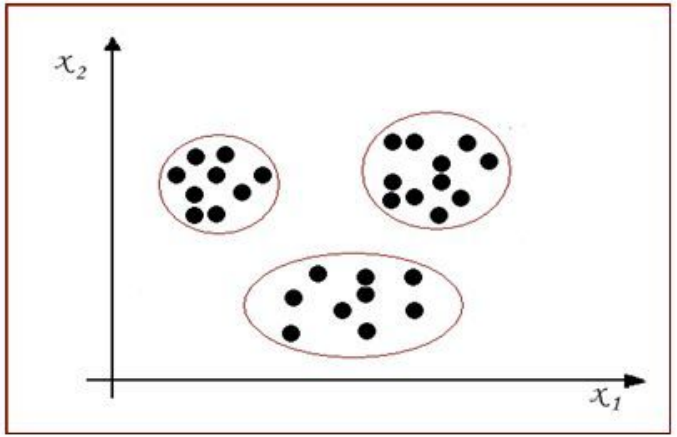
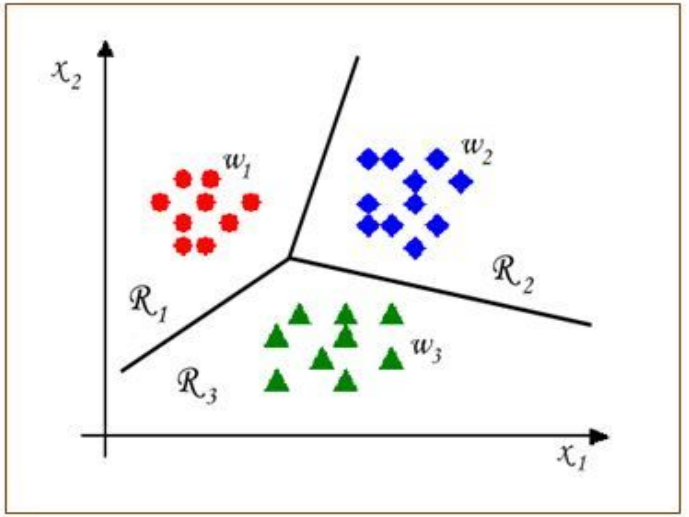
Clustering vs. Classification



Given some patterns, the goal is to discover the underlying structure (categories) in the data based on inter-pattern similarities



Given some training patterns from each class, the goal is to construct decision boundaries or to partition the feature space

Clustering

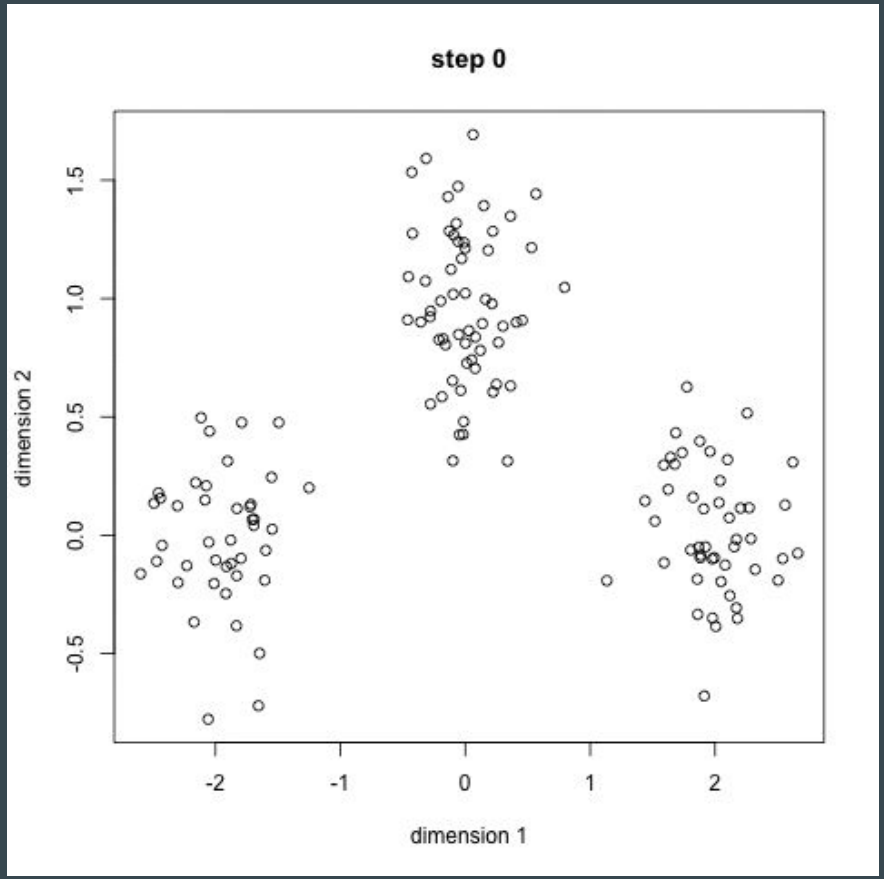
- Clustering techniques apply when there is ***no class to be predicted*** but rather when the ***instances are to be divided into natural groups***.
- It involves automatically discovering natural grouping in data.
- Clustering can be helpful as a data analysis activity in order to learn more about the problem domain, so-called ***pattern discovery*** or knowledge discovery.
- Clustering is an ***unsupervised learning*** technique, so it is hard to evaluate the quality of the output of any given method.
- Examples:
    - Market Segmentation
    - News Grouping
    - Social Network Analysis

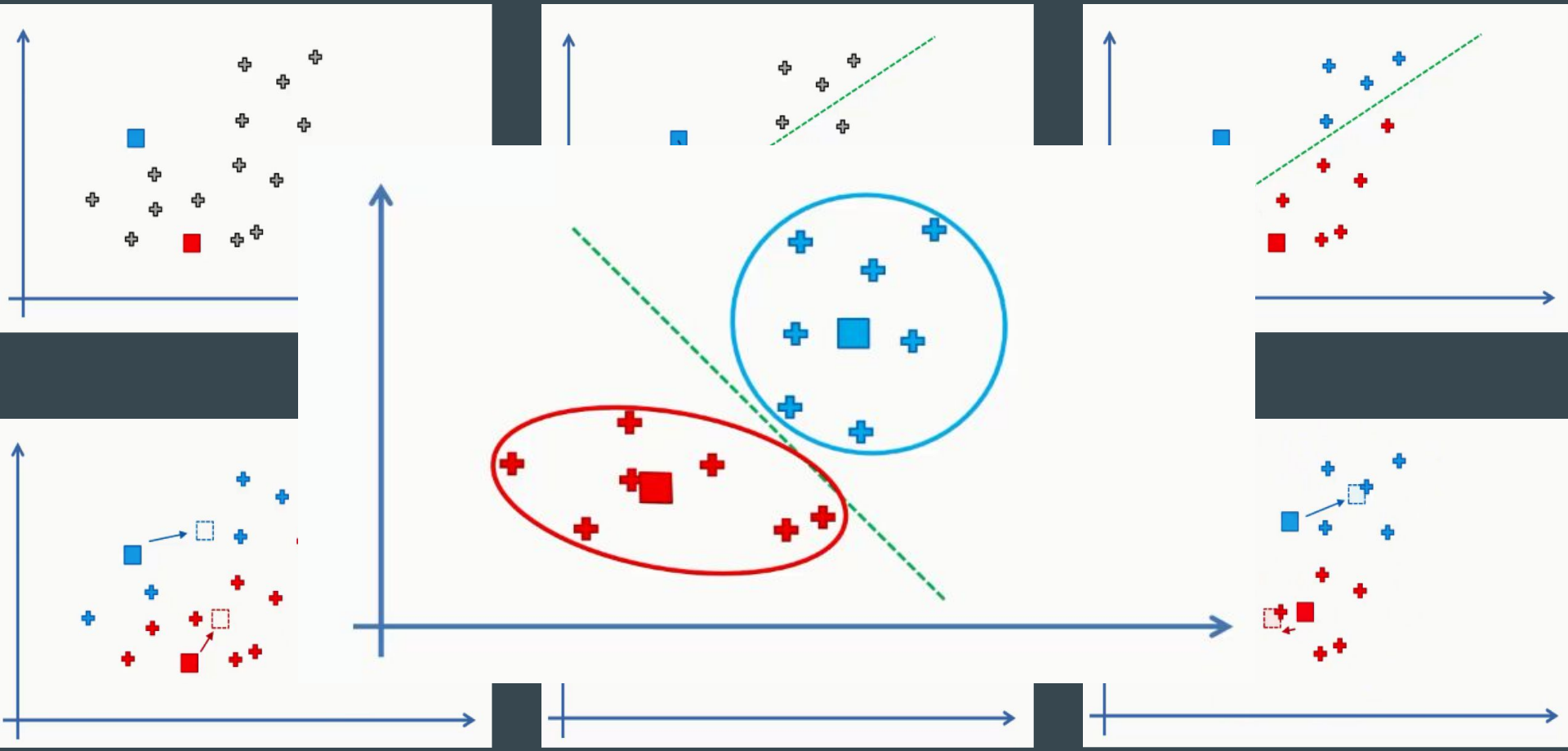
Clustering

- Many algorithms ***use similarity or distance measures*** between examples to discover dense regions of observations.
- As such, it is often good practice to ***scale data*** prior to use clustering algorithms.
- The scikit-learn library provides a suite of different clustering algorithms to choose from, you can check here: Clustering@Sklearn
- ***K-Means Clustering*** may be the most widely known clustering algorithm and involves assigning examples to clusters in an effort to minimize the variance within each cluster.

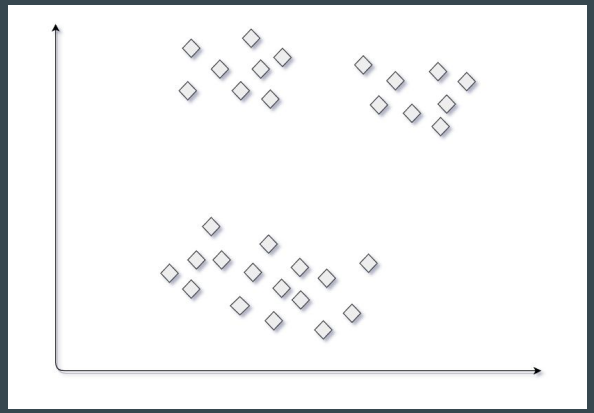
K-means Clustering Algorithm

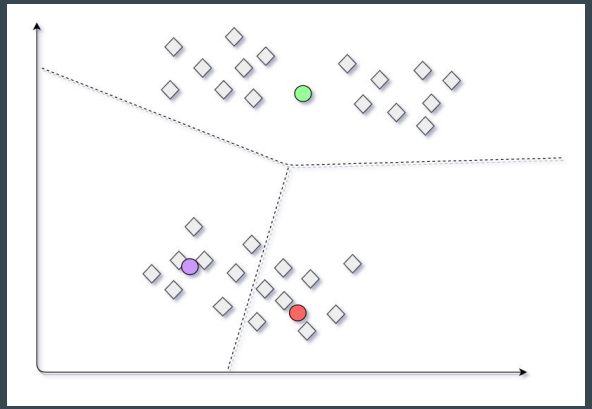
1. *Choose the number k of clusters*
2. *Select at random k points, the centroids*
3. *Assign each data point to the closest centroid, so we have k clusters*
   - ➔ *Euclidean distance is generally used.*
4. *Compute and place the new centroid of each cluster*
5. *Repeat steps 3, 4 till no change occurs*

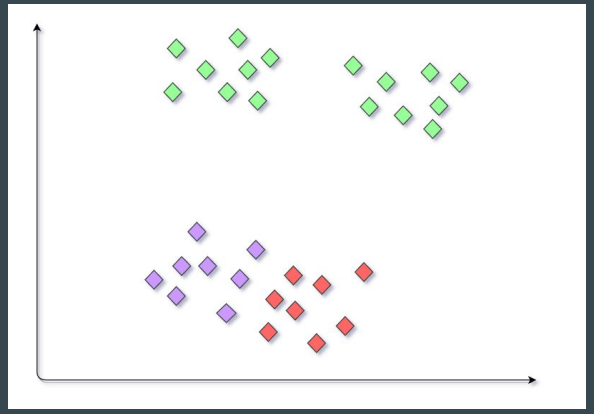


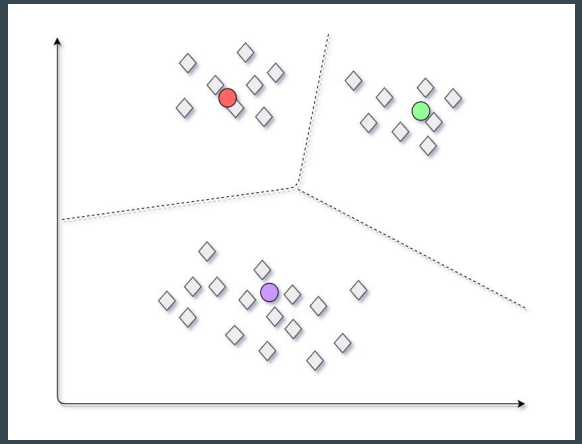


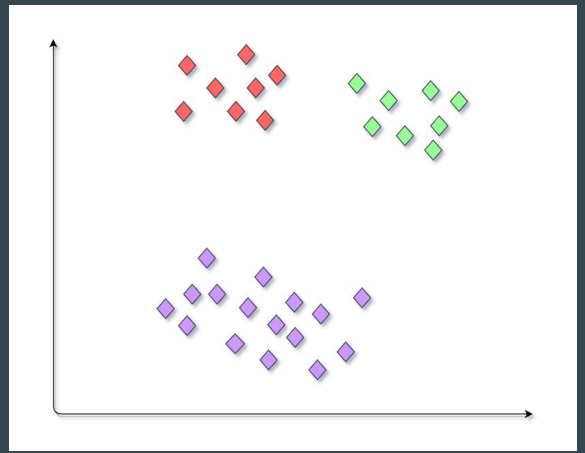
Random Initialization Trap in K-means Algorithm









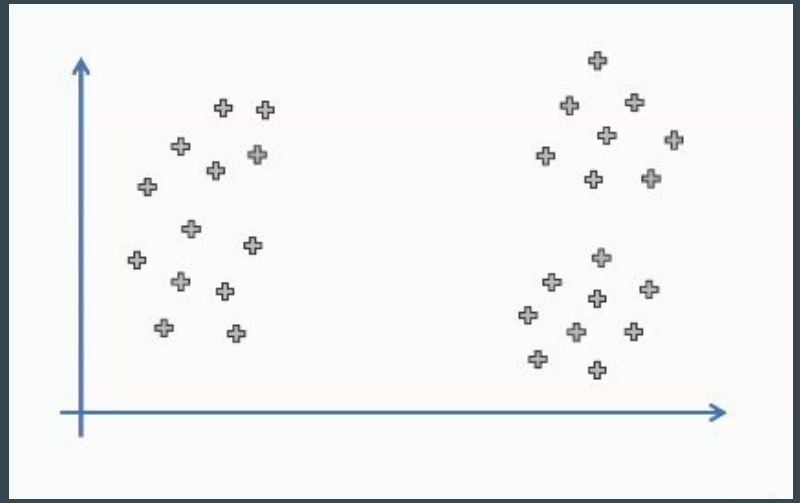


Random Initialization Trap Solution : K-means ++

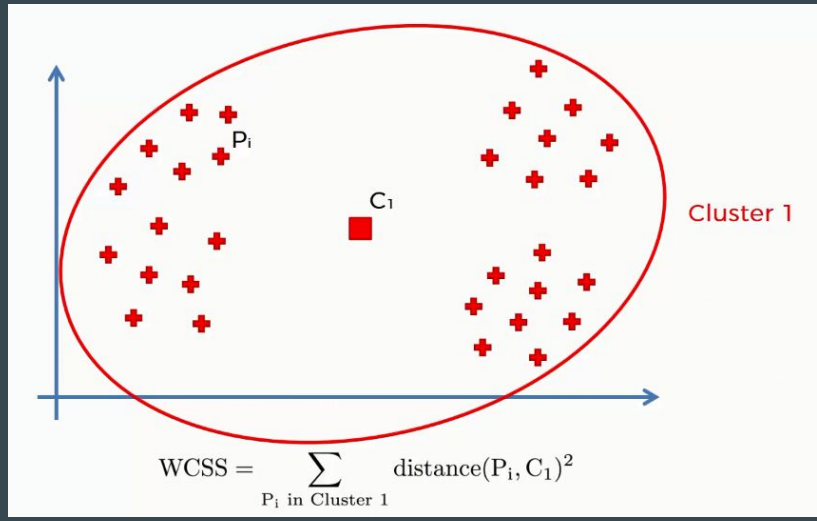
- The more optimal the positioning of initial centroids, the fewer iterations will be required for convergence.
- ***How does k means ++ work to solve the issue?***

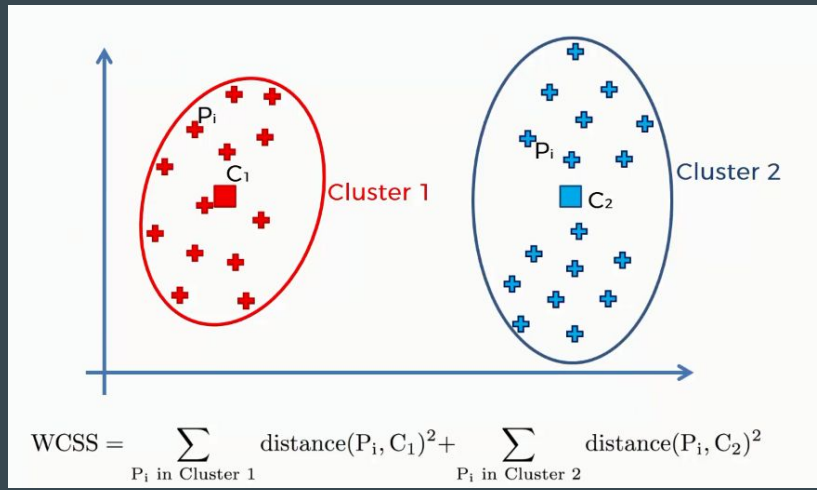
1. Randomly select the first centroid from the data points.
2. For each data point compute its distance from the nearest, previously chosen centroid.
3. Select the next centroid from the data points such that the probability of choosing a point as centroid is directly proportional to its distance from the nearest, previously chosen centroid. (i.e. the point having maximum distance from the nearest centroid is most likely to be selected next as a centroid)
4. Repeat steps 2 and 3 until k centroids have been sampled

Elbow Method for optimal value of k

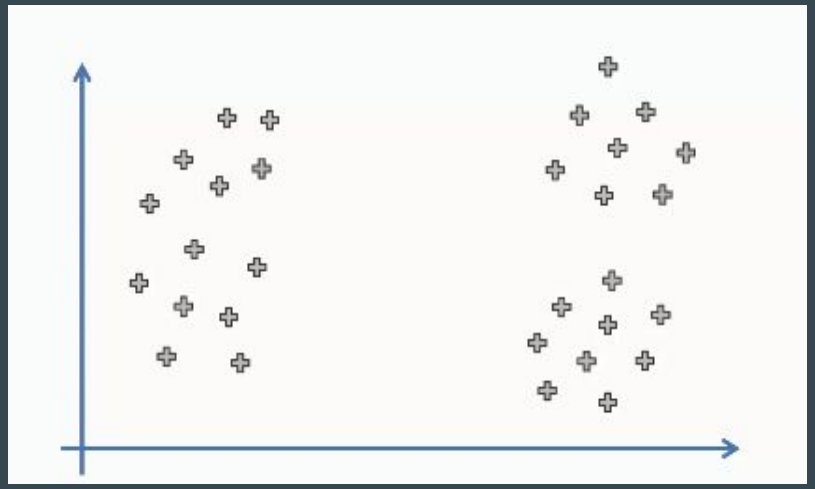


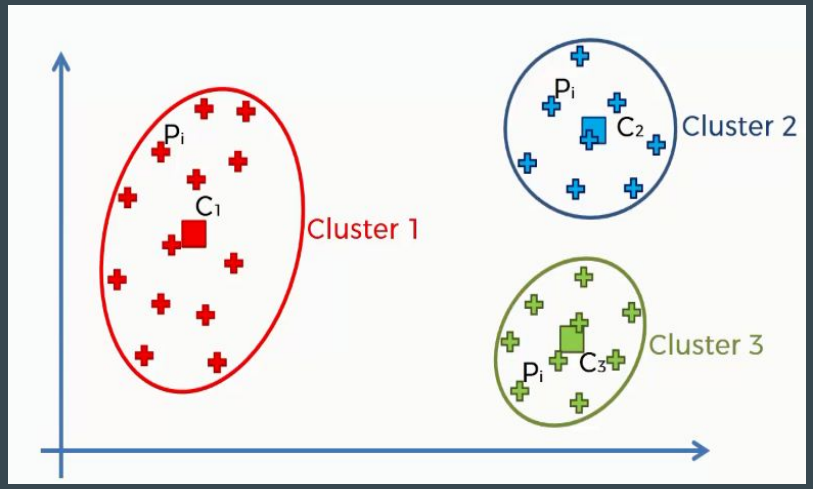
- WCSS is the sum of squares of the distances of each data point in all clusters to their respective centroids.
- It is also known as the inertia





Elbow Method for optimal value of k





$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

Elbow Method for optimal value of k

- k-means attempts to minimize the inertia (WCSS) when choosing Clusters
- More clusters means lower inertia
- Choose an "elbow" in the inertia plot where inertia begins to decrease more slowly