Peoples Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

AKLI MOHAND OULHADJ UNIVERSITY of BOUIRA

Faculty of Sciences and Applied Sciences

Computer Science Department

TD Project

In Information retrieval systems SRI

Specialty: *ISIL*

Theme

Clustering algorithms – K-Means

Supervised by

DR. Bal Kamal

Realized by

GRINE Lyes

2023/2024

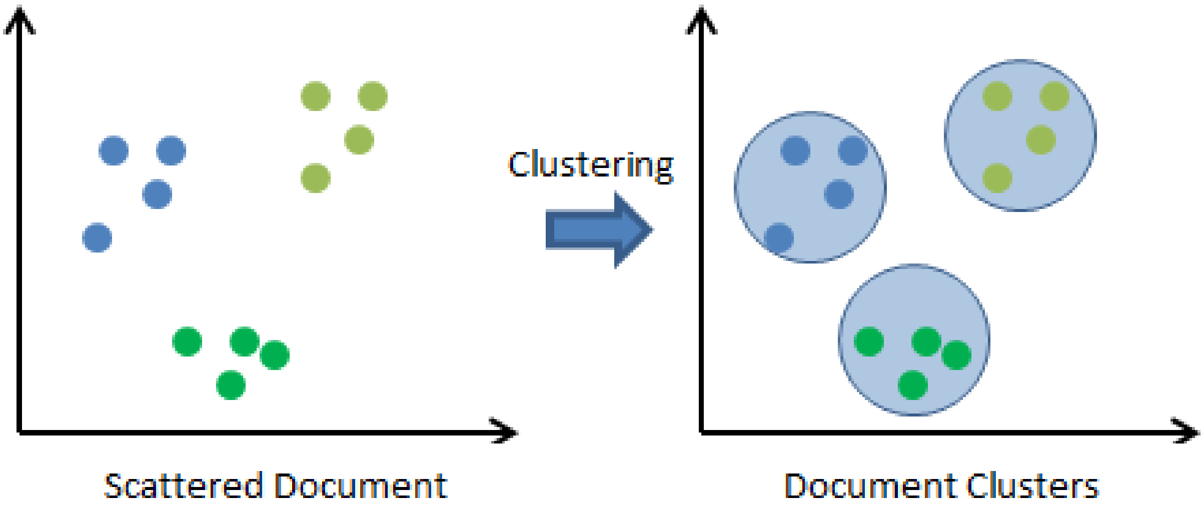
Contents

Introduction

Definition:

Clustering algorithms are a machine-learning technique that discovers patterns and groups together similar data points. It is used to group sets of data points into a number of clusters, which helps extract underlying patterns in data and transforms raw data into meaningful knowledge. [1]

Clustering algorithms are unsupervised learning approaches that group comparable data points into clusters based on their similarity. The identification of such clusters leads to segmentation of data points into a number of distinct groups. [1]



Types of clustering algorithms:

There are several types of clustering algorithms, like centroid-based, density-based, distribution-based, and hierarchical clustering. [1]

Each type of clustering algorithm is best suited to a particular data distribution and has its own advantages and disadvantages. [1]

Examples:

- **Centroid-based clustering** : This algorithm organizes data into non-hierarchical clusters. The most widely used centroid-based clustering algorithm is **k-means**. [1]
- **Density-based clustering** : This algorithm connects areas of high example density into clusters. It allows for arbitrary-shaped distributions as long as dense areas can be connected. [1]
- **Distribution-based clustering** : This algorithm assumes data is composed of distributions, such as Gaussian distributions. [1]
- **Hierarchical clustering** : This algorithm creates a tree of clusters. Hierarchical clustering is well suited to hierarchical data, such as taxonomies. [1]

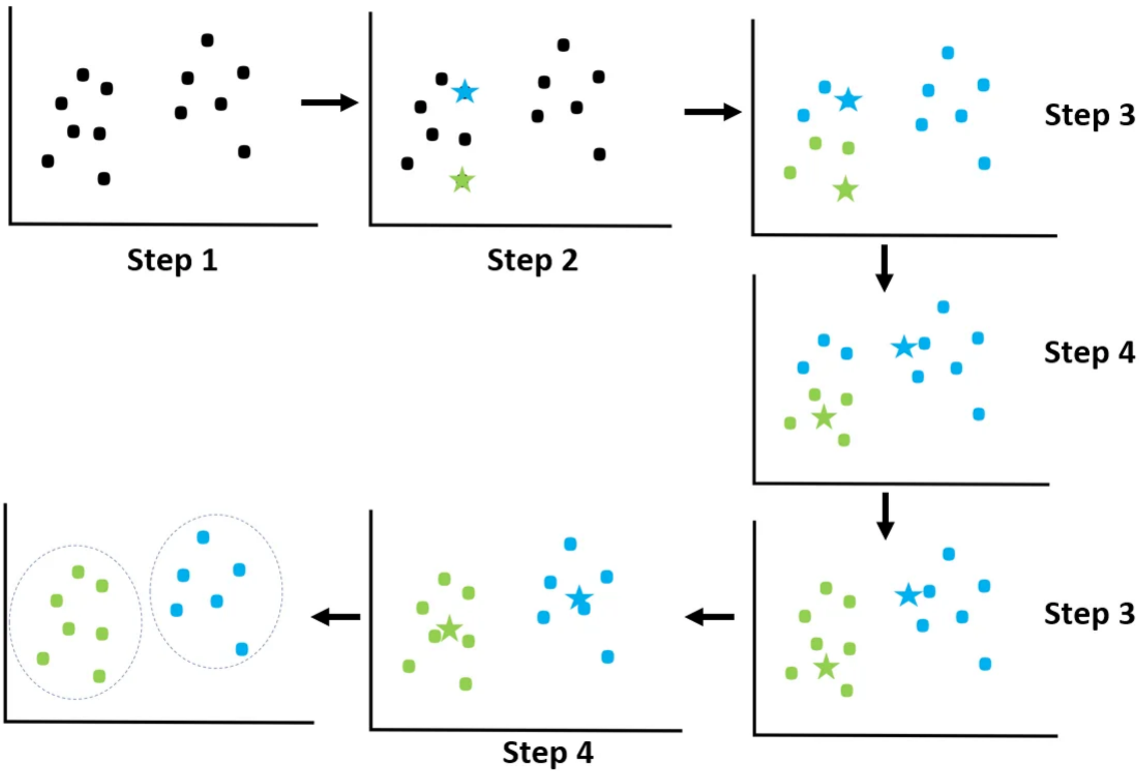
K-Means

The K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. [1]

Were K defines the number of pre-defined clusters that need to be created in the process, so if K=5, there will be five clusters, and so on. [1]

K-Means Steps:

1. Select the number of clusters K.
2. Select random K points or centroids.
3. Assign each data point to their closest centroid.
4. Calculate the variance and place a new centroid of each cluster.
5. Repeat Step 3 and Step 4 for N number of times.
6. Stop if you loop N times or if the values of Step 4 do not change.



Optimal Value of K (Elbow Method):

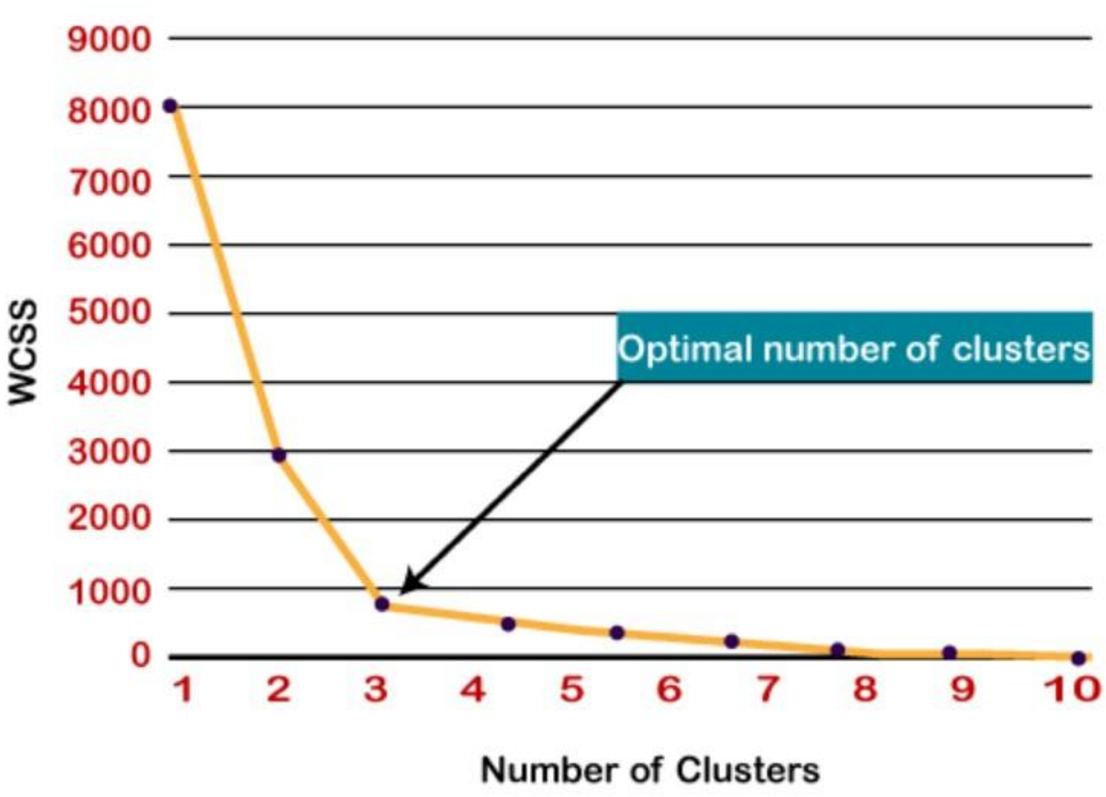
The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS (Within Cluster Sum of Squares) value. [1]

- WCSS= $\sum_{\text{Pi in Cluster1}}$ distance(P_i C_1)^2 +$\sum_{\text{Pi in Cluster2}}$distance($P_i$ C_2)^2 with k=2.

We executes the K-means clustering on a given dataset for different K values (ranges from 1-15).

For each value of K, we calculates the WCSS value.

Plots a curve between calculated WCSS values and the number of clusters K.



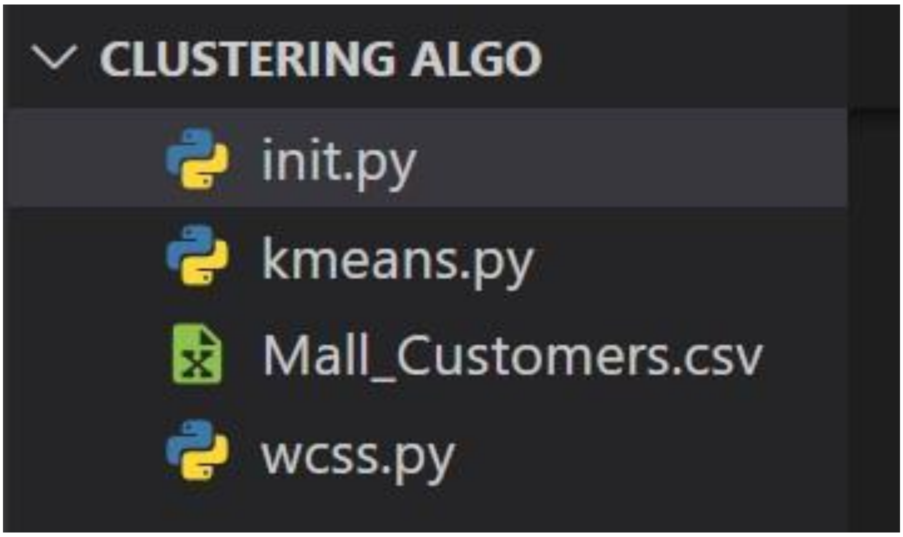
In this graph, the optimal k is Three.

Implementation

Prerequisite:

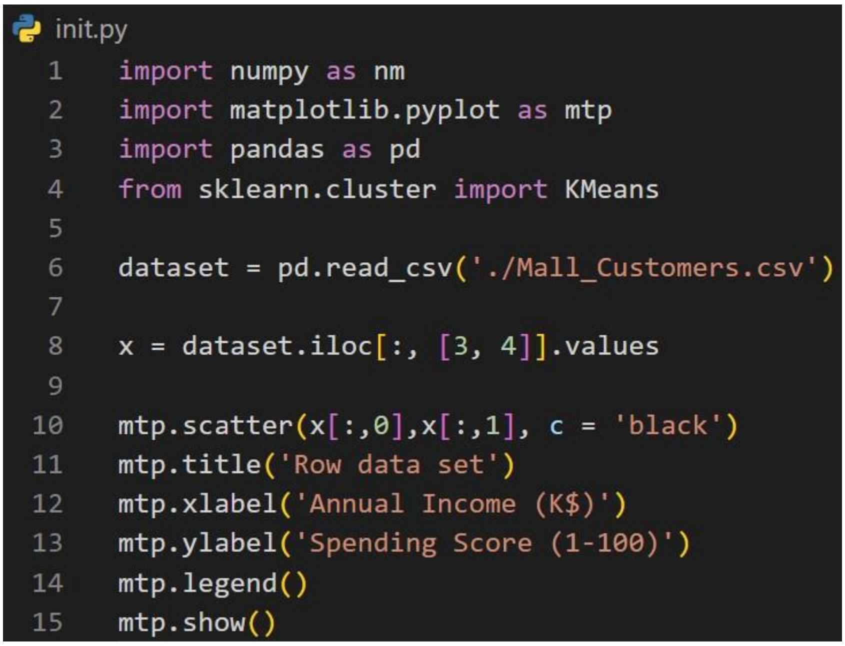
- Python 3.10
- pip install numpy
- pip install matplotlib
- pip install pandas
- pip install sklearn

We will use the free **mall customer csv [2]**, after extracting the .csv file, you can drop it in your working directory, and then we will need three new files.



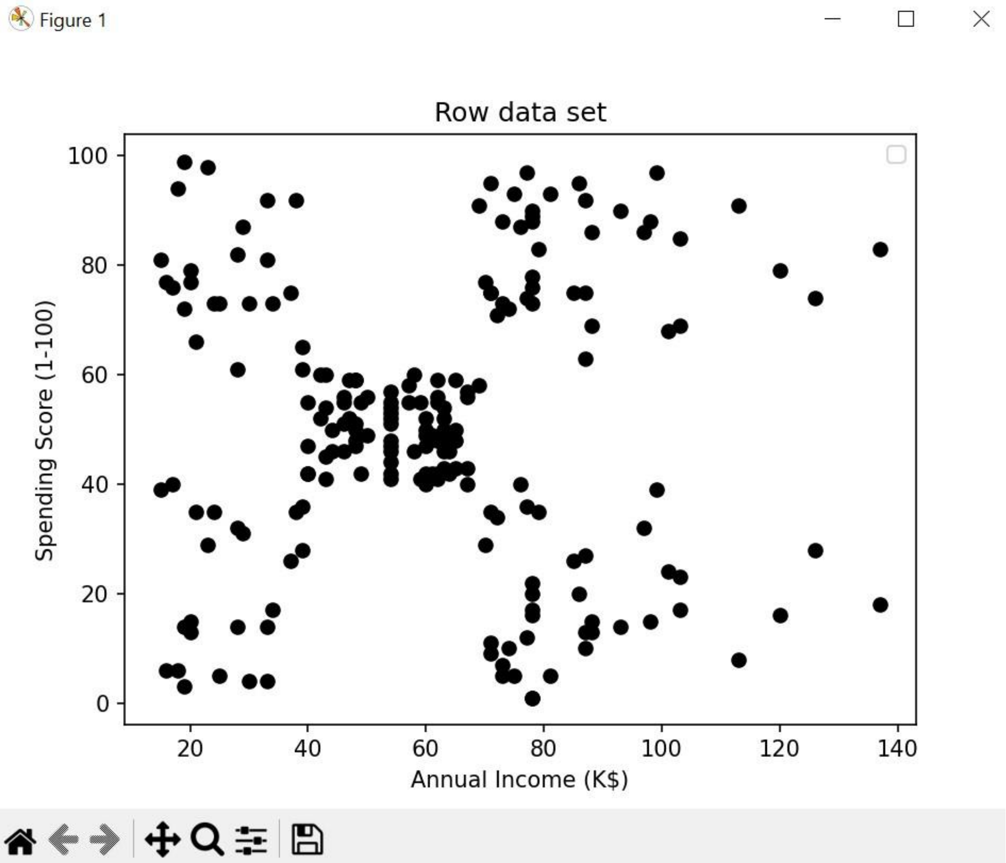
```
CLUSTERING ALGO
    init.py
    kmeans.py
    Mall_Customers.csv
    wcss.py
```

init.py: This file is mandatory, it serve to help us visualize how we transformed the data into a graph.

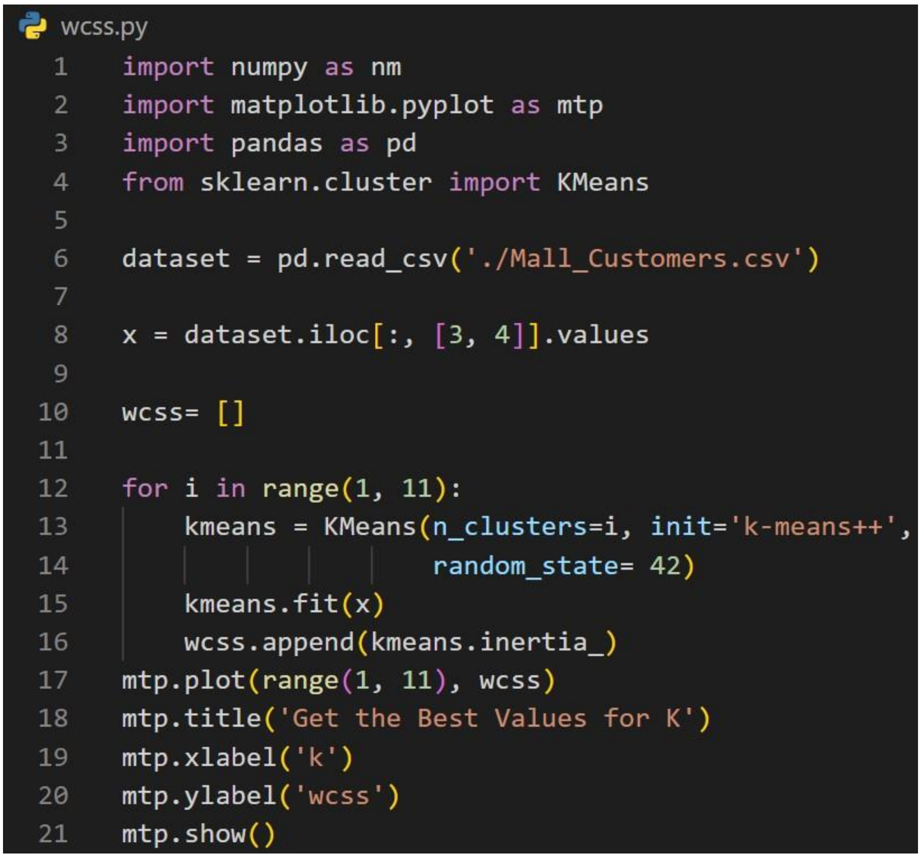


init.py

```python
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
from sklearn.cluster import KMeans

dataset = pd.read_csv('./Mall_Customers.csv')

x = dataset.iloc[:, [3, 4]].values

mtp.scatter(x[:,0],x[:,1], c = 'black')
mtp.title('Row data set')
mtp.xlabel('Annual Income (K$)')
mtp.ylabel('Spending Score (1-100)')
mtp.legend()
mtp.show()
```

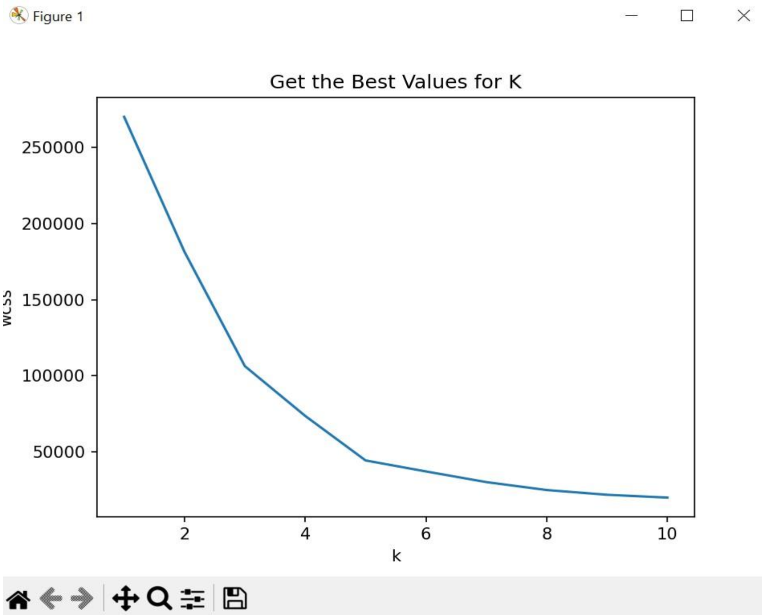


wcss.py: This file is responsible for finding the optimal K using the WSCC method. Is our example the optimal K = 5.

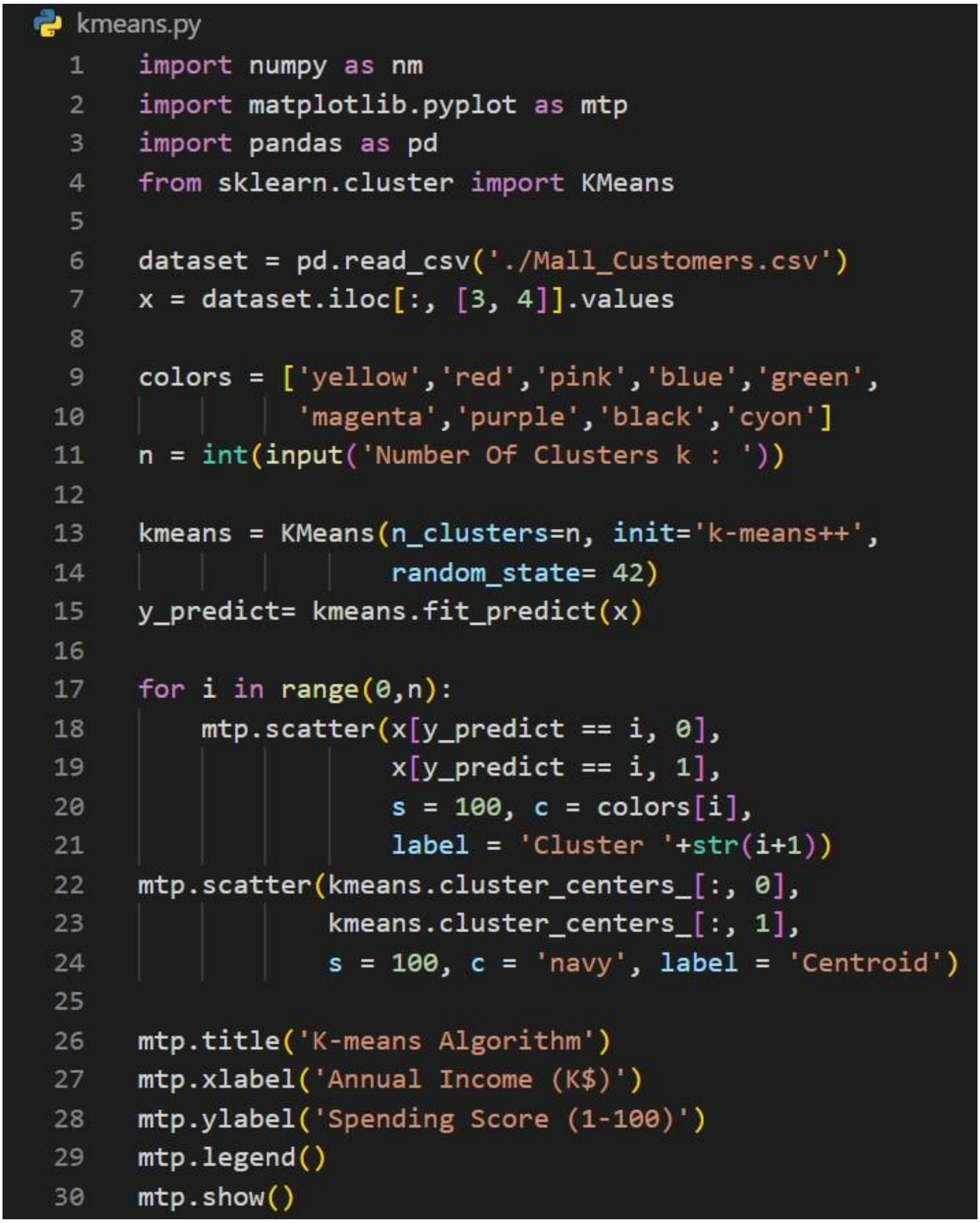


wcss.py

```python
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
from sklearn.cluster import KMeans

dataset = pd.read_csv('./Mall_Customers.csv')

x = dataset.iloc[:, [3, 4]].values

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_)
mtp.plot(range(1, 11), wcss)
mtp.title('Get the Best Values for K')
mtp.xlabel('k')
mtp.ylabel('wcss')
mtp.show()
```

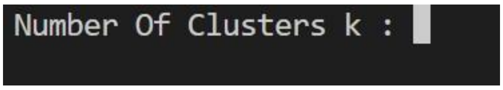


kmeans.py: This file is our main algorithm for transforming the date into relevant clusters.



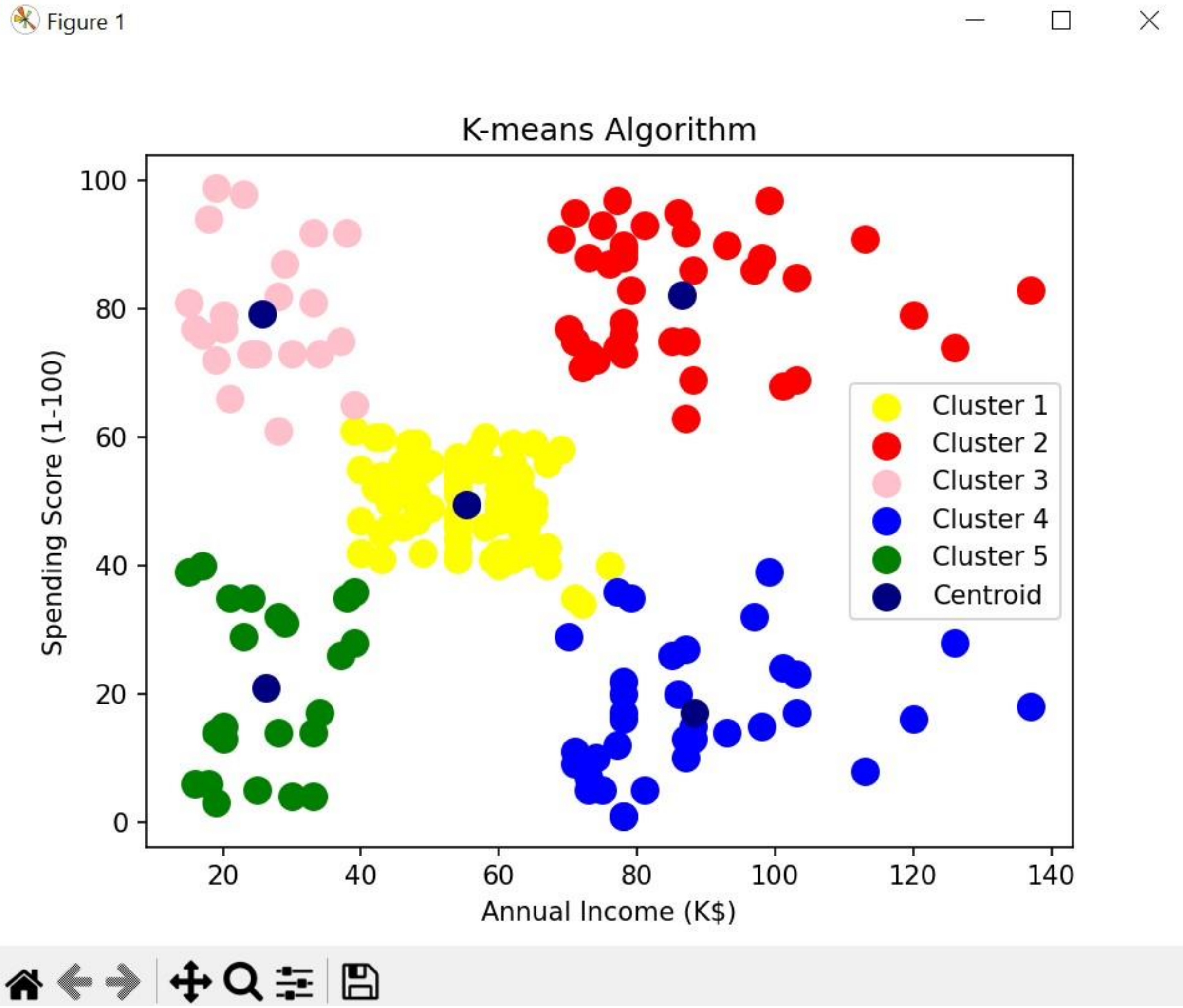
```python
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
from sklearn.cluster import KMeans

dataset = pd.read_csv('./Mall_Customers.csv')
x = dataset.iloc[:, [3, 4]].values

colors = ['yellow','red','pink','blue','green',
     'magenta','purple','black','cyon']
n = int(input('Number Of Clusters k : '))

kmeans = KMeans(n_clusters=n, init='k-means++',
        random_state= 42)
y_predict= kmeans.fit_predict(x)

for i in range(0,n):
  mtp.scatter(x[y_predict == i, 0],
        x[y_predict == i, 1],
        s = 100, c = colors[i],
        label = 'Cluster '+str(i+1))
mtp.scatter(kmeans.cluster_centers_[:, 0],
      kmeans.cluster_centers_[:, 1],
      s = 100, c = 'navy', label = 'Centroid')

mtp.title('K-means Algorithm')
mtp.xlabel('Annual Income (K$)')
mtp.ylabel('Spending Score (1-100)')
mtp.legend()
mtp.show()
```

You will be prompted for the K values, which is 5.



```
Number Of Clusters k : 
```

Type 5 and then hit enter.



As you can see, we have five different clusters.

Bibliography

[1] ChatGPT , https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning

[2] https://gist.github.com/pravalliyaram/5c05f43d2351249927b8a3f3cc3e5ecf