Stevens Institute of Technology

Diabetes Prediction

Prediction of Diabetes using Machine learning algorithms

Manish Yadav & Shrinidhi Athanikar

Multivariate Analysis BIA – 652D

Dec 4, 2018

Project Proposal

Abstract

Today due to modern life style the people have more stress of work, less physical activities, changed eating habits due to this reason people facing many health related problem. The diabetes is one reason behind the death of people. Diabetes may lead to kidney, eye problems heart problem also[1]. Hence it is better to detect Diabetes in early stage to avoid other health risks.Early detection of diabetes can reduce patient's health risk. Physicians, patients, and patient's relatives can be benefited from the prediction's outcomes. In low resource clinical settings, it is necessary to predict the patient's condition after the admission to allocate resources appropriately[2]. Using different machine learning algorithm such as Linear Discriminant Analysis, K Nearest Neighbors, Logistic Regression, Naïve Bayes Classifier, Principal Component Analysis various models were developed and depending on accuracy and other factors the best model is selected. The outcome of the model is 0 or 1 indicating that the person is diabetic or not respectively.

Introduction

1.1 Diabetes Mellitus

Diabetes is one of deadliest disease in the world. It is not only a disease but also a creator of different kinds of diseases like heart attack, blindness, kidney diseases, etc. The normal identifying process is that patients need to visit a diagnostic center, consult their doctor, and sit tight for a day or more to get their reports. Moreover, every time they want to get their diagnosis report, they have to waste their money in vain. Diabetes Mellitus (DM) is defined as a group of metabolic disorders mainly caused by abnormal insulin secretion and/or action. Insulin deficiency results in elevated blood glucose levels (hyperglycemia) and impaired metabolism of carbohydrates, fat and proteins. DM is one of the most common endocrine disorders, affecting more than 200 million people worldwide. The onset of diabetes is estimated to rise dramatically in the upcoming years. DM can be divided into several distinct types. (Maturity Onset Diabetes of the Young), neonatal, mitochondrial, and pregnancy diabetes. The symptoms of DM include polyuria, polydipsia, and significant weight loss among others. Diagnosis depends on blood glucose levels (fasting plasma glucose = 7.0 mmol/L[3].

Machine Learning

The supervised learning algorithms are classified into different types such as probability-based, function-based, rule-based, tree-based, instance-based, etc. The unsurprised learning is the descriptive type learning. This learning is used to describe or summarize the data. The examples of the unsupervised learning algorithms are clustering, association rule mining, etc. The semi-supervised learning is the combination of supervised and unsupervised. This report presents a diabetes prediction system to diagnosis the diabetics. Moreover, the supervised learning algorithm is used to learn the diabetes data and to develop diabetes prediction system for diagnosing diabetes. The accuracy of this prediction system is improved using pre-processing technique[4].

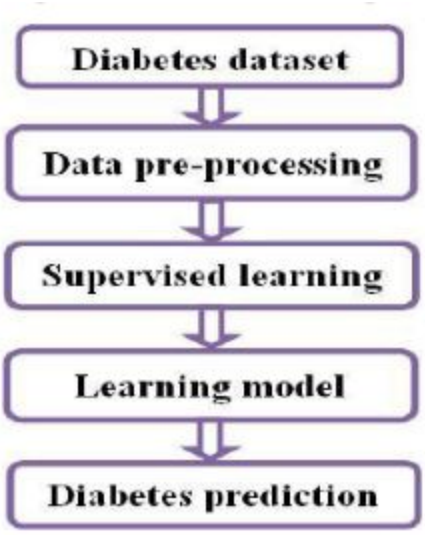


Figure 1 Flowchart representation of diabetes predication system

Problem Description:

According to the Centers for Disease Control and Prevention statistics, one among seven adults are diabetic, this rate would be skyrocketing by 2050 to as many as one in three. Most of us don't know the dangers of being a diabetic. It's important to separate out facts from fiction in order to understand diabetes and help to contain this leading cause of disability and death. So in order to do that a successful solution would be to more effectively predict the diabetic condition of a person using machine learning algorithms.

Data Pre-processing:

To begin with data pre-processing the data set was imported. Then the independent variables namely Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes, Age were separated from the dependent variable which is the outcome of the experiment. Then the data was standardized using SciKit learn library in python by using StandardScaler function. The idea behind StandardScaler is that it will transform your data such that its distribution will have a mean value 0 and standard deviation of 1. Given the distribution of the data, each value in the dataset will have the sample mean value subtracted, and then divided by the standard deviation of the whole dataset.

How to calculate it manually:

Standardization:

$$z = \frac{x - \mu}{\sigma}$$

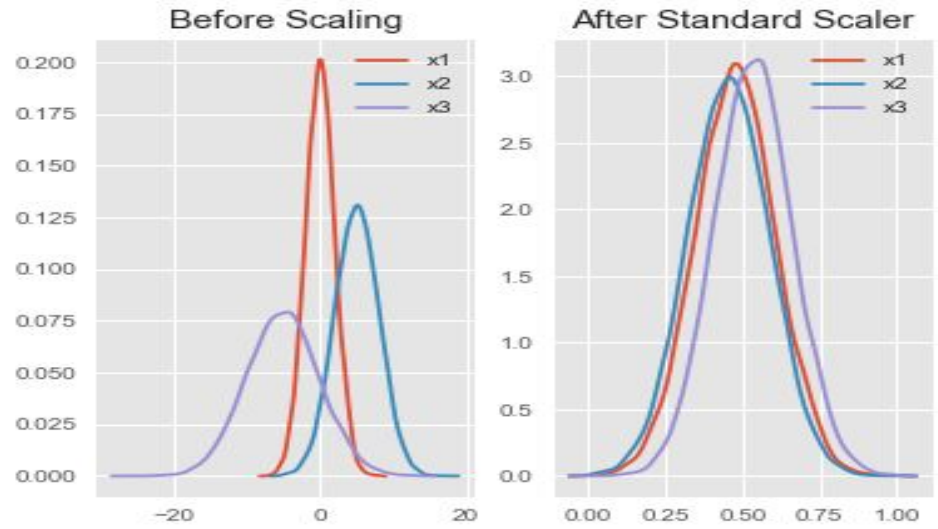
with mean:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i)$$

and standard deviation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

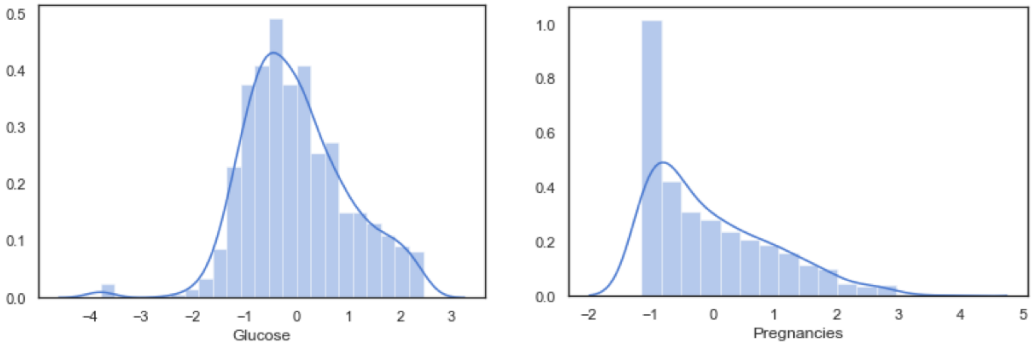
An example of what StandardScaler does to your dataset is shown below

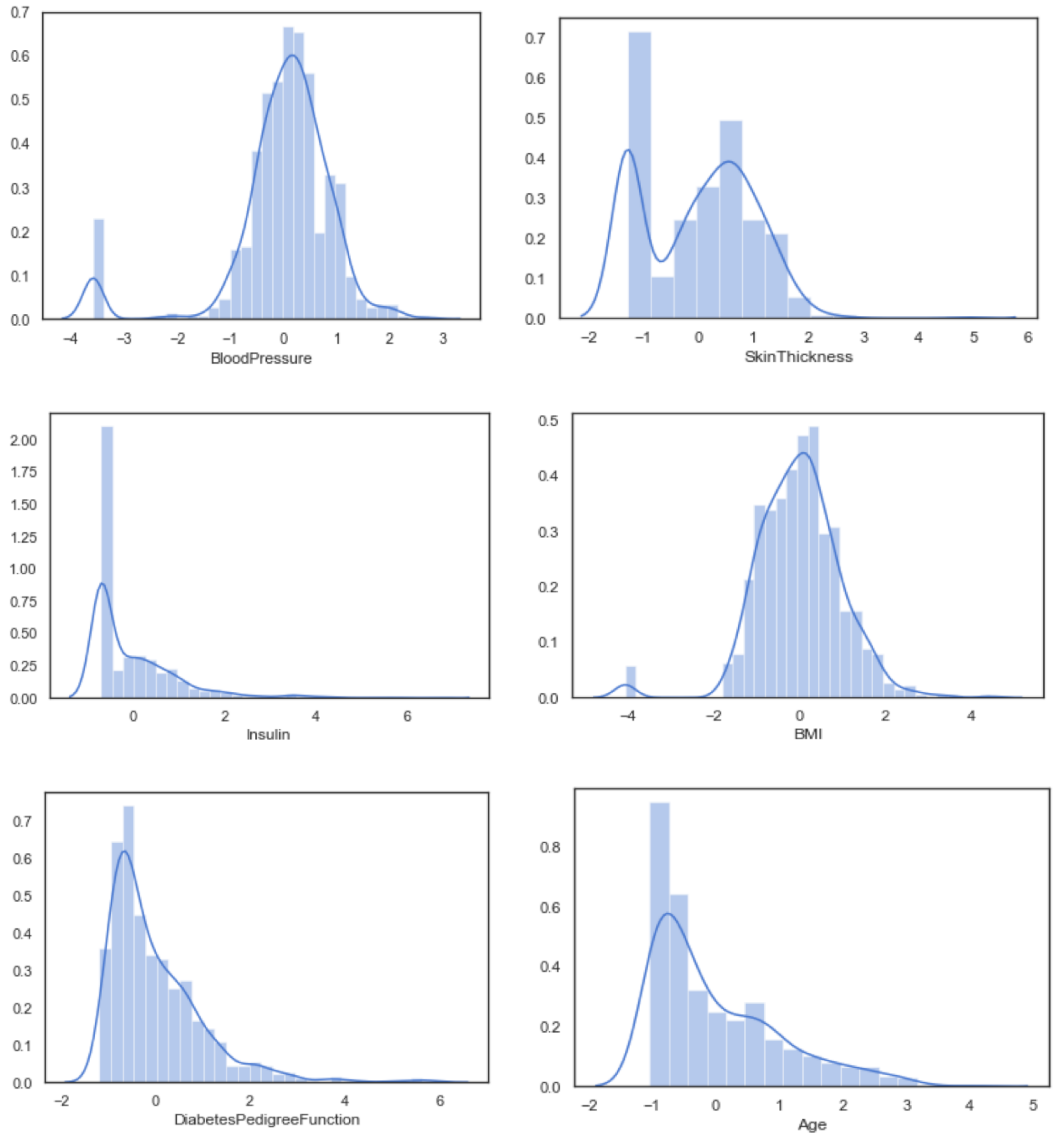


All features are now on the same scale relative to one another.

Identifying the outliers

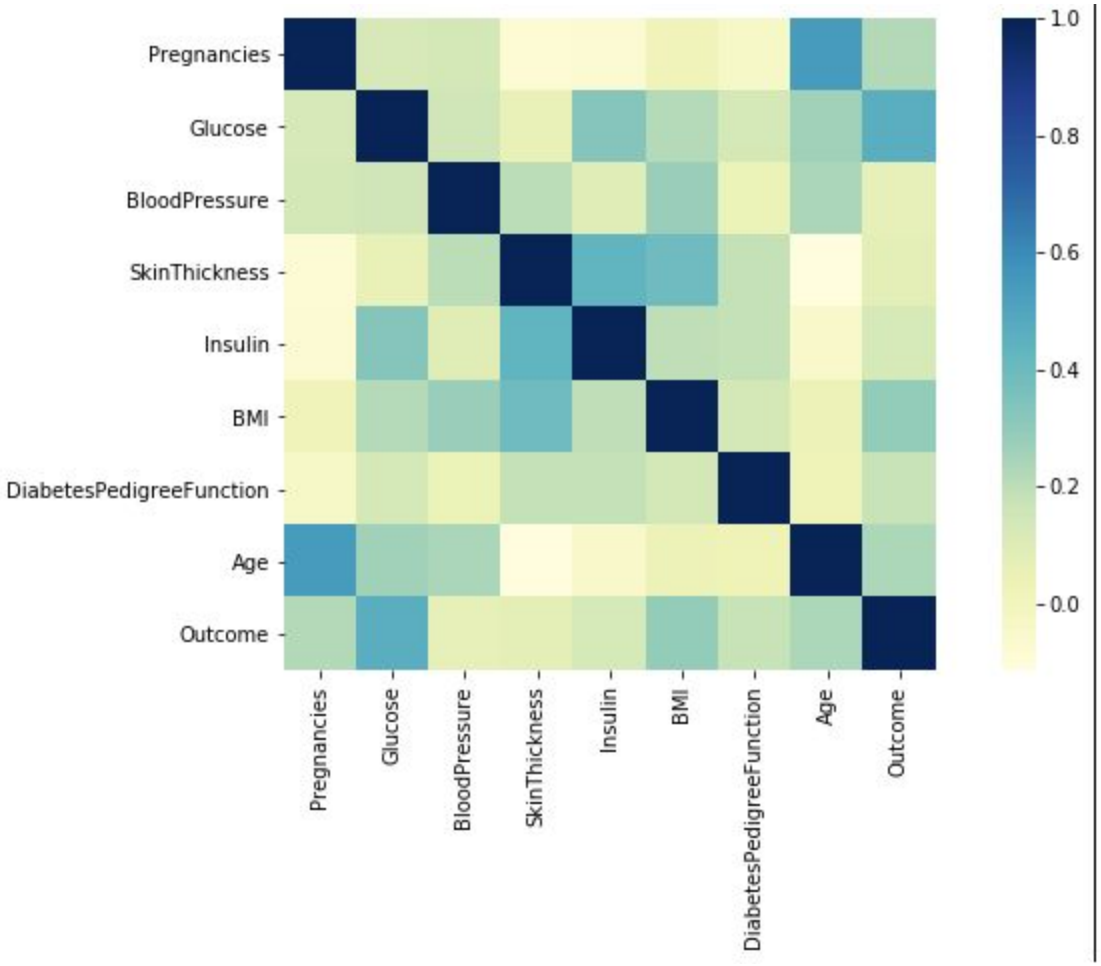
The outliers were identified and removed using the histogram shown below





Identifying correlation

Subsequent to distinguishing and expelling outliers, the following step taken was to run a correlation analysis to have a deeper and better understanding of how every one of the variables are related with one another. Multicollinearity happens when independent variables are highly related with one another. This issue can decrease the impact of an predictor variable on a predictive model and can be especially hazardous for logistic regression models.The heatmap for correation matrix is as shown below:



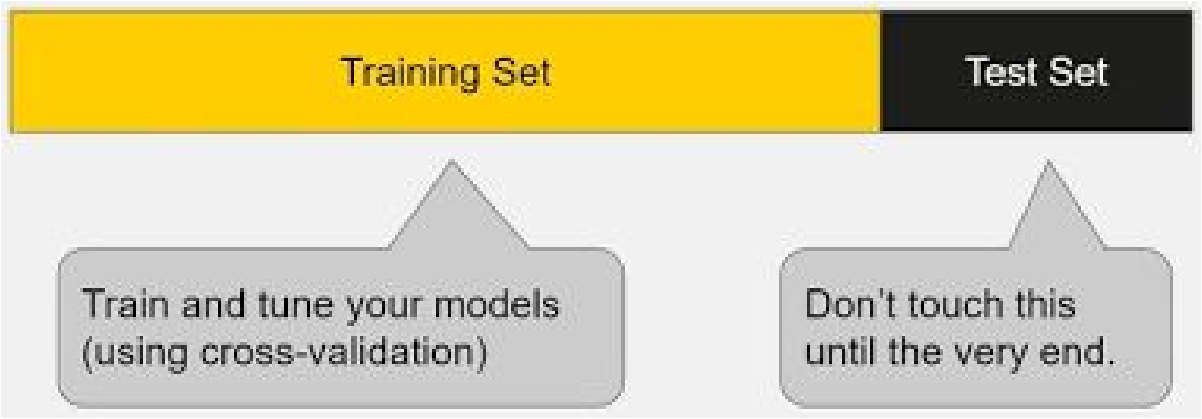
	Pregnancies	*Glucose*	*BloodPressure*	*SkinThickness*	*Insulin*	*BMI*	*D_Pedigree*	*Age*	*Outcome*
Pregnancies	1								
Glucose	0.129458671	1							
BloodPressure	0.141281977	0.15259	1						
SkinThickness	-0.081671774	0.057328	0.207370538	1					
Insulin	-0.073534614	0.331357	0.088933378	0.43678257	1				
BMI	0.017683091	0.221071	0.281805289	0.392573204	0.197859	1			
DiabetesPedigree	-0.033522673	0.137337	0.041264948	0.183927573	0.185071	0.140647	1		
Age	0.544341228	0.263514	0.239527946	-0.113970262	-0.04216	0.036242	0.03356131	1	
Outcome	0.221898153	0.466581	0.06506836	0.074752232	0.130548	0.292695	0.17384407	0.238356	1

The values in red indicate that the variables are highly correlated to each other.

Splitting data into Training vs. Test set

To best analyze the prediction power of most models best practices advise splitting the dataset into a “Training” & “Test” set. By building the model on a training dataset and testing the model’s prediction ability on a test dataset we can more significantly test the model’s accuracy correcting for overfitting to the training dataset, helping eliminate training bias.

To know how good your model is, i.e in order to know the prediction power of the model the best approach would be to divide the model into training set and testing set. The training data set would be used to train the model and the testing data set would be used to check the model’s accuracy also checking for overfitting of the training dataset.



Accuracy measured on a Test dataset is an indicator of a more power predictor than accuracy results measured on a Training dataset. The following models were built using the ‘Training’ data, but accuracy was measured on the prediction of the ‘Test’ dataset.

Analysis Methods:

After the data was fully cleaned and prepared for analysis, our next step was to develop a model. Given that the dependent variable is binary the initial model used for prediction were Linear Discriminant Analysis(LDA), K-Nearest Neighbour(KNN), Logistic Regression and Naive Bayes algorithm.

Before developing the model, the first step is to read the excel file. For doing this, we need to import the library “pandas” and use “read_excel()” function to read our dataset.

After reading the dataset, the dataset was divided into two variables X and Y. The X variable will consist of all the independent variables and the Y variable will consist of the dependent or the outcome variable. This can be done using “dataframe.iloc()” function.

Initially the model was divided into two sets: Training Dataset and Testing Dataset. The model was trained on the training dataset and the tested on the testing dataset. Taking into consideration every possible split, we concluded that the model performs better when the test size is 29% of the whole data. This was implemented using the function “train_test_split()” from sklearn.model_selection library.

Linear Discriminant Analysis

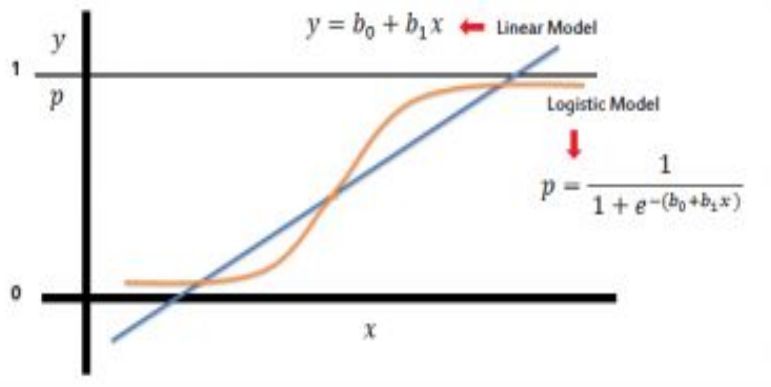
Linear Discriminant Analysis is a method which is used to find a linear combination of features that characterizes two or more classes of objects. The resulting combination may be used as a linear classifier , or for dimensionality reduction before later classification.

K-Nearest Neighbor

K-Nearest Neighbor is another classification technique which we used for our analysis. For this we created an array of “accuracies[]” which will keep record of all the values of K. The range for the value of K is from 1 to 100 which means the testing data will compared with K number of neighbours in order to predict the outcome for that particular instance.

Logistic Regression

Logistic Regression is a powerful machine learning algorithm widely used for predicting binary outcomes. This is because it serves as a better predictor than a traditional linear regression model.



Naive Bayes Classifier

Types:

Gaussian Naïve Bayes: In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution.

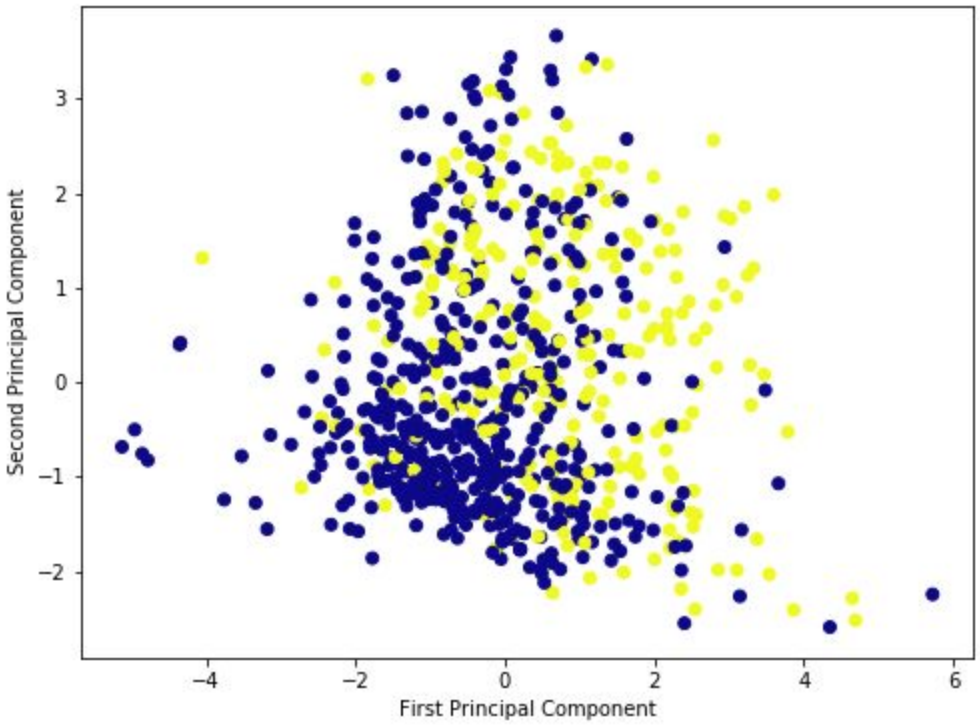
Multinomial Naïve Bayes: It is used for discrete counts.

Bernoulli Naïve Bayes: The binomial model is useful if your feature vectors are binary.

In our case **Gaussian Naive Bayes** is used because the data is normalised and which is evident from the histogram.

Principal Component Analysis

Principal Component Analysis is basically a dimension reduction technique. In this algorithm, an Orthogonal transformation is done to convert a set of observations of possibly correlated variables into a set of values into a set of values of linearly uncorrelated principal components. For our dataset, we had in all 8 independent variables so it is quite obvious that we are going to deal with 8 principal components. But for our analysis we worked on only first two components as it explained nearly 48% of variance within the data. Using PCA technique it is very easy to visualize the data. After considering the two components, we tried to plot the scatter plot diagram in 2D. The result is as shown below:

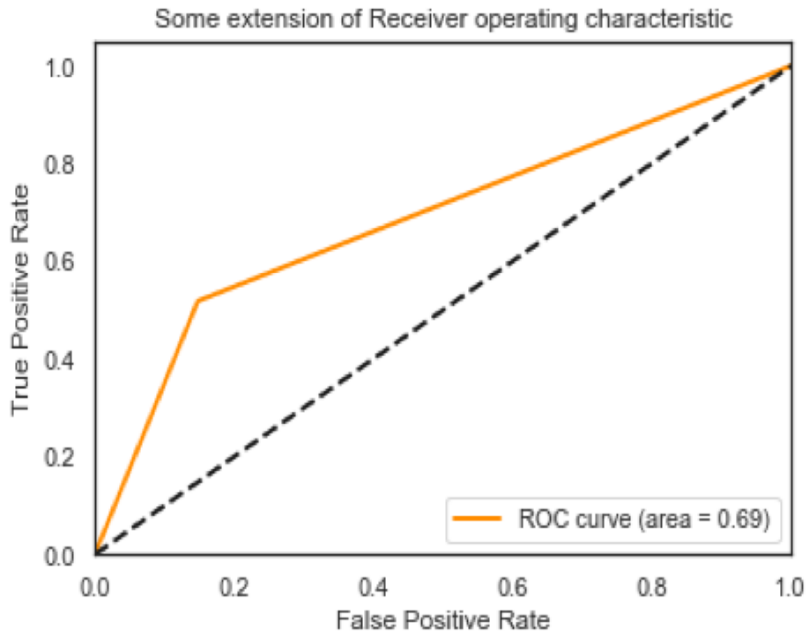


As we can conclude from the graph that the dots which are located to the left on the X-axis belongs to one group and dots to the right on the X-axis belongs to another group.

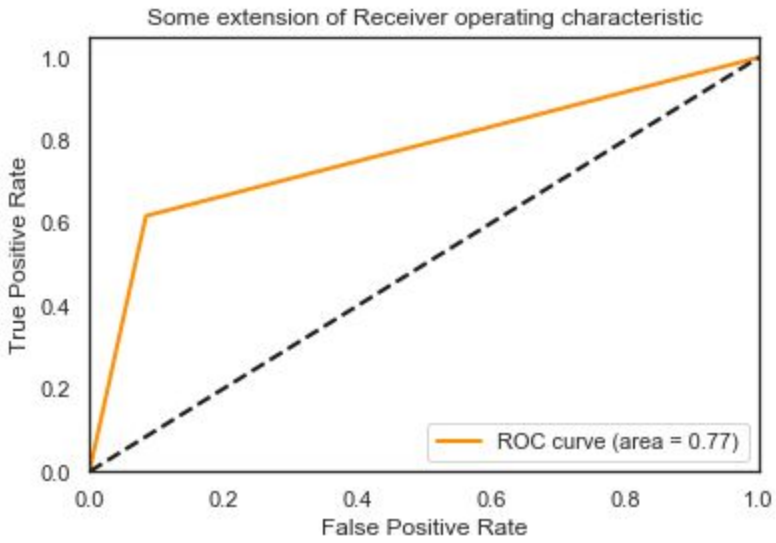
We then tried to train and test the two components and realised that there wasn't any significant difference in the accuracy of different algorithm.

Receiver Operating Characteristics

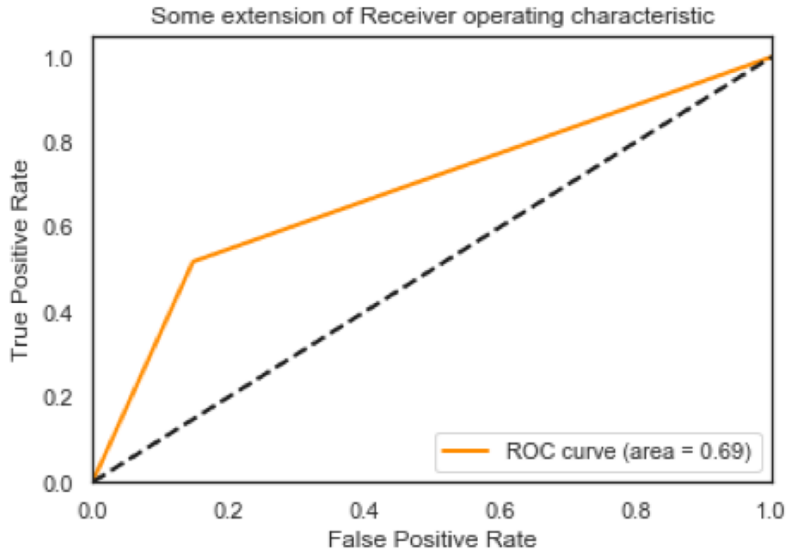
ROC is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate against the false positive rate at various threshold settings. The area under the ROC curve is a measure to how well a parameter can distinguish between two diagnostic groups.



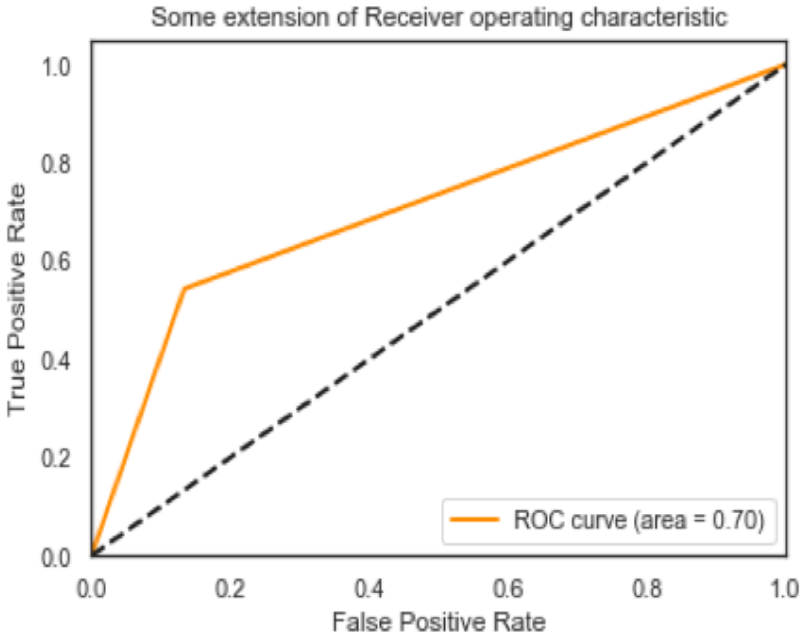
This is the ROC for Naive Bayes



This is the ROC for K Nearest Neighbour



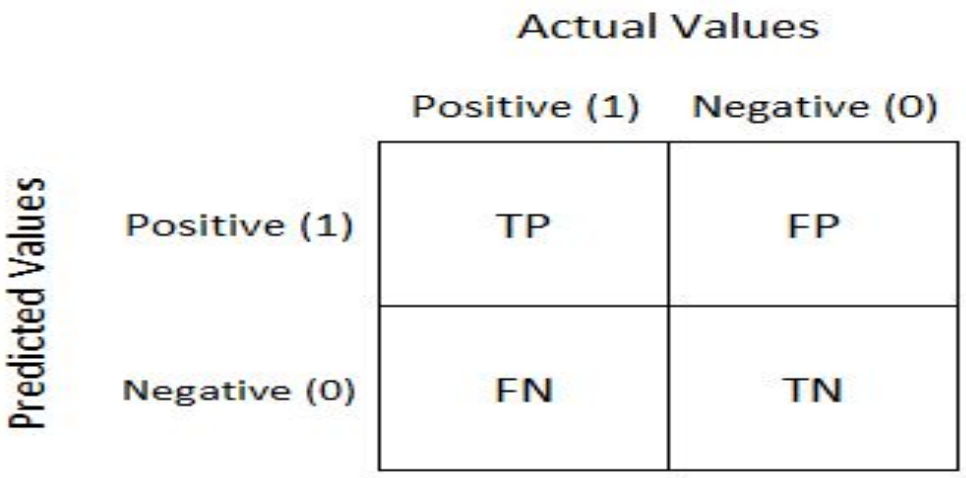
This is the ROC for Linear Discriminant Analysis



This is the ROC for Logistic Regression.

Confusion Matrix

A confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one.



The following can be calculated using the confusion matrix

- $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$
- $\text{Recall} = \frac{TP}{TP+FN}$
- $\text{Precision} = \frac{TP}{TP+FP}$
- $\text{F1-Score} = 2\frac{Precision * Recall}{Precision+Recall}$

Few examples of confusion matrix are as follows:

```
Accuracy of LDA:  0.7847533632286996
The Confusion Matrix for LDA is:
 [[128  14]
 [ 34  47]]


Accuracy of KNN: 0.8116591928251121 when k = 9
The Confusion Matrix for KNN is:
 [[132  10]
 [ 32  49]]


Accuracy of Logistic Regression:  0.7847533632286996
The Confusion Matrix for LR is:
 [[128  14]
 [ 34  47]]


Accuracy of GaussianNB:  0.7802690582959642
The Confusion Matrix for GNB is:
 [[123  19]
 [ 30  51]]
```

K Folds Cross Validation

In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the validation data. The k results can then be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once. 10-fold cross-validation is commonly used, but in general k remains an unfixed parameter.
In our case the value of k=10 that is 10 fold cross validation is used.

========== K-Folds Cross Validation ==========

Linear Discriminant Analysis:
Maximum Accuracy: 0.8909090909090909
Variance in accuracies: 0.060301236761037 82

K Nearest Neighbors:
Maximum Accuracy: 0.8
Variance in accuracies: 0.06374321223640805

Logistic Regression:
Maximum Accuracy: 0.8909090909090909
Variance in accuracies: 0.060301236761037 82

Naive Bayes Classifier:
Maximum Accuracy: 0.8363636363636363
Variance in accuracies: 0.07779435614374863

Cross-Validation Accuracy Before PCA

========== K-Folds Cross Validation after PCA ==========

Linear Discriminant Analysis:
Maximum Accuracy: 0.8
Variance in accuracies: 0.05825267836183487

K Nearest Neighbors:
Maximum Accuracy: 0.7272727272727273
Variance in accuracies: 0.042369581763826235

Logistic Regression:
Maximum Accuracy: 0.8
Variance in accuracies: 0.05509353743603544

GaussianNB:
Maximum Accuracy: 0.8363636363636363
Variance in accuracies: 0.06743440829059912

Cross-Validation Accuracy After PCA

Conclusion

On comparing all the techniques, we can surely say that Gaussian Naïve Bayes Algorithm would work efficiently for this dataset if we reduce the dimensions of the dataset using Principal Component Analysis. IT is also evident from the ROC curve.
Also, we can say that glucose is having a maximum influence on the outcome. So higher the glucose intake more the chances that the person is diabetic.
Gaussian Naïve Bayes algorithm gives the highest accuracy of 83.63% for cross-validation technique after reducing the dimensions. Also it has lowest FP value.

Future Work:
The next step in this analysis would be to continue to add and refine variables. Adding food habits, location identifiers and season identifiers such as zip code can improve the analysis to specific geographies and times. Further refining and transforming variables can add to the model's prediction accuracy. Also testing the results of other machine learning tools such as a decision tree, ensemble techniques, or boosting techniques would be logical next steps to improve prediction.

References:

1. Yoshua Bengio, “Learning Deep Architectures for AI”, volume 2. 2009
2. https://www.ijcaonline.org/archives/volume180/number5/islam-2017-ijca-916020.pdf
3. S. Dewangan.et.al. Int. Journal of Engineering Research and Application www.ijera.com ISSN: 2248-9622, Vol. 8, Issue 1, (Part -II) January 2018, pp.-09-13
4. https://www.researchgate.net/publication/316432650_Diabetes_Prediction_Using_Medical_Data [accessed Dec 04 2018].
5. https://wellness.allinahealth.org/library/content/0/6571
6. https://stackoverflow.com/questions/40758562/can-anyone-explain-me-standards caler
7. http://benalexkeen.com/feature-scaling-with-scikit-learn/
8. https://www.quora.com/In-machine-learning-what-s-the-purpose-of-splitting-data-up-into-test-sets-and-training-sets
9. https://en.wikipedia.org/wiki/Naive_Bayes_classifier#Gaussian_naive_Bayes
10. https://en.wikipedia.org/wiki/Cross-validation_(statistics)
11. McLachlan, Geoffrey J.; Do, Kim-Anh; Ambroise, Christophe (2004). Analyzing microarray gene expression data. Wiley.