

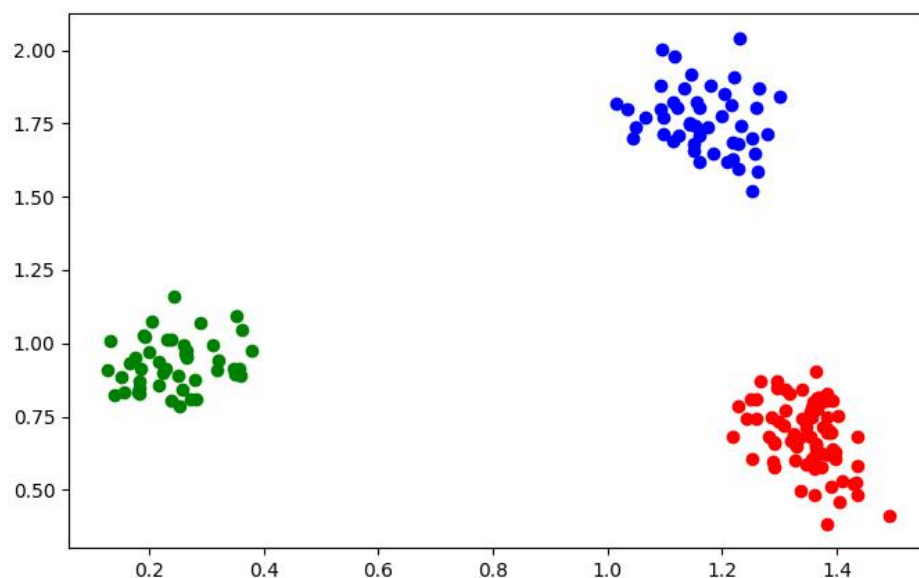
1. Applications of Clustering Algorithm are as follows
 - **Recommendation Engines** : The recommendation engines used in Amazon ,Flipkart etc.
 - **Market and Customer Segmentation** : A telecom company makes a cluster of prepaid users to understand the pattern/behaviour in the form of recharging amount, sending SMS, and using the internet .
 - **Search Results** : Google search engine uses clustering to suggest similar search results while typing in search box.
 - **Biological Data Analysis, Medical Imaging Analysis and Identification of Cancer Cells**
 - **Identifying Spam Mails** : Clustering is used to identify spam mails.
2. As the value of K increases the value of Objective function for K-Means **Decreases** non-linearly .
 - As K increases no of points in a cluster decreases and the points are closer to its cluster centre and hence distance is less , therefore $|x_i - c_i|^2$ decreases for every point in space and objective function value is decreased . If we have no of clusters k equal to no of points in space then we will have distortion function as zero since every point will be cluster centre of its own.

3. After applying K-Means to the data set provided we get the following results .

1)

```
Console x PyUnit  
kmeans.py [C:\Users\gutti\AppData\Local\Programs\Python\Python39\python  
The Value of Distortion Function is 2.60399663544384
```

2)



The three different colors represent three different clusters.

Hence the value of the distortion function for the given data set is = **2.60399663544384**

The code for the above output is as follows

```
import matplotlib.pyplot as plt
import numpy as np
import math
from sklearn.cluster import KMeans

array_txt = np.loadtxt(r"C:\Users\gutti\Downloads\A.txt",usecols=(0, 1),
skiprows=1)
df = array_txt[:,0:2]

kmeans = KMeans(n_clusters=3, random_state=100)
kmeans.fit(df)
y_kmeans = kmeans.predict(df)

print('The Value of Distortion Function is' ,kmeans.inertia_)

plt.figure(figsize = (8,5))
plt.scatter(df[y_kmeans == 0 ,0 ] ,df[y_kmeans == 0,1 ] , c = 'red')
plt.scatter(df[y_kmeans == 1 ,0] ,df[y_kmeans == 1,1 ] , c = 'green')
plt.scatter(df[y_kmeans == 2 ,0] ,df[y_kmeans == 2,1 ] , c = 'blue')
plt.show()

'''x=[]      #distortion function is calculated as follows
for m in range(0,3) :
    x.append(df[y_kmeans == m])
sum = 0
for i in range(0,3) :
    for j in range(0,Len(x[i])) :
        sum =sum + math.pow(math.dist(x[i][j] ,
kmeans.cluster_centers_[i]),2)
print(sum)'''
```