Unsupervised Learning and K-Means Clustering

UNSUPERVISED LEARNING:

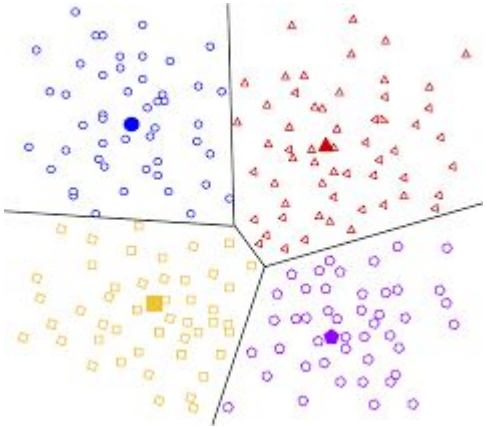
- **Unsupervised learning** is a machine learning task in which patterns are learnt from datasets consisting of samples without labels i.e., there exist only features but no targets variables

- Unsupervised learning is useful for understanding the distribution of data or patterns in data which when there is no explicit measure or guiding signal

- Unsupervised learning tasks include clustering, association mining and outlier or anomaly detection

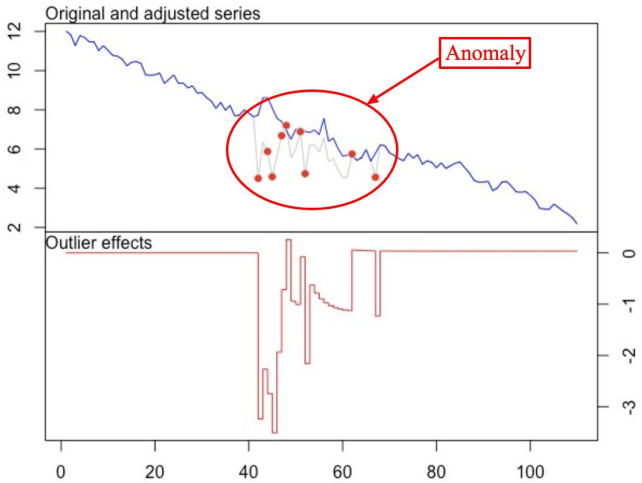
APPLICATIONS

Clustering is trying to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure. It is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data.



APPLICATIONS

Anomaly detection can automatically discover unusual data points in your dataset. This is useful in pinpointing fraudulent transactions, discovering faulty pieces of hardware, or identifying an outlier caused by a human error during data entry.

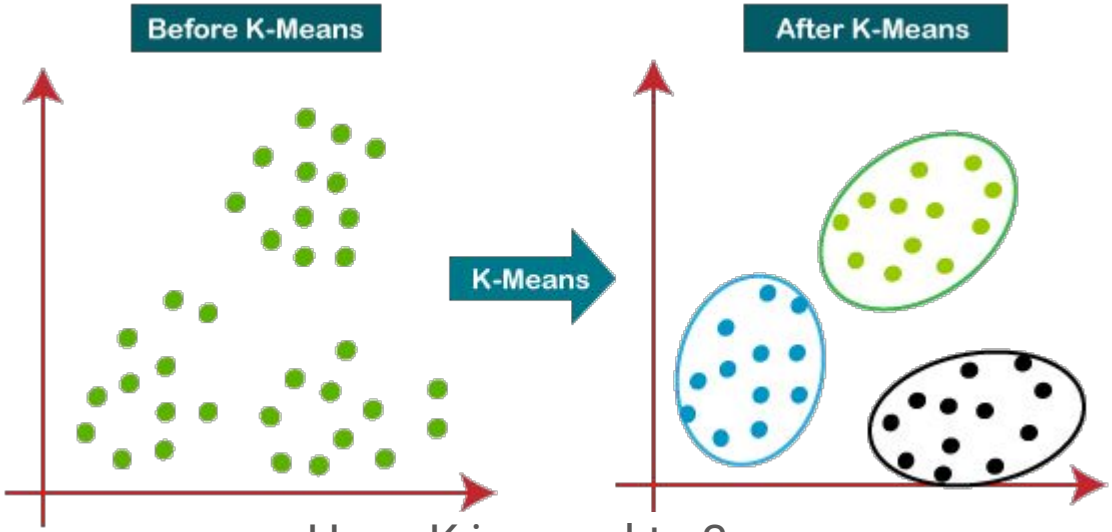


APPLICATIONS

Association mining identifies sets of items that frequently occur together in your dataset. Retailers often use it for basket analysis, because it allows analysts to discover goods often purchased at the same time and develop more effective marketing and merchandising strategies.

K Means Clustering

- It is one of the simplest **clustering** algorithms.
- K-means is a centroid-based clustering algorithm to group the data points into k clusters
- Here, distances are calculated based on which a point is assigned to a cluster



Here, K is equal to 3

K Means Clustering Algorithm

Step 1: Choose the number of clusters *k*

- The first step in k-means is to pick the number of clusters, k.
- Data visualisation can be used to choose the number of clusters but it is tough when the data is high dimensional. In such a case, it can be done by plotting an **elbow curve** (described later).

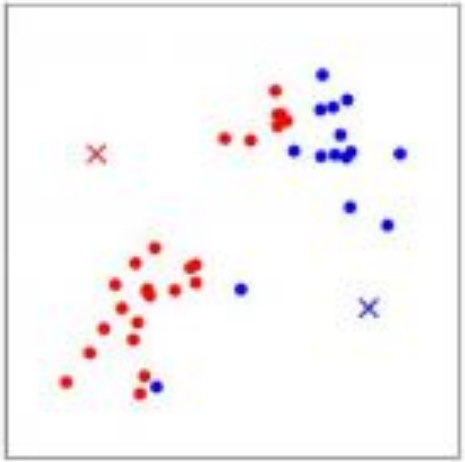
Step 2: Select k random points from the data as centroids

- Next, we randomly select the centroid for each cluster. Let's say we want to have 2 clusters, so k is equal to 2 here. We then randomly select the centroids.

Here the blue and red crosses represent the initial cluster centroids

Step 3: Assign all the points to the closest cluster centroid

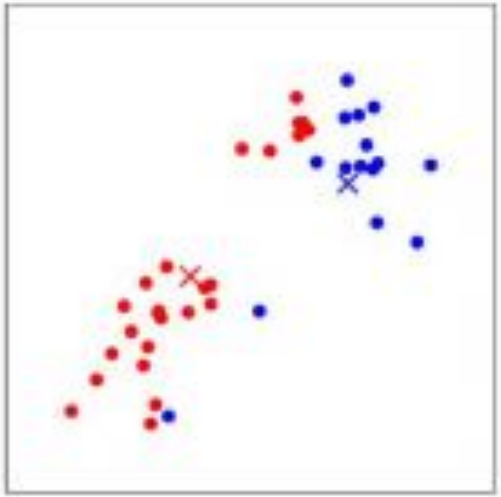
- After initializing k centroids, assign each point to the closest cluster centroid by calculating the distances between the point and the k centroids



Here the points closer to the blue cross (centroid) are assigned to the blue cluster and so is the case with red centroid

Step 4: Recompute the centroids of newly formed clusters

- Once we have assigned all of the points to either cluster, the next step is to compute the centroids of newly formed clusters



Here the new clusters are represented by red and blue clusters. Do note the shift in the centroids.

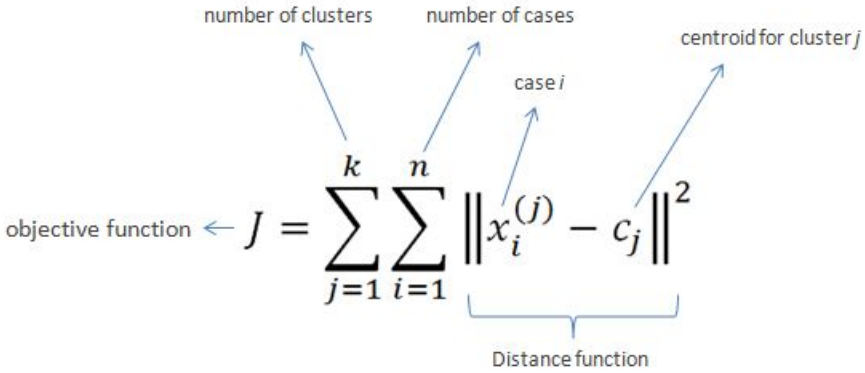
Step 5: Repeat steps 3 and 4

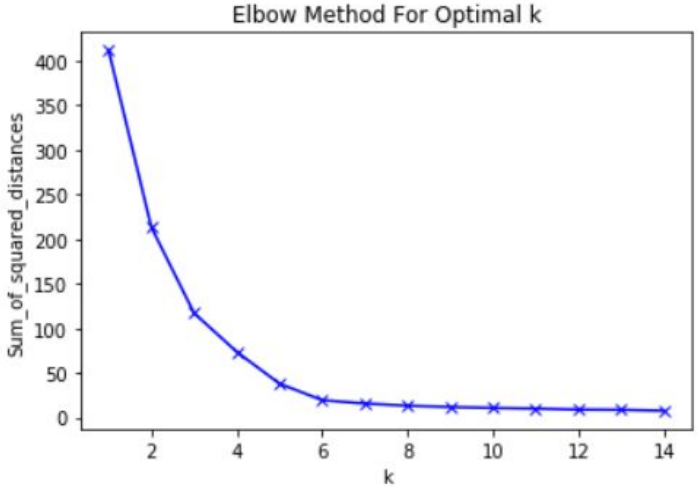
- Keep iterating over the steps 3 and 4 till the stopping criteria is met
- Stopping criteria:
  a. Centroids of newly formed clusters do not change: Even after multiple iterations, if the cluster centroids don’t shift or shift by a small amount, the iterations can stop
  b. Points remain in the same cluster: Another criteria to stop the training process is if the points remain in the same cluster even after training the algorithm for multiple iterations
  c. Maximum number of iterations are reached: Alternately, the number of iterations can be set to a maximum value and the algorithm stops after those iterations

Heuristic method to find Optimal k

Elbow Method:

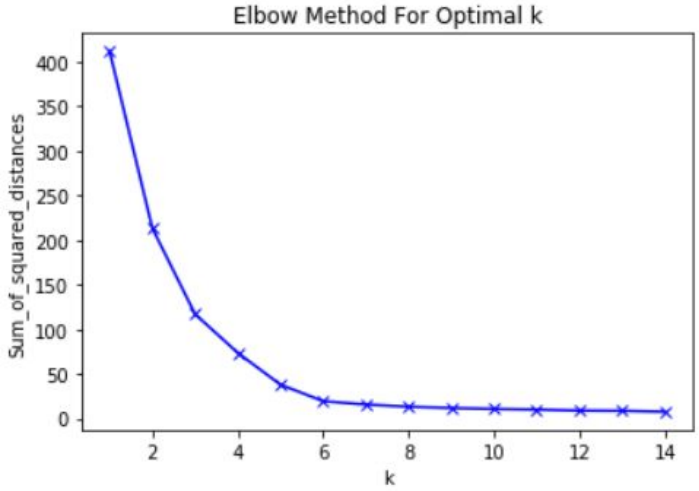
- This metric is based on the **within Cluster Sum of Squared (WCSS)** distances.
- The sum of squared distance between data points and their assigned clusters' centroids are calculated. It can be repeated for different values of *k* and a curve is plotted to find an optimal *k*.



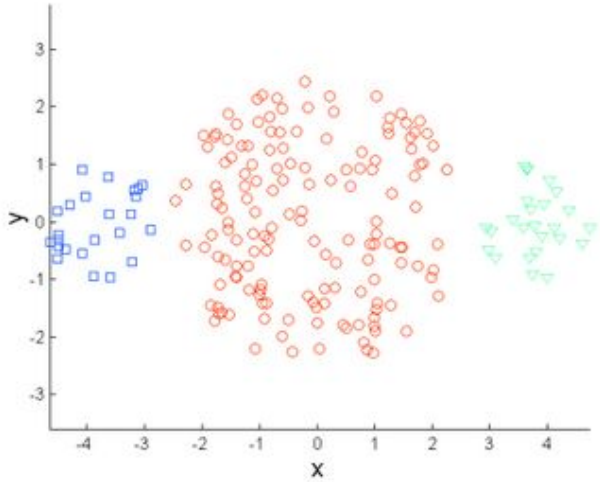


Drawbacks

1) **Choosing k manually**: The elbow needs to be drawn to determine an optimal value for k which might be computationally intensive

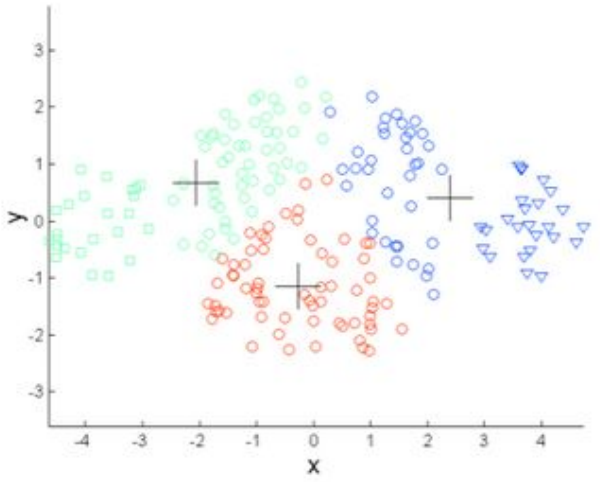


2) Dependant in initial values of centroids: Bad initialization may lead to unwanted clusters as shown in the figure below. This can be mitigated by trying multiple initializations or using kmeans++ algorithm (https://en.wikipedia.org/wiki/K-means%2B%2B)



Original Points

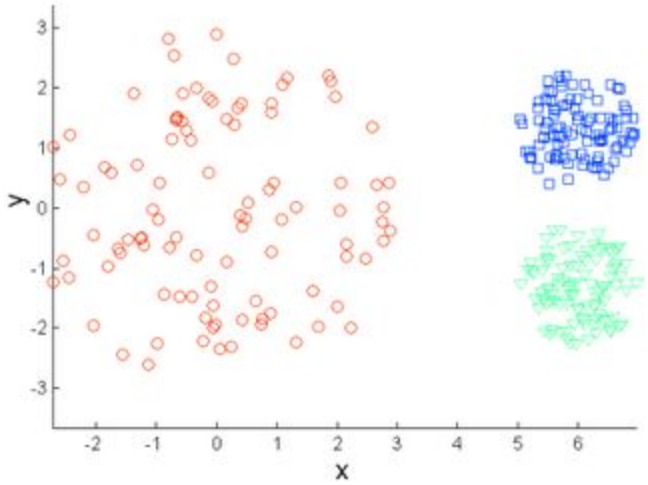
Expected Clusters



K-means (k = 3)

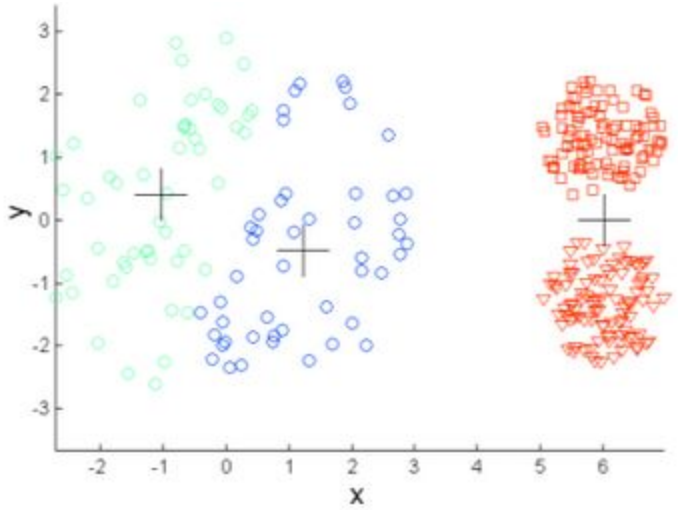
Clustering by K-Means

3) **Clustering data of varying sizes and density:** K means has trouble clustering data where clusters are of varying sizes and density. Alternately, density based clustering algorithms can be used.



Original Points

Expected Clusters



K-means (k = 3)

Clustering by K-Means