Final Report
Early Detection of Diabetes
Springboard Capstone

By Rime Saad

Table of Contents:

Introduction:

Problem Statement:

Data Wrangling:

Exploratory Data Analysis:

Data processing and model selection:

Modeling:

Conclusion:

Introduction:

Diabetes is a disease that is becoming more common. People's unhealthy habits, like diets full of sugar and unhealthy fats in addition to no exercise, is contributing to this increase. The CDC reports that 34.2 million people or 10% of the US population have it. There are 26.9 million people diagnosed and 7.3 million people or 21.4% of the 10% are undiagnosed. These are alarming numbers.

But why is diabetes so dangerous? Diabetes is a condition in which the body does not properly process food for use as energy. When we eat, food gets digested by our bodies and turned into energy in the form of glucose, which in turn gets absorbed by the cells with the help of the hormone insulin. Without insulin glucose stays in the bloodstream and that can cause incredible damage and devastate the body. The problem is that diabetic patients' pancreas does not produce enough insulin or even if it does, their cells do not respond to it, so glucose continues to build up in their cells and this build up can cause heart disease, nerve damage, kidney disease, dementia, eye problems, the list goes on. Patients need to constantly monitor their blood glucose to make sure it does not rise or drop under a certain level.

Problem Statement:

Can this disease be detected early?

There is no cure for diabetes but if detected early, it can be managed successfully and even reversed. The key is to detect it early enough. By using the UCI Early stage diabetes risk prediction dataset I created a tool that helps doctors determine if a patient is at risk of developing diabetes by using only a few symptoms. This dataset has been collected using direct questionnaires from the patients of Sylhet Diabetes Hospital in Sylhet, Bangladesh and approved by a doctor.

It is important to not misclassify the sick so I need to optimise for false negatives or recall. At the same time, healthy patients should not be misclassified as sick, so I need to also optimise for false positives or precision. F1 metric is the harmonic mean of precision and recall so that is the metric that I am going to be looking at in addition to the ROC_AUC curve and the precision recall curve. After running logistic regression, Knearest neighbor, tree algorithms, and RandomForestClassifer, RandomForestClassifer gave the highest f1 score of 100% followed by Knearest neighbor with f1 score of 98% with three cases being misclassified as a false negatives.

Data Wrangling:

This dataset has been collected from the UCI Machine Learning Repository, using direct questionnaires from the patients of Sylhet Diabetes Hospital in Sylhet, Bangladesh and approved by a doctor.

The dataset consisted of 16 variables plus the response variable that indicated if a patient was positive or negative for the disease and there are 520 patients. So this is a classification problem and all the variables were binary except for Age which was continuous. The data set was mostly clean with no missing values.

Exploratory Data Analysis:

Since this is a classification problem I looked at the response variable to see if it was balanced. This is the plot:

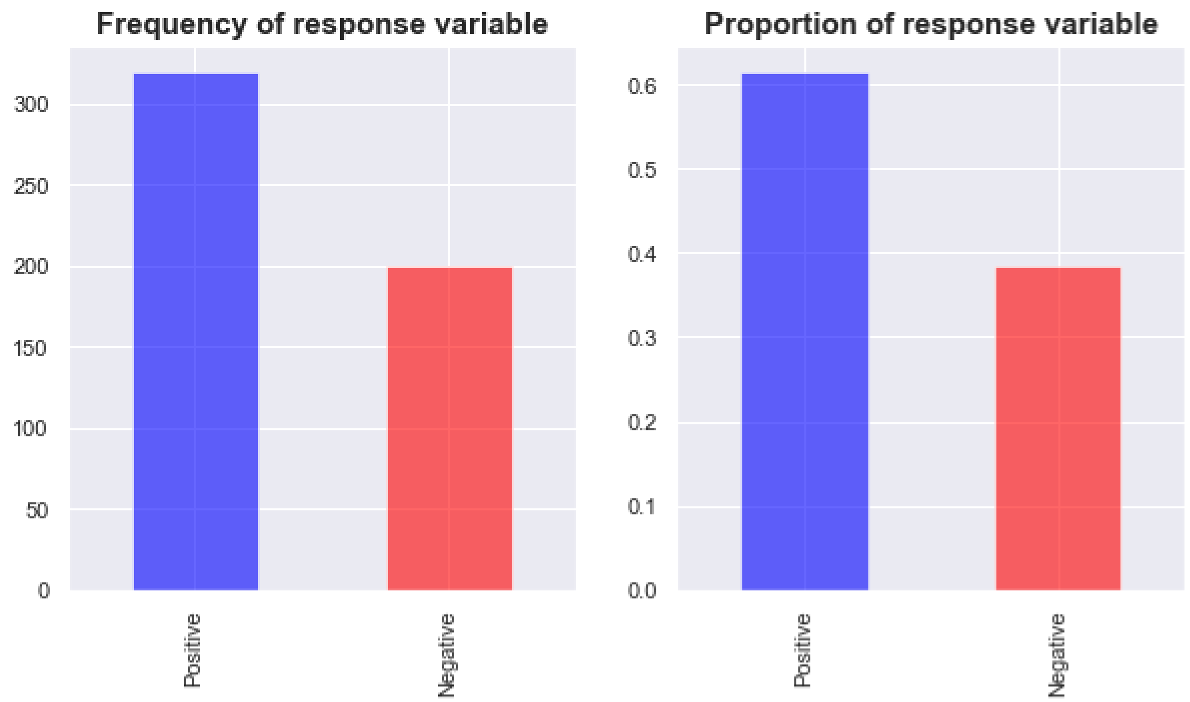

Figure 1: Frequency and proportion of response variable

The dataset is a bit imbalanced which is to be expected, since it was collected in a hospital, so most of the people performing the survey were already not feeling well.

These are all the symptoms that are used in the dataset.

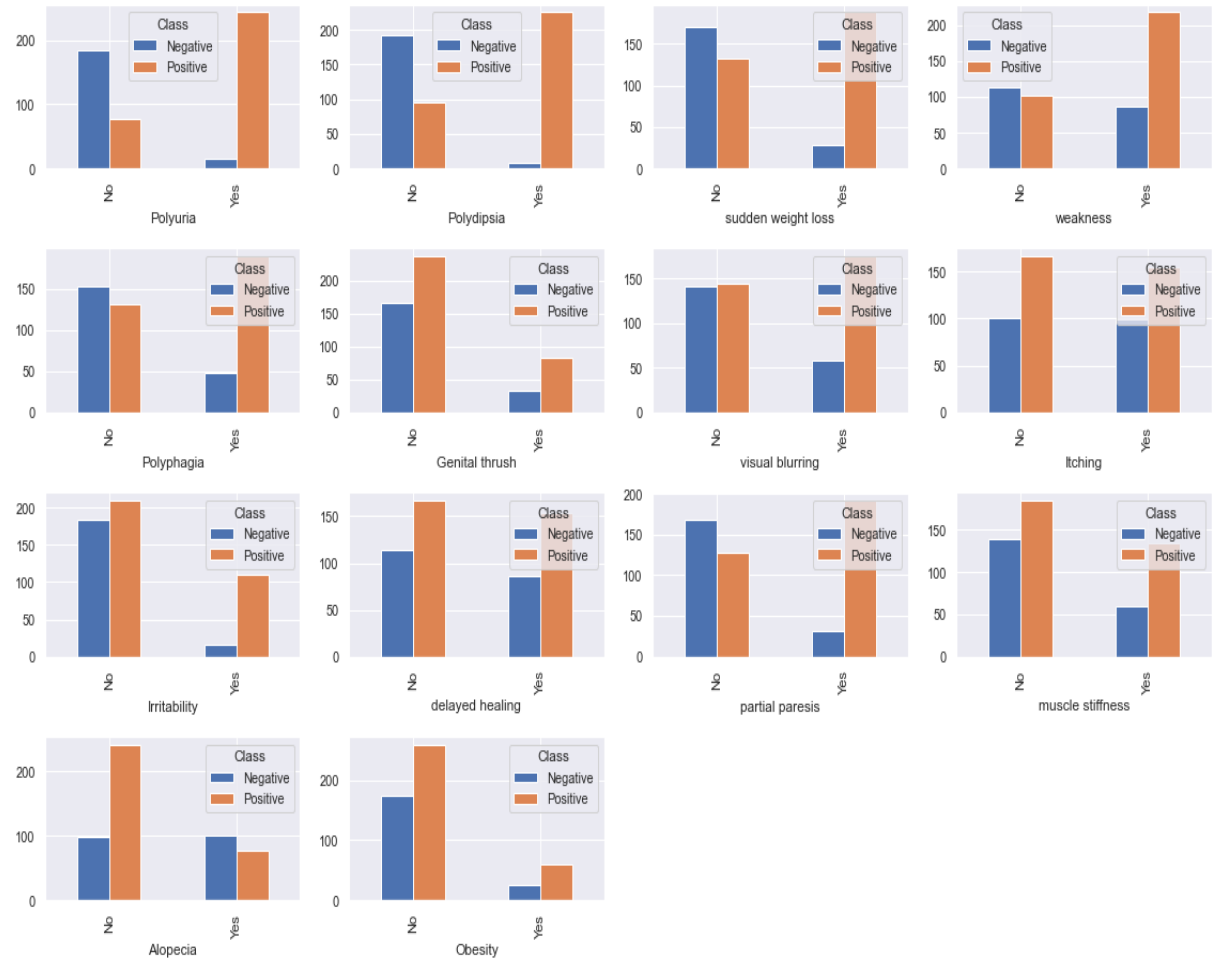

Figure 2: The features and how they are expressed for patients. Negative class is someone who is not sick and positive is someone is sick

Some of the symptoms are more important symptoms than others in telling if someone is sick, like polyuria, which is increased urination, weakness, polyphagia, which means increased hunger, and some of the symptoms are not that common for someone who is diabetic, like alopecia, which means hair loss, and obesity.

In this dataset there are more females than males who are positive, as is shown in figure 3.

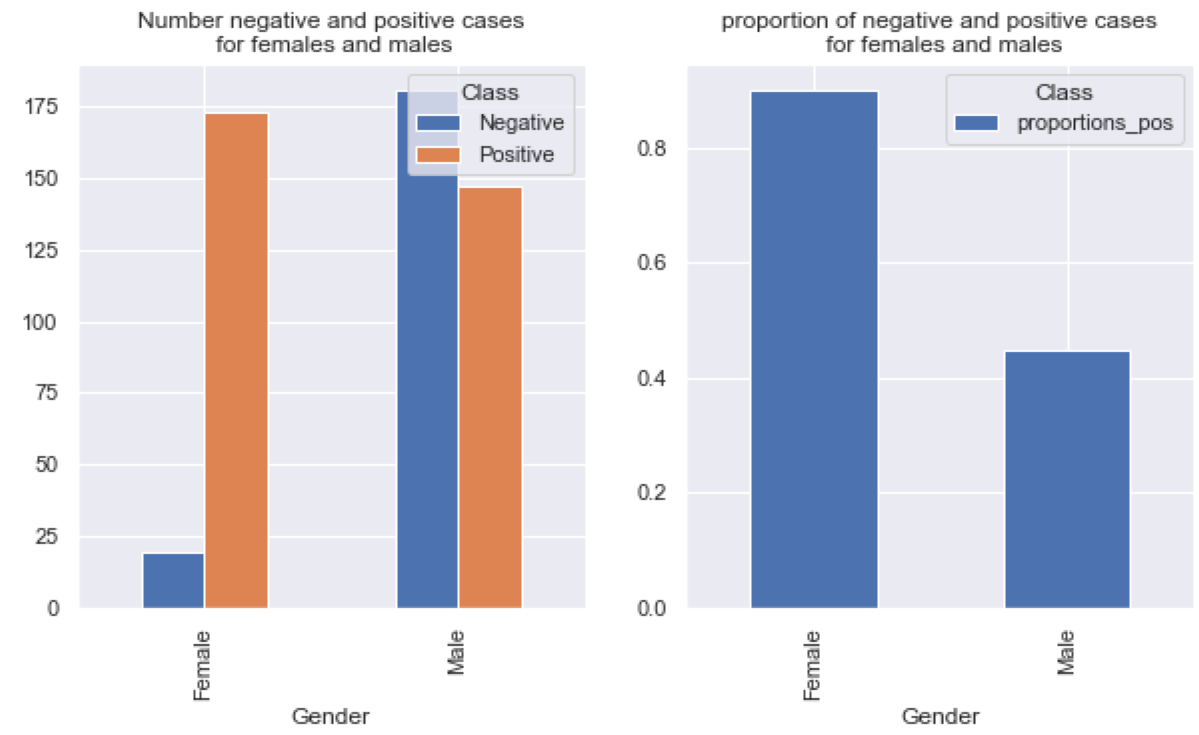

Figure 3: Number and Proportion of females to males who are positive

	Negative	Positive	total_gender
Female	19	173	192
Male	181	147	328

Table 1: Numbers of females to males in dataset

The total number of females is 192 versus 328 males is, so the big difference in proportion of positive women to men could be because there are more males than females. To test the null hypothesis that there is no difference between the fraction of men and women who tested positive, I did a proportions z_test which gave me a significant p_value, which means that I reject the null hypothesis and can deduce that females are more likely to develop diabetes . However there is no literature to support that females are more likely to develop diabetes, unless we consider pregnancy, which can increase the likelihood. I think these results have to do with the dataset and how it was collected. The high number of males to females tested suggests that women might not go to the hospital until they are very sick, whereas men go more frequently.

The second hypothesis to be tested is if there is a relationship between age and getting the disease, so I perform a chi squared test.This is how positive cases are spread across the age groups for the dataset.

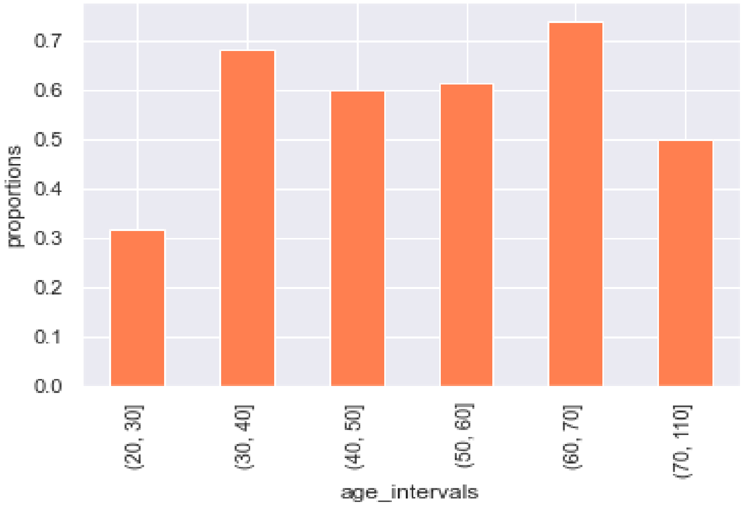

Figure 4: Proportions of positive cases by age

After running the test, the result was significant which means I can reject the null hypothesis that there is no connection, and that age matters. Older individuals are at higher risk of getting the disease and they need to be careful with what they eat and how much they exercise. That's why it is important to be able to predict if someone is at risk early.

Data processing and model selection:

The first thing I did was standardized the age variable and hot encoded all the other variables. The first model I ran was Logistic regression where I tested if there is a difference between standardizing or normalizing the age variable and found that normalizing gives slightly better results in logistic regression so I used normalized data for all the other models.

Then I ran PCA or principal component analysis, to see if some features are more important than others. There was no clear elbow so I used all the variables in the models

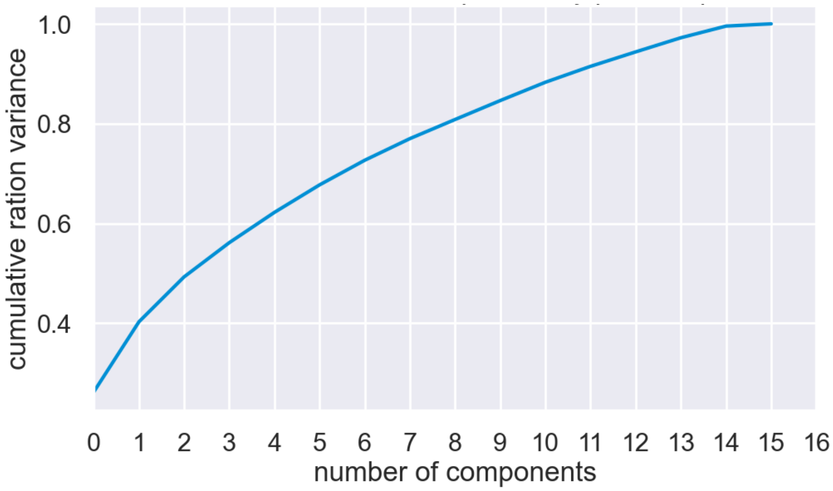

Figure 5: Cumulative variance ratio explained by pca components

Modeling:

The model is supposed to predict if someone is sick, so false negatives have to be eliminated because a sick person should not be predicted to be healthy. Also, a person who is healthy should not be predicted to be sick, since that is a waste of resources and time, so I have to optimize for false positives also. In this case the best metrics to use are precision and recall. f1 score is a great measure of both, since it is their harmonic mean, and the precision_recall curve since there is a little bit of imbalance in the dataset.

The first model I ran was Logistic regression using Gridsearch to find the optimal hyper parameters. This model gave me an f1 score of 96% and a recall of 95% and precision of 97%. The model made 5 mistakes 2 false positives and 3 false negatives.

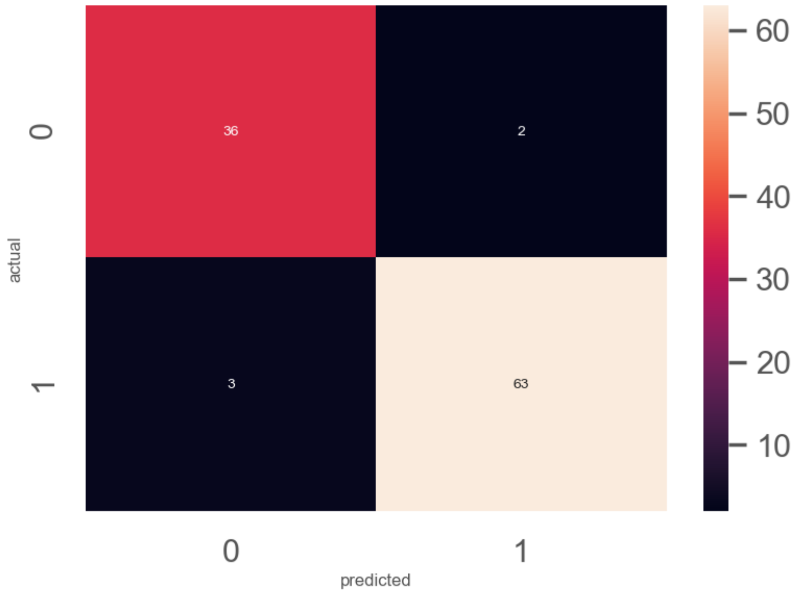

Figure 6: Logistic Regression Classifier confusion Matrix

Next I ran a KNeighbors classifier, using GridSearch for hyperparameter optimization, which scored 100% for precision for positives and 95% for recall and f1 score of 98%.

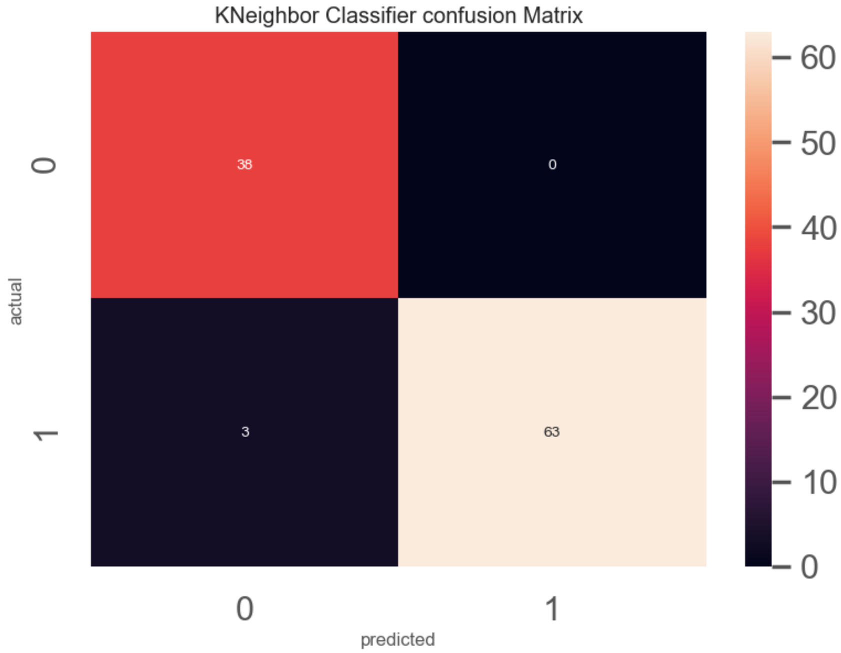

Figure 7: KNeighbor Classifier confusion Matrix

This model made only three mistakes and they were all false negatives. Since I am optimizing for false negatives, I needed to try another model to try to get better results.

Next is decision tree classifier with GridSearch and it scored 97% for precision and 94% for recall and it made 6 mistakes, a much worse result than KNeighbors classifier.

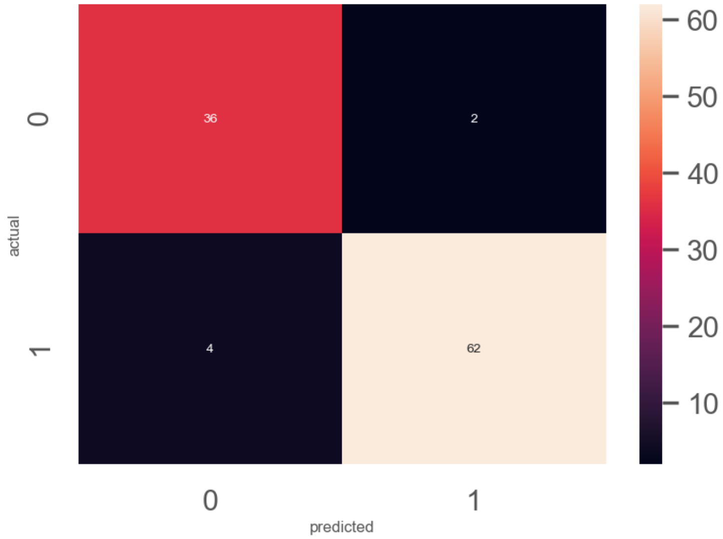

Figure 8: Decision Tree confusion Matrix

Next is RandomForestClassifer using RandomSearchCV to search for the optimal hyperparameters. This classifier gave me a 100% recall and 100% precision.

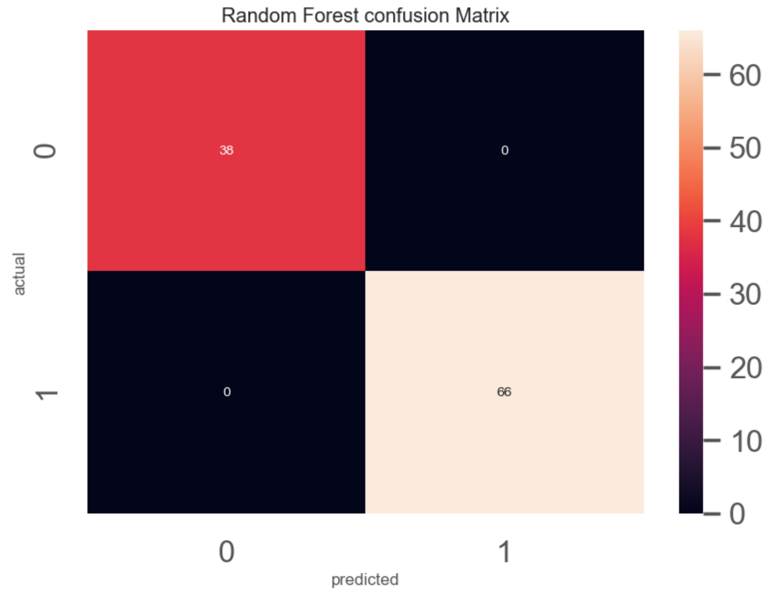

Figure 9: Random Forest confusion Matrix

Obviously this is the best algorithm so far since it gava a 100% results, and made no mistakes.

This is how the ROC curve looks for all the models used

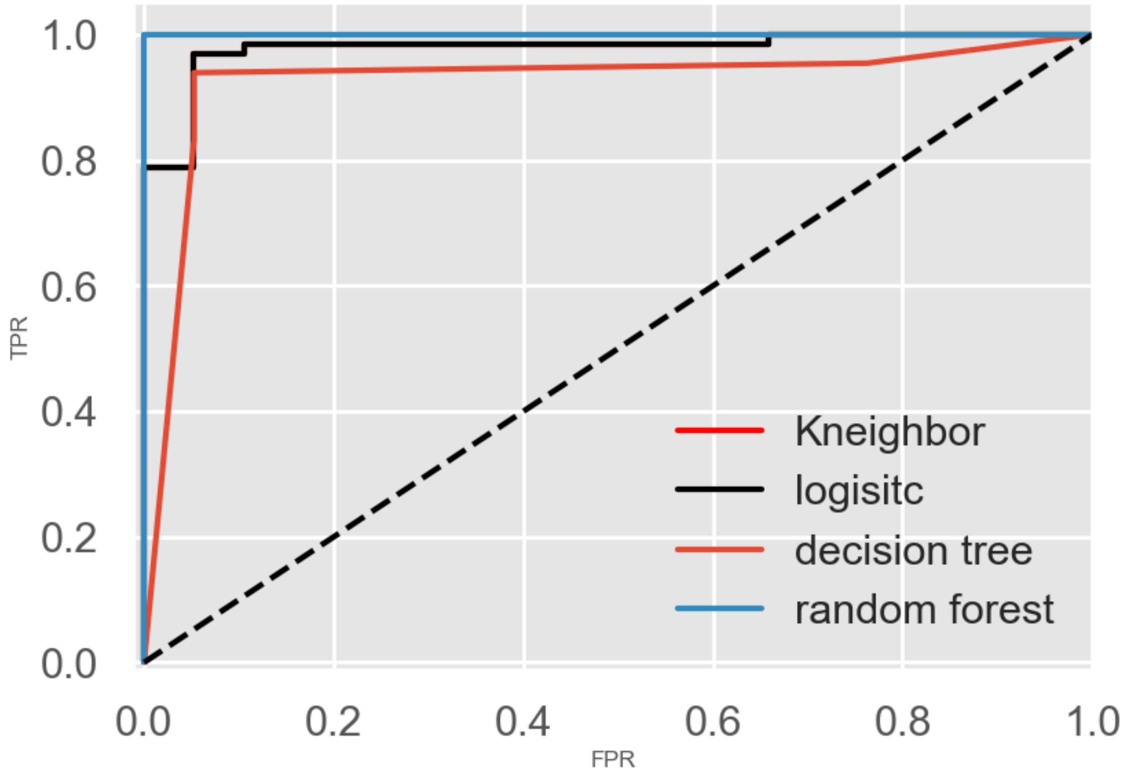

Figure10: ROC curve for all models

And this is the precision recall curve:

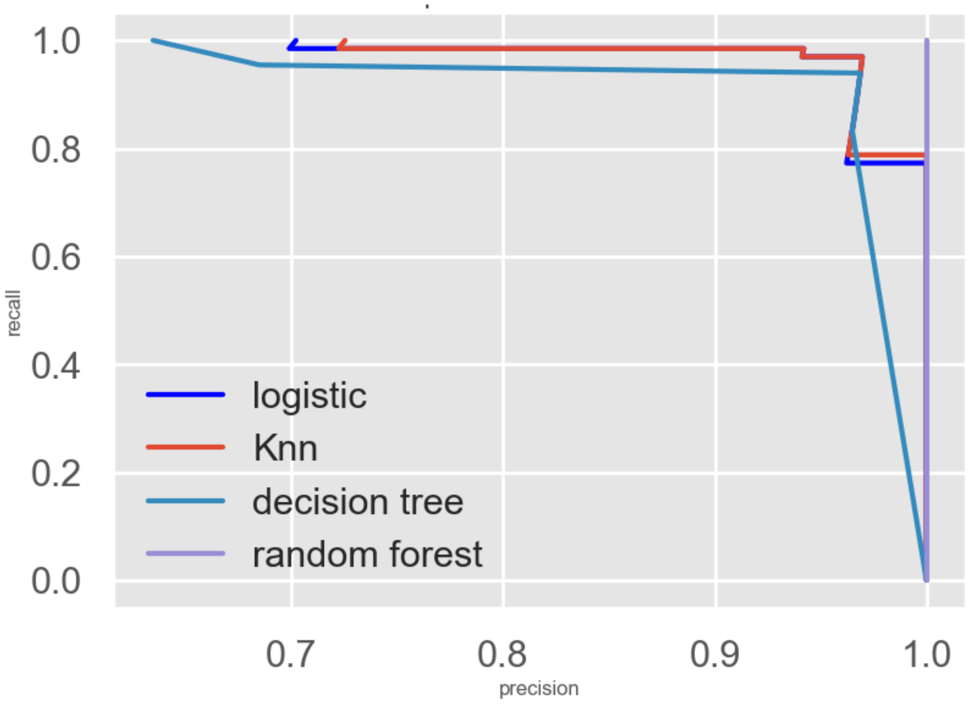

Figure 11: precision recall curve

It is obvious that the best roc curve and the best precision_recall curve is the one belonging to the random forest model. I also tried the same model but using Age as a categorical variable instead of continuous and it gave the same results.

Conclusion:

The data set was somewhat clean, which allowed me to explore more than one model and experiment with their hyperparameters. However, I would like to add blood and insulin tests to the dataset and test it on a larger number of patients to improve its efficacy. In addition, when the models were run on different training and test sets, they gave slightly different results with Kneighbors classifier sometimes outperforming Random forest. It is better to run these two models a 100 times on 100 different train and test sets and then perform a t_test to see if there is a significant difference between the mean of the metrics. This will give a more accurate result