Course name: **Python programming and analytics by Rahul Sir**

Topic name: **topic 28**

Video name**: python by Rahul sir hierarchical clustering & TSA intro**

Video length: **1 hour 10 minutes 20 seconds**

Hierarchical clustering is a type of unsupervised machine learning algorithm used to cluster unlabeled data points. Like K-means clustering, hierarchical clustering also groups together the data points with similar characteristics. There are two types of hierarchical clustering: Agglomerative and Divisive. In the former, data points are clustered using a bottom-up approach starting with individual data points, while in the latter top-down approach is followed where all the data points are treated as one big cluster and the clustering process involves dividing the one big cluster into several small clusters.

Steps to Perform Hierarchical Clustering

Following are the steps involved in agglomerative clustering:

1. At the start, treat each data point as one cluster. Therefore, the number of clusters at the start will be K, while K is an integer representing the number of data points.

2. Form a cluster by joining the two closest data points resulting in K-1 clusters.

3. Form more clusters by joining the two closest clusters resulting in K-2 clusters.

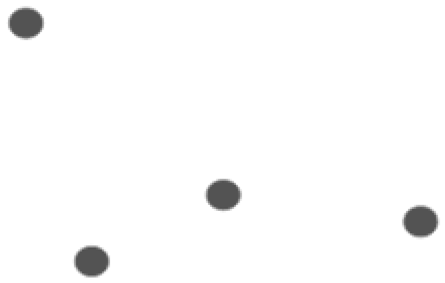
4. Repeat the above three steps until one big cluster is formed.

5. Once single cluster is formed, dendrograms are used to divide into multiple clusters depending upon the problem. We will study the concept of dendrogram in detail in an upcoming section.

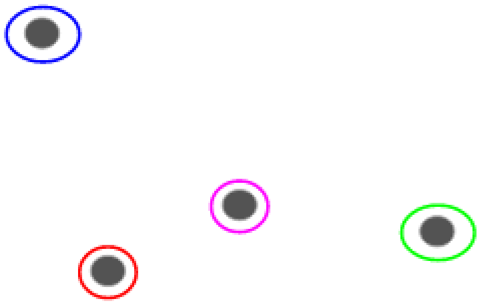
There are different ways to find distance between the clusters. The distance itself can be Euclidean or Manhattan distance. Following are some of the options to measure distance between two clusters:

1. Measure the distance between the close's points of two clusters.
2. Measure the distance between the farthest points of two clusters.
3. Measure the distance between the centroids of two clusters.
4. Measure the distance between all possible combination of points between the two clusters and take the mean.

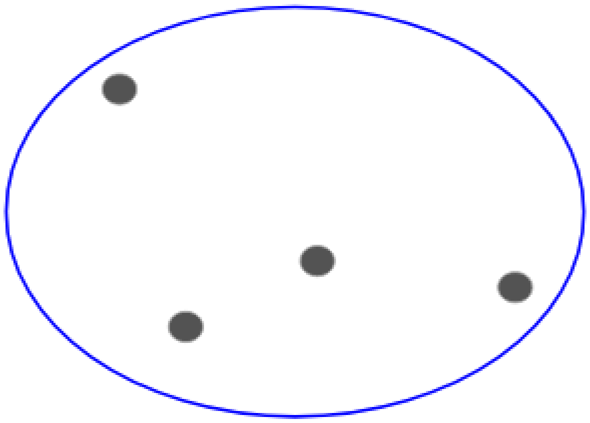
Let's say we have the below points and we want to cluster them into groups:



We can assign each of these points to a separate cluster:

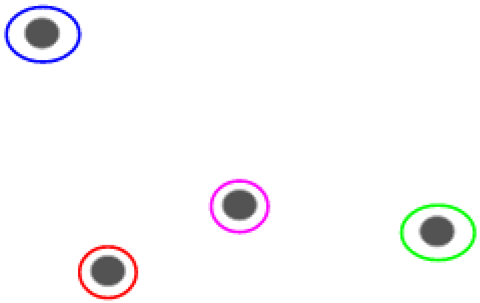


Now, based on the similarity of these clusters, we can combine the most similar clusters together and repeat this process until only a single cluster is left:

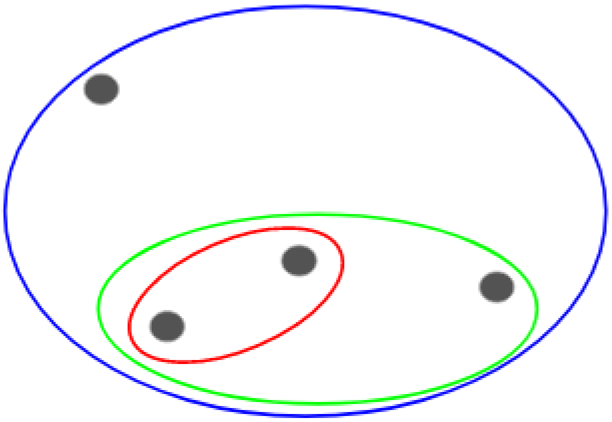


Agglomerative Hierarchical Clustering

We assign each point to an individual cluster in this technique. Suppose there are 4 data points. We will assign each of these points to a cluster and hence will have 4 clusters in the beginning:



Then, at each iteration, we merge the closest pair of clusters and repeat this step until only a single cluster is left:

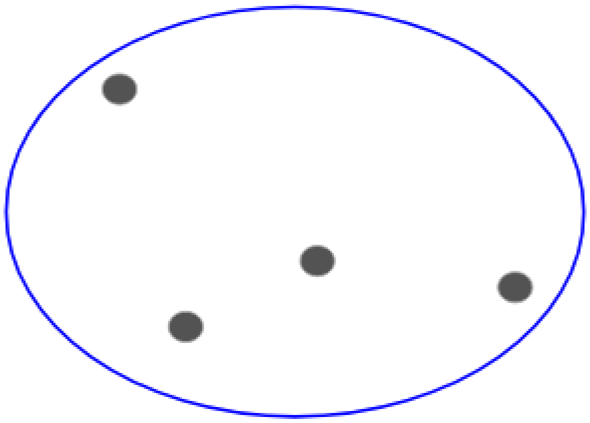


We are merging (or adding) the clusters at each step, right? Hence, this type of clustering is also known as **additive hierarchical clustering**.

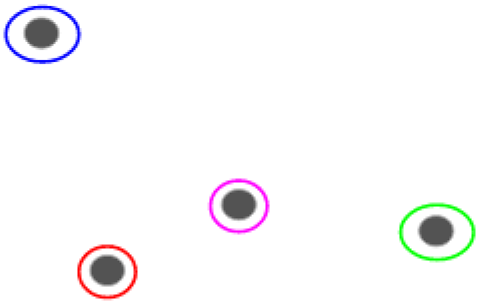
Divisive Hierarchical Clustering

Divisive hierarchical clustering works in the opposite way. Instead of starting with n clusters (in case of n observations), we start with a single cluster and assign all the points to that cluster.

So, it doesn't matter if we have 10 or 1000 data points. All these points will belong to the same cluster at the beginning:



Now, at each iteration, we split the farthest point in the cluster and repeat this process until each cluster only contains a single point:



We are splitting (or dividing) the clusters at each step, hence the name divisive hierarchical clustering.

Agglomerative Clustering is widely used in the industry and that will be the focus in this article. Divisive hierarchical clustering will be a piece of cake once we have a handle on the agglomerative type.

Steps to perform Hierarchical Clustering

At the start, treat each data point as one cluster. Therefore, the number of clusters at the start will be K, while K is an integer representing the number of data points.

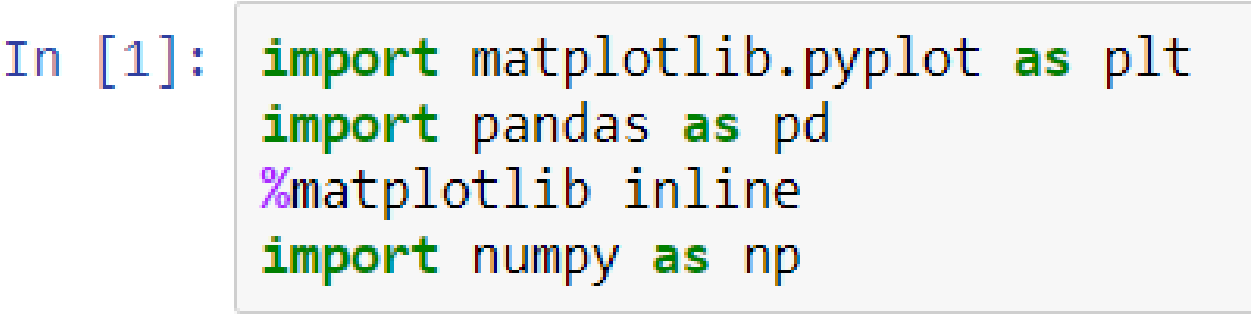
Form a cluster by joining the two closest data points resulting in K-1 clusters.

Form more clusters by joining the two closest clusters resulting in K-2 clusters.

Repeat the above three steps until one big cluster is formed.

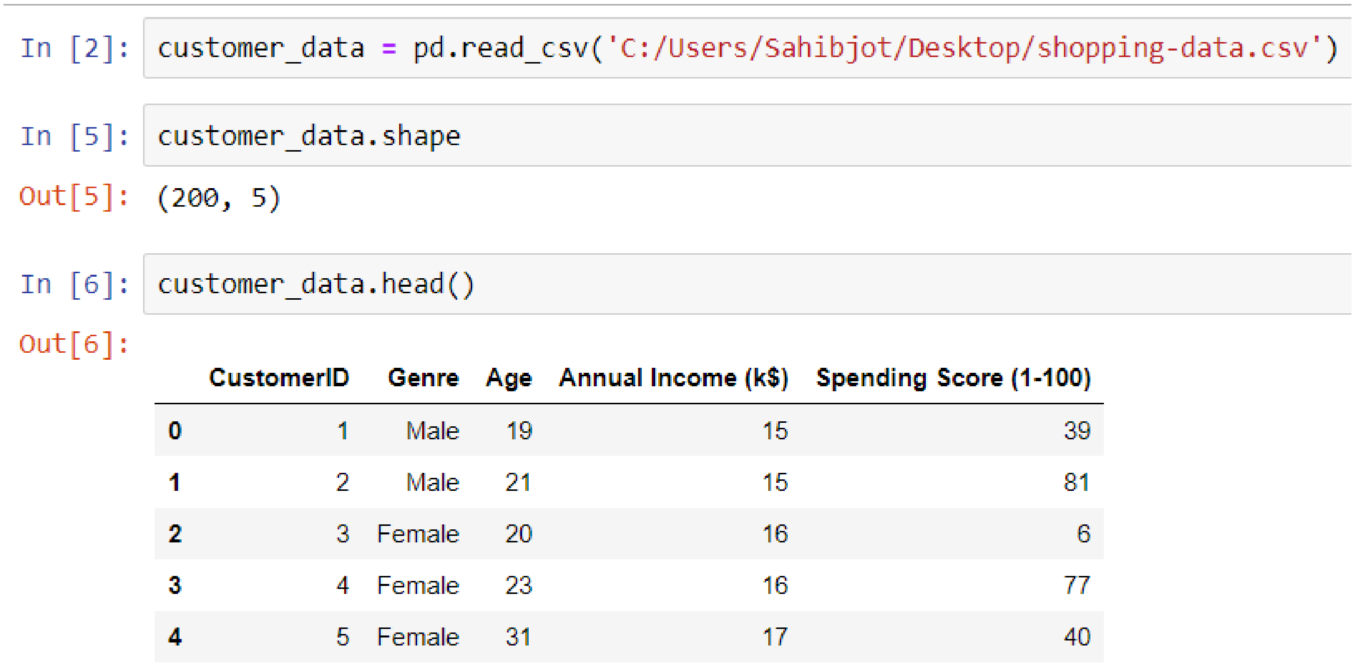
Once single cluster is formed, dendrograms are used to divide into multiple clusters depending upon the problem

Import the libraries



```
In [1]: import matplotlib.pyplot as plt
        import pandas as pd
        %matplotlib inline
        import numpy as np
```

Importing the data

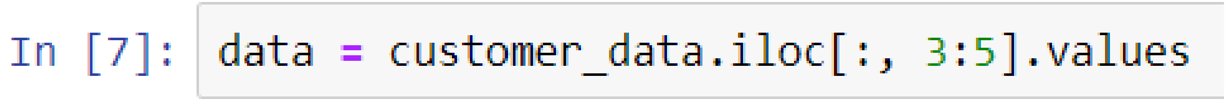


```
In [2]: customer_data = pd.read_csv('C:/Users/Sahibjot/Desktop/shopping-data.csv')

In [5]: customer_data.shape

Out[5]: (200, 5)

In [6]: customer_data.head()

Out[6]:
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

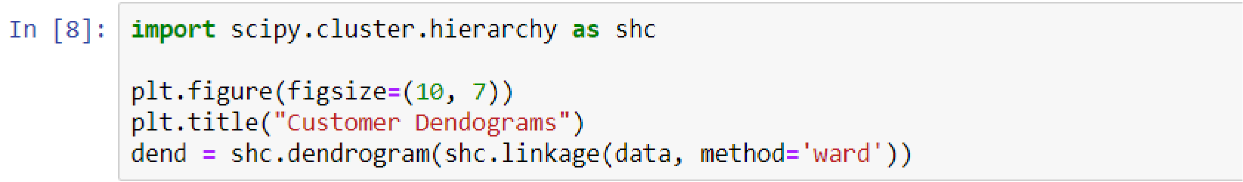
Our dataset has five columns: Customer ID, Genre, Age, Annual Income, and Spending Score. To view the results in two-dimensional feature space, we will retain only two of these five columns. We can remove Customer ID column, Genre, and Age column. We will retain the Annual Income (in thousands of dollars) and Spending Score (1-100) columns. The Spending Score column signifies how often a person spends money in a mall on a scale of 1 to 100 with 100 being the highest spender

Execute the following script to filter the first three columns from our dataset:

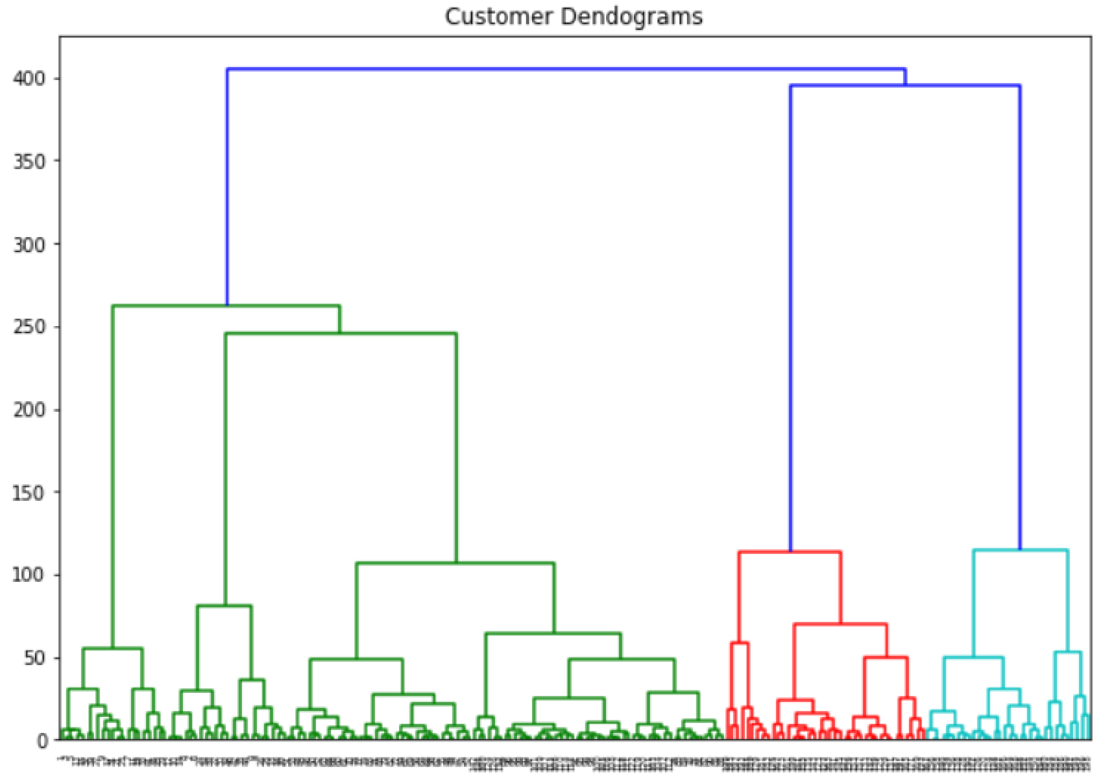


```
In [7]: data = customer_data.iloc[:, 3:5].values
```

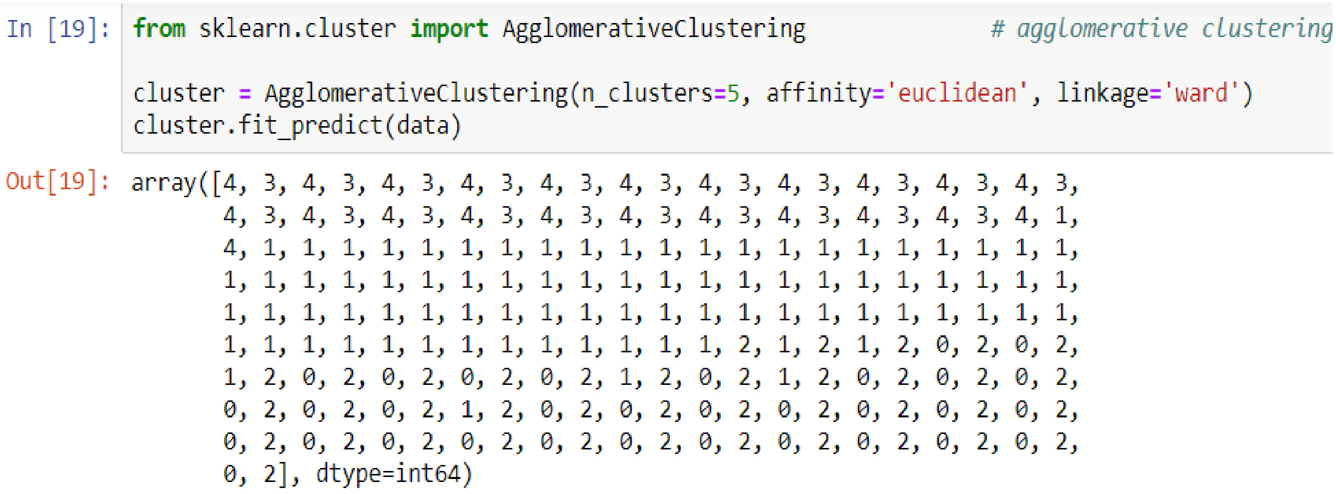
Next, we need to know the clusters that we want our data to be split to. We will use the SciPy library to create the dendrograms for our dataset. Execute the following script to do so:



```
In [8]: import scipy.cluster.hierarchy as shc

        plt.figure(figsize=(10, 7))
        plt.title("Customer Dendograms")
        dend = shc.dendrogram(shc.linkage(data, method='ward'))
```



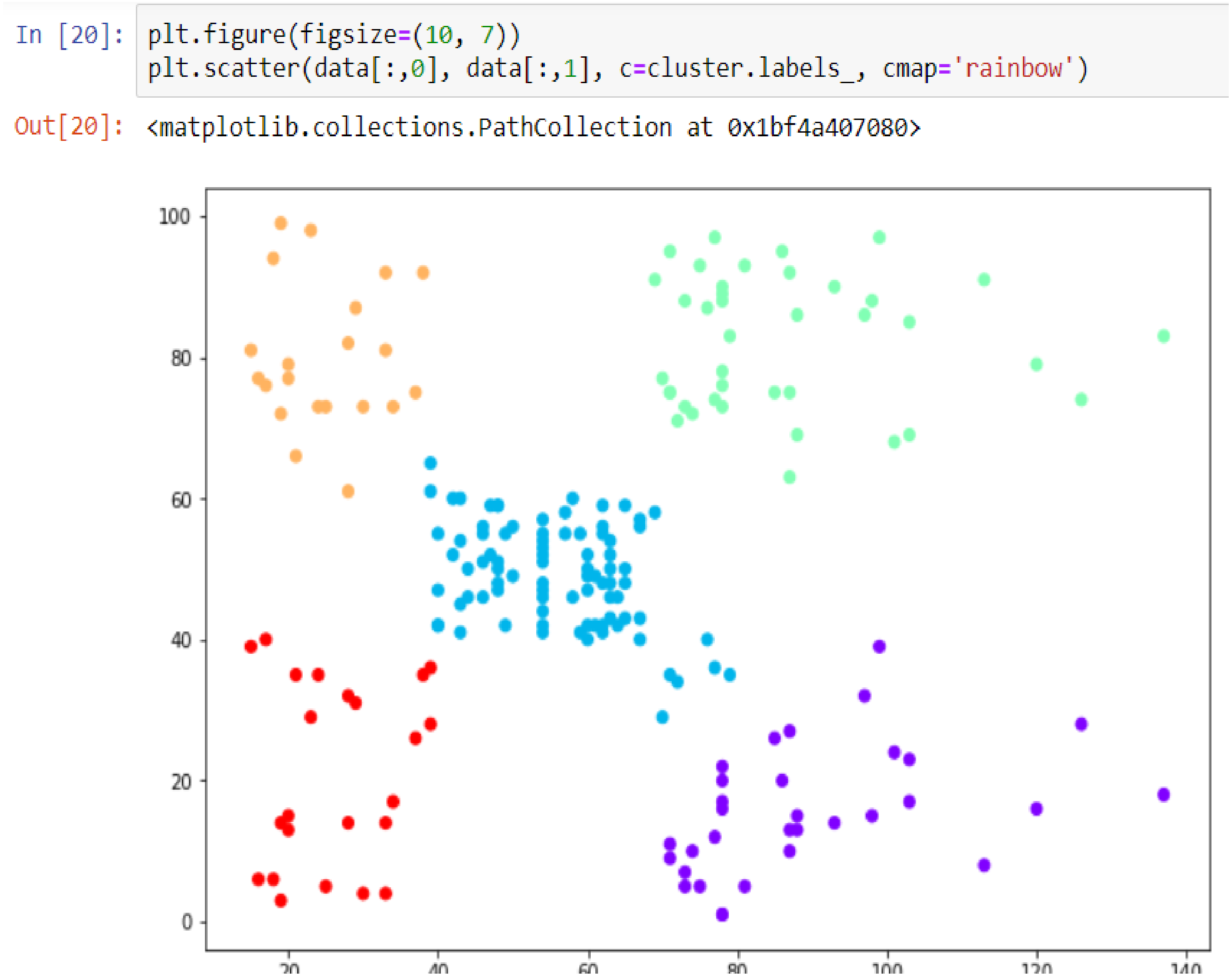
Now we know the number of clusters for our dataset, the next step is to group the data points into these five clusters. To do so we will again use the Agglomerative Clustering class of the sklearn.cluster library. Take a look at the following script:



```
In [19]: from sklearn.cluster import AgglomerativeClustering              # agglomerative clustering

         cluster = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='ward')
         cluster.fit_predict(data)

Out[19]: array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3,
                4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 1,
                4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 0, 2, 0, 2,
                1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2,
                0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
                0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
                0, 2], dtype=int64)
```

You can see the cluster labels from all of your data points. Since we had five clusters, we have five labels in the output i.e. 0 to 4.

As a final step, let's plot the clusters to see how actually our data has been clustered:



```
In [20]: plt.figure(figsize=(10, 7))
         plt.scatter(data[:,0], data[:,1], c=cluster.labels_, cmap='rainbow')

Out[20]: <matplotlib.collections.PathCollection at 0x1bf4a407080>
```

You can see the data points in the form of five clusters. The data points in the bottom right belong to the customers with high salaries but low spending. These are the customers that spend their money carefully. Similarly, the customers at top right (green data points), these are the customers with high salaries and high spending. These are the type of customers that companies target. The customers in the middle (blue data points) are the ones with average income and average salaries. The highest numbers of customers belong to this category. Companies can also target these customers given the fact that they are in huge numbers, etc.

Time series analysis

Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values.

Time series are widely used for non-stationary data, like economic, weather, stock price, and retail sales