VJEŽBA 5: NENADZIRANO UČENJE - GRUPIRANJE PODATAKA.

I. Cilj vježbe: *Upoznati se s algoritmom k srednjih vrijednosti za grupiranje podataka.*

II. Opis vježbe:

Nenadzirano učenje (engl. *unsupervised learning*) je vrsta strojnog učenja gdje je cilj izvući zaključke o raspoloživom skupu podataka koji se sastoji samo od ulaznih veličina, bez odgovarajuće izlazne veličine (npr. otkriti interesantna svojstva). Najčešće se analiza podataka provodi u skladu sa sljedećim pitanjima:

- Mogu li se otkriti grupe u podacima?
- Može li se otkriti skrivena struktura u podacima?
- Mogu li se podaci predstaviti na drugačiji način?
- Mogu li se podaci efikasno komprimirati

Dva najvažnija problema nenadziranog učenja:

- Grupiranje podataka (engl. *clustering*)
- Smanjivanje dimenzionalnosti (engl. *dimensionality reduction*)

II.1. Grupiranje podataka *Kmeans* algoritmom

Grupiranje podataka ili klaster analiza jedan je najčešćih problema nenadziranog strojnog učenja. Koristi se kako bi se pronašle grupe ili skrivene zakonitosti i obrasci u podacima odnosno pokušava se naučiti optimalna podjela podataka. Podaci su neoznačeni - jednom kada se pronađu grupe u podacima moguće je nove mjerne uzorke dodijeliti odgovarajućoj grupi. Primjene su raznolike: segmentacija korisničkog ponašanja (npr. prema povijesti kupovine, aktivnosti u aplikaciji i sl.), detektiranje "botova" i anomalija, *text mining*, obrada medicinskih slika, segmentacija slika, sustavi preporuka itd. U ovoj vježbi razmatra se problem nenadgledanog učenja gdje je cilj grupirati podatke koji se sastoje *m* ulaznih veličina $X = [x_1, x_2, \dots, x_m]$. Stoga, svaki podatak je vektor vrijednosti ulaznih i zapisuje se u obliku:

$$\mathbf{x}^{(i)} = \left[x_1^{(i)}, x_2^{(i)}, \dots, x_m^{(i)}\right]^T . \tag{5-1}$$

Skup koji se sastoji od *n* raspoloživih mjernih podataka može se zapisati u matričnom obliku:

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_m^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_m^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(n)} & x_2^{(n)} & \dots & x_m^{(n)} \end{bmatrix} \tag{5-2}$$

Algoritam *k* srednjih vrijednosti (engl. *Kmeans*) je jednostavan i često korišten algoritam grupiranja koji kao rezultat daje *K* klastera koji su zapisani kao *m* dimenzionalni vektori $\mathbf{c}^{(k)}, k = 1, \dots, K$. Algoritam se zasniva na dvije pretpostavke:

- Centar nekog *k*-tog klastera $\mathbf{c}^{(k)}$ je aritmetička sredina svih podataka $\mathbf{x}^{(i)}$ koji pripadaju tom klasteru
- Svaki podatak je bliže svom klasteru nego centrima ostalih klastera

Pomoću navedenih pretpostavki moguće je napisati kriterijsku funkciju algoritma *Kmeans*:

$$J(\mathbf{c}^{(k)}, k = 1, \dots, K; \mathbf{X}) = \sum_{i=1}^{n} \sum_{k=1}^{K} b_k^{(i)} \left\| \mathbf{x}^{(i)} - \mathbf{c}^{(k)} \right\|^2 \tag{5-3}$$

pri čemu je $b_k^{(i)}$ jednak 1 ili 0 ovisno pripada li podatak $\mathbf{x}^{(i)}$ centru $\mathbf{c}^{(k)}$.
U nastavku je dan pseudokod *Kmeans* algoritma koji sa svakom iteracijom smanjuju vrijednost kriterijske funkcije (5-3). Postoje razni načini inicijalizacije centara odnosno određivanje početnih vrijednosti centara. Npr. kao početne vrijednosti cenatara uzima se nasumično *K* podataka iz dostupnog skupa podataka $\mathbf{X}$. Kao kriterij zaustavljanja obično se uzima promjena centara u dvije uzastopne iteracije ili se unaprijed zadaje broj iteracija.

Kmeans algoritam

1. odredi broj centara *K*. Odredi početne vrijednosti centara klastera $\mathbf{c}^{(k)}, k = 1, \dots, K$.

2. Sve dok nije zadovoljen kriterij zaustavljanja

3. Za svaki podatak $\mathbf{x}^{(i)}$ odredi kojem centru (klasteru pripada)

$$b_k^{(i)} = \begin{cases} 1 \; ako \; je \; \left\|\mathbf{x}^{(i)} - \mathbf{c}^{(k)}\right\| \text{ najmanja od svih udaljenosti za } k = 1, \dots, K \\ \text{u suprotnom } 0 \end{cases}$$

4. Osvježi vrijednosti svih centara $k = 1, \dots, K$

$$\mathbf{c}^{(k)} \leftarrow \frac{\sum_{i=1}^{n} b_k^{(i)} \mathbf{x}^{(i)}}{\sum_{i=1}^{n} b_k^{(i)}}$$

Kmeans je iterativna procedura jer jednom kad se izračunaju novi centri $\mathbf{c}^{(k)}$ mijenjaju se i pripadnosti pojeding podatka $\mathbf{x}^{(i)}$ pa se njihovim preračunavanjem opet utječe na centre. Ova dva koraka se izmjenjuju sve dok se vrijednosti centara ne stabiliziraju.

III. Priprema za vježbu:

Nema posebne pripreme za vježbu.

IV. Rad na vježbi:

Riješite dana zadatke.

Zadatak 1

U prilogu vježbe nalazi se funkcija 5.1. koja služi za generiranje umjetnih podataka kako bi se demonstriralo grupiranje podataka. Funkcija prima cijeli broj koji definira željeni broju uzoraka u skupu i cijeli broj (od 1 do 5) koji definira na koji način će se generirati podaci, a vraća generirani skup podataka u obliku numpy polja pri čemu su prvi i drugi stupac vrijednosti prve odnosno druge ulazne veličine za svaki podatak.
Generirajte 500 podataka i prikažite ih na slici. Pomoću scikit-learn ugrađene metode za *kmeans* odredite centre klastera te svaki podatak obojite ovisno o njegovoj pripadnosti pojedinom klasteru (grupi). Nekoliko puta pokrenite napisani kod. Što primjećujete? Što se događa ako mijenjate način kako se generiraju podaci?

Zadatak 2

Scikit-learn *kmeans* metoda vraća i vrijednost kriterijske funkcije (5-3). Za broj klastera od 1 do 20 odredite vrijednost kriterijske funkcije za podatke iz Zadatka 1. Prikažite dobivene vrijednosti pri čemu je na x-osi broj klastera (npr. od 2 do 20), a na y-osi vrijednost kriterijske funkcije. Kako komentirate dobivene rezultate? Kako biste pomoću dobivenog grafa odredili optimalni broj klastera?

Zadatak 3

Primijenite scikit-learn *kmeans* metodu za kvantizaciju boje na slici. Skripta 5.2. iz priloga vježbe učitava sliku u sivim tonovima koja dolazi kao prilog ovoj vježbi (`example_grayscale.png`). Dodajte kod na TODO mjesta koji će izvršiti kvantizaciju boje primjenom scikit-learn *kmeans*. Prikažite kvantiziranu sliku. Mijenjajte broj klastera. Što primjećujete? Izračunajte kolika se kompresija ove slike može postići ako se za zapis slike koristi samo 4 klastera.

Zadatak 4

Na temelju rješenja prethodnog zadatka primijenite scikit-learn *kmeans* metodu za kvantizaciju boje na slici `example.png` koja dolazi kao prilog ovoj vježbi. Prikažite originalnu i kvantiziranu sliku.

V. Izvještaj s vježbe

Kao izvještaj s vježbe prihvaća se web link na repozitorij pod nazivom `PSU_LV`.

VI. Dodatak

Funkcija 5.1. – generiranje podataka

```
from sklearn import datasets
import numpy as np

def generate_data(n_samples, flagc):

    if flagc == 1:
        random_state = 365
        X,y = datasets.make_blobs(n_samples=n_samples, random_state=random_state)

    elif flagc == 2:
        random_state = 148
        X,y = make_blobs(n_samples=n_samples, random_state=random_state)
        transformation = [[0.60834549, -0.63667341], [-0.40887718, 0.85253229]]
        X = np.dot(X, transformation)

    elif flagc == 3:
        random_state = 148
        X, y = make_blobs(n_samples=n_samples,
                          centers=4,
                          cluster_std=[1.0, 2.5, 0.5, 3.0],
                          random_state=random_state)

    elif flagc == 4:
        X, y = datasets.make_circles(n_samples=n_samples, factor=.5, noise=.05)

    elif flagc == 5:
        X, y = datasets.make_moons(n_samples=n_samples, noise=.05)

    else:
        X = []

    return X
```

Kod 5.2. – kvantizacija boje

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as image
from sklearn.cluster import KMeans

# ucitaj sliku
img = image.imread("example_grayscale.png")

# prikazi sliku
plt.figure()
plt.title('Original image')
plt.imshow(img, cmap='gray')
plt.show()


# TODO: predstavi sliku kao vektor



# TODO: primijeni K-means na vektor (sliku)



# TODO: zamijeni svjetlinu svakog piksela s najblizim centrom



# TODO: prikazi dobivenu aproksimaciju (sliku)
```