Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кафедра системотехніки

Дисципліна: «Методи та системи штучного інтелекту»

ПРАКТИЧНА РОБОТА № 5

«Кластеризація даних за допомогою метода k-means»

Виконала:
ст. гр. ІТКН-18-4
Левченко А.С.

Прийняв:
к.т.н., ст. викл. каф.СТ
Жернова П.Є.

Харків 2020

Мета роботи:

Отримання практичних навичок при кластеризації даних з використанням методу машинного навчання, а саме методу k-means.

Хід роботи:

1. Необхідно імпортувати необхідні пакети і класи: Pandas, Numpy, Sklearn, matplotlib.pyplot.

1. Імпорт необхідних пакетів

In [1]:
```
import sklearn as sk
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Рисунок 1 – Імпорт необхідних пакетів та класів

2. Імпортувати набір даних для подальшої роботи.

Для завантаження .csv файлу з даними в pandas використовується функція read_csv (). Далі розбиваємо наші дані на частини за допомогою функції т записуємо до змінної X (дані з останніх двох таблиць).

2. Імпорт набору даних для подальшої роботи

In [2]:
```
dataset = pd.read_csv('Mall_Customers.csv')
X = dataset.iloc[:, [3, 4]].values
```

Рисунок 2 – Імпорт набору даних для подальшої роботи

3. Використовуючи метод ліктьовий метод (Elbow method) знайти оптимальну кількість кластерів для набору даних.

На відміну від завдання класифікації або регресії, в разі кластеризації складніше вибрати критерій, за допомогою якого було б просто уявити завдання кластеризації як задачу оптимізації.

Якщо справжня мітка заздалегідь не відома (як в нашому випадку), то k-means clustering можна оцінити за допомогою критерію ліктя або коефіцієнта силуету. Ідея методу ліктя полягає в тому, щоб виконати кластеризацію k-

середніх по заданому набору даних для діапазону значень k (num_clusters, наприклад k = 1-10) і для кожного значення k обчислити суму квадратів помилок (SSE).

3. За допомогою ліктьового методу знаходимо оптимальну кількість скупчень

In [3]:
```
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

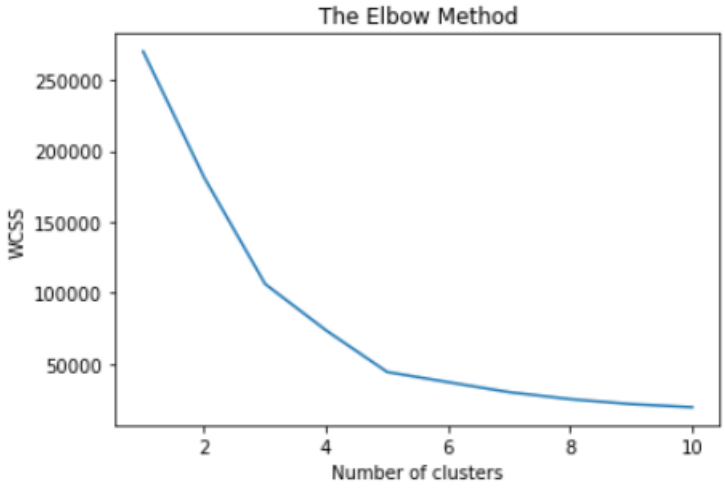

Рисунок 3 – Пошук оптимальної кількості кластерів для набору даних з використанням ліктьового методу

На рис.3 бачимо, що WCSS падає сильно при збільшенні числа кластерів з 1 до 2 і з 2 до 4 і з 4 до 5 і вже не так сильно - при зміні з 5 до 6. Значить, в цьому завданню оптимально задати 5 кластерів.

4. Провести класетеризацію набору даних.

Точки даних стають кластерізованний на підставі подібності їхніх ознак. Наведемо деякі результати алгоритму кластеризації засобів:

– ви можете використовувати центр кластерів До для маркування нових даних.

– кожному кластеру присвоюються навчальні мітки даних.

Замість того, щоб визначати групи перед вивченням даних, цей алгоритм дозволяє шукати і розуміти органічно сформовані групи.

4. Навчання моделі K-Means на наборі даних

In [4]:
```
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(X)
```

Рисунок 4 – Кластеризація даних за допомогою навчання моделі K-Means на наборі даних

5. Візуалізація кластерів

In [5]:
```
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 100, alpha = 0.9, c = 'yellow', label = 'Centr
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

Рисунок 5 – Візуалізація результатів кластеризації

Центри кластерів представлені у вигляді жовтих кіл, в той час як точки даних відображаються у вигляді кіл інших кольорів.

5. Виконати масштабування даних та провести кластеризацію даних ще раз.

5. Маштабування даних та повторна кластеризація

In [6]:
```
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range=(0,1))
sc.fit(X)
X = sc.transform(X)
print(X)
```

```
[[0.         0.3877551 ]
 [0.         0.81632653]
 [0.00819672 0.05102041]
 [0.00819672 0.7755102 ]
 [0.01639344 0.39795918]
 [0.01639344 0.76530612]
 [0.02459016 0.05102041]
```

Рисунок 6 – Масштабування даних

Навчання моделі після маштабування

In [7]:
```
kmeans_scale = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
y_kmeans_scale = kmeans_scale.fit_predict(X)
```

Рисунок 7 – Навчання моделі кластеризації після масштабування

In [8]:
```
plt.scatter(X[y_kmeans_scale == 0, 0], X[y_kmeans_scale == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans_scale == 1, 0], X[y_kmeans_scale == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_kmeans_scale == 2, 0], X[y_kmeans_scale == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans_scale == 3, 0], X[y_kmeans_scale == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_kmeans_scale == 4, 0], X[y_kmeans_scale == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.scatter(kmeans_scale.cluster_centers_[:, 0], kmeans_scale.cluster_centers_[:, 1], s = 100, alpha = 0.9, c = 'yellow', la
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

Рисунок 8 – Візуалізація результатів кластеризації після маштабування

6. Порівняти результат кластеризації до масштабування даних та після.

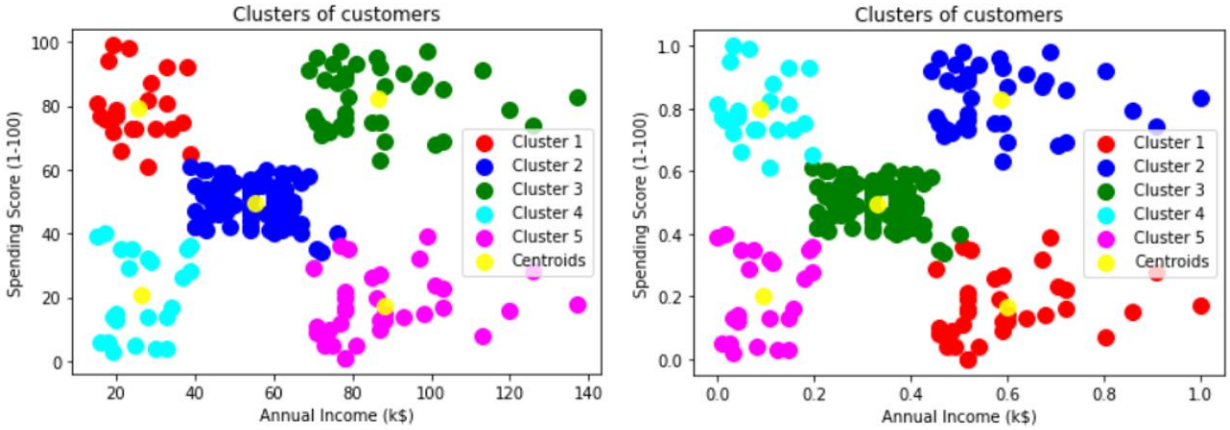

Рисунок 9 – Порівняння візуалізації до масштабування та після (зліва до, праворуч після)

Основна відмінність – зміна кольорів кластеризації. А все тому, що число кластерів та кількість ітерацій не змінилася.

Висновки:

В ході виконання практичної роботи було отримано практичні навички при аналізі даних при кластеризації даних з використанням методу машинного навчання, а саме методу k-means. Метод k-means потребує точну кількість кластерів, для того щоб цей метод працював визначили кластери за допомогою ліктьового методу та побудували графік на якому визначили найліпшу кількість кластерів, а саме п’ять.

Під час візуалізації до та після масштабування даних ніякої різниці крім зміни кольору відображення не визначили.

Основні недоліки:

- трохи складно передбачити кількість кластерів, тобто значення k.
- на вихід сильно впливають вихідні дані, такі як кількість кластерів (значення k)
- порядок даних буде мати сильний вплив на кінцевий результат.
- у кластеризації погано працювати, якщо кластери мають складну геометричну форму.