upGrad

Table of Contents

Table of Contents 1

K-MEANS CLUSTERING 2

Introduction to Unsupervised Learning and Clustering 2
- Supervised vs Unsupervised Learning 2

K-Means Clustering 3
- Distance as a Similarity Measure 3
- Centroids or Cluster Centers 4
- Good vs Bad Clusters 5
- Working Process 5
- Practical Considerations 6
  - Features on Different Scales 6
  - Choice of K 6
  - Influence of Outliers 7
  - Initial Centroid Values 8
  - Stopping Criteria 9
  - Dimensionality 9
  - Presence of Categorical Variables 9

Other Forms of Clustering 10

Applications of Clustering 11

K-Means Clustering

Introduction to Unsupervised Learning and Clustering

Unsupervised learning is a way for machines to discover hidden patterns in data without any labels or guidance. Instead of being told what to look for, the model explores the data on its own to find structure and relationships that naturally exist. This makes it especially useful when there are no clear categories or outcomes available, but we still want to understand how the data is organised or what insights it might hold.

In simple terms, it's like letting someone wander through a new city without a map. They start noticing which areas look alike, which shops attract similar crowds, and where distinct regions begin to form. Similarly, a computer analyses data to spot similarities, differences, and trends that humans might overlook, especially when the dataset is large or complex.

Clustering is one of the most common and important forms of unsupervised learning. Its main goal is to group similar data points together without any labels or predefined categories. By uncovering natural patterns and relationships within data, clustering helps reveal structure that isn't immediately visible and makes it easier to interpret complex datasets. It is widely used for exploring data, identifying segments, and discovering meaningful patterns across fields such as marketing, healthcare, and security.

While clustering is the most familiar example, unsupervised learning also includes other methods such as dimensionality reduction and association learning. These techniques further support data exploration by simplifying information or finding relationships between items. Together, they help reveal the hidden organisation within unlabelled data and guide deeper understanding.

Supervised vs Unsupervised Learning

Given below are the general differences between supervised and unsupervised learning.

Aspect	Supervised Learning	Unsupervised Learning
Data Type	Uses labelled data where inputs have known outputs	Uses unlabelled data with no predefined answers

Goal	Learns to predict outcomes or classify new data based on past examples	Finds hidden patterns, structures, or groupings within data
Examples of Use	Spam detection, price prediction, sentiment analysis	Customer segmentation, topic modelling, anomaly detection
Feedback and Accuracy	Model performance can be measured against known results	No direct measure of accuracy since true labels are unknown

K-Means Clustering

K-means clustering is one of the simplest and most popular techniques used to group data in unsupervised learning. The idea is to divide data points into a set number of clusters, where each cluster contains points that are more similar to each other than to those in other clusters.

The algorithm works by repeatedly assigning points to the nearest cluster centre (called a *centroid*) and then updating these centres based on the average position of the points within each cluster. This process continues until the groupings no longer change significantly.

K-means is widely used because it is fast, easy to understand, and effective for finding clear patterns or natural divisions in data, such as grouping customers by behaviour or segmenting images by colour.

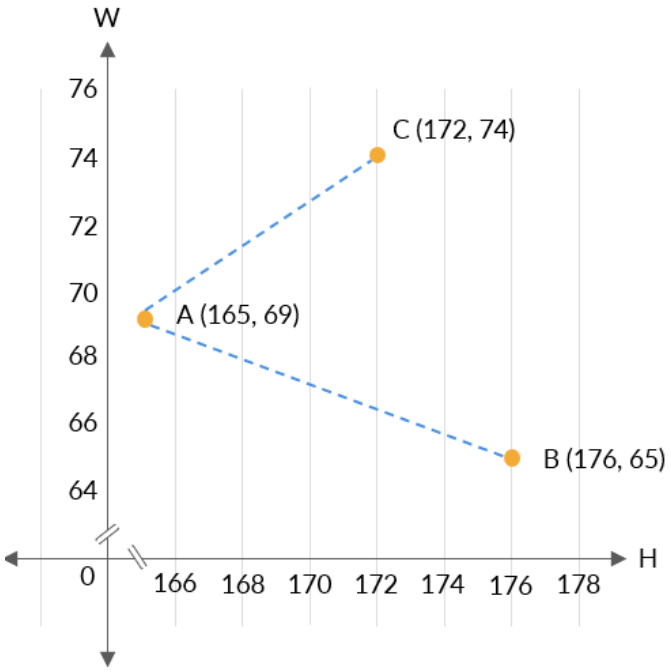
To understand how K-means actually forms these groups, we need to look at the key ideas it relies on, such as measures of distance, the concept of similarity, and the role of centroids in defining each cluster.

Distance as a Similarity Measure

Distance is the foundation of similarity in K-means clustering — it tells us how close or far apart two data points are. The most common choice is **Euclidean distance**, which measures the straight-line distance between points in space, much like measuring with a ruler. Points that are closer together are considered more similar and are grouped into the same cluster.

The Euclidean distance between two points can be found by applying the formula $d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$. Here, the terms x_i and y_i represent the coordinates (or feature values) of the two points being compared.

For instance, consider the following as a depiction of how distance can be calculated.



Person	Height (cm)	Weight (kg)
A	165	69
B	176	65
C	172	74

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$d_{AB} = 11.7$

$d_{AC} = 8.6$

Centroids or Cluster Centers

A **centroid** or **cluster centre** represents the average position of all points within a cluster. It is calculated by taking the mean value of each feature for the points assigned to that cluster, giving a single representative point in the data space.

During clustering, each data point is assigned to the nearest centroid using a distance measure, often the Euclidean distance. The centroids are then updated to reflect the new cluster assignments, and this process repeats until the centroids no longer change significantly.

Centroids help describe the main characteristics of each cluster and make it easier to compare groups within a dataset. However, since they are based on averages, they may not fully capture the variation or irregular shapes that exist in real data.

Good vs Bad Clusters

A cluster may be said to be good if:

- Internal (intra-cluster or within the cluster) distances should be small
    - This can be captured by using the within-cluster sum of squares (WCSS) or inertia which is given by the formula $\text{WCSS} = \sum_{i=1}^{n} \|C - x_i\|^2$ where C is the centroid
- External (inter-cluster or between-cluster) distances should be lerge

Refer to the following table to understand the difference between good and bad clusters.

Aspect	Supervised Learning	Unsupervised Learning
Compactness	Low WCSS, meaning points are tightly grouped around the centroid	High WCSS, indicating large variation within clusters
Separation	High distances between clusters, showing clusters are well-separated	Low distances between clusters, suggesting that clusters are not distinct from one another
Silhouette Score	High (close to 1), meaning points are well matched to their own cluster	Low or negative, suggesting poor or overlapping clustering
Cluster Shape	Compact and roughly spherical (for distance-based methods like K-Means)	Irregular or elongated, violating algorithm assumptions
Interpretability	Clusters make sense in the problem context	Clusters are hard to explain or lack real meaning

Working Process

The K-means clustering algorithm works as follows

- Choose the **number of clusters** (*K*) to form
- **Randomly initialise** *K* centroids from the dataset
- **Assign** each data point to the **nearest centroid** based on distance
- **Recalculate centroids** as the mean of assigned points
- Repeat assignment and update steps until centroids **stabilise**
- Evaluate clustering quality using WCSS or silhouette score

In addition, you may also tune for the parameters of individual decision trees used by the forest.

Practical Considerations

The following are some of the practical considerations one should take into account while performing K-means clustering.

Features on Different Scales

K-means clustering is highly sensitive to the scale of features because it uses distance-based calculations, typically the Euclidean distance. When features have different ranges or units, those with larger values can dominate the distance computation, leading to biased clustering results.

For example, in a customer segmentation task, annual income might be measured in lakhs of rupees, while customer satisfaction rating is on a scale of 1 to 5. Without scaling, the large numeric range of income would dominate the clustering process, causing the algorithm to form clusters mainly based on income and largely ignore satisfaction levels, even if those ratings are crucial for meaningful insights.

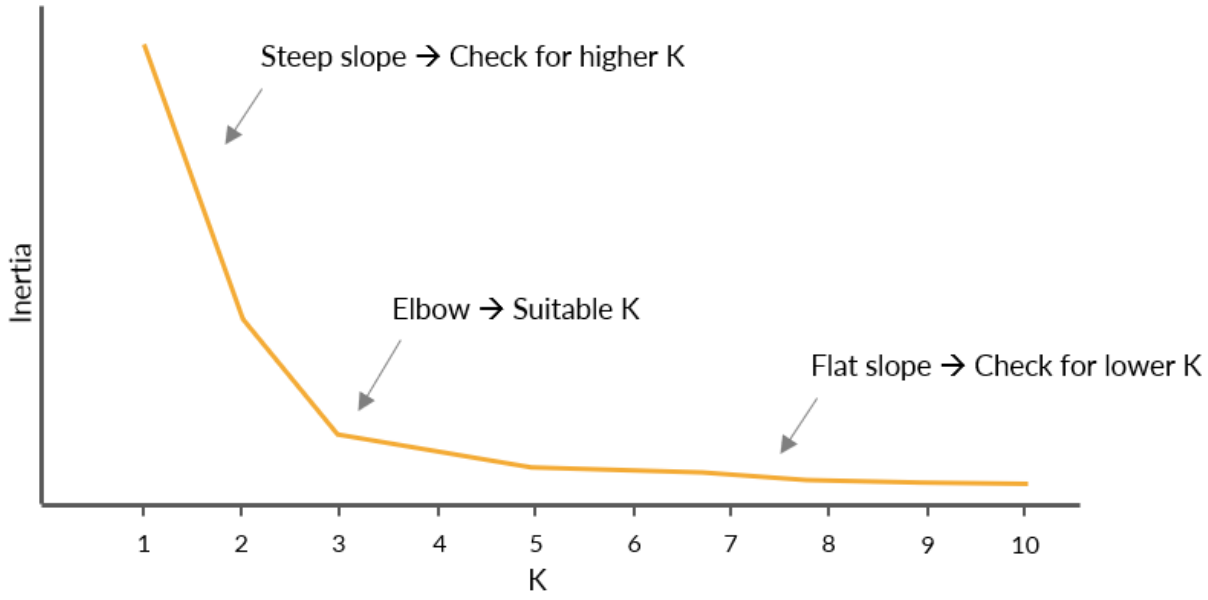
To avoid this imbalance, it is essential to scale features before applying K-means. Standardisation (converting values to have zero mean and unit variance) or normalisation (scaling values to a fixed range, such as 0 to 1) ensures that all features contribute equally. This step helps produce clusters that genuinely represent the structure of the data rather than the influence of measurement scales.

Choice of K

Choosing the right number of clusters, *K*, is one of the most important decisions in K-means clustering. The **elbow method** is a common approach for this. It involves plotting the Within-Cluster Sum of Squares (WCSS) against different values of K and looking for a

point where the rate of decrease sharply slows down — the "elbow." This point suggests a balance between compact clusters and model simplicity.

Consider the following plot of inertia versus *K*. The "elbow" being formed here suggests an optimal value of *K*.



Another useful technique is the **silhouette score**, which measures how similar each point is to its own cluster compared to other clusters. The score ranges from –1 to 1, with higher values indicating better-defined clusters. By comparing silhouette scores across different K values, one can identify the configuration that provides the most cohesive and well-separated clustering.

Note that, in practical applications, the ideal *K* may not always align with what these methods suggest. **Business domain knowledge** often plays a key role. For instance, a marketing team might prefer three customer segments because they match existing strategies, even if the data-driven optimum is slightly different.

Influence of Outliers

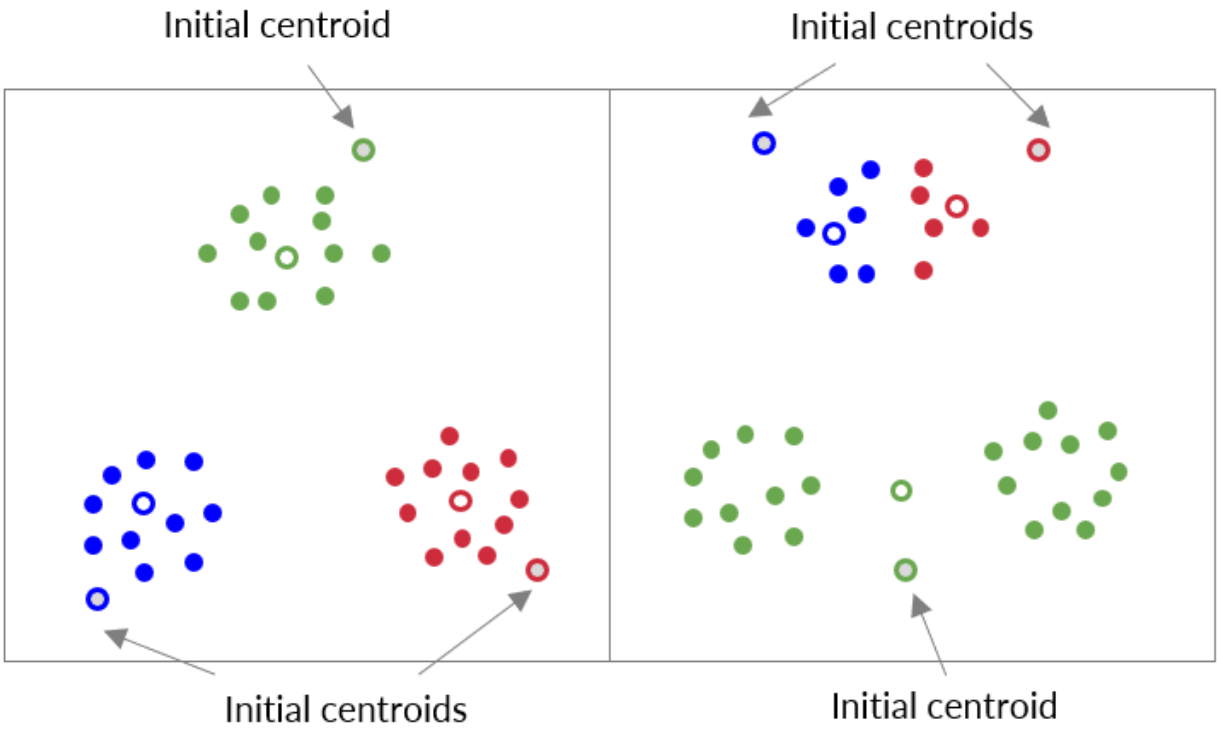
K-means is sensitive to outliers, as it relies on the mean to determine cluster centres. Even a single extreme data point can pull a centroid towards it, distorting the cluster's position and affecting the assignment of nearby points. This can lead to inaccurate or unbalanced clusters, especially when the dataset contains rare but extreme values.

To reduce this influence, it is important to detect and handle outliers before applying K-means. This can be done by removing extreme values, applying transformations such as logarithmic scaling, or by using alternative clustering models.

Initial Centroid Values

The choice of **initial centroid values** can significantly affect the outcome of K-means clustering. Since the algorithm starts with randomly selected centroids, different initialisations can lead to different final clusters, sometimes getting stuck in local minima rather than finding the best overall solution. This makes the results unstable if the process is repeated multiple times.

Consider the following as an example of how the outcome of K-means clustering can differ depending on the initialisation of centroids.



To address this, methods like K-means++ are used to improve initialisation. K-means++ selects initial centroids in a way that spreads them out across the data space, reducing the likelihood of poor starting positions. Running the algorithm several times with different seeds and comparing results is also a good practice to ensure stability and consistency in clustering outcomes. Scikit-learn's *KMeans()* function uses K-means++ as its default method for initialisation.

Stopping Criteria

The stopping criteria in K-Means determine when the algorithm should end its iterative process. Typically, it stops when the centroids no longer move significantly between iterations, meaning the clusters have stabilised.

Another common condition is when the change in the WCSS falls below a defined threshold, indicating minimal improvement in cluster compactness.

Alternatively, a maximum number of iterations can be set to prevent the algorithm from running indefinitely, especially with large or complex datasets.

Combining these criteria ensures that K-Means stops efficiently, balancing computational effort with the need for stable and meaningful clustering results.

Dimensionality

As the dimensionality of data increases, K-means becomes more challenging to apply effectively. In high-dimensional spaces, data points tend to become **sparse,** meaning they are far apart from one another. This reduces the contrast between distances, making it harder for the algorithm to distinguish meaningful clusters.

Higher dimensions also lead to **greater computational costs** and make results more difficult to interpret, as visualising or explaining clusters becomes complex. To address this, dimensionality reduction techniques are often used before clustering, helping to simplify the data while retaining its key patterns.

Presence of Categorical Variables

K-means is designed for numerical data, as it relies on calculating means and distances such as the Euclidean distance.

When categorical variables are included, these operations lose meaning because categories cannot be averaged or meaningfully compared through standard distance metrics. For instance, consider a category with the following levels: “Android”, “Apple”, and “Other”.

If we encode “Android”, “Apple”, and “Other” as 0, 1, and 2, respectively, the mean of this category cannot reasonably be interpreted as “Apple” (value 1).

Similarly, if we apply one-hot encoding — [1, 0, 0], [0, 1, 0], [0, 0, 1] — the mean would be [⅓, ⅓, ⅓], which does not correspond to any valid category.

Moreover, Euclidean distance assumes continuous numerical space, so it cannot properly capture relationships between categories.

Because of this, categorical variables should be handled using alternative approaches such as K-modes or K-prototypes, which replace means with modes and adapt the distance metric to work meaningfully with categorical data.

Other Forms of Clustering

K-Means is efficient and simple but has clear limitations. It assumes numerical data, spherical clusters, and similar cluster sizes. It also struggles with categorical features, irregular shapes, and outliers, which can distort the results.

K-modes clustering addresses this by working with categorical data. Instead of computing a mean, it identifies the most frequent value (mode) in each cluster, making it suitable for non-numeric features such as colours, brands, or categories.

K-prototypes clustering combines the logic of K-means and K-modes to handle mixed datasets that contain both numerical and categorical variables, ensuring meaningful clustering across different data types.

Hierarchical clustering takes a different approach by gradually merging or splitting clusters, producing a tree-like structure that allows exploration of data at various levels of detail.

Density-based spatial clustering of applications with noise (DBSCAN) groups data points based on density rather than distance from a centroid, enabling it to find clusters of irregular shapes and identify noise or outliers effectively.

Fuzzy clustering allows data points to belong to multiple clusters by assigning degrees of membership, which is useful in cases where boundaries between groups are not clearly defined.

Beyond these, other techniques such as **gaussian mixture models (GMM)** and **spectral clustering** offer more flexibility for complex data structures.

Together, these methods show that clustering is a broad and adaptable field, where choosing the right technique depends on the data type, distribution, and the insights one aims to uncover.

Applications of Clustering

Clustering has wide-ranging applications across industries because it helps uncover natural groupings within data without requiring predefined labels. In **marketing and customer analytics**, it is commonly used for **customer segmentation**, where buyers are grouped based on behaviour, demographics, or spending patterns. This allows businesses to tailor marketing strategies, personalise offers, and improve customer retention through targeted engagement.

In **healthcare**, clustering is applied to identify patterns in patient data, such as grouping individuals with similar symptoms, genetic markers, or treatment responses. This aids in disease diagnosis, drug discovery, and the development of personalised treatment plans. It also helps detect anomalies in medical imaging or patient monitoring data, supporting early intervention.

In technology and research, clustering is used in areas such as **image recognition**, **document classification**, and **anomaly detection**. For example, it helps group similar images in large datasets, cluster research papers by topic, or identify unusual network activity in cybersecurity. Across domains, clustering provides structure to complex data, revealing hidden relationships that support better decision-making and deeper insight.