120A3051

Shreya Idate

Batch: E3

Experiment 6

Aim: Implementing K-mean clustering algorithm using Python.

Theory:

- K-means is an unsupervised learning method for clustering data points.
- The algorithm iteratively divides data points into K clusters by minimizing the variance in each cluster.
- First, each data point is randomly assigned to one of the K clusters. Then, we compute the centroid (functionally the center) of each cluster, and reassign each data point to the cluster with the closest centroid.
- We repeat this process until the cluster assignments for each data point are no longer changing.
- K-means clustering requires us to select K, the number of clusters we want to group the data into.
- The elbow method lets us graph the inertia (a distance-based metric) and visualize the point at which it starts decreasing linearly. This point is referred to as the "eblow" and is a good estimate for the best value for K based on our data.

Output:

```
!pip install KMeans
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: KMeans in /usr/local/lib/python3.9/dist-packages (1.0.2)
```

```
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
path='/content/Countryclusters.csv'
```

```
[12] data = pd.read_csv(path,error_bad_lines=False)
     data.iloc[:,:-1]
```

	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

```
x = data.iloc[:,1:3] # 1t for rows and second for columns
x
```

	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

```
[5] kmeans = KMeans(3)
    kmeans.fit(x)
```

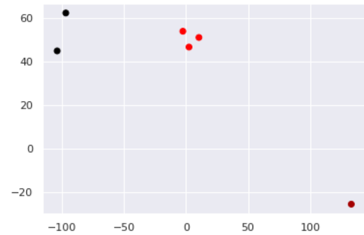
```
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will ch
  warnings.warn(
KMeans(n_clusters=3)
```

```
identified_clusters = kmeans.fit_predict(x)
identified_clusters
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will ch
  warnings.warn(
array([2, 2, 0, 0, 0, 1], dtype=int32)
```

```
[18] 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='flag')
```

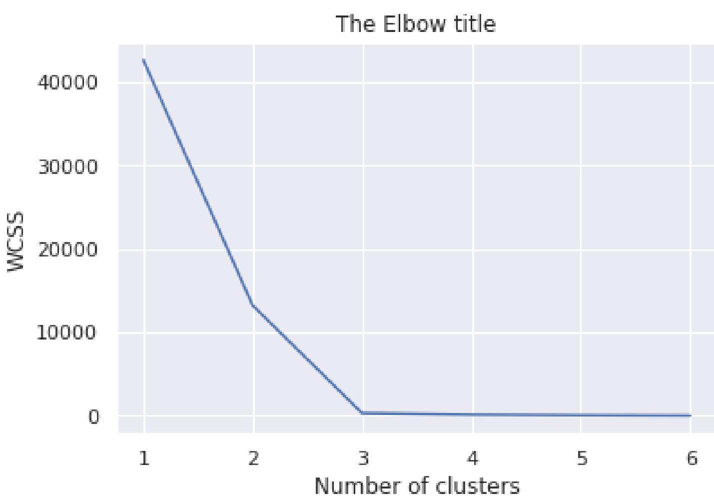
```
<matplotlib.collections.PathCollection at 0x7fda0b39eac0>
```




```
[8]  wcss=[]
     for i in range(1,7):
       kmeans=KMeans(i)
       kmeans.fit(x)
       wcss_iter = kmeans.inertia_
       wcss.append(wcss_iter)

     number_clusters = range(1,7)
     plt.plot(number_clusters,wcss)
     plt.title('The Elbow title')
     plt.xlabel('Number of clusters')
     plt.ylabel('WCSS')
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will ch
  warnings.warn(
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will ch
  warnings.warn(
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will ch
  warnings.warn(
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will ch
  warnings.warn(
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will ch
  warnings.warn(
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will ch
  warnings.warn(
Text(0, 0.5, 'WCSS')
```

Conclusion:

Successfully implemented K-mean clustering algorithm using Python.