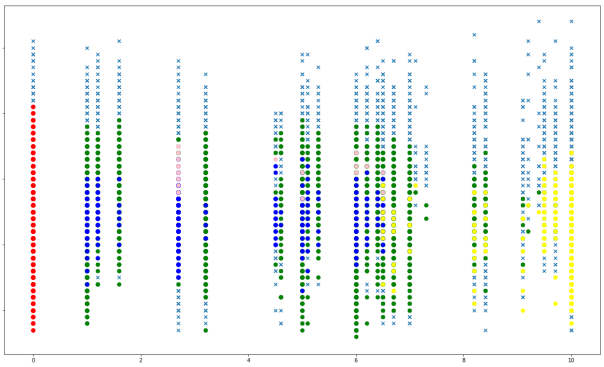


Slika 6: 11-means Position $\sim$ Overall

Sličan rezultat[3] je dobijen i ovde, tako da umesto dobijemo 11 klastera vezanih za svaku poziciju, dobili smo 4 veća klastera po pozicijama i posle podele tih klastera na manje u odnosu na kvalitet igrača na tim pozicijama. *Senka koeficijent* dobijen ovakvim klasterovanjem je 0.176189407.

3.2 DBSCAN

Ako bi samo primenili algoritam na ceo skup, sa ulaznim parametrima $\epsilon = 0.2$ i MINSAMPLE=0.25 rezultati su katastrofalni.



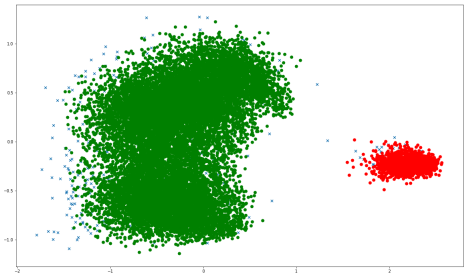
Slika 7: dbscan Position $\sim$ Overall

Ovaj algoritam je dao najlošije rezultate u radu sa celim skupom, pa je uz pomoć algoritma PCA razbijen na 5 komponenata. Pa je nad njima

[3]U oba slučaja na grafiku je prikazan dosta manji slučajan uzorak u odnosu na ceo skup podataka

izvršeno testirnje za kombinacije parametara. I dobijeni su sledeći rezultati za *senka koeficijent*:
EPS: 0.2, MINSAMPLE 15, -0.1869359996151717
EPS: 0.2, MINSAMPLE 17, -0.16759585821982464
EPS: 0.2, MINSAMPLE 19, -0.2173351910035472
EPS: 0.2, MINSAMPLE 22, -0.11936531330028984
EPS: 0.2, MINSAMPLE 25, -0.1105914961339661
EPS: 0.25, MINSAMPLE 15, 0.020336130534418833
EPS: 0.25, MINSAMPLE 17, -0.07882488012519194
EPS: 0.25, MINSAMPLE 19, -0.013699088736421575
EPS: 0.25, MINSAMPLE 22, 0.08744345584290553
EPS: 0.25, MINSAMPLE 25, -0.000881245667701518
EPS: 0.28, MINSAMPLE 15, 0.1888508085038652
EPS: 0.28, MINSAMPLE 17, 0.10032136758996933
EPS: 0.28, MINSAMPLE 19, 0.10049425982754563
EPS: 0.28, MINSAMPLE 22, 0.05128258301759307
EPS: 0.28, MINSAMPLE 25, 0.10400024000298268
EPS: 0.3, MINSAMPLE 15, 0.21026581196557972
EPS: 0.3, MINSAMPLE 17, 0.09208329293314058
EPS: 0.3, MINSAMPLE 19, 0.20001618066160468
EPS: 0.3, MINSAMPLE 22, 0.12226020063683611
EPS: 0.3, MINSAMPLE 25, 0.1196769498036405
EPS: 0.35, MINSAMPLE 15, 0.2683946717146196
EPS: 0.35, MINSAMPLE 17, 0.26255846338140093
EPS: 0.35, MINSAMPLE 19, 0.2588889497655407
EPS: 0.35, MINSAMPLE 22, 0.24745750354415014
EPS: 0.35, MINSAMPLE 25, 0.19363694433779471

Kada grafički predstavimo klastere dobijene za najbolje ulazne parametre vidimo da on jeste dobar ali da pravi samo 2 klastera. I to verovatno samo na onu najveću podelu golman-igrači u polju. Pa nam i ne daje neke značajnije informacije.

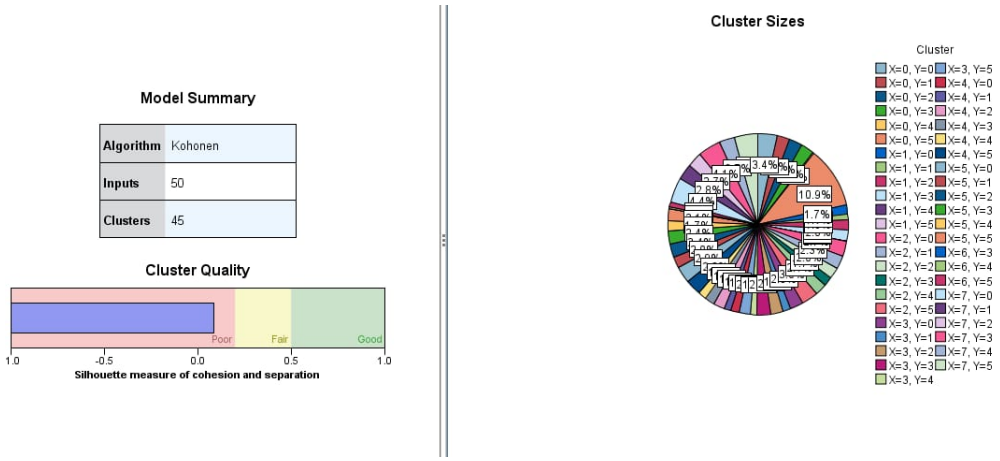


Slika 8: dbscan pca1 $\sim$ pca2

Jedan od razloga za ovakvo ponašanje ovog algoritma bi mogao da bude taj što vrednosti atributa nisu gusto raspoređene po celom skupu. Većina ih se u početku nalazi u opsegu od 40 celobrojnih vrednosti. Pa i kad se skaliraju nisu skroz gusto postavljeni na realnoj pravoj. Zato je teško naći razmeru ϵ i MINSAMPLE u cilju povećanja broja klastera, a ne narušavanju njihovog kvaliteta.

3.3 Self Organizing Maps

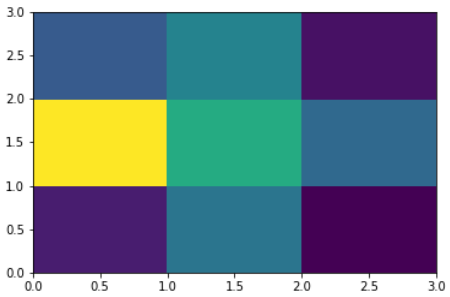
Algoritam Kohonen primenjen u spss na sve podatke daje rezultate:



Slika 9: SPSS Kohonen

Na žalost ovoliki broj klastera nam ne znači. Važnost atributa u svakom od klastera moguće je videti ovde.

Pokušalo se i primenom istog algoritma uz pomoć PCA, iz Python modula *minisom* na mreži 3x3.

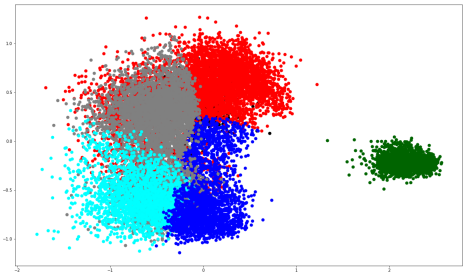


Slika 10: Distance Map

Vidimo da je *senka koeficijent* sličan onom dobijenom pomoću softvera SPSS Modeler. Tako da i ovaj pokušaj možemo podvesti pod neuspešan.

3.4 Hijerahijsko klasterovanje

Aglomerativni algoritam je odmah testiran za kombinacije ulaznih parametara.



Slika 11: SOM pca

complete 3: 0.16111395901587666

complete 4: 0.15463340908414164

complete 5: 0.10867388502724341

complete 6: 0.10899681823968788

complete 7: 0.1163346869599695

complete 8: 0.08395732844533542

complete 9: 0.07767909227419069

complete 10: 0.06966049001459615

average 3: 0.2709627878327729

average 4: 0.24091917722313702

average 5: 0.1728780322782311

average 6: 0.11372869517045374

average 7: 0.0960059077583922

average 8: 0.08104887570408535

average 9: 0.07844464901824208

average 10: 0.060803301819689466

single 3: 0.4411833673662859

single 4: 0.41806554059568624

single 5: 0.4043615186449922

single 6: 0.25612808527919223

single 7: 0.2544154870862858

single 8: 0.25318604389230687

single 9: 0.24743314095736932

single 10: 0.19687832678703504

ward 3: 0.24043039484247533

ward 4: 0.2380921126774734

ward 5: 0.1895722289010395

ward 6: 0.19807178307014062

ward 7: 0.16971785435757866

ward 8: 0.16611285555102587

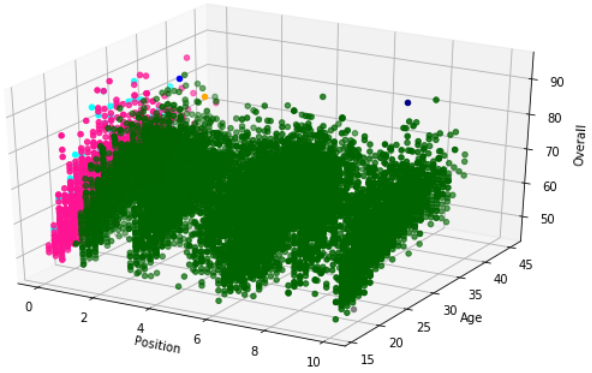
ward 9: 0.15481562166653126

ward 10: 0.15596833147946337

Nakon što je primećeno da single veza daje najbolje rezultate. Primenjen je algoritam ponovo za broj klastera 11.

Očigledno i ovog put razdvaja dva najveća klastera, čak i među golmanima pronalazi potklastere.

Isprobano je sa istim parametrima primeniti algoritm i na podatke redukovane sa PCA:



Slika 12: Agg11 Overall ≈ Age ≈ Position

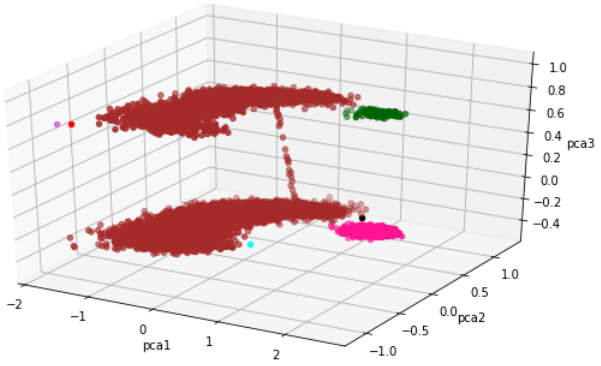
Jedna zanimljivost, pri ovakvoj klasterizaciji pojavljuju se 3 klastera sa samo po jednim objektom. U kojima se nalaze redom : po mnogima najbolji fudbaler svih vremena Lionel Messi, golman iz Japana od 42 godine i golman iz Indije koji je trenutno bez kluba, kom zapravo u skupu podataka i nedostaje vrednost za atribut pozicije na kojoj igra. Pa je pri interpolaciji svrstan u igrače u polju to jest vrednost *Position* je > 0

3.5 Mean-shfit

Mean-shift algoritam je još algoritam kod kog nije moguće kao parametar zadati broj klastera koji želimo da izdvojimo. Već zadajemo okolinu (*engl. bandwidth*).

```
Input: bandwith, skup podataka
WHILE postoji objekat koji nije dodeljen nijednom klasteru DO:
    izaberi jedan od nedodeljenih objekata i ozna i
    da pripada novom klasteru
    REPEAT:
      azuriraj srednju tacku(centroid) u trenutnom klasteru;
      sve ta ke koje se nalaze na razdaljini manjoj
      od bandwidth oznaci ih da pripadaju trenutnom
      klasteru
    UNTIL postoji promena na trenutnom klasteru
```

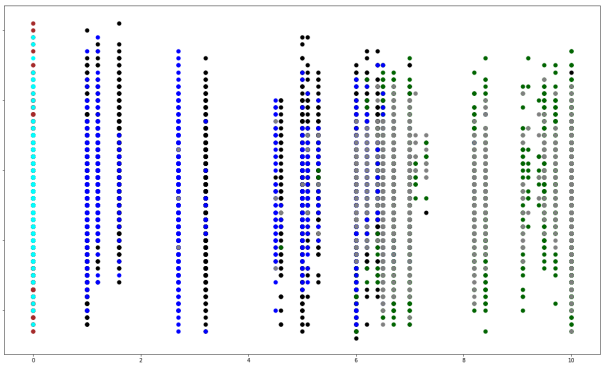
Paramtear bandwidth je u prvom testu izabran uz pomoć *sklearn.estimate_bandwidth()*, rezultat klasterovanja sa ovakvim izborom je odličan *senka koeficijent*, ali sa samo 2 klastera. Pa je smanjen parametar sa 1.5 na 0.5, smanjen je *senka koeficijent*, ali je broj klastera bio 7.



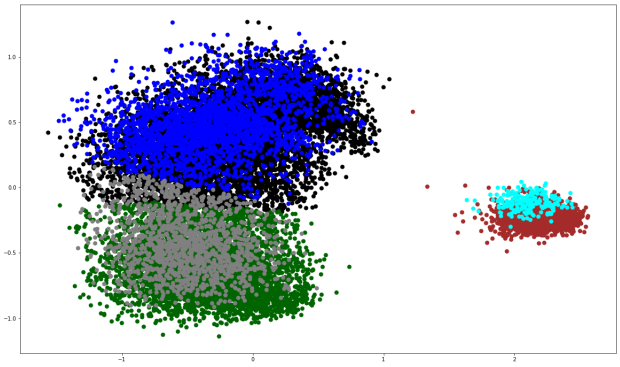
Slika 13: Agg11 PCA

Čak i sa ovim pogoršanjem *senka koeficijenta* on ostaje iznad vrednosit 0.5 Klastera ima više nego pri 4-means algoritmu, ali manje su raspršeni po terenu.

3.6 BIRCH



Slika 14: Meanshift



Slika 15: Meanshift PCA