K-Nearest Neighbor

Klasifikasi K-Nearest Neighbor (KNN) is first developed by Evelyn Fix and Joseph Holder in 1951 and later explained by Thomas Cover.

KNN is an Supervised Machine Learning Technique Algorithm used for both Classification and Regression Problems.

It uses existing data points (i.e dataset) and classify new data points with respect to similarity and features of existing data points.

One of Real World Scenario of KNN is “when you do shopping in online sites and selects a product after that it will get similar kind of options where your items is being selected”.

KNN will be used in some form of “find items similar to this one”.

It is a Non-Parametric Algorithm

It finds pattern recognition, data mining and intrusion detection.

## Process of KNN

* Before assigning the k value to the data, algorithm is used to handle the data and preprocess it before generating the training and testing.

Then, Assigning the k-value is important. After assigning the k value algorithm starts to select the datapoints from the existing data points with respect to Similarity and Features.

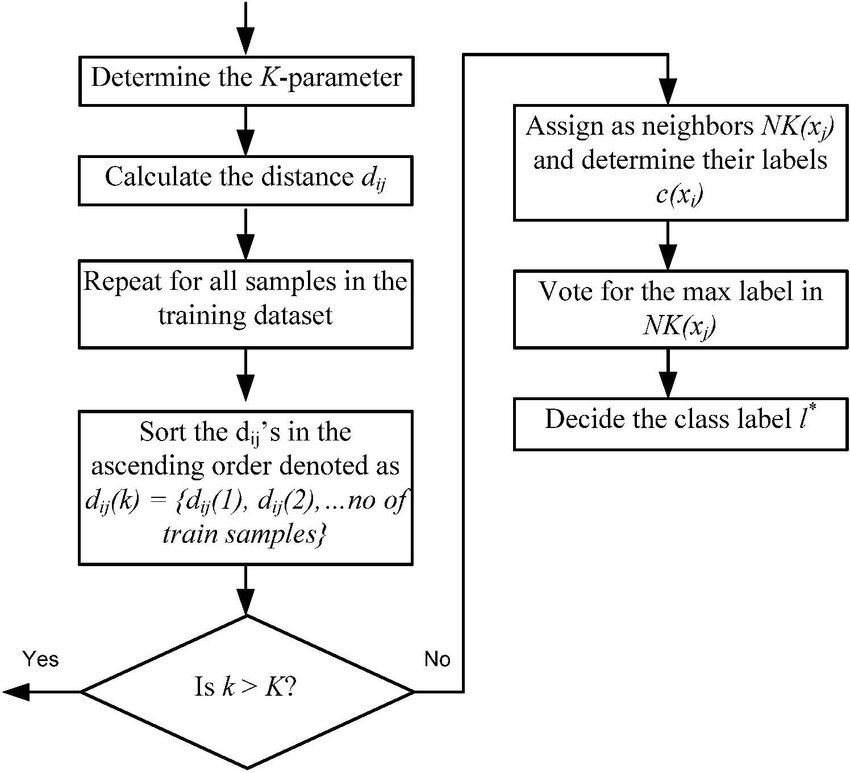
### Note

* A small value of k means that noise will have higher influence on the result.

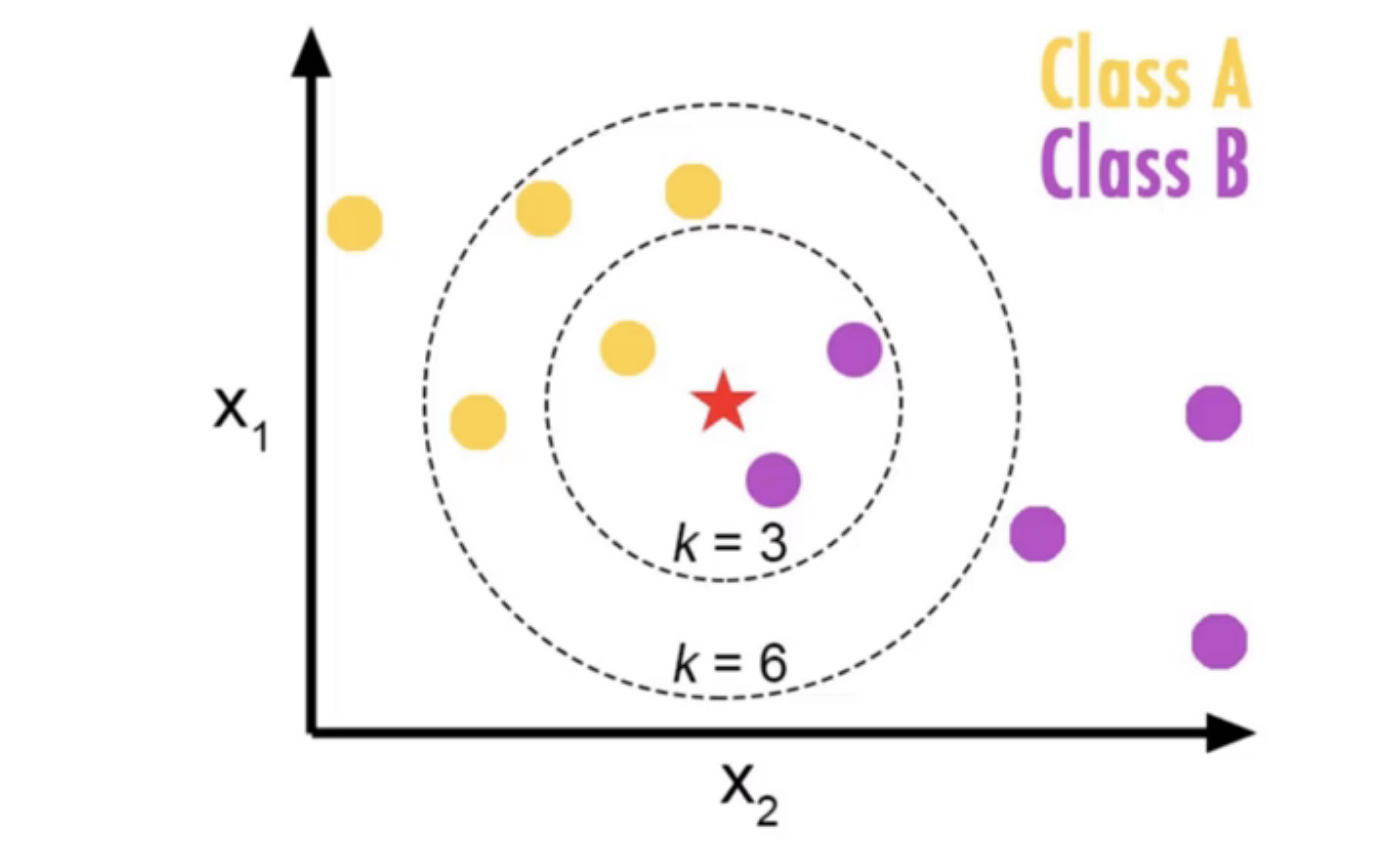
A large value of k means it is expensive and defeats the basic philosophy of KNN.

Simple Approach to select k is assign k=n^(1/2) where n={1,2,3,4,5,…..}

KNN is a lazy-learning algorithm. Its purpose is to use a database in which the datapoint is separated into several classes to predict the classification of a new sample Point.



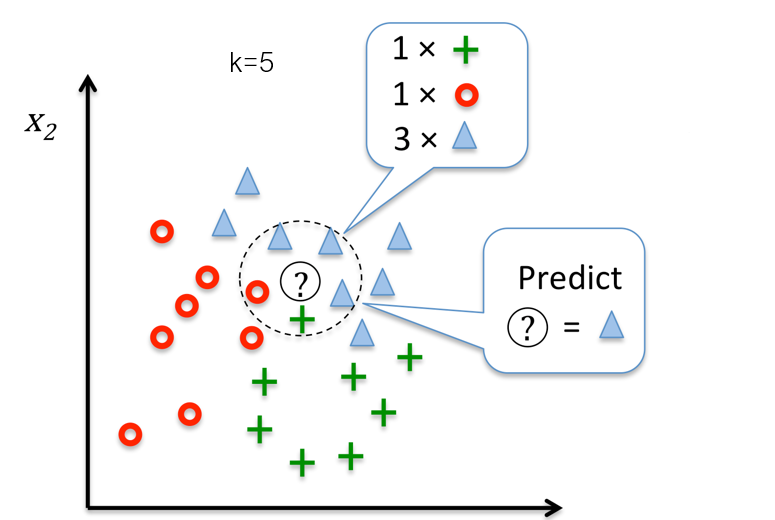
From the above fig, we can say that first determine the k-parameter, then calculate the distances with different types of techniques like Euclidean or Mahalanobis Distances,then repeat with all the samples and sort the data in ascending order.



When we plot the data points in the mathematical Space, based on the data ,data records are classified and first set the k value as 3,when k value set to 3,algorithm will search 3 data points near the datapoints i.e from the above fig we can see there are 3 categories and target class is star class.

KNN algorithm will check the classes near to the star class and find the majority of the classes.

Once we get the k nearest neighbors the new data point that is star datapoint will be use the class labels nearest neighbors to determine the class labels of the unknown record/data.



The above fig is the best example that, how a K-Nearest Neighbor Algorithm works.

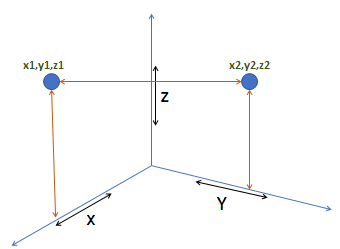
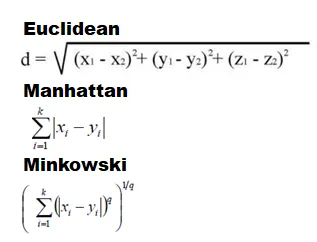
In K-nearest Neighbor algorithm there are many types of distance methods in that we discuss about 3 types

* Euclidean Distance
* Mahantan
* Minikowski

Euclidean Distance is the of the distance measuring technique which is used to measure the true straight line. In KNN it is used to measure the distance between a new datapoint and existing datapoint.

Mahantan distance is also know as taxicab distance or city cab distance. It is also one of the distance measuring technique used to measure “distance between two data points is sum of absolute differences of their cartesian coordinates.

Minikowski is the generalization of both Euclidean and Mahantan Distances.It is a metric used to measure the distance between two data points in the normed vector space.



## Practical Implementation of KNN using Python

### Lets code

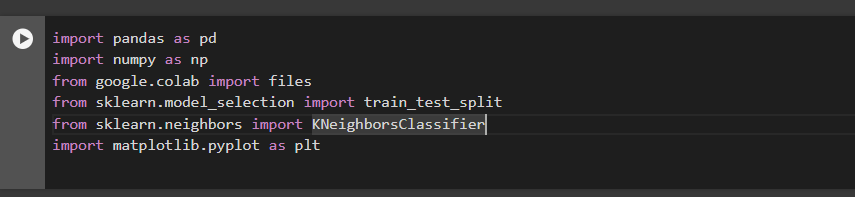
KNN refers to Supervised machine learning

To process the KNN algorithm first need to import necessary libraries and read the data set in and preprocess the data.

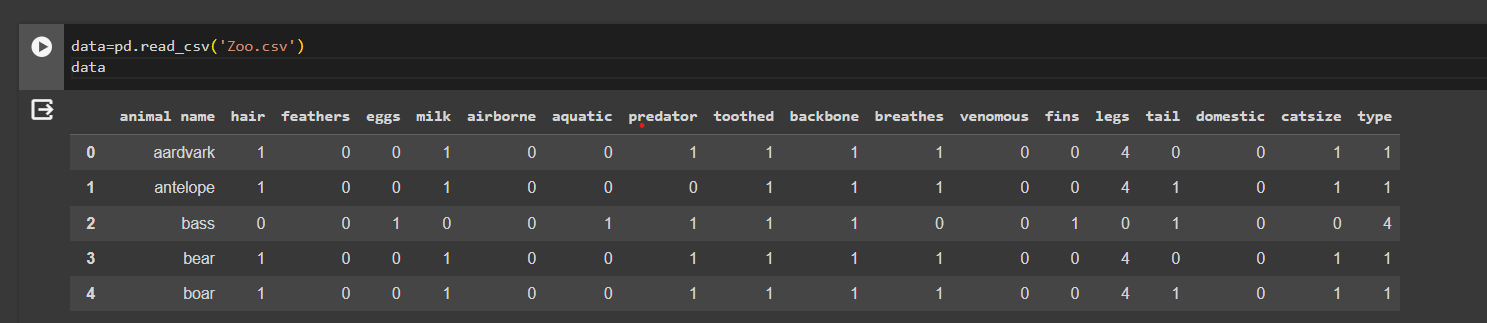
Before implementation of KNN Algorithm first the data should be splitted into dependent and independent variables and should split that dependent and target variables into training and testing data.

Then select the k and set the value of k in range of whole numbers and implement the KNN algorithm and fit the training and testing data in the KNN algorithm validate the model.

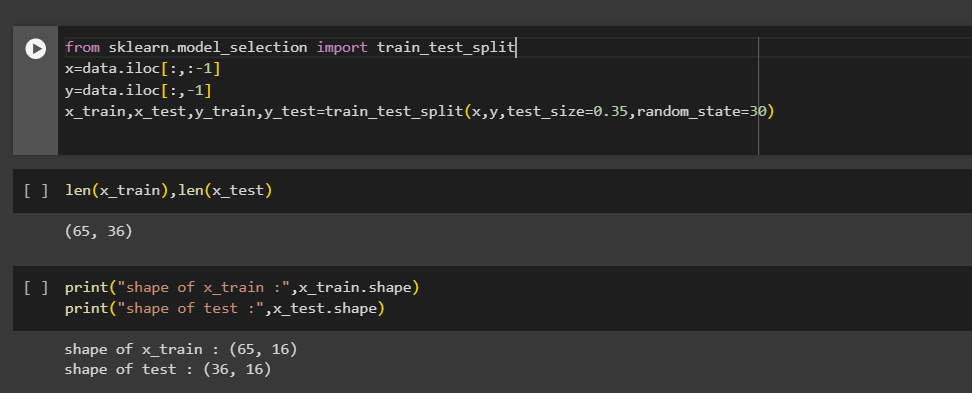
* First step is to Implement the necessary Libraries to perform KNN Algorithm.



* Second step is to load the necessary dataset

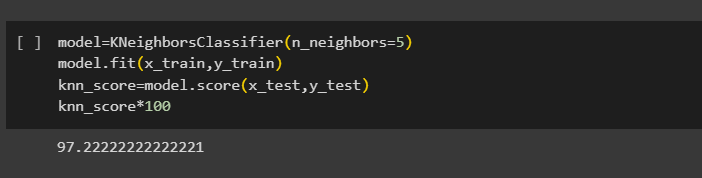


* Third step is to clean, normalize and split the data into train and test split



Build the model with different k values to check the optimal value.

Check the miss classification errors vs k for cross validation.



Full python is [here](https://github.com/kiranpasunuti/K-NearestNeighbor)

Thankyou..