ANALIZA VELIKIH PODATAKA

školska 2024/2025 godina

Vežba 7: Klaster analiza i uvod u nenadgledano učenje

U mašinskom učenju razlikujemo dve glavne grupe problema:

- **Nadgledano učenje (supervised learning)**
  Imamo **označene podatke** (npr. recenzije sa poznatim etiketama kao što su „pozitivna“ ili „negativna“) i cilj nam je da predvidimo oznake za nove primere.
- **Nenadgledano učenje (unsupervised learning)**:
  Podaci **nemaju unapred poznate oznake**. Cilj je da **otkrijemo strukturu ili obrazac** unutar podataka.

U nenadgledanom učenju **ne postoji „tačan odgovor“**, već se algoritam koristi da:

- grupiše slične podatke (klaster analiza),
- otkrije neobične (anomalne) primere,
- smanji dimenzionalnost (redukuje broj atributa),
- pronađe obrasce u nestrukturiranim podacima.

Klaster analiza je tehnika kojom se automatski grupišu podaci u **grupe koje sadrže slične podatke**. Svaka grupa naziva se **klaster**.

Glavni cilj:

- Slični primeri budu **u istom klasteru**,
- Različiti primeri budu **u različitim klasterima**.

Primeri primene:

- Grupisanje korisnika na osnovu ponašanja (marketing),
- Segmentacija proizvoda,
- Grupisanje tekstova, slika ili biometrijskih podataka,
- Otkrivanje anomalija (recenzije koje ne liče na druge).

🔢 K-Means algoritam

K-Means je jedan od najpoznatijih i najjednostavnijih algoritama za klaster analizu. Njegova osnovna ideja je da **podeli podatke u K grupa (klastera)** tako da **rastojanje između tačaka u klasteru i centra tog klastera (centroida) bude minimalno**.

📍 Ključni koraci algoritma:

1. **Inicijalizacija**: Nasumično se izaberu K tačaka kao početni centri (centroidi).
2. **Dodeljivanje**: Svaka tačka se dodeljuje najbližem centru (po euklidskom rastojanju).
3. **Ažuriranje centara**: Računa se novi centar svakog klastera kao srednja vrednost svih tačaka u tom klasteru.
4. **Ponavljanje**: Koraci 2 i 3 se ponavljaju dok se centri ne stabilizuju (konvergencija).

✏️ Matematička ideja:

KMeans minimizuje **inerciju** (sumu kvadratnih rastojanja između svake tačke i njenog centra):

$$J = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2$$

gde je C_i zapravo i-ti klaster, a μ_i njegov centroid

⚠️ Važne pretpostavke i ograničenja

- KMeans funkcioniše **samo za numeričke podatke**, jer koristi geometrijska rastojanja.
- Zahteva da unapred znamo broj klastera K (što ponekad nije lako).
- Osetljiv je na **inicijalne vrednosti** i **šum** u podacima.
- Pretpostavlja da su klasteri **sferni i približno iste veličine**, što ne mora biti tačno u realnim podacima.

⚙️ Priprema podataka

Pre klaster analize, podatke treba:

1. **Očistiti** – ukloniti prazne vrednosti.
2. **Normalizovati** – jer različiti opsezi (npr. ocena 1–5 i broj recenzija 0–10,000) mogu uticati na rastojanje. Najčešće se koristi MinMaxScaler koji skalira vrednosti u opseg [0, 1].

📊 Vizuelizacija klastera

Vizuelizacija klastera je važan deo analize jer omogućuje **brzu i intuitivnu interpretaciju rezultata**. Cilj je prikazati kako su podaci raspoređeni u klastere, koje su grupe slične, da li postoje preklapanja ili izdvojene tačke.

Najčešći način vizuelizacije klastera je **scatter plot (dvodimenzionalni grafikon)**, gde:

- **Svaka tačka predstavlja jedan podatak**, pozicioniran u odnosu na dve odabrane osobine.
- **Boje označavaju koji klasteru podatak pripada**, što pruža vizuelno razlikovanje grupa.

Ako su podaci višedimenzionalni (npr. imaju više od dve ili tri osobine), koristimo **tehnike smanjenja dimenzionalnosti**, kao što je:

🔄 PCA – Principal Component Analysis

- Smanjuje kompleksnost podataka tako što pronalazi nove osobine (glavne komponente) koje zadržavaju **najveći deo varijabilnosti podataka**.
- Omogućuje projektovanje podataka na **2D prostor**, što je idealno za vizuelizaciju.
- U praksi se često koristi zajedno sa K-Means algoritmom kako bi se **jasno prikazali rezultati klasterisanja**.

📈 Kako odabrati pravi broj klastera (K)?

Broj klastera **ne treba birati nasumično**. Ako se odabere premalo klastera, različite grupe se mogu spojiti. Ako se odabere previše, moguće je preterano razbijanje podataka na manje delove bez smisla. Postoje metode koje pomažu da se izabere optimalna vrednost K:

📌 Elbow metoda (metoda "lakta")

- Za više vrednosti K (npr. K = 1 do 10), izračunava se **inercija** (suma kvadratnih rastojanja tačaka do svog centra).
- Pravi se graf sa brojem klastera na X osi i inercijom na Y osi.
- Traži se **"lakat" – tačka gde brzina opadanja greške naglo usporava**.
- Ova tačka se smatra idealnim brojem klastera, jer dodatni klasteri više ne donose veliko poboljšanje.

📌 Silueta skor

- Za svaku tačku meri koliko je dobro pozicionirana u svom klasteru u odnosu na susedne klastere.
- Silueta skor se kreće od **-1 do 1**:
  - Blizu **1** → tačka je dobro uklopljena u klaster.
  - Oko **0** → tačka je na granici između klastera.
  - Manje od **0** → tačka je verovatno pogrešno klasifikovana.
- **Prosečna vrednost** svih silueta se koristi kao indikator kvaliteta klasterisanja.

🧠 Interpretacija klastera

Nakon što su podaci uspešno podeljeni u klastere, ključni korak je razumevanje značenja tih klastera. Sam algoritam ne daje opise klastera – on samo grupiše podatke prema sličnostima. Na nama je da analiziramo i tumačimo rezultate.

Za svaki klaster možemo izračunati:

- **Prosečne vrednosti osobina** (npr. prosečna ocena hotela, broj recenzija, broj slika).
- **Broj članova u klasteru** – koliko podataka pripada toj grupi.
- **Raspodelu podataka** – da li su podaci slični unutar klastera ili veoma različiti?

Na osnovu toga dajemo **značenje klasterima** (npr. „popularni hoteli“, „loše ocenjeni“, „mali broj recenzija“). Tako dobijene grupe (rezultati klasterovanja) mogu poslužiti za:

- Ciljano oglašavanje
- Preporuku sličnih objekata
- Otkrivanje neobičnih obrazaca

🛠️ Primeri primene u praksi

1. **Booking / TripAdvisor** – grupisanje recenzija, segmentacija korisnika prema ponašanju ili ocenjivanju.
2. **Marketing** – segmentacija tržišta na osnovu potrošačkih navika.
3. **Zdravstvo** – grupisanje pacijenata sa sličnim simptomima radi bolje dijagnoze.
4. **Obrazovanje** – klasterisanje učenika po stilu učenja ili nivou uspeha za personalizaciju nastave.
5. **Finansije** – otkrivanje obrazaca trošenja, grupa klijenata, pa čak i prevara.

Osim **K-Means** algoritma, postoji više drugih metoda klasterovanja koje se koriste u nenadgledanom učenju, u zavisnosti od vrste i prirode podataka. Dve najpoznatije su:

🔷 1. Hierarhijsko klasterovanje (Hierarchical Clustering)

Gradi hijerarhiju klastera (npr. kao drvo) spajanjem ili razdvajanjem klastera.

Osnovne metode:

- **Aglomerativno** (bottom-up): Svaka tačka počinje kao poseban klaster i postepeno se spajaju.
- **Divizivno** (top-down): Počinje sa jednim velikim klasterom i postepeno ga deli.

Prednosti:

- Ne mora se unapred zadati broj klastera.
- Može se vizuelizovati kao **dendrogram** – drvo koje pokazuje redosled spajanja.

Nedostaci:

- Teško skalira na velike skupove podataka.
- Osetljivo na šum i anomalije.

🔷 2. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Klasteri su oblasti visoke gustine podataka. Odbacuju se tačke koje se nalaze u niskoj gustini (anomalije).

Osobine:

- Klasteri mogu biti nepravilnog oblika.
- Nije potrebno zadavati broj klastera.
- Detektuje **outliere (šum)** automatski.

Ključni parametri:

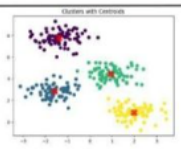
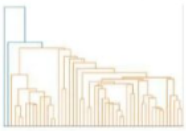
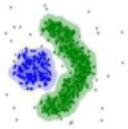
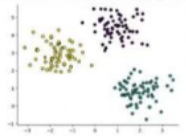
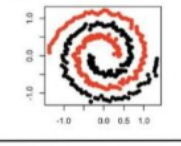
- eps: maksimalna udaljenost za susedstvo tačke
- min_samples: minimalan broj tačaka da bi se oblast smatrala klasterom

Prednosti:

- Odličan za otkrivanje anomalija.
- Radi dobro kod klastera različitih oblika.

Nedostaci:

- Teže odabrati prave parametre.
- Ne radi dobro kad su gustine klastera različite.

Representation	Algorithm Name	Hyperparameter
	K-Means Clustering	Partitions data into K clusters by minimizing variance within each cluster.
	Hierarchical Clustering	Builds a hierarchy of clusters by iteratively merging or splitting existing groups.
	DBSCAN	Forms clusters based on density; groups densely packed points and marks outliers.
	Mean Shift	Finds clusters by locating and adapting to the centroids of data points.
	Spectral Clustering	Uses eigenvalues of similarity matrix to reduce dimensions before clustering.

🔷 **Mean Shift** je metoda klasterovanja koja ne zahteva unapred definisan broj klastera, već automatski pronalazi gustinske vrhove u prostoru podataka. Funkcioniše tako što "pomeranjem sredine" (centra) identifikuje najgušće oblasti – odnosno centre klastera. Koristi kernel funkciju (najčešće Gaussovu) za procenu gustine u okolini svake tačke i iterativno pomera tačke prema lokalnim maksimumima gustine. Pogodna je za skupove podataka gde klasteri nisu nužno sferični niti jednake veličine.

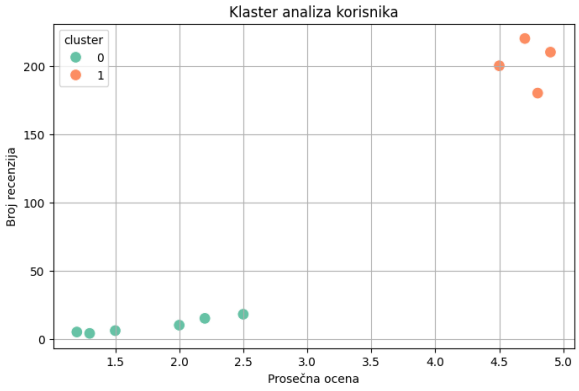
🔷 **Spectral Clustering** koristi matematičke principe iz teorije grafova i linearne algebre za identifikaciju klastera. Umesto da se oslanja samo na udaljenosti, on koristi matricu sličnosti između podataka kako bi izgradio graf i zatim analizira sopstvene vrednosti Laplasijana tog grafa (otuda naziv „spektralno“). Ova metoda je posebno korisna kada klasteri imaju nepravilne oblike ili kada se nalaze u zamršenim odnosima u višedimenzionalnom prostoru. Spectral Clustering je moćna tehnika, ali zahteva odgovarajuće podešavanje matrice sličnosti i može biti manje efikasan za vrlo velike skupove podataka.

K-Means primer: Grupisanje korisnika na osnovu ocena i broja recenzija

Ovaj primer ilustruje kako možemo primeniti K-Means algoritam za klasterisanje gostiju hotela na osnovu njihove prosečne ocene i broja recenzija koje su ostavili.

```
# Instalacija biblioteke
!pip install matplotlib seaborn scikit-learn

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans

# 1. Simulirani skup podataka
data = {
    'user_id': range(1, 11),
    'average_rating': [4.5, 4.8, 1.2, 2.0, 4.7, 1.5, 2.2, 4.9, 1.3, 2.5],
    'review_count': [200, 180, 5, 10, 220, 6, 15, 210, 4, 18]
}
df = pd.DataFrame(data)

# 2. Normalizacija podataka
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(df[['average_rating', 'review_count']])

# 3. KMeans klasterisanje (npr. K=2)
kmeans = KMeans(n_clusters=2, random_state=42)
df['cluster'] = kmeans.fit_predict(scaled_data)

# 4. Vizuelizacija
plt.figure(figsize=(8, 5))
sns.scatterplot(x='average_rating',y='review_count',
hue='cluster', data=df, palette='Set2', s=100)
plt.title('Klaster analiza korisnika')
plt.xlabel('Prosečna ocena')
plt.ylabel('Broj recenzija')
plt.grid(True)
plt.show()

# 5. Pregled rezultata
print(df)
```

	user_id	average_rating	review_count	cluster
0	1	4.5	200	1
1	2	4.8	180	1
2	3	1.2	5	0
3	4	2.0	10	0
4	5	4.7	220	1
5	6	1.5	6	0
6	7	2.2	15	0
7	8	4.9	210	1
8	9	1.3	4	0
9	10	2.5	18	0



Korisnici će biti podeljeni u 2 grupe:

- Jedna grupa sa visokim prosekom i mnogo recenzija,
- Druga sa niskim ocenama i malo recenzija.