Unsupervised learning

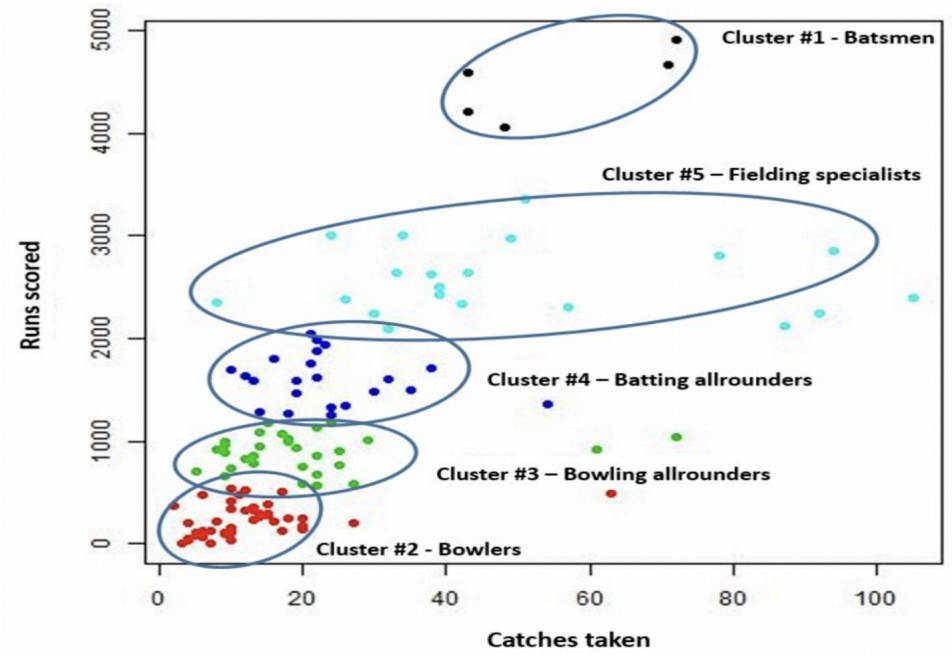
- In the case of both classification and regression problems, we are trying to find a function that is used to predict y_i when x_is are given as the input. Both of these are supervised learning problems, where the models are trained with a target variable.

- The basic and simplest definition of Unsupervised learning is:
    - Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled data sets.
    - These algorithms discover hidden patterns in data without the need for human intervention (hence, they are “unsupervised”).
- Unsupervised learning deals with data that is unlabelled or hasn’t a target variable.

Clustering

- The process of grouping any kind of data based on the similarity in their features, automatically, without human expertise, is called **clustering**. It is a type of **unsupervised learning**.

- Intuitively, clustering is dividing a population into groups such that the points in one group are similar to each other. Each group is called a **cluster**.
    - The points in the same cluster are more closer and similar to each other.
    - The points in different clusters are more distant and distinct from each other.
- So, the task in clustering is grouping the points of a similar kind based on our definition of similarity. For example,



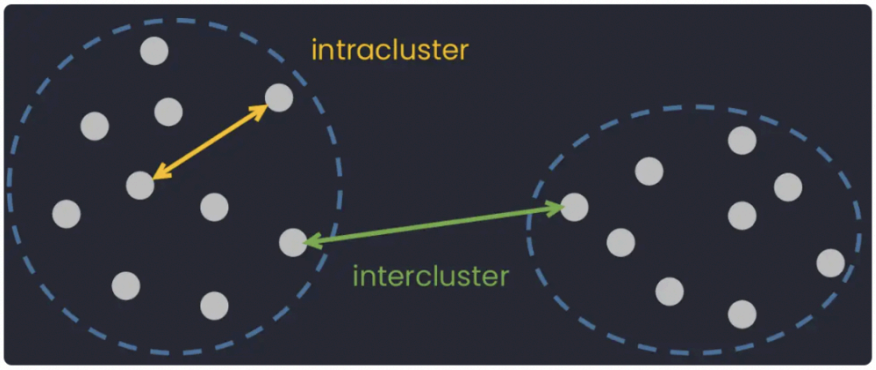
In the image, we can see clusters of different kind of players on the basis of runs scored and catch taken.

- Since there is no ground truth data and nothing to compare with, we decide if a cluster is good or bad if it simply makes some **business sense.**

- The similarity in clustering can be defined as the closeness of data points with each other.
- **Similarity** can be measured using different distance metrics like euclidean distance, manhattan distance, and hamming distance.

Distances used while clustering:

- **Inter-cluste**r distance represents the distance between two clusters
    - Distance between average values of the clusters.
    - Distance between closest points from the clusters (min distance)
    - Distance between farthest points from the clusters (max distance)
- **Intra-cluster** distance represents the distance within a certain cluster. Basically, it measures how tightly the points of clusters are packed.
    - Average distance between the points of a cluster.
    - Distance between farthest points of a cluster



- Having only one inter or intra-cluster distance won't tell us how good or bad our clusters therefore we need a metric to evaluate our clusters.

Dunn Index

- **Dunn Index** is a metric for the evaluation of clustering algorithms. It is calculated as a ratio of the **smallest** inter-cluster distance to the **largest** intra-cluster distance.

i.e. $$D = \frac{minimum \;\; inter-cluster \;\; distance}{maximum \;\; intra-cluster \;\; distance}$$

- The objective of the Dunn index is to identify clusters that are:
  - compact with a small variance between members of the cluster
  - and well separated

- A higher Dunn Index means better clustering since observations in each cluster are closer together, while clusters themselves are further away from each other.

- Dunn Index is **unbound**, so it can only be interpreted in a relative sense.

- For analyzing models using Dunn-index:
  - we can find the values of the metric for a random clustering algorithm and

    - then compare it with the values of the metric for different clustering algorithms to find out whether the algorithm is performing better or worse than a random model.

Introduction to K-Means

- K-Means clustering is one of the most popular and simplest clustering algorithms. The value 'K' in the K-means algorithm denotes the number of clusters.

- In k-means data is divided into k clusters where each cluster has a centroid which is basically the average of all the points in the cluster.

- The centroid (C_i) of the cluster (S_i) can be defined as

$$C_i = \frac{1}{|S_i|} \sum_{x_j \epsilon S_i} x_j$$

, where $|S_i|$ represents the number of points belonging to the i^{th} cluster.

- K-Means assign only one cluster to each point.
- In K-Means, we perform a certain number of iterations and in each iteration, we perform the following steps.
    - Every point is assigned to the cluster centroid closest to it.
    - Update the centroid.
    - Repeat the above two steps until convergence.

Improving centroids

- If after each iteration, the value of the Dunn Index **increases**, it is a sign that the clusters have become better after each iteration.

- The **objective** for the optimization of clusters can be represented as

$$L : \max_{c} \; dunn(x, argmin(c, x))$$

- The above equation can be read as
  - ➔ Take a point **x** and calculate its distance from all the **centroids,**
  - ➔ Choose the center having minimum distance from the point, that would be the cluster to which the point **x** would be assigned,
  - ➔ Using the cluster and the data point, calculate the Dunn index and maximize that over the centers.

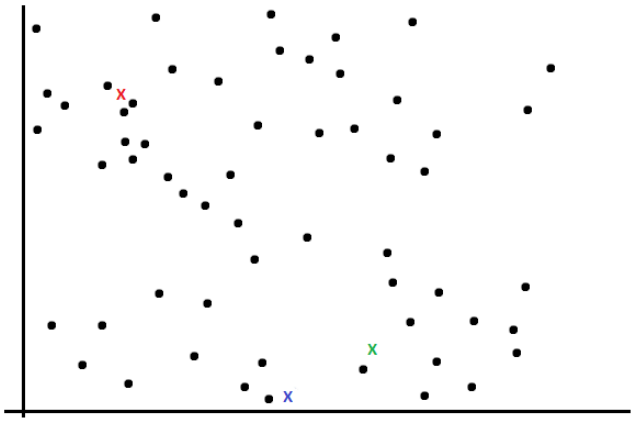
- But there are some **problems** while performing the above optimization steps:
  1. The objective function is not differentiable.
  2. Assigning a point **x** to a group is a discrete problem, and there is no good way to convert the expression into a function that is continuous and differentiable.

- Taking the above problems into consideration, we cannot use calculus or gradient descent for the purpose of optimization. Gradient descent can lead to fractional assignments.

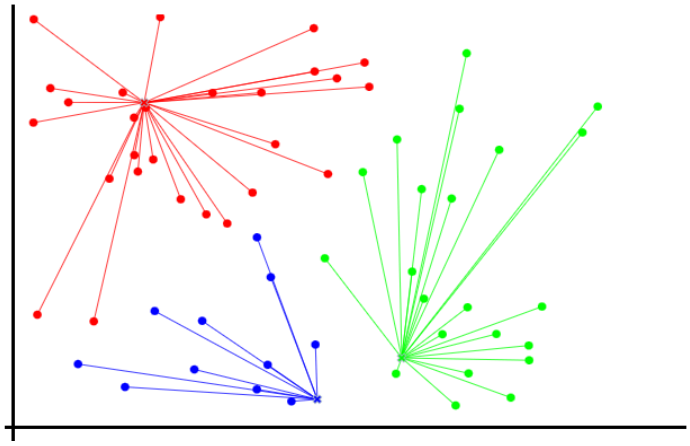
Lloyd's algorithm (K-means algorithm)

- This algorithm is used to cope with the problem of updating the centers.
- It has 4 basic steps:

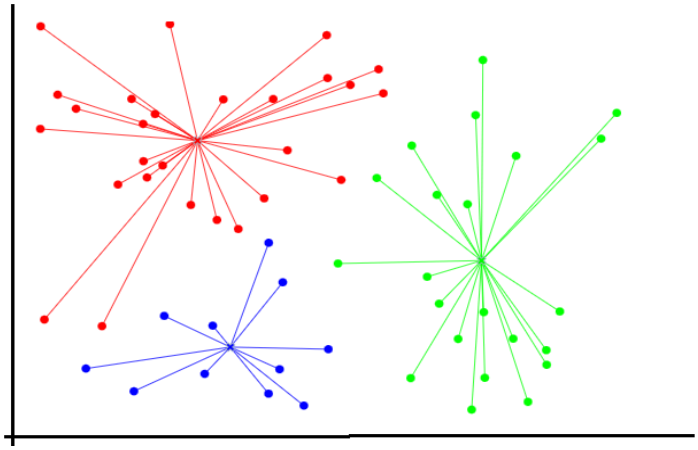
  - ➔ **Initialization**: Randomly initialize **k** centers from the dataset.



➔ **Assignment**: For each point, we find the distance of existing centroids from it and assign the point to that cluster whose centroid has the minimum distance.



➔ **Update** the centroids of the clusters by taking the average of points from each cluster.



➔ Repeat the previous two steps until convergence (the center of new cluster centroids stops changing their positions).

ANIMATION LINK: http://tech.nitoyon.com/en/blog/2013/11/07/k-means/

Within-cluster sum of squares (WCSS)

- The within-cluster sum of squares is a measure of the variability of the data points within each cluster. It is given as

$$wcss = \sum_{i=1}^{k} \sum_{j=1}^{m_i} (x_{ij} - c_i)^2$$

where $\mathbf{x_{ij}}$ is the $\mathbf{j^{th}}$ point belonging to the $\mathbf{i^{th}}$ cluster and $\mathbf{m_i}$ is the number of points in the $\mathbf{i^{th}}$ cluster.

- A **variation** of the above formula can be as follows:

$$wcss = \sum_{i=1}^{k} \sum_{j=1}^{m_i} d(x_{ij}, c_i)$$

where, $\mathbf{d(x_{ij}, c_i)}$ is representing a distance metric (any of euclidean, manhattan, etc.) that is calculating the distance between the point $\mathbf{x_{ij}}$ and the centroid $\mathbf{c_i}$ of the cluster.

Silhouette score

- The silhouette score is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).

$$S(x_i) = \frac{b - a}{max(b, a)}$$

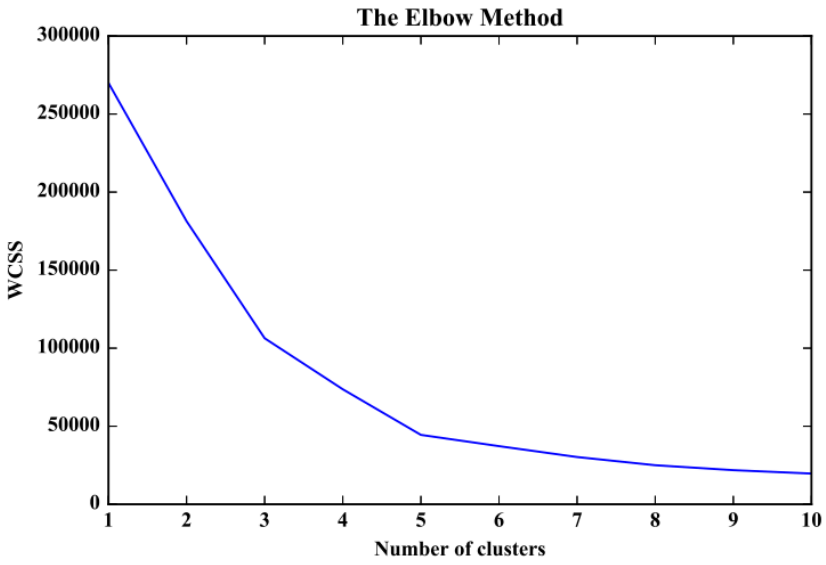
where **a** = average distance of point $\mathbf{x_i}$ from points in its own cluster and,

b = average distance of point $\mathbf{x_i}$ from all the points of the nearest cluster.

- The range of the Silhouette score is **[-1, 1]**.
  - ➔ A Silhouette score near +1 indicates that the sample is far away from its neighboring cluster.
  - ➔ A value near 0 represents overlapping clusters with samples very close to the decision boundary of the neighboring clusters.
  - ➔ A Silhouette score of -1 indicates that the samples have been assigned to the wrong clusters.

Elbow method

- It is a method to determine the optimal number of clusters (**k**) for k-means clustering.
- We perform the k-means clustering for a range of values of **k** and for each iteration, we calculate the value of the WCSS metric.
- When the value of WCSS is plotted against a range of **k** values, we get a plot looking like an elbow.



- We can clearly see that the WCSS value decreases as the number of clusters (k) increases.

- At some point on the graph, there is a sharp change in the slope (k = 5) after which the change in slope is very small. The k value corresponding to this point is the optimal K value or an **optimal** number of clusters.

- If we do not get a sharp change in the slope of the elbow plot while using the WCSS metric on the y-axis, we can try using the **Silhouette score** to get significant results or to get confidence in our decision.