Graphic Era
Deemed to be University

Project Report

Machine Learning in HealthCare [Diabetes Prediction]

Submitted by: Rohan Tripathi
Univ. Roll. No: 2015268
Section: ML

UNDER THE GUIDANCE OF
:

ACKNOWLEDGEMENT

I am extremely thankful to my Honorable President sir Prof. (Dr.) Kamal Ghanshala of GRAPHIC ERA DEEMED TO BE UNIVERSITY for supplying me the opportunity to work on this project. I would also like to thank our head of department, Dr Devesh P. Singh and all faculties of Computer Science and Engineering for their valuable guidance, as well as for the educational and infrastructural support. I would like to express my sincere gratitude and respect to my project guide Mr. Kireet Joshi for his stimulating guidance and continuous supervision, monitoring and constant encouragement throughout the project completion. The blessing, help and guidance given by him time to time shall help me a long way in the journey of life on which I am about to embark. I am obliged to my project team members for the valuable information provided by then in their respective fields. I am grateful for everyone’s cooperation during the period of this project assignment.

ABSTRACT

- Diabetes is an illness caused because of high glucose level in a human body. Diabetes should not be ignored if it is untreated then Diabetes may cause some major issues in a person like: heart related problems, kidney problem, blood pressure, eye damage and it can also affects other organs of human body.
- Diabetes can be controlled if it is predicted earlier. To achieve this goal this project work we will do early prediction of Diabetes in a human body or a patient for a higher accuracy through applying, Various Machine Learning Techniques.

- Machine learning techniques Provide better result for prediction by con- structing models from datasets collected from patients. In this work we will use Machine Learning Classification and ensemble techniques on a dataset to predict diabetes. Which are K Neighbors classifier , Logistic Regression (LR), Random Forest (RF). The accuracy is different for every model when compared to other models. The Project work gives the accurate or higher accuracy model shows that the model is capable of predicting diabetes effectively.

1.1. INTRODUCTION

Diabetes is noxious diseases in the world. Diabetes caused because of obesity or high blood glucose level, and so forth. It affects the hormone insulin, resulting in abnormal metabolism of crabs and improves level of sugar in the blood. Diabetes occurs when body does not make enough insulin. According to (WHO) World Health Organization about 422 million people suffering from diabetes particu- larly from low or idle income countries. And this could be increased to 490 billion up to the year of 2030. However prevalence of diabetes is found among various Countries like Canada, China, and India etc.

Population of India is now more than 100 million so the actual number of diabet- ics in India is 40 million. Diabetes is major cause of death in the world. Early prediction of disease like diabetes can be controlled and save the human life. To accomplish this, this work explores prediction of diabetes by taking various attributes related to diabetes disease.

Diabetes mellitus is one of the non-communicable diseases that pose a threat to human health. It has become a major global health problem. It is a chronic disease that occurs either when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin which it produces. It is found that diabetes causes blindness, amputation and kidney failure. Lack of awareness about diabetes, insufficient access to health services and essential medicines can lead to the above mentioned complications. According to a study by the World Health Organization (WHO), number of diabetic patients will raise to 552 million by 2030, which means that one in 10 adults will have diabetes by 2030. In 2014, the global prevalence of diabetes was estimated to be 9 % among adults aged 18+ years . WHO insisted with an alarm that Diabetes is the 7th leading cause of death in the world. In 2012, an estimated 1.5 million deaths were directly caused by diabetes. Total deaths due to diabetes are projected to rise by more than 50 % in the next 10 years. Moreover, the International Diabetes Federation said that nearly 52 % of Indians are not aware that they are suffering from high blood sugar. More than 62 million diabetic individuals are currently diagnosed with the disease. It is predicted that, by 2030 diabetes mellitus may affect up to 79.4 million individuals in India. A nation-wide study, conducted by the Indian Council of Medical Research`s INDIAB (India Diabetes) has confirmed that one out of 10 people in Tamil Nadu is affected by diabetes, and every two persons with age group of 25 are in the pre-diabetic stage. It is stated that 14.8 per cent of urban population and 11 per cent of rural population of Tamil Nadu are affected by diabetes. Madras Diabetes Research Foundation suggested that about 42 lakh individuals have diabetes and 30 lakh people are in pre-diabetes stage. At least, 1,000 people avail treatment for diabetes out of the 12,000 outpatients who visit Rajiv Gandhi Government General Hospital (RGGGH).

For this purpose we apply various Machine Learning classification and ensemble Techniques to predict diabetes.

Machine Learning Is a method that is used to train computers or machines explicitly. Various Machine Learning Techniques provide efficient result to collect Knowledge by building various classification and ensemble models from collected dataset. Such collected data can be useful to predict diabetes.

Various techniques of Machine Learning can capable to do prediction, however its tough to choose best technique. Thus for this purpose we apply popular classification and ensemble methods on dataset for prediction.

1.2. BACKGROUND OF PROJECT

Diabetes: Its Beginnings

The first known mention of diabetes symptoms was in 1552 B.C., when Hesy-Ra, an Egyptian physician, documented frequent urination as a symptom of a mysterious disease that also caused emaciation. Also around this time, ancient healers noted that ants seemed to be attracted to the urine of people who had this disease.

In 150 AD, the Greek physician Arateus described what we now call diabetes as "the melting down of flesh and limbs into urine." From then on, physicians began to gain a better understanding about diabetes.

Centuries later, people known as "water tasters" diagnosed diabetes by tasting the urine of people suspected to have it. If urine tasted sweet, diabetes was diagnosed. To acknowledge this feature, in 1675 the word "mellitus," meaning honey, was added to the name "diabetes," meaning siphon. It wasn't until the 1800s that scientists developed chemical tests to detect the presence of sugar in the urine.

Diabetes: Early Treatments

As physicians learned more about diabetes, they began to understand how it could be managed. The first diabetes treatment involved prescribed exercise, often horseback riding, which was thought to relieve excessive urination.

In the 1700s and 1800s, physicians began to realize that dietary changes could help manage diabetes, and they advised their patients to do things like eat only the fat and meat of animals or consume large amounts of sugar. During the Franco-Prussian War of the early 1870s, the French physician Apollinaire Bouchardat noted that his diabetic patients' symptoms improved due to war-related food rationing, and he developed individualized diets as diabetes treatments. This led to the fad diets of the early 1900s, which included the "oat-cure," "potato therapy," and the "starvation diet."

In 1916, Boston scientist Elliott Joslin established himself as one of the world's leading diabetes experts by creating the textbook *The Treatment of Diabetes Mellitus*, which reported that a fasting diet combined with regular exercise could significantly reduce the risk of death in diabetes patients. Today, doctors and diabetes educators still use these principles when teaching their patients about lifestyle changes for the management of diabetes.

Diabetes: Where We Are Today

Today, insulin is still the primary therapy used to treat type 1 diabetes; other medications have since been developed to help control blood glucose levels. Diabetic patients can now test their blood sugar levels at home, and use dietary changes, regular exercise, insulin, and other medications to precisely control their blood glucose levels, thereby reducing their risk of health complications.

- **What is this project?**

This project comprises of various models that analyse a given data set and categorize it into subparts:

1. Diabetes Positive i.e 1
2. Diabetes Negative i.e 0

2.1. IMPORTING LIBRARIES AND FUNCTIONS

1 Scikit-learn:

Scikit-learn is written in Python and uses NumPy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Python to improve performance. It provides many unsupervised and supervised learning algorithms. Scikitlearn integrates well with many other Python libraries, such as Matplotlib and plotly for plotting, NumPy for array vectorization, Pandas data frames, SciPy, and many more.

2 Pandas:

Pandas is a software library written for the Python programming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself. Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010. It is a popular python library for data analysis. It can be imported as a Julia package too. It is not related to Machine Learning. It is an opensource library that allows to perform data manipulation in python. It is used for data retrievals of csv files and datafiles. It provides an effortless way to create, manipulate, and wrangle the data. It is designed to make working with ‘relational’ or ‘labelled’ data both easy and intuitive. pd is an object of panda’s library.

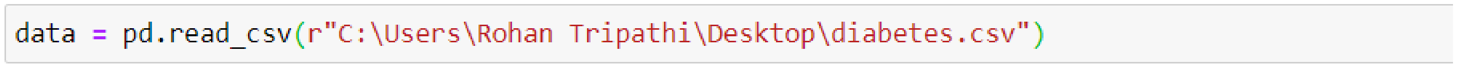
3 Seaborn:

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

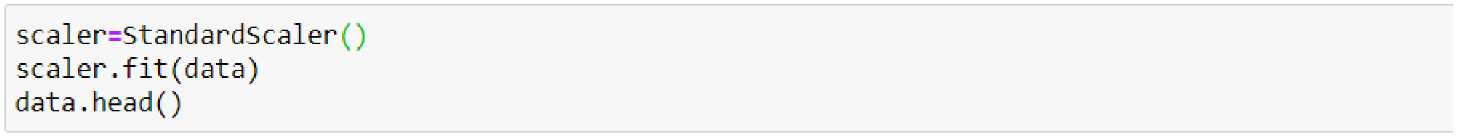
4 Matplotlib:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

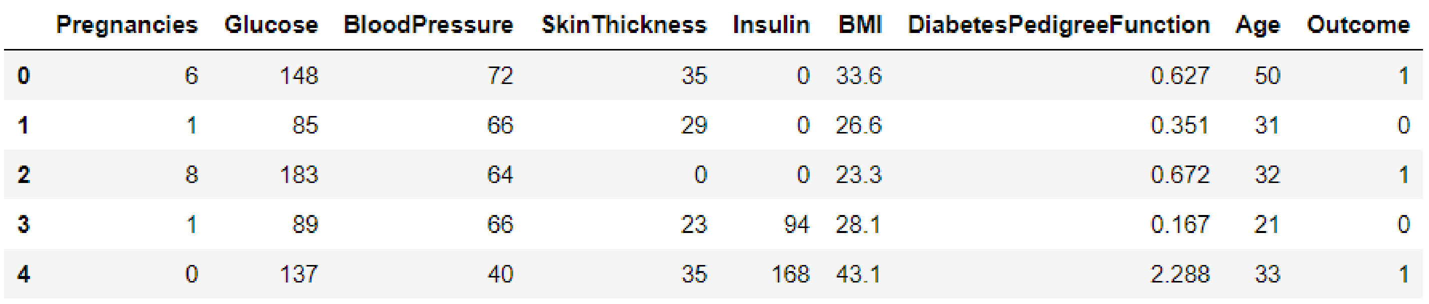
2.2. ANALYZING THE DATA



```
data = pd.read_csv(r"C:\Users\Rohan Tripathi\Desktop\diabetes.csv")
```



```
scaler=StandardScaler()
scaler.fit(data)
data.head()
```



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

So,

Here we have different columns or distinct factors which are responsible for categorizing a person having Diabetes or not into 2 categories present in the Outcome column:

0 for Diabetes Negative

1 for Diabetes Positive

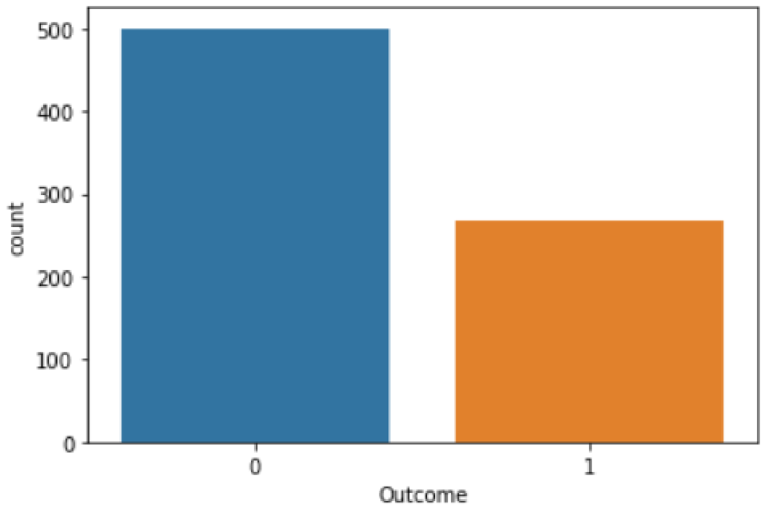
2.3. DATA VISUALIZATION

HISTOGRAM PLOT

A histogram is a plot that lets you discover, and show, the underlying frequency distribution (shape) of a set of continuous data. This allows the inspection of the data for its underlying distribution (e.g., normal distribution), outliers, skewness, etc. It is a graphical display of data using bars of different heights. In a histogram, each bar groups numbers into ranges. Taller bars show that more data falls in that range. A histogram displays the shape and spread of continuous sample data.

```
sns.countplot(x='Outcome',data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2d060688e80>
```



2.4. LOGISTIC REGRESSION

Logistic Regression is a classification algorithm used to assign observations to a discrete set of classes. Logistic Regression transforms its output using the logistic sigmoid function to return a probability value . There are two types are: BINARY AND MUILTI – LINEAR FUNCTION FAILSCLASS LOGISTIC REGRESSION . It is used for the classification problems , it is a predictive analysis algorithm and based on the concept of probability .The hypothesis of logistic regression tends it to limit the cost function between o and 1 . Therefore linear function fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression . We can call a Logistic Regression a linear Regression model but the logistic regression uses a more complex cost function can be defined as the 'SIGMOID FUNCTION' or also known as the 'logistic function' instead of a linear function .It is used to map predictions to probabilities . The func. maps any real value into another value between 0 and 1 .t is a technique to analyze a data-set which has a dependent variable and one or more independent variables to predict the outcome in a binary variable, meaning it will have only two outcomes.

The dependent variable is **categorical** in nature. Dependent variable is also referred as **target variable** and the independent variables are called the **predictors**.
Logistic regression is a special case of linear regression where we only predict the outcome in a categorical variable. It predicts the probability of the event using the log function.
We use the **Sigmoid function/curve** to predict the categorical value. The threshold value decides the outcome(win/lose).
Linear regression equation**:** **$y = \beta_0 + \beta_1X_1 + \beta_2X_2 \ldots. + \beta_nX_n$**

- Y stands for the dependent variable that needs to be predicted.

- β0 is the Y-intercept, which is basically the point on the line which touches the y-axis.

- β1 is the slope of the line (the slope can be negative or positive depending on the relationship between the dependent variable and the independent variable.)

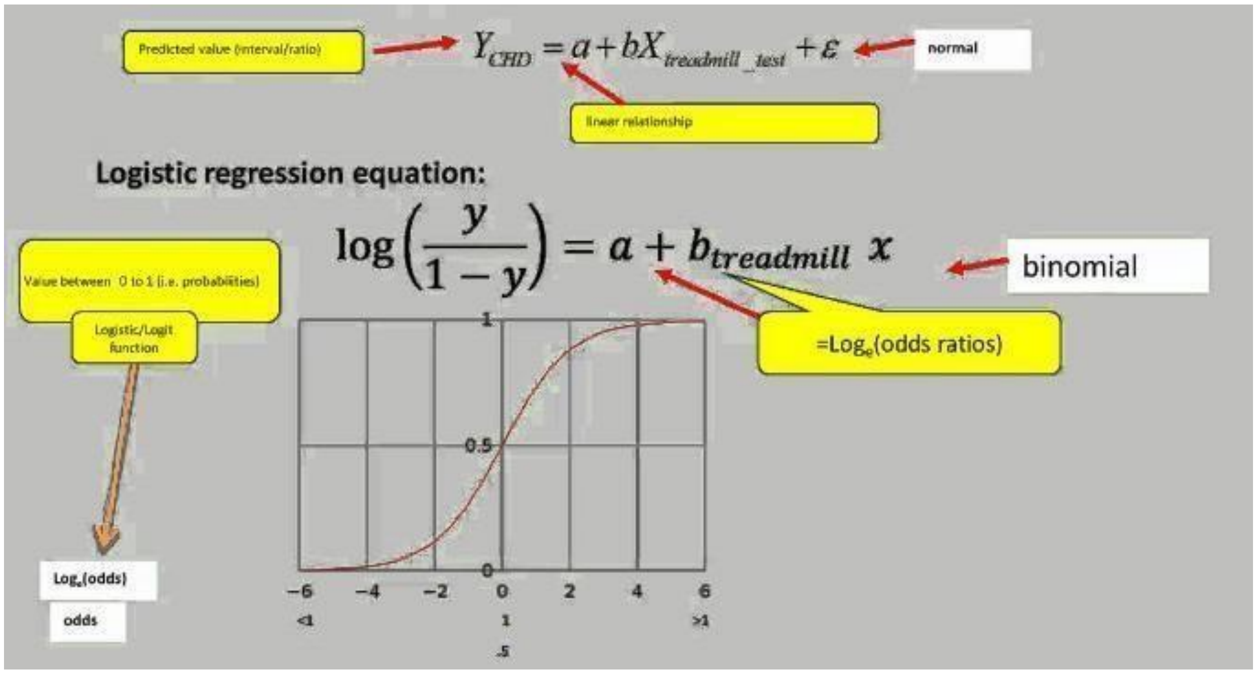
- X here represents the independent variable that is used to predict our resultant dependent value.

Sigmoid function: **$p = 1 / 1 + e^{-y}$**
Logistic Regression equation: **$p = 1 / 1 + e^{-(\beta_0 + \beta_1X_1 + \beta_2X_2 \ldots. + \beta_nX_n)}$**
Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function. It uses a black box function to understand the relation between the categorical dependent variable and the independent variables. Assumptions of Linear Regression. Linear regression is an analysis that assesses whether one or more predictor variables explain the dependent (criterion) variable. The regression has five key assumptions:

- Linear relationship.
- Multivariate normality.
- No or little multicollinearity.

- No autocorrelation.
- Homoscedasticity.

Logistic regression is considered a generalized **linear model** because the outcome always depends on the sum of the inputs and parameters. Or in other words, the output cannot depend on the product (or quotient, etc.) ... "A statistician calls a model "**linear**" if the mean of the response is a **linear** function of the parameter, and this is clearly violated for **logistic regression**.



Predicted value (interval/ratio)
$Y_{CHD} = a + bX_{treadmill\_test} + \varepsilon$
normal
linear relationship
Logistic regression equation:
$\log\left(\frac{y}{1-y}\right) = a + b_{treadmill}\, x$
binomial
Value between 0 to 1 (i.e. probabilities)
Logistic/Logit function
=Log$_e$(odds ratios)
1
0.5
0
−6 −4 −2 0 2 4 6
<1 1 >1
.5
Log$_e$(odds)
odds

Consider a model with two predictors, and , and one binary (Bernoulli) response variable , which we denote . We assume a linear relationship between the predictor variables, and the log-odds of the event that . This linear relationship can be written in the following mathematical form (where ℓ is the log-odds, is the base of the logarithm, and are parameters of the model):

$$\ell = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

We can recover the odds by exponentiating the log-odds:

$$\frac{p}{1-p} = b^{\beta_0+\beta_1 x_1+\beta_2 x_2}.$$

By simple algebraic manipulation, the probability that is

$$p = \frac{b^{\beta_0+\beta_1 x_1+\beta_2 x_2}}{b^{\beta_0+\beta_1 x_1+\beta_2 x_2}+1} = \frac{1}{1+b^{-(\beta_0+\beta_1 x_1+\beta_2 x_2)}}.$$

The above formula shows that once are fixed, we can easily compute either the log-odds that for a given observation, or the probability that for a given observation. The main use-case of a logistic model is to be given an observation , and estimate the probability that . In most applications, the base of the logarithm is usually taken to be e. However in some cases it can be easier to communicate results by working in base 2, or base 10.
We consider an example with , and coefficients , , and . To be concrete, the model is

$$\log_{10} \frac{p}{1-p} = \ell = -3 + x_1 + 2x_2$$

where is the probability of the event that .Y =1 .
This can be interpreted as follows:

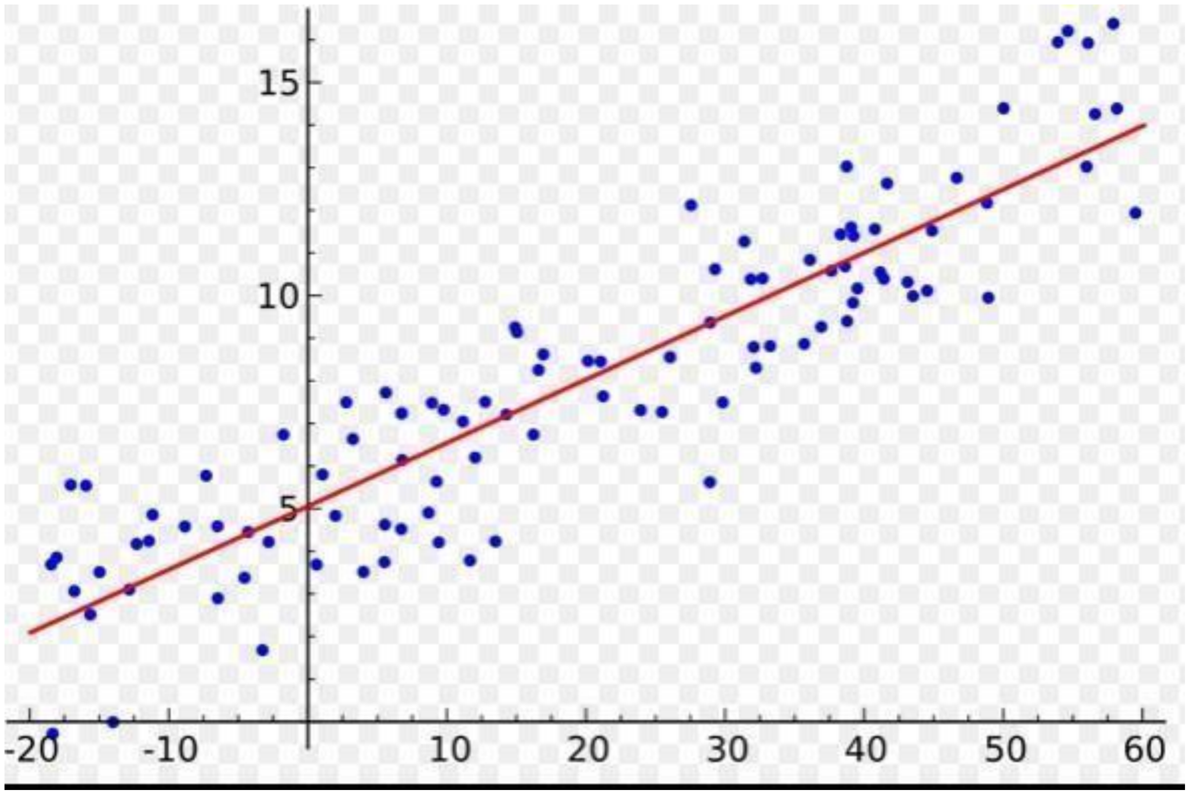
- is the y-intercept. It is the log-odds of the event that , when the predictors . By exponentiating, we can see that when the odds of the event that are 1-to-1000, or . Similarly, the probability of the event that when can be computed as .

- means that increasing by 1 increases the log-odds by . So if increases by 1, the odds that increase by a factor of .

- means that increasing by 1 increases the log-odds by . So if increases by 1, the odds that increase by a factor of Note how the effect of on the log-odds is twice as great as the effect of , but the effect on the odds is 10 times greater.

Logistic regression can be seen as a special case of generalized linear model and thus analogous to linear regression. The model of logistic regression, however, is based on quite different assumptions (about the relationship between dependent and independent variables) from those of linear regression.

Multivariate logistic regression is like simple **logistic regression** but with multiple predictors.
Logistic regression is like linear regression but you can use it when your response variable is binary. As in linear regression let's represent our hypothesis(Prediction Of Dependent Variable) in classification. In classification our hypothesis representation which tries to predict the binary outcome of either o or 1, will look like, $h\theta(x) = g(\theta T x) = 1/ 1 + e -\theta T x$,
Here $g(z) = 1/(1 + e$ ^$-z)$, is called the logistic function or the sigmoid function.



Here dots are scatterplot suggest the form and strength of the relationship between the dependent variable and regressors.

It can be used as a dot product between any two observations. The formula of linear kernel is as below − *K(x,xi)=sum(x∗xi)K(x,xi)=sum(x∗xi)*

From the above formula, we can see that the product between two vectors say $x\&xi$ is the sum of the multiplication of each pair of input values.

Polynomial Kernel

It is more generalized form of linear kernel and distinguish curved or nonlinear input space. Following is the formula for polynomial kernel − *k(X,Xi)=1+sum(X∗Xi)^dk(X,Xi)=1+sum(X∗Xi)^d*

Here d is the degree of polynomial, which we need to specify manually in the learning algorithm.

Radial Basis Function (RBF) Kernel

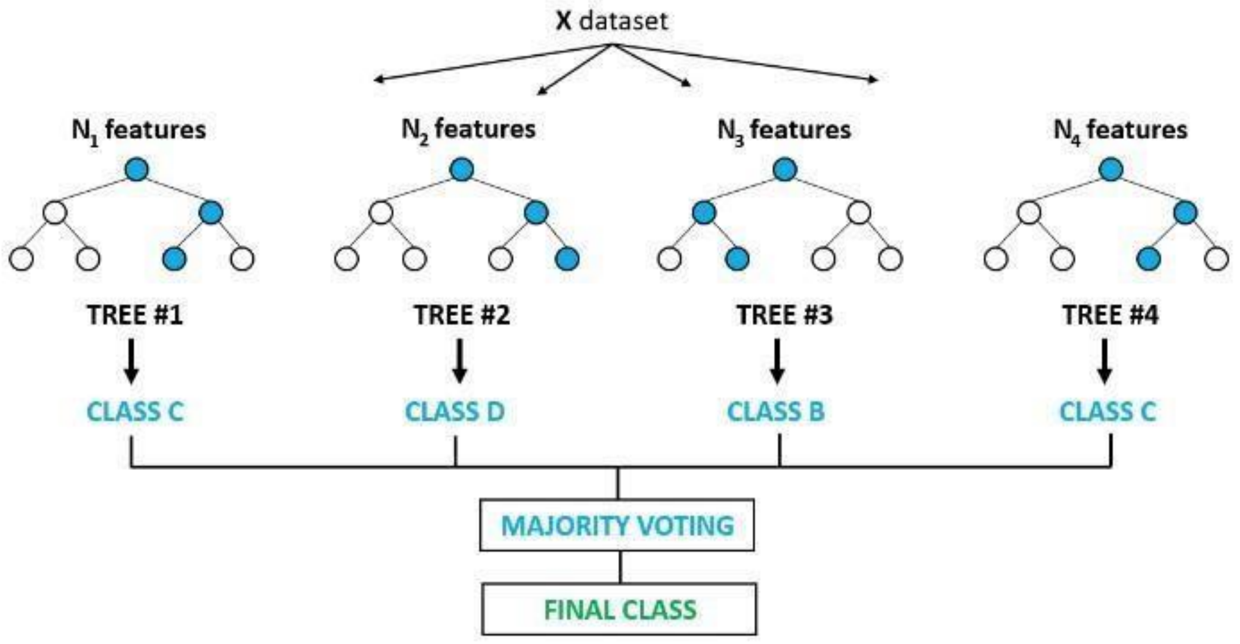
RBF kernel, mostly used in SVM classification, maps input space in indefinite dimensional space. Following formula explains it mathematically −

$K(x,xi)=exp(-gamma*sum(x-xi^2))K(x,xi)=exp(-gamma*sum(x-xi^2))$

Here, *gamma* ranges from 0 to 1. We need to manually specify it in the learning algorithm. A good default value of *gamma* is 0.1.

2.5. RANDOM FOREST

Random forests or random decision forests are an **ensemble learning method for classification, regression** and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.



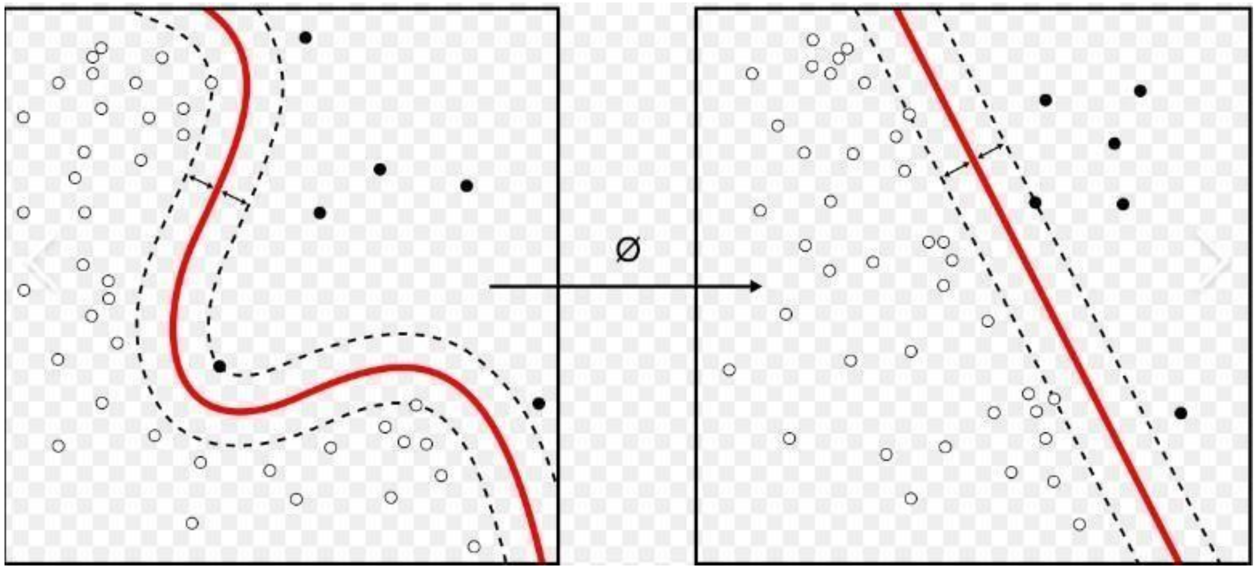
Random Forests are an improvement over bagged decision trees.
A problem with decision trees like CART is that they are greedy. They choose which variable to split on using a greedy algorithm that minimizes error. As such, even with Bagging, the decision trees can have a lot of structural similarities and in turn have high correlation in their predictions.
Combining predictions from multiple models in ensembles works better if the predictions from the sub-models are uncorrelated or at best weakly correlated.

Random forest changes the algorithm for the way that the sub-trees are learned so that the resulting predictions from all the subtrees have less correlation.
It is a simple tweak. In CART, when selecting a split point, the learning algorithm is allowed to look through all variables and all variable values to select the most optimal split-point. The random forest algorithm changes this procedure so that the learning algorithm is limited to a random sample of features of which to search.

- The number of features that can be searched at each split point (m) must be specified as a parameter to the algorithm. You can try different values and tune it using cross validation. For classification, a good default is m = sqrt(p)
- For regression, a good default is m = p/3

Where m is the number of randomly selected features that can be searched at a split point and p is the number of input variables. For example, if a dataset had 25 input variables for a classification problem, then:

- m = sqrt (25)
- m = 5
- simple mathematical representation.
- Random forest works by building decision trees & then aggregating them & hence the Beta values have no counterpart in random forest. Though you do get the 'Variable Importance /Gini Index' values for the forest, which can be used for making sense of the model but not as a multiplication factor.
- Now to answer your question, from your code snippet it seems that you have stored you model as rf. To use this model for prediction, you can simply call the predict method in python associated with the random forest class.
- use:
- prediction = rf.predict(test)
- This will give you the predictions for you new data (test here) based on the model rf. The predict method won't build a new model, it'll use the model rf to use for prediction on new data.



Classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, based on a training set of data containing observations (or instances) whose category membership is known. Examples are assigning a given email to the "spam" or "nonspam" class and assigning a diagnosis to a given patient based on observed characteristics of the patient (sex, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition.

In the terminology of machine learning,[1] classification is considered an instance of supervised learning, i.e., learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering and involves grouping data into categories based on some measure of inherent similarity or distance.

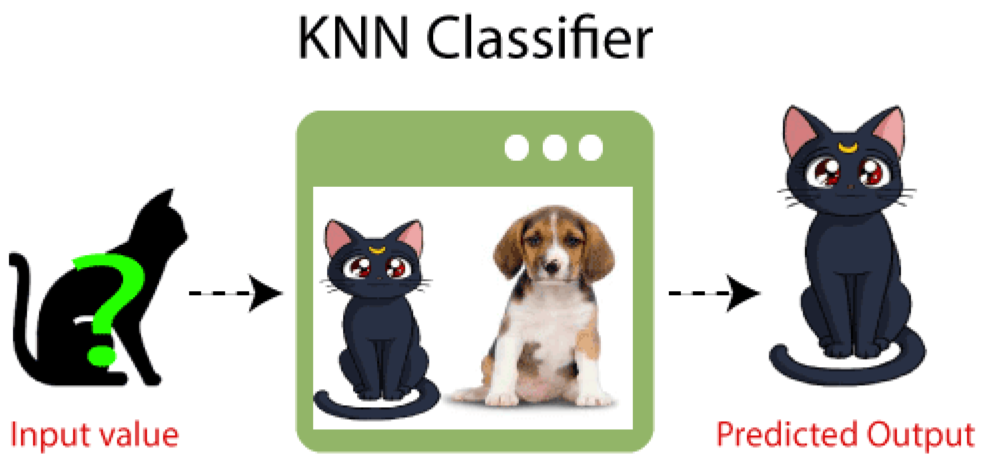
Often, the individual observations are analyzed into a set of quantifiable properties, known variously as explanatory variables or features. These properties may variously be categorical (e.g., "A", "B", "AB" or "O", for blood type), ordinal (e.g. "large", "medium" or "small"), integer-valued (e.g. the number of occurrences of a particular word in an email) or real-valued (e.g. a measurement of blood pressure). Other classifiers work by comparing observations to previous observations by means of a similarity or distance function.

An algorithm that implements classification, especially in a concrete implementation, is known as a **classifier**. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm, that maps input data to a category.

Terminology across fields is quite varied. In statistics, where classification is often done with logistic regression or a similar procedure, the properties of observations are termed explanatory variables (or independent variables, regressors, etc.), and the categories to be predicted are known as outcomes, which are possible values of the dependent variable. In machine learning, the observations are often known as instances, the explanatory variables are termed features (grouped into a feature vector), and the possible categories to be predicted are classes. Other fields may use different terminology: e.g., in community ecology, the term "classification" normally refers to cluster analysis, i.e., a type of unsupervised learning, rather than the supervised learning.

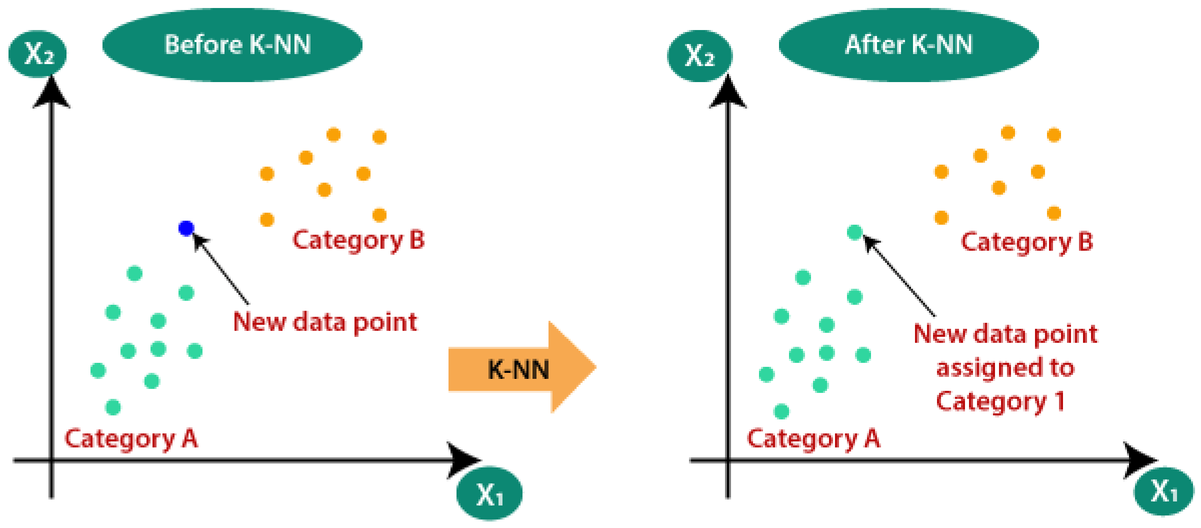
2.6. KNeighborsClassifier

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- 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 also called 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.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

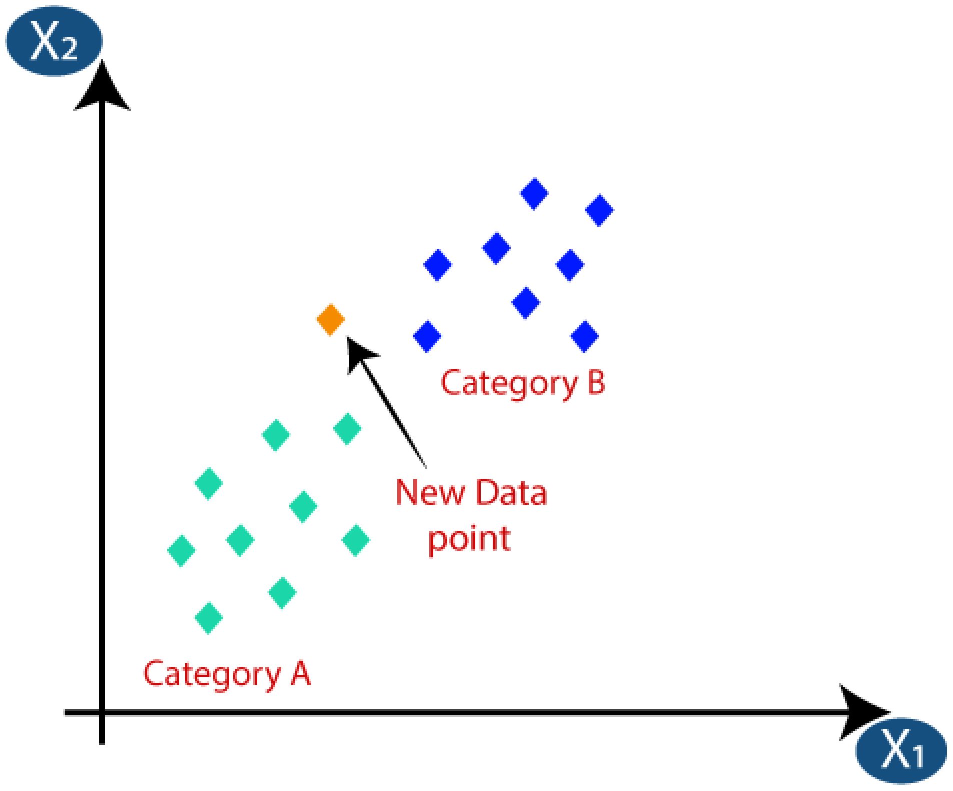


How does K-NN work?

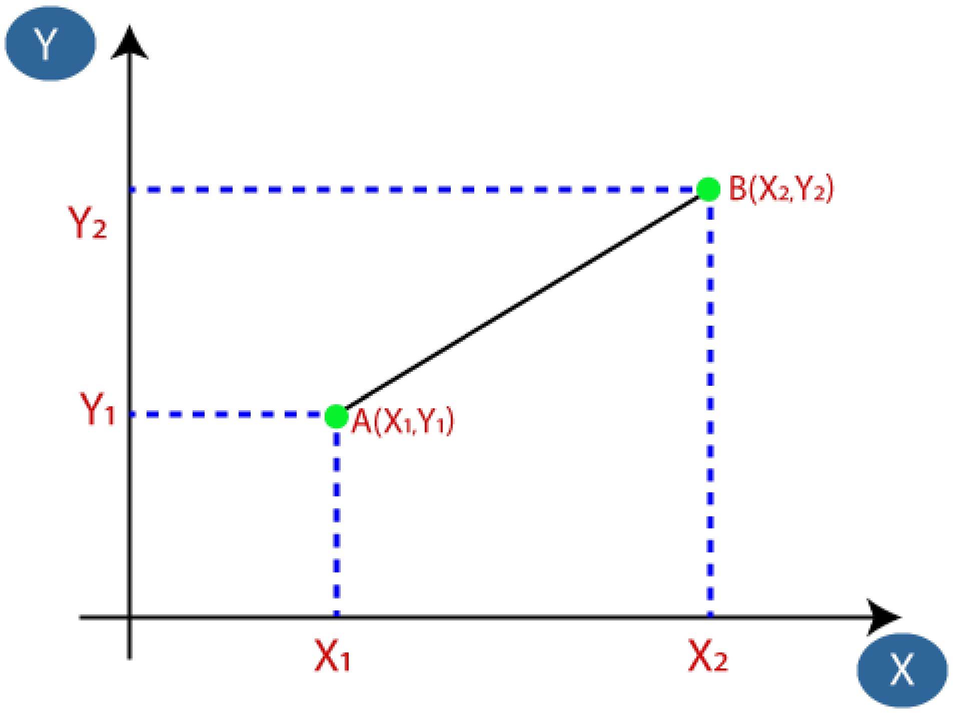
The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:

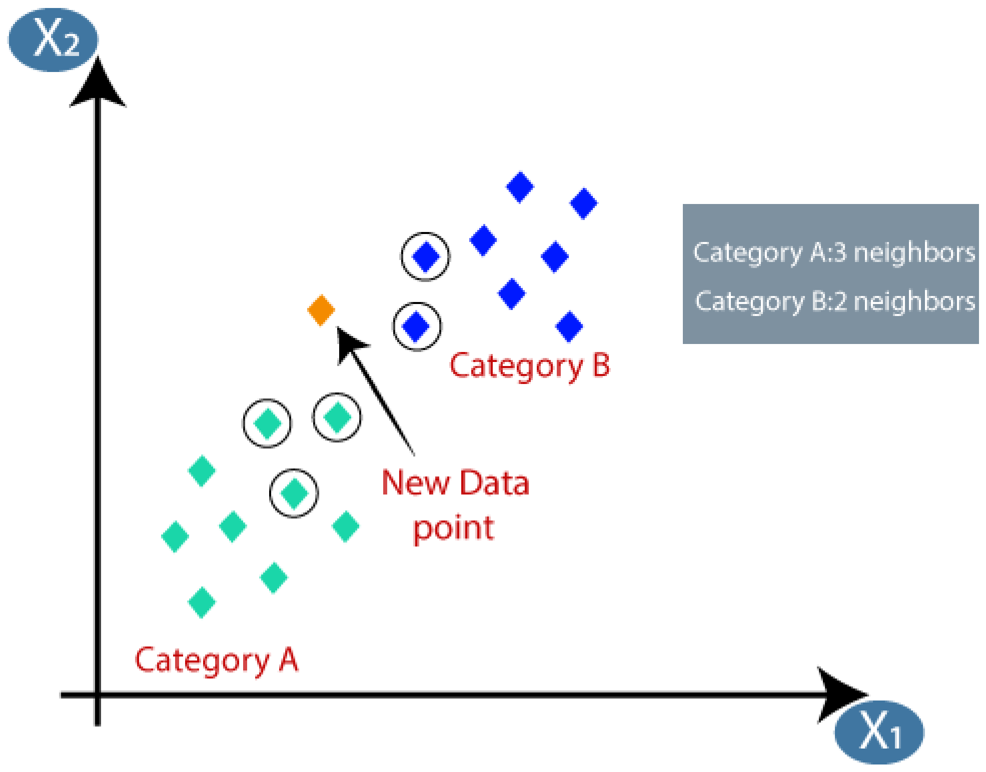


- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



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

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

How to select the value of K in the K-NN Algorithm?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- 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.
- Large values for K are good, but it may find some difficulties.

Advantages of KNN Algorithm:

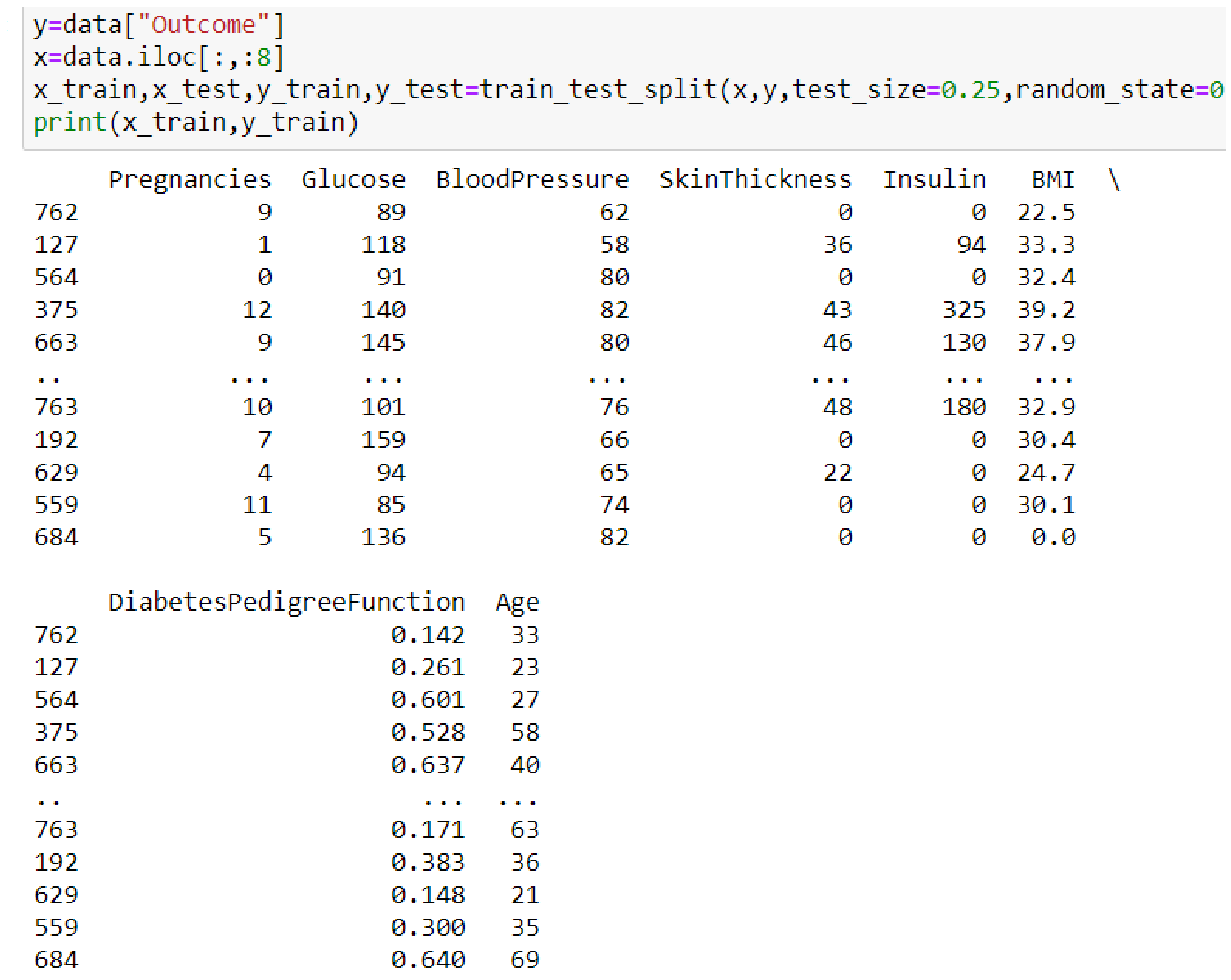
- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

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.

4.1. TRAINING THE MODEL

For training model , I trained 75% of the data from the dataset

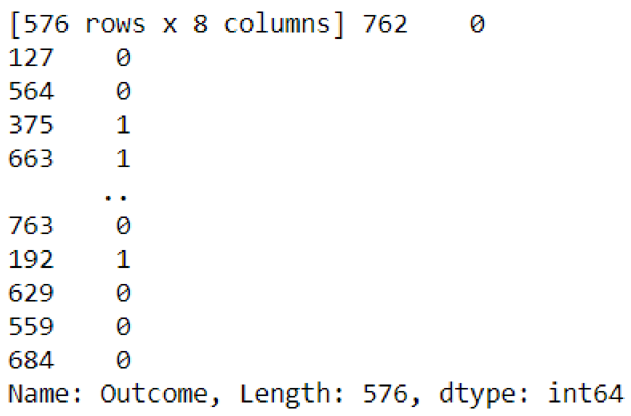


```
y=data["Outcome"]
x=data.iloc[:,:8]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0
print(x_train,y_train)
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
762            9       89             62              0        0  22.5
127            1      118             58             36       94  33.3
564            0       91             80              0        0  32.4
375           12      140             82             43      325  39.2
663            9      145             80             46      130  37.9
..           ...      ...            ...            ...      ...   ...
763           10      101             76             48      180  32.9
192            7      159             66              0        0  30.4
629            4       94             65             22        0  24.7
559           11       85             74              0        0  30.1
684            5      136             82              0        0   0.0

     DiabetesPedigreeFunction  Age
762                     0.142   33
127                     0.261   23
564                     0.601   27
375                     0.528   58
663                     0.637   40
..                        ...  ...
763                     0.171   63
192                     0.383   36
629                     0.148   21
559                     0.300   35
684                     0.640   69
```

The above columns such as Pregnancies , Glucose ,Bloodpressure etc contains 75% of total data of the whole dataset.
And this data is the trained one.



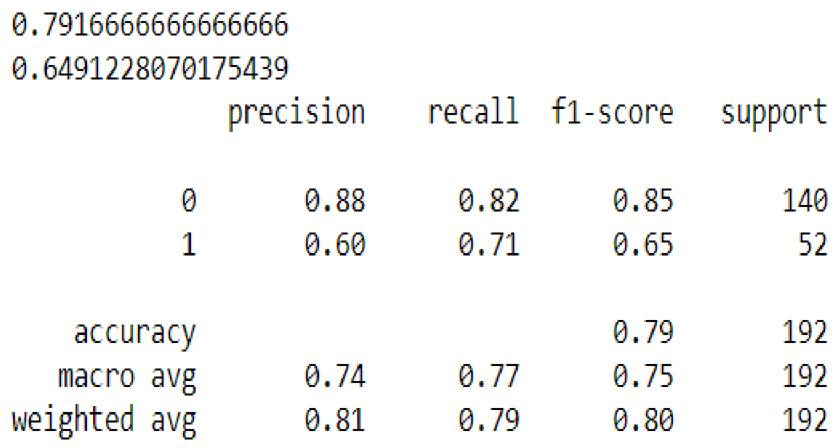
```
[576 rows x 8 columns] 762    0
127    0
564    0
375    1
663    1
      ..
763    0
192    1
629    0
559    0
684    0
Name: Outcome, Length: 576, dtype: int64
```

Above is 75% of the output data which was saved in data set in column outcome as shown above.

4.2. ACCURACY OF THE MODEL

Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions. predictions are found by testing model on test data.
The below image shows the prediction of our model and we use *classification report* and *accuracy score* to check our model accuracy.

- Logistic Regression:



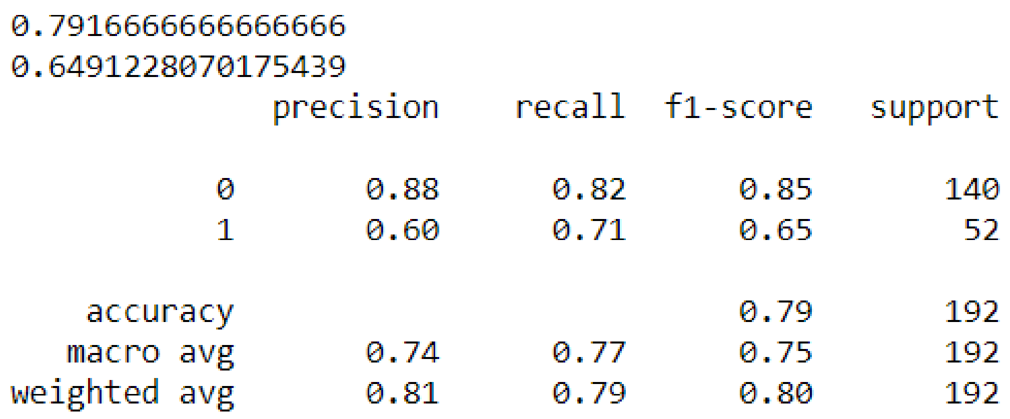
```
0.7916666666666666
0.6491228070175439
              precision    recall  f1-score   support

           0       0.88      0.82      0.85       140
           1       0.60      0.71      0.65        52

    accuracy                           0.79       192
   macro avg       0.74      0.77      0.75       192
weighted avg       0.81      0.79      0.80       192
```

Accuracy for this model is 0.7916666666666666.

- Random Forest:



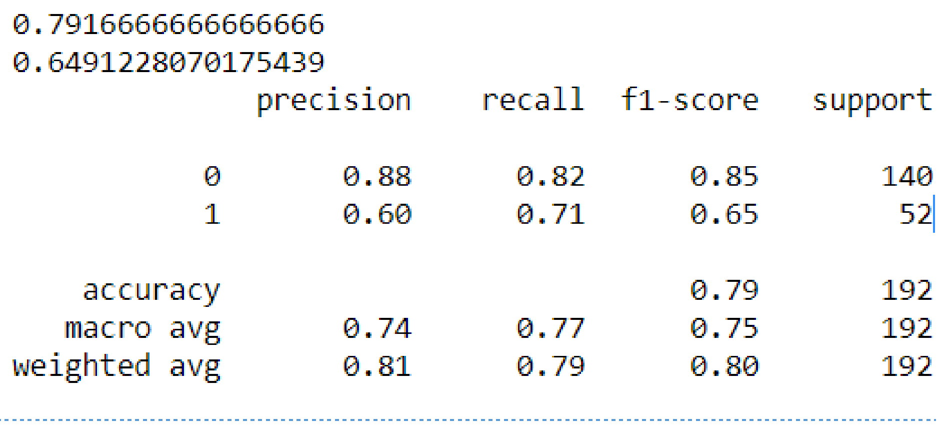
```
0.7916666666666666
0.6491228070175439
              precision    recall  f1-score   support

           0       0.88      0.82      0.85       140
           1       0.60      0.71      0.65        52

    accuracy                           0.79       192
   macro avg       0.74      0.77      0.75       192
weighted avg       0.81      0.79      0.80       192
```

Accuracy for this model is also 0.7916666666666666.

- KNeighborsClassifier:



```
0.7916666666666666
0.6491228070175439
              precision    recall  f1-score   support

           0       0.88      0.82      0.85       140
           1       0.60      0.71      0.65        52

    accuracy                           0.79       192
   macro avg       0.74      0.77      0.75       192
weighted avg       0.81      0.79      0.80       192
```

Accuracy for this model is also 0.7916666666666666.

5.1. CONFUSION MATRIX

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. **The confusion matrix shows the ways in which your classification models confused when it makes predictions.** There could be four possible outcomes. Let us look at all four:
True Positives (TP) - These are the correctly predicted positive values which means that the value of actual class is yes, and the value of predicted class is also yes. E.g., if actual class value indicates that this passenger survived and predicted class tells you the same thing.
True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no, and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing. False positives and false negatives, these values occur when your actual class contradicts with the predicted class. **False Positives (FP)** – When actual class is no, and predicted class is yes. E.g., if actual class says this passenger did not survive but predicted class tells you that this passenger will survive. **False Negatives (FN)** – When actual class is yes but predicted class in no. E.g., if actual class value indicates that this passenger survived and predicted class tells you that passenger will die. Once you understand these four parameters then we can calculate Accuracy, Precision, Recall and F1 score.
Accuracy - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where

values of false positive 34 and false negatives are almost same. Therefore, you must look at other parameters to evaluate the performance of your model. For our
model, we have got 0.803 which means our model is approx. 80% accurate. *Accuracy = TP+TN/TP+FP+FN+TN*

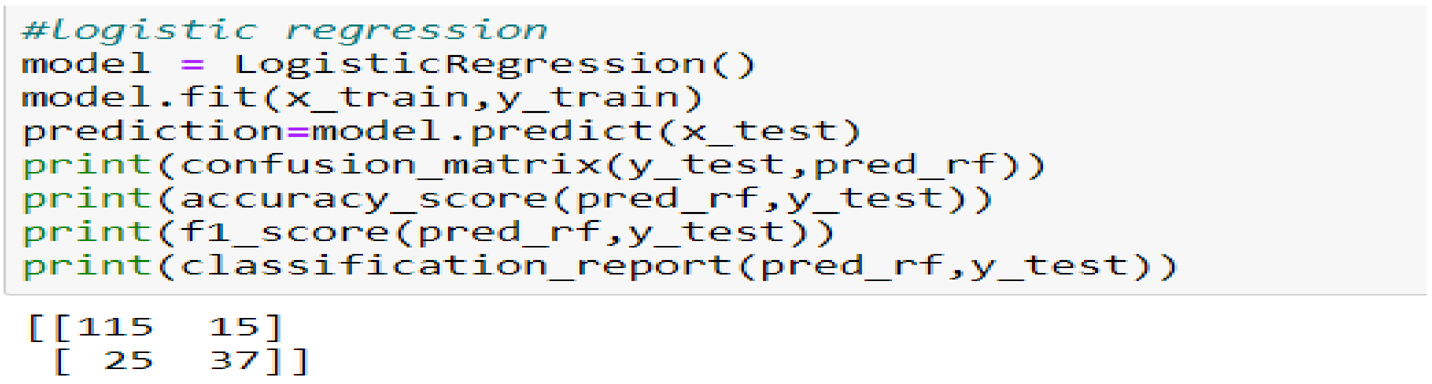
Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many survived? High precision relates to the low false positive rate. We have 0.788 precision which is good.
Precision = TP/TP+FP

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes. The question recall answers is: Of all the passengers that truly survived, how many did we label? We have got recall of 0.631 which is good for this model as it's above 0.5. *Recall = TP/TP+FN*

F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are quite different, it's better to look at both Precision and Recall. In our case, F1 score is 0.701.
F1 Score = 2(Recall * Precision) / (Recall + Precision)*

The support is the number of samples of the true response that lie in that class.
Confusion matrix output:

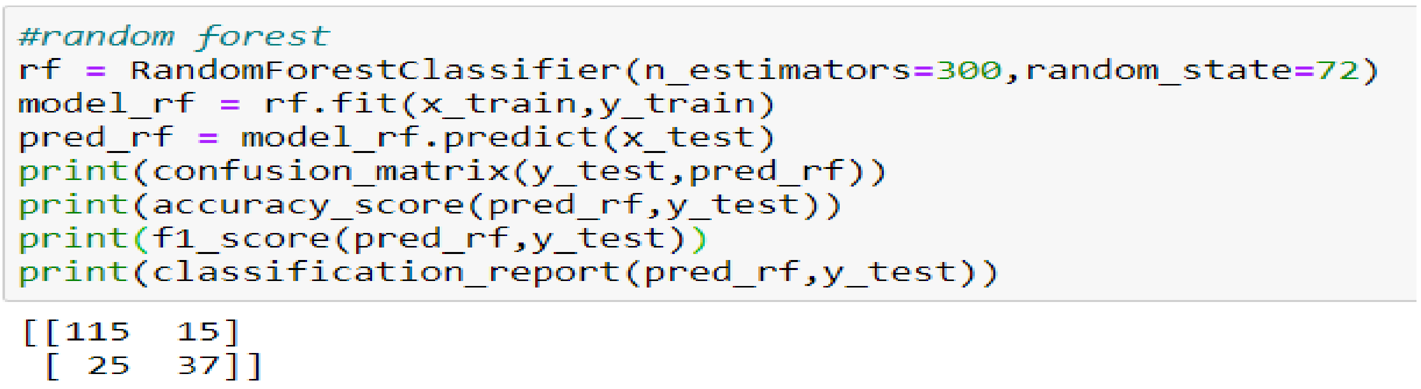
- Logistic Regression:



```
#logistic regression
model = LogisticRegression()
model.fit(x_train,y_train)
prediction=model.predict(x_test)
print(confusion_matrix(y_test,pred_rf))
print(accuracy_score(pred_rf,y_test))
print(f1_score(pred_rf,y_test))
print(classification_report(pred_rf,y_test))
```

```
[[115  15]
 [ 25  37]]
```

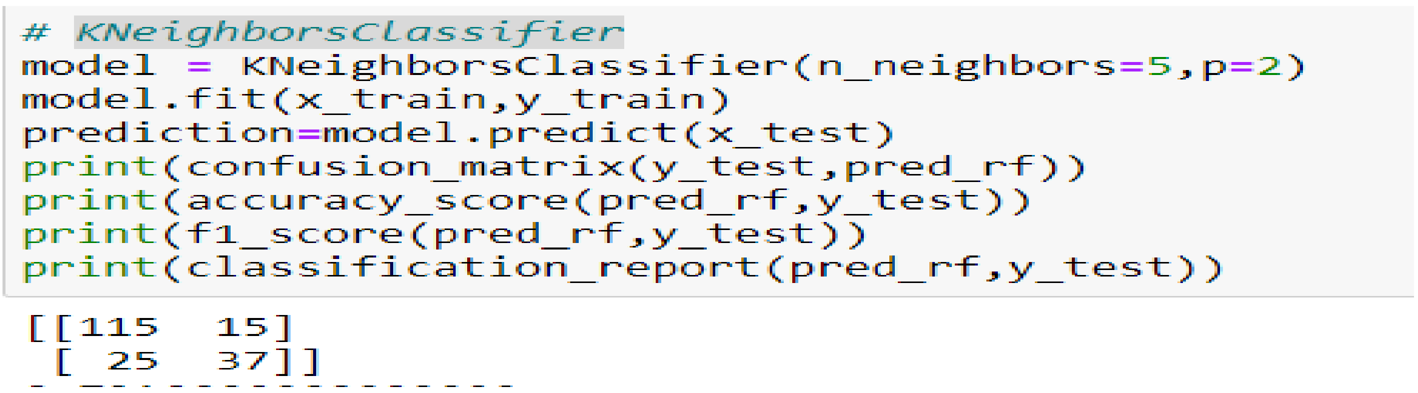
- Random Forest:



```
#random forest
rf = RandomForestClassifier(n_estimators=300,random_state=72)
model_rf = rf.fit(x_train,y_train)
pred_rf = model_rf.predict(x_test)
print(confusion_matrix(y_test,pred_rf))
print(accuracy_score(pred_rf,y_test))
print(f1_score(pred_rf,y_test))
print(classification_report(pred_rf,y_test))
```

```
[[115  15]
 [ 25  37]]
```

- KNeighborsClassifier:



```
# KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=5,p=2)
model.fit(x_train,y_train)
prediction=model.predict(x_test)
print(confusion_matrix(y_test,pred_rf))
print(accuracy_score(pred_rf,y_test))
print(f1_score(pred_rf,y_test))
print(classification_report(pred_rf,y_test))
```

```
[[115  15]
 [ 25  37]]
```

CONCLUSION

SO,

Till now we have 3 models named Random Forest, Logistic Regression and KNeighborsClassifier

That take inputs of Pregnancies , Glucose ,BloodPressure ,SkinThickness ,Insulin ,DiabetesPedigreeFunction ,Age and Bmi and show the output in the form of outcome which detects a person is Diabetes positive or negative.

REFRENCES

1. 1. Debadri Dutta, Debpriyo Paul, Parthajeet Ghosh, "Analyzing Feature Importances for Diabetes Prediction using Machine Learning". IEEE, pp 942-928, 2018.

2. K.VijiyaKumar, B.Lavanya, I.Nirmala, S.Sofia Caroline, "Random Forest Algorithm for the Prediction of Diabetes ".Proceeding of International Conference on Systems Compu- tation Automation and Networking, 2019.

3. Md. Faisal Faruque, Asaduzzaman, Iqbal H. Sarker, "Perfor- mance Analysis of Machine Learning Techniques to Predict Diabetes Mellitus". International Conference on Electrical, Computer and Communication Engineering (ECCE), 7-9 Feb- ruary, 2019.

4. Tejas N. Joshi, Prof. Pramila M. Chawan, "Diabetes Prediction Using Machine Learning Techniques".Int. Journal of Engineer- ing Research and Application, Vol. 8, Issue 1, (Part -II) Janu- ary 2018, pp.-09-13

5. Nonso Nnamoko, Abir Hussain, David England, "Predicting Diabetes Onset: an Ensemble Supervised Learning Approach ". IEEE Congress on Evolutionary Computation (CEC).

6. Deeraj Shetty, Kishor Rit, Sohail Shaikh, Nikita Patil, "Diabe- tes Disease Prediction Using Data Mining ".International Con- ference on Innovations in Information, Embedded and Com- munication Systems (ICIIECS).

7. Nahla B., Andrew et al,"Intelligible support vector machines for diagnosis of diabetes mellitus. Information Technology in Biomedicine".