EXPERMINT: 07

- **Aim: Implementation of Clustering algorithm (K-means/K-medoids).**
- **Theory:**

K-Means Clustering:

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

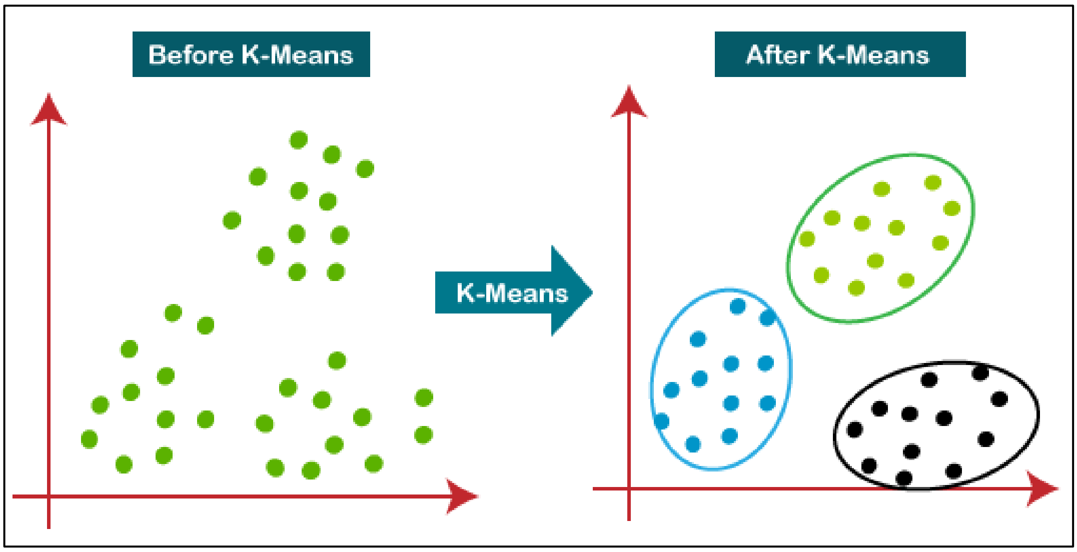
It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.



How does the K-Means Algorithm Work?

- The working of the K-Means algorithm is explained in the below steps:
- Step-1: Select the number K to decide the number of clusters.
- Step-2: Select random K points or centroids. (It can be other from the input dataset).
- Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.
- Step-4: Calculate the variance and place a new centroid of each cluster.
- Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.
- Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.
- Step-7: The model is ready.

Advantages of k-means:

Simple: It is easy to implement k-means and identify unknown groups of data from complex data sets.

Flexible: K-means algorithm can easily adjust to the changes. If there are any problems, adjusting the cluster segment will allow changes to easily occur on the algorithm.

Suitable in a large dataset: K-means is suitable for a large number of datasets and it’s computed much faster than the smaller dataset. It can also produce higher clusters.

Efficient: The algorithm used is good at segmenting the large data set. Its efficiency depends on the shape of the clusters. K-means works well in hyper-spherical clusters.

Time complexity: K-means segmentation is linear in the number of data objects thus increasing execution time. It doesn’t take more time in classifying similar characteristics in data like hierarchical algorithms.

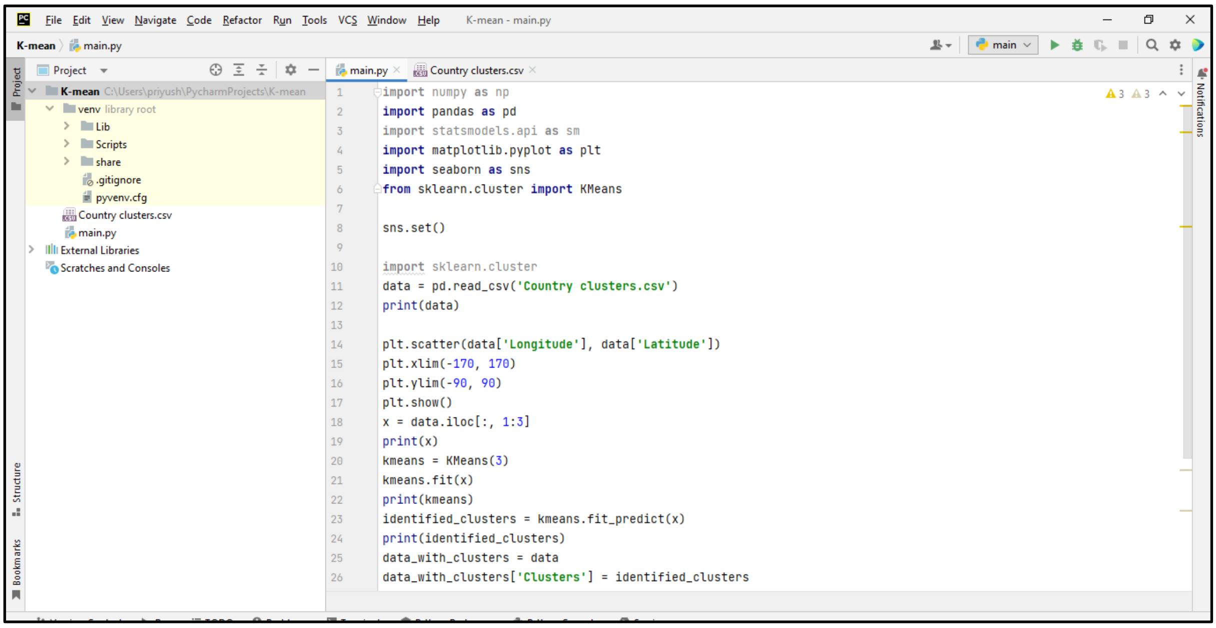
Disadvantages of k-means:

No-optimal set of clusters: K-means doesn’t allow the development of an optimal set of clusters and for effective results, you should decide on the clusters before.

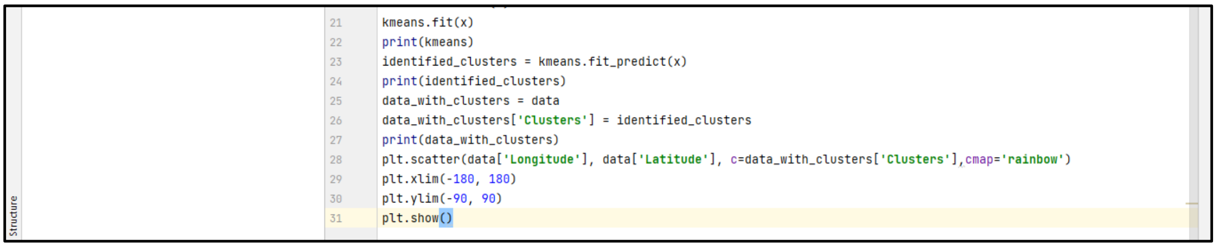
Lacks consistency: K-means clustering gives varying results on different runs of an algorithm. A random choice of cluster patterns yields different clustering results resulting in inconsistency.

Uniform effect: It produces clusters with uniform sizes even when the input data has different sizes.

Source code:

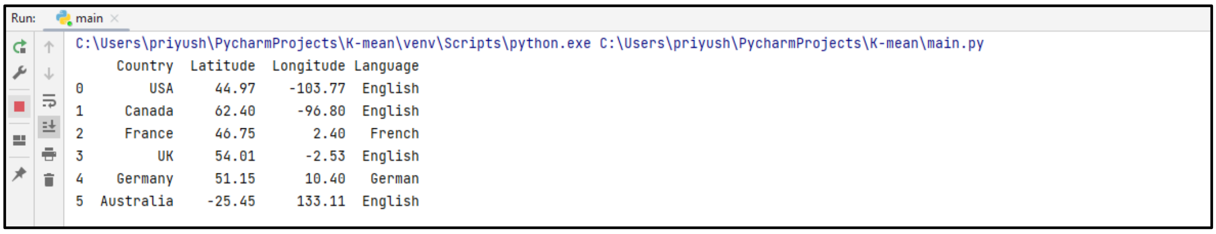


```python
import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans

sns.set()

import sklearn.cluster
data = pd.read_csv('Country clusters.csv')
print(data)

plt.scatter(data['Longitude'], data['Latitude'])
plt.xlim(-170, 170)
plt.ylim(-90, 90)
plt.show()
x = data.iloc[:, 1:3]
print(x)
kmeans = KMeans(3)
kmeans.fit(x)
print(kmeans)
identified_clusters = kmeans.fit_predict(x)
print(identified_clusters)
data_with_clusters = data
data_with_clusters['Clusters'] = identified_clusters
```

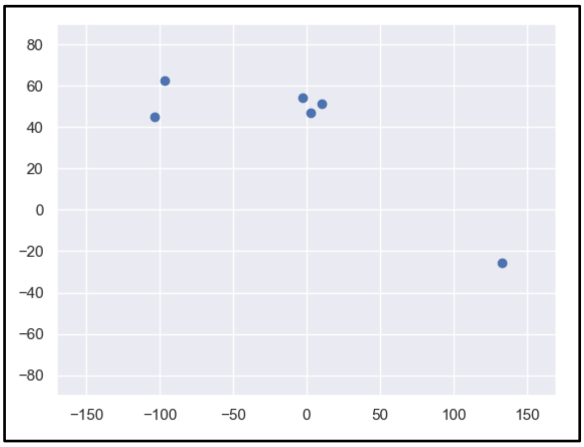


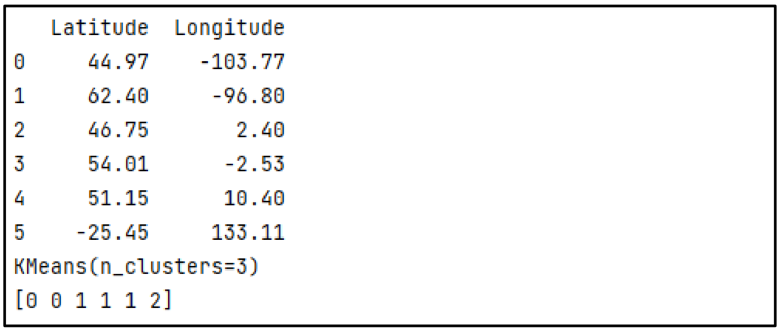
```python
kmeans.fit(x)
print(kmeans)
identified_clusters = kmeans.fit_predict(x)
print(identified_clusters)
data_with_clusters = data
data_with_clusters['Clusters'] = identified_clusters
print(data_with_clusters)
plt.scatter(data['Longitude'], data['Latitude'], c=data_with_clusters['Clusters'],cmap='rainbow')
plt.xlim(-180, 180)
plt.ylim(-90, 90)
plt.show()
```

Output

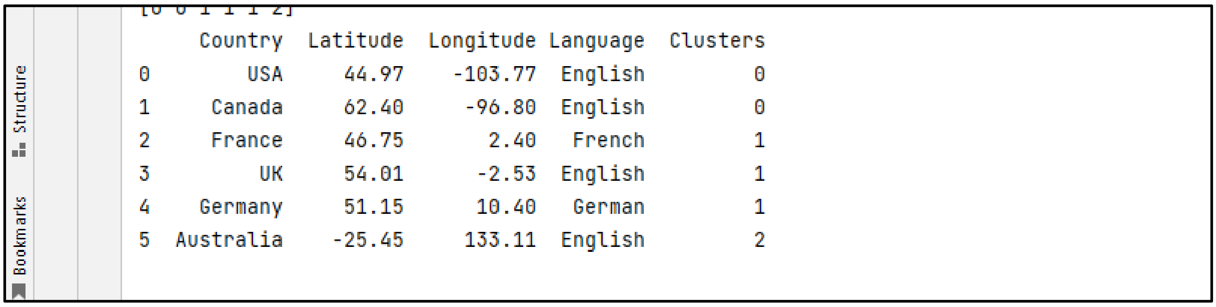


```
C:\Users\priyush\PycharmProjects\K-mean\venv\Scripts\python.exe C:\Users\priyush\PycharmProjects\K-mean\main.py
     Country  Latitude  Longitude Language
0        USA     44.97    -103.77  English
1     Canada     62.40     -96.80  English
2     France     46.75       2.40   French
3         UK     54.01      -2.53  English
4    Germany     51.15      10.40   German
5  Australia    -25.45     133.11  English
```

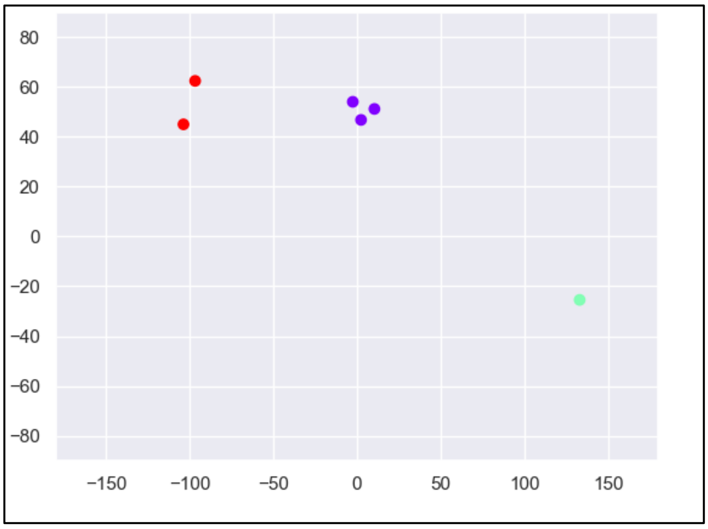




```
    Latitude  Longitude
0      44.97    -103.77
1      62.40     -96.80
2      46.75       2.40
3      54.01      -2.53
4      51.15      10.40
5     -25.45     133.11
KMeans(n_clusters=3)
[0 0 1 1 1 2]
```



```
[0 0 1 1 1 2]
     Country  Latitude  Longitude Language  Clusters
0        USA     44.97    -103.77  English         0
1     Canada     62.40     -96.80  English         0
2     France     46.75       2.40   French         1
3         UK     54.01      -2.53  English         1
4    Germany     51.15      10.40   German         1
5  Australia    -25.45     133.11  English         2
```



Advantages K-Means Algorithm	Disadvantages K-Means Algorithm
High Performance	Result repeatability
Unlabeled Data	Manual Work
Easy to Use	Spherical Clustering Only

● **Conclusion:** K-means clustering is the unsupervised machine learning algorithm that is part of a much deep pool of data techniques and operations in the realm of Data Science. It is the fastest and most efficient algorithm to categorize data points into groups even when very little information is available about data.