UNIVERSITY OF KARACHI

UBIT DEPARTMENT

DATA MINING PROJECT

MCS FINALE (EVENING)

Course:
CS-626

Instructor:
Tehseen Ahmed Jilani

Group Members:

- Omama Imran (EP-20101045)
- Maham Rashid (EP-20101022)
- Muhammad Raza (EP-20101037)
- Syed Umer Ali (EP-19101084)

Project's Name:-

Customer segmentation using k-means and agglomerative clustering algorithms

Table of Contents

Customer Segmentation

Customer segmentation is the practice of dividing a company's customers into groups that reflect similarity among customers in each group. The goal of segmenting customers is to decide how to relate to customers in each segment in order to maximize the value of each customer to the business.

The Importance of Customer Segmentation:

Customer segmentation has the potential to allow marketers to address each customer in the most effective way. Using the large amount of data available on customers (and potential customers), a customer segmentation analysis allows marketers to identify discrete groups of customers with a high degree of accuracy based on demographic, behavioral and other indicators.

There are four main types of market segmentation:

First, we have demographic segmentation this is arguably the most basic of the segmentation types demographic segmentation divides a group or population based on variables like age, gender, income, occupation, education level, marital status, religion or nationality and it's usually one of the first stops when brands begin to segment their users because demographic information is easy to collect simple to measure and it is cost effective.

Next up we have geographic segmentation, geographic segmentation basically groups users based on their geographic details similar to demographics, geographic tends to be objective and essentially based on facts about the individual the five key areas that marketers take into account when segmenting geographically our location meaning the world regions, states, counties, cities, our neighborhoods urban estate so when the individual is in an urban suburban exurban or rural area climate the culture can be formed or influenced by things like religion, environment and social norms so brands take this segmentation into account because it focuses more on the person's values and beliefs rather than just where they are on the map lastly we have language if a brand is looking to expand into a global market it's so important to consider linguistic differences brands and marketers need to ensure that the language in their promotion and communication is precise and error free no brand wants to experience any embarrassing translation blunders

Thirdly, we have behavioral segmentation: behavioral segmentation does what it says, this segments users based on their behavior in a store or a website or in an app. Behavior data tends to be gathered and analyzed through tools like Google Analytics or by using particular algorithms. These tools and algorithms are able to track things like user's time on a website or website bounce rate online or in store actions communications with the brand and whether they are new or returning users. Big brands such as Spotify and Netflix also use this segmentation. Interests and lifestyles psychographic data can be collected through surveys interviews or most commonly using analytics from Google or social media where users often freely reveal personal information like their favorite TV shows their political views.

Lastly, we have psychographic segmentation, the benefits of using psychographic data in your marketing strategy can't be overlooked psychographics offer such a personal look into what consumers like dislike, need, want and love but it's not all roses because psychographic information is definitely the most subjective of the segmentation types and psychographics in particular can be harder to collect and analyze.

Since the marketer’s goal is usually to maximize the value (revenue and/or profit) from each customer, it is critical to know in advance how any particular marketing action will influence the customer. Ideally, such “action-centric” customer segmentation will not focus on the short-term value of a marketing action, but rather the long-term customer lifetime value (CLV) impact that such a marketing action will have. Thus, it is necessary to group, or segment, customers according to their CLV.

Let's consider that a mall wants to get insights about their customers and they have their customer data regarding their approaches, behaviors and other aspects to it so as a data scientist you can build a system that can cluster customers into different groups so one group of customer that tend to purchase more in a month and some other group may represent a group of customers that don't purchase that much in a month so you know having these groups of customers will give us better insights and that helps the mall to make Better Business decisions to make better marketing strategies. This clustering comes under unsupervised learning.

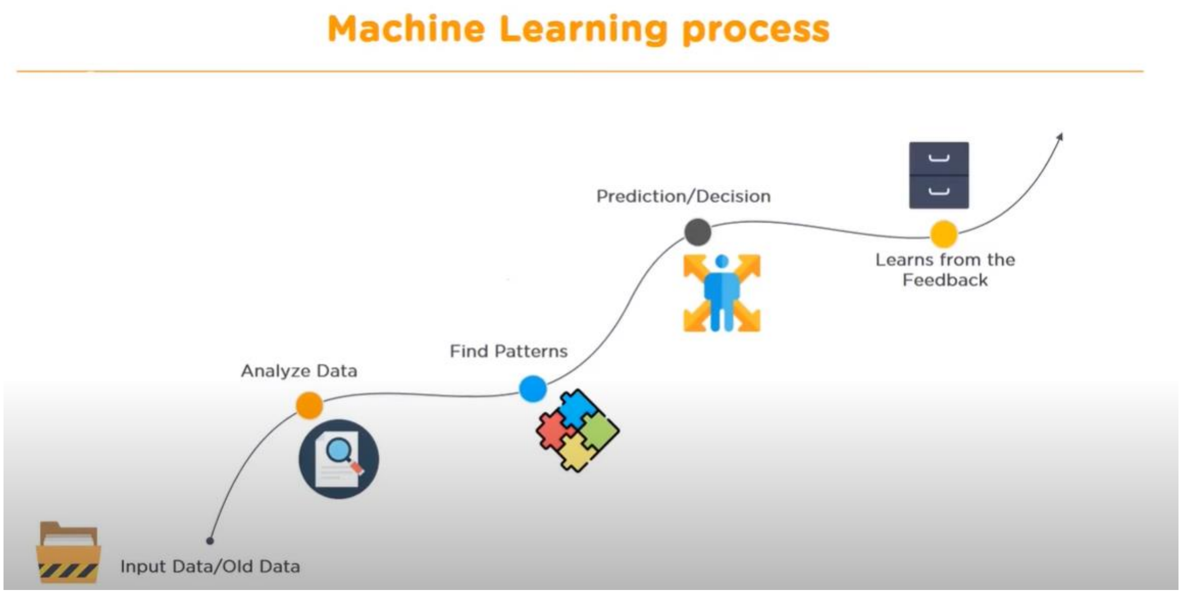
it's an application of artificial intelligence that provides systems ability to learn on their own and improve from experience without being programmed externally.

Let’s see how machine learning works:

Let’s say you provide a system with the input data that carries the photos of various kinds of fruits now you want the system to figure out what are the different fruits and grouped them accordingly. So what the system does it analyzes the input data then it tries to find patterns like shapes size and color based on these patterns the system will try to predict the different types of fruit and segregate them finally it keeps track of all such decisions it took in the process to make sure it's learning the next time you ask the same system to predict and segregate the different types of fruits it won't have to go through the entire process again that's how machine learning works.

Machine Learning

Machine learning is a crucial part of the rapidly expanding discipline of data science. Algorithms are trained to generate classifications or predictions using statistical approaches, revealing crucial insights in data mining initiatives. Following that, these insights drive decision-making within applications and enterprises, with the goal of influencing important growth KPIs. As big data expands and grows, the demand for data scientists will rise, necessitating their assistance in identifying the most relevant business questions and, as a result, the data needed to answer them.



Machine Learning Algorithms:

The following are some of the elements that assist us in selecting the best algorithm.

- **Factors Help To Choose Algorithm:-**

1. Types Of Algorithms
2. Parameterization
3. Memory Size
4. Overfitting Tendency
5. Time Of Learning
6. Time Of Predicting

- **Types of Algorithms:-**

Following are the types of algorithms.

1. Regression:

It's a method for predicting a dependent variable from a set of independent variables. The algorithms that are subject to regression are

i. Linear Regression
ii. Decision Tree
iii. Random Forest
iv. Boosting

2. Classification:

It is a method for estimating a mapping function (f) from discrete output variables to input variables (X) (Y). These are the algorithms that are classified.

i. Logistics Regression
ii. Naïve Bayes
iii. SVM
iv. Neural Networks
v. Decision Tree
vi. Random Forest
vii. Boosting

3. Clustering:

It's a method of separating a population or set of data points into groups such that data points in the same group are more comparable to other data points in the same group and different from data points in other groups. The K-means method is a popular clustering algorithm.

❖ Types Of Machine Learning:-

Machine learning is primarily consists of three types. First one is supervised machine learning as the names suggests you have to supervise the machine learning while you train it to work on its own it requires label training data.

- **Supervised Learning:-**

Its use of labelled datasets to train algorithms that accurately classify data or predict outcomes defines it. As input data is fed into the model, the weights are adjusted until the model is properly fitted, which happens during the cross-validation phase. Organizations can use supervised learning to tackle a range of real-world problems at scale, such as spam classification in a distinct folder from your email.

- **Reinforcement Learning:-**

The study of decision-making is called Reinforcement Learning (RL). It's all about figuring out how to behave optimally in a given situation in order to maximize your reward. This optimal behavior is acquired by encounters with the environment and observations of how it responds, similar to how toddlers explore the world around them and learn the activities that assist them in achieving a goal.

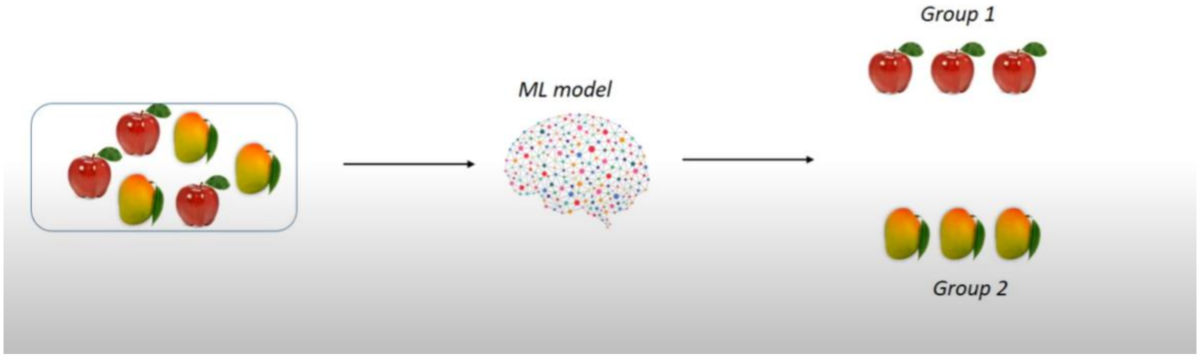
In the absence of a supervisor, the learner must figure out the best way to maximize the reward on his or her own. This search resembles a trial-and-error procedure. The value of an activity is determined not just by the immediate reward it provides, but also by the potential for a later payoff. Reinforcement learning is a particularly strong algorithm because it can learn actions that lead to eventual success in an unknown environment without the assistance of a supervisor.

Since our course and project is based on unsupervised learning, we will cover this specific type of machine learning:

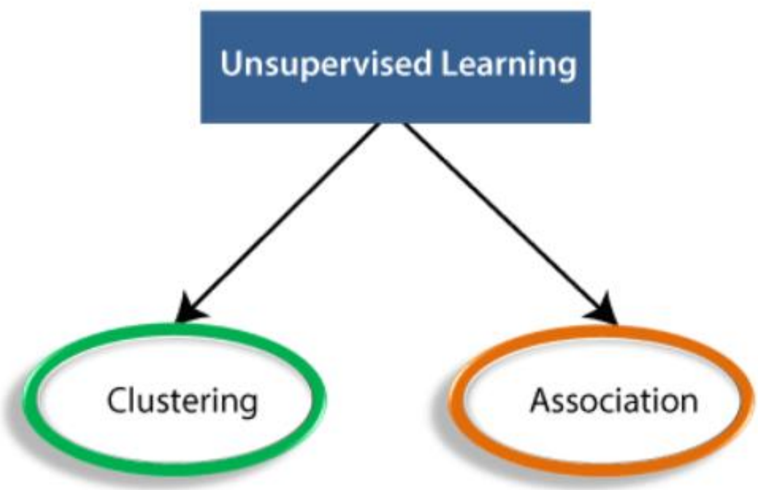
- **Unsupervised Learning**

In unsupervised learning we train our model with unlabeled data. let's try to understand this with an example:

We have several images of apples and mangoes and so once we feed this data to our machine learning model what happens is that it can group the group the data based on similar patterns so it can group the apples in one group and it can group the mangoes in the second group OK So what happens here is we are not telling the model that these images represent apples and these images represents mangoes so we are not giving that label whereas in the supervised learning we will tell the machine that these images represents apple supervised and unsupervised learning. Unsupervised learning algorithms automatically finds the pattern between those images, and it groups similar items in one group and other items in another group, this is the basic idea behind unsupervised learning.



➢ **Types of Unsupervised learning:**



- **Clustering:**

A clustering problem is where you want to want to discover the inherent groupings / categories in the data.

For example, grouping customers by purchasing behavior.

- **Association:**

An association rule learning problem is where you want to discover rules that describe large portions of data.

For example, people that tend to buy X (milk) also tend to buy Y (bread).

Customer segmentation provides insight into the landscape of the market revealing customer characteristics that can be used to group customers into segments that have something in common this process is also known as clustering and the techniques used to develop these models are called clustering algorithms.

In this project, K means clustering algorithm and hierarchical clustering is used to do customer segmentation. In customer segmentation we will try to group customers based on their spending patterns and purchase behavior.

Data clustering:

Clustering is grouping data points and creating partitions based on similarity if two things are similar in some ways, they often share other characteristics almost everything we perceive is in the form of clusters. A cluster is a set of similar data points or a set of points that are more similar to each other than two points in other clusters. It is classified as unsupervised learning technique and the key difference from other machine learning techniques is that clustering does not have a response class, after grouping observations a human needs to visually look at the clusters and optionally associate meaning to each cluster.

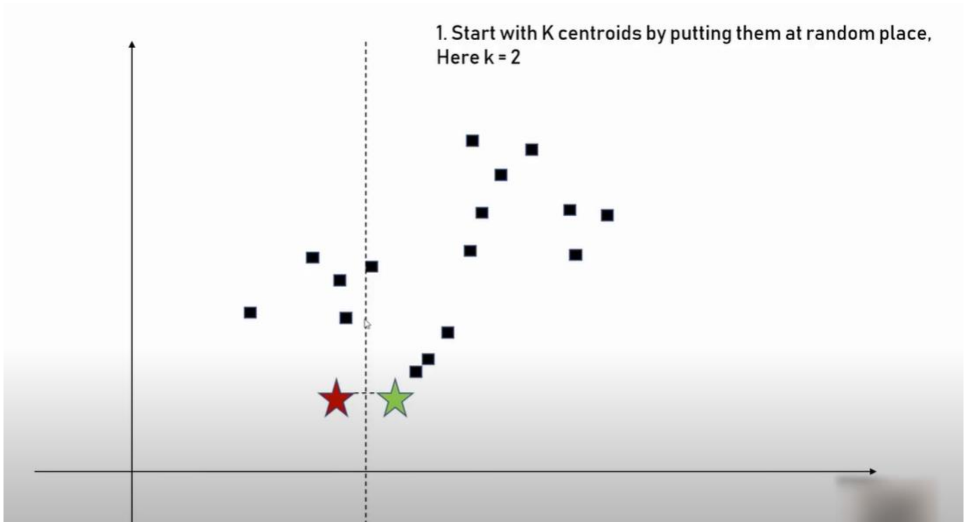
The ultimate prediction is the set of clusters themselves and this technique works only with data that is in numeric form this means that any categorical variable needs to be converted to a numeric variable by binarization.

The K Means Clustering Algorithm:

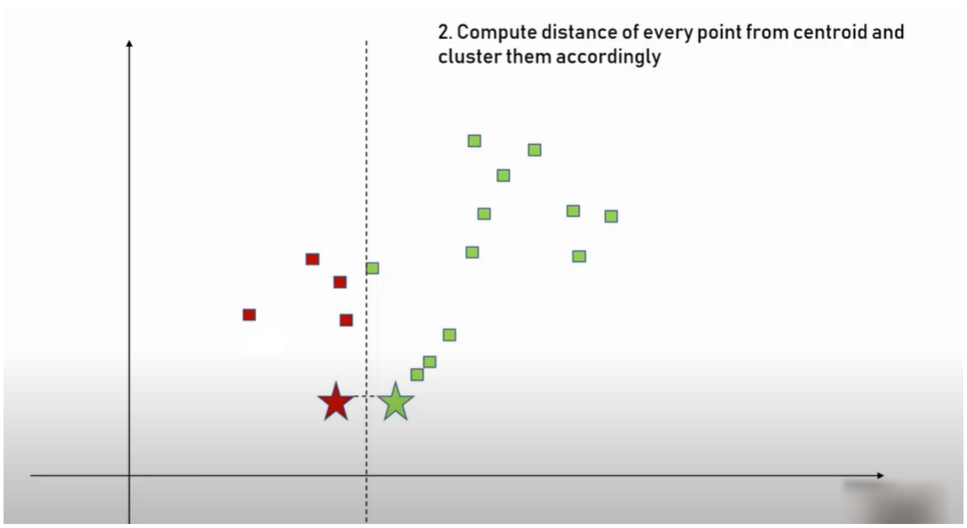
Let’s say you have a data set where X&Y axis represent two different features and you want to identify clusters in this data set now when the data set is given to you , you don't have any information on target variables and you don't know what you're looking for, all your trying to do is identify some structure into it and one way of looking into it are these two clusters just by the visual examination we can say that this data set has these two clusters and K-means helps you identify these clusters.

K in K-means is a free parameter. Before the algorithm starts, you have to tell the algorithm the value of K. Suppose we start with K=2 and the first step is to identify two random points which you consider as the center of those two clusters and we call them centroids. The next step is to identify the distance between each of these data points from the centroids.

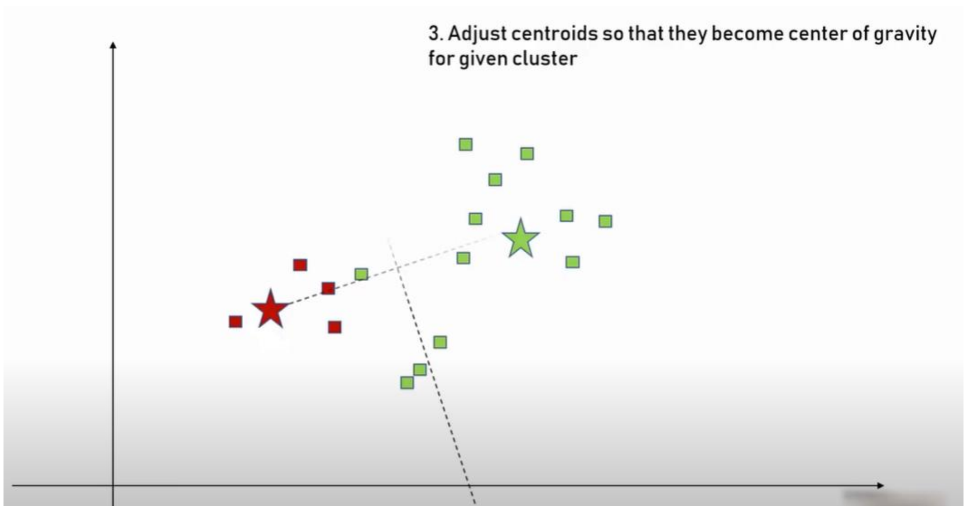
1) So first, the K means algorithm starts with K centroids by putting them at random place. In this case we have k =2



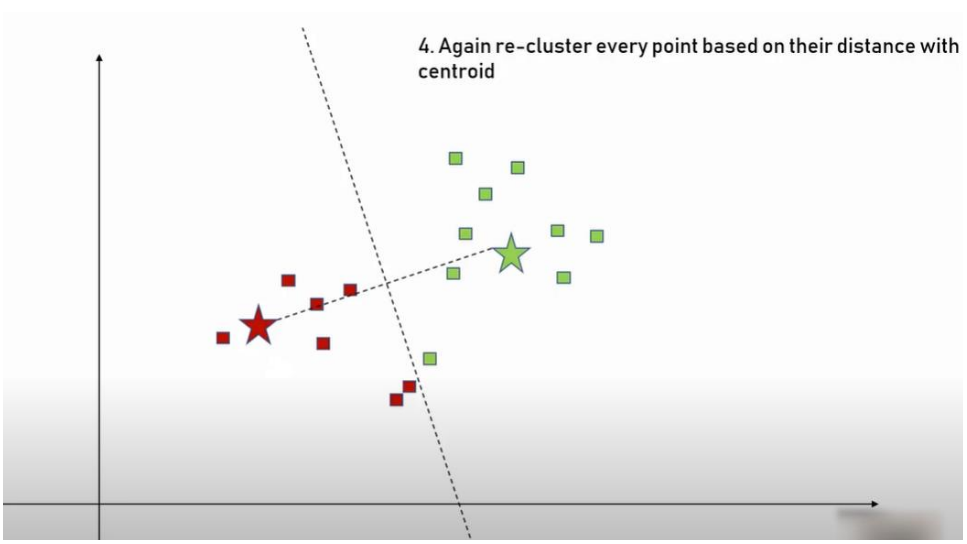
2) Adjust clusters so that they become center of gravity for given cluster.



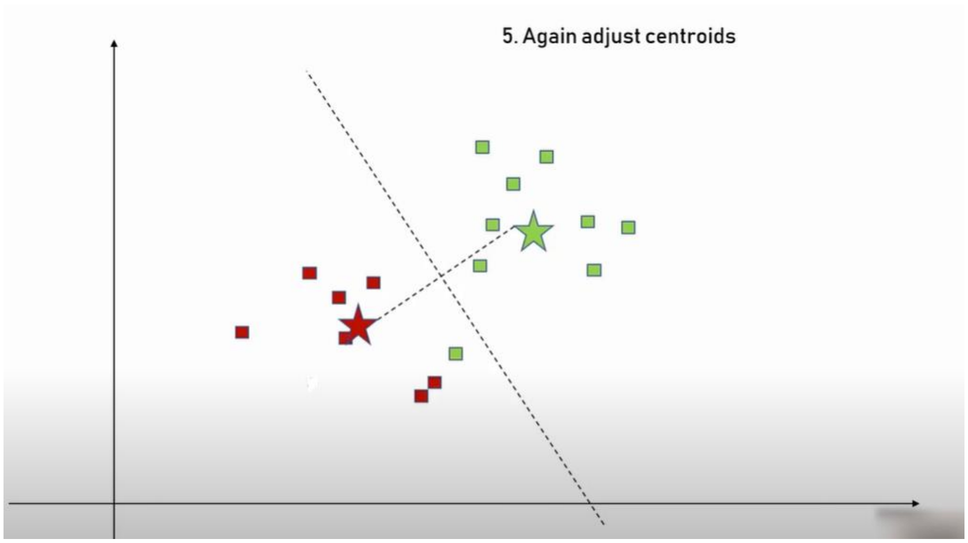
3) Again, re-cluster every point based on their distance with the centroid.



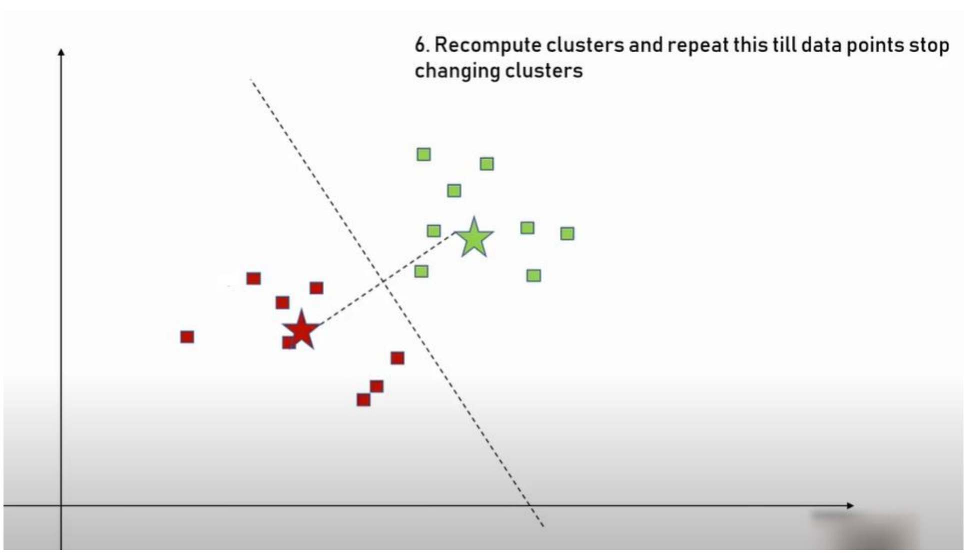
4) Again re-cluster every point based on their distance with centroid.



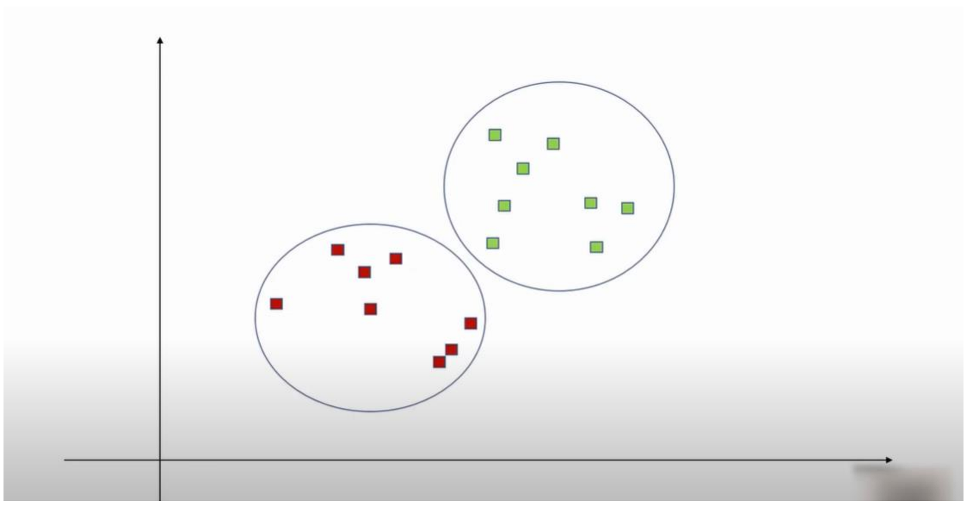
5) Again adjust centroids



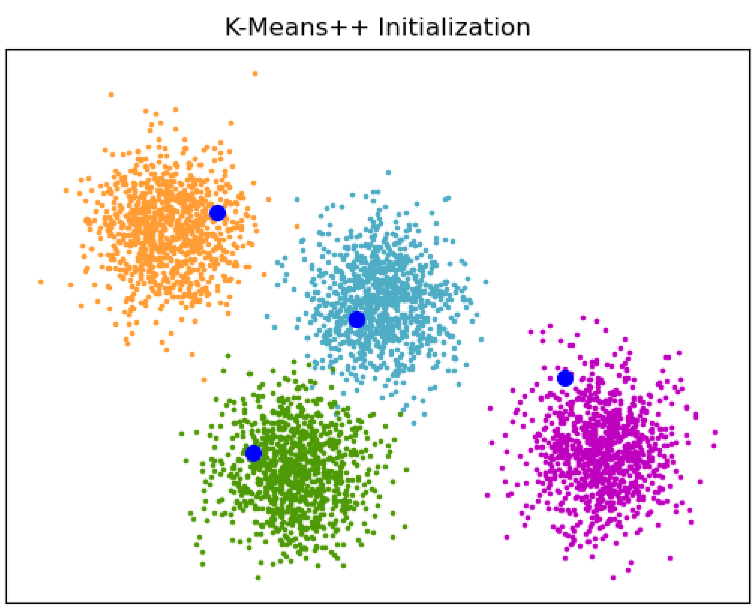
6) Recompute clusters and repeat this till data points stop changing clusters.



7) We get the following group of clusters :



In this project we use k-means ++, K-means starts with allocating cluster centers randomly and then looks for preferable solutions. K-means++ starts with allocating one cluster center randomly and then searches for other centers which helps us improve the outcome of our clustering.



How to determine the correct number of clusters?

To find the correct number of K, there is a technique called elbow method and the way that method works is you start with some K value let's say we start with k=2 and we try to compute sum of square error but what it means is for each of the clusters you try to compute the distance of individual data points from the centroid you square it and then you sum it up so for this cluster we got square error = 1 similarly for the second cluster you will get the error = 2 and you do that for all your cluster and then you get the total sum of squared errors.

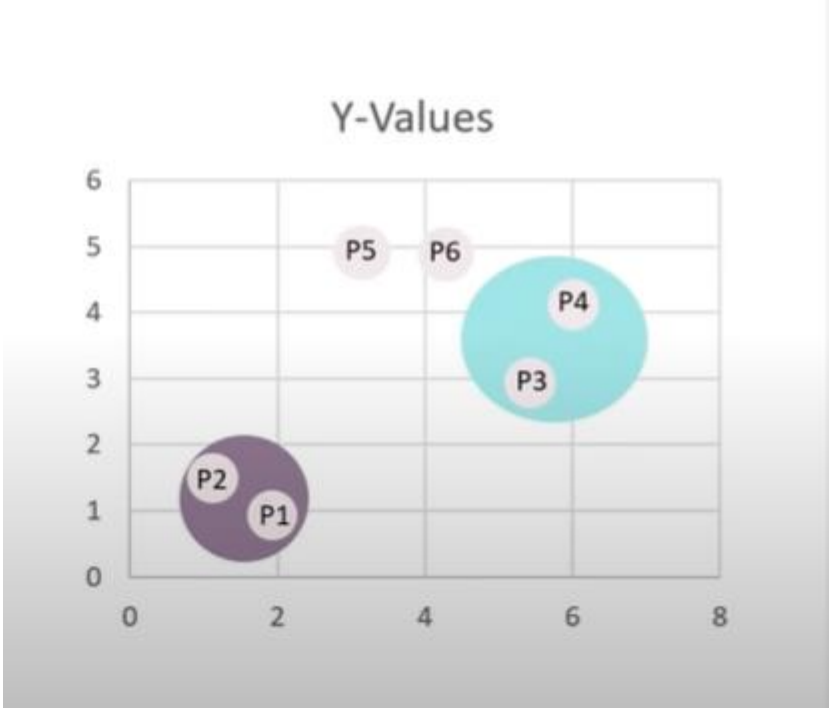
How to find out the distance measures?

Distance measures will determine the similarity between two elements and it will influence the shape of the clusters and we have Euclidean distance measure we have squared Euclidean distance measure which is almost the same thing but with less computations and we had the Manhattan distance measure which will give you slightly different results and we have the cosine distance measure which again is very similar to the Euclidean.

Hierarchical Clustering:

Hierarchical clustering is separating data into different groups based on some measure of similarity.

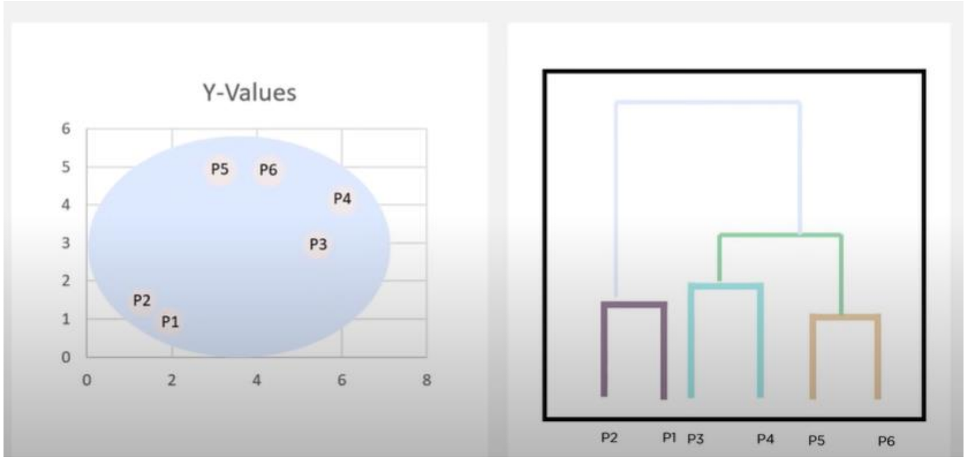
Let’s see how hierarchical clustering works let's consider we have a few points on a plane and this plane is a 2D plane and it has XY coordinates.



First, we compute the distance between each point and each point is a cluster of its own then we have each point being it's own cluster we try to find the least distance between two data points to form a cluster.

Once we find those with the least distance between them we start grouping them together so we start forming clusters of multiple points this is represented in a tree like structure called dendogram.

We terminate when are left with one cluster.



The Agglomerative Clustering Algorithm:

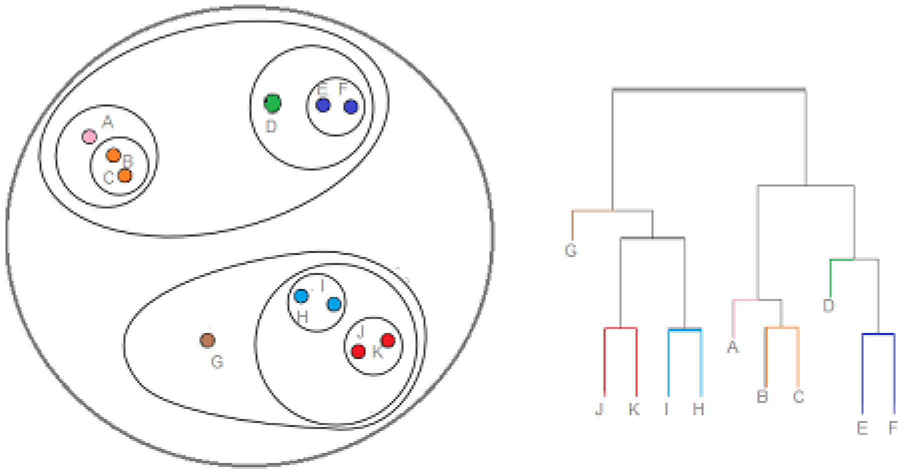
- The agglomerative clustering begins with each element as a separate cluster and then we merge them into a larger cluster.

- The hierarchical agglomerative starts with one cluster individual item and its own cluster and iteratively merging all the clusters until all items belong to one cluster.

- It follows a bottom-up approach and pictorially dendrograms used to represent the HAC.

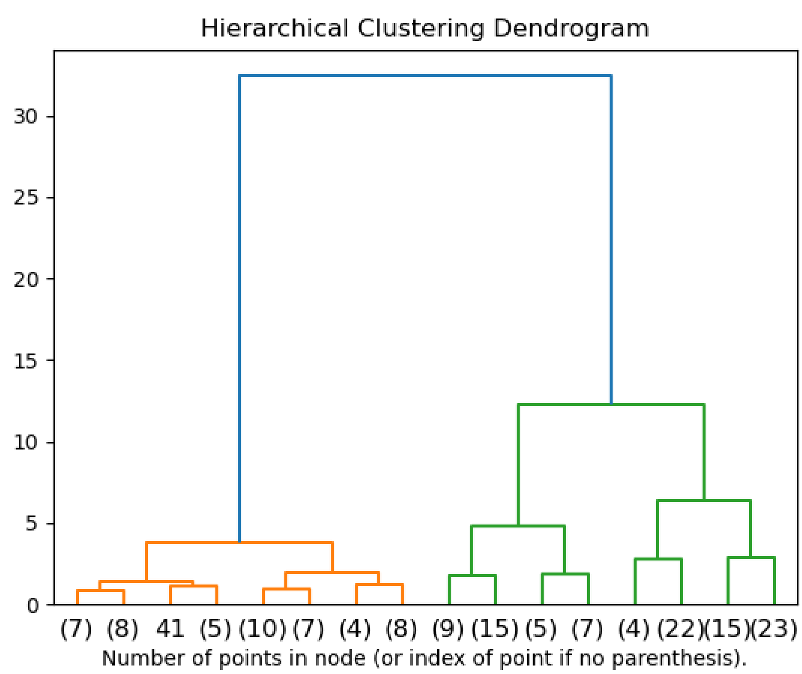
- It follows three techniques:

    - The single nearest distance for single linkage : Single linkage is the distance between the closest members of the two clusters so it will use the min distance.
    - Complete farthest distance for complete linkage: complete linkage this is the distance between the members that are the farthest apart so it uses the Max distance
    - Average-average distance or average linkage : Average linkage involves looking at the distance between all the pairs and averages of all these distances. It uses the average linkage this is also called unweighted pair group mean averaging.



Dendrograms:-

Dendrograms it is a tree like structure which represents hierarchical technique, each individual item is known as leaf and when these leaves combine together it forms a cluster and it is joined at a root one cluster. A cluster at a level one is the merger of its child cluster at level i + 1.



Source Code:

so now let's try to understand the workflow which we are going to follow first we need the customer data because we need this data to train our machine learning model and once, we have this data we need to process this data as we cannot feed this data directly to our machine learning model.

First, we need to do import the dependencies so dependencies are nothing but the libraries and the functions which we will use for this particular project. For example, the matplotlib library is useful for making plots so these are data visualization libraries and finally we load the Kmeans and other clustering libraries.

We'll import some basic python Libraries like:

Importing the Dependencies

```
[ ]  import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.cluster import KMeans
     from sklearn.cluster import AgglomerativeClustering
     from sklearn.preprocessing import normalize
     from sklearn.metrics import confusion_matrix
```

Pandas:

Pandas one of the most important and widely used libraries used for Data analysis. It has different data structure and operation to manipulate and analyse data. Some of the mostly used data manipulation operations in pandas are merging, reshaping, selecting, as well as data cleaning, and data wrangling.

Matplotlib:

Matplotlib is an extension to Numpy library. It is used for Data Visualization and graphical plotting. It can also provide object-oriented APIs incorporating plots into applications. Also there are alternatives to matplotlib like Seaborn and plotly, but matplotlib is the most frequently used one.

Sklearn:

This particular library has almost all the machine learning algorithms, ranging from unsupervised to supervised as well as ensembling algorithms in it. Sklearn is a one stop solution for most if not all of the machine learning algorithms, validations and preprocessing techniques. It also does offer model accuracy tools such as confusion matrix, precision recall, cross validation etc

The next step is about data collection and analysis. We load the file and the `read CSV` function will read the CSV file and load all the contents of the CSV file to data frame.

We need to mention the path of the file:

```
[ ]  # loading the data from csv file to a Pandas DataFrame
     customer_data = pd.read_csv('/Mall_Customers.csv')
```

We load the file to the data frame named as customer_data.

Data frames are nothing but structured table so the data set which we have is in the form of a CSV file. CSV represents, separated values so it is better to load the data into pandas for a structured table for better processing and analysis.

we have about 5 columns and the first column is customer ID so each customer has a unique ID and we have the gender of each customers and we also have their age and their annual income in $1000 or 15 represents $15,000 seventeen represents $17,000 annual income and finally we have spending score and the value of spending scores range from one to 100.

we have the data for 200 different customers. This is a good sample size and total 4 features or four columns.

We need to select new features and then we need to also analyze the data to see which features are important for us and what are the various features that particular data set contains so that will be covered in data analysis part of the code :

```
[ ] # first 5 rows in the dataframe
    customer_data.head()
```

	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

Choosing the Annual Income Column & Spending Score column

```
[ ] customer_data = customer_data.rename(columns={"Genre":"Gender"})
```

```
[ ] customer_data.drop(columns=['CustomerID', 'Gender'], inplace=True)
```

The next step is to find the number of rows and columns and getting basic information about the data set:

```
# finding the number of rows and columns
customer_data.shape
```

```
(200, 3)
```

```
# getting some informations about the dataset
customer_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 3 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Age                     200 non-null    int64
 1   Annual Income (k$)      200 non-null    int64
 2   Spending Score (1-100)  200 non-null    int64
dtypes: int64(3)
memory usage: 4.8 KB
```

Now we check for null or missing values:

```
[ ]  # checking for missing values
     customer_data.isnull().sum()

     Age                       0
     Annual Income (k$)        0
     Spending Score (1-100)    0
     dtype: int64
```

The next step is to choose which columns we are going to need for clustering. We have total five columns the first column is customer ID, we don't want this customer ID as it's not required because we are going to group the customers based on their spending behavior and for that customer ID, gender and age is not required. In this case we're just going to use the two features annual income and spending score. So, what we will get is group of customers and these groups of customers will be classified based on their annual income and spending score.

so now let's extract these two columns alone from this customer data. Doing this we are locating particular columns the columns are nothing but three and four here. Indexing in Python starts from zero so number count doesn't start from one but it starts from zero in Python so the index of the first column i.e. customer ID column is zero and the index of gender column is one and for ages 2 for annual income three and four so totally we have 5 columns and this is 0123 and four so we are going to take these two columns so the index of this annual income column is three and the index our spending score column is 4.

so, we have a list of values and each list contains 2 values the first value represents the annual income. Suppose annual income is 15 and spending score is 39 you can see at 15 and 39 so all the values in this first column represents the annual income of specific customers and the second value in each list represents their spending score so these two values are the ones which we need for our clustering.

So in our X axis we will take the annual income and then Y axis we will take the spending so we won't be taking any other values like their age or gender so we will just take these two values.

After that we need to choose the correct number of clusters, so we need to tell the machine learning model that I want three clusters or five clusters or this many groups of customers.

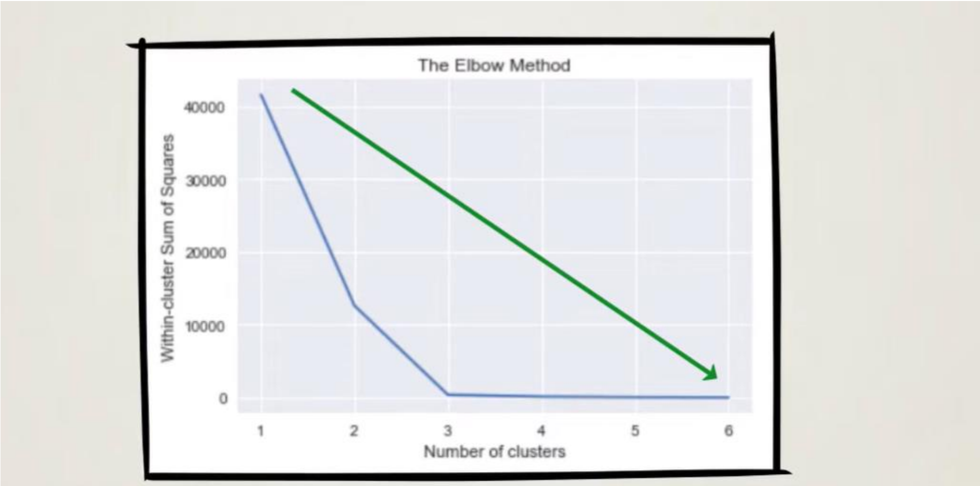
We want to divide the data into these many groups, so we find that number by using a method called within cluster sum of squares which tells us the correct number of clusters that is suitable for this particular data set.

As we know that clustering is about minimizing the distance between points in a cluster and maximizing the distance between clusters it turns out that for K means these two occur simultaneously if we minimize the distance between points in a cluster, we are automatically maximizing the distance between clusters. Now, the distance is measured in somewhat squares and the academic term is within clusters sum of squares or WCSS, similar to SST SSR and SSE from regressions WCSS is a measure developed within the ANOVA framework.

If we minimize WCSS we have reached a perfect clustering solution but the problem if we have the same six countries and each one of them is a different cluster so a total of 6 clusters then WCSS is zero that's because there is just one point in each cluster and we can't have it within cluster sum of squares furthermore the clusters are as far as they can possibly be imagine this with one million observations a 1,000,000 cluster solution is definitely of no use.

Similarly, if all observations are in the same cluster the solution is useless and WCSS is at its maximum there must be some middle ground and that is, we don't really want WCSS to be minimized instead we want it to be as low as possible while we can still have a small number of clusters so we can interpret them.

If we plot WCSS against the number of clusters, we get this graph which looks like an elbow and is known as the elbow method:



The within cluster sum of squares is a monotonously decreasing function which is lower for a bigger number of clusters.

Once we have the number of clusters we need we can feed this data to a K means and agglomerative clustering algorithm so once you feed this algorithm to this model it can group the data depending on the similarities depending on the similar spending pattern etc.

Optimum Number of Clusters = 5

Training the k-Means Clustering Model

```
[ ]  kmeans = KMeans(n_clusters=5, init='k-means++', random_state=0)

     # return a label for each data point based on their cluster
     Y = kmeans.fit_predict(X)

     print(Y)
```

```
[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 3 4 1 4 3 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 0 2 1 2 0 2 0 2 1 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 0 2 0
 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2]
```

5 Clusters - 0, 1, 2, 3, 4

// number of clusters for agglomerative algorithm :

```
[ ]  cluster = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='ward')
     cluster.fit_predict(customer_data)
```

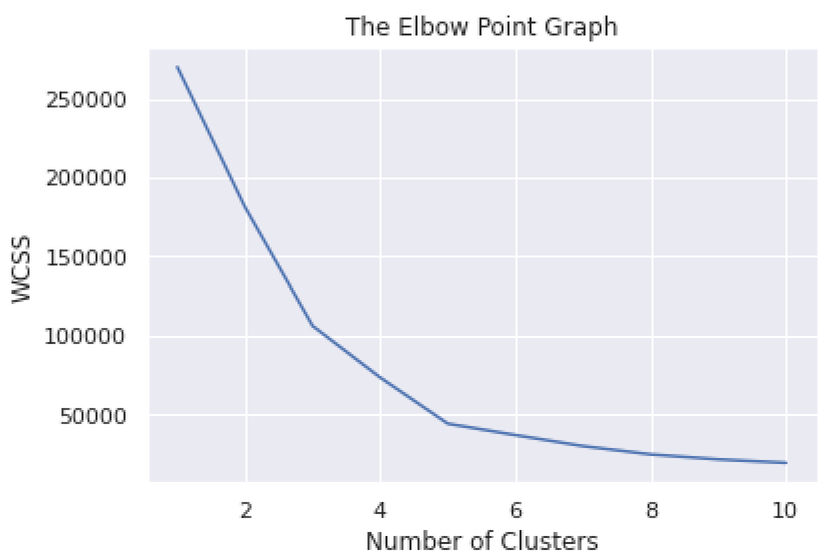
```
array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3,
       4, 3, 4, 3, 4, 0, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 0,
       4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    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

```
[ ]  # finding wcss value for different number of clusters

     wcss = []

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

       wcss.append(kmeans.inertia_)
```

We will plot all the data points in the graph having the centroid so each clusters as their own centroid so in this case the number of clusters we have is 3 so when you try to find this value it tries to find the distance between each data point and the centroid of those clusters the WCSS value should be very less that means the distance between the date and point and the centroid should be very less for this we will find this value and you know build plot like this so you can see here this plot or this graph is called as an elbow graph:

```
[ ]  # plot an elbow graph

     sns.set()
     plt.plot(range(1,11), wcss)
     plt.title('The Elbow Point Graph')
     plt.xlabel('Number of Clusters')
     plt.ylabel('WCSS')
     plt.show()
```

The visualization looks something like this :



After that we can visualize these clusters by putting the data or the predictions made by this particular K means clustering model into plots to get better insights about this data.

so, we have 5 clusters so each of these represents data point of each five clusters and here the zero represents the X axis value which is annual income and here the 1 represents spending score so this will give us all the data points along with their annual income and spending scores as well as they are labels of clusters now, we need to plot the centroids so the dots are almost the center point of each cluster.

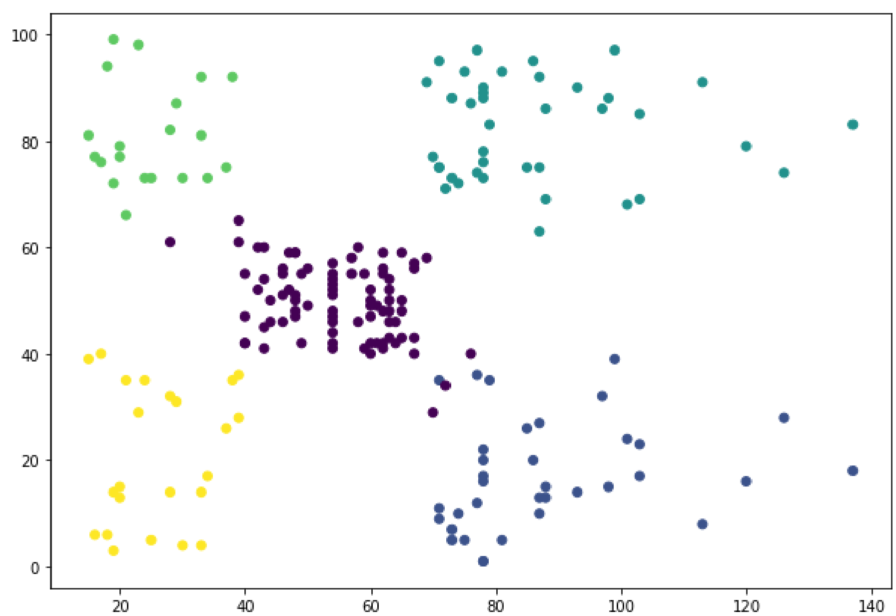
Visualizing all the Clusters

```
# plotting all the clusters and their Centroids

plt.figure(figsize=(8,8))
plt.scatter(X[Y==0,0], X[Y==0,1], s=50, c='green', label='Cluster 1')
plt.scatter(X[Y==1,0], X[Y==1,1], s=50, c='red', label='Cluster 2')
plt.scatter(X[Y==2,0], X[Y==2,1], s=50, c='yellow', label='Cluster 3')
plt.scatter(X[Y==3,0], X[Y==3,1], s=50, c='violet', label='Cluster 4')
plt.scatter(X[Y==4,0], X[Y==4,1], s=50, c='blue', label='Cluster 5')

# plot the centroids
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], s=100, c='cyan', label='Centroids')

plt.title('Customer Groups')
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.show()
```

The clustering visualization looks something like:

For agglomerative clustering:

```
plt.figure(figsize=(10, 7))
plt.scatter(customer_data['Annual Income (k$)'], customer_data['Spending Score (1-100)'], c=cluster.labels_)
```

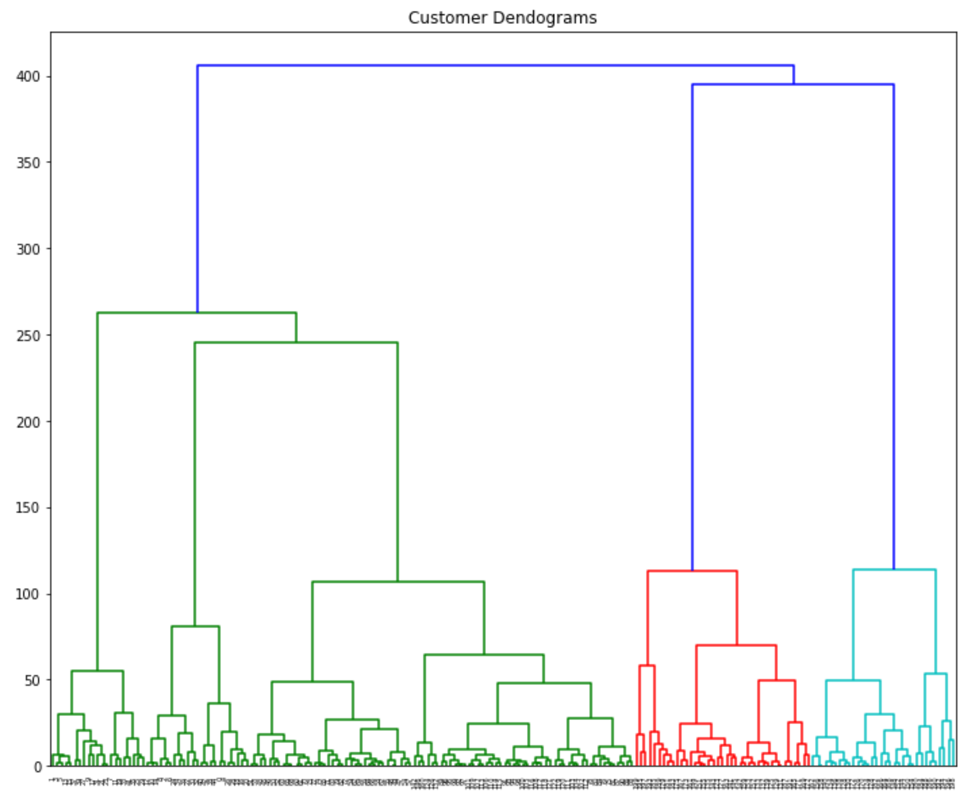


For K-means:

Each point on the scatterplot represents where customer lies with respect to their age and income. There are five distinct segments in the data set there are also some points with extreme values which may be interpreted as outliers clustering algorithms. We have to find these segments in the data and label each record with the cluster or segment that it belongs to based on an understanding of the variables that characterize each cluster, you can assign a name or meaning to each of these clusters.

Plotting the Dendrogram:

```
[ ]  import scipy.cluster.hierarchy as shc

     plt.figure(figsize=(12, 10))
     plt.title("Customer Dendograms")
     dend = shc.dendrogram(shc.linkage(X, method='ward'))
```


Customer Dendograms
400
350
300
250
200
150
100
50
0