Unsupervised Learning: Clustering

Module 1 : Introduction to Clustering

Introduction:-

Welcome to the module on 'Unsupervised Learning'. In the previous modules, you learnt about several supervised learning techniques such as regression and classification. These techniques use a training set to make the algorithm learn, and then apply what is learnt to new, unseen data points.

In this module, you will be introduced to unsupervised learning.

In this session

You will start by learning what “clustering” is. It is an unsupervised learning technique, where you try to find patterns based on similarities in the data. Then, you will be introduced to a case study that shows the applicability of clustering in the industry.

You will learn the two most commonly used types of clustering algorithms - **K-Means Clustering** and **Hierarchical Clustering**, as well as their application in Python. Then, you will also look at what segmentation is and how it is different from clustering.

Understanding Clustering:-

In the previous modules, you saw various supervised machine learning algorithms. Supervised machine learning algorithms make use of labelled data to make predictions.

For example, an email will be classified as spam or ham, or a bank’s customer will be predicted as ‘good’ or ‘bad’. You have a target variable Y which needs to be predicted.

On the other hand, in unsupervised learning, you are not interested in prediction because you do not have a target or outcome variable. The objective is to discover interesting patterns in the data, e.g. are there any subgroups or ‘clusters’ among the bank’s customers?

Let’s learn clustering in detail.

So you saw the favourite tourist destinations of Prof. Dinesh and Rohit. You also saw the emerging pattern in the places preferred by the Professor and Rohit.

However, how does all this relate to the concept of unsupervised learning?

PRACTICAL APPLICATIONS OF CLUSTERING

1. **Customer Insight:** Say, a retail chain with so many stores across locations wants to manage stores at best and increase the sales and performance. Cluster analysis can help the retail chain to get desired insights on customer demographics, purchase behaviour and demand patterns across locations. This will help the retail chain for assortment planning, planning promotional activities and store benchmarking for better performance and higher returns.
2. **Marketing:** Cluster Analysis can help with In the field of marketing, Cluster Analysis can help in market segmentation and positioning, and to identify test markets for new product development.
3. **Social Media:** In the areas of social networking and social media, Cluster Analysis is used to identify similar communities within larger groups.
4. **Medical**: Cluster Analysis has also been widely used in the field of biology and medical science like human genetic clustering, sequencing into gene families, building groups of genes, and clustering of organisms at species.

In the next segment, you will be introduced to a real-life application of clustering — grouping customers of an online store into different clusters and making a separate targeted marketing strategy for each group. We will be using this example throughout the module.

Practical Example of Clustering - Customer Segmentation:-

In the last segment, you got a basic idea of what clustering is. So let's consider a real-life application of the unsupervised clustering algorithm.

You can download the data set for the case study from below. We will be using the same data for Python Lab.

Customer segmentation for targeted marketing is one of the most vital applications of the clustering algorithm. Here, as a manager of the online store, you would want to group the customers into different clusters, so that you can make a customised marketing campaign for each of the group. You do not have any label in mind, such as good customer or bad customer. You want to just look at patterns in customer data and then try and find segments. This is where clustering techniques can help you with segmenting the customers. Clustering techniques use raw data to form clusters based on common factors among various data points. This is exactly what will also be done in segmentation, where various people or products will be grouped together on the basis of

similarities and differences between them.

As a manager, you would have to decide what the important business criteria are on which you would want to segregate the customers. So, you would need a method or an algorithm that itself decides which customers to group together based on these criteria.

Sounds interesting? Well, that is the beauty of unsupervised learning, especially clustering. But before we conclude this introductory session, it would be best to get an industry perspective on the application of clustering in the world of analytics.

You saw that, for successful segmentation, the segments formed must be stable. This means that the same person should not fall under different segments upon segmenting the data on the same criteria. You also saw that segments should have **intra-segment homogeneity** and **inter-segment heterogeneity**. You will see in later sessions how this can be defined mathematically.

Now you will see what types of market segmentations are commonly used.

You saw that mainly 3 types of segmentation are used for customer segmentation:

- **Behavioural segmentation**: Segmentation is based on the actual patterns displayed by the consumer
- **Attitudinal segmentation**: Segmentation is based on the beliefs or the intents of people, which may not translate into similar action
- **Demographic segmentation**: Segmentation is based on a person's profile and uses information such as age, gender, residence locality, income, etc.

You will also learn in later sessions about the different types of behavioural segmentations used in the industry.

Additional reading

You can read more about business cases where clustering is used here

Summary:-

In this session, we covered the basics of unsupervised learning and also got a little idea about how clustering works. In the next sessions, you will go deeper into the details of clustering and learn about the 2 common clustering algorithms — the K-Means algorithm and the Hierarchical clustering algorithm.

Module 2 : K Means Clustering

Introduction:-

Welcome to the session on 'K-Means Clustering'. In the previous session, you got a basic idea of what unsupervised learning is. You also learnt about one such unsupervised technique called clustering. Now let's dive deeper into the concept and learn about the first common algorithm to achieve this unsupervised clustering — the K-Means algorithm.

In this session

You will learn about:

- The steps in the K-Means algorithm
- How to graphically visualise the steps of K-Means algorithm
- Practical considerations while using the K-Means algorithm

Euclidean Distance:-

In the previous segments, you got an idea about how clustering works - it groups the objects on the basis of their similarity or closeness to each other.

Now, the next important thing is to get into the nitty-gritty of how clustering algorithms generally work. You will learn about the 2 types of clustering methods - K-means and Hierarchical and how they go about doing the clustering process.

We have learnt that clustering works on the basis of grouping the observations which are the most similar to each other. What does this exactly mean?

In simple terms, the algorithm needs to find data points whose values are **similar** to each other and therefore these points would then belong to the same cluster. The method in which any clustering algorithm goes about doing that is through the method of finding something called a "**distance measure**". The distance measure that is used in K-means clustering is called the **Euclidean Distance** measure. Let's look at the following lecture to understand how this value is calculated.

As mentioned in the video above, the Euclidean Distance between the 2 points is measured as follows: If there are 2 points X and Y having n dimensions

$$X = (X_1, X_2, X_3, \ldots X_n)$$
$$Y = (Y_1, Y_2, Y_3, \ldots . Y_n)$$

Then the **Euclidean Distance D** is given as

$$D = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2 + \ldots (X_n - Y_n)^2}$$

The idea of distance measure is quite intuitive. Essentially, the observations which are closer or more similar to each other would have a low Euclidean distance and the observations which are farther or less similar to each other would have a higher Euclidean distance. **So can you now guess how the Clustering process would work based on the Euclidean distance?**

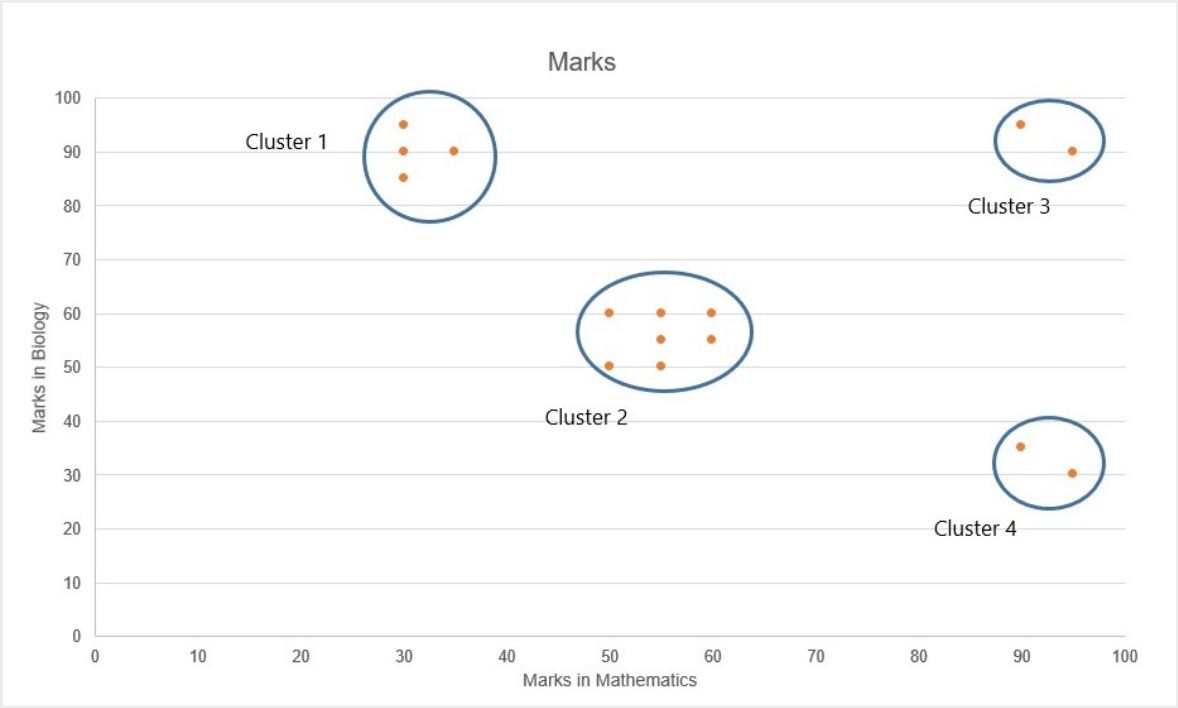
Now once you've computed the Euclidean distance, the next step is pretty straightforward for the Clustering Algorithm. All it has to do is compute these distances and then find out which observations or points have a low Euclidean distance between them, i.e. are closer to each other and then cluster them together.

Now answer the following questions

Centroid:-

The next concept that is crucial for understanding how clustering generally works is the idea of centroids. If you remember your high school geometry, centroids are essentially the centre points of triangles. Similarly, in the case of clustering, centroids are the **centre points of the clusters** that are being formed.

Now before going to the formula part, here is an intuition for the need of a centroid. Imagine you have the following clusters of the marks of a group of students in Mathematics and Biology and someone asks you to explain them. From a glance, you can easily interpret the 4 clusters that are being formed.



So the four clusters that are being formed are as follows:

Cluster 1: Students who have scored high marks in Bio, but poor marks in Maths
Cluster 2: Students who have scored average marks in Bio and Maths
Cluster 3: Students who have scored high marks in both Bio and Maths
Cluster 4: Students who have scored high marks in Maths, but poor marks in Bio

Now the above representation is fine and correct, but it is missing one crucial information - **the numerical order**. For example, when you want to compare two clusters say Cluster 1 and Cluster 2 can you say by how much marks on average do the students from Cluster 1 outperform or underperform the Cluster 2 students in a particular subject just by taking a look at the above visualisation alone? Is it by 10 marks? Or 15?

This is where the concept of **Centroids** come in handy. Listen to the following lecture to understand its importance and how it is calculated.

Therefore, as mentioned in the video, the Centroids are essentially **the cluster centres** of a group of observations that help us in **summarising the cluster's properties**. Thus as you saw in the video, the centroid value in the case of clustering is essentially the mean of all the observations that belong to a particular cluster. For example, in the dataset that you saw here,

Observation	Height	Weight	Age
A	175	83	22
B	165	74	25
C	183	98	24
D	172	80	24

The centroid is calculated by computing the mean of each and every column/ dimension that you have and then ordering them in the same way as above.
Therefore, Height-mean = ((175+165+183+172))/4 = 173.75
Weight-mean = ((83+74+98+80))/4 = 83.75
Age - mean = ((22+25+24+24))/4 =23.75
Thus the centroid of the above group of observations is (173.75, 83.75 and 23.75)

Now that you've understood how the centroids are calculated, answer the following question.

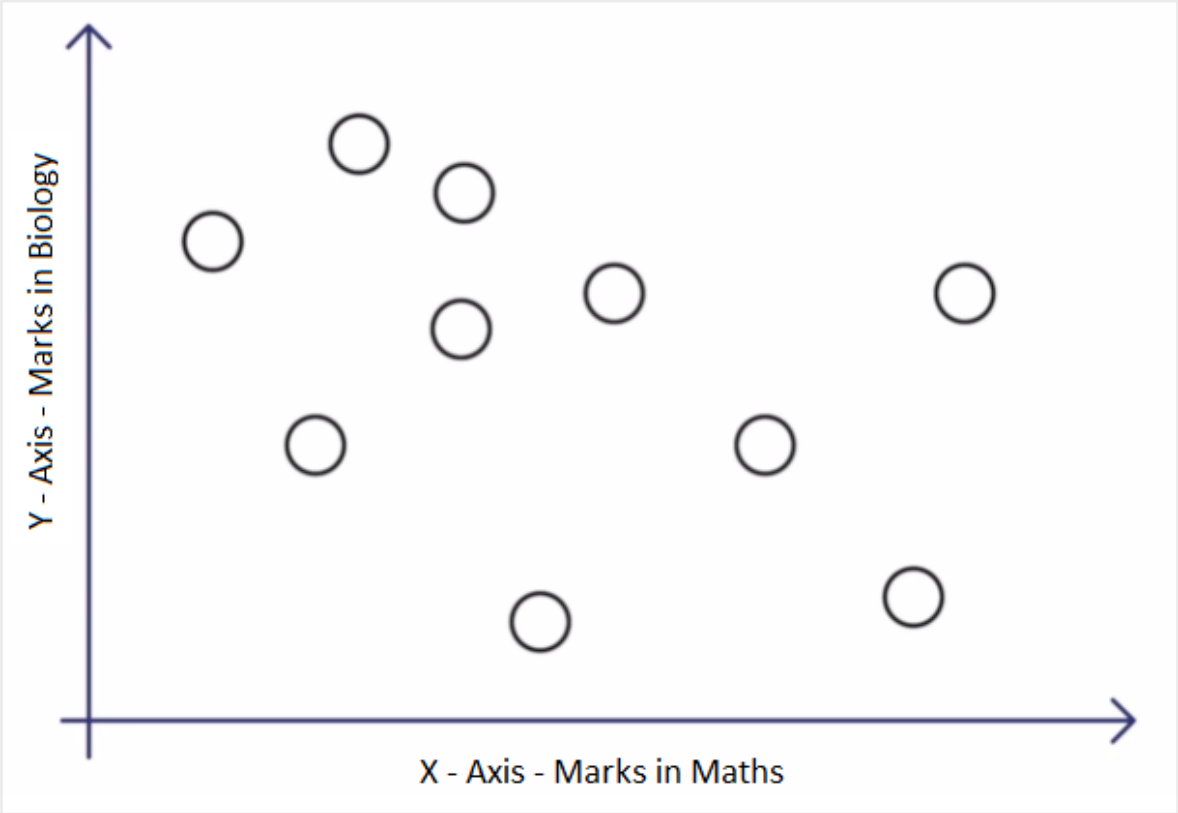
Steps of the Algorithm:-

Let's go through the K-Means algorithm using a very simple example. Let's consider a set of 10 points on a plane and try to group these points into, say, 2 clusters. So let's see how the K-Means algorithm achieves this goal.

Before moving ahead, think about the following problem. Let's say you have the data of 10 students and their marks in Biology and Math (as shown in the plot below). You want to divide them into two clusters so that you can see what kind of students are there in the class.

The y-axis shows the marks in Biology, and the x-axis shows the marks in Math.

Imagine two clusters dividing this data — one red and the other yellow. How many points would each cluster have?



Centroid

The K-Means algorithm uses the concept of the centroid to create K clusters. Before you move ahead, it will be useful to recall the concept of the centroid.

In simple terms, a centroid of n points on an x-y plane is another point having its own x and y coordinates and is often referred to as the geometric centre of the n points.

For example, consider three points having coordinates (x1, y1), (x2, y2) and (x3, y3). The centroid of these three points is the average of the x and y coordinates of the three points, i.e.
(x1 + x2 + x3 / 3, y1 + y2 + y3 / 3).

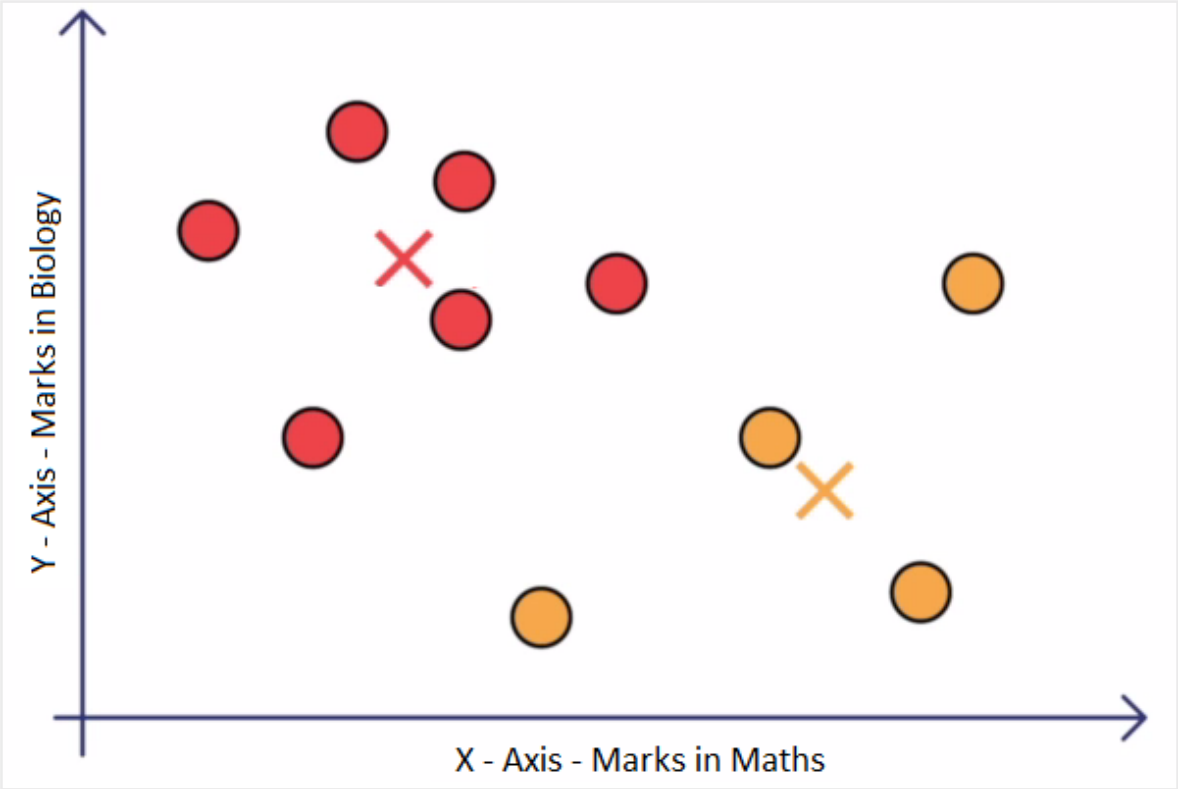
Similarly, if you have n points, the formula (coordinates) of the centroid will be:
(x1+x2.....+xn / n, y1+y2.....+yn / n).

So let's see how the K-Means algorithm achieves this goal.

Each time the clusters are made, the centroid is updated. The updated centroid is the centre of all the points which fall in the cluster associated with the centroid. This process continues till the centroid no longer changes, i.e. the solution converges.

Thus, you can see that the K-means algorithm is a clustering algorithm that takes N data points and groups them into K clusters. In this example, we had N

=10 points and we used the K-means algorithm to group these 10 points into K = 2 clusters.



Download the Excel file below. It is designed to give you hands-on practice of k-means clustering algorithm. The file contains a set of 10 points (with x and y coordinates in column A and B respectively) and two initial centres 1 and 2 (in columns F and G). Answer the questions below based on the Excel file.

You can download the solution worksheet for the above clustering activity here

K Means Algorithm:-

In the previous segment, we learned about K-means clustering and how the algorithm works using a simple example. We learned about how assignment and the optimisation work in K Means clustering, Now in this lecture, we will look K-means more algorithmically. We will be learning how the K Means algorithm proceeds with the assignment step and then with the optimisation step and will also be looking at the cost of function for the K-means algorithm.

Let's understand the K-means algorithm in more detail.

From the previous lecture, we understood that the algorithm's inner-loop iterates over two steps:

1. Assign each observation Xi(ith of x) to the closest cluster centroid μk(mew k)
2. Update each centroid to the mean of the points assigned to it.

In the next lecture, we will learn about the Kmeans cost function and will also see how to compute the cost function for each iteration in the K-means algorithm

So the cost function for the K-Means algorithm is given as:

$$J = \sum_{i=1}^{n} ||X_i - \mu_{k(i)}||^2 = \sum_{k=1}^{K} \sum_{i?C_k} ||X_i - \mu_k||^2$$

Now in the next video, we will learn what exactly happens in the assignment step? and we will also look at how to assign each data point to a cluster using the K-Means algorithm assignment step.

[Note: At 1:43 where the Prof explains the optimisation step, the values in the column - μ_1 and μ_2 should be **X1** and **X2**]

In the assignment step, we assign every data point to K clusters. The algorithm goes through each of the data points and depending on which cluster is closer, in our case, whether the green cluster centroid or the blue cluster centroid; It assigns the data points to one of the 2 cluster centroids.

The equation for the assignment step is as follows:

$$Z_i = argmin||X_i - \mu_k||^2$$

Now having assigned each data point to a cluster, now we need to recompute the cluster centroids. In the next lecture, Prof.Dinesh will explain how to recompute the cluster centroids or the mean of each cluster.

In the optimisation step, the algorithm calculates the average of all the points in a cluster and moves the centroid to that average location.

The equation for optimisation is as follows:

$$\mu_k = \frac{1}{n_k} \sum_{i:z_i=k} X_i$$

The process of assignment and optimisation is repeated until there is no change in the clusters or possibly until the algorithm converges.

In the next segment, we will learn how to optimise the K-Means algorithm even further using the K-Means ++ algorithm

Additional Reading

- You can also look K-Means algorithm as a coordinate descent problem and how to achieve global minima in the K-Means cost function. Please take a look at this optional segment to learn further

K Means++ Algorithm:-

We looked in the previous segment that for K-Means optimisation problem, the algorithm iterates between two steps and tries to minimise the objective function given as,

$$Z_i = argmin||X_i - \mu_k||^2$$

To choose the cluster centres smartly, we will learn about K-Mean++ algorithm. K-means++ is just an initialisation procedure for K-means. In K-means++ you pick the initial centroids using an algorithm that tries to initialise centroids that are far apart from each other.

Let's understand the algorithm in detail in the next lecture.

To summarise, In K-Means++ algorithm,

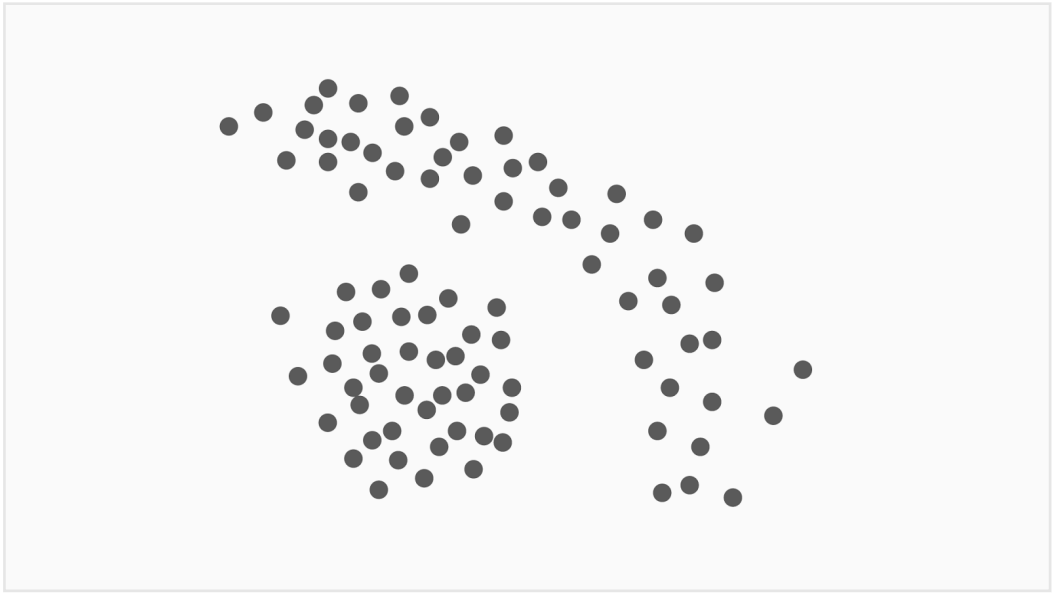
1. We choose one data point as the cluster centre at random.
2. For each data point X_i, We compute the distance between X_i and the nearest centre that had already been chosen.
3. Now, we choose the next cluster centre using the weighted probability distribution where a point X is chosen with probability proportional to $d(X)^2$.
4. Repeat Steps 2 and 3 until K centres have been chosen.

Visualising the K Means Algorithm:-

Let's see the K-Means algorithm in action using a visualisation tool. This tool can be found on naftaliharris.com. You can go to this link after watching the video below and play around with the different options available to get an intuitive feel of the K-Means algorithm.

Upon trying the different options, you may have noticed that the final clusters that you obtain vary depending on many factors, such as choice of the initial cluster centres and the value of K, i.e. the number of clusters that you want. You will understand these factors and other practical considerations while using the K-means algorithm in more detail in the next segment.

Now look at the given image and answer the question that follows:



Practical Consideration in K Means Algorithm:-

Let's understand some of the factors that can impact the final clusters that you obtain from the K-means algorithm. This would also give you an idea about the issues that you must keep in mind before you start to make clusters to solve your business problem.

Thus, the major practical considerations involved in K-Means clustering are:

- The number of clusters that you want to divide your data points into, i.e. the value of K has to be pre-determined.
- The choice of the initial cluster centres can have an impact on the final cluster formation.
- The clustering process is very sensitive to the presence of outliers in the data.
- Since the distance metric used in the clustering process is the Euclidean distance, you need to bring all your attributes on the same scale. This can be achieved through standardisation.
- The K-Means algorithm does not work with categorical data.
- The process may not converge in the given number of iterations. You should always check for convergence.

You will understand some of these issues in detail and also see the ways to deal with them when you implement the K-means algorithm in Python.

Now let's look in detail how to choose K for K-Means algorithm.

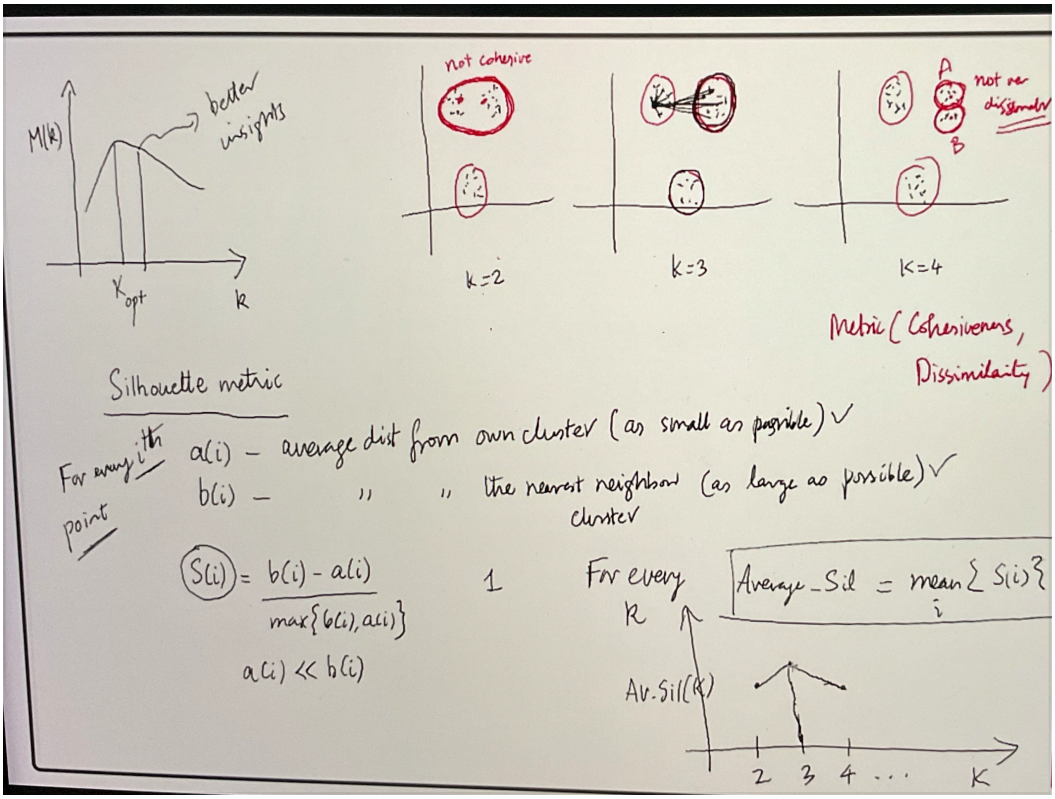
Having understood about the approach of choosing K for K-Means algorithm, we will now look at silhouette analysis or silhouette coefficient. Silhouette coefficient is a measure of how similar a data point is to its own cluster (cohesion) compared to other clusters (separation).

Let's look at it in detail in the next lecture.

So to compute silhouette metric, we need to compute two measures i.e. $a(i)$ and $b(i)$ where,

- $a(i)$ is the average distance from its own cluster(Cohesion).
- $b(i)$ is the average distance from the nearest neighbour cluster(Separation).

Now, let's look at how to combine cohesion and separation to compute the silhouette metric.



Additional reading
You can read more about K-Mode clustering here, We will be covering it in detail in the next section.

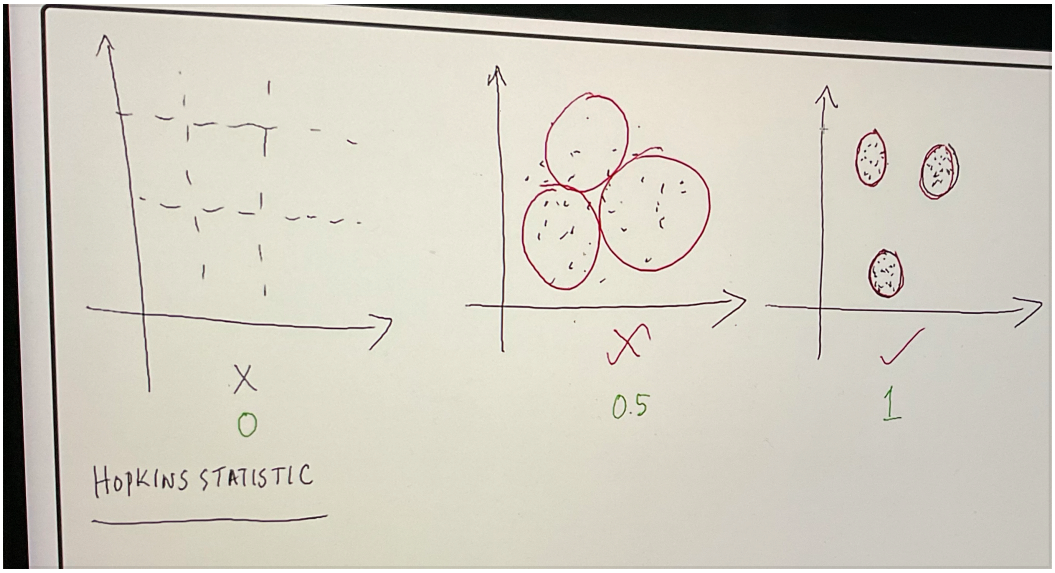
Cluster Tendency:-

Before we apply any clustering algorithm to the given data, it's important to check whether the given data has some meaningful clusters or not? which in general means the given data is not random. The process to evaluate the data to check if the data is feasible for clustering or not is known as the clustering tendency.

As we have already discussed in the previous lecture that the clustering algorithm will return K clusters even if that data does not have any clusters or have any meaningful clusters. So before proceeding for clustering, we should not blindly apply the clustering method and we should check the clustering tendency.

Let's look in detail at how it works.

To check cluster tendency, we use Hopkins test. Hopkins test examines whether data points differ significantly from uniformly distributed data in the multidimensional space.



Additional Resources
To read about Hopkins test in detail, please follow this link1, link2, remember that the document is described using R programming, please ignore it.

Summary:-

We covered a lot in this session. We started with understanding the K-Means intuitively by grouping the 10 random points in 2 clusters.

The algorithm begins with choosing K random cluster centres.

Then the 2 steps of **Assignment and Optimisation** continue iteratively till the clusters stop updating. This gives you the most optimal clusters — the clusters with minimum intra-cluster distance and maximum inter-cluster distance.

You also saw the different practical issues that need to be considered while employing clustering to your data set. You need to choose **how many clusters** you want to group your data points into. Secondly, the K-means algorithm is **non-deterministic**. This means that the final outcome of clustering can be different each time the algorithm is run even on the same data set. This is because, as you saw, the final cluster that you get can vary by the choice of the initial cluster centres.

You also saw that the **outliers** have an impact on the clusters and thus outlier-infested data may not give you the most optimal clusters. Similarly, since the most common measure of the distance is the Euclidean distance, you would need to bring all the attributes into the same scale using **standardisation**.

You also saw that you cannot use categorical data for the K-Means algorithm. There are other customised algorithms for such categorical data.

Module 3 : Executing K Means in Python

Introduction:-

Welcome to the session on 'Executing K-Means in Python'. In the previous sessions, you got a basic understanding of what clustering is and how you can use the K-Means algorithm to cluster objects. In this session, you will see the implementation of the K-Means algorithm in Python on the Online Retail case study that was introduced earlier.

In this session

You will learn about

- Data preparation
- How to make the clusters
- Decide the optimal number of clusters
- How to interpret the results

Data Understanding and Data Cleaning:-

In this session, you'll learn how to create clusters on the basis of K-means clustering algorithm in Python. Before that, you need to understand the dataset that we'll be using for this demonstration. **You can download the data set for the case study from this link here.**

Also, you can get the final analysis python notebook from the link given below.

Let's hear from Prof. Dinesh as he explains the dataset to us.

Now let's observe how to understand the data using Python.

Since you'd be doing a pretty huge analysis it is always a good idea to write the steps first.

Now that you've understood what to do with the dataset, it's always a good idea to start with cleaning the data first.

In the next segment, we'll start with the data preparation part.

Data Preparation - I:-

Now that you've cleaned the data, the next step is to prepare it for the

modelling part. Take a look at the RFM part once again.

The next thing is monetary and frequency column creation.

Finally we need to analyse the recency part once again.

In the next segment we'll take a look at the scaling part of the analysis.

Data Preparation - II:-

The next important concepts that need to be applied in the data preparation stage are outlier treatment and standardisation of the data. Let's understand both of them in detail.

Now let's about outlier treatment.

Now let's go ahead and complete the preprocessing part of standardisation

Hopkins Statistics

One more important data preparation technique that we also need to do but have skipped in the demonstration is the calculation of the Hopkins Statistic. In python, you can use the following code snippet to pass a dataframe to the Hopkins statistic function to find if the dataset is suitable for clustering or not. You can simply copy-paste the code present in the code given below to the main dataset and analyse the Hopkins statistic value.

Notes regarding Hopkins Statistic

- You don't need to know how the algorithm of Hopkins Statistic works. The algorithm is pretty advanced and hence you don't need to know its workings but rather only interpret the value that it assigns to the dataframe.
- On multiple iterations of Hopkins Statistic, you would be getting multiple values since the algorithm uses some randomisation in the initialisation part of the code. Therefore it is advised to run it a couple of times before confirming whether the data is suitable for clustering or not.

Making the Clusters:-

Now let's begin the modelling part by creating the clusters using the SKlearn's K-means algorithm package.

Optimal Number of Clusters:-

Now you might be thinking why the number of clusters is taken as 4 and not

any other number. To find the optimum number of clusters, we use two techniques - the elbow curve method and the silhouette score method. Let's learn about both of them in detail in the following lecture.

Next take a look at the silhouette score.

Let's go ahead and take it a step further in Python.

Now that you've understood the concept of finding the optimal number of clusters, the next segment would deal with analysing these clusters for further understanding our segmentation process.

Cluster Analysis:-

First , we need to assign the Cluster IDs that we generated to each of the datapoints that we have with us. Let's go ahead and do that.

The next step is interesting because we need to perform a bit of outlier analysis once again to understand how the dataset works here.

Now once the outlier analysis is completed, let's go ahead and analyse all the clusters that we have with us.

Let's Have Some Fun:-

You have learnt about how to make clusters using the K-Means algorithm. Let's use that knowledge to play around with clustering using K-Means.

Given below is a data set on the education status of Indian states.

The data contains state-level information on attributes such as the number of literates, illiterates, the number of literates who are graduate and above, etc.

But there’s a problem — the number of variables is quite large, and after forming the clusters, it may get difficult to describe each cluster’s characteristics.

This is not an uncommon problem. You may have noticed data sets having as many as 100-200 variables. There are techniques which are used to ‘reduce the number of variables while retaining as much information as possible’.

Two most common techniques, also called variable reduction techniques, are factor analysis and principal component analysis.

Lawmakers can cluster the states to find out which states have similar education statistics, and thus assign the budget accordingly. Clustering can

also help them figure out the best policies to make for these clusters.

You can download the data set and run the K-Means algorithm on this. You can try to make the clusters on different attributes.

You can see the effect of various elements of the K-Means clustering on the clusters formed. You can zoom in by selecting and double-clicking the map to look at the clusters.

For the purpose of creating the above visualisation, we have cleaned the data to include only the 2 factors under consideration. You could have chosen any other factors as well. Factors here mean the variables that you will use to build the clustering model. You can download the file below to answer the questions that follow:

Other Behavioural Segmentation Types:-

You have seen what RFM segmentation is. Now, you will look at other segmentation types commonly used in the industry.

You looked at RPI segmentation, which looks at what kind of relationship you have had with the person before, what type of person he/she is, and the intent of the person at the time of buying.

You also looked at the CDJ segmentation, which looks at the path that customers take while experiencing your product.

Now, let us turn our attention to another clustering technique — hierarchical clustering — in the next session.

Summary:-

So what did you learn in this session?

You learnt how to create clusters using the K-means algorithm in Python with the analysis of the Online Store data set. We wanted to group the customers of the store into different clusters based on their purchasing habits. The different steps involved were:

- Missing values treatment
- Data transformation
- Outlier treatment
- Data standardisation
- Finding the optimal value of K
- Implementing K Means algorithm
- Analysing the clusters of customers to obtain business insights

Once we are through with the data preparation, the K-means algorithm is quite easy to implement. All it takes is running the KMeans() function. The only ambiguous point you may notice here is that you need to decide the number of required clusters beforehand and in fact run the algorithm multiple times with a different number K before you can figure out the most optimal number of clusters.

This is also what happens in the industry practices that we run the algorithm multiple times with different values of K and then pick the clusters which make the most business sense. In fact, the k-means algorithm finds a large application in the industry. For example, it can be used to find out the most optimal centre to install mobile towers by clustering the customers geographically. Similarly, it has wide application in medical science, where say the patients can be clustered together on the basis of their symptoms, and then analysed to figure out the cause of their illness.

However, K means was just one of the clustering algorithm. In the next session, we will learn about another clustering algorithm called hierarchical clustering, which does not require you to decide the number of clusters beforehand.

Module 4 : Hierarchical Clustering

Introduction:-

Welcome to the session on 'Hierarchical Clustering'. In the previous sessions, you got a basic understanding of what clustering is and how you can use the K-Means algorithm to create clusters in your data set. You also saw the execution of the K-Means algorithm in Python.

In this session

You will learn about another algorithm to achieve unsupervised clustering. This is called **Hierarchical Clustering**. Here, instead of pre-defining the number of clusters, you first have to visually describe the similarity or dissimilarity between the different data points and then decide the appropriate number of clusters on the basis of these similarities or dissimilarities.

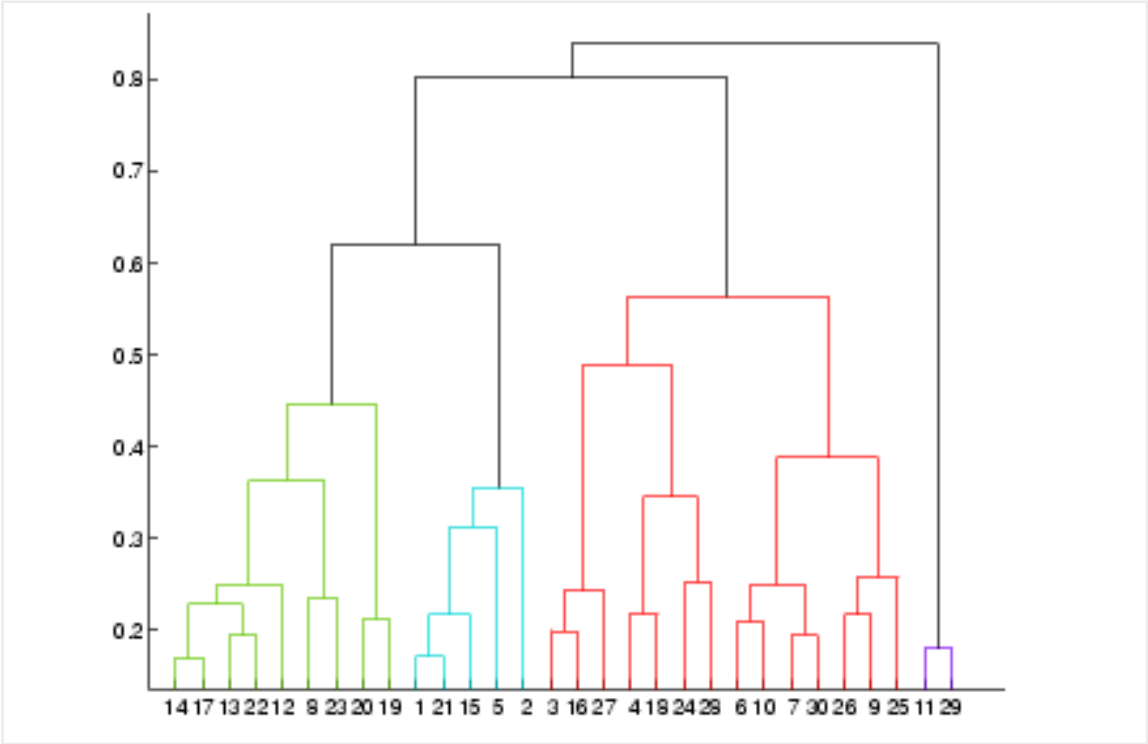
You will learn about:

- Hierarchical clustering algorithm
- Interpreting the dendrogram
- Cutting the dendrogram
- Types of linkages

Hierarchical Clustering Algorithm:-

One of the major considerations in using the K-means algorithm is deciding the value of K beforehand. The hierarchical clustering algorithm does not have this restriction.

The output of the hierarchical clustering algorithm is quite different from the K-mean algorithm as well. It results in an inverted tree-shaped structure, called the dendrogram. An example of a dendrogram is shown below.



Let's see how hierarchical clustering works.

In the K-Means algorithm, you divided the data in the first step itself. In the subsequent steps, you refined our clusters to get the most optimal grouping. In hierarchical clustering, the data is not partitioned into a particular cluster in a single step. Instead, a series of partitions/merges take place, which may run from a single cluster containing all objects to n clusters that each contain a single object or vice-versa.

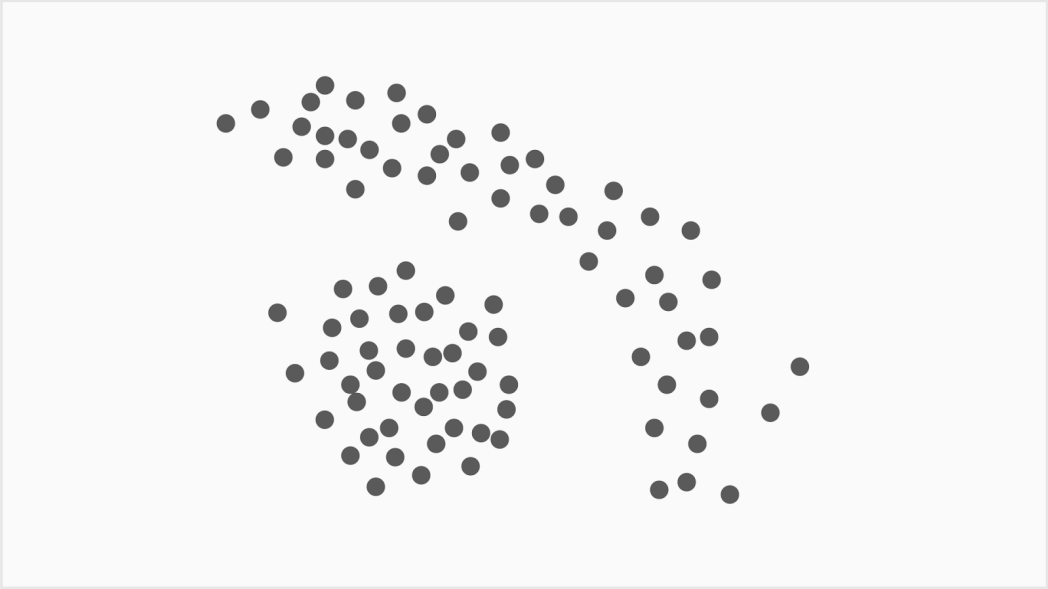
This is very helpful since you don't have to specify the number of clusters beforehand.

Given a set of N items to be clustered, the steps in hierarchical clustering are:

1. Calculate the NxN distance (similarity) matrix, which calculates the distance of each data point from the other
2. Each item is first assigned to its own cluster, i.e. N clusters are formed
3. The clusters which are closest to each other are merged to form a single cluster
4. The same step of computing the distance and merging the closest clusters is repeated till all the points become part of a single cluster

Thus, what you have at the end is the dendrogram, which shows you which data points group together in which cluster at what distance. You will learn more about interpreting the dendrogram in the next segment.

Look at the image given below and answer the question that follows.



Types of Linkages:-

In our example, we took the minimum of all the pairwise distances between the data points as the representative of the distance between 2 clusters. This measure of the distance is called a single linkage. Apart from using the minimum, you can use other methods to compute the distance between the clusters.

Let's see once again the different types of linkages.

- **Single Linkage:** Here, the distance between 2 clusters is defined as the shortest distance between points in the two clusters
- **Complete Linkage:** Here, the distance between 2 clusters is defined as the maximum distance between any 2 points in the clusters
- **Average Linkage:** Here, the distance between 2 clusters is defined as

the average distance between every point of one cluster to every other point of the other cluster.

You have to decide what type of linkage should be used by looking at the data. One convenient way to decide is to look at how the dendrogram looks. Usually, a single linkage-type will produce dendrograms which are not structured properly , whereas complete or average linkage will produce clusters which have a proper tree-like structure. You will see later what this means when you run the hierarchical clustering algorithm in Python.

Additional reading

You can read more about the type of linkages here, here and here.

Use the excel file given below to answer the questions that follow:

Let's recall what you have learnt in this session so far. You learnt about another clustering technique called Hierarchical clustering. You saw how it is different from K-Means clustering. One major advantage is that you do not have to pre-define the number of clusters. However, since you compute the distance of each point from every other point, it is time-consuming and needs a lot of processing power.

In the next segment, you will use the hierarchical clustering technique to actually make clusters using Python.

Hierarchical Clustering in Python:-

We will use the same online retail case study and data set that we used for the K-Means algorithm. For making the customer segments this time, we will use the hierarchical algorithm.

We will start at the point where we are done with the data preparation and already have the RFM dataset which has been treated for missing values and outliers, and is also standardised.

The hierarchical clustering involves 2 basic steps:

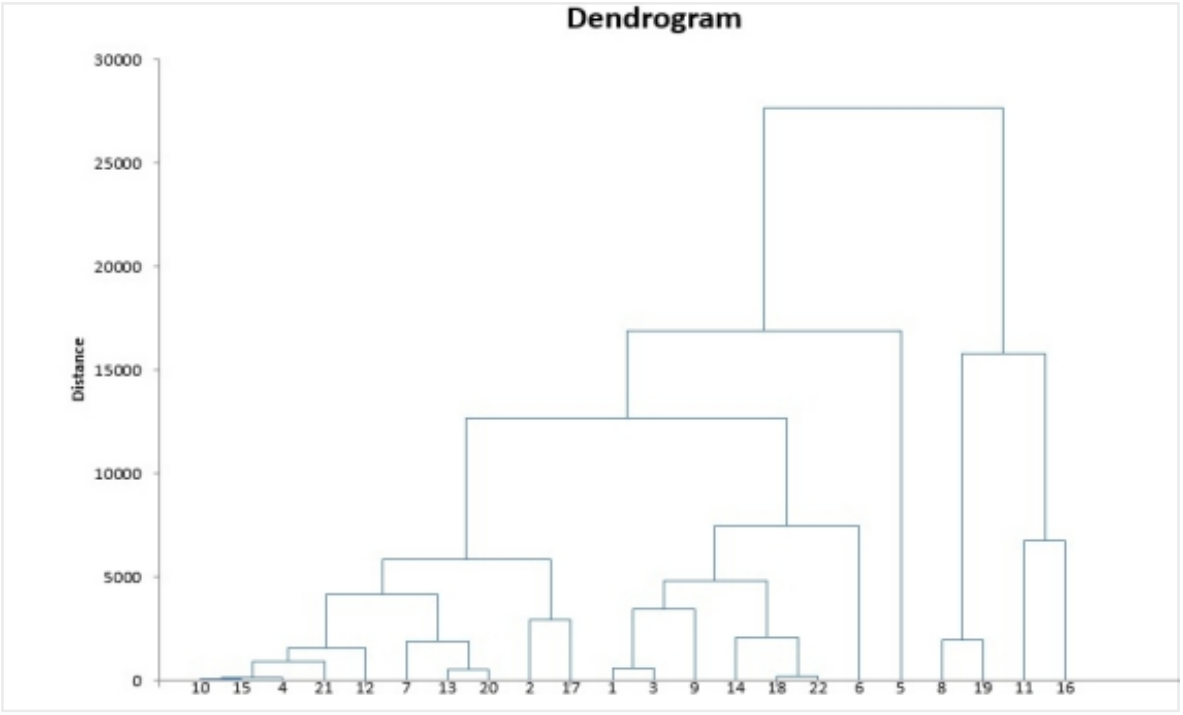
1. Creating the dendrogram
2. Cutting the dendrogram at an appropriate level

Now let's go ahead and utilise the single linkage method for clustering this dataset.

As you can clearly see, single linkage doesn't produce a good enough result for us to analyse the clusters. Hence, we need to go ahead and utilise the complete linkage method and then analyse the clusters once again.

After we got the clusterIDs for each customer, we then appended the obtained ClusterIDs to the RFM data set, and analysed the characteristics of each cluster to derive the business insights from the different customer segments or clusters, in the same way as you did for the K-Means algorithm.

Now look at the following dendrogram and answer the questions that follow.



Industry Insights:-

Now let's hear from our industry experts regarding the comparison between the K-Means algorithm and the Hierarchical clustering algorithm, before learning how to choose between the two based on your business problem.

So, you learnt that whether you use k-means or hierarchical clustering algorithm depends on your hardware and the data that you are dealing with.

Now, you will look at a good statistical hack to solve segmentation problems so that you get meaningful segments, where you will use both hierarchical and k-means algorithms to complement each other.

These insights were really helpful. You looked at how these clustering methods can be used to complement each other. You also looked at the differences between these methods, and the cases where you would prefer one method over the other.

Let's have some fun:-

You have learnt how about how to make clusters using the hierarchical clustering algorithm. Let's use that knowledge to play around with clusters. Again we will be using the example of education in Indian states.

Given below is a data set on the education status of Indian states.

You can download the data set and run the hierarchical algorithm on this. You can try to make the clusters on different attributes.

You can see the effect of various elements of the hierarchical clustering on the clusters formed.

For the purpose of creating the above visualisation, we have cleaned the data to include only the 2 factors under consideration. You can download the file below.

Summary:-

So what did you learn in this session?