K-Means Clustering

Target To understand the K-Means clustering and ways to find the best value of k.

What is K-Means Clustering

Let's say someone asks you to show all toys you use to play or their equipment and you start showing it

Let's say you are out somewhere at any destination with your complete family. Your father asks "who wants to eat strawberry ice cream

In a bucket, you have a different color of marble and you want to put similar marble in one bucket and another bucket. So you start separating them based on their size, color, and other parameters. Now you have 5 different buckets of marble and each bucket contains similar to marble. So, the buckets are known as clusters and the process of separation and creating buckets is known as clustering.

In our case, we have 5 buckets that are `k=5`.

K-means clustering is an Unsupervised machine learning algorithm.

Definition Clustering is a process of separating the data points into a number of clusters (groups), in such a way that data points in the same cluster are more similar to other data points in the same group and dissimilar to another cluster (groups).

How K-Means Clustering works??

- 1. It picks any random points(k points) from data as cluster center, that points also known as **Centroids**.
- 2. Start assigning each data point to the nearest centroid.
- 3. Recalculate the centroid of each cluster based on the mean of the data points assigned to it.
- 4. Repeat steps 2 and 3 until the cluster assignment no longer changes or a maximum number of iterations is reached.

As Centroind move, the calculation of **Squared Euclidean Distance** is used to measure the similarity between the sample points and centroids.

Ways to find the Best value for K

There are two main methods for calculating the optimal value of K(optimal number of clusters).

Elbow Curve Method

- It is a graphical technique for determining the optimal number of clusters in a dataset.
- working of method: as the number of clusters increases, the sum of the squared distance between the data points and their assigned cluster centroids(aka within-cluster sum of square or inertia) decreases, since each cluster is more specialized and compact.

- 1. Initialize the value of k(number of the cluster). A good range for k is from 10 to 0.
- 2. Now compute the **within-cluster sum of the square(aka inertia)**. A within-cluster sum of squares is defined as the `sum of squared distances between each data point and its assigned cluster centroid`. This measure the compactness of the cluster.
- 3. Plot the within-cluster sum of the square as a function of a number of clusters (k).
- 4. Identify the elbow point visually, the value of K at which the rate of decrease in the within-cluster sum of square start level-off.
- 5. Now choose the optimal number of clusters based on the elbow point.

code

```
x = iris_df.iloc[:, [0, 1, 2, 3]].values

from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++',
                    max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(x)
    wcss.append(kmeans.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('The elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS') # Within cluster sum of squares
plt.show()
```