Fergusson College (Autonomous), Pune

K-Means Clustering Algorithm

Dataset Name: - MS Admission Prediction

Subject: - Linear Algebra **By: -Purva Deepak Kolhe**

Class: - F.Y. M.Sc. Data Science **Roll No.:- 226518**

Introduction

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 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.

Data Collection

I have taken data from kaggle, which is related to 'MS Admission Prediction'. This file contains 300 students data In which there are 8 columns headed as GRE Score, TOEFL Score, University Rating, SOP, LOR, CGPA, Research, and Chance of Admit. From above I took two data frames are as follows:

1. GRE Score
2. TOEFL Score.

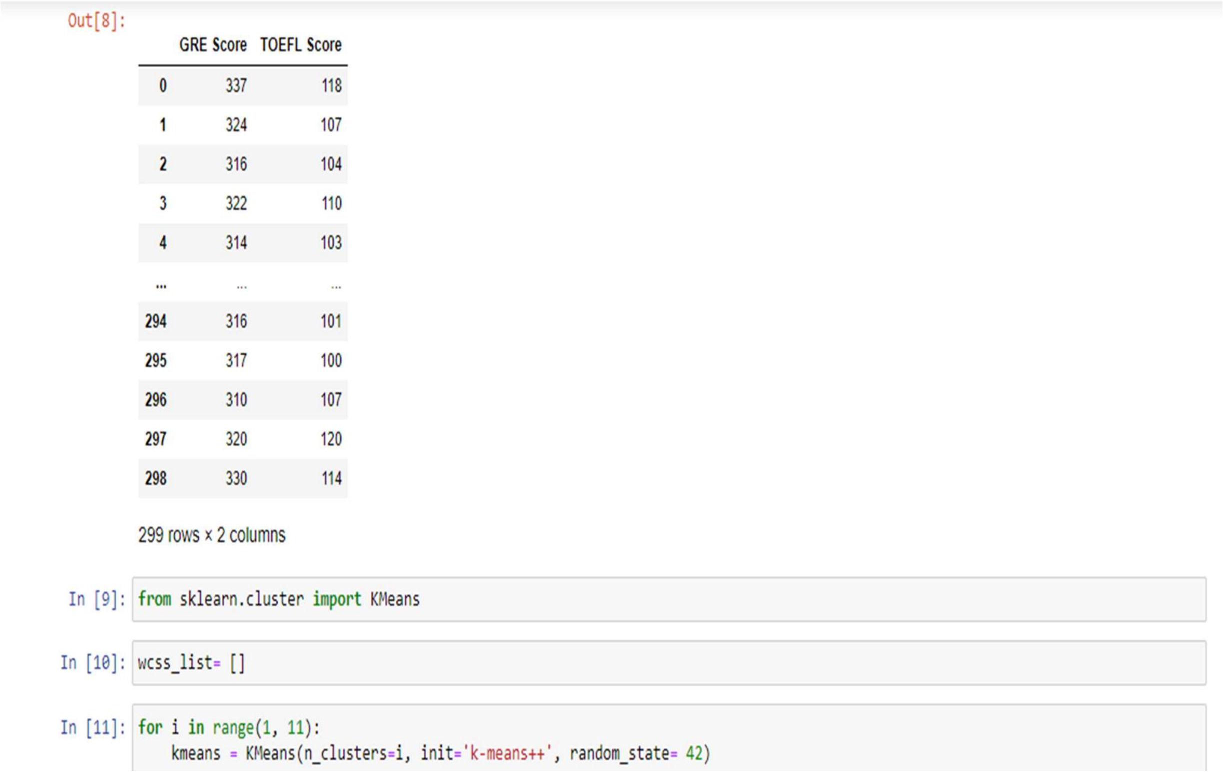
Source: https://www.kaggle.com/datasets/mukeshmanral/graduates-admission-prediction

Actual Implementation

```
import numpy as nm
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv(r"C:\Users\LENOVO\OneDrive\Desktop\Admission_Predict.csv")
df.head()

X = df[['GRE Score','TOEFL Score']]
X

from sklearn.cluster import KMeans
wcss_list= []

for i in range(1, 11):
   kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)
   kmeans.fit(X)
   wcss_list.append(kmeans.inertia_)
```

```
plt.plot(range(1, 11), wcss_list)
plt.title('The Elobw Method Graph')
plt.xlabel('Number of clusters(k)')
plt.ylabel('wcss_list')
plt.show()

plt.scatter(X['GRE Score'],X['TOEFL Score'])

model = KMeans(n_clusters = 4)
model.fit(X)

model.cluster_centers_

cluster_number = model.predict(X)

len(cluster_number)

len(X)

c0 = X[cluster_number==0]
c1 = X[cluster_number==1]
c2 = X[cluster_number==2]
c3 = X[cluster_number==3]


plt.scatter(c0['GRE Score'], c0['TOEFL Score'],c = 'blue')
plt.scatter(c1['GRE Score'], c1['TOEFL Score'],c = 'green')
plt.scatter(c2['GRE Score'], c2['TOEFL Score'],c = 'red')
plt.scatter(c3['GRE Score'], c3['TOEFL Score'],c = 'purple')
plt.scatter(model.cluster_centers_[:, 0], model.cluster_centers_[:,1], s = 100, c = 'yellow', label =
'Centroids')
plt.title('Clusters of Admission prediction')
plt.xlabel('GRE Score(0-340)')
plt.ylabel('TOEFL Score(0-120)')
plt.legend()
plt.show()
```

Output:



Out[8]:

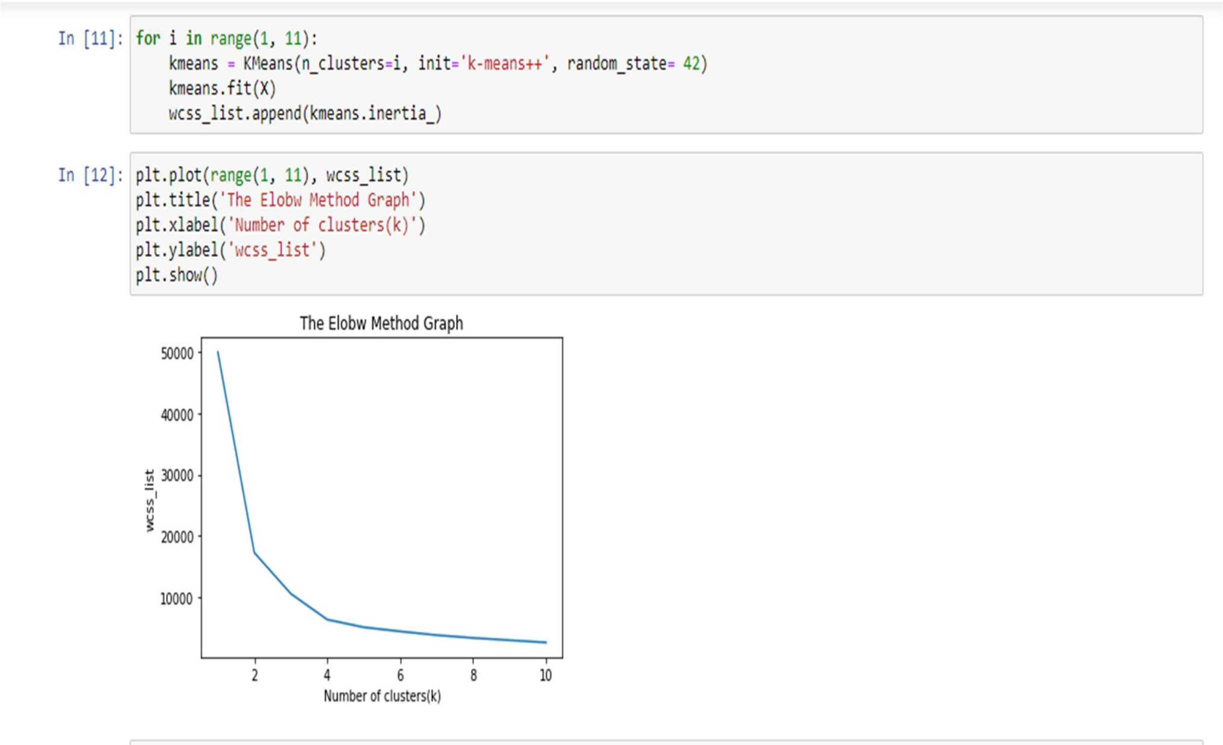
	GRE Score	TOEFL Score
0	337	118
1	324	107
2	316	104
3	322	110
4	314	103
...	...	...
294	316	101
295	317	100
296	310	107
297	320	120
298	330	114

299 rows × 2 columns

In [9]:
```
from sklearn.cluster import KMeans
```

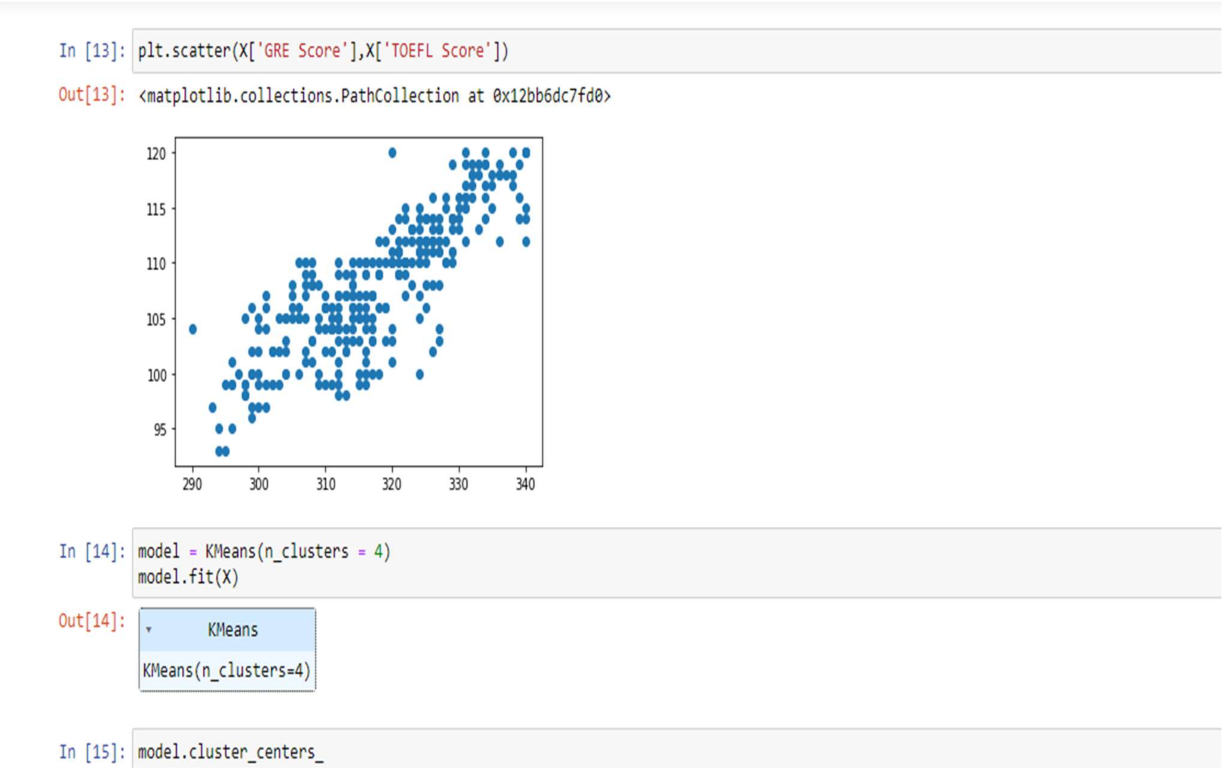
In [10]:
```
wcss_list= []
```

In [11]:
```
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)
```



In [11]:
```
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)
    kmeans.fit(X)
    wcss_list.append(kmeans.inertia_)
```

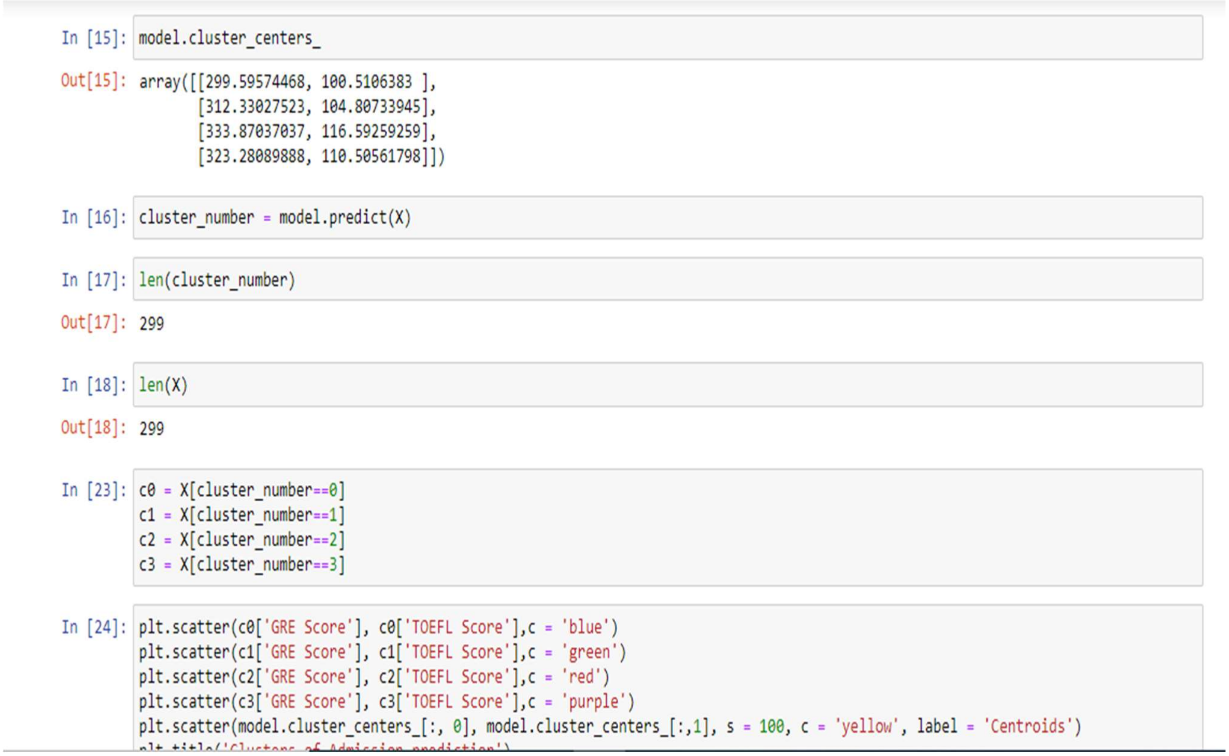
In [12]:
```
plt.plot(range(1, 11), wcss_list)
plt.title('The Elobw Method Graph')
plt.xlabel('Number of clusters(k)')
plt.ylabel('wcss_list')
plt.show()
```



```
In [13]: plt.scatter(X['GRE Score'],X['TOEFL Score'])

Out[13]: <matplotlib.collections.PathCollection at 0x12bb6dc7fd0>
```

```
In [14]: model = KMeans(n_clusters = 4)
         model.fit(X)

Out[14]: ▾ KMeans
         KMeans(n_clusters=4)
```

```
In [15]: model.cluster_centers_
```



```
In [15]: model.cluster_centers_

Out[15]: array([[299.59574468, 100.5106383 ],
                [312.33027523, 104.80733945],
                [333.87037037, 116.59259259],
                [323.28089888, 110.50561798]])
```

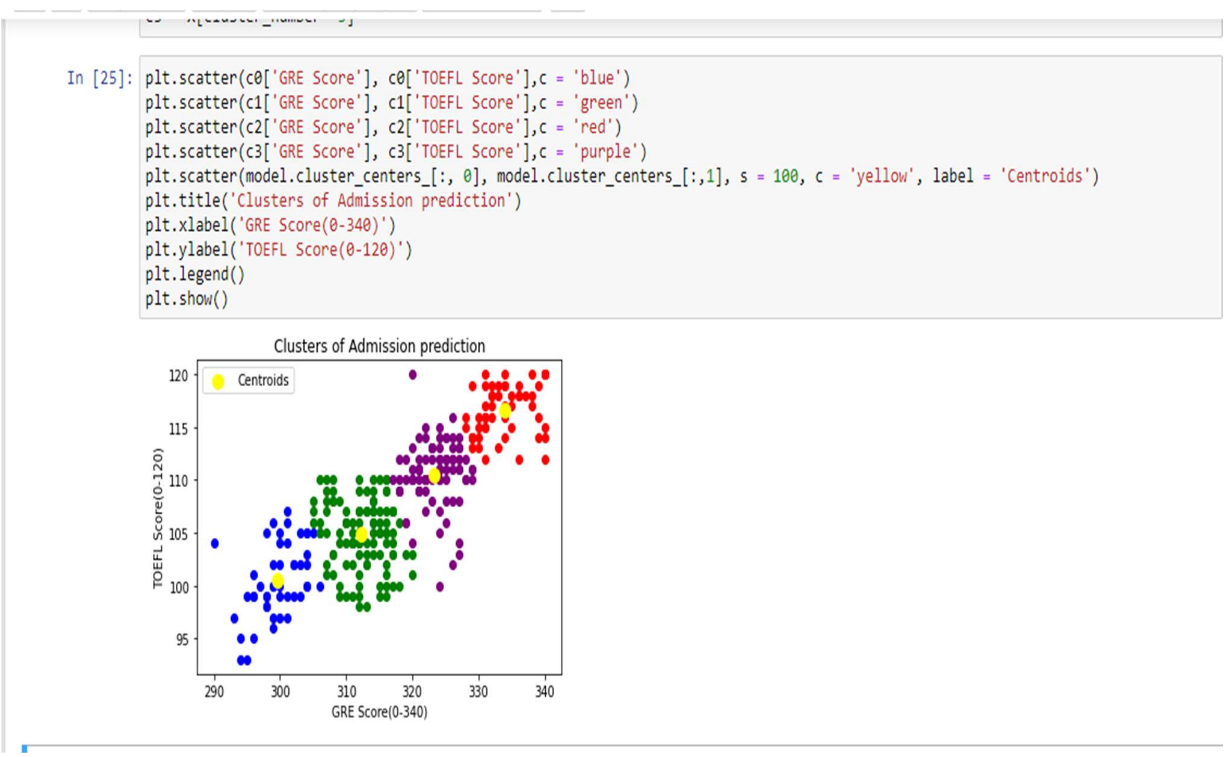
```
In [16]: cluster_number = model.predict(X)
```

```
In [17]: len(cluster_number)

Out[17]: 299
```

```
In [18]: len(X)

Out[18]: 299
```

```
In [23]: c0 = X[cluster_number==0]
         c1 = X[cluster_number==1]
         c2 = X[cluster_number==2]
         c3 = X[cluster_number==3]
```

```
In [24]: plt.scatter(c0['GRE Score'], c0['TOEFL Score'],c = 'blue')
         plt.scatter(c1['GRE Score'], c1['TOEFL Score'],c = 'green')
         plt.scatter(c2['GRE Score'], c2['TOEFL Score'],c = 'red')
         plt.scatter(c3['GRE Score'], c3['TOEFL Score'],c = 'purple')
         plt.scatter(model.cluster_centers_[:, 0], model.cluster_centers_[:,1], s = 100, c = 'yellow', label = 'Centroids')
         plt.title('Clusters of Admission prediction')
```



```
c3 = X[cluster_number == 3]
```

```
In [25]: plt.scatter(c0['GRE Score'], c0['TOEFL Score'],c = 'blue')
         plt.scatter(c1['GRE Score'], c1['TOEFL Score'],c = 'green')
         plt.scatter(c2['GRE Score'], c2['TOEFL Score'],c = 'red')
         plt.scatter(c3['GRE Score'], c3['TOEFL Score'],c = 'purple')
         plt.scatter(model.cluster_centers_[:, 0], model.cluster_centers_[:,1], s = 100, c = 'yellow', label = 'Centroids')
         plt.title('Clusters of Admission prediction')
         plt.xlabel('GRE Score(0-340)')
         plt.ylabel('TOEFL Score(0-120)')
         plt.legend()
         plt.show()
```

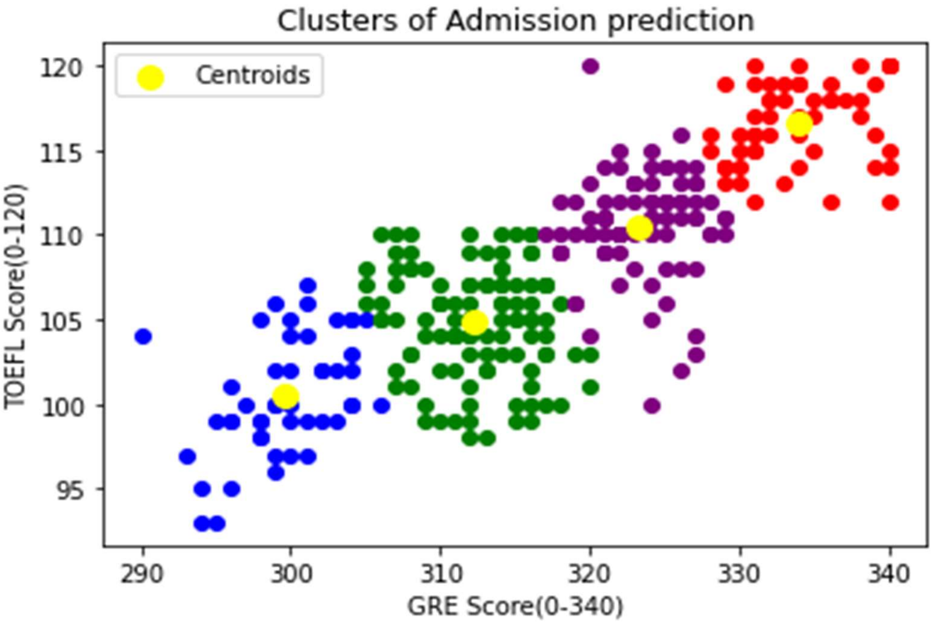
Conclusion

It is the fastest and most efficient algorithm to categorize data points into groups even when very little information is available about data.

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.

This cluster is a type of Centroid-based clustering, 4 clusters are made of this data in 4 colors that are Red, Green, Purple, Blue which represents GRE Score and 'TOEFL Score'. The score of GRE is between 0 to 340 and the score of TOEFEL is between 0 to 120**.** From this clustering we can see that the student who have good scored in GRE they also have good scored in TOEFEL and their CGPA is also good. One Outlier is present in purple color at TOEFEL score is 120 and GRE score is 320 and another outlier is of blue color at TOEFEL score is 104 and GRE score is 290.

Exam score for admission represents in colors	GRE Score	TOEFEL Score
Red	0-309	0-109
Green	309-320	97-110
Purple	319-330	98-120
Blue	329-340	111-120


Clusters of Admission prediction
Centroids
TOEFL Score(0-120)
GRE Score(0-340)