K-Nearest Neighbor Algorithm

M. Shahidur Rahman

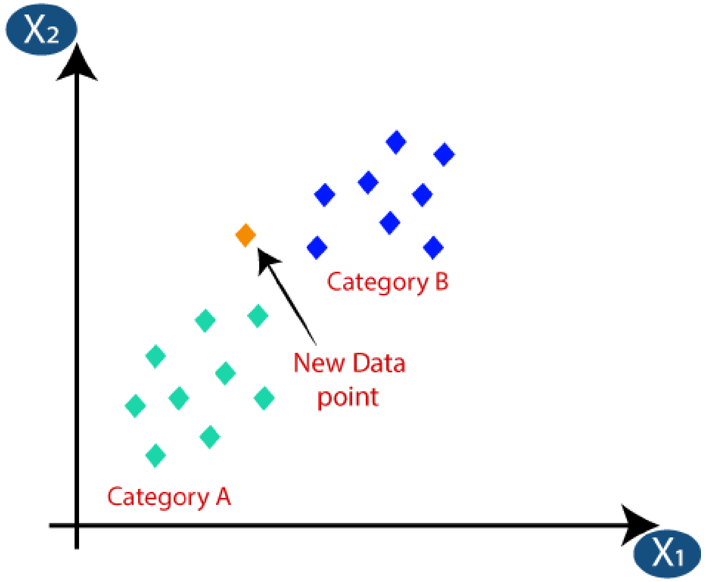
Professor, CSE, SUST

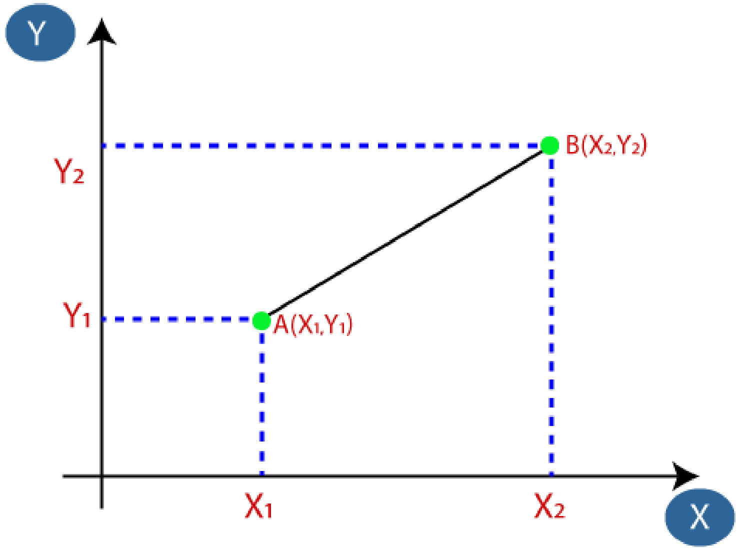
K-Nearest Neighbor (KNN) Algorithm

- K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on **Supervised Learning** technique.
- K-NN algorithm assumes the similarity between the new data and available data and put the new data into the category that is most similar to the available categories.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

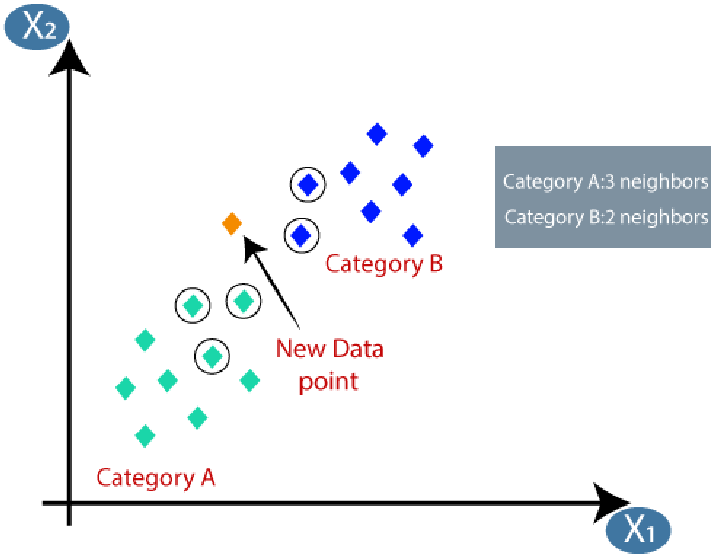
How does K-NN work?

- Suppose we have a new data point and we need to put it in the required category.





Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

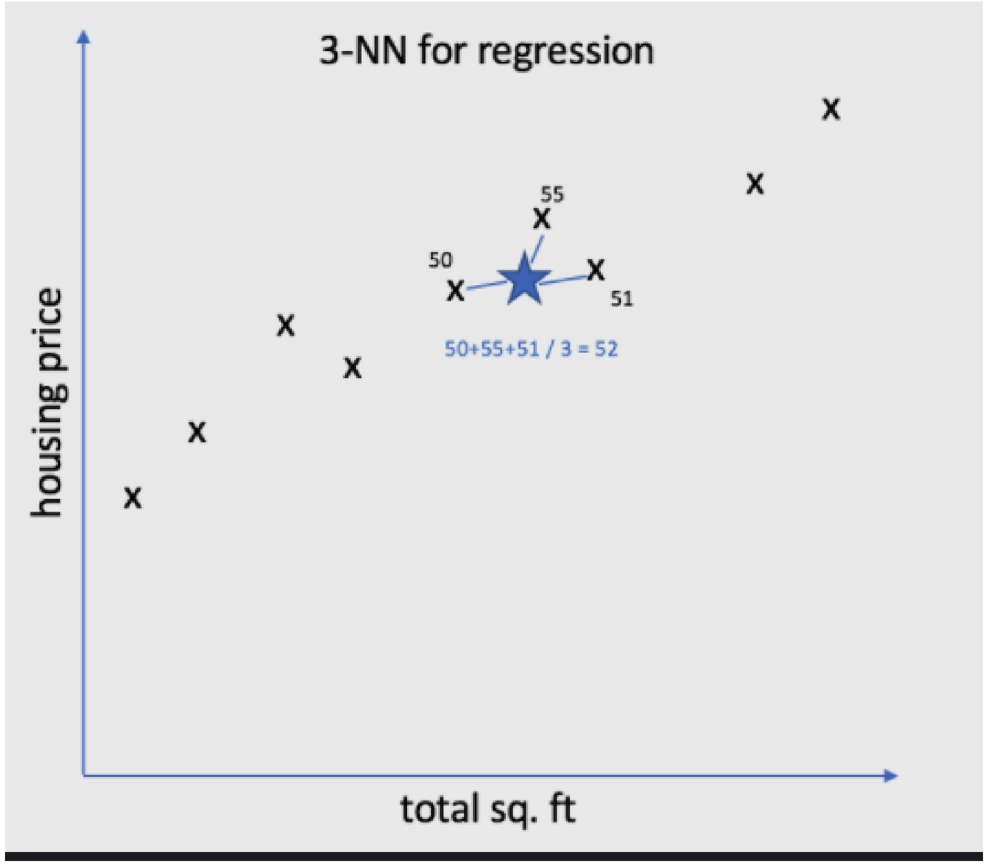


How to select the value of K?

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- For a larger value of K, underfitting occurs. In this case, the model would be unable to correctly learn on the training data.

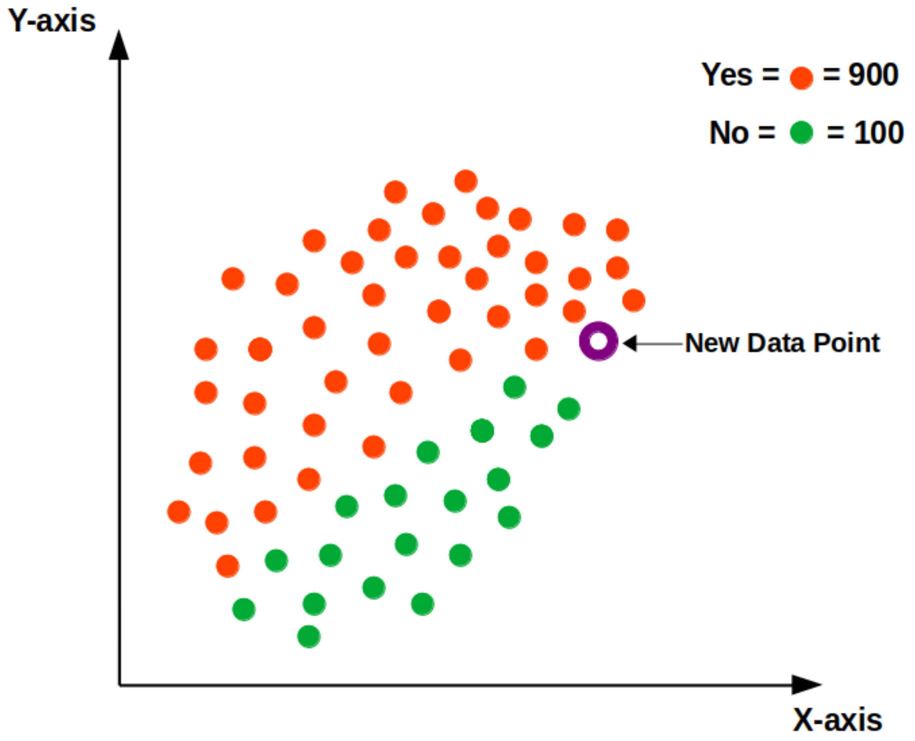
How does KNN work for ‘Regression’ problem?

- KNN employs an average method for predicting the value of new data. Based on the value of K, it would consider all of the nearest neighbors.



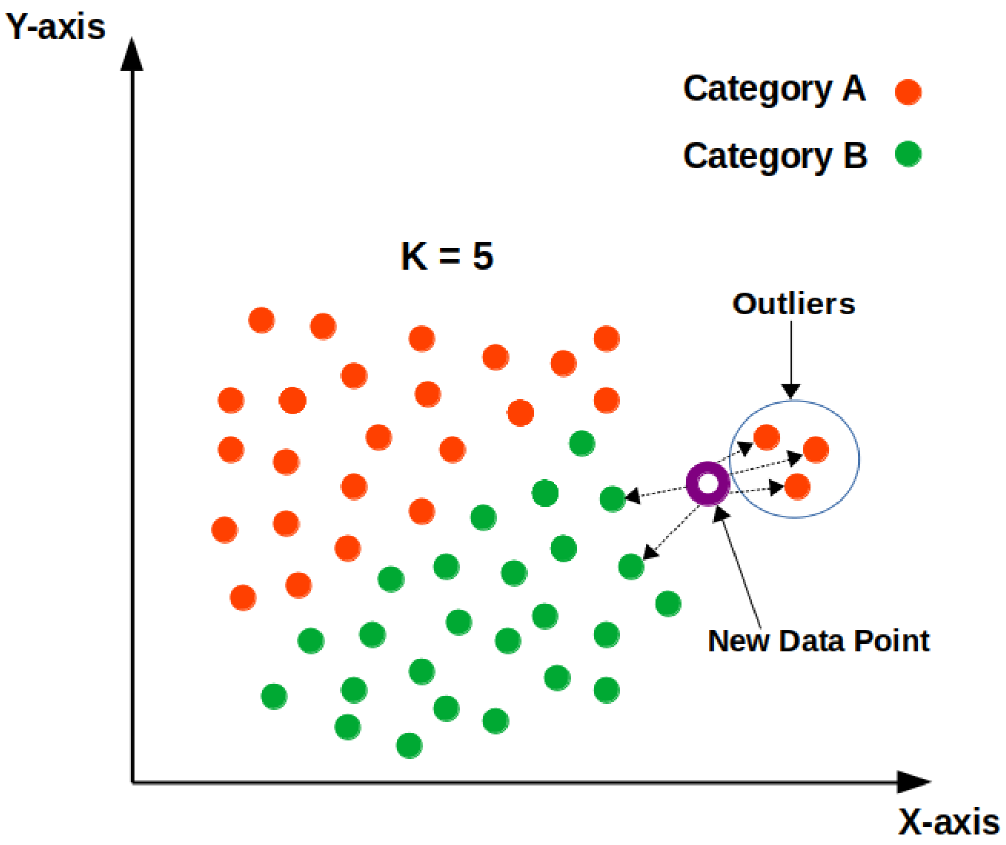
Impact of Imbalanced dataset on KNN

- When dealing with an imbalanced data set, the model will become biased
- The bulk of the closest neighbors to this new point will be from the dominant class.
- Because of this, we must balance data set using either an “Upscaling” or “Downscaling” strategy.

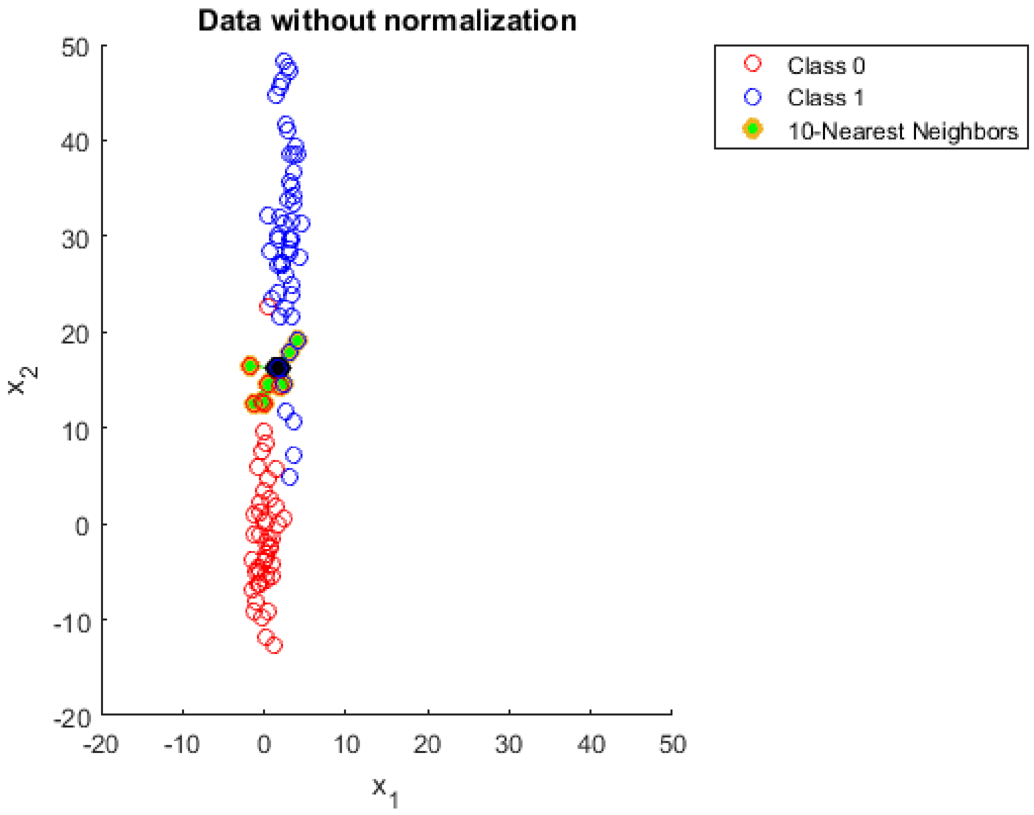


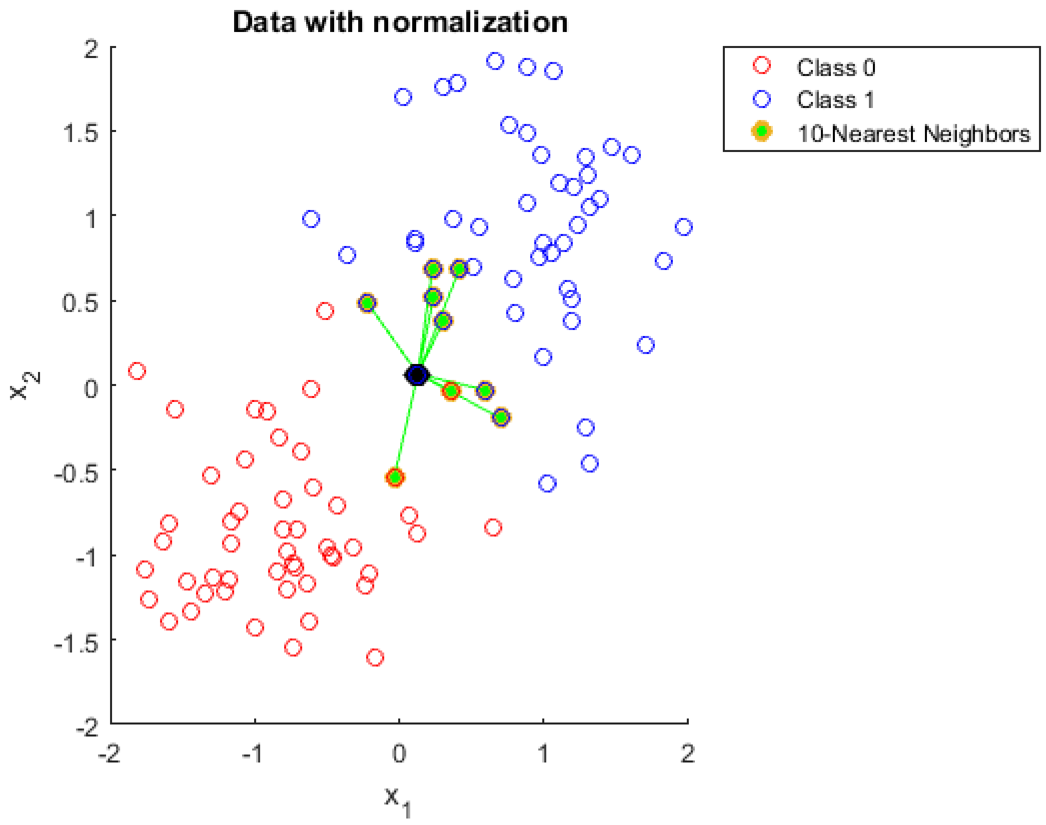
Impact of Outliers on KNN

- Outliers are the points that differ significantly from the rest of the data points.
- The appropriate class for the new data point should be "Category B" in green.
- The model, however, would be unable to have the appropriate classification due to the existence of outliers.
- Removing outliers before using KNN is recommended.



Importance of scaling down the numeric variables to the same level





Ways to calculate the distance in KNN

The distance can be calculated using different ways including

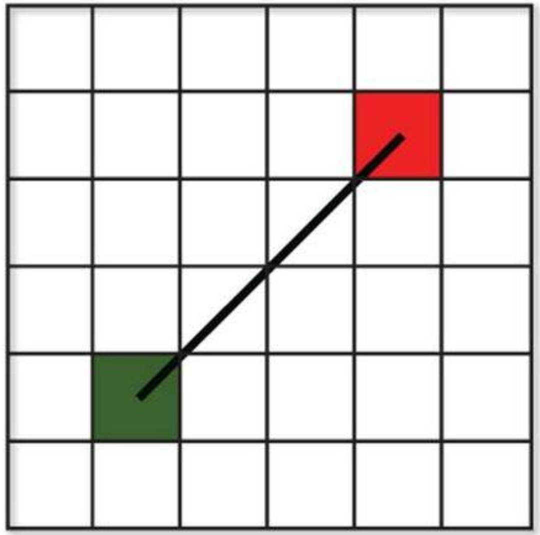
- Euclidean Method (most popular one)
- Manhattan Method
- Minkowski Method

Euclidean Distance

- It is a measure of the true straight line distance between two points in Euclidean space.
- This distance is the most widely used one as it is the default metric that SKlearn library of Python uses for K-Nearest Neighbor.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

- The distance using Euclidean distance metric of red (4,4) and the green (1,1) is 4.24.



Euclidean Distance

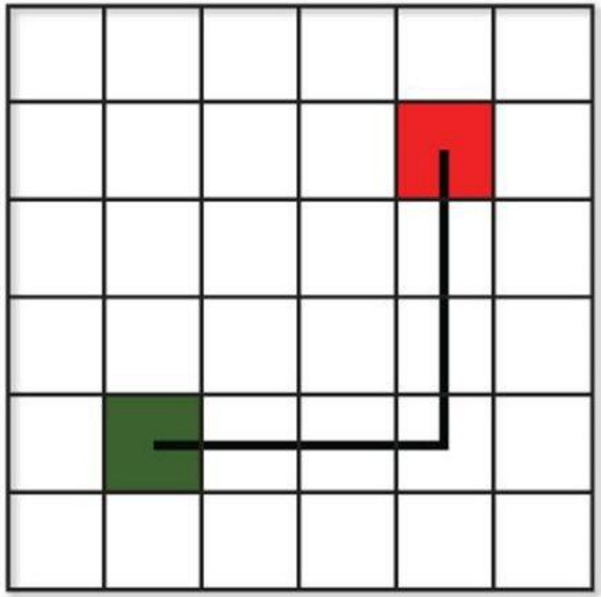
Manhattan Distance

- This distance is also known as taxicab distance or city block distance, that is because the way this distance is calculated.
- The distance between two points is the sum of the absolute differences of their Cartesian coordinates.

$$d=\sum_{i=1}^{n} |x_i - y_i|$$

- The distance between the red(4,4) and the green(1,1) points using Manhattan distance metric is

  d = |4-1| + |4-1| = 6
- This distance is preferred over Euclidean distance when we have a case of high dimensionality.



Manhattan Distance

Minkowski Distance

- It is a metric intended for real-valued vector spaces.
- There are a few conditions that the distance metric must satisfy:
  - i. Non-negativity: d(x, y) >= 0
  - ii. Identity: d(x, y) = 0 if and only if x == y
  - iii. Symmetry: d(x, y) = d(y, x)
  - iv. Triangle Inequality: d(x, y) + d(y, z) >= d(x, z)

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

- The p value in the formula can be manipulated to give us different distances like:
  - p = 1, when p is set to 1 we get Manhattan distance
  - p = 2, when p is set to 2 we get Euclidean distance

Pros and Cons of KNN algorithm

- Benefits of using KNN algorithm
  - KNN algorithm is widely used for different kinds of learnings because of its easy to apply nature.
  - There are only two metrics to provide in the algorithm. value of k and distance metric.
  - Work with any number of classes not just binary classifiers.
- Disadvantages of KNN Algorithm
  - Always needs to determine the value of K which may be complex some time.
  - The computation cost is high because of calculating the distance between the data points for all the training samples.