UNSUPERVISED MACHINE LEARNING

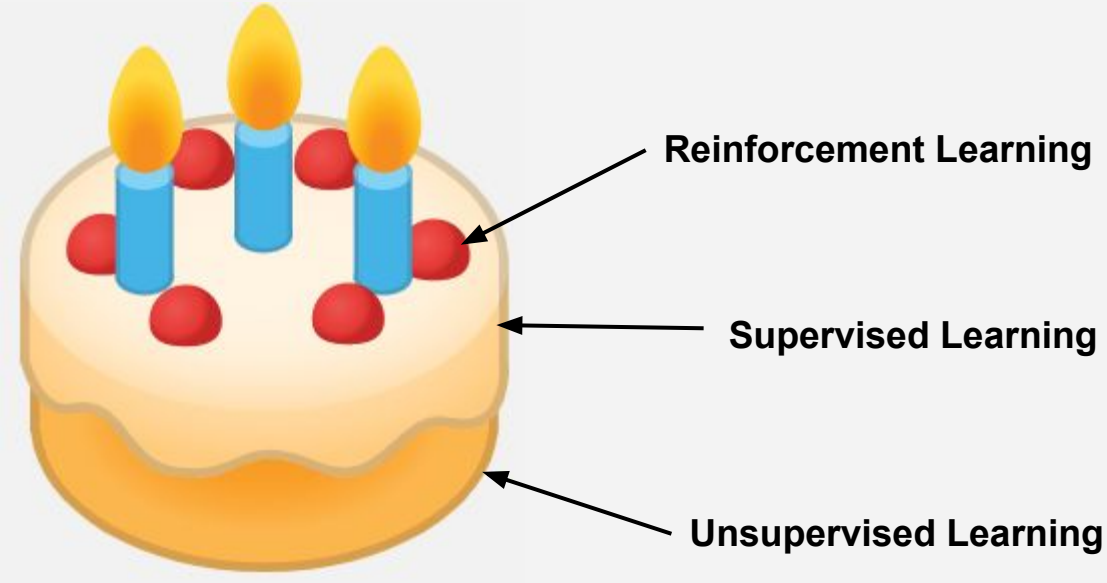
K-Means Clustering

Contents

Supervised Vs Unsupervised Machine Learning

- Supervised learning:
  - Deals with labelled dataset
  - Supervised learning works by feeding the machine sample data with various features (represented as “X”) and the correct value output of the data (represented as “y”). The fact that the output and feature values are known qualifies the dataset as **labeled**.
  - The algorithm then deciphers patterns that exist in the data and creates a model that can reproduce the same underlying rules with new data.
- Unsupervised learning:
  - In the case of unsupervised learning, the datasets are not labeled. Here the machine must uncover hidden patterns and create labels through the use of unsupervised algorithms.

UNSUPERVISED MACHINE LEARNING

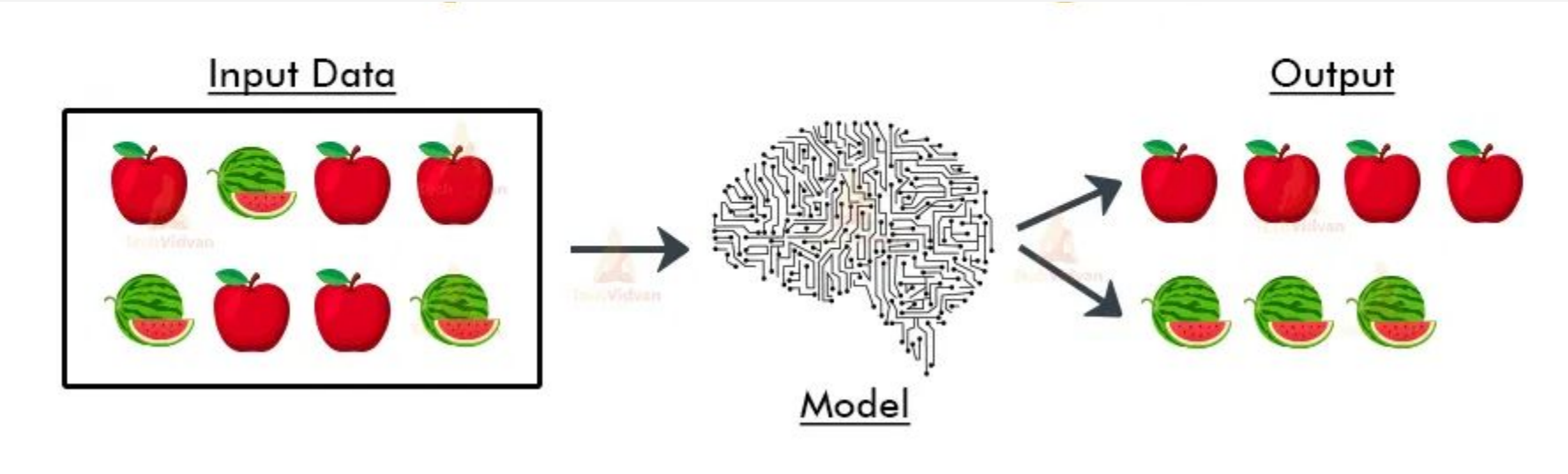


UNSUPERVISED LEARNING TECHNIQUES

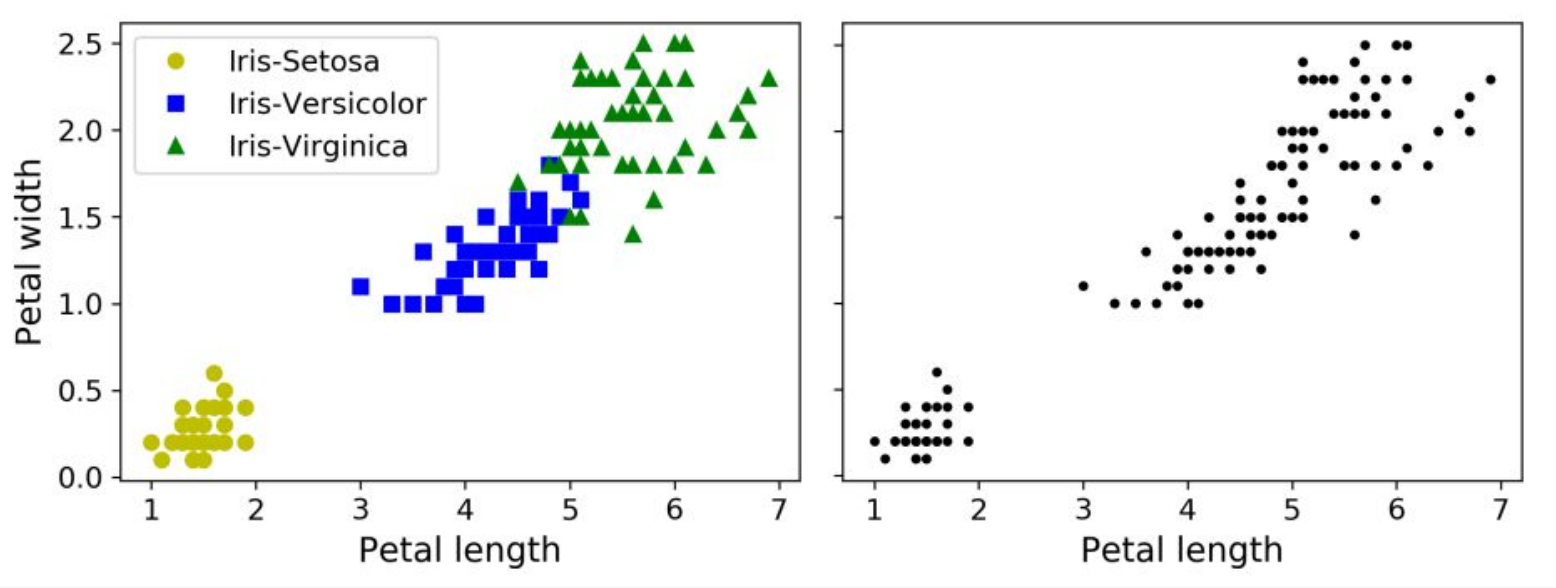
- **Clustering:** The goal is to group similar instances together into clusters. This is a great tool for data analysis, customer segmentation, recommender systems, search engines, image segmentation, semi-supervised learning, dimensionality reduction, and more.
- **Anomaly Detection:** The objective is to learn what “normal” data looks like, and use this to detect abnormal instances, such as defective items on a production line or a new trend in a time series.
- **Density Estimation:** This is the task of estimating the probability density function (PDF) of the random process that generated the dataset. This is commonly used for anomaly detection: instances located in very low-density regions are likely to be anomalies. It is also useful for data analysis and visualization.

CLUSTERING

A cluster refers to a **collection of data points aggregated together** because of certain similarities.



CLUSTERING


Iris-Setosa
Iris-Versicolor
Iris-Virginica
Petal width
Petal length
Petal length

APPLICATIONS OF CLUSTERING

Market Segmentation

Search Engines

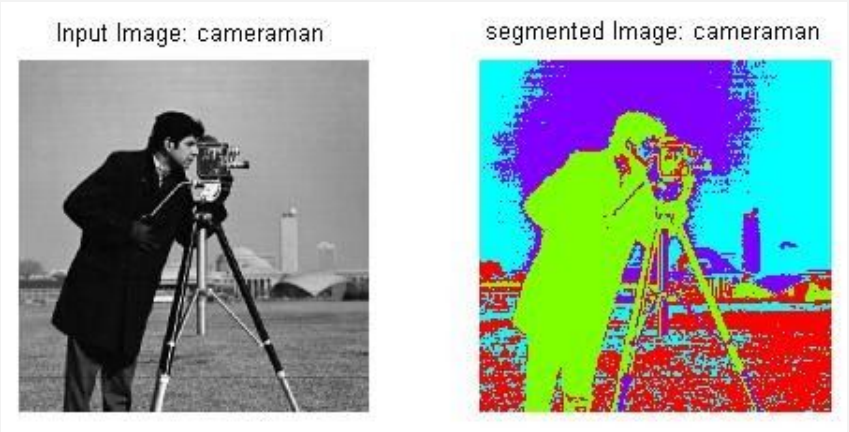
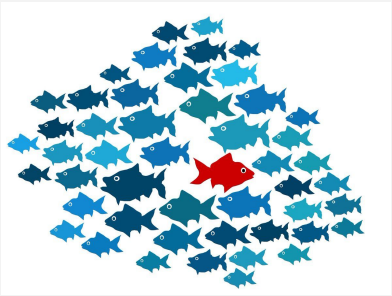


Image Segmentation



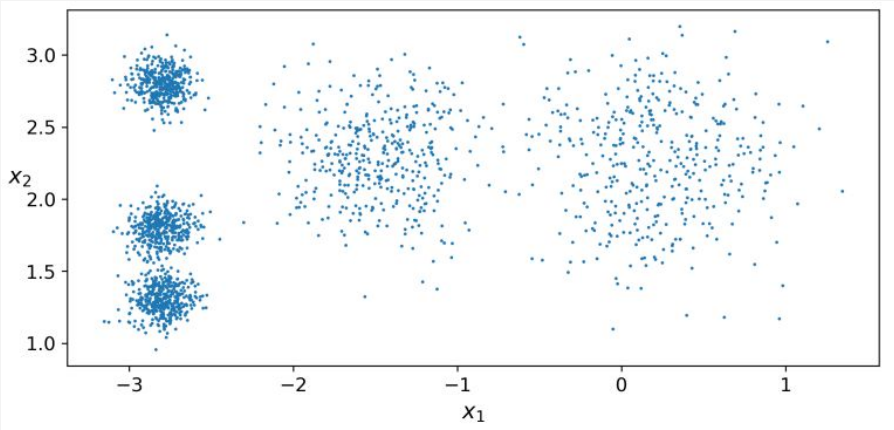
Anomaly Detection

K- MEANS CLUSTERING

Objective: Group similar data points together and discover underlying patterns. To achieve this, K-means looks for a fixed number (*k*) of clusters in a dataset.

- It is an iterative algorithm
- It does two things:
  - Cluster Assignment
  - Root Centroid Step

A **centroid** is the imaginary or real location representing the center of the cluster.



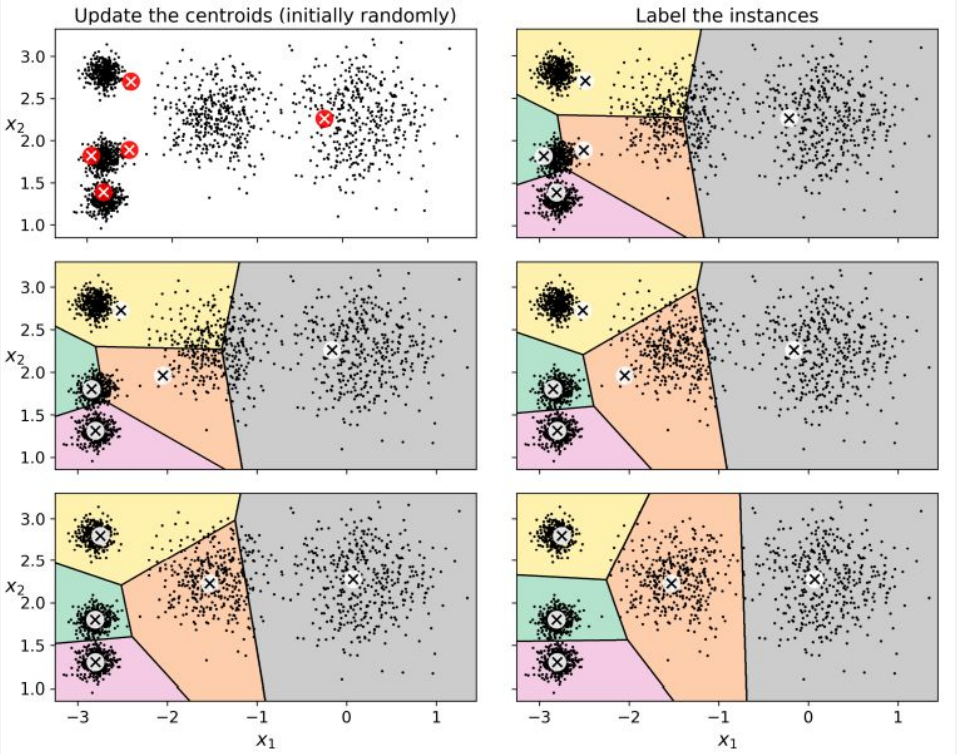
K-means algorithm identifies *k* number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

The *'means'* in the K-means refers to averaging of the data; that is, finding the centroid.

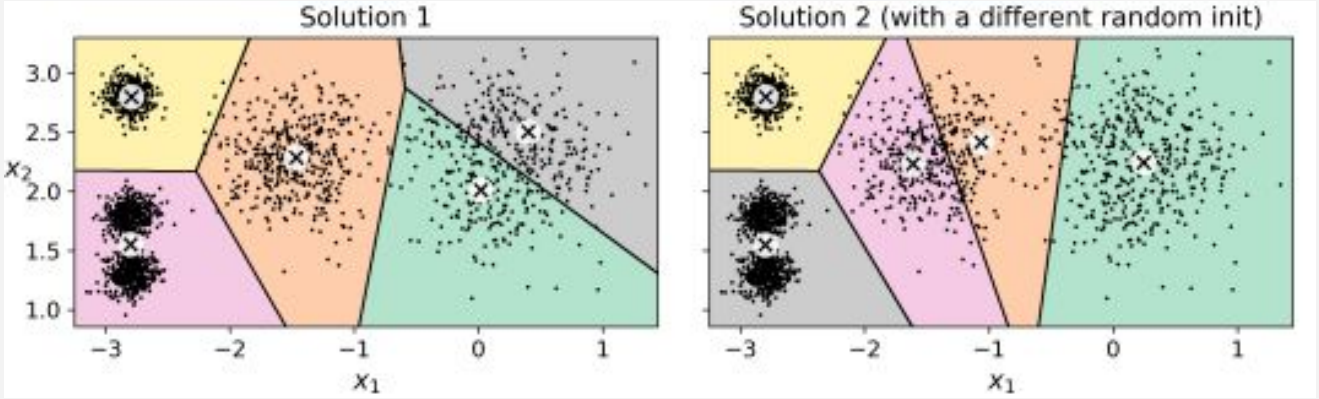
K- MEANS CLUSTERING ALGORITHM

Steps:

1. Start with K centroids placed at random
2. Compute distance of every point from centroid and cluster them accordingly
3. Adjust the centroids so that they become center for given cluster
4. Again re-cluster every point based on their distance with centroid
5. Again adjust centroid
6. Recompute clusters and repeat till the data points stop changing clusters

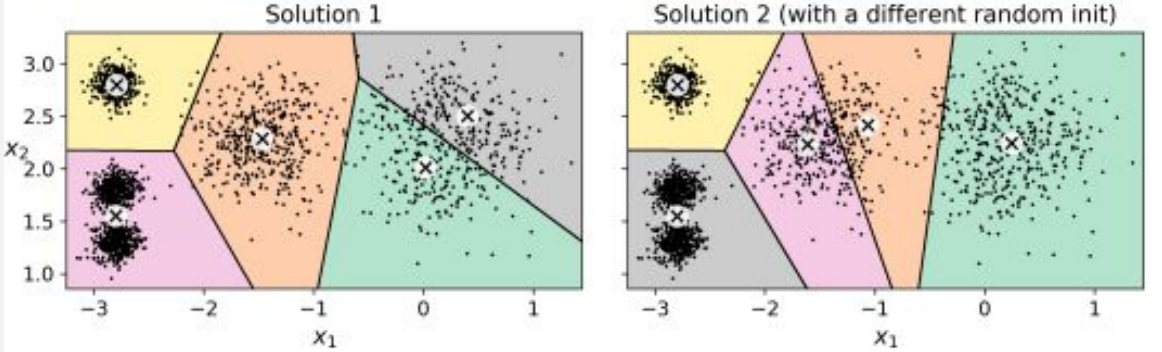


Although the algorithm is guaranteed to converge, it may not converge to the right solution (i.e., it may converge to a local optimum) and this depends on the centroid initialization.



Sub-optimal solutions due to unlucky centroid initializations

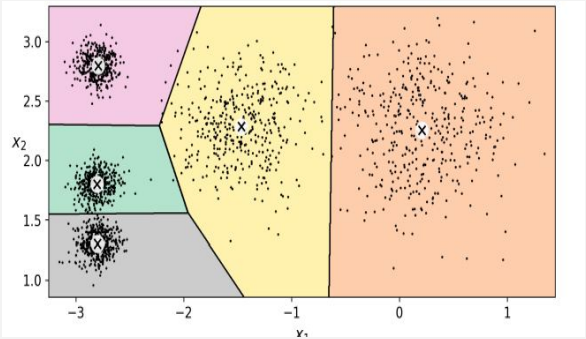
Centroid Initialization Methods



223.3

237.5

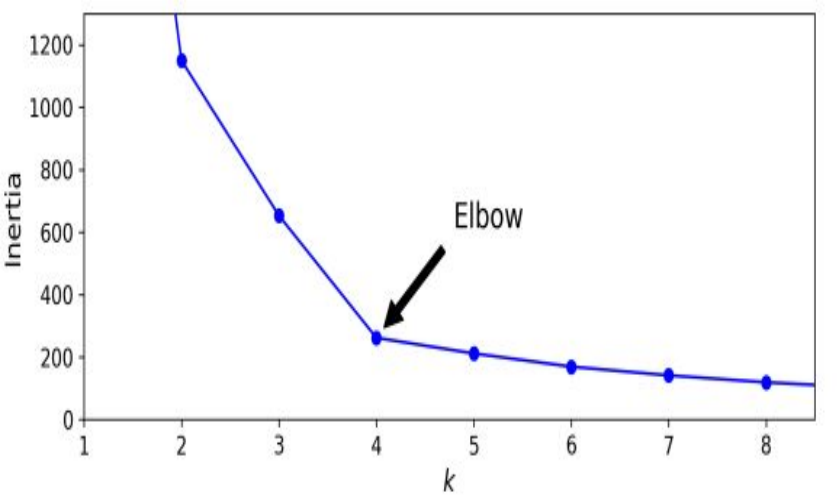
```
kmeans = KMeans(n_clusters=5, init=good_init, n_init=1)
```



211.6

Elbow Method

- Inertia is not a good performance metric when trying to choose k
- The more clusters there are, the closer each instance will be to its closest centroid, and therefore the lower the inertia will be
- Inertia drops very quickly as we increase k up to 4, but then it decreases much more slowly as we keep increasing k



Selecting the number of clusters k using the "elbow rule"

When using the K-means algorithm, keep the following points in mind:

- It is suggested to normalize the data while dealing with clustering algorithms such as K-Means since such algorithms employ distance-based measurement to identify the similarity between data points.
- Because of the iterative nature of K-Means and the random initialization of centroids, K-Means may become stuck in a local optimum and fail to converge to the global optimum. As a result, it is advised to employ distinct centroids' initializations.

MERITS:

- Fast
- K-means would be faster than Hierarchical clustering if we had a high number of variables.
- It enables you to discover patterns in the data that you were unaware existed
- When compared to Hierarchical clustering, K-means produces tighter clusters.

DEMERITS:

- K-Means is not perfect. We need to run the algorithm several times to avoid sub-optimal solutions, plus you need to specify the number of clusters, which can be quite a hassle.
- It's quite sensitive to rescaling. If we rescale our data using normalization or standards, the outcome will be drastically different.
- It is not advisable to do clustering tasks if the clusters have a sophisticated geometric shape.

** It is important to scale the input features before running K-Means, or else the clusters may be very stretched, and K-Means will perform poorly. Scaling the features does not guarantee that all the clusters will be nice and spherical, but it generally improves things*