**K Nearest Neighbor – KNN Algorithm**

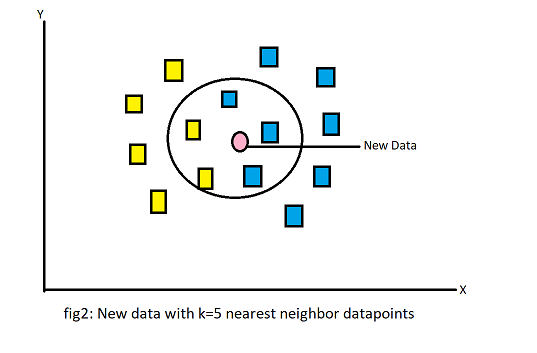
Bibek Lamsal

CSE student from BVC Engineering College, Amalapuram, Andhra Pradesh

**Abstract:** This paper proposes the introduction to the K Nearest Neighbor (KNN) algorithm and its Applications. It is a basically classification problem.KNN Algorithm currently provide the best solutions to many classification problems like Face Recognition, Recommendation Systems (Amazon, Netflix), Text Mining, Diabetes in patients.

**1. Introduction:**

KNN algorithm is a powerful, top text mining classification algorithm.  KNN is a supervised learning algorithm used both as a classification and regression. In this article we will cover the classification side of things for now. KNN algorithm uses simple technique to classify the problem like how the data point is similar to its neighbour datapoints.

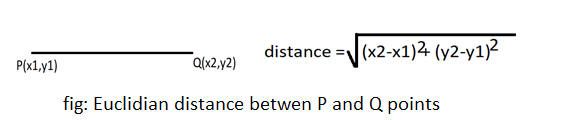
 

Let’s us take one example from figure 1, Where two different types of categories are there like yellow datapoints and blue datapoints, the pink dataset is new and this pink data point is whether yellow or blue point.

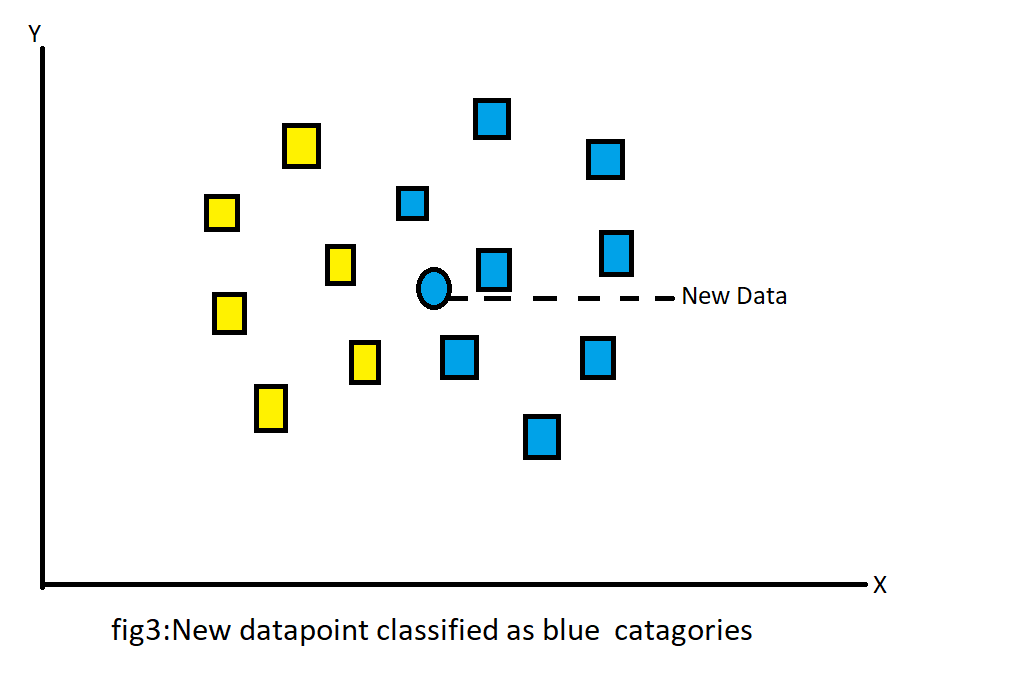
How many nearest neighbours will be required is totally based on the K-Value. If we specified the value of K is 1 means only one nearest neighbour. In this there are five neighbours(k) and based on k value which is in figure 2.

The nearest neighbour can be calculated using the Euclidian distance or Manhattan distance.

Euclidian distance between two data point P(x1,y1) and Q(x2,y2) is



Based on this Euclidian distance we find the best five nearest datapoints and on the majority number of the similar type of neighbours the data point will be classify. From the above example there are 3 blue data points and 2 yellow data points and the majority is the 3 blue data points. So, the new data points are classified as the blue data point.



**2. Steps involves in KNN Algorithm:**

This are the different steps we have to follow to do the KNN model predictions.

Step 1: Import the modules and dataset

Step 2: Process the data

Step 3: Create the model

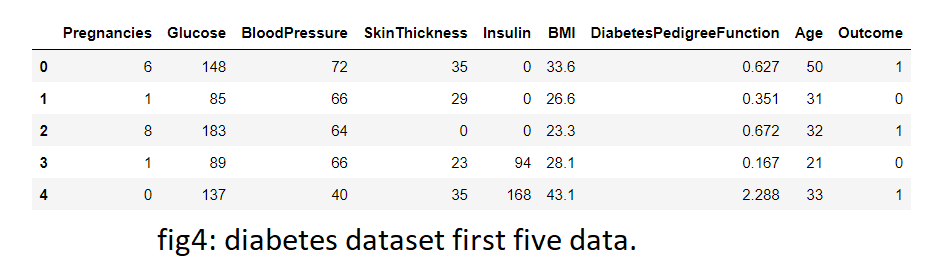
Step 4: Train the model

Step 5: Evaluate the model

Let’s take one dataset and implement KNN stepwise.

**3. Classification of Diabetes Patient Using KNN:**

For this classification we need the diabetes dataset where some of the patient have this problem and some of them haven’t. We have to classify the new patient based on his/her conditions. The dataset looks like



Based on the given characteristics of the patient we can classify them as a diabetes patient or not.

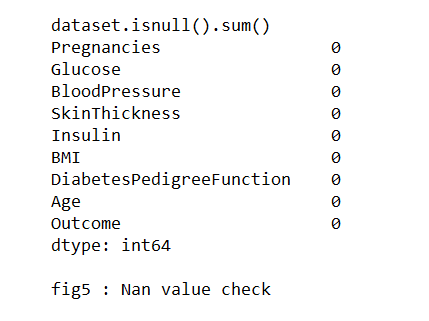
### Step 1: Importing dataset

This dataset is available in the Kaggle site with the name of [diabetes.csv](https://storage.googleapis.com/kaggle-data-sets/818300/1400440/bundle/archive.zip?X-Goog-Algorithm=GOOG4-RSA-SHA256&X-Goog-Credential=gcp-kaggle-com%40kaggle-161607.iam.gserviceaccount.com%2F20210606%2Fauto%2Fstorage%2Fgoog4_request&X-Goog-Date=20210606T090953Z&X-Goog-Expires=259199&X-Goog-SignedHeaders=host&X-Goog-Signature=) dataset.

**dataset = pd.read\_csv(“diabetes.csv”)**

Step 2: Process dataset

Checking the empty or nan values present in the dataset.



Feature scaling is the most important factor in any type of the regression and classification problem dataset. It helps the gradient descent to reach the global minimum in less time compare to without feature scaling and also helps to optimize the problem.

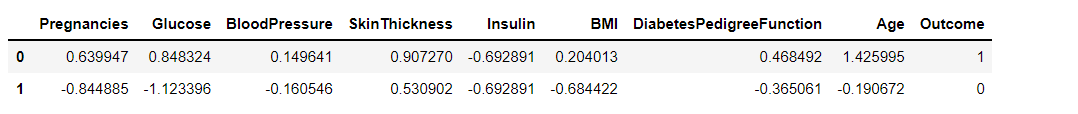
**from sklearn.preprocessing import StandardScaler**

**scale\_data = dataset.columns[:-1]**

**sc = StandardScaler()**

**dataset[scale\_data] = pd.DataFrame(sc.fit\_transform(dataset.iloc[:,:1]),columns=scale\_data)**

**dataset.head(2)**



Now splitting the dataset into the dependent and independent features

**X = dataset.iloc[:,:-1]**

**Y = dataset.iloc[:,-1]**

**X.shape, Y.shape**

**((768, 8), (768,))**

Again, split the dependent and independent features into the training and testing dataset.

**from sklearn.model\_selection import train\_test\_split**

**x\_train,x\_test,y\_train,y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state=0)**

**x\_train.shape,x\_test.shape,y\_train.shape,y\_test.shape**

**((614, 8), (154, 8), (614,), (154,))**

Step 3: Create the model:

**from sklearn.neighbors import KNeighborsClassifier**

**kn = KNeighborsClassifier(n\_neighbors = 1)**

Step 4: Train the model:

kn.fit(x\_train,y\_train)

Step 5: Evaluate the model:

**kn.score(x\_test,y\_test) #testing on the test dataset**

**0.8051948051948052**

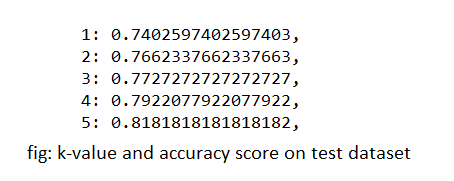
**kn.score(x\_train,y\_train)**

**0.8127035830618893**

**4. Optimization:**

Optimization refers to hyper parameter tuning in KNN Algorithm. Hyper parameter tuning is the process finding the best value of K (K number of nearest neighbors).Lets find out

The best value of k in the above diabetes problem.



Based on the test sample we can tune up to 0.81% accuracy score from the given dataset.

**5. Summary:**

K-Nearest Neighbours is a [machine learning](https://www.unite.ai/what-is-machine-learning/) technique and algorithm that can be used for both classification and regression problem. [K-Nearest Neighbours](https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/) examines the labels of a chosen number of data points surrounding a target data point, in order to make a prediction about the class that the data point falls into. K-Nearest Neighbours (KNN) is a conceptually simple yet very powerful algorithm, and for those reasons, it’s one of the most popular machine learning algorithms. It does not make any model like the linear and SVM regression algorithm. So, it is the lazy type of algorithm. It can be very useful when we have to work with the small dataset.

When you use KNN, you must ensure that you have to check the various values of K in order to find the number that provides the highest accuracy.

**GitHub Link :** <https://github.com/bibeklamsa1/Python-Machine-Learning-And-AI/tree/main/Introduction%20and%20Implementation_KNN_Algorithm>

**Reference:**

<https://medium.com/@arman_hussain786/k-nearest-neighbors-knn-and-its-applications-7891a4a916c6>

<https://www.youtube.com/watch?v=4HKqjENq9OU>

<https://www.educba.com/knn-algorithm/>

<https://www.unite.ai/what-is-k-nearest-neighbors/>