Performance of KNN Algorithm on Woman’s Fertility Dataset

Shrikant patro,

School of Computer Science and Engineering, VIT Chennai, Tamil Nadu, India 600127

Email:shrikantjagannath.2018@vitstudent.ac.in

Abstract

KNN stands for the K nearest neighbor. It is a supervised learning algorithm. It is a simple algorithm that store all the available cases and classifies new cases based on similarity measure. The newly introduced data point without class label, it will predict the class label for the classless data point. This paper will introduce the dataset of woman’s fertility from UCI repository and will propose the mechanism of KNN. The performance of the KNN using on data point whether it is donner or not based on the features of the data point.

Introduction

KNN can be used for both classification and regression. However, it is widely used in classification of problems in the industry. KNN algorithm generally fairs across all parameters of considerations. It is used for ease of interpretation and low cost calculation time. It is typically called as lazy learner. It’s doesn’t learn from the dataset. It is doesn’t learn from the training data but it memorizes the training dataset instead. Let’s understand it’s working from the dataset.

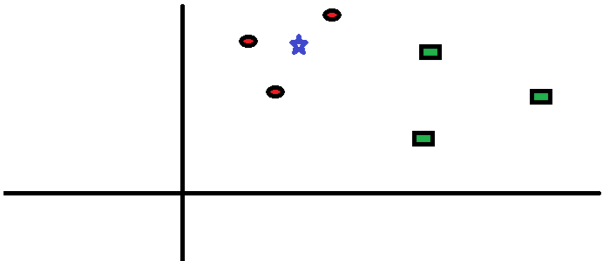

Fig 1 Sample data point

We intend to find out the class of the blue star (BS). BS can either be RC or GS and nothing else. The “K” is KNN algorithm is the nearest neighbors we wish to take vote from. Let’s say K = 3. Hence, we will now make a circle with BS as center just as big as to enclose only three data points on the plane. Refer to following diagram for more details

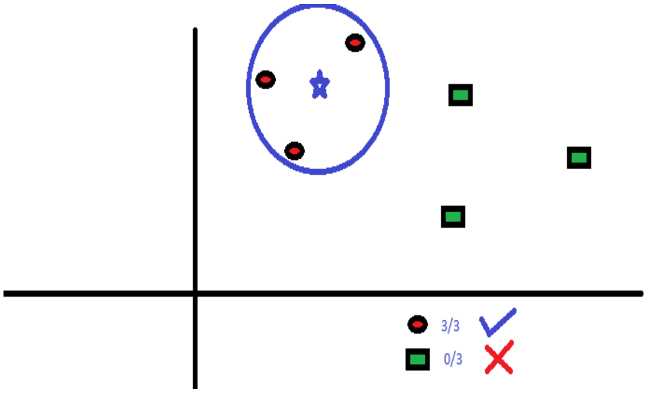

Fig 2 Identify the class label for the data point

Performance of the KNN algorithm changes with the selection of the K. The boundary between the selection of the K will decide whether the division will be smoother or not. As the value of K increases the boundary become smoother. Here, the overfitting takes place at the k=1 because the training error rate is minimum and validation error rate is maximum. This situation stabilizes with slight increase in training error rate and decrease in validation error rate. This situation is observed during the K=10, when the validation error rate comes down and increase in training phase is small.

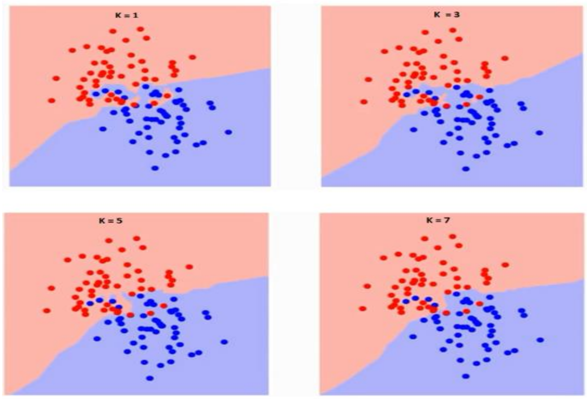

Fig 3 Change in value of k

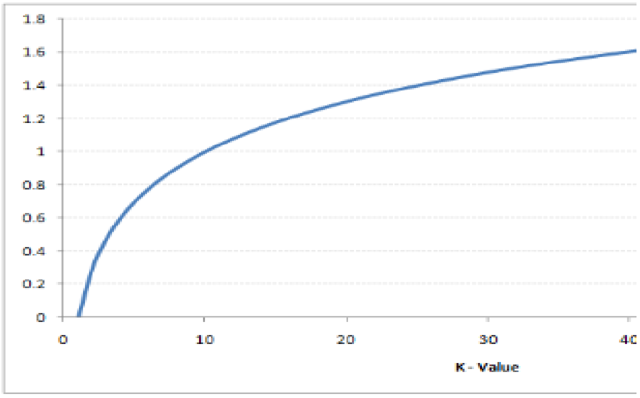

Fig 4. Training error

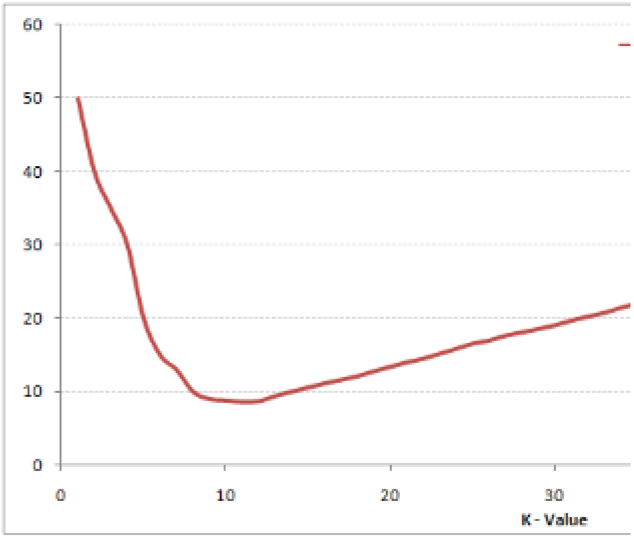

Fig 5. Validation error

Methodology

The KNN algorithm steps include the following steps:

1. Choose the number of k and a distance metric.
2. Find k nearest neighbor of the given data point.
3. Assign the class label by majority vote.

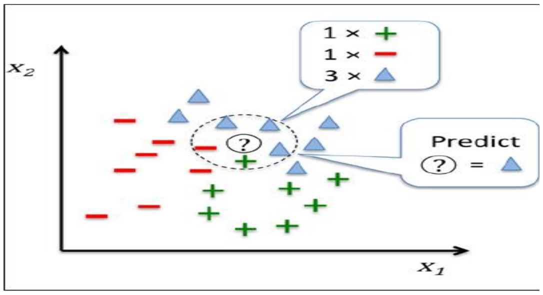

Fig 6 KNN Demonstration

Based on the chosen distance metric, the KNN algorithm finds the *k* samples in the training dataset that are closest (most similar) to the point that we want to classify. The class label of the new data point is then determined by a majority vote among its *k* nearest neighbors.

The main advantage of such a memory-based approach is that the classifier immediately adapts as we collect new training data. However, the downside is that the computational complexity for classifying new samples grows linearly with the number of samples in the training dataset in the worst-case scenario—unless the dataset has very few dimensions (features) and the algorithm has been implemented using efficient data structures such as KD-trees.

Experiment

The implementation of KNN model in scikit learn using the Euclidean distance matric. Import KNeighborsClassifier from sklearn.neighbors. use n_neighbor to specify the value of k, use p for the distance measure and metric=’minkowaski’.

Conclusion

This is very simple algorithm to identify the class label for the data point. The maximum vote and value of k, is principle component for identification of class label of the data point.