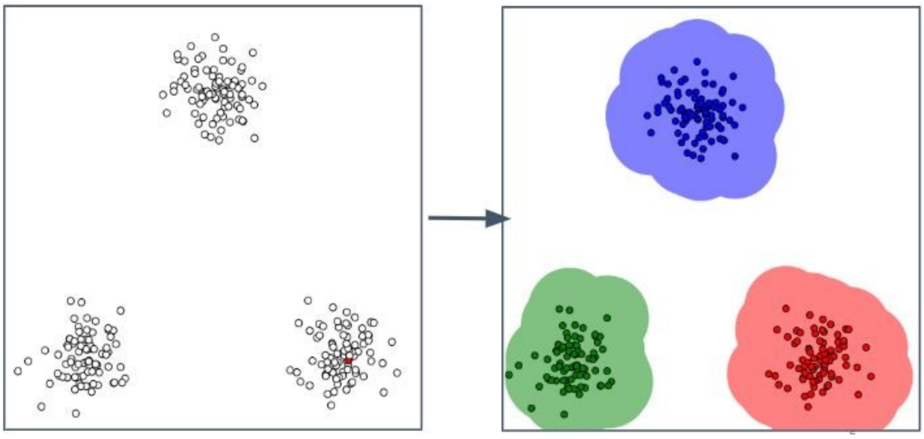
Summary

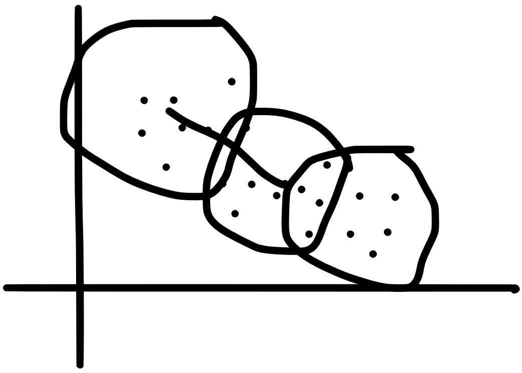
Introduction to Clustering and the K-means Algorithm

In this session, you learnt about clustering. It is an unsupervised learning technique wherein you try to find patterns based on similarities in data. You also learnt about the types of clustering methods and its various applications in practical life. You gained an understanding of the process of segmentation and learnt how it differs from clustering. Finally, you learnt about one of the most popular clustering algorithms known as the K-means algorithm.

Understanding Clustering

Clustering is the grouping of objects such that those belonging to the same cluster or group are similar. This similarity is a measure of how alike two data points are. This similarity measure is quantified by the distance measured between two points. Consider the scatter plot given below. Using a clustering algorithm, we have grouped the data points into three clusters and depicted them in red, blue and green.

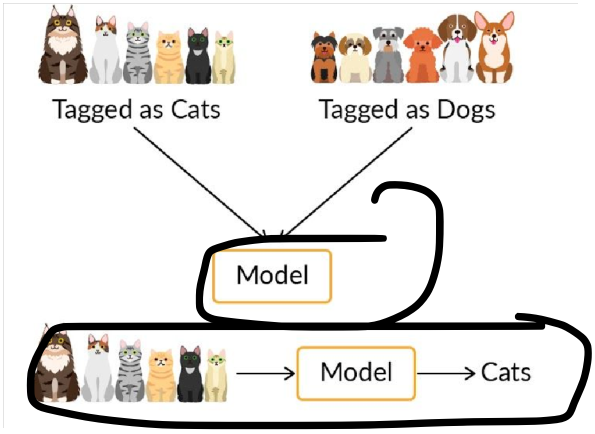
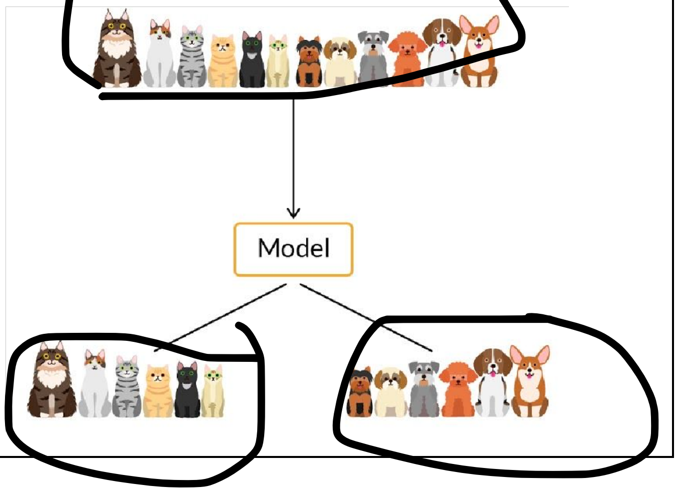




Classification vs clustering

Clustering and classification are both fundamental approaches in data mining, and students often tend to get confused between the two. The following table lists the differences between classification and clustering.

Classification is generally supervised (i.e., an objective or a dependent variable is given).	Clustering is usually unsupervised.
Classification is generally used for predictive purposes.	Clustering is generally used to discover new categories/groups within a given population of interest.

In classification, specific categories are explicitly given.	In clustering, new categories are defined based on the similarity between the instances. Instances that are similar to each other are grouped together.
Example: Differentiating between cats and dogs  	Example: Clustering animals based on similar features 

Applications of clustering

1. **E-commerce**: In this industry, clustering is used to identify different segments of customers within the customer base. The characteristics/profiles of these customer clusters are used to devise target (cluster)-specific marketing campaigns.

2. **Banking**: Data of fraudulent customers tends to act as outliers in the data set. These outliers can be easily detected through clustering algorithms, as they can identify the large dissimilarity between the outlier and the rest of the data. This can help in various use cases such as loyalty tiering and fraud.

3. **Market research**: Before launching a certain product into the market, market research is performed in order to understand customer needs and improve product features accordingly. Clustering is performed as part of market research to identify specific groups within a population who would be more likely to purchase the product.

4. **Telecom**: In this industry, clustering is used to identify network congestion within specific markets. It helps in estimating the capacity for network expansion.

5. **Marketing**: Cluster analysis can help in the field of marketing. It can also help in market segmentation and positioning and to identify test markets for new product development.

6. **Social media**: In the areas of social networking and social media, cluster analysis is used to identify similar communities within larger groups.

7. **Healthcare**: Cluster analysis has also been widely used in the field of biology and medical science such as human genetic clustering, sequencing into gene families, building groups of genes, and clustering organisms at species level.
8. Other applications
    a. **Document clustering**: Clustering is performed as part of natural language processing (NLP) to cluster similar types of documents or text together. One such use case is topic clustering, wherein you group pieces of information or text that share a similar topic, such as sports or politics.

    b. **Image clustering**: Image clustering is the process of clustering similar-looking images. Consider the example of running a clustering algorithm on a data set consisting of pictures of dogs and cats. Since the pictures of cats would be more similar to each other, they would be grouped in one cluster. Likewise, all the pictures of dogs would be grouped in another cluster.

Segmentation

Segmentation is a business problem, whereas clustering is an analytics technique. Customer segmentation for targeted marketing is one of the most significant applications of clustering algorithms. As a manager of an online store, you would want to group customers into different clusters so that you can create a customised marketing campaign for each group. You learnt that mainly three types of segmentation are used for customer segmentation, which are as follows:

- **Behavioural segmentation**: This segmentation is based on the actual patterns displayed by the consumer. In this module, you learnt about the following three major types of behavioural segmentation.

○ **RFM analysis:** The three attributes on which you may want to build the model, which are monetary, recency and frequency, are not part of the data set; so, you would need to build or derive these from the existing pool of features.

- **Recency**: It measures the recency of a customer visiting the store or making a purchase.
- **Frequency**: It measures the frequency of the transactions made by customers.
- **Monetary**: It measures the amount of money that a customer has spent on their purchases.

○ **RPI analysis**: The three attributes on which you may want to build the model are the following:

■ **Relationship**: Past interaction with the company
■ **Persona**: Type of person
■ **Intent**: Intention at the time of purchase

E-Comm: Amazon, Flipkart, Myntra

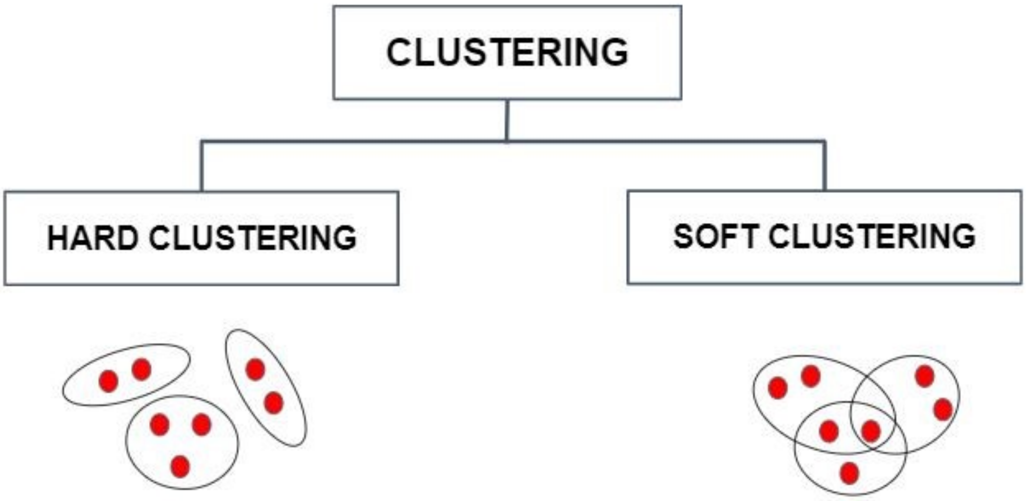
    - **CDJ analysis**: CJD stands for ‘**consumer decision journey** ’. It connects a customer’s life journey with a certain brand or product.
- **Attitudinal segmentation**: This segmentation is based on the beliefs or intents of people, which may not translate into a similar action.
- **Demographic segmentation**: This segmentation is based on a person’s profile and uses information such as age, gender, residential address, locality and income.

\

Types of Clustering

Clustering is of the following two types:

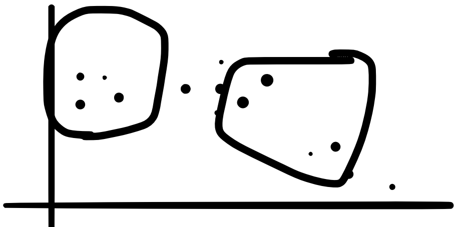
- **Hard clustering:** In a hard cluster, a data point would either belong to a cluster or would not belong to it at all. There is no in-between.
- **Soft clustering**: In a soft cluster, a data point could belong to more than one cluster simultaneously. A data point is associated with a cluster with a likelihood.



Types of clustering algorithms

Similar to the different types of clusters, there exist different types of clustering algorithms. You can differentiate between algorithms based on their clustering model. Some important techniques are as follows:

- **Partition clustering:** In this technique, a data set is divided into fixed sets of partitions/clusters. It is also known as a centroid-based cluster. The number of clusters depends on the number of cluster centroids defined by the user. A cluster centroid is iteratively modified by learning data to form final clusters.



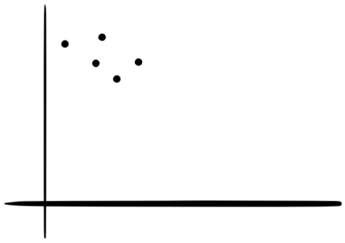
- **Hierarchical clustering:** In this technique, the user does not specify the number of clusters (as in the previous method). This type of clustering is known as connectivity-based clustering. It provides a hierarchy of clusters, as clusters merge together to form new ones. The output is represented in the form of a tree or a dendrogram.
- **Density-based clustering:** In this technique, clusters are formed by segregation of various density-based regions depending on the density in the data set. DBSCAN is one of the popular techniques that belongs to this type of clustering. This method works on any type of cluster shape.
- **Distribution model-based clustering:** In this technique, clusters are formed by assigning the observation to the cluster with the highest probability of belonging to that cluster. The most popular algorithm in this technique is the expectation-maximisation (EM) algorithm.
- **Fuzzy clustering:** This technique is part of soft clustering methods. In this type of clustering, a point can belong to multiple clusters. Each point is assigned a certain membership in each cluster, which indicates the degree to which the point belongs to the cluster. Fuzzy c-means is an example of fuzzy clustering.

Distance Measures

A clustering algorithm needs to find data points whose values are similar to each other; therefore, these points would then belong to the same cluster. The method in which any clustering algorithm does that is by utilising a ‘distance measure’. There are three major types of classical distance measures, which are as follows:

- **Euclidean distance measure**: The Euclidean distance is the distance between two points that can be measured using a ruler. In a two-dimensional space, the Euclidean distance between the two points (x1, y1) and (x2, y2) is given as follows:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



For higher dimensions, this distance can be generalised as follows:

d(p, q) = $$\sqrt{\sum (p_i - q_i)^2}$$

where p and q are observations, and i denotes a feature.

- **Manhattan distance measure**: The Manhattan distance is the distance between two points measured along axes at right angles. In a two-dimensional space, the Manhattan distance between the two points (x1, y1) and (x2, y2) is given as follows:

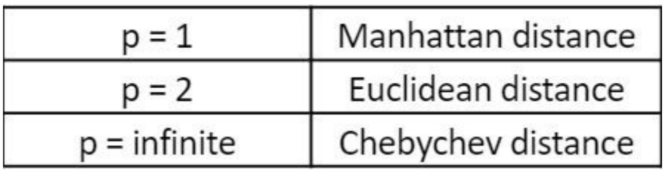
$$|x_2 - x_1| + |y_2 - y_1|$$

For higher dimensions, this distance can be generalised as follows:

d(p, q) = $\sum |p_i - q_i|$

- **Minkowski distance measure**:
  The Minkowski distance is a generalisation of both Euclidean and Manhattan distance measures. For n dimensions, the Minkowski distance between two observations, p and q, can be given as follows:
  $$(\sum |p_i - q_i|^p)^{1/p}$$

  The following table lists the distance measures for different p values:

p = 1	Manhattan distance
p = 2	Euclidean distance
p = infinite	Chebychev distance

  For p = 1 and p = 2, the Minkowski distance transforms into Manhattan and Euclidean distances, respectively.

  At p = ∞, the distance measure is transformed into the Chebyshev distance. This distance is also known as the maximum value distance. For two points, p and q, the Chebyshev distance is given as follows:
  $$d(p, q) = max(|p_i - q_i|)$$

Standardisation of data, that is, converting them into z-scores with mean 0 and standard deviation 1, is important for the following two reasons in the K-Means algorithm:

- Since you need to compute the Euclidean distance between data points, it is important to ensure that the attributes with a larger range of values do not outweigh the ones with a smaller range. Thus, scaling down all attributes to the same normal scale helps in this process.
- Different attributes will have measures in different units. Thus, standardisation helps in making the attributes unit-free and uniform.
- The formula for standardisation is given by:

$$x_{scaled} = \frac{x - mean}{sd}$$

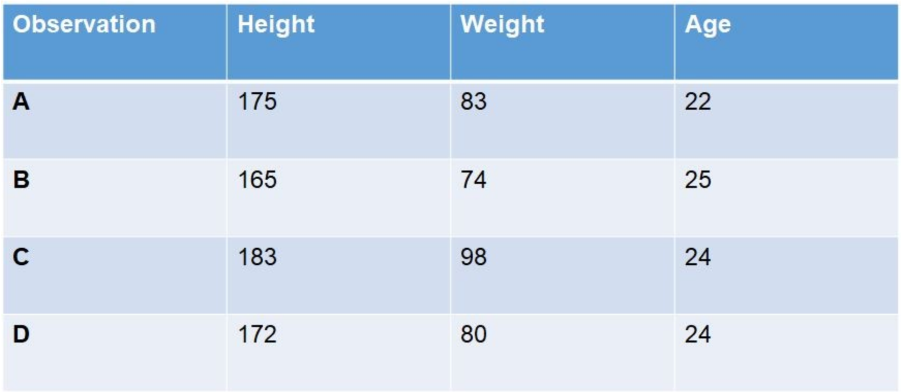
Hopkins Statistic

The Hopkins statistic is used to measure cluster tendency by measuring the probability of the given data being generated using uniform data distribution. If the value of the Hopkins statistic is close to 1, then it implies that the data set is clusterable.

K-means Algorithm

K-means is a centroid-based algorithm using which you can calculate the distances to assign a point to a cluster, and each cluster is associated with a centroid.

The centroid is calculated by computing the mean of each and every column/dimension that you have and then arranging them in order in the same way as shown in the table below.



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

.

For the data represented above:

1. Mean of height = ((175+165+183+172)/)/4 = 173.75
2. Mean of weight = ((83+74+98+80))/4 = 83.75
3. Mean of age = ((22+25+24+24))/4 =23.75

Thus, the centroid of the aforementioned group of observations is (173.75, 83.75 and 23.75).

To use the algorithm, you will first need to state the number of clusters ‘K' that will be present in your result. The steps in the algorithm are as follows:

- The algorithm first selects K objects randomly to act as initial cluster centres. These objects are called cluster centroids or means.
- Then, you assign the remaining objects to their closest centroids. The Euclidean distance between the cluster centroids and the objects determines their proximity.
- After you assign the objects to their respective centroids, the algorithm calculates the mean value of the clusters.
- After this re-computation, you recheck the observations to determine whether or not they might be closer to a different cluster. Then, you reassign the objects to centroids accordingly.
- Continue repeating these steps until the assigning clusters stop. This means that you stop repeating the iterations when the clusters that are formed in an iteration are the same as those in their previous iteration.

Ideally, if the assignment of points to cluster centroids in the last two consequent iterations are the same, then the algorithm is said to have converged. At this point, you can conclude that the K-means algorithm has come to a stop, as the points remain in the same cluster, and the clusters at hand are the final optimal clusters.

Cost Function

The cost function for the K-Means algorithm is mathematically given by the sum of squared errors (SSE), which is given as follows:

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

The Euclidean distance is calculated from each data point to its nearest centroid. These distances are squared and summed to obtain the SSE. This type of distance is also known as the intracluster or within-cluster distance. Essentially, the cost function tries to minimise the SSE (i.e., the intra-cluster sum of distances), and by minimising the cost function, the K-Means algorithm aims to form clusters of data points that are similar to each other.

Mathematical representation of the assignment and optimisation steps

In the assignment step, you assign every data point to K clusters. The algorithm goes through each of the data points, and depending on the cluster that is closer, it assigns the data points to one of the closest cluster centroids. The equation for the assignment step can be given as follows:

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

After the assignment step is complete, the next one is the optimisation step wherein the algorithm computes the new cluster centroids. In the optimisation step, the algorithm calculates the average of all the points in a cluster and moves the centroid to that average location.

Once you have all the points for each cluster, you can directly compute the new centroid values using the equation given below:

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

Optimal K

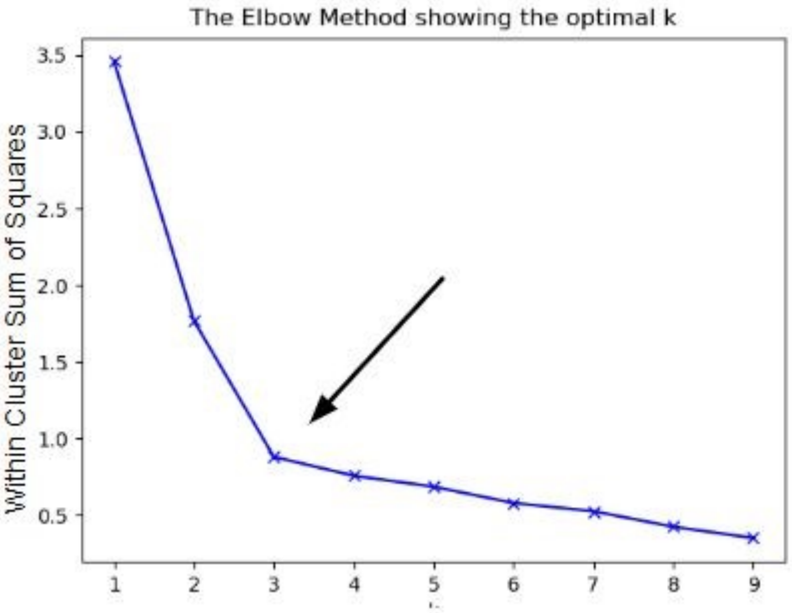
Determining the value of K is a fundamental problem in K-means. The two methods that can help in identifying the correct value of K are as follows:

- Elbow method
- Silhouette method

Elbow method

The steps involved in using the elbow method can be broadly broken down into the following three steps:

- Calculate the within-cluster sum of squares (WCSS) for different values of K. The WCSS measures the squared average distance between all the points that are within a cluster and the cluster centroid.
- Create a plot with the WCSS on the y axis and the number of clusters on the x axis.
- Choose the value of K at which the curve begins to flatten. This point is also known as the elbow point. Essentially, the elbow point represents the value of K at which the addition of clusters does not improve the WCSS.



Silhouette method

The silhouette method measures the extent of similarity of the point with its own cluster compared with that with the nearest cluster. For a point xi, the silhouette value is given by

$$S(i) = \frac{b(i) - a(i)}{max(b(i), a(i))}$$

where, b(i) = Average distance of a point to the points in the next closest cluster a(i) = Average distance of a point to other points within the cluster

The range of this measure is between -1 and 1. The silhouette coefficient is the average silhouette value of all the observations.

Silhouette Coefficient	Conclusion
Close to 1	Properly clustered
Close to 0	Overlapping clusters
Close to -1	Assigned to wrong clusters

Plot the curve of the average silhouette values against the number of clusters, K. The location of the highest value of silhouette coefficient is considered as the appropriate number of clusters

Choosing the metrics

Ideally, both methods should be used together. The elbow method is a decision rule, whereas the silhouette method validates the clustering algorithm. When both these methods are used together, you can be more confident with the final output.

Pros and Cons of the K-means Algorithm

The pros of this algorithm are as follows:

1) Compared with all the clustering algorithms, the K-means algorithm is considered as one of the **simplest algorithms to implement**.

2) In the previous session, you learnt that one of the important requirements of clustering algorithms is **scalability**. The K-means algorithm can be used with large and small data sets.

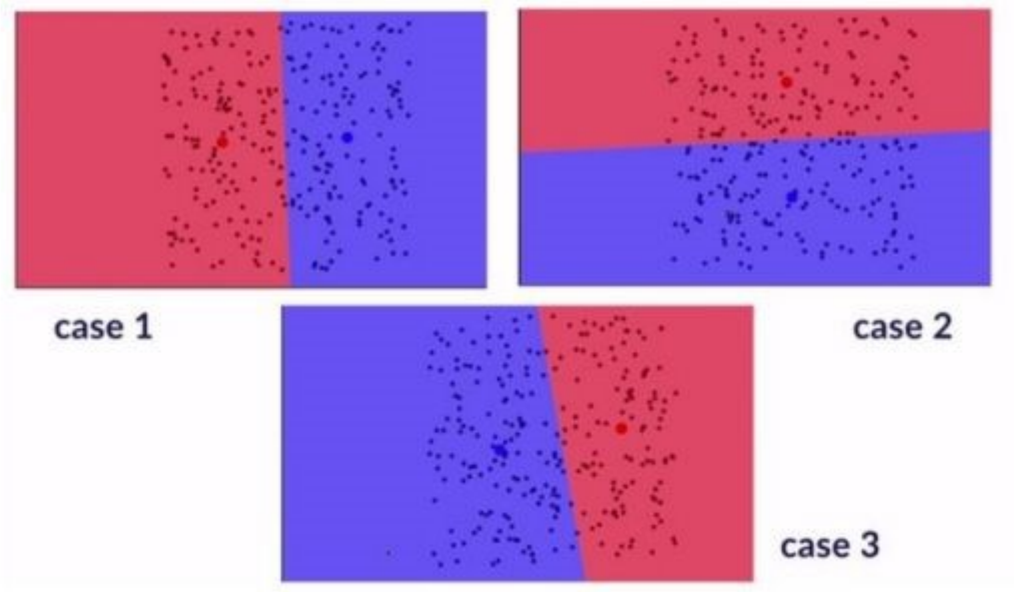
3) The K-means algorithm **guarantees convergence** after a certain number of iterations. At the point of convergence, the cost function has reached its minima, and the centroids will not be updated with anymore iterations.

The cons of this algorithm are as follows:

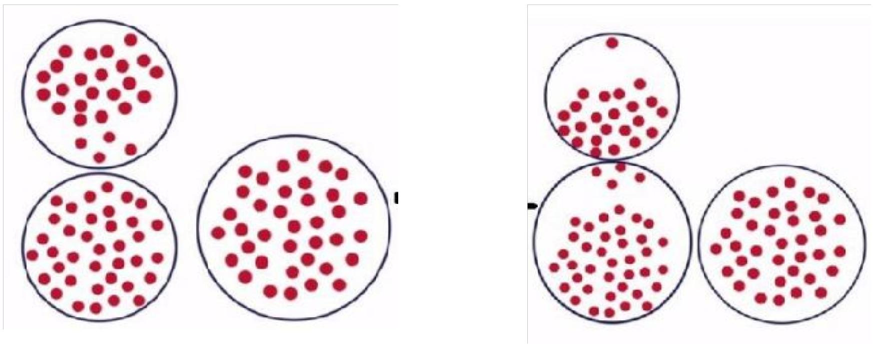
1) The value of K needs to be **chosen manually** .

2) K-means works best when clusters are spherical. It does a **poor job for clusters of complicated shapes**.

3) The K-Means algorithm is highly **dependent on the initial value of the centroids** . Three cases with different sets of initial cluster centres are shown below. In these, we obtained three different clusters at the end.



4) **Impact of outliers:** Since the K-Means algorithm tries to allocate each of the data points to one of the clusters, outliers have a serious impact on the performance of the algorithm and prevent optimal clustering.



5) **Categorical data:** The K-Means algorithm cannot be used when dealing with categorical data, as the concept of distance for categorical data does not make much sense. So, instead of the K-Means algorithm, you need to use different algorithms. In the next segment, you will learn about one such algorithm, the K-Modes algorithm.

6) **Scaling with number of dimensions:** As the number of dimensions increases, distance-based measures such as the Euclidean distance measure converge to a constant value between any given data points. Hence, for large dimensions, it is advisable to reduce the dimensionality using algorithms such as PCA.

K-means++

K-means++ is only 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. Apart from initialisation, both the algorithms work in the same way.

The steps involved in the K-Means++ algorithm are as follows:

1. You choose a centre as one of the data points at random.
2. For each data point Xi, you compute the distance between Xi and the nearest centre that was already chosen.
3. Then, you choose the next cluster centre using the weighted probability distribution where a point X is chosen with a probability proportional to $d(X)^2$.
4. Repeat steps 2 and 3 until K centres have been chosen.

Summary

Hierarchical Clustering and Case Study

In this segment, you learnt about hierarchical clustering algorithms. In hierarchical clustering, instead of pre-defining the number of clusters, you first need to visually describe the similarity or dissimilarity between different data points and then decide the appropriate number of clusters based on these similarities or dissimilarities. Then, you also learnt about the types of linkages, wherein you gained an understanding of the following three different types of linkages: single, complete and average linkages. Then, you learnt about the Bisecting K-means algorithm that is popularly used for clustering big data. Finally, you went through a case study on a PUBG data set to understand the different strategies used by gamers while playing PUBG.

Hierarchical Clustering Algorithm

Hierarchical clustering is an unsupervised clustering algorithm. It creates clusters based on the similarity between observations. It generally falls in the two categories that are given below:

1) Hierarchical agglomerative clustering
2) Hierarchical divisive clustering

In this segment, you learnt about the hierarchical agglomerative clustering (HAC) algorithm. The steps involved in this clustering algorithm are as follows:

- Calculate the distance matrix between all the points in a data set.
- Treat each point as a separate cluster. If you have n data points, then you will have n clusters at the start.
- Merge two clusters based on the lowest distance on the distance matrix. The total cluster at the end of this step is n-1.
- Update the distance matrix.
- Now, continue repeating the above two points until the number of clusters becomes 1.
- Repeat these steps until there are no more clusters to join.

Building the Dendrogram

The result of cluster analysis is shown by a dendrogram, which starts with all the data points as separate clusters and indicates the level of dissimilarity at which any two clusters were joined. Let's consider this example: take 10 points and try to apply a hierarchical clustering algorithm over them.

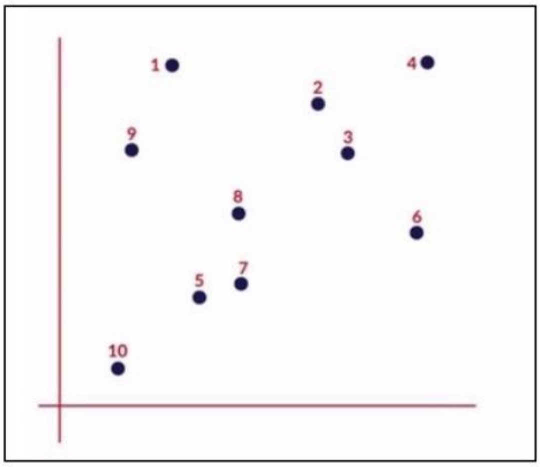


Fig 1: 10 random points

Initially, treat each of these points as individual clusters. Thus, you will begin with 10 different clusters.

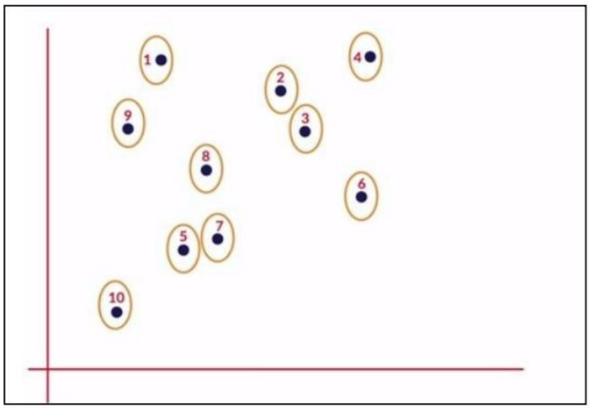


Fig 2: 10 initial clusters

Now, calculate the distance matrix for the 10 clusters, that is, the distance of each cluster from every other cluster. Then, combine the two clusters that have the minimum distance between them. In this case, points 5 and 7 were the closest cluster to each other. Thus, they would be merged first. Correspondingly, they appear at the lowest level in the dendrogram.

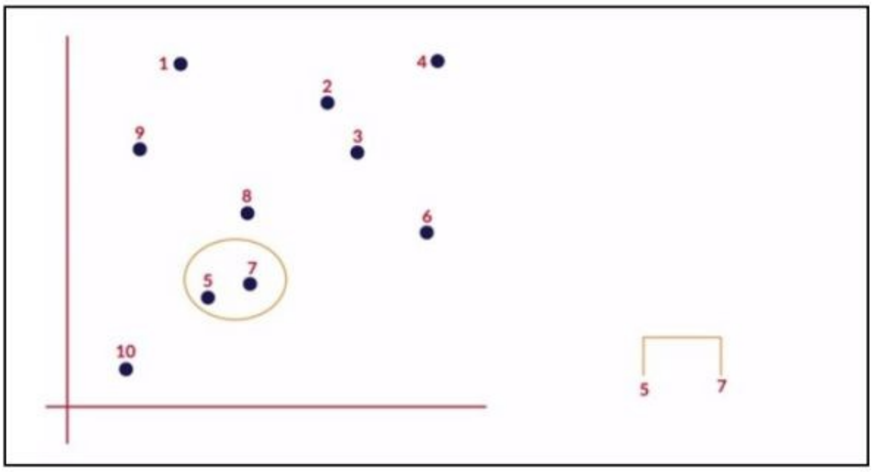


Fig 3: Merging of point 5 & 7 to form a single cluster

You are now left with only nine clusters. Eight of them have a single element, whereas one of them has two elements, which are 5 and 7. Calculate the distance of each cluster from every other cluster again. However, the problem here is the measurement of the distance between a cluster having two points and a cluster having a single point. It is here that the concept of linkage gains importance. Linkage is the measure of dissimilarity or similarity between the clusters having multiple observations.

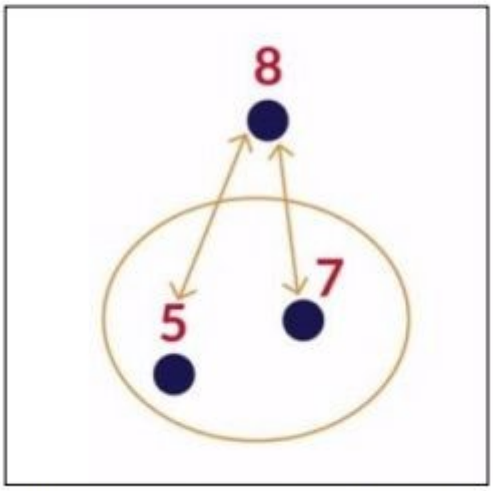


Fig 4: Calculating dissimilarity measure between 2 clusters .

Here, calculate the distance between the points 5 and 8 and then between the points 7 and 8; the minimum of these two distances is taken as the distance between the two clusters. Thus, in the next iteration, you will obtain eight clusters.

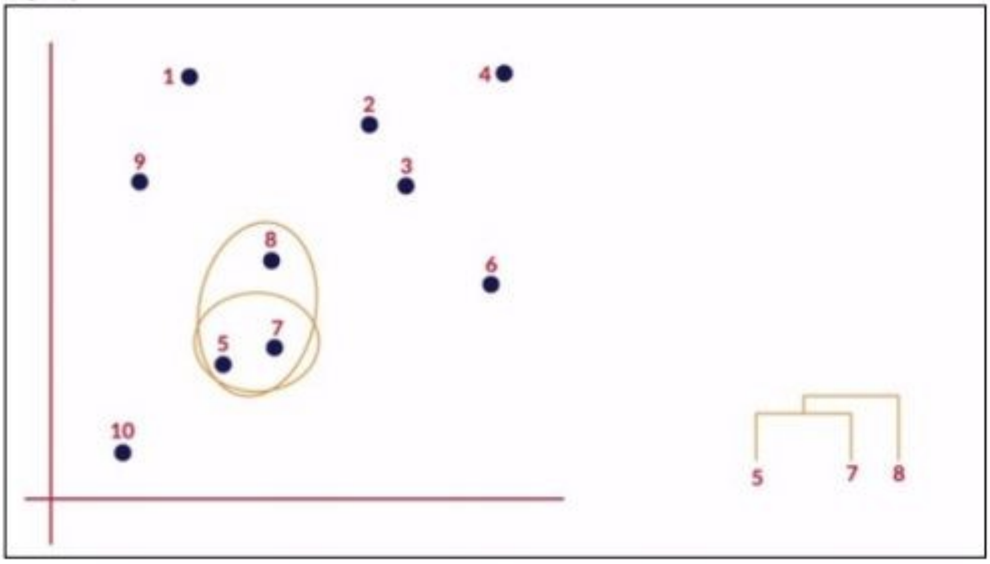
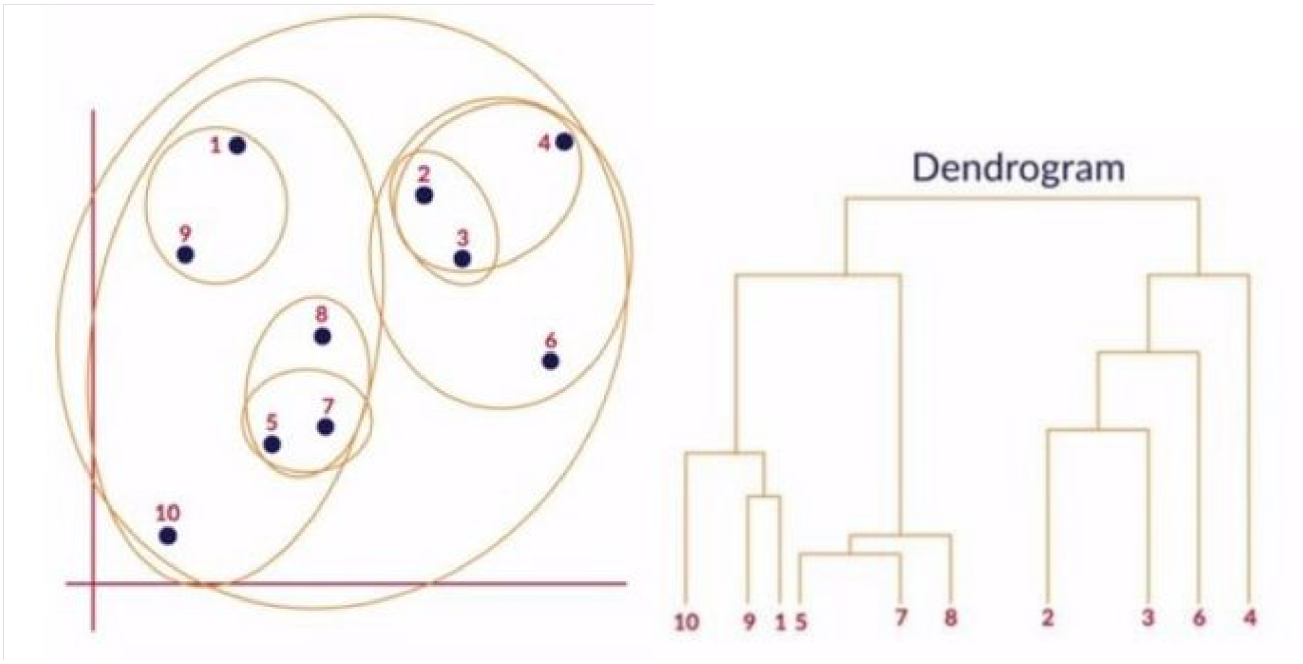
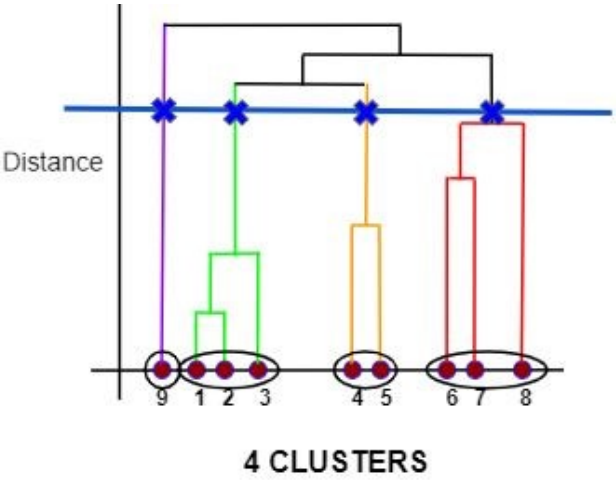


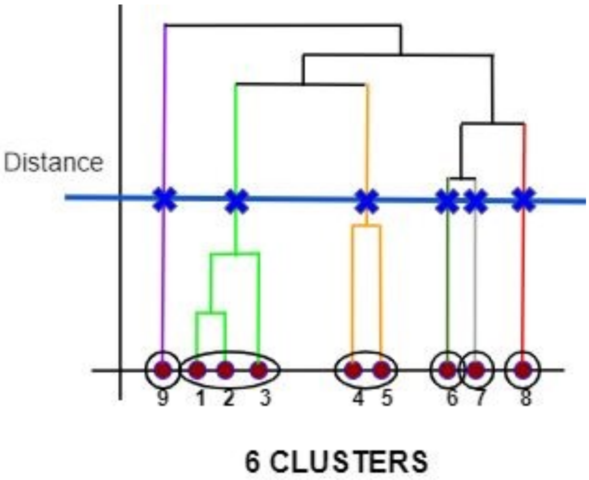
Fig 5: After iteration 2, we have 8 clusters

These iterations continue until you arrive at one giant cluster.



The y axis of the dendrogram is the measure of the dissimilarity or distance at which clusters join. In the image given below, you will notice that cutting the dendrogram at two points has resulted in four and six clusters.



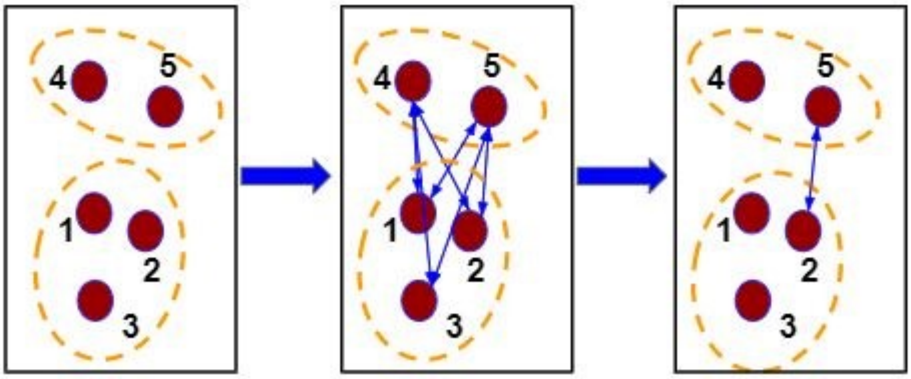


Types of Linkages

In the earlier example, you took the minimum of all the pairwise distances between the data points as the representative of the distance between two clusters. This measure of distance is called single linkage. Apart from the minimum, you can use other methods to compute the distance between the clusters. Let's consider the following common types of linkages:

a) Single linkage
b) Complete linkage
c) Average linkage

Single linkage: Here, the distance between two clusters is defined as the shortest distance between points in the two clusters.



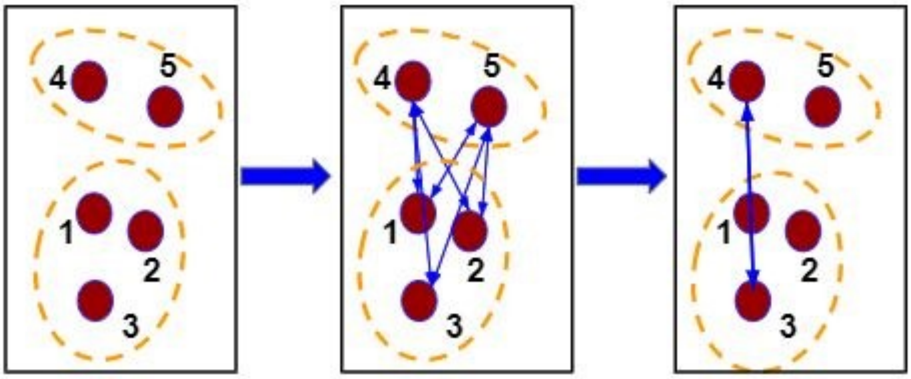
The **advantage** of this is as follows:

- It is highly effective in handling non-elliptical shapes.

The **disadvantages** of this type of linkage are as follows:

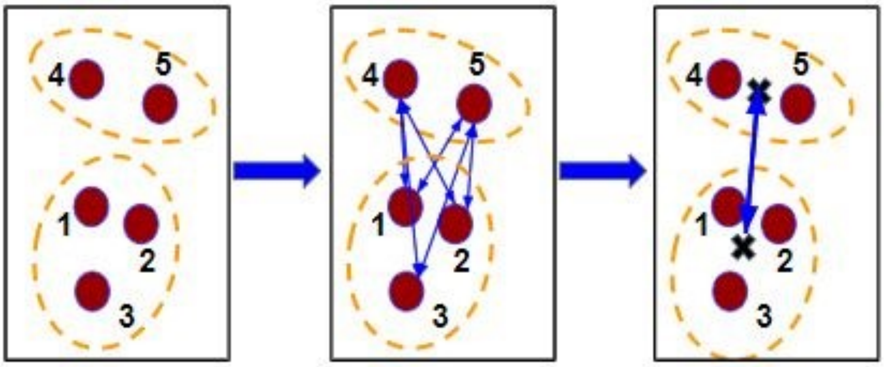
- **Chaining effect**: Since the distance between two clusters (or a cluster and a point) is recognised by the minimum distance between them, they are one of the first clusters to be merged when climbing up the ladder and cannot be distinguished after the dendrogram is created. This leads to a very high tendency of merging clusters that are close to each other, resulting in a chaining effect.
- **Sensitive to noise and outliers**: Adding a single outlier can drastically change the cluster sizes.

Complete linkage: Here, the distance between two clusters is defined as the maximum distance between any two points in the clusters.



- The **advantage** of this is as follows:
  a) Complete linkage is less susceptible to noise and outliers.
- **The drawbacks** of this are as follows:
  a) Linkages of this type tend to break large clusters.
  b) Complete linkage is biased towards global clusters.

Average linkage: Here, the distance between two clusters is defined as the average distance between every point of one cluster and every other point of another cluster.



- The **advantage** of this is as follows: It is less susceptible to noise and outliers.
- The **drawback** of this is as follows: It is biased towards globular clusters.

Advantages and Disadvantages of Hierarchical Clustering

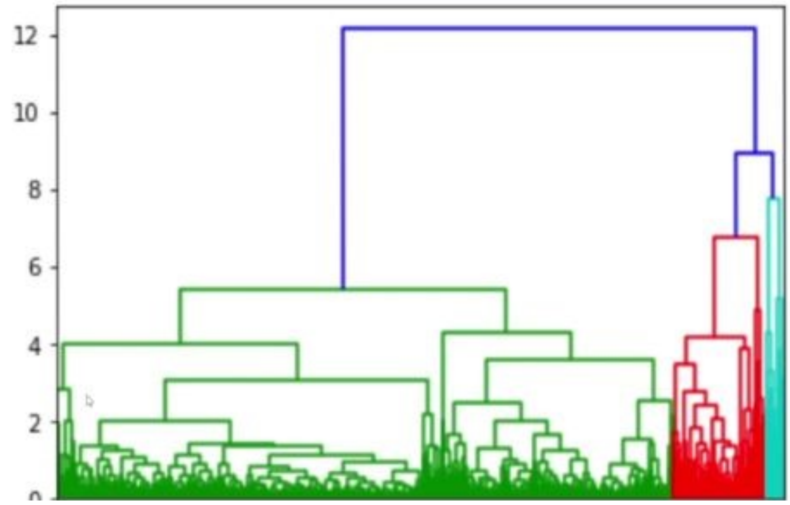
In this segment, you learnt about the advantages and disadvantages of hierarchical clustering. They aren listed below:

Its advantages are as follows:

- Unlike the K-means algorithm, you do not need to specify the number of clusters at the start of the algorithm.
- Compared with other clustering algorithms, the hierarchical clustering algorithms are relatively simpler to implement.
- Dendrograms are useful for visualising the data and the clusters formed after cutting the dendrogram.

Its disadvantages are as follows:

- Errors made in the early stages of agglomerative clustering algorithms cannot be recovered.
- Time complexity can result in long computation times. This is mostly because of the creation of the distance matrix. Consider using a data set with 1 lakh points. In order to create the distance matrix for these 1 lakh points, the system would have to compute the pairwise distance from each point to the remaining 99, 999 points. Similarly, after completing the first iteration, the distance matrix would have to be updated, which would require an extremely high computing power.
- For large data sets, it is difficult to use dendrograms. The following dendrogram is obtained from a data set of merely 700 points. You can imagine the complexity of the dendrogram if the data set is scaled to approximately a million points.



Bisecting K-means

A major disadvantage of the HAC algorithms is their large time complexity. This time complexity arises owing to the creation and updation of the distance matrix. Though the Bisecting K-means algorithm is a type of divisive clustering, it avoids the formation of a distance matrix. The run time of the Bisecting K-means is attractive compared with that of the agglomerative hierarchical clustering techniques, making it more suitable for big data processing.

The Spark ML library has an inbuilt implementation of the Bisecting K-means algorithm. To build a BisectingKMeans model, create an object of the BisectingKmeans class first and then fit the training data to generate your model.

The steps involved in the bisecting K-means algorithm are as follows:

- Set all data points to a single cluster.
- Using K = 2, create two subclusters.
- Measure the intra-cluster distance for both the clusters.
- The clusters with the highest intra-cluster distance are chosen for breaking next.
- The chosen cluster is further broken into two sub clusters using K-means.
- These steps are repeated until all the observations are individual clusters.

Why is the Bisecting k-means algorithm more efficient than the K-means algorithm for large data sets?

The computation for the K-means algorithm involves every data point of the data set and k centroids. On the other hand, in each Bisecting step in the Bisecting k-means algorithm, only the data points of one cluster and two centroids are involved in the computation. Thus, the computation time is reduced.

Python Commands for Clustering

You should remember the following important commands that are used to cluster data:

Scaling/standardising

```
standard_scaler = StandardScaler()
```

K Means clustering

```
model_clus = KMeans(n_clusters = num_clusters, max_iter=_)
```

Hierarchical clustering

```
mergings = linkage(X, method = "single/complete/average", metric='euclidean') dendrogram(mergings)
```

Cutting the cluster

```
clusterCut = pd.Series(cut_tree(mergings, n_clusters = num_clusters).reshape(-1,))
```

PySpark clustering

```
from pyspark.ml.clustering import KMeans from
pyspark.ml.evaluation import ClusteringEvaluator

# Loads data.
dataset = spark.read.format("libsvm").load("data/mllib/sample_kmeans_data.txt")

# Trains a k-means model. kmeans =
KMeans().setK(2).setSeed(1) model =
kmeans.fit(dataset)

# Make predictions
predictions = model.transform(dataset)

# Evaluate clustering by computing Silhouette score evaluator
= ClusteringEvaluator()
```