K-Means Clustering Algorithm

K-means clustering is a popular method for grouping data by assigning observations to clusters based on proximity to the cluster's center. This article explores k-means clustering, its importance, applications, and workings, providing a clear understanding of its role in data analysis. In this article, you will explore k-means clustering, an unsupervised learning technique that groups data points into clusters based on similarity. A k means clustering example illustrates how this method assigns data points to the nearest centroid, refining the clusters iteratively. Understanding what is k-means clustering will enhance your grasp of **data analysis** and pattern recognition.

What is K-Means Clustering?

K-means clustering is a popular unsupervised **machine learning algorithm** used for partitioning a dataset into a pre-defined number of clusters. The goal is to group similar data points together and discover underlying patterns or structures within the data.

- Recall the first property of clusters – it states that the points within a cluster should be similar to each other. So, our aim here is to minimize the distance between the points within a cluster.
- There is an algorithm that tries to minimize the distance of the points in a cluster with their centroid – the k-means clustering technique.
- K-means is a centroid-based algorithm or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid.

The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.

Optimization plays a crucial role in the k-means clustering algorithm. The goal of the optimization process is to find the best set of centroids that minimizes the sum of squared distances between each data point and its closest centroid.

How K-Means Clustering Works?

Here's how it works:

1. **Initialization**: Start by randomly selecting K points from the dataset. These points will act as the initial cluster centroids.
2. **Assignment**: For each data point in the dataset, calculate the distance between that point and each of the K centroids. Assign the data point to the cluster whose centroid is closest to it. This step effectively forms K clusters.
3. **Update centroids**: Once all data points have been assigned to clusters, recalculate the centroids of the clusters by taking the mean of all data points assigned to each cluster.
4. **Repeat**: Repeat steps 2 and 3 until convergence. Convergence occurs when the centroids no longer change significantly or when a specified number of iterations is reached.
5. **Final Result**: Once convergence is achieved, the algorithm outputs the final cluster centroids and the assignment of each data point to a cluster.

Objective of k means Clustering

The main objective of k-means clustering is to partition your data into a specific number (k) of groups, where data points within each group are similar and dissimilar to points in other groups. It achieves this by minimizing the distance between data points and their assigned cluster’s center, called the centroid.

Here’s an objective:

- **Grouping similar data points:** K-means aims to identify patterns in your data by grouping data points that share similar characteristics together. This allows you to discover underlying structures within the data.
- **Minimizing within-cluster distance:** The algorithm strives to make sure data points within a cluster are as close as possible to each other, as measured by a distance metric (usually Euclidean distance). This ensures tight-knit clusters with high cohesiveness.
- **Maximizing between-cluster distance:** Conversely, k-means also tries to maximize the separation between clusters. Ideally, data points from

different clusters should be far apart, making the clusters distinct from each other.

What is Clustering?

Cluster analysis is a technique in data mining and **machine learning** that groups similar objects into clusters. K-means clustering, a popular method, aims to divide a set of objects into K clusters, minimizing the sum of squared distances between the objects and their respective cluster centers.

Hierarchical clustering and k-means clustering are two popular techniques in the field of unsupervised learning used for clustering data points into distinct groups. While k-means clustering divides data into a predefined number of clusters, hierarchical clustering creates a hierarchical tree-like structure to represent the relationships between the clusters.

Example of Clustering

Let’s try understanding this with a simple example. A bank wants to give credit card offers to its customers. Currently, they look at the details of each customer and, based on this information, decide which offer should be given to which customer.

Now, the bank can potentially have millions of customers. Does it make sense to look at the details of each customer separately and then make a decision? Certainly not! It is a manual process and will take a huge amount of time.

So what can the bank do? One option is to segment its customers into different groups. For instance, the bank can group the customers based on their income:

High Income
Average Income
Low Income

Can you see where I’m going with this? The bank can now make three different strategies or offers, one for each group. Here, instead of creating different strategies for individual customers, they only have to make 3 strategies. This will reduce the effort as well as the time.

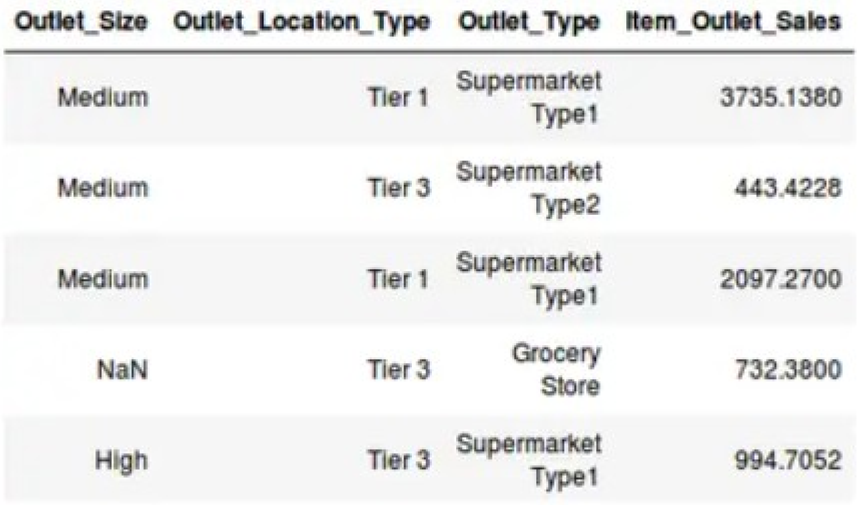
The groups I have shown above are known as clusters, and the process of creating these groups is known as clustering. Formally, we can say that:

- Clustering is the process of dividing the entire data into groups (also known as clusters) based on the patterns in the data.
- Can you guess which type of learning problem clustering is? Is it a **supervised or unsupervised learning** problem?

Think about it for a moment and use the example we just saw. Got it? Clustering is an unsupervised learning problem!

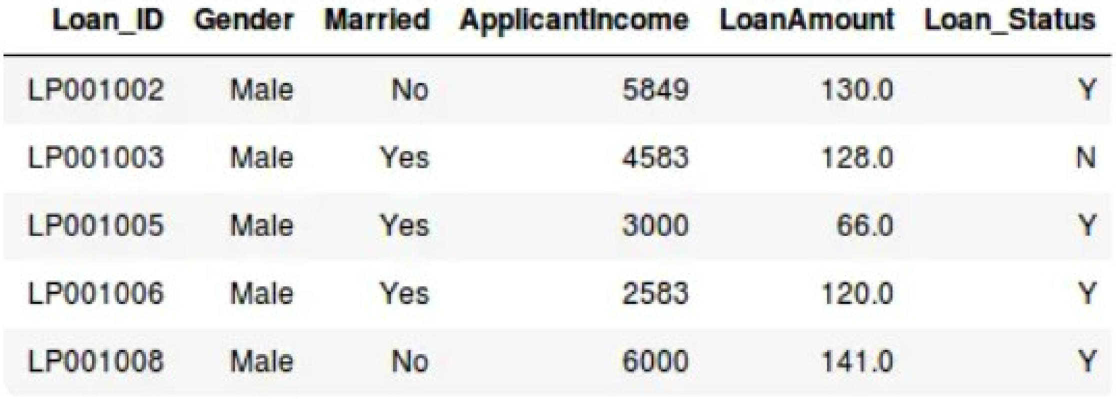
How is Clustering an Unsupervised Learning Problem?

Let’s say you are working on a project where you need to predict the sales of a big mart:



Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales
Medium	Tier 1	Supermarket Type1	3735.1380
Medium	Tier 3	Supermarket Type2	443.4228
Medium	Tier 1	Supermarket Type1	2097.2700
NaN	Tier 3	Grocery Store	732.3800
High	Tier 3	Supermarket Type1	994.7052

Or, a project where your task is to predict whether a loan will be approved or not:



	Loan_ID	Gender	Married	ApplicantIncome	LoanAmount	Loan_Status
	LP001002	Male	No	5849	130.0	Y
	LP001003	Male	Yes	4583	128.0	N
	LP001005	Male	Yes	3000	66.0	Y
	LP001006	Male	Yes	2583	120.0	Y
	LP001008	Male	No	6000	141.0	Y

We have a fixed target to predict in both of these situations. In the sales prediction problem, we have to predict the *Item_Outlet_Sales* based on *outlet_size, outlet_location_type*, etc., and in the loan approval problem, we have to predict the *Loan_Status* depending on the Gender, marital status, the income of the customers, etc.

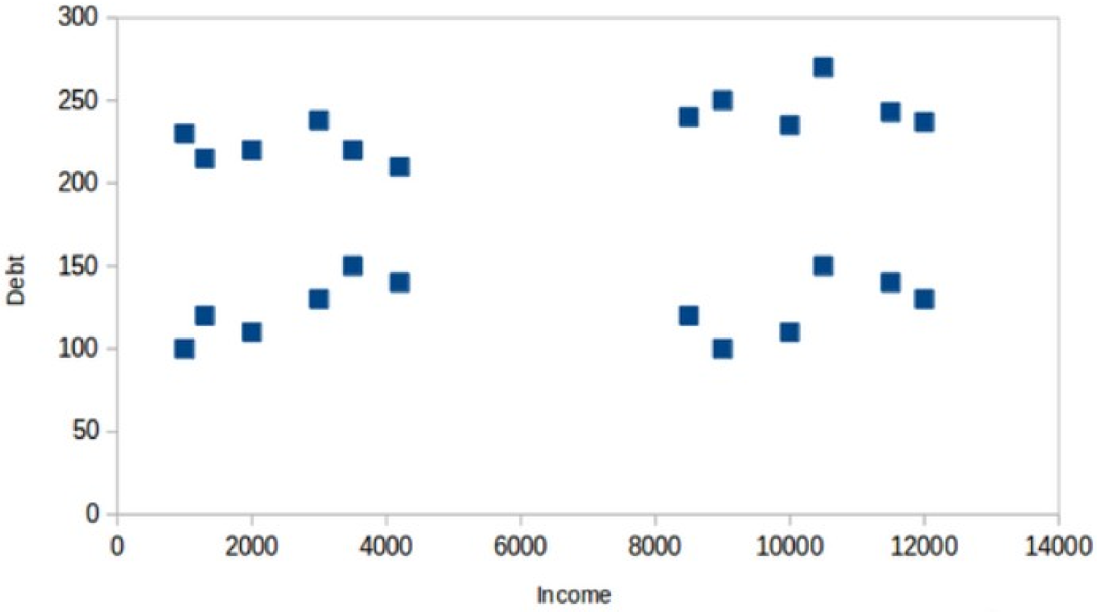
- So, when we have a target variable to predict based on a given set of predictors or independent variables, such problems are called supervised learning problems.
- Now, there might be situations where we do *not* have any target variable to predict.
- Such problems, without any fixed target variable, are known as unsupervised learning problems. In these problems, we only have the independent variables and no target/dependent variable.

In clustering, we do not have a target to predict. We look at the data, try to club similar observations, and form different groups. Hence it is an unsupervised learning problem.

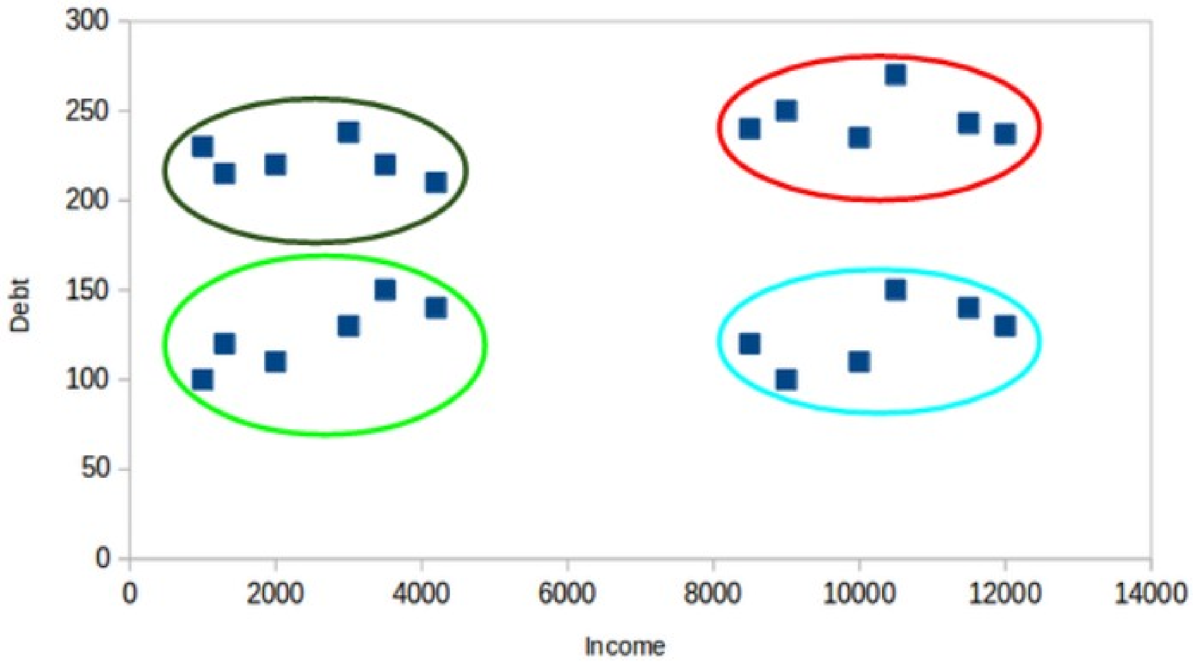
We now know what clusters are and the concept of clustering. Next, let's look at the properties of these clusters, which we must consider while forming the clusters.

Properties of K means Clustering

How about another example of k-means clustering algorithm? We'll take the same bank as before, which wants to segment its customers. For simplicity purposes, let's say the bank only wants to use the income and debt to make the **segmentation**. They collected the customer data and used a scatter plot to visualize it:



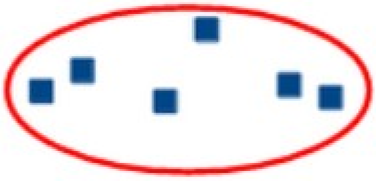
On the X-axis, we have the income of the customer, and the y-axis represents the amount of debt. Here, we can clearly visualize that these customers can be segmented into 4 different clusters, as shown below:



This is how clustering helps to create segments (clusters) from the data. The bank can further use these clusters to make strategies and offer discounts to its customers. So let's look at the properties of these clusters.

First Property of K-Means Clustering Algorithm

All the data points in a cluster should be similar to each other. Let me illustrate it using the above example:

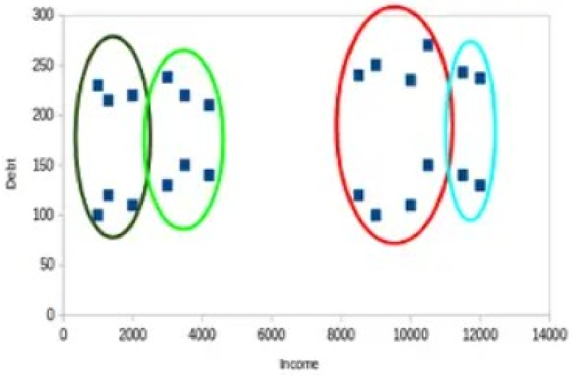


If the customers in a particular cluster are not similar to each other, then their requirements might vary, right? If the bank gives them the same offer, they might not like it, and their interest in the bank might reduce. Not ideal.

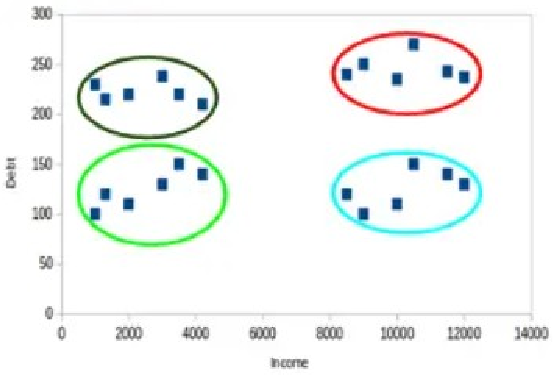
Having similar data points within the same cluster helps the bank to use targeted marketing. You can think of similar examples from your everyday life and consider how clustering will (or already does) impact the business strategy.

Second Property of K-Means Clustering Algorithm

The data points from different clusters should be as different as possible. This will intuitively make sense if you've grasped the above property. Let's again take the same example to understand this property:



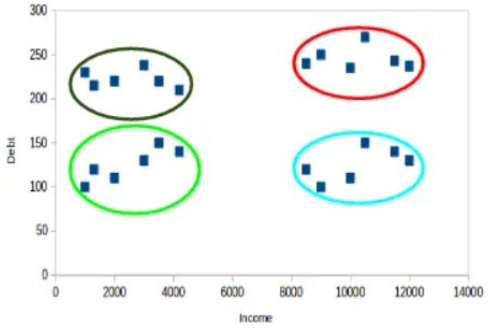
Case - I



Case - II

Which of these cases do you think will give us the better clusters? If you look at case I:

Customers in the red and blue clusters are quite similar to each other. The top four points in the red cluster share similar properties to those of the blue cluster's top two customers. They have high incomes and high debt values. Here, we have clustered them differently. Whereas, if you look at case II:



Case - II

Points in the red cluster completely differ from the customers in the blue cluster. All the customers in the red cluster have high income and high debt, while the customers in the blue cluster have high income and low debt value. Clearly, we have a better clustering of customers in this case.

Hence, data points from different clusters should be as different from each other as possible to have more meaningful clusters. The k-means algorithm uses an iterative approach to find the optimal cluster assignments by minimizing the sum of squared distances between data points and their assigned cluster centroid