Klasterovanje

Tim [sigma]:

- Marko Vukotić SW-71/2018
- Miloš Popović SW-24/2018

Zadatak:

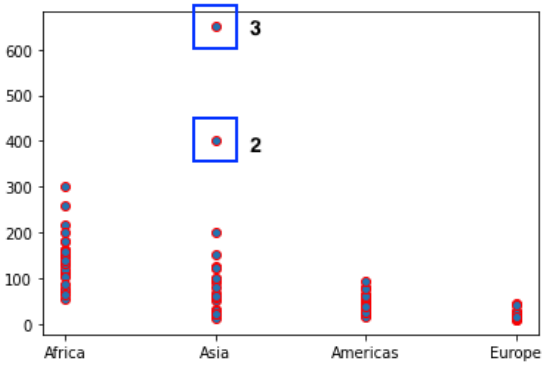
Klasterovati države na osnovu njihovih karakteristika u klastere koji predstavljaju geografske regione. Regioni u pitanju su: Europe, Asia, Americas & Africa.

Analiza podataka:

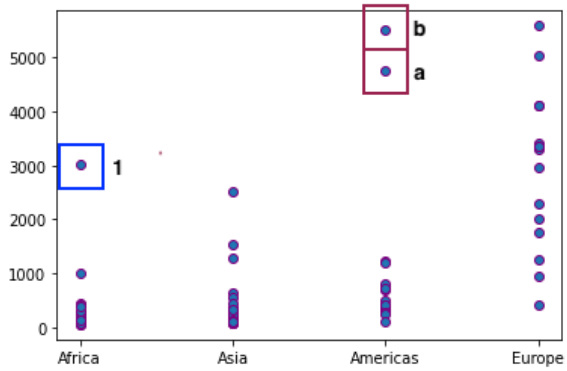
Podaci koji su dati sadrže kolone:

- Income: Prihod po glavi stanovništva u dolarima
- Infant: Smrtnost novorođenčadi na 1000 živorođenih
- Oil - Da li je država izvoznik naftnih derivata (Da/Ne)
- Region - Kontinent države

Nakon plotovanja train skupa vide se neki outlier-i.

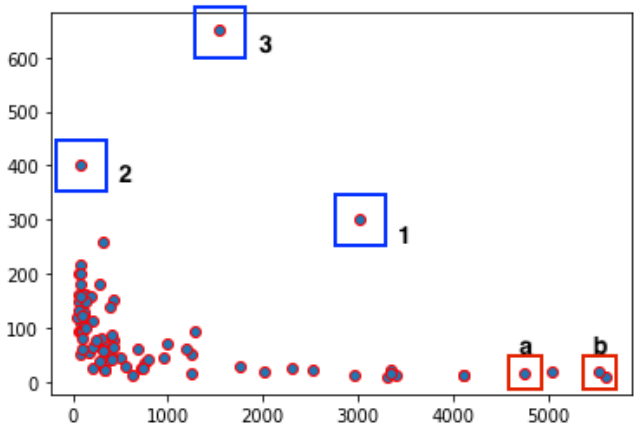


infant po regionima



income po regionima

Imamo problem pošto nemamo dovoljno podataka u testnom skupu, prema tome nismo sigurni da li se stvarno radi o šumu. Na primer u Americi je realno da postoje podaci sa platom preko $5000 iako na prvi pogled ti podaci izgledaju kao očigledan outlier..



odnost infant i income

Kriterijum za izbacivanje outlajera nam je bio da odnos između income i infant bude nekonzistentan u odnostu na druge podatke.

Izbacili smo outlier-e 1,2 i 3

'a' i 'b' podatke ipak nismo izbacili.

Pretprocesiranje podataka:

Postoje ukupno 4 nepotpuna podatka za kolonu infant.

Popunili smo ih medijalnom vrednostima ostalih podataka filtriranih sa identičnim oil i region vrednostima. Na primer ako nam nedostaje podatak infant u državi iz Azije koja je izvoznik nafte, tražimo medianu za infant nad svim podacima država gde je region Asia i jeste izvoznik nafte)

Međutim znatno bolji rezultat (za ~0.06) na validacionom skupu se dobio ako samo te redove izbacimo. (ti podaci čine ~4% skupa, pa neće biti preterano značajno ako se izbaci)

Nad oil kolonom je urađen one hot encoding (zbog ulaza algoritma)

Rešenje

Za klasterovanje smo koristili GaussianMixture metod iz scikit-learn biblioteke

```
parameters = {
    'covariance_type':('full', 'diag', 'tied', 'spherical'),
    'n_components':[5,10,20,50],
    'tol': [1e-6,1e-7,1e-8,1e-9],
    'reg_covar': [1e-10,1e-12,1e-14],
    'max_iter': [5000],
    'n_init': [1],
    'random_state': [1]
    }

gaus = GaussianMixture()
clf = GridSearchCV(gaus, parameters)
clf.fit(train_data[selected_columns], train_data['region'])

print(clf.best_params_)
```

Parametre smo dobili inicijalno Grid Search-om i ručnim menjanjem i testiranjem.

Odabrani parametri:

n_components =	44
covariance_type =	'diag'
tol =	0.0000001
reg_covar =	1e-8
max_iter =	10000

Rezultati

Train skup smo podelili na 60% za train i 40% za test skup

V Measure Score:

Test skup - nedostajuće vrednosti zamenjeni sa medijalnim vrednostima
0.5623789681180245

Test skup - izbačeni redovi sa nedostajućim vrednostima
0.6285316010753187

Test skup sa platforme:
0.741116146111634

Drugi pokušaj:

Parametar n_components postavljen na 4 pošto treba da imamo samo 4 klastera međutim rezultat je iz nekog razloga posta dosta lošiji (nismo stigli da fine tune-ujemo ostale parametre)

Najbolji random state nad splitovanim skupom je sada bio 90

Test skup - 60% train, 40% test
0.4521181858589563

Test skup sa platforme:
0.6996752594191554