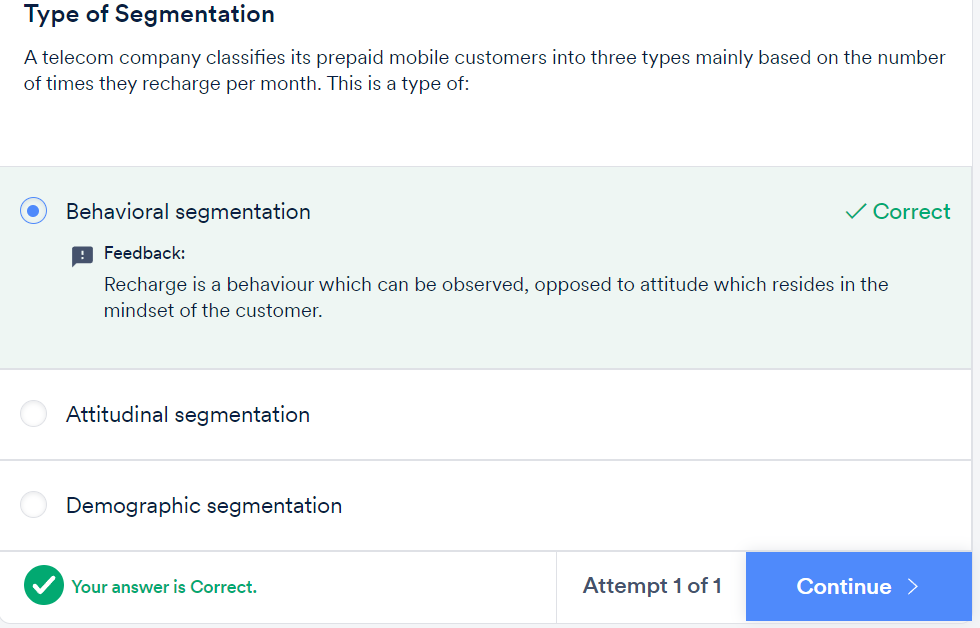
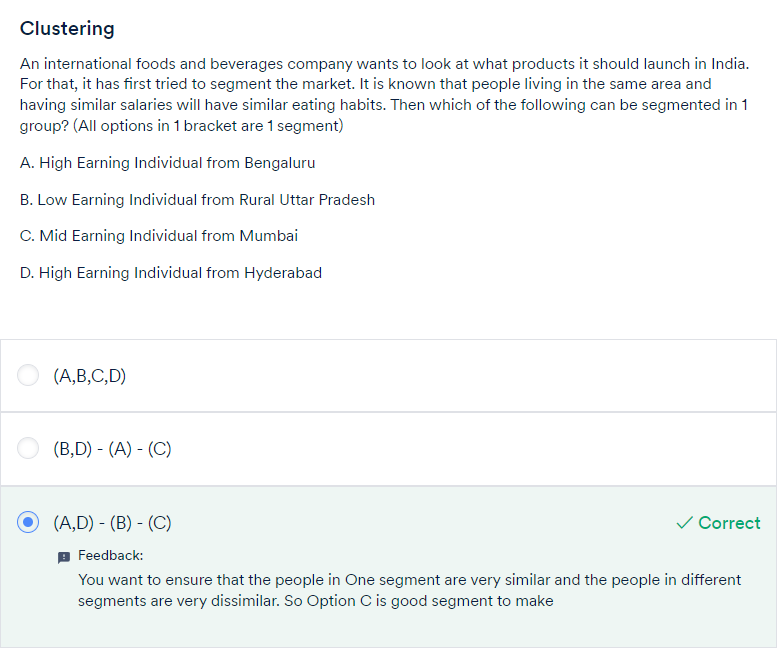
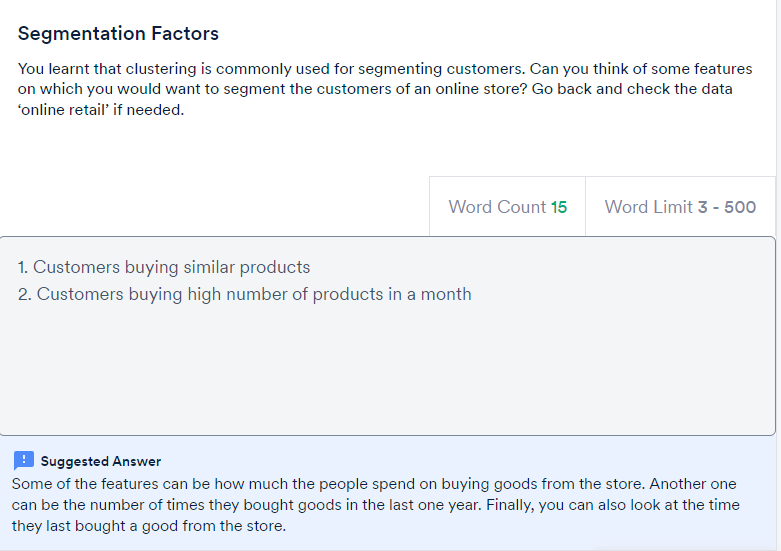
You saw that, for successful segmentation, the segments formed must be stable. This means that the same person should not fall under different segments upon segmenting the data on the same criteria. You also saw that segments should have **intra-segment homogeneity** and **inter-segment heterogeneity**.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

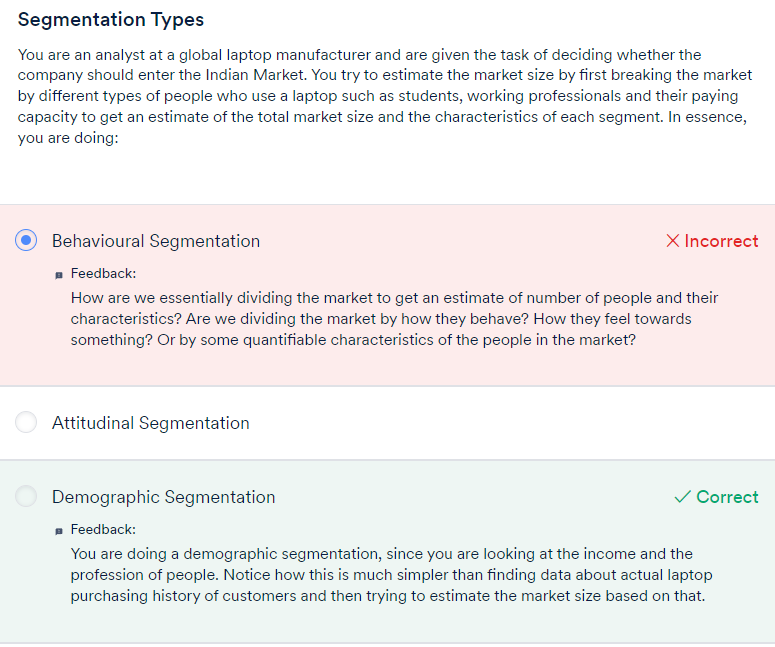
You saw that mainly 3 types of segmentation are used for customer segmentation:

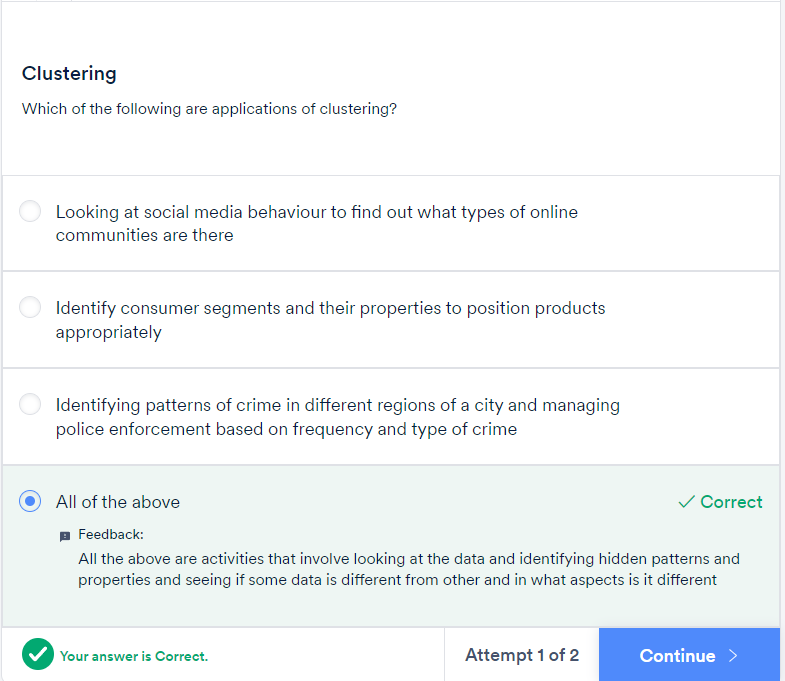
* **Behavioural segmentation**: Segmentation is based on the actual patterns displayed by the consumer
* **Attitudinal segmentation**: Segmentation is based on the beliefs or the intents of people, which may not translate into similar action
* **Demographic segmentation**: Segmentation is based on the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

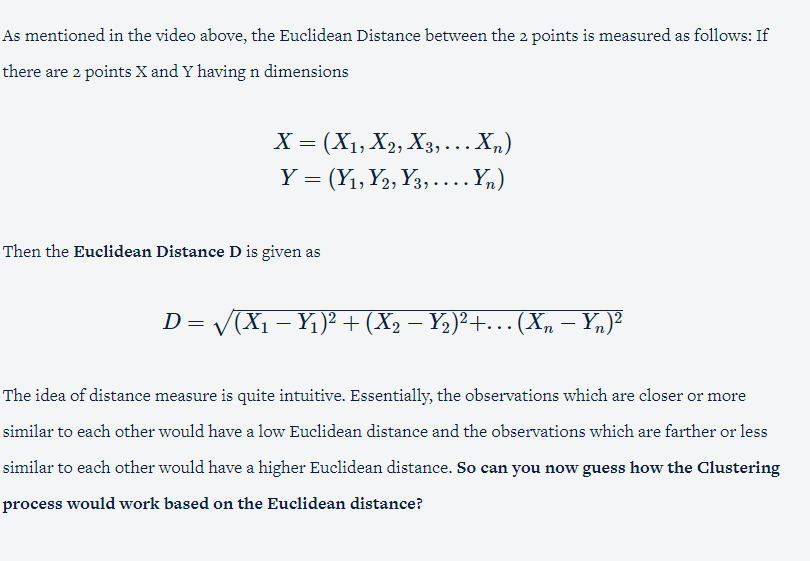


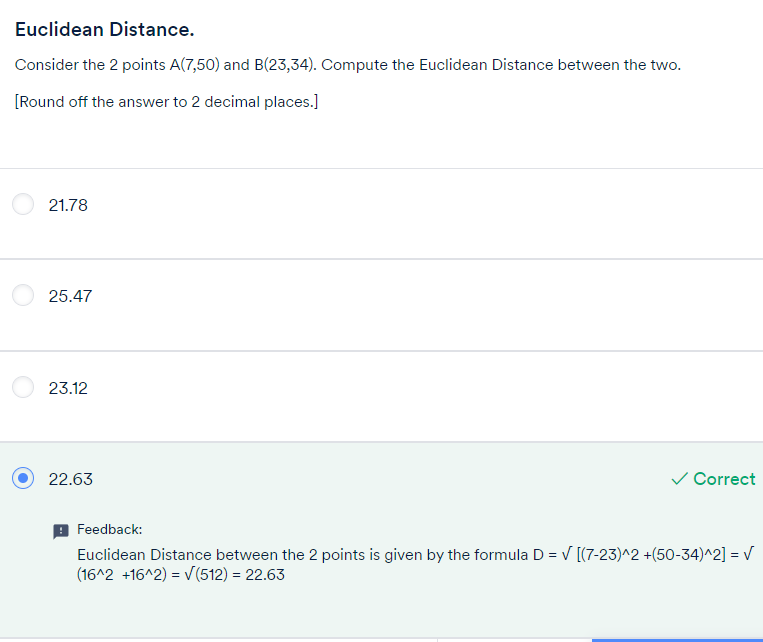




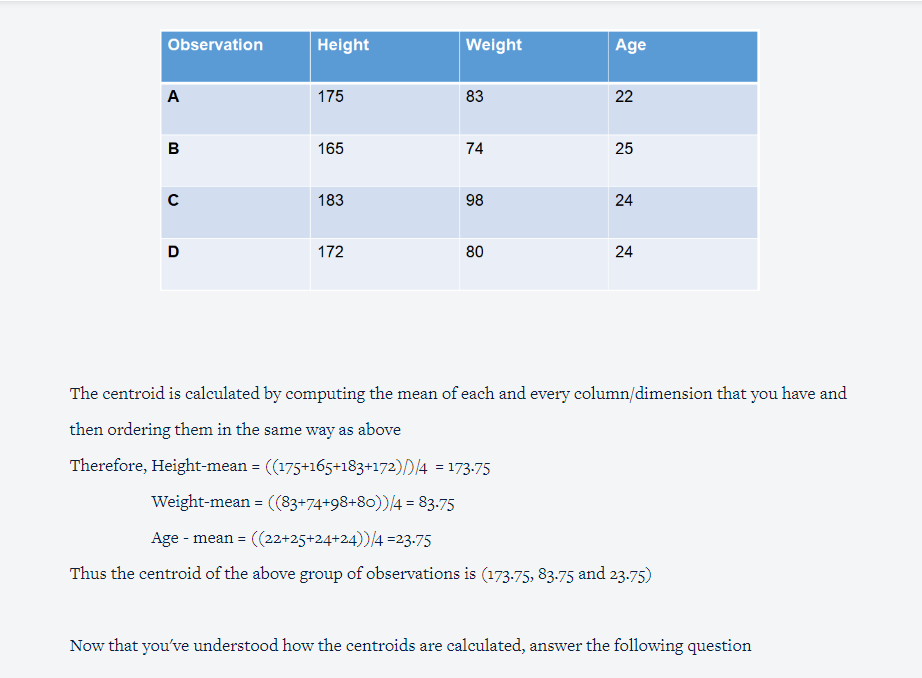


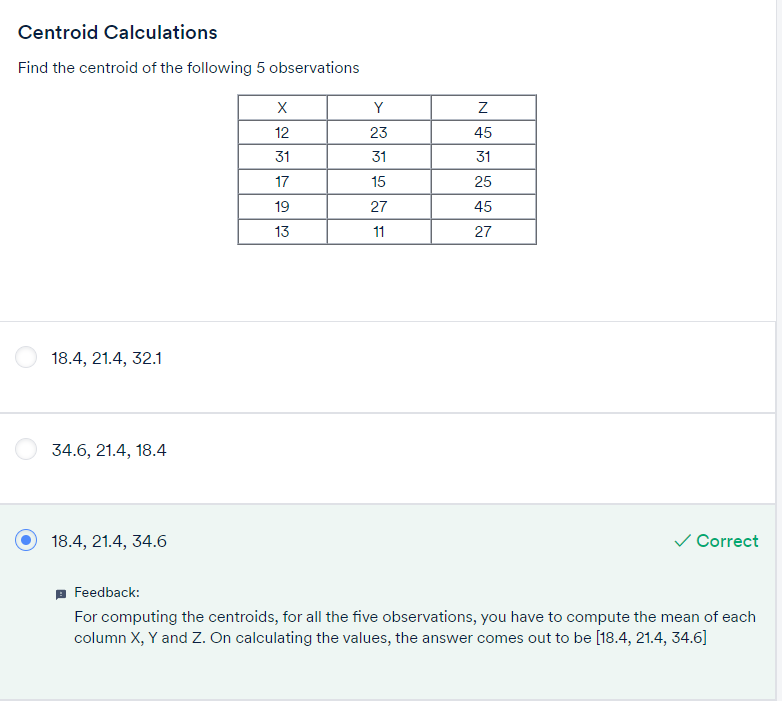






Therefore, as mentioned in the video, the Centroids are essentially**the cluster centres** of a group of observations that help us in **summarising the cluster's properties**. Thus as you saw in the video, the centroid value in the case of clustering is essentially the mean of all the observations that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

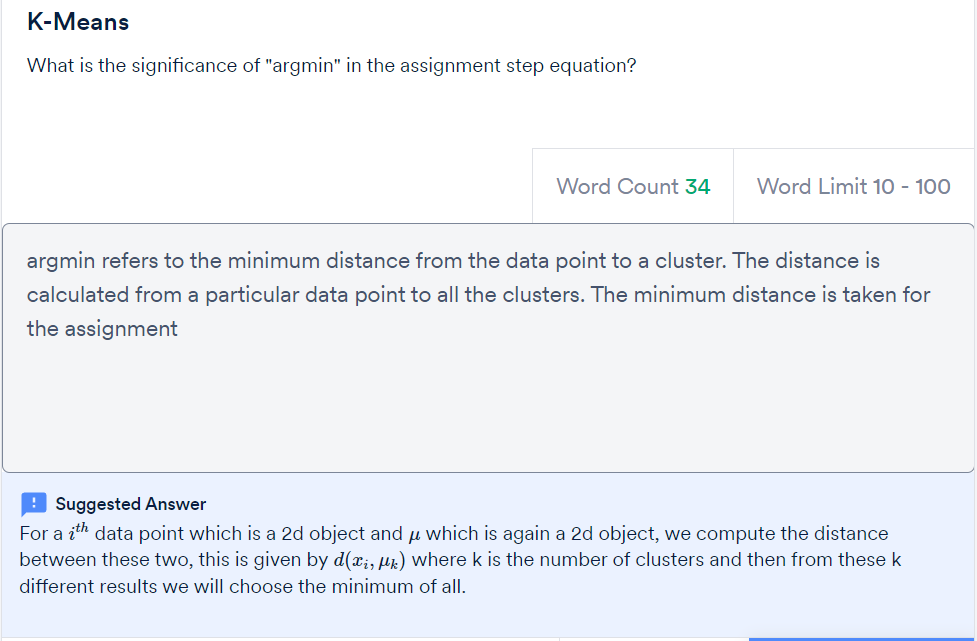




From the previous lecture, we understood that the algorithm’s inner-loop iterates over two steps:

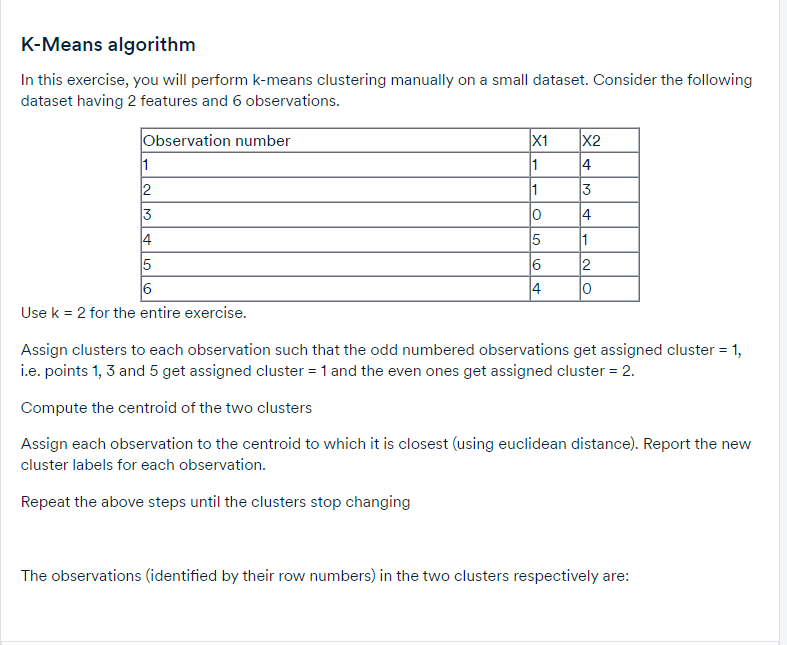
1. Assign each observation Xi to the closest cluster centroid μk
2. Update each centroid to the mean of the points assigned to it.

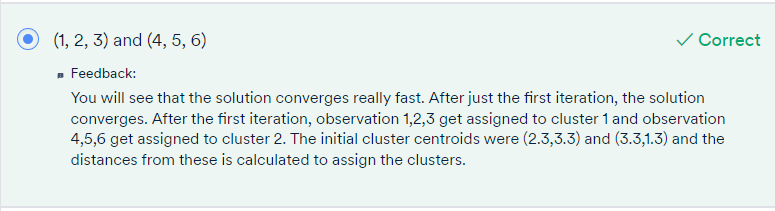
In the next lecture, we will learn about the Kmeans cost function and will also see how to compute the cost function for each iteration in the K-means algorithm.

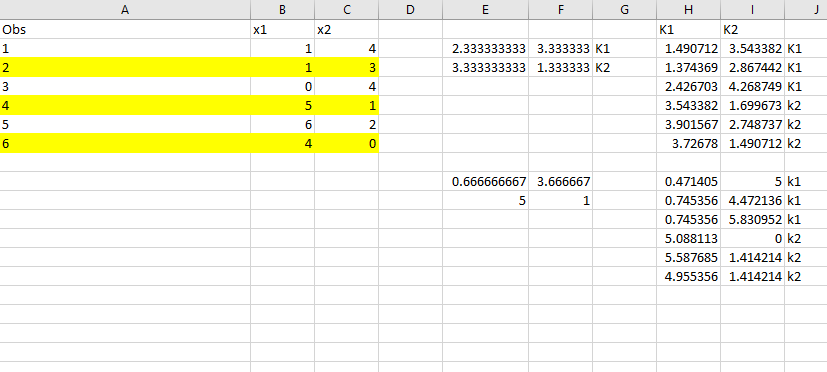


To summarise, In K-Means++ algorithm,

1. We choose one center as one of the data points at random.
2. For each data point Xi, We compute the distance between Xi and the nearest center that had already been chosen.
3. Now, we choose the next cluster center 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 centers have been chosen.





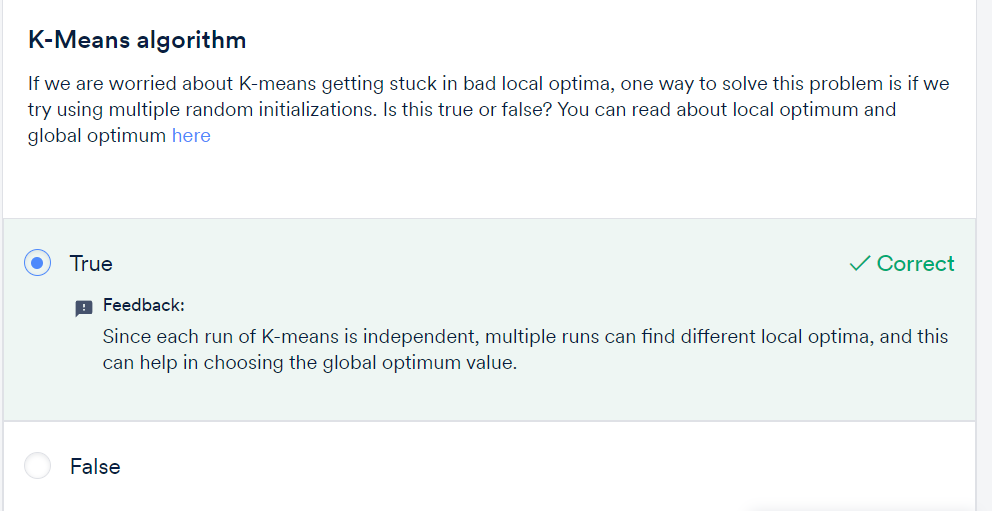


Thus, the major practical considerations involved in K-Means clustering are:

* 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.

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 own cluster(Cohesion).
* b(i) is the average distance from the nearest neighbour cluster(Separation).



The algorithm begins with choosing K random cluster centres.

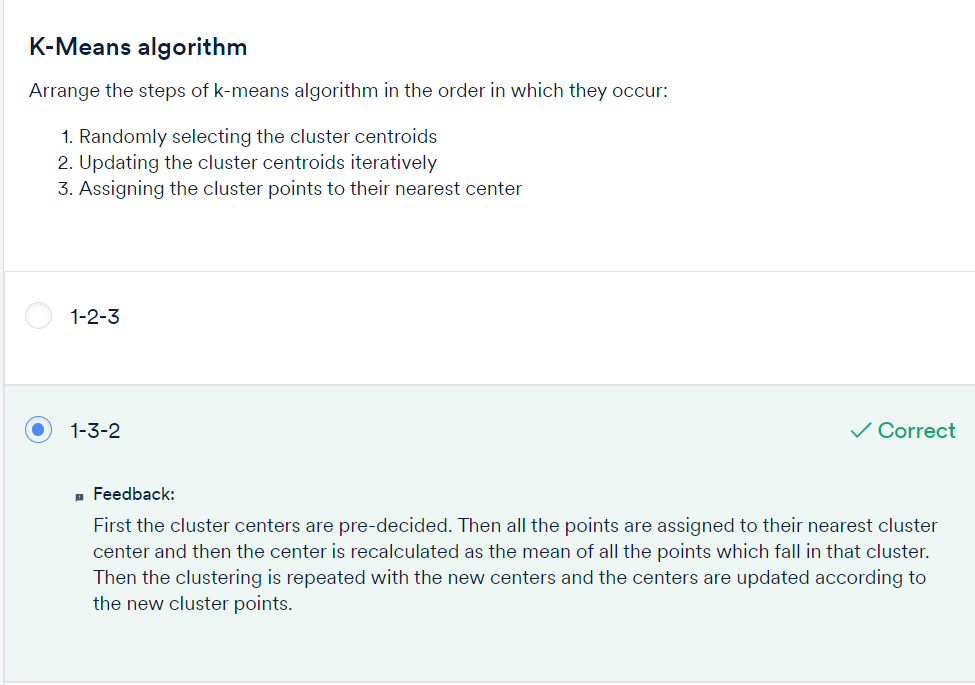
Then the 2 steps of **Assignment and Optimisation** continue iteratively till the clusters stop updating. This gives you the most optimal clusters — the clusters with minimum intra-cluster distance and maximum inter-cluster distance.

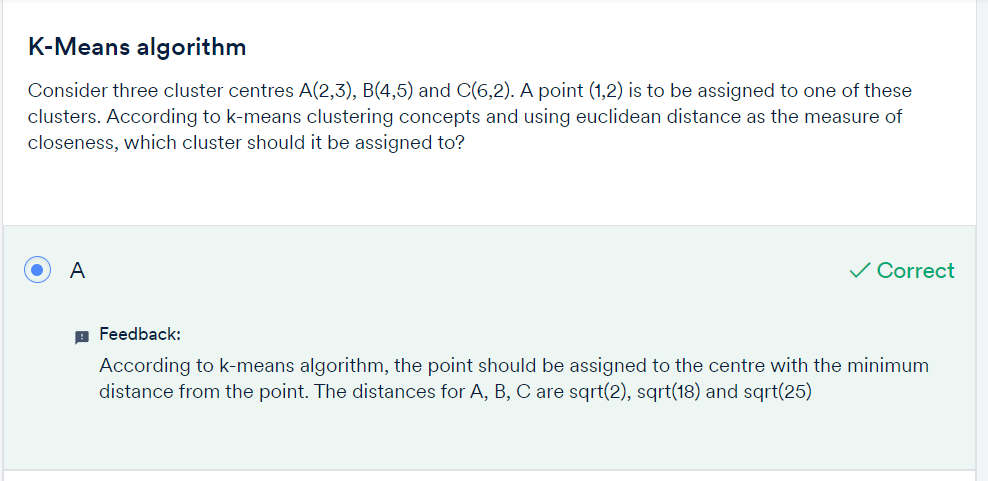
You also saw the different practical issues that need to be considered while employing clustering to your data set. You need to choose **how many clusters**you want to group your data points into. Secondly, the K-means algorithm is **non-deterministic**. This means that the final outcome of clustering 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.

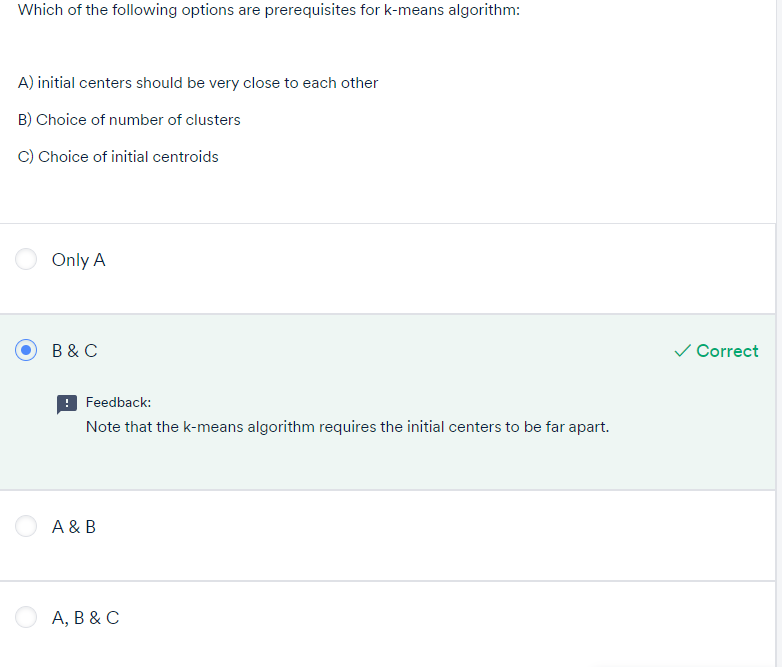
You also saw that the **outliers** have an impact on the clusters and thus outlier-infested data may not give you the most optimal clusters. Similarly, since the most common measure of the distance is the Euclidean distance, you would need to bring all the attributes into the same scale using **standardisation**.

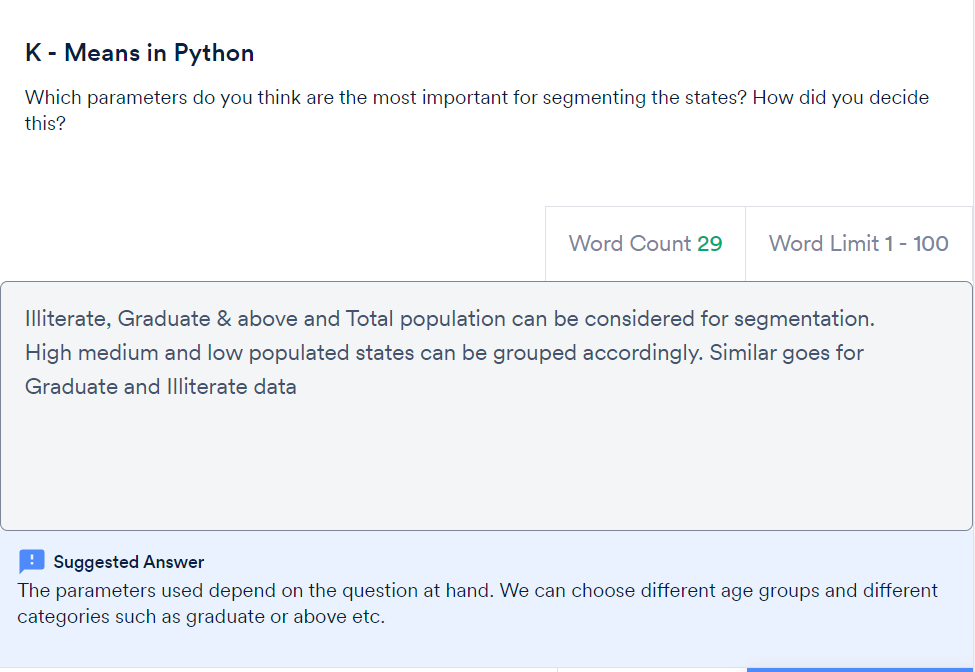
You also saw that you cannot use categorical data for the K-Means algorithm. There are other customised algorithms for such categorical data.

GRADED QUESTIONS:

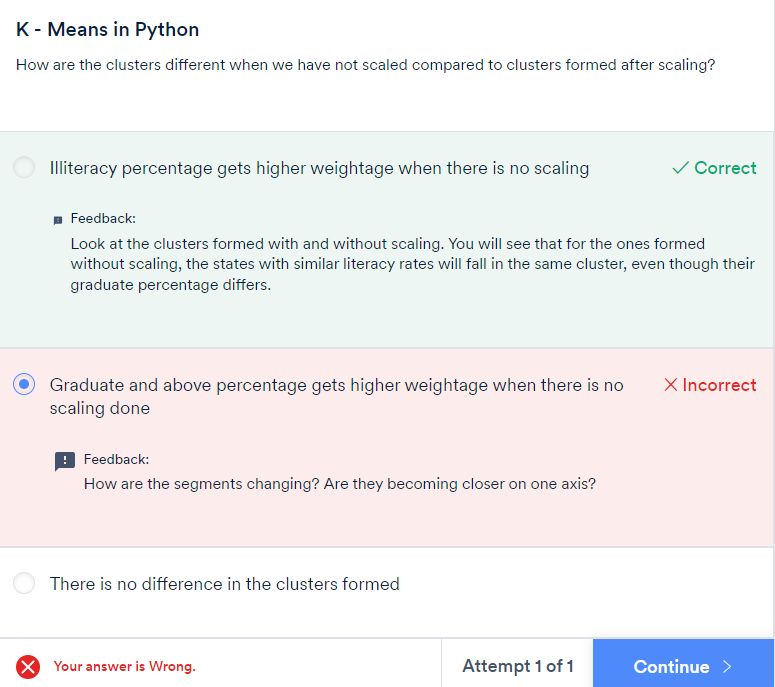












Behavior Segmentation:

1.

R – Recency : How recently did the customer purchase

F – Frequency : How often do they purchase

M : Monetary Value : How much do they spend

2.

R – Relationship: Past interaction with the company

P – Persona: Type of person

I – Intent: Intension at the time of purchase

3.

CDJ: Consumer Decision Journey : Based on customer life journey with brand or product

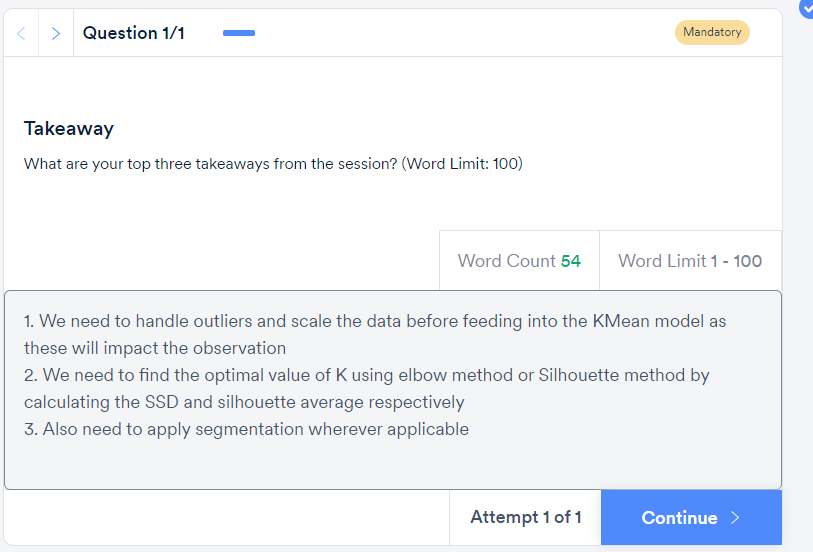
Summary:

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:

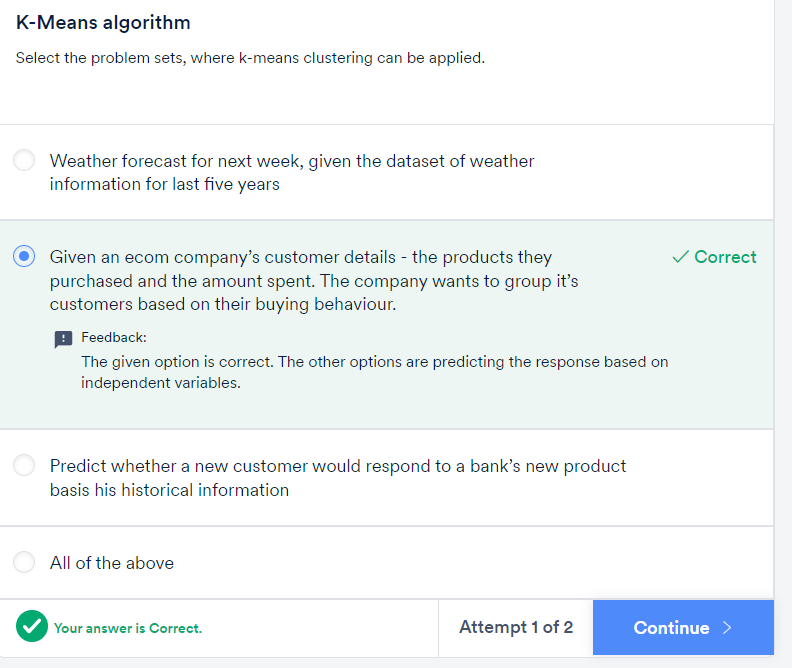
* Missing values treatment
* Data transformation
* Outlier treatment
* Data standardisation
* Finding the optimal value of K
* Implementing K Means algorithm
* Analysing the clusters of customers to obtain business insights

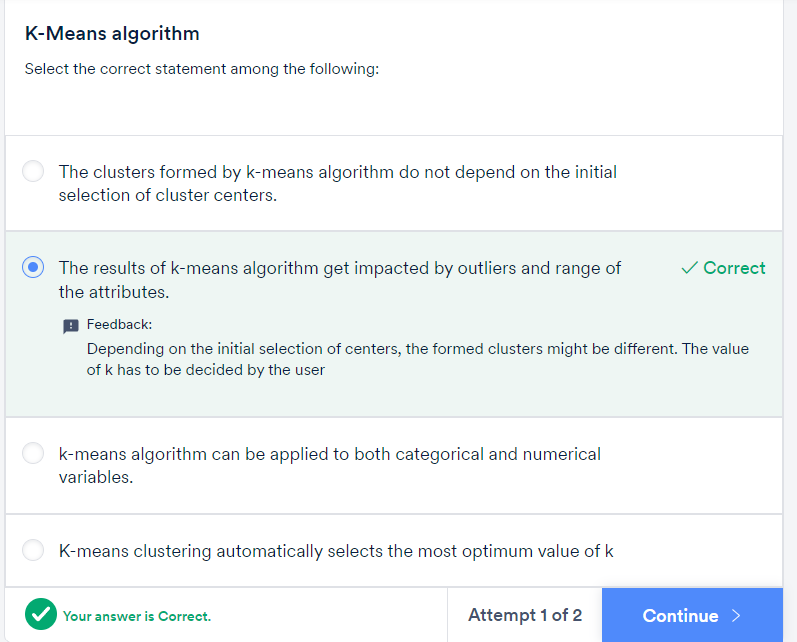
Once we are through with the data preparation, the K-means algorithm is quite easy to implement. All it takes is running the KMeans() function. The only ambiguous point you may notice here is that you need to decide the number of required clusters beforehand and in fact run the algorithm multiple times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

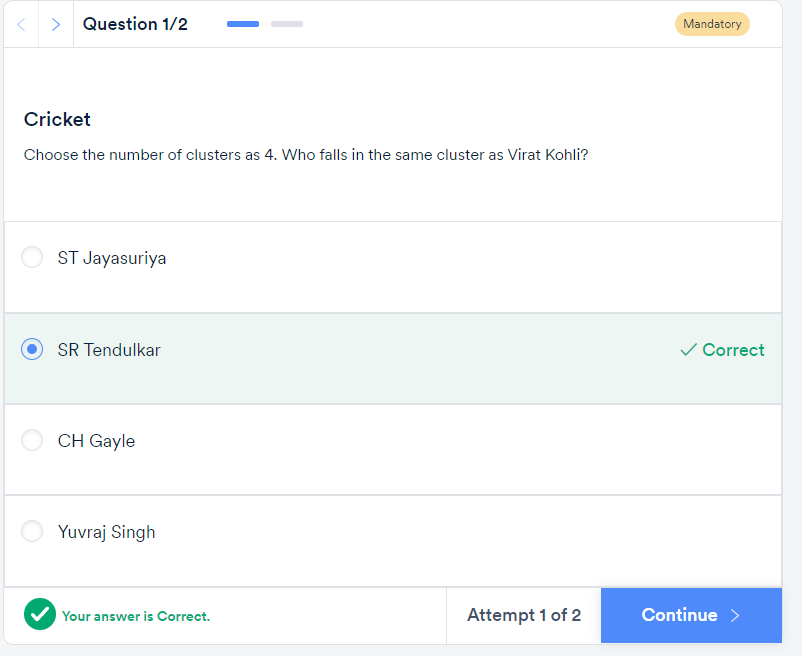
This is also what happens in the industry practices that we run the algorithm multiple times with different values of K and then pick the clusters which make the most business sense. In fact, the k-means algorithm finds large application in the industry. For example, it can be used to find out the most optimal centre to install the 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.

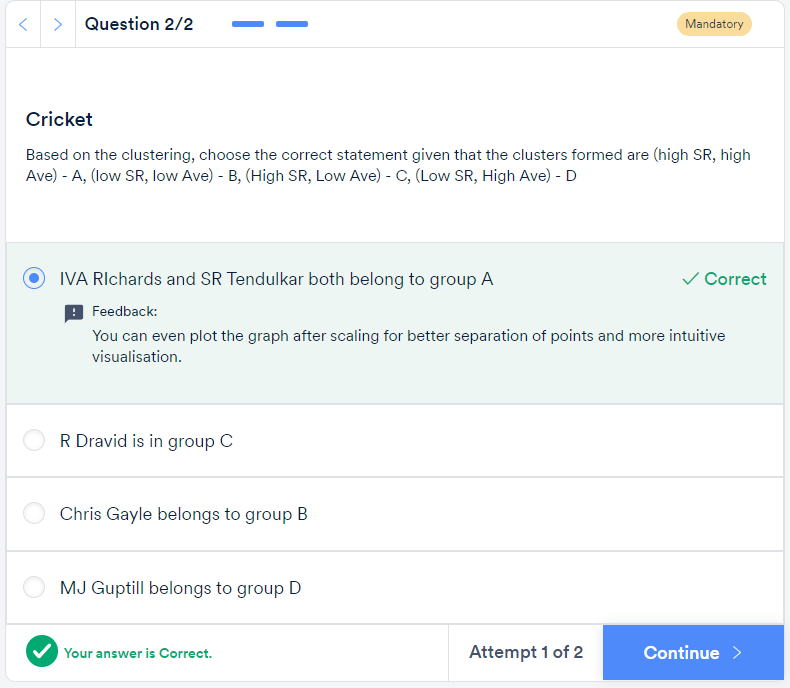


Graded Questions:





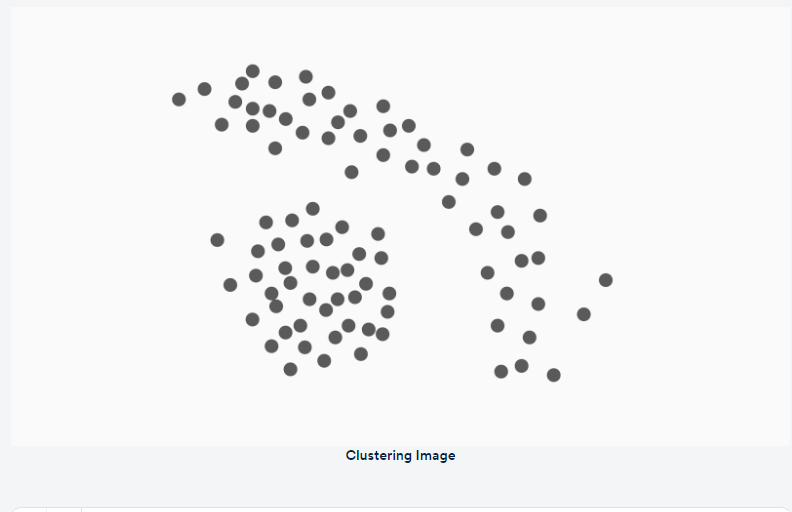


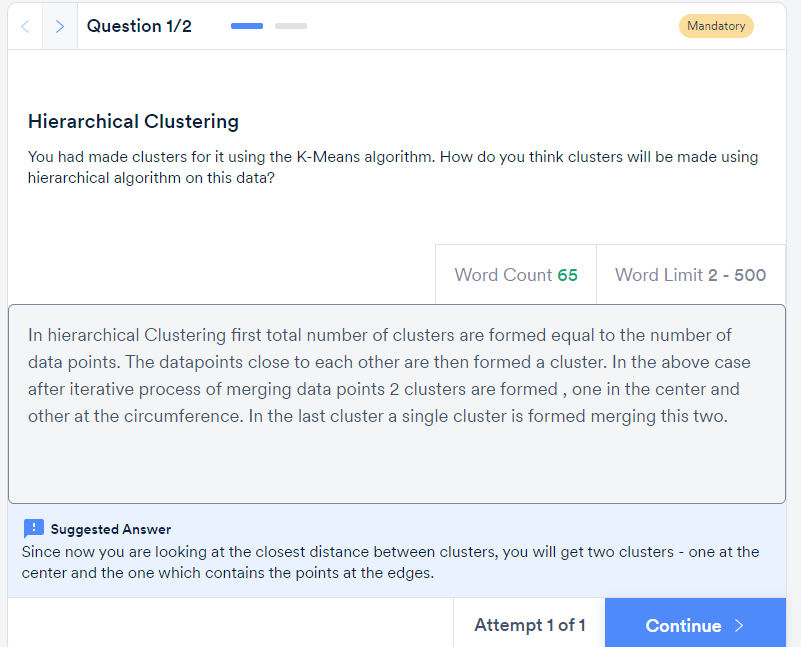


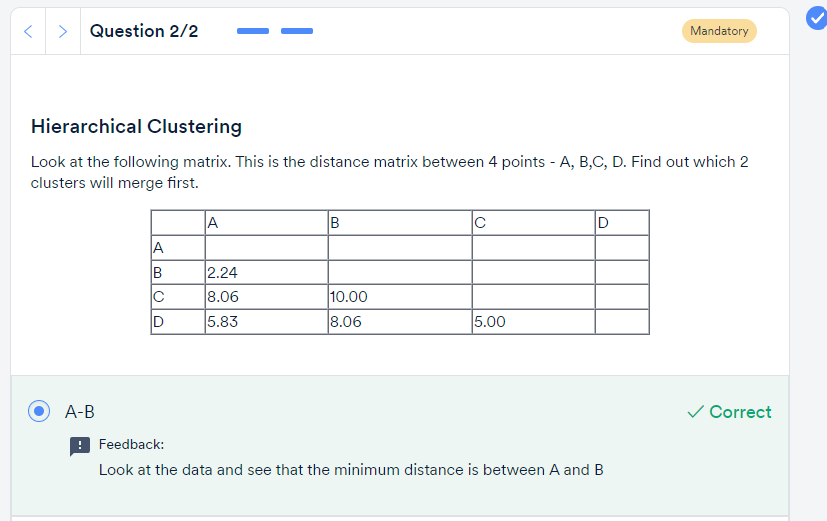
# Hierarchical Clustering:

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

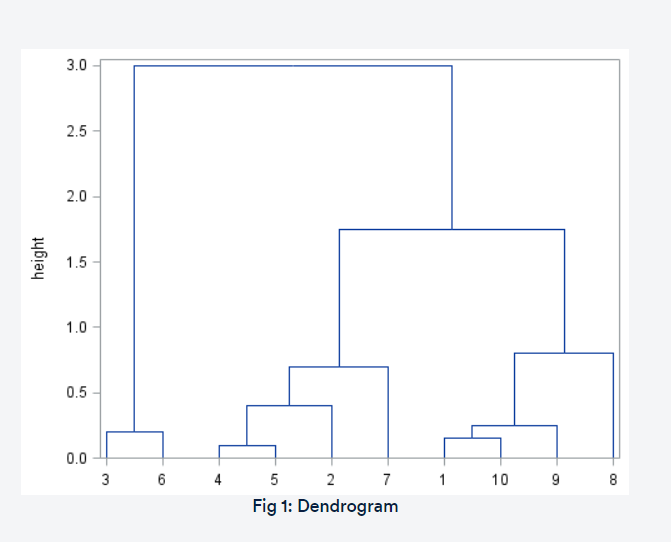
1. Calculate the NxN distance (similarity) matrix, which calculates the distance of each data point from the other
2. Each item is first assigned to its own cluster, i.e. N clusters are formed
3. The clusters which are closest to each other are merged to form a single cluster
4. The same step of computing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







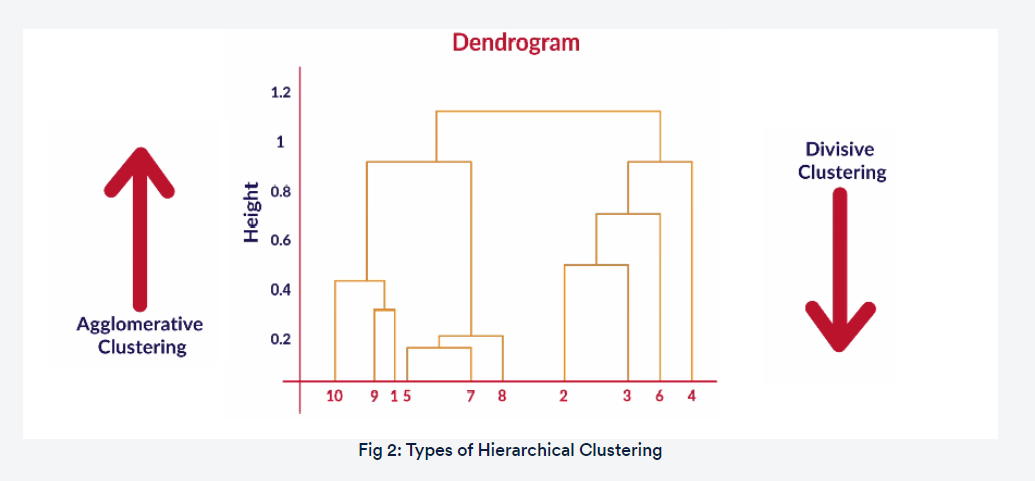
# Interpreting the Dendrogram

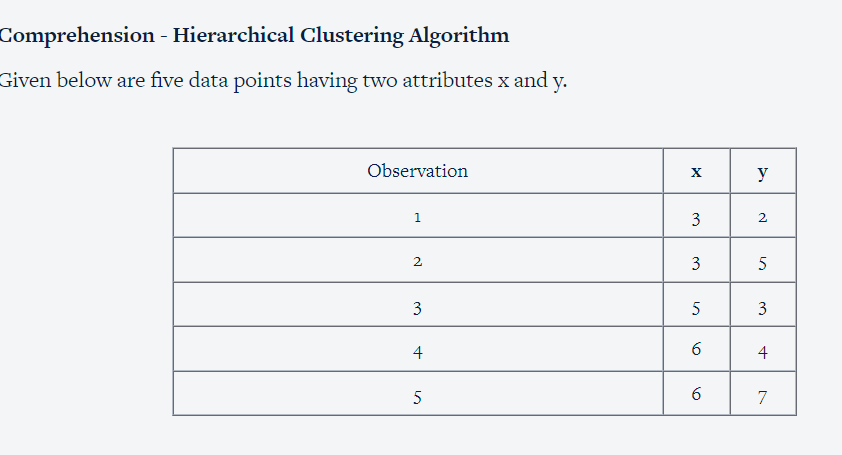


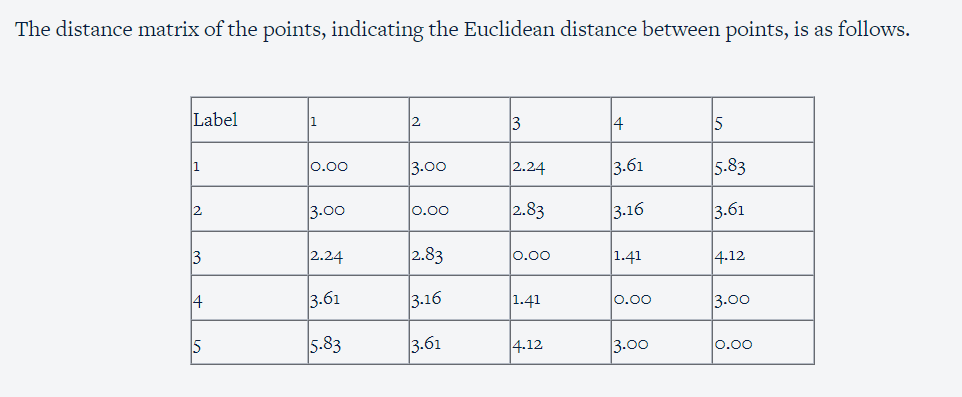
In the dendrogram shown above, samples 4 and 5 are the most similar and join to form the first cluster, followed by samples 1 and 10. The last two clusters to fuse together to form the final single cluster are 3-6 and 4-5-2-7-1-10-9-8.

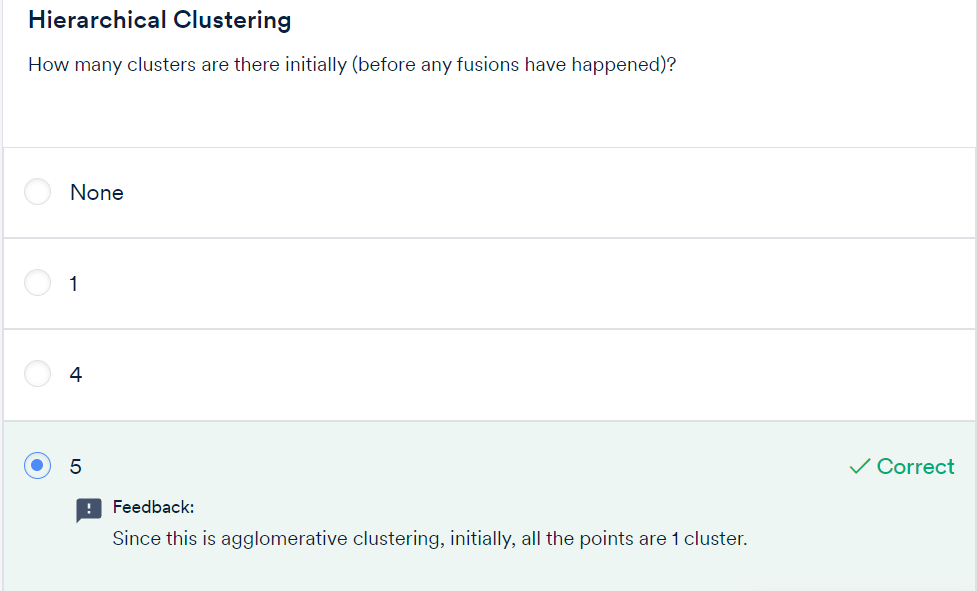
Determining the number of groups in a cluster analysis is often the primary goal. Typically, one looks for natural groupings defined by long stems. Here, by observation, you can identify that there are 3 major groupings: 3-6, 4-5-2-7 and 1-10-9-8.

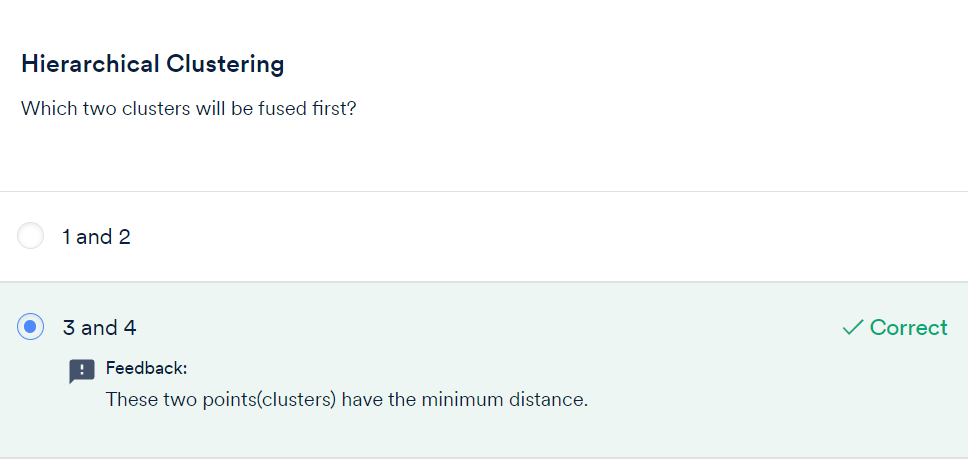
You also saw that hierarchical clustering can proceed in 2 ways — agglomerative and divisive. If you start with n distinct clusters and iteratively reach to a point where you have only 1 cluster in the end, it is called agglomerative clustering. On the other hand, if you start with 1 big cluster and subsequently keep on partitioning this cluster to reach n clusters, each containing 1 element, it is called divisive clustering.

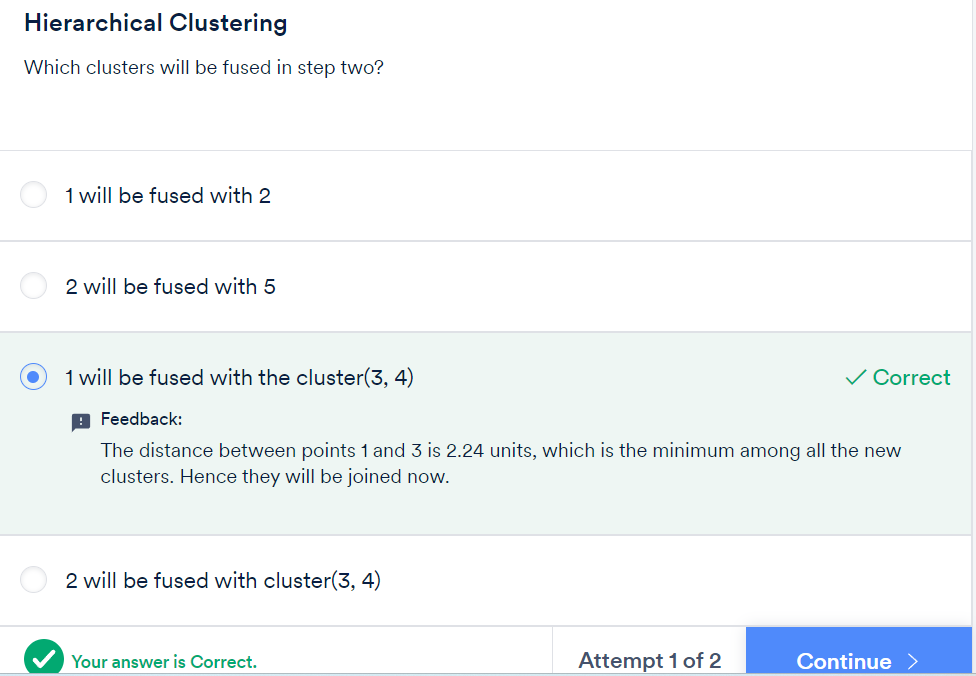


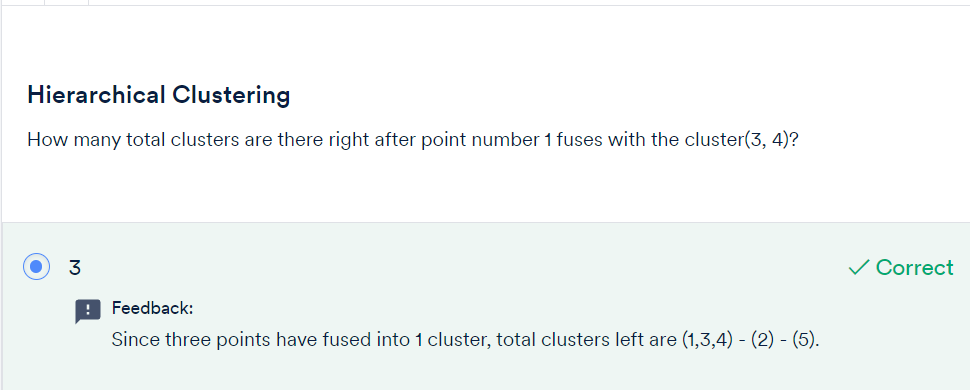


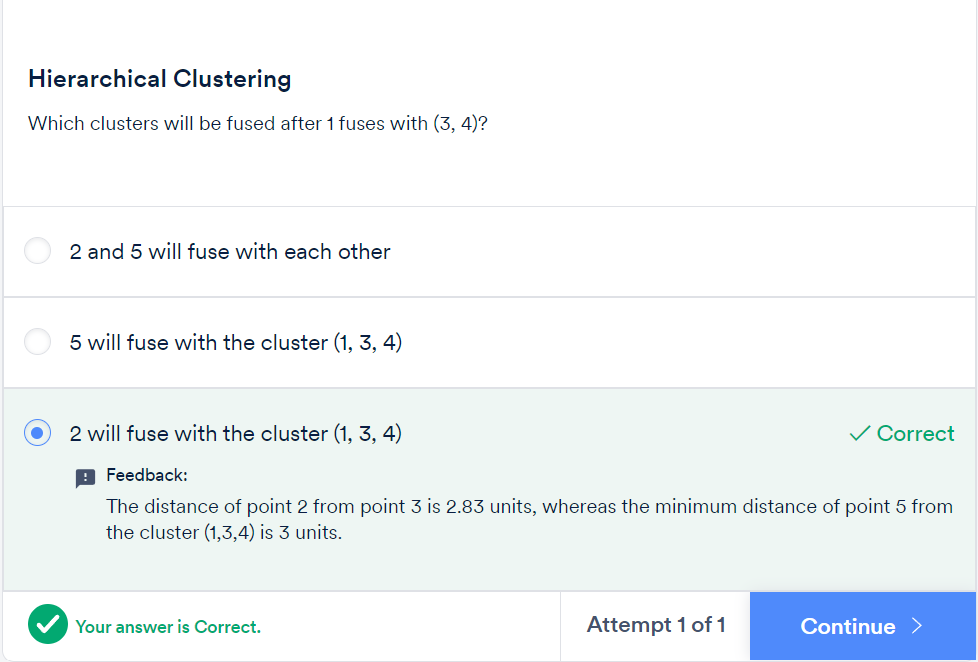


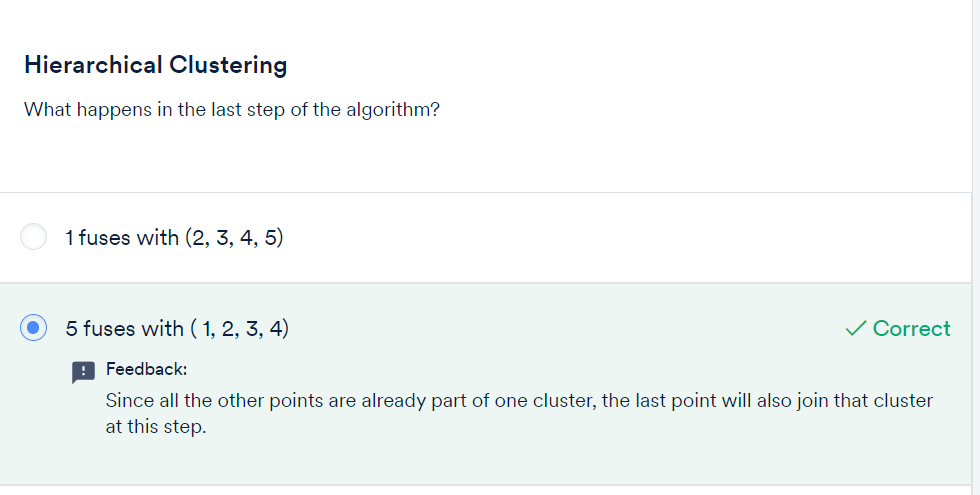










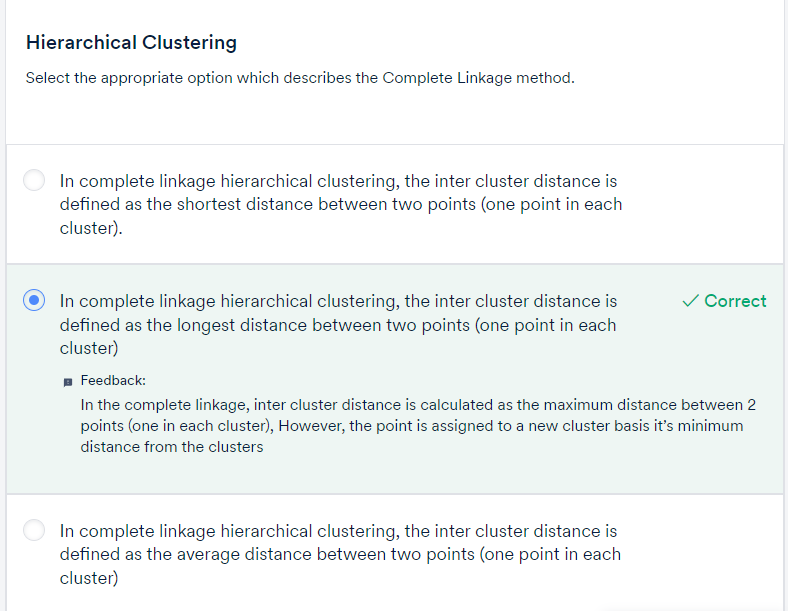


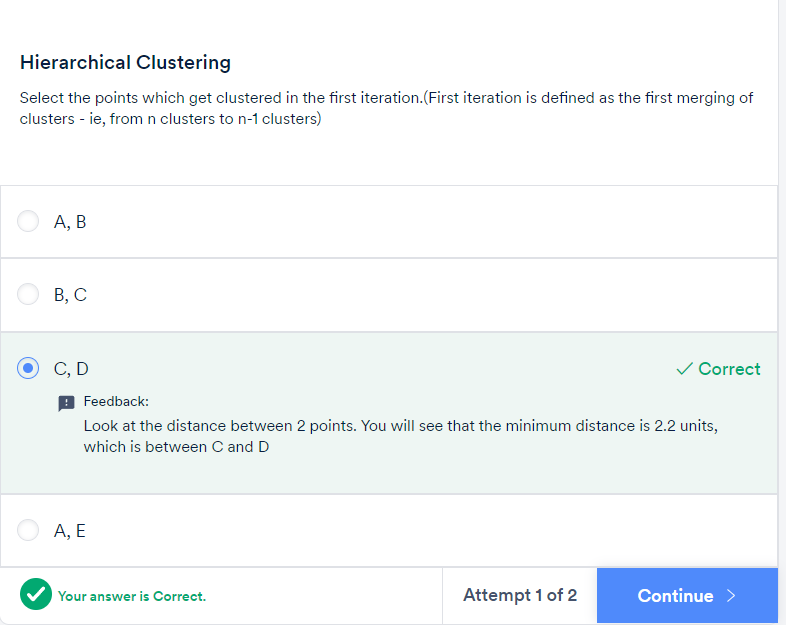
# Types of Linkages:

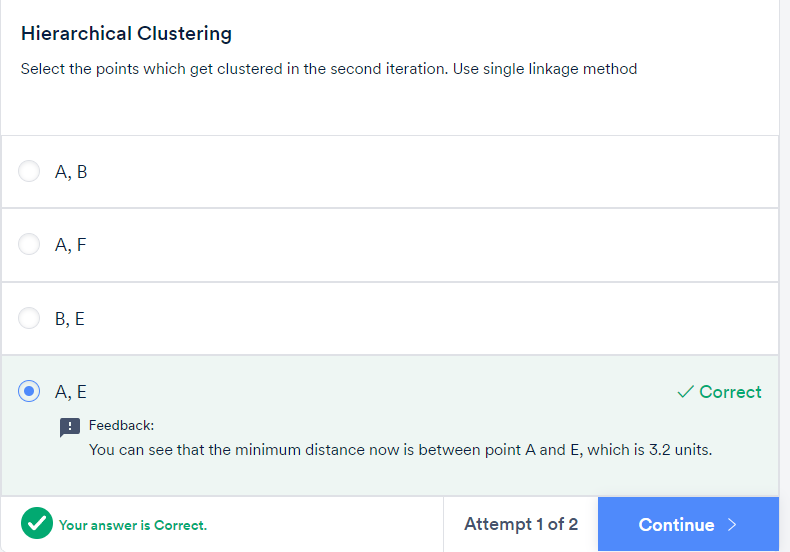
Let’s see once again the different types of linkages.

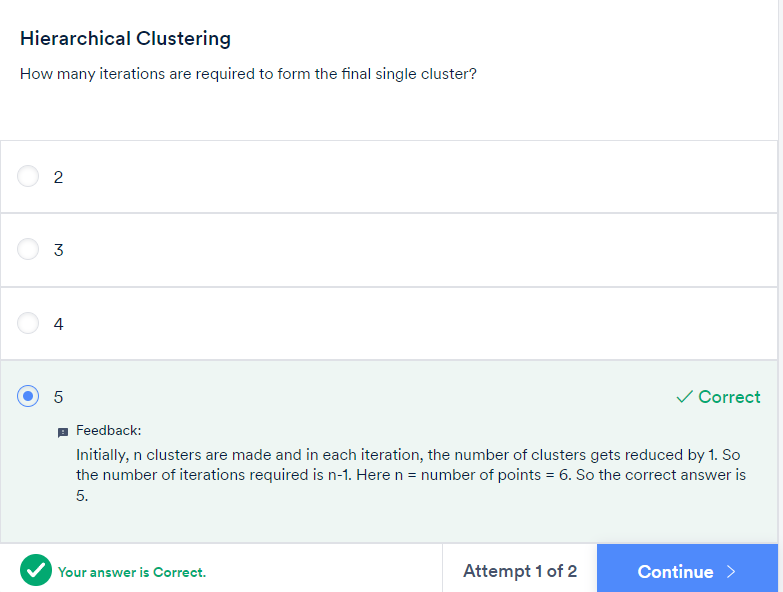
* **Single Linkage:**Here, the distance between 2 clusters is defined as the shortest distance between points in the two clusters
* **Complete Linkage:**Here, the distance between 2 clusters is defined as the maximum distance between any 2 points in the clusters
* **Average Linkage:**Here, the distance between 2 clusters is defined as the average distance between every point of one cluster to every other point of the other cluster.

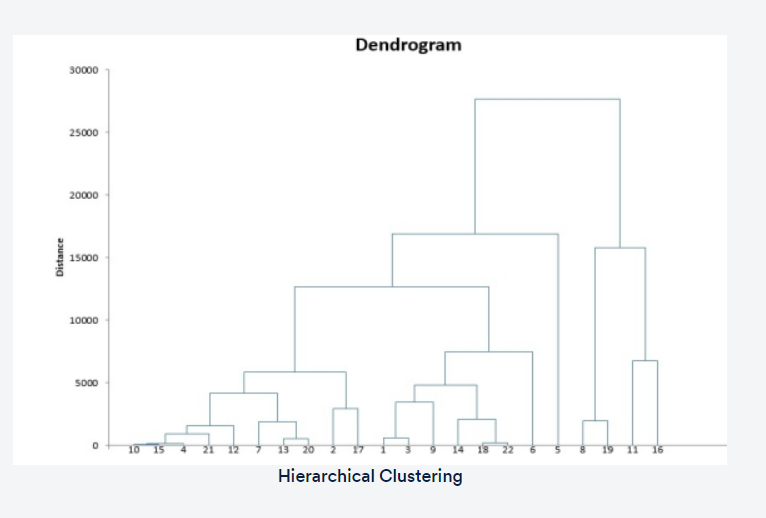
You have to decide what type of linkage should be used by looking at the data. One convenient way to decide is to look at how the dendrogram looks. Usually, single linkage type will produce dendrograms which are not structured properly, whereas complete or average linkage will produce clusters which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

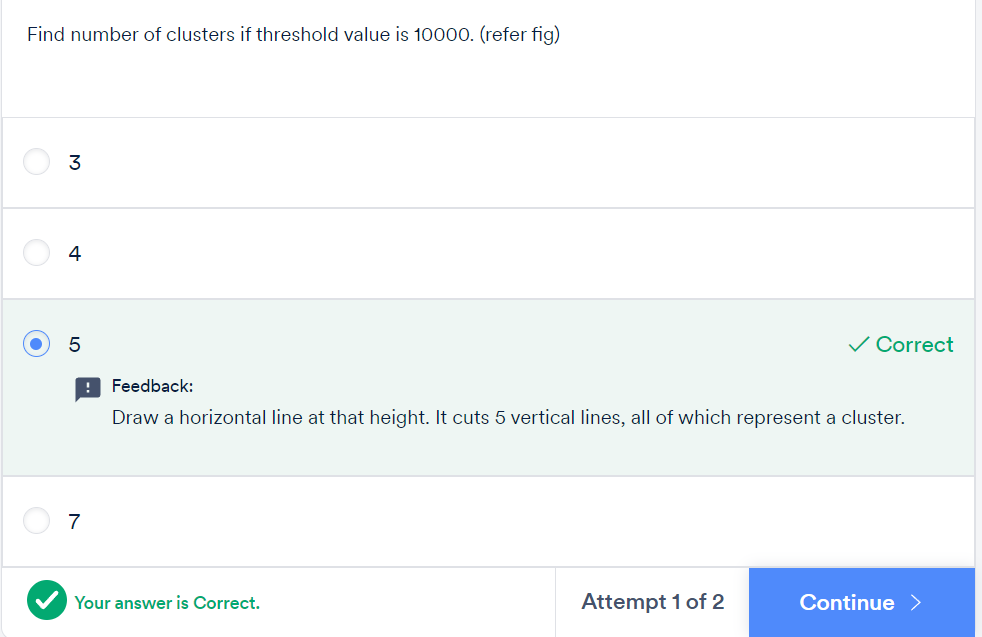


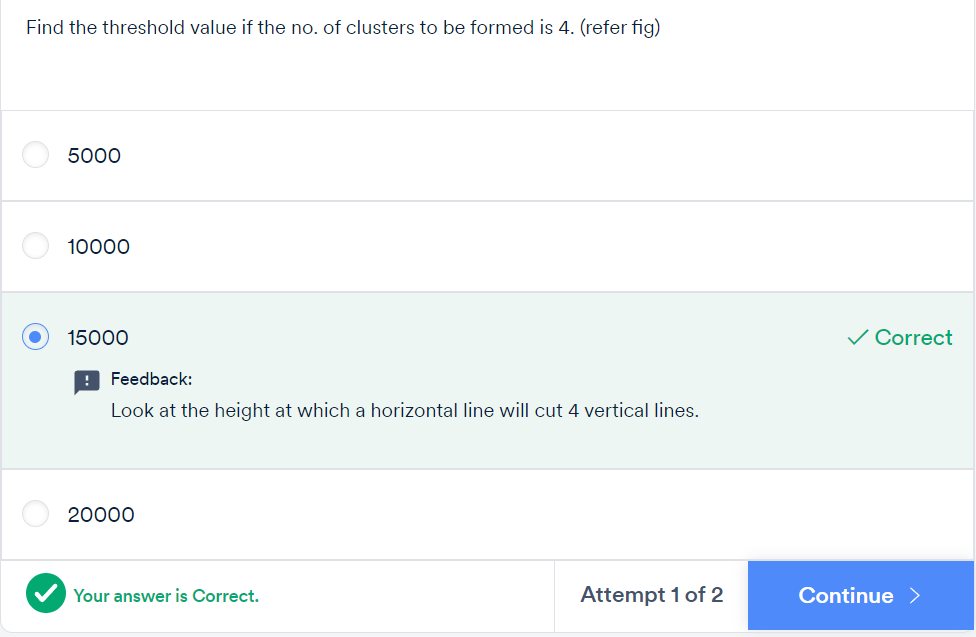












**For a better way , we can start with Hierarchical clustering. Get a range of K and then apply the K mean clustering**

