Session – Summary

Clustering – K-Means algorithm

In the previous modules, you saw various supervised machine learning algorithms. Supervised learning is a type of machine learning algorithm that uses a known dataset to preform predictions. This dataset (referred to as the training dataset) includes both response values and input data. From this, the supervised learning algorithm seeks to build a model that can predict the response values for a new dataset.

If you are training your machine learning task only with a set of inputs, it is called unsupervised learning, which will be able to find the structure or relationships between different inputs. The most important unsupervised learning technique is clustering, which creates different groups or clusters of the given set of inputs and is also able to put any new input in the appropriate cluster. While carrying out clustering, the basic objective is to group the input points in such a way as to **maximise the inter-cluster distance and minimise the intra-cluster variance**.

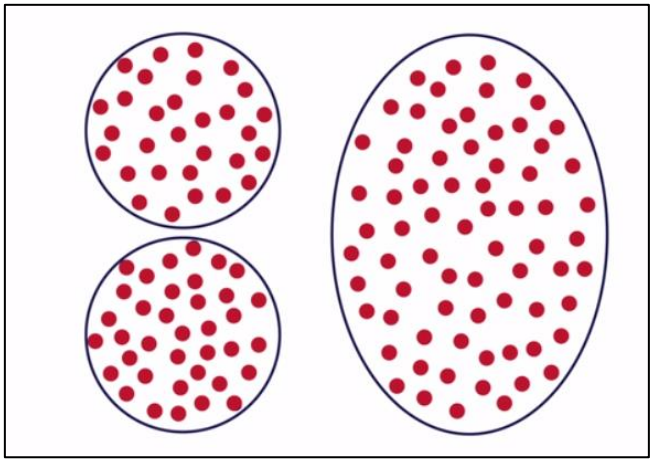

Fig 1: Objective of clustering is to **maximise the inter-cluster distance and minimise the intra-cluster variance**

The two most important methods of clustering are the K-Means algorithm & the Hierarchical clustering algorithm.

K-Means Algorithm

K-Means algorithm is the process of dividing the N data points into K groups or clusters. Here the steps of the algorithm are:

1. Start by choosing K random points the initial cluster centres.
2. Assign each data point to their nearest cluster centre. The most common way of measuring the distance between the points is the Euclidean distance.
3. For each cluster, compute the new cluster centre which will be the mean of all cluster members.
4. Now re-assign all the data points to the diffrent clusters by taking into account the new cluster centres.
5. Keep iterating through the step 3 & 4 until there are no further changes possible.

At this point, you arrive at the optimal clusters.

Let's apply the K-Means algorithm on a set of 10 points, which we want to divide into 2 clusters. Thus the value of K here is 2.

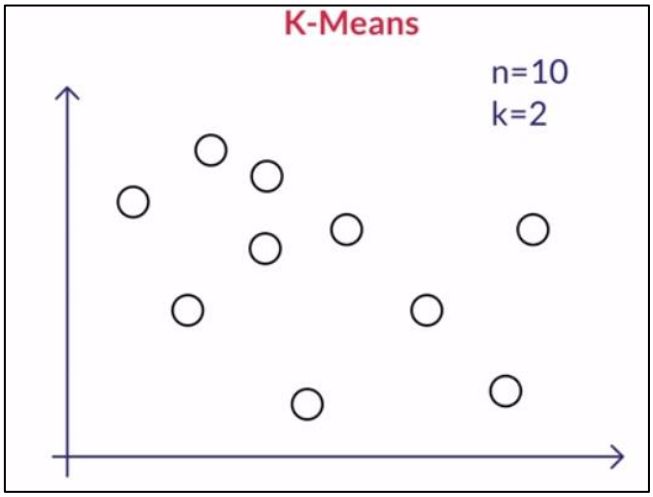

Fig 2: A set of 10 points to be divided into 2 clusters

We begin with choosing 2 random points as the 2 cluster centres.

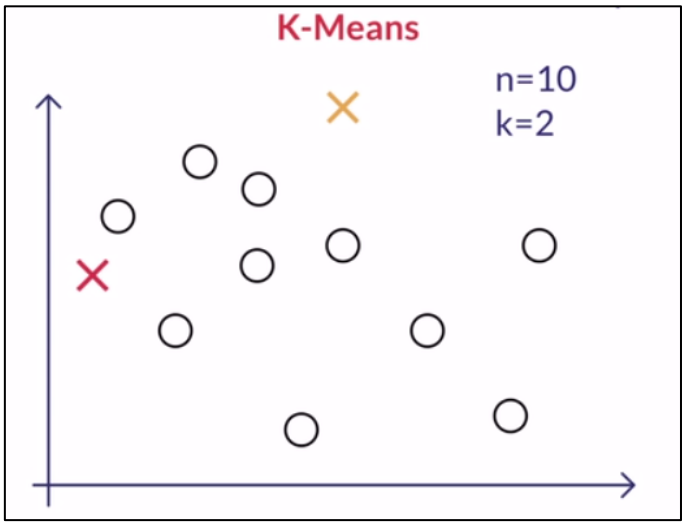

Fig 3: Choosing K random initial cluster centres

We then assign each of the data points to their nearest cluster centres based on the Euclidean distance. This way all the points are divided among the K clusters.

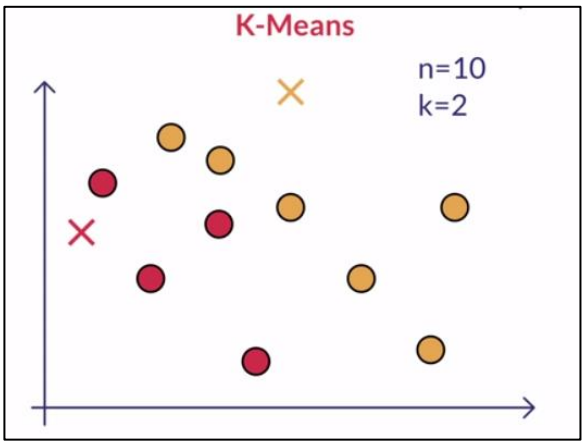

Fig 4: Assigning each data point to their nearest cluster centre

Now we update the position of each of the cluster centres to reflect the mean of each cluster.

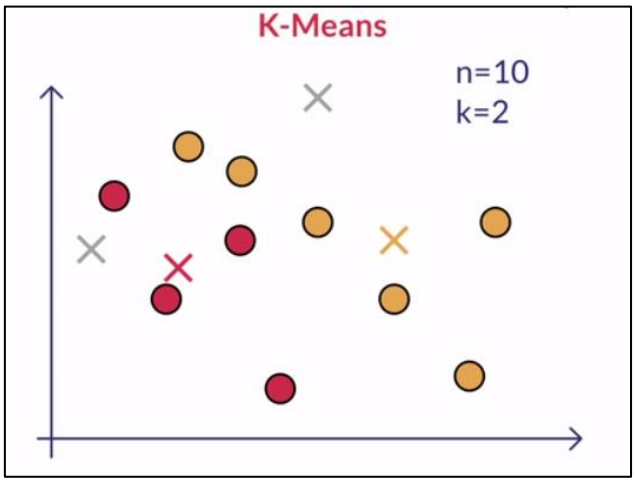

Fig 5: Updating the cluster centres

This process continues iteratively till the clusters converge; that is, there are no more changes possible in the position of the cluster centres. At this point, we achieve the two optimal clusters.

Practical considerations in K-Means algorithm

Some of the points to be considered while implementing K-Means algorithm are:

1.) **The choice of initial cluster centre has an impact on the final cluster composition.**

You saw the impact of the initial cluster centres through the visualisation tool. In the 3 cases with a different set of initial cluster centres, we obtained 3 different clusters at the end.

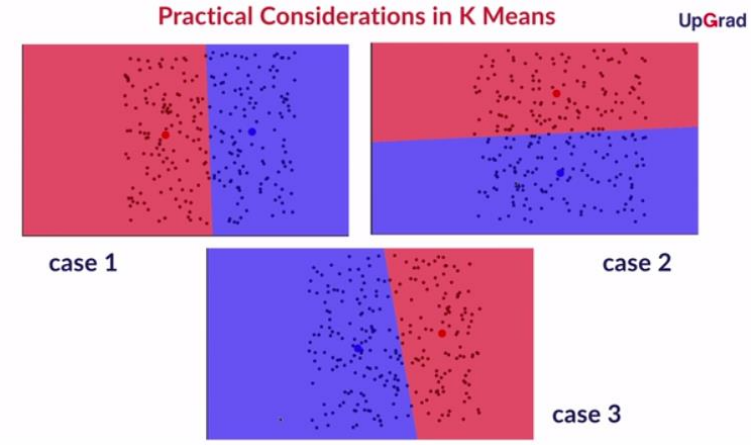

Fig 6: Impact of initial cluster centres on the final result

2.) **Choosing the number of clusters K in advance**

There are a number of pointers that can help us decide the K for our K-means algorithm:-

- The business problem/understanding/constraints
- Mathematically, we aim to maximise the inter-cluster distance and minimise the intra-cluster variance. This can be achieved in R using the R_sq statistic. Here the output of the kmeans() function is utilised. R_sq measures the fraction of the total sum of the squared distance between the data points that can be explained by the sum of squared distance between the clusters. The higher this fraction, the better the clustering. Thus, we plot the curve for the R_sq vs the number of clusters and use the elbow method to find the optimal range of K. You will learn more about it when you implement the K-Means algorithm in R.
- Also, the division of points among different clusters should make business sense and should be feasible and viable.

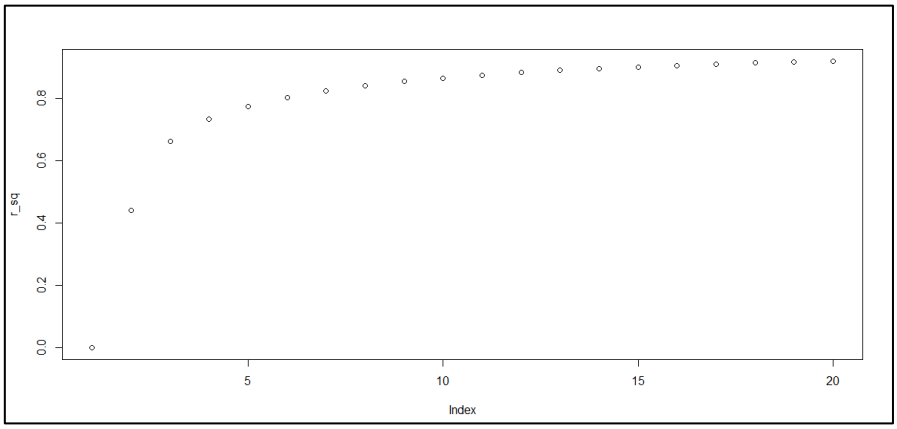

Fig 7: R_sq vs the K (number of clusters)

3.) **Impact of outliers**

Since, the K-Means algorithm tries to allocate each of the data point to one of the clusters, outliers have serious impact on the performance of the algorithm and prevent optimal clustering.

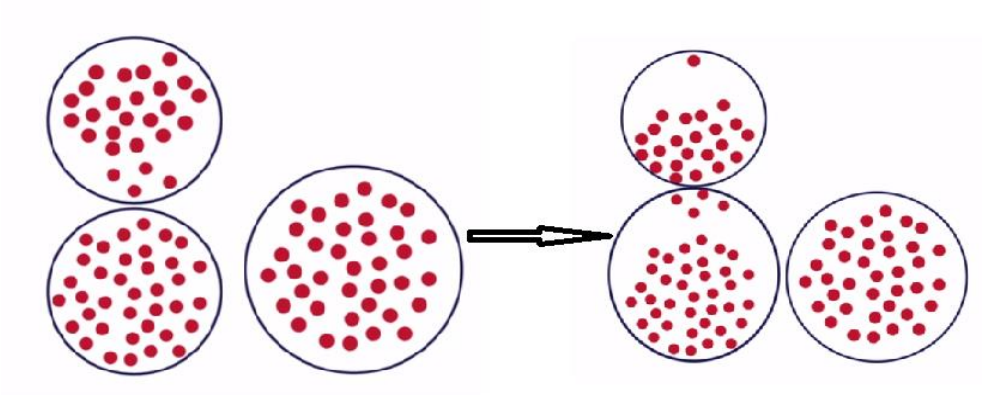

Fig 8: Impact of outliers on clustering

4.) **Standardisation of data**

Standardisation of data, that is, converting them into z-scores with mean 0 and standard deviation 1, is important for 2 reasons in K-Means algorithm:

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

5.) Non-applicability with the categorical data

The K-Means algorithm cannot be used when dealing with categorical data as the concept of distance for categorical data doesn't make much sense. So, instead of K-Means, we need to use different algorithms.