Klasterovanje mobilnih aplikacija

Zadatak ovog rada je klasterovanje mobilnih aplikacija korišćenjem podataka iz AppleStore-a, pomocu 3 različita algoritma - Dbscan, K sredina i Gaussian mixture.

1. Učitavanje podataka:

```
url = 'https://raw.githubusercontent.com/marinaborozan/IP_projekat/master/app-store-apple-data-set-10k-apps/AppleStore.csv'
df = pd.read_csv(url)
```

2. Uvid u tipove podataka:

```
df.info()
```

```
RangeIndex: 7197 entries, 0 to 7196
Data columns (total 17 columns):
Unnamed: 0          7197 non-null int64
id                  7197 non-null int64
track_name          7197 non-null object
size_bytes          7197 non-null int64
currency            7197 non-null object
price               7197 non-null float64
rating_count_tot    7197 non-null int64
rating_count_ver    7197 non-null int64
user_rating         7197 non-null float64
user_rating_ver     7197 non-null float64
ver                 7197 non-null object
cont_rating         7197 non-null object
prime_genre         7197 non-null object
sup_devices.num     7197 non-null int64
ipadSc_urls.num     7197 non-null int64
lang.num            7197 non-null int64
vpp_lic             7197 non-null int64
dtypes: float64(3), int64(9), object(5)
```

3. Provera null vrednosti:

Nema null vrednosti ni u jednoj od kolona.

4. Upoznavanje sa podacima:

Izdvojićemo 5 aplikacija sa najvišim vrednostima atributa rating_count_tot, kao i 5 aplikacija sa najvišim vrednostima atributa user_rating:

```
rating_count_sorted = df.sort_values(by = ['rating_count_tot'], ascending = False)
rating_count_top5 = rating_count_sorted[:5]
rating_count_top5.iloc[:,[2,5,6,7,8,9,12]]
```

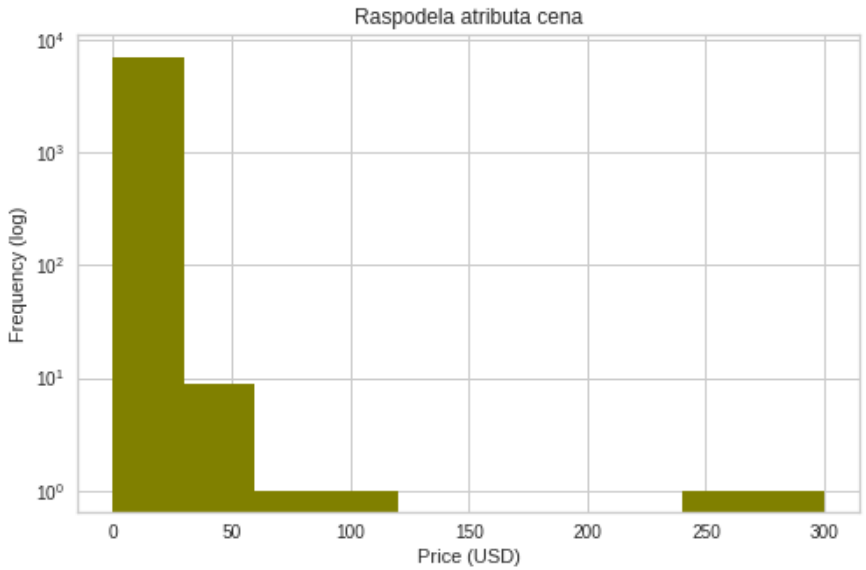
```
user_rating_sorted = df.sort_values(by = ['user_rating'], ascending = False)
user_rating_top5 = user_rating_sorted[:5]
user_rating_top5.iloc[:,[2,5,6,7,8,9,12]]
```

track_name	price	rating_count_tot	rating_count_ver	user_rating	user_rating_ver	prime_genre
Facebook	0.0	2974676	212	3.5	3.5	Social Networking
Instagram	0.0	2161558	1289	4.5	4.0	Photo & Video
Clash of Clans	0.0	2130805	579	4.5	4.5	Games
Temple Run	0.0	1724546	3842	4.5	4.0	Games
Pandora - Music & Radio	0.0	1126879	3594	4.0	4.5	Music

track_name	price	rating_count_tot	rating_count_ver	user_rating	user_rating_ver	prime_genre
Escape the Sweet Shop Series	0.00	3	3	5.0	5.0	Games
激おこ!! はじめしゃちょーなんなんですか!?	0.00	1	1	5.0	5.0	Games
Mini Metro	4.99	4064	338	5.0	5.0	Games
Wayfair - Shop Furniture, Home Decor, Daily Sales	0.00	12578	146	5.0	5.0	Shopping
Mystic Castle - the Simplest & Best RPG and Ad...	0.00	650	488	5.0	4.5	Games

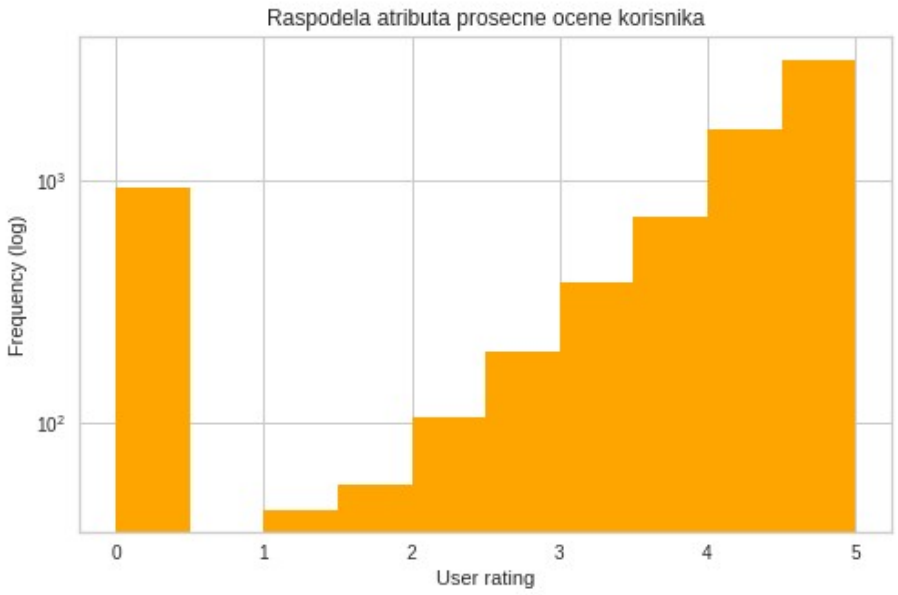
Vidimo da su korisnici jako dobro ocenili aplikacije u kategoriji igrica, a da su najpopularnije aplikacije prema broju glasova Facebook i Instagram.

Prikazaćemo par grafika da bi uočili neke zavisnosti među atributima.



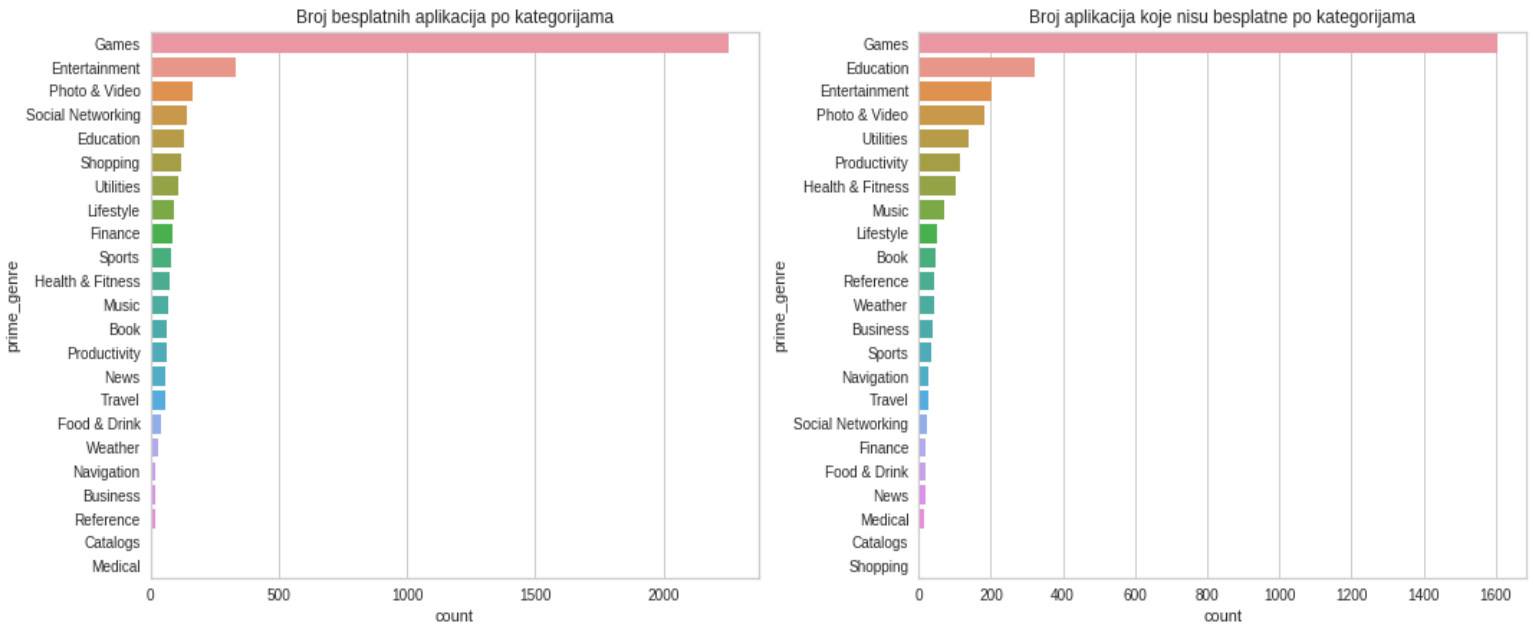
Grafik 1: Raspodela atributa cena

Vidimo da najveći broj aplikacija ima cenu do 50 USD, a da postoji i mali broj aplikacija sa visokom cenom - 250 do 300 USD. To su outlieri i njih ćemo izbaciti iz seta podataka u pretprocesiranju.



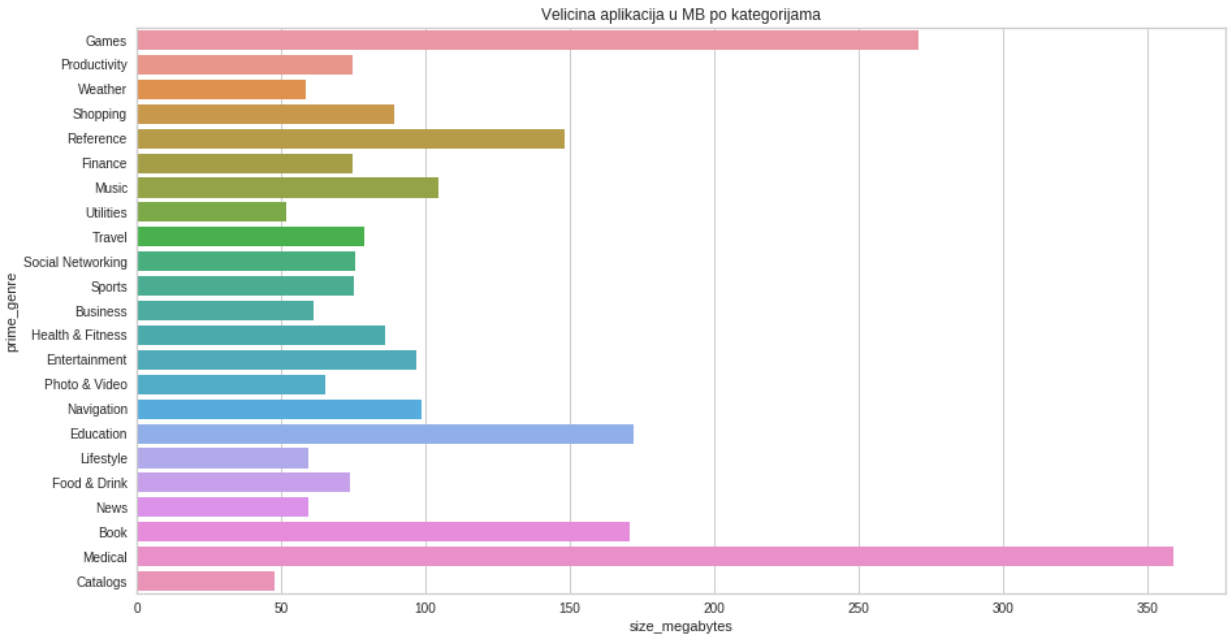
Grafik 2: Raspodela atributa ocene korisnika

Vidimo da najveći broj aplikacija ima ocenu između 3 i 5, najčešće 4.5.



Grafik 3: Broj aplikacija koje (ni)su besplatne u svakoj od kategorija

Dodajemo kolonu free koja oznacava da li je aplikacija besplatna.
Vidimo da se oko 60% igrica naplaćuje, a da u kategoriji Education ima znatno više aplikacija koje se naplaćuju, za razliku od kategorije Entertainment. Social Networking aplikacije su uglavnom besplatne.



Grafik 4: Veličina aplikacija po kategorijama

Da bi lakše radili sa podacima, prebacićemo atribut velicina u MB.
Vidimo da su aplikacije u proseku najveće u kategoriji Medical, zatim Games, Education, Book...

5. Pretprocesiranje:

Najbrojnije 4 kategorije prebacujemo u numeričke vrednosti I sa jos jednom vrednošću označavamo ostale kategorije.
Dodajemo kolonu game koja označava da li je u pitanju igrica ili ne, zbog jako velikog broja igrica u setu podataka.

```
def reduce_genre(x):
  if(x == "Games"):
    return str(1)
  elif(x == "Entertainment"):
    return str(2)
  elif(x == "Education"):
    return str(3)
  elif(x == "Photo & Video"):
    return str(4)
  else:
    return str(5)

df['prime_genre'] = (df['prime_genre']
            .apply (lambda x: reduce_genre(x))
            .astype (int))

#is game
df['game'] = df['prime_genre'].apply(lambda x: 1 if x == 1 else 0)
```

Izbacujemo kolone koje nisu relevantne, id aplikacije, track_name(naziv aplikacije), currency(samo USD vrednost), a neke prebacujemo u numeričke (vpp_lic, prime_genre, cont_rating), ver svodimo na jedan broj.

Izbacujemo i sve redove u kojima je vrednost user_rating jednak 0, jer nam ne daju bitne informacije o aplikaciji.

Sada nam set podataka izgleda ovako:

```
df.info()
```

```
Int64Index: 6261 entries, 0 to 7196
Data columns (total 15 columns):
price               6261 non-null float64
rating_count_tot    6261 non-null int64
rating_count_ver    6261 non-null int64
user_rating         6261 non-null float64
user_rating_ver     6261 non-null float64
ver                 6261 non-null int64
cont_rating         6261 non-null int64
prime_genre         6261 non-null int64
sup_devices.num     6261 non-null int64
ipadSc_urls.num     6261 non-null int64
lang.num            6261 non-null int64
vpp_lic             6261 non-null bool
free                6261 non-null int64
size_megabytes      6261 non-null float64
game                6261 non-null int64
dtypes: bool(1), float64(4), int64(10)
```

6. Skaliranje:

Koristimo min max skaliranje jer je u toku rada dalo bolje rezultate.

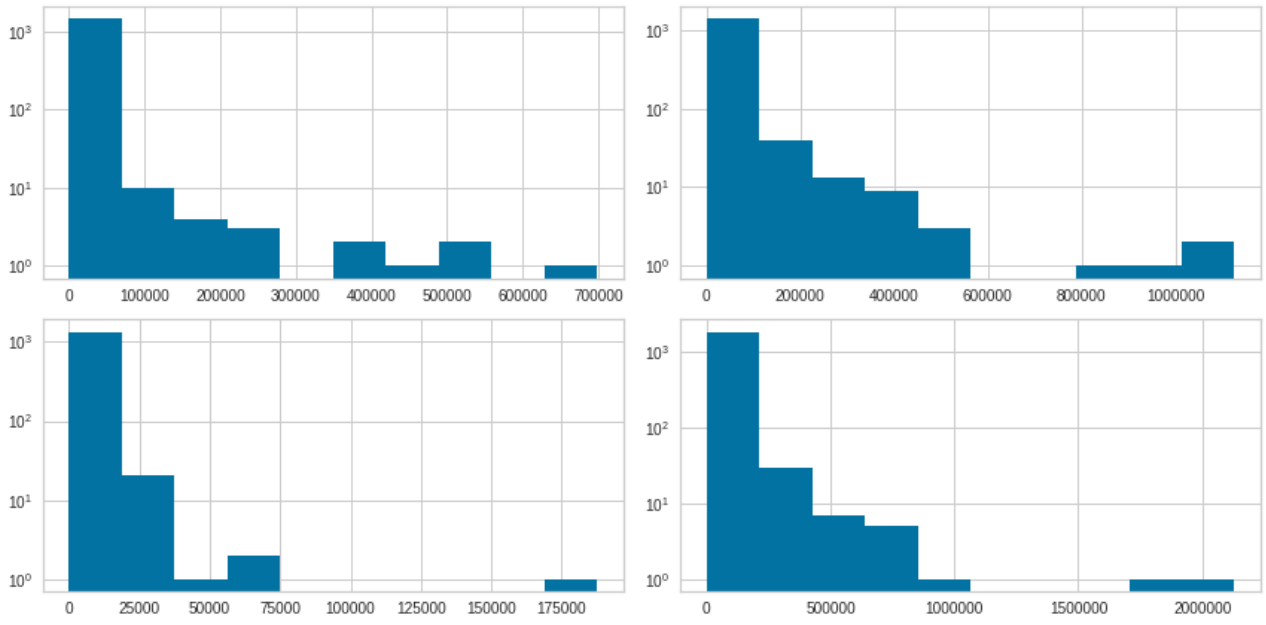
```
features = df.columns
scaler = MinMaxScaler().fit(df[features])
x = pd.DataFrame(scaler.transform(df[features]))
x.columns = features
```

7. Klasterovanje:

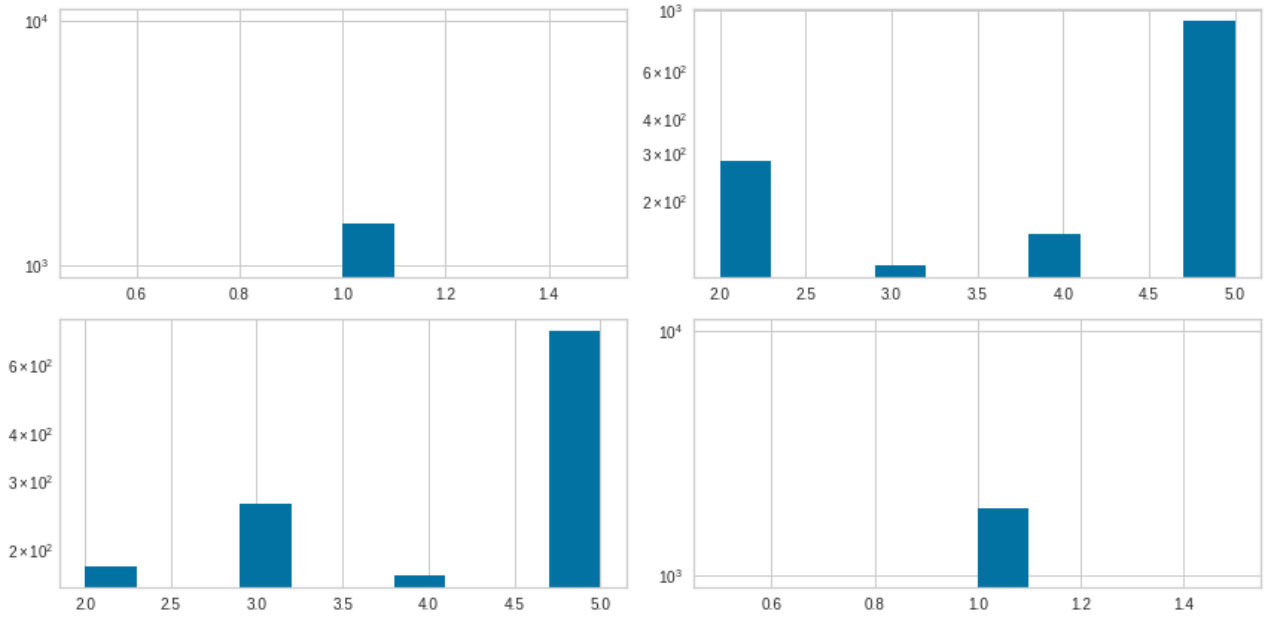
DBSCAN algoritam

```
dbscan = DBSCAN(eps=0.5, min_samples=10)
dbscan.fit_predict(x)
```

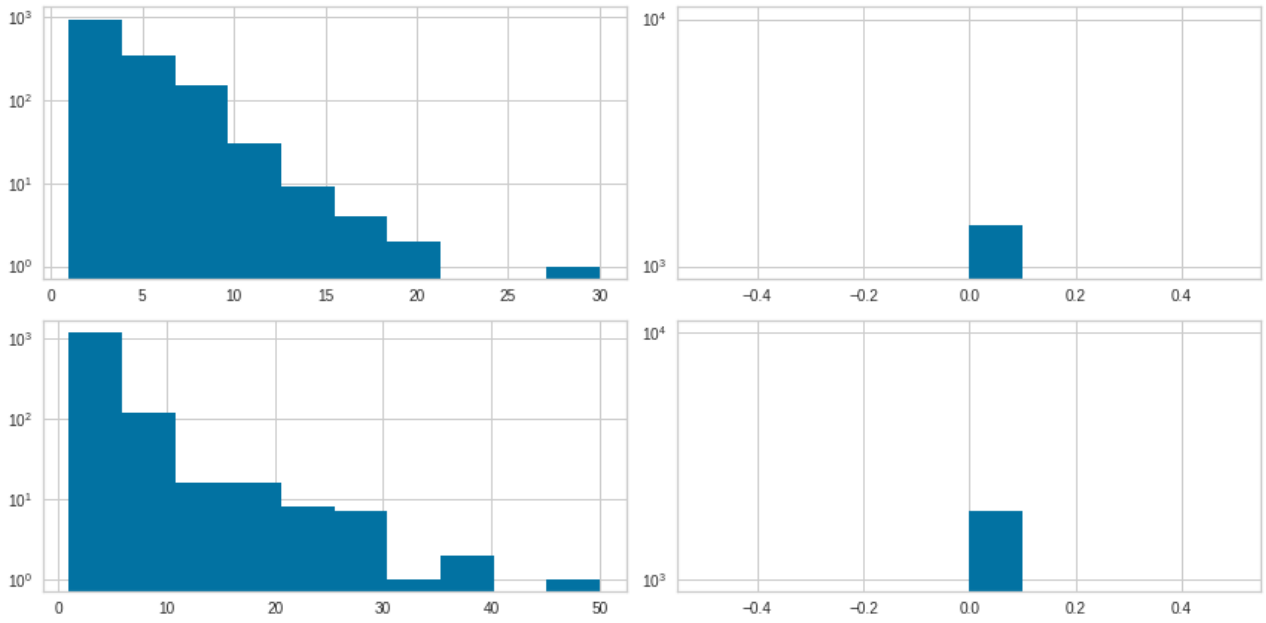
Isprobano je više vrednosti parametara eps i min_samples, od kojih su se najbolje pokazale 0.5 za eps i 10 za min_samples.
Dobili smo 4 klastera, a 78 instanci nam je klasterovano kao noise.
Pri pravljenju histograma za svaki klaster za važnije atribute(velicina, cena, kategorija, broj ocena, prosečna ocena), primećujemo da su interesantne raspodele atributa broj ocena, kategorija i cena. Zaključujemo da je broj ocena mnogo veći kad su u pitanju besplatne aplikacije, kao i kada su u pitanju igrice. Među besplatnim aplikacijama ima i znatan broj njih koje imaju izuzetno veliki broj ocena.



Grafik 5.1: Raspodela atributa rating_count_tot



Grafik 5.2: Raspodela atributa prime_genre



Grafik 5.3: Raspodela atributa price

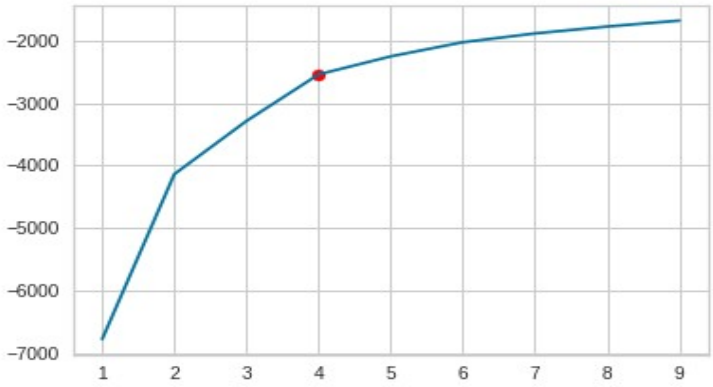
Evaluiramo pomocu silhouette_score metode.

```
sil_score_dbscan = silhouette_score (x, dbscan.labels_)
sil_score_dbscan
```

```
0.397276794904324
```

KMEANS algoritam

```
Ks = range(1, 10)
km = [KMeans(n_clusters=i) for i in Ks]
score = [km[i].fit(x).score(x) for i in range(len(km))]
```

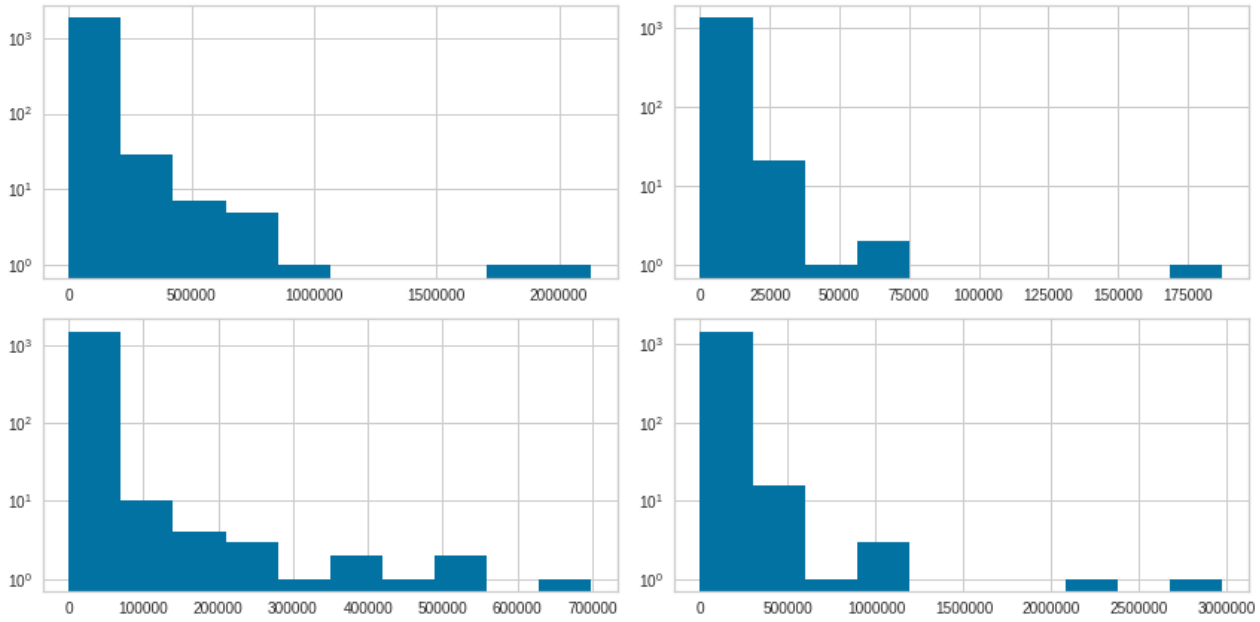


Grafik 6: Elbow metod

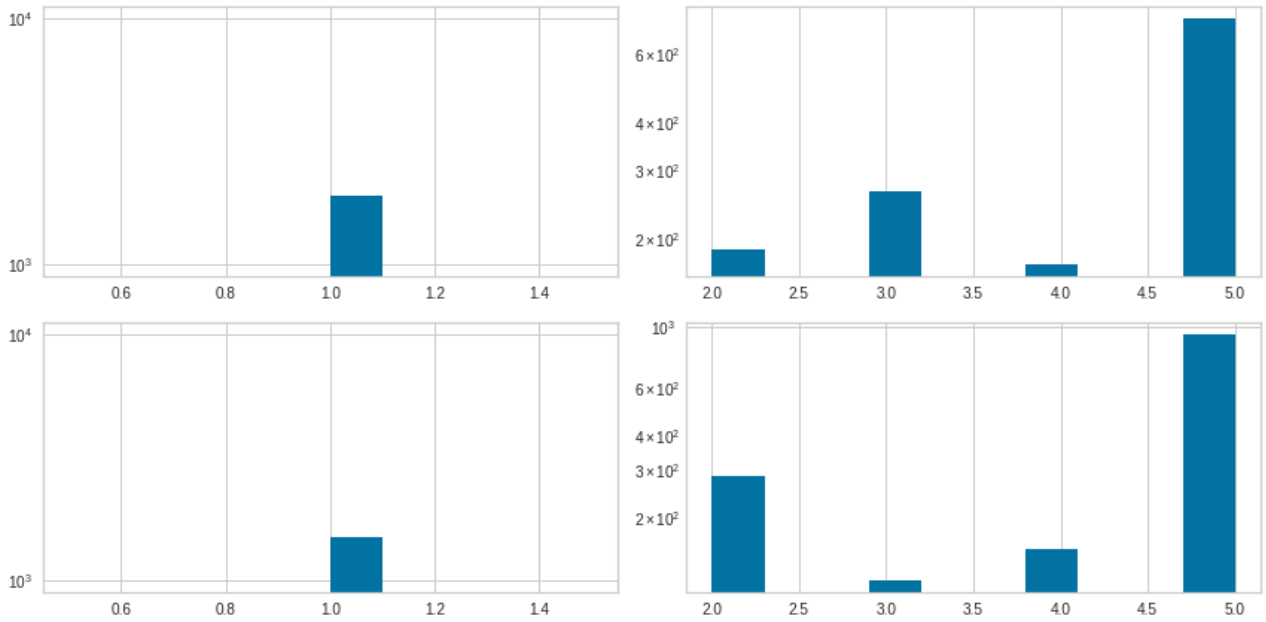
Za predviđanje broja klastera koristimo elbow method i sa grafika vidimo da je najbolja vrednost n = 4. Primenom algoritma sa tim parametrom dobijamo 4 klastera.

```
kmeans = KMeans(n_clusters=4)
kmeans.fit(x)
```

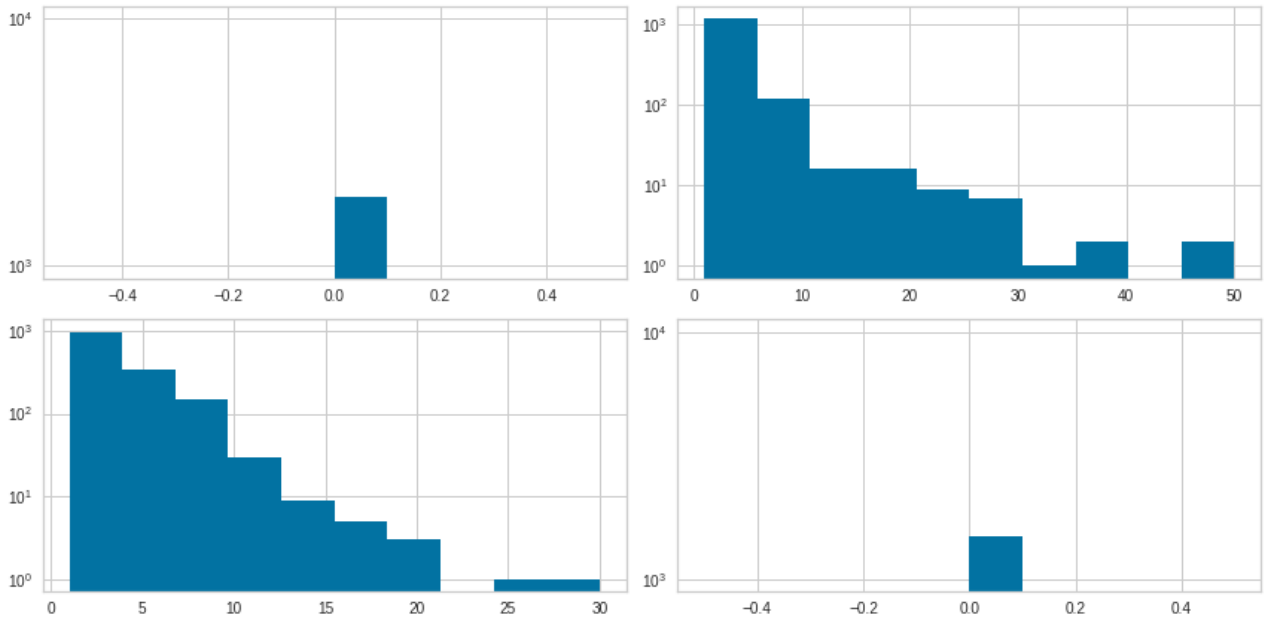
Slično kao kod dbscan algoritma, interesantne su raspodele atributa broj ocena, kategorija i cena. Rezultate možemo interpretirati isto kao kod dbscan. Takođe, vidimo da se igrice koje nisu besplatne kreću u nižem cenovnom rangu nego ostale kategorije aplikacija.



Grafik 7.1: Raspodela atributa rating_count_tot



Grafik 7.2: Raspodela atributa prime_range



Grafik 7.3: Raspodela atributa price

Evaluiramo pomocu silhouette_score metode.

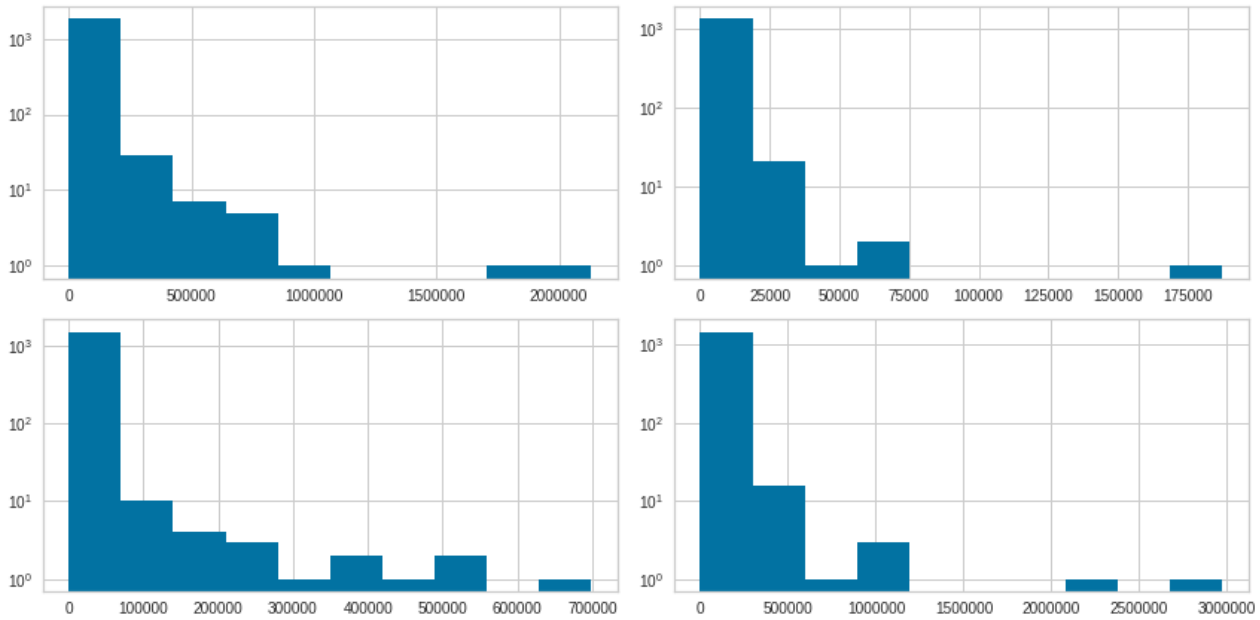
```
sil_score_kmeans = silhouette_score (x, kmeans.labels_)
sil_score_kmeans
```

```
0.3999265029181792
```

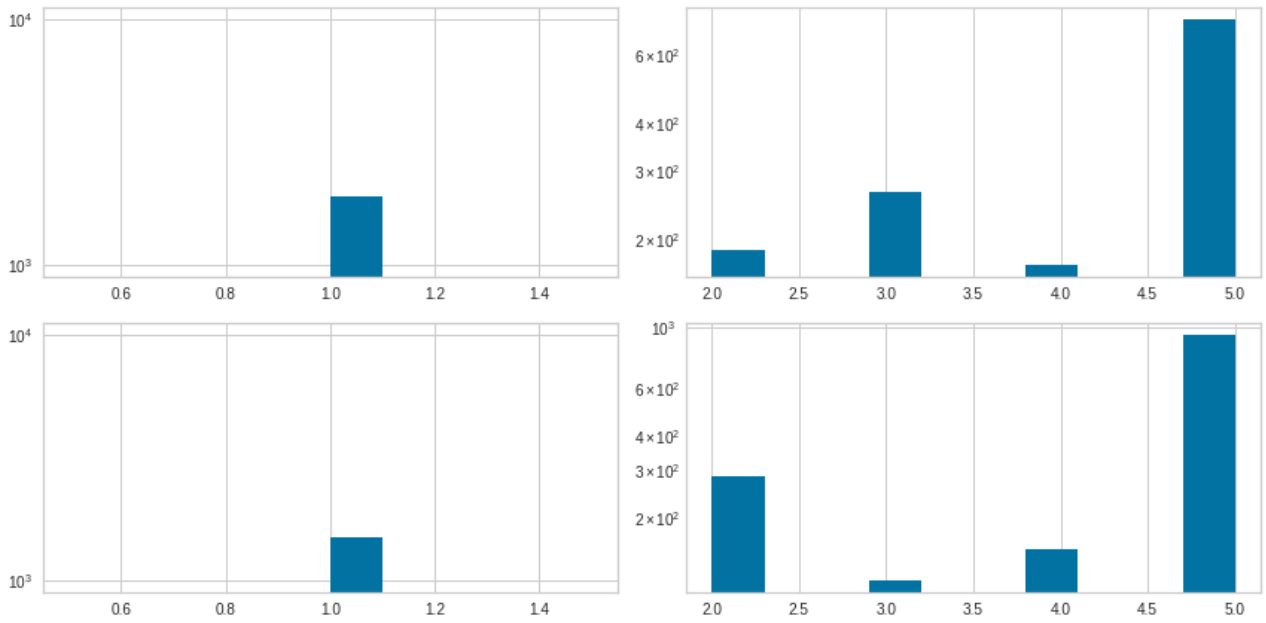
GAUSSIAN MIXTURE algoritam

Probamo razne vrednosti za parametar n koji označava broj klastera, najbolje rezultate daje n = 4, a toliko klastera i dobijamo primenom algoritma sa tim parametrom.
Vidimo da su rezultati vrlo slični kao kod prethodna dva algoritma, pa isto važi i ovde.

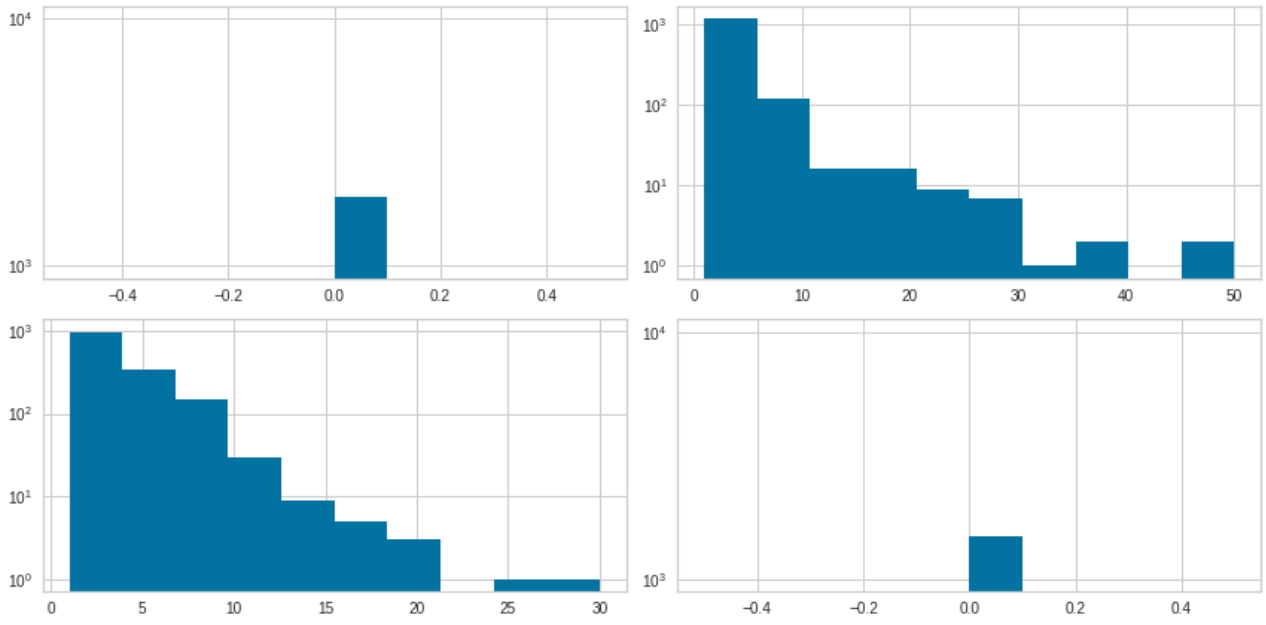
```
gaussian = GaussianMixture(n_components=4)
labels = gaussian.fit_predict(x)
```



Grafik 8.1: Raspodela atributa rating_count_tot



Grafik 8.2: Raspodela atributa prime_genre



Grafik 8.3: Raspodela atributa price

Evaluiramo pomocu silhouette_score metode.

```
sil_score_gaussian = silhouette_score (x, labels)
sil_score_gaussian
```

```
0.3999265029181792
```

8. Zaključak:

Broj ocena se ispostavlja kao najbitniji parametar za popularnost aplikacija. Besplatne aplikacije u proseku imaju 5 do 10 puta veći broj ocena, a određeni deo njih ima i vrlo visok broj ocena. To su neke najpopularnije aplikacije poput Facebook-a i Instagram-a.
Oko 50% svih aplikacija čine igrice, najbrojnija kategorija. Pokazuje se da su igrice popularnije od ostalih vrsta aplikacija nezavisno od toga da li su besplatne ili ne. Cene igrica su u proseku niže nego cene aplikacija koje su u drugim kategorijama.
Kad su u pitanju ostale brojnije kategorije, najveći procenat aplikacija u kategoriji Entertainment je besplatno, dok je u Education dosta veći procenat aplikacija koje se naplaćuju.
Prosečne ocene su visoke, najveći procenat aplikacija ima ocenu oko 4.5.