AIM: IMPLEMENTATION AND ANALYSIS OF CLUSTERING ALGORITHMS LIKE K-MEANS

THEORY:

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.

What is K-Means Algorithm?

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.
It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.
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.

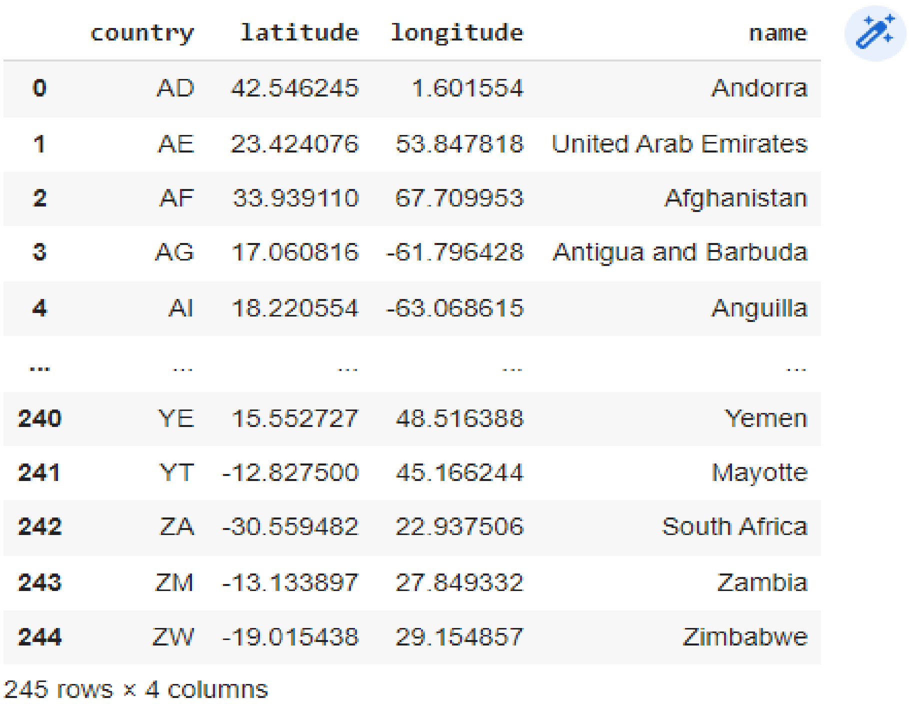
Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

1) IMPORTING LIBRARIES:

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

2) DATA PREPROCESSING:

```
data = pd.read_csv('Countrries.csv')
data = data.replace((np.inf, -np.inf, np.nan), 0).reset_index(drop=True)
data
```

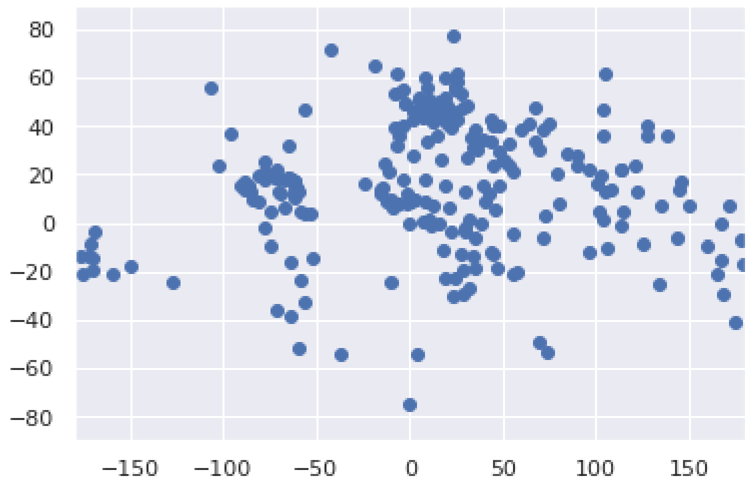


	country	latitude	longitude	name
0	AD	42.546245	1.601554	Andorra
1	AE	23.424076	53.847818	United Arab Emirates
2	AF	33.939110	67.709953	Afghanistan
3	AG	17.060816	-61.796428	Antigua and Barbuda
4	AI	18.220554	-63.068615	Anguilla
...	...	...	...	...
240	YE	15.552727	48.516388	Yemen
241	YT	-12.827500	45.166244	Mayotte
242	ZA	-30.559482	22.937506	South Africa
243	ZM	-13.133897	27.849332	Zambia
244	ZW	-19.015438	29.154857	Zimbabwe

245 rows × 4 columns

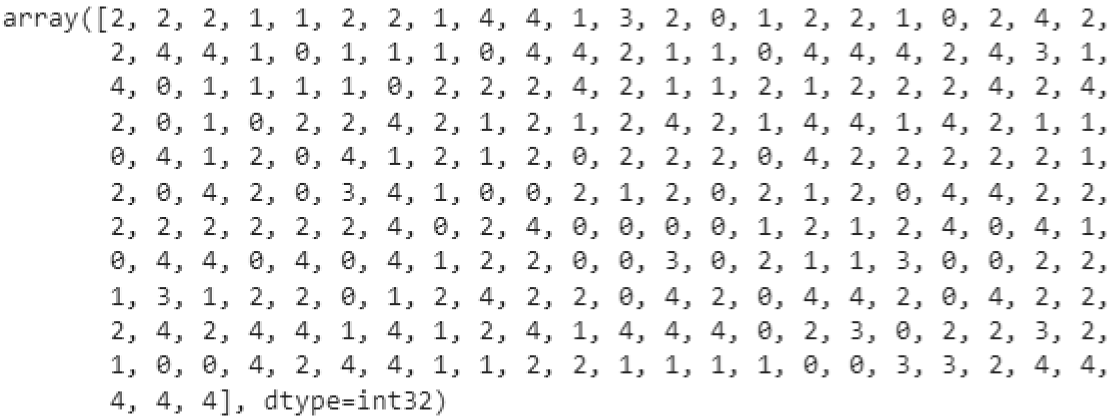
3) **PLOTTING GRAPH:**

```
plt.scatter(data['longitude'],data['latitude'])
plt.xlim(-180,180)
plt.ylim(-90,90)
plt.show()
```



4) **DATA SPLITTING INTO TRAINING DATASET & TESTING DATASET & CREATING CLUSTERS:**

```
x = data.iloc[:,1:3]
kmeans = KMeans(5)
kmeans.fit(x)
identified_clusters = kmeans.fit_predict(x)
identified_clusters
```

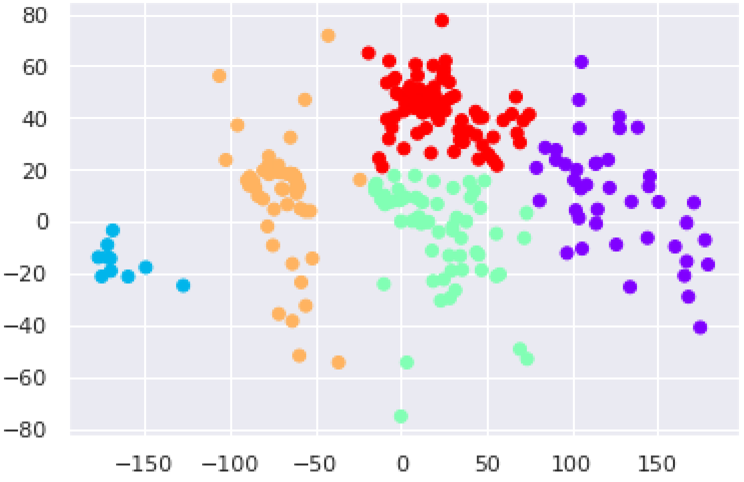


```
array([2, 2, 2, 1, 1, 2, 2, 1, 4, 4, 1, 3, 2, 0, 1, 2, 2, 1, 0, 2, 4, 2,
       2, 4, 4, 1, 0, 1, 1, 1, 0, 4, 4, 2, 1, 1, 0, 4, 4, 4, 2, 4, 3, 1,
       4, 0, 1, 1, 1, 1, 0, 2, 2, 2, 4, 2, 1, 1, 2, 1, 2, 2, 2, 4, 2, 4,
       2, 0, 1, 0, 2, 2, 4, 2, 1, 2, 1, 2, 4, 2, 1, 4, 4, 1, 4, 2, 1, 1,
       0, 4, 1, 2, 0, 4, 1, 2, 1, 2, 0, 2, 2, 2, 0, 4, 2, 2, 2, 2, 2, 1,
       2, 0, 4, 2, 0, 3, 4, 1, 0, 0, 2, 1, 2, 0, 2, 1, 2, 0, 4, 4, 2, 2,
       2, 2, 2, 2, 2, 2, 4, 0, 2, 4, 0, 0, 0, 0, 1, 2, 1, 2, 4, 0, 4, 1,
       0, 4, 4, 0, 4, 0, 4, 1, 2, 2, 0, 0, 3, 0, 2, 1, 1, 3, 0, 0, 2, 2,
       1, 3, 1, 2, 2, 0, 1, 2, 4, 2, 2, 0, 4, 2, 0, 4, 4, 2, 0, 4, 2, 2,
       2, 4, 2, 4, 4, 1, 4, 1, 2, 4, 1, 4, 4, 4, 0, 2, 3, 0, 2, 2, 3, 2,
       1, 0, 0, 4, 2, 4, 4, 1, 1, 2, 2, 1, 1, 1, 1, 0, 0, 3, 3, 2, 4, 4,
       4, 4, 4], dtype=int32)
```

5) **PLOTTING CLUSTERS:**

```
data_with_clusters = data.copy()
data_with_clusters['clusters'] = identified_clusters
plt.scatter(data_with_clusters['longitude'],data_with_clusters['latitude'],c=data_with_clusters['clusters'],cmap='rainbow')
```

```
<matplotlib.collections.PathCollection at 0x7fa01d1aa110>
```



CONCLUSION:
From this practical, I have learned the implementation of k-means in python.