Predicting Diabetes Using Medical Parameters

By

Muhammad Umar Salman

2 Credit Hours

Directed Research Project

2020

Lahore University of Management Sciences (LUMS)

Table of Contents

ABSTRACT

The aim of this study is to look at the medical parameters and the nutritional consumption behavior of individuals and see whether we are able to find some correlation either among these features or find some sort of causation for the presence of diabetes. Diabetes is a disease that may arise when the levels blood sugar in your body are too high. Basic **exploratory data analysis** is done to analyze different features and their distribution for 8 different type of medical parameters. The study also using these parameters tries to find the underlying structure of the data for any meaningful results using **cluster analysis**. It then further looks at the **predictive models** and compares them to see which performs better with the data at hand.

INTRODUCTION

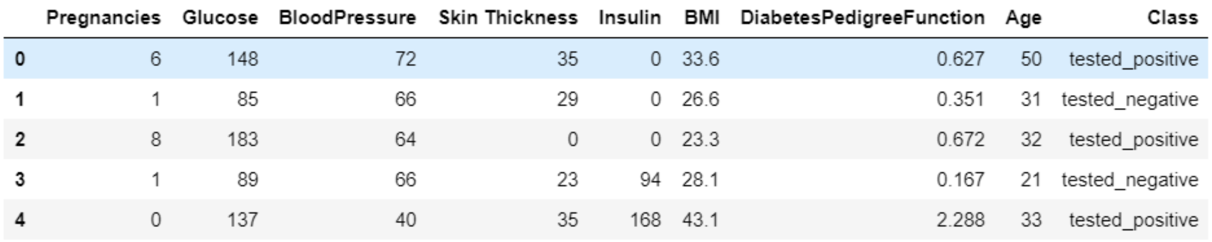
What is diabetes?

Diabetes is a disease that occurs when your blood glucose, also called blood sugar, is too high. Blood glucose is your main source of energy and comes from the food you eat. Insulin, a hormone made by the pancreas, helps glucose from food get into your cells to be used for energy. Sometimes your body doesn't make enough—or any—insulin or doesn't use insulin well. Glucose then stays in your blood and doesn't reach your cells. Diabetes causes many fatal health problems and has no cure, that is why it is necessary that it is caught before things get out of hand. There are 3 types of diabetes and they are the following:

1. Type 1 Diabetes
2. Type 2 Diabetes
3. Gestational Diabetes

The Dataset

The data that the study was initially supposed to use has still not been received as of yet. Thus, the following has been done using the diabetes dataset originated from the University of California, Irvine (UCI)'s Machine Learning Repository. This dataset only consisted of a few medical parameters which is why no such analysis could be conducted on the nutritional consumption taken by an individual. The dataset consists of 8 medical parameters and one column which identifies if the individual has **Positive** or **Negative** diabetes. The dataset has 768 rows each identified by different individuals and 9 columns (8 Medical Parameters and 1 Diabetes Status). The figure below shows a screenshot of the first 5 rows of the dataset.



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

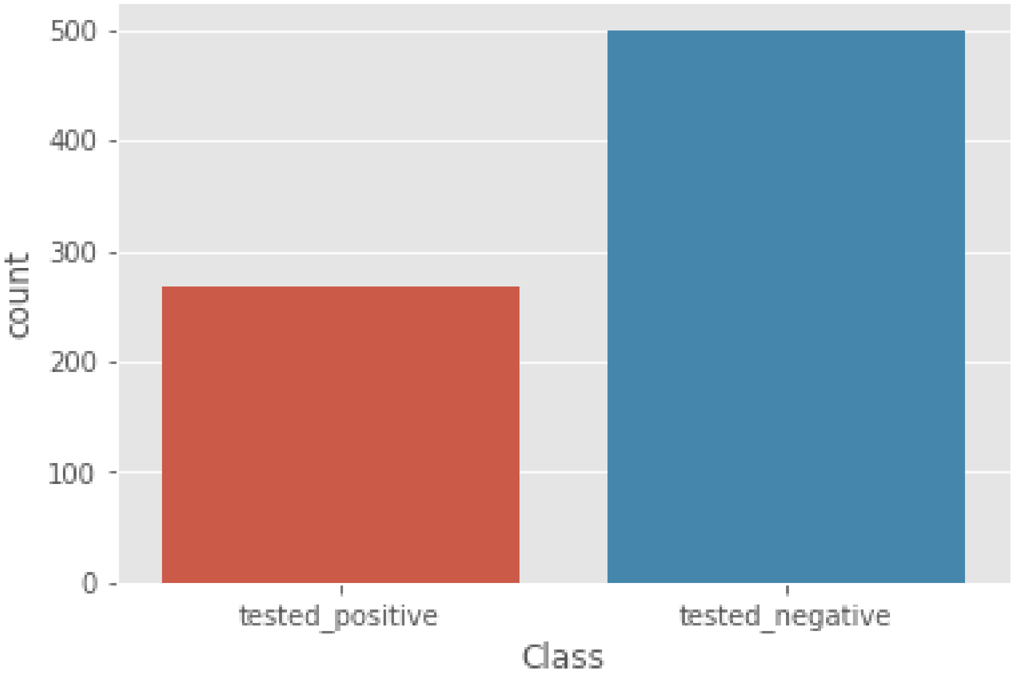
The list below shows the columns:

1. Pregnancies	Count of Pregnancies
2. Glucose	(mg/dL)
3.Blood Pressure	(mm Hg)
4.Skin Thickness	(mm)
5.Insulin	(mlU/L)
6.BMI	Body Mass Index (Kg/m^2)
7.Diabetes Pedigree Function	Probability of having diabetes or not
8.Age	-

There are some confusions within the data as they were not mentioned. Such as we are not sure whether the Glucose levels are after fasting or after a meal as the normal range for their values shift under different circumstances. The same case goes with Insulin levels. Similarly, it isn't mention if the blood pressure measured is the Systolic (Upper number) or Diastolic (Lower number), however, from the distribution of values it can be seen that it is most like to be Diastolic. Further, it isn't specified what variables were used to calculate the Diabetes Pedigree Function which could be misleading.

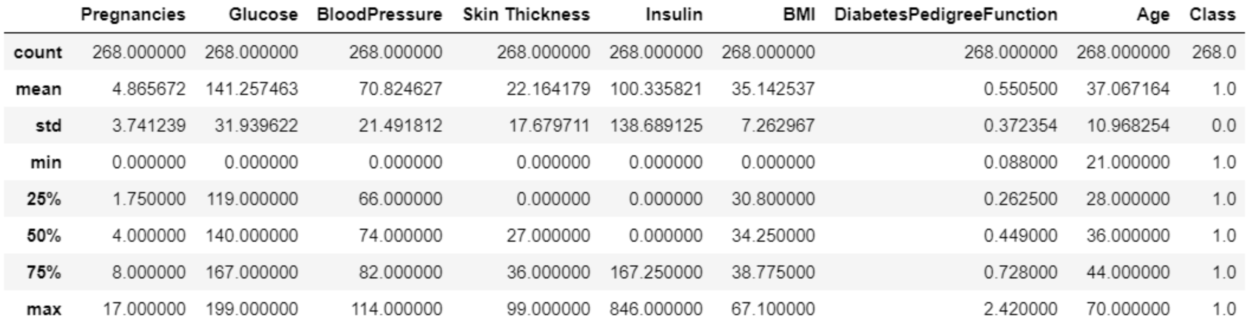
EXPLORATARY DATA ANALYSIS

Here we will look and the distributions and observable patterns within the data and see if we there is some meaningful information which can be extracted. The figure below shows the counts of positive and negative diabetes records in the data. (Positive: 268, Negative: 500)



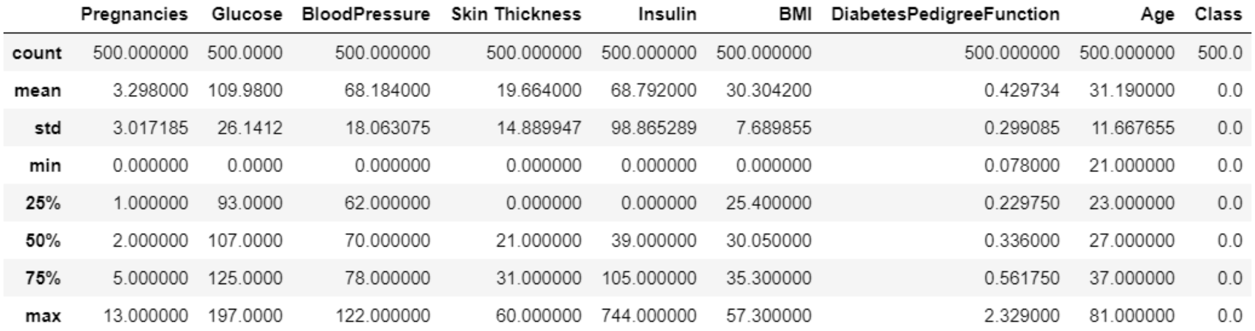
For both classes positive and negative, we will see some basic statistics for each medical parameter.

Positive:



	Pregnancies	Glucose	BloodPressure	Skin Thickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Class
count	268.000000	268.000000	268.000000	268.000000	268.000000	268.000000	268.000000	268.000000	268.0
mean	4.865672	141.257463	70.824627	22.164179	100.335821	35.142537	0.550500	37.067164	1.0
std	3.741239	31.939622	21.491812	17.679711	138.689125	7.262967	0.372354	10.968254	0.0
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.088000	21.000000	1.0
25%	1.750000	119.000000	66.000000	0.000000	0.000000	30.800000	0.262500	28.000000	1.0
50%	4.000000	140.000000	74.000000	27.000000	0.000000	34.250000	0.449000	36.000000	1.0
75%	8.000000	167.000000	82.000000	36.000000	167.250000	38.775000	0.728000	44.000000	1.0
max	17.000000	199.000000	114.000000	99.000000	846.000000	67.100000	2.420000	70.000000	1.0

Negative:



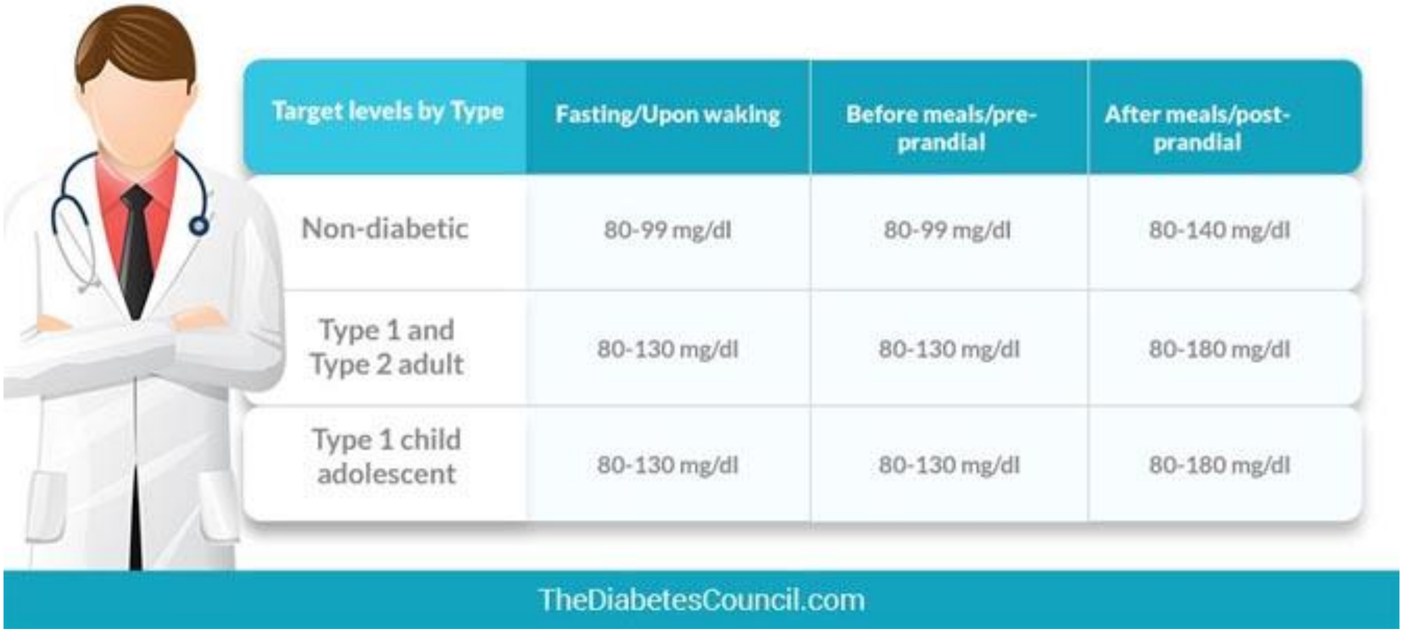
	Pregnancies	Glucose	BloodPressure	Skin Thickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Class
count	500.000000	500.0000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.0
mean	3.298000	109.9800	68.184000	19.664000	68.792000	30.304200	0.429734	31.190000	0.0
std	3.017185	26.1412	18.063075	14.889947	98.865289	7.689855	0.299085	11.667655	0.0
min	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.0
25%	1.000000	93.0000	62.000000	0.000000	0.000000	25.400000	0.229750	23.000000	0.0
50%	2.000000	107.0000	70.000000	21.000000	39.000000	30.050000	0.336000	27.000000	0.0
75%	5.000000	125.0000	78.000000	31.000000	105.000000	35.300000	0.561750	37.000000	0.0
max	13.000000	197.0000	122.000000	60.000000	744.000000	57.300000	2.329000	81.000000	0.0

From the above results we can see some very obvious relations with the presence and absence of diabetes. Firstly, we can see that in positive cases the mean values for **Glucose, BMI, DPF, Skin Thickness and Age** are all significantly higher than those of negative cases. We can see that **Pregnancies** is also higher, but this analysis can be only made if we know the total amount of females in both cases which isn’t mentioned. Next we can see that the **Insulin** levels are also higher however that seems like an error in data as facts shows that Insulin levels are much lower for non-diabetic individuals, yet the data summary shows something different. We can confirm our hypothesis by looking at the 25% and 50% which show that their values are also 0. This led to the idea that maybe since there were no missing values all missing values may be filled as zeros. Below we will see the count of zeros for each feature.

Pregnancies:	111
Glucose:	5
Blood Pressure:	35
Skin Thickness:	227
Insulin:	374
BMI:	11
DiabetesPedigreeFunction:	0
Age:	0

From the list above we can see that there are a lot of erroneous entries in the form of 0's in the data. From domain knowledge we know that values of **Glucose, Blood Pressure, Skin Thickness, Insulin and BMI** can never be 0. We will now see the normal ranges for each of these features to confirm that they can't be marked as 0 and that in fact they are missing values. As for BMI which follows the formula of Weight divided by Height, a 0 in that case would imply that the weight is 0 which is impossible. Hence proven that for BMI you can't have a 0 value. As for skin thickness the subcutaneous tissue **thickness range** in males is from 1.65 mm to 14.65 mm, whereas it is from 3.30 mm to 18.20 mm in females, but no case can a skin thickness value be 0.
The following ranges for the other 3 can be seen below.

Glucose:

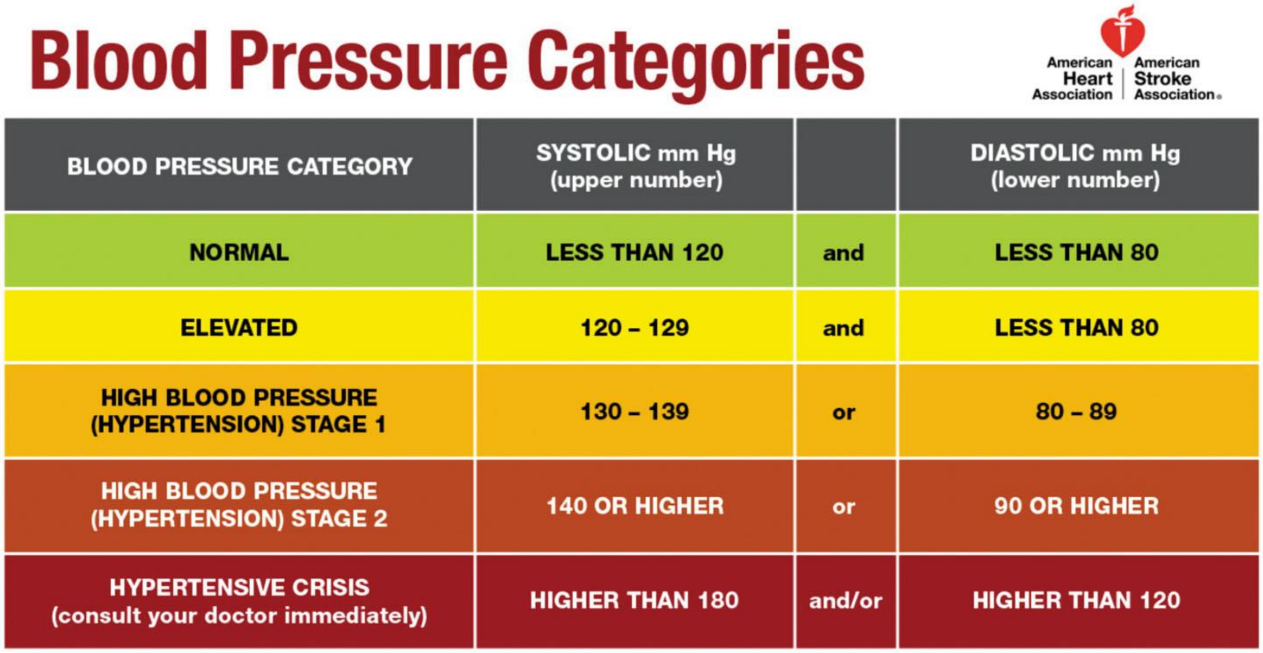


Target levels by Type	Fasting/Upon waking	Before meals/pre-prandial	After meals/post-prandial
Non-diabetic	80-99 mg/dl	80-99 mg/dl	80-140 mg/dl
Type 1 and Type 2 adult	80-130 mg/dl	80-130 mg/dl	80-180 mg/dl
Type 1 child adolescent	80-130 mg/dl	80-130 mg/dl	80-180 mg/dl

TheDiabetesCouncil.com

We can clearly see from the tables that in no range does a 0 value exist for glucose. However, from this we can observe another limitation within our data. The fact that we are unaware of how long before the person had a meal matters a lot as in what values you get for glucose levels.

Blood Pressure:

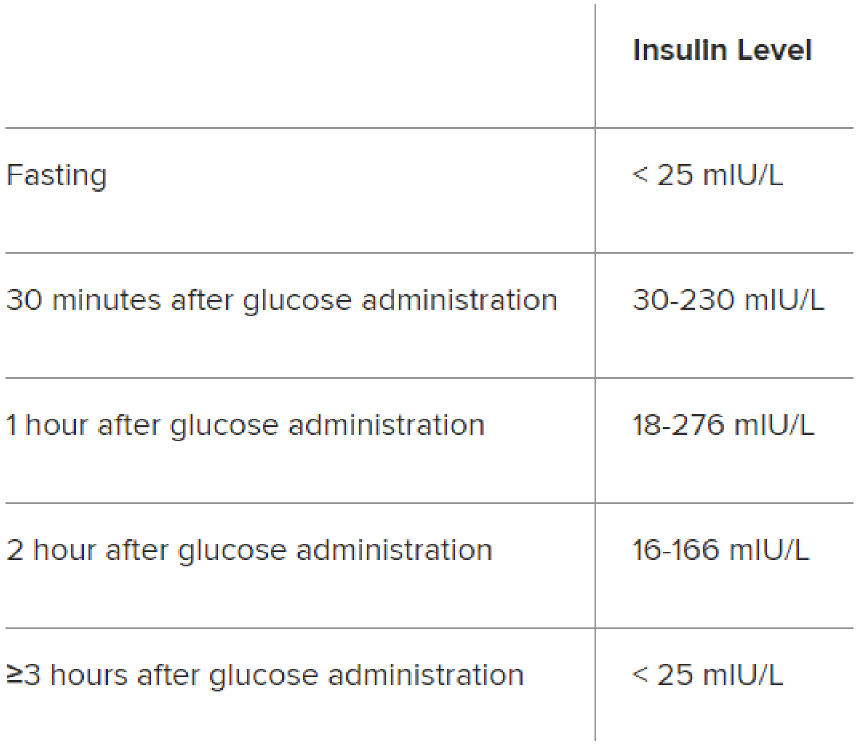


Blood Pressure Categories

American Heart Association | American Stroke Association

BLOOD PRESSURE CATEGORY	SYSTOLIC mm Hg (upper number)		DIASTOLIC mm Hg (lower number)
NORMAL	LESS THAN 120	and	LESS THAN 80
ELEVATED	120 – 129	and	LESS THAN 80
HIGH BLOOD PRESSURE (HYPERTENSION) STAGE 1	130 – 139	or	80 – 89
HIGH BLOOD PRESSURE (HYPERTENSION) STAGE 2	140 OR HIGHER	or	90 OR HIGHER
HYPERTENSIVE CRISIS (consult your doctor immediately)	HIGHER THAN 180	and/or	HIGHER THAN 120

Insulin:

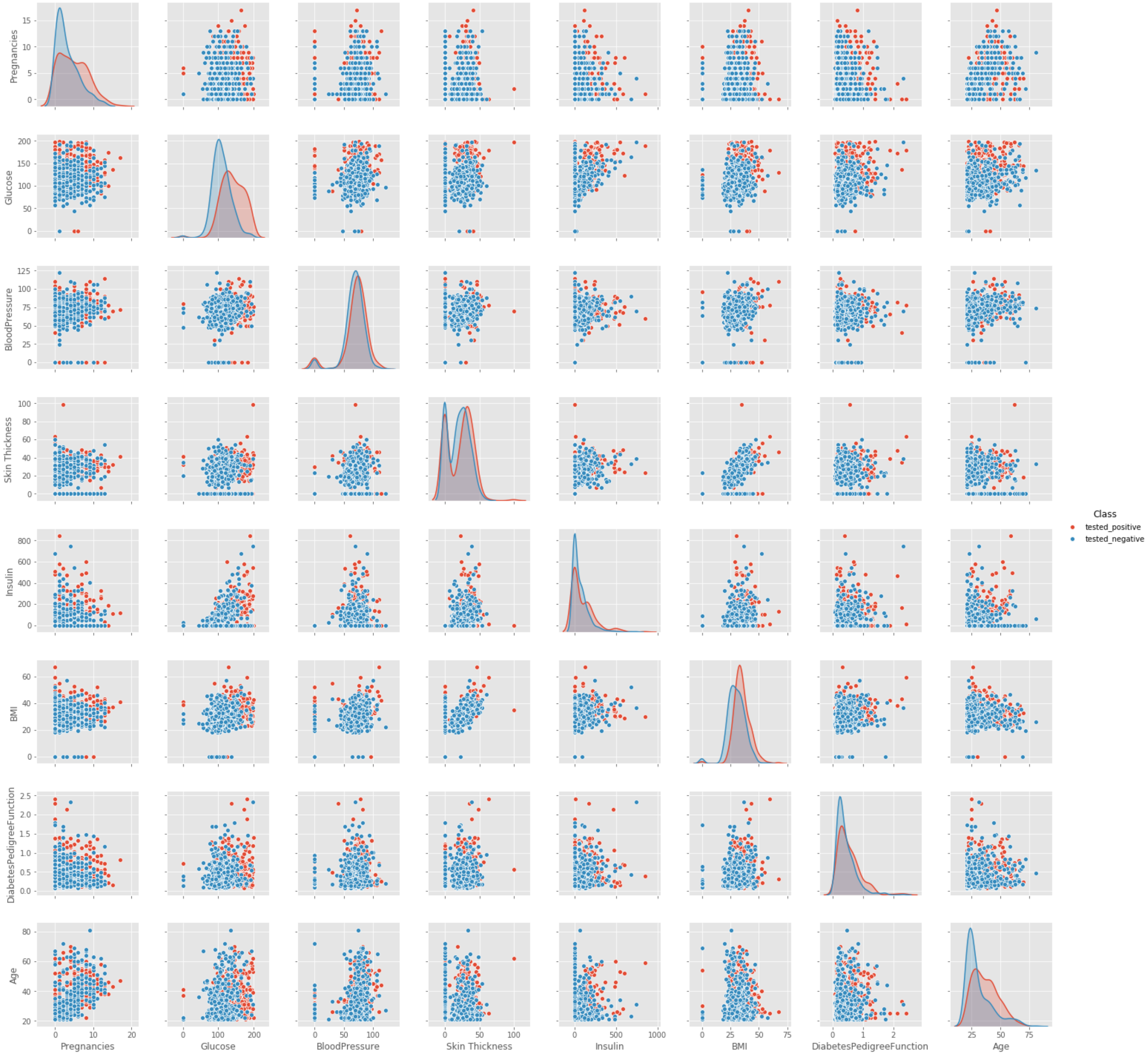


	Insulin Level
Fasting	< 25 mIU/L
30 minutes after glucose administration	30-230 mIU/L
1 hour after glucose administration	18-276 mIU/L
2 hour after glucose administration	16-166 mIU/L
≥3 hours after glucose administration	< 25 mIU/L

Again, we can see that there are no such 0 values for Blood Pressure or Insulin in any range. Even for a fasting insulin level should never be 0, It shouldn't go below 3. However, there are some case where the insulin levels may drop to 0 for Type 1 Diabetes but a count of 374 0s with only 268 positive records shows these are in fact missing entries filled as 0. This also creates doubt for the features where 0's are acceptable if they are all filled or not.

However, if we remove all unique records with missing values or in this case 0 values, we would lose out on 376 records which is almost half of the data that is present. So, most analysis, clustering and models are created from the whole dataset present. We will now see the relations of each feature with the other using a **pair plot** for the two different classes. Different features are compared by scatter plots and the same one by Kernel Density Estimates.

PAIR PLOT



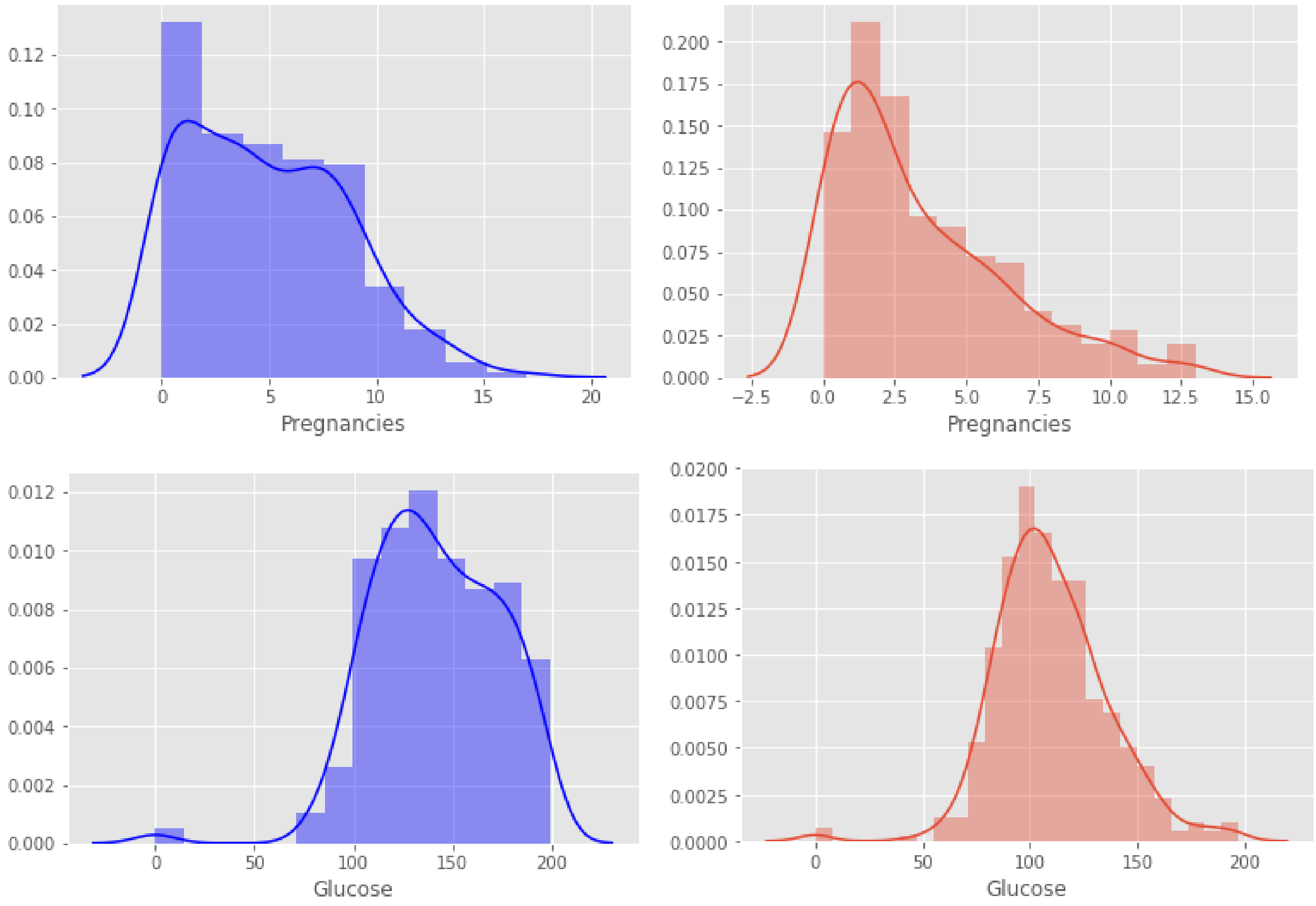
Red: Positive

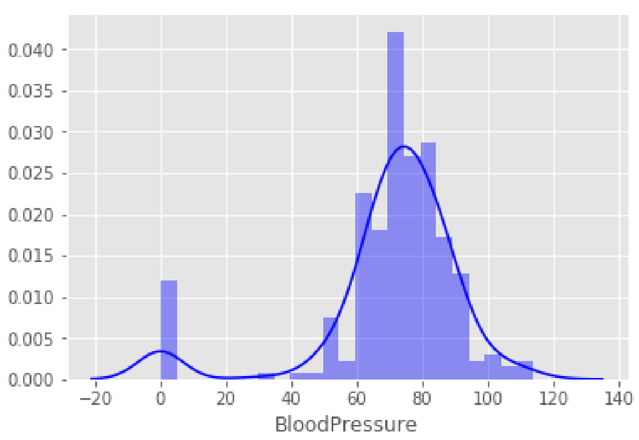
Blue: Negative

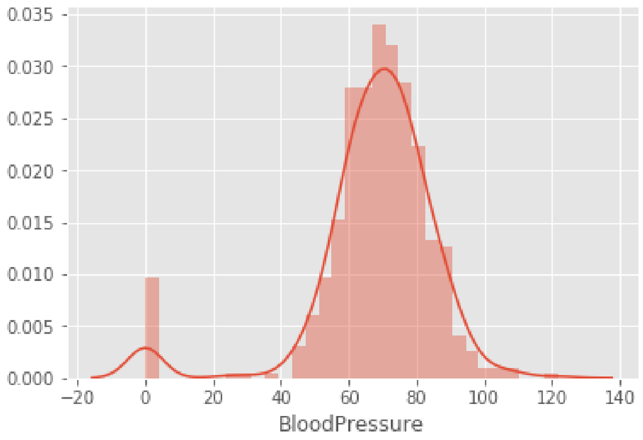
We can see that most values show the same distribution from the plot above, there are only very few extreme positive case values which can be seen. To get a better idea we will compare **distributions** for positive and negative cases.

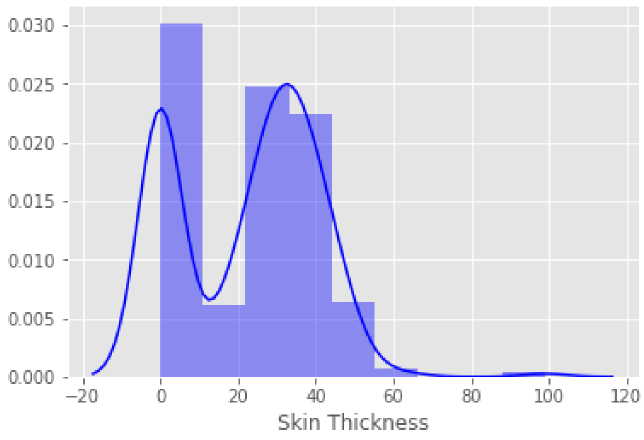
Red: Negative
Blue: Positive

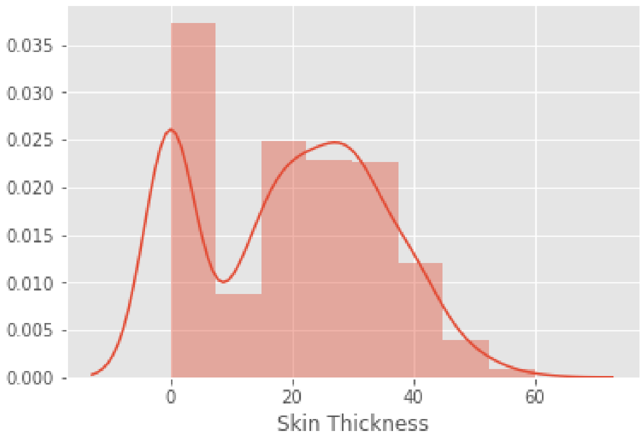
DISTRIBUTIONS

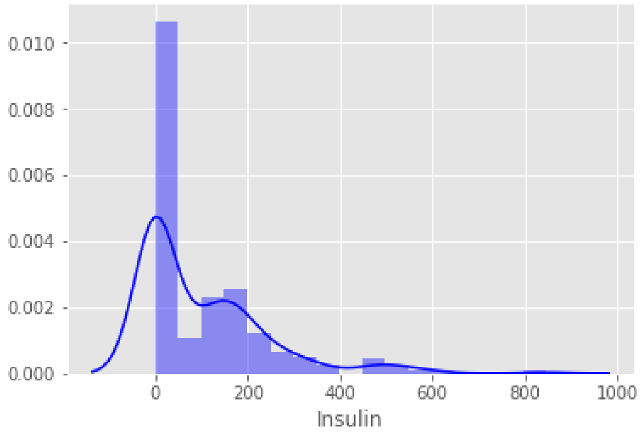


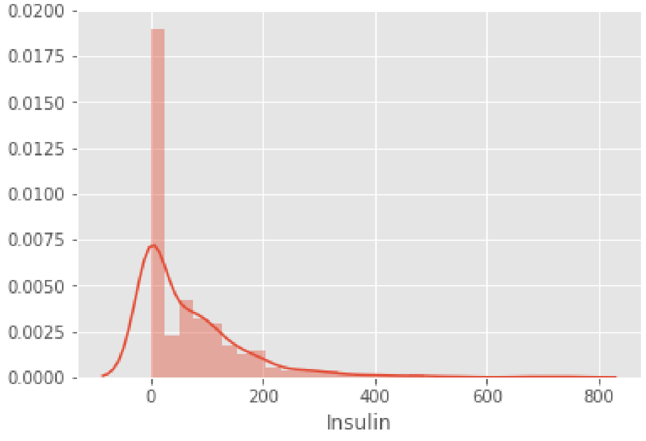

0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
−20
0
20
40
60
80
100
120
140
BloodPressure

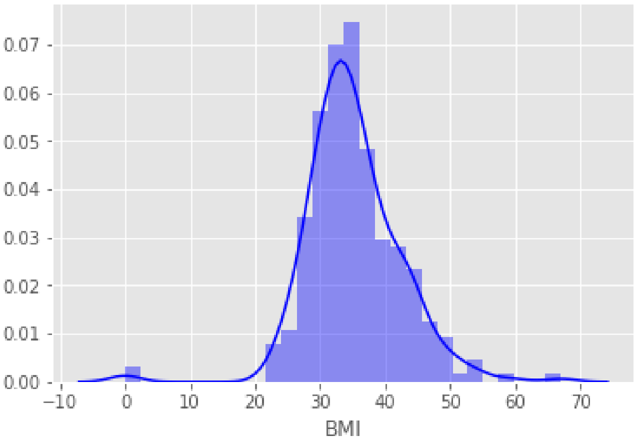

0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
−20
0
20
40
60
80
100
120
140
BloodPressure

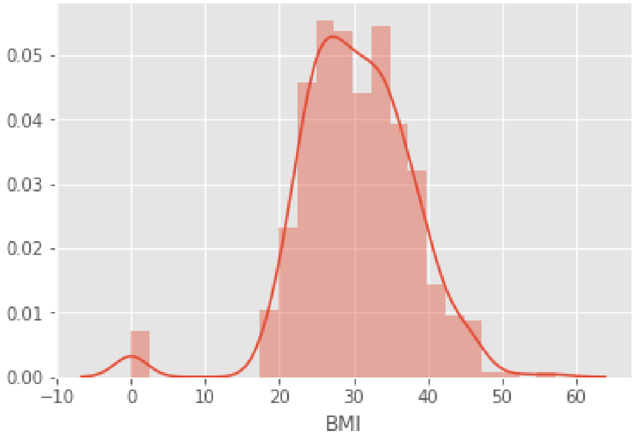

0.030
0.025
0.020
0.015
0.010
0.005
0.000
−20
0
20
40
60
80
100
120
Skin Thickness

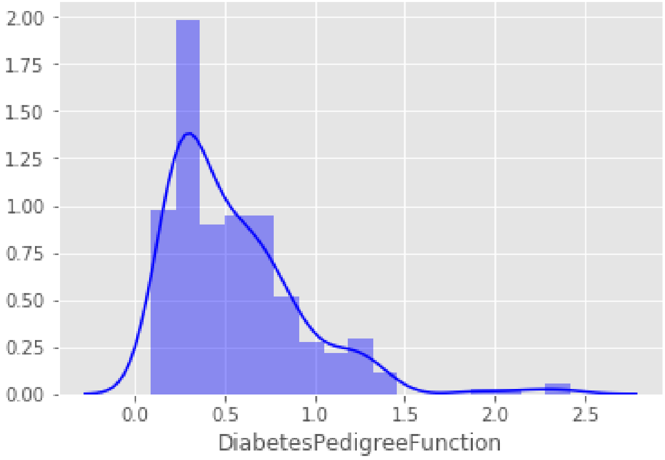

0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
20
40
60
Skin Thickness

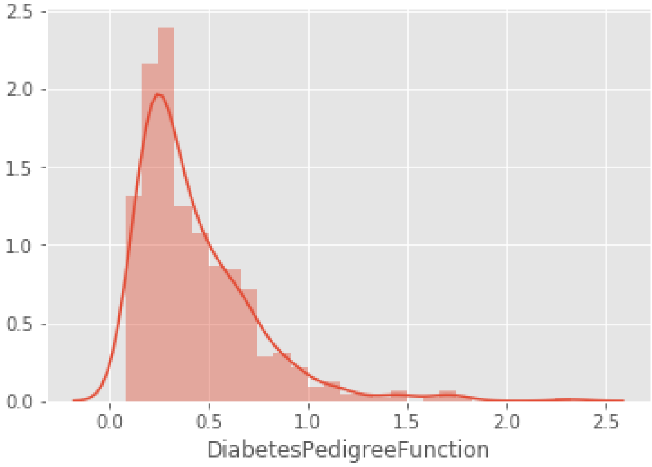

0.010
0.008
0.006
0.004
0.002
0.000
0
200
400
600
800
1000
Insulin

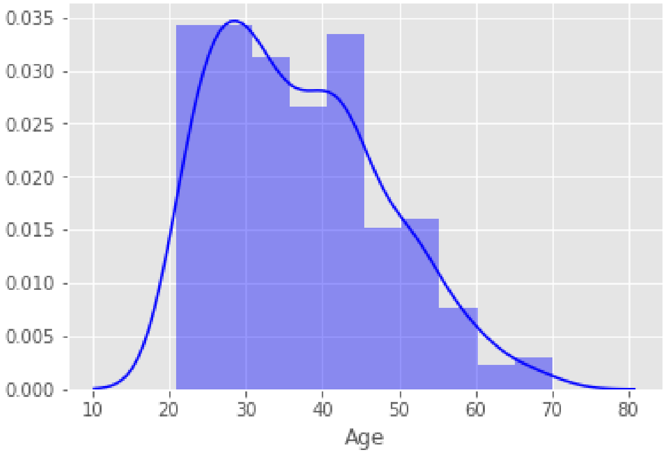

0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
200
400
600
800
Insulin

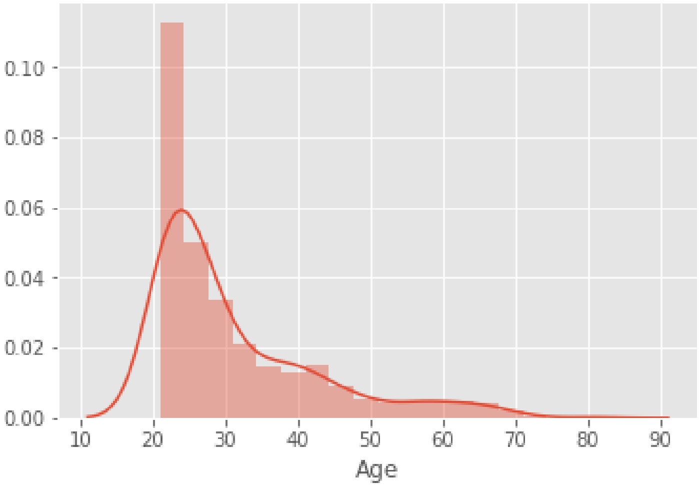

0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
−10
0
10
20
30
40
50
60
70
BMI


0.05
0.04
0.03
0.02
0.01
0.00
−10
0
10
20
30
40
50
60
BMI


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0
0.5
1.0
1.5
2.0
2.5
DiabetesPedigreeFunction

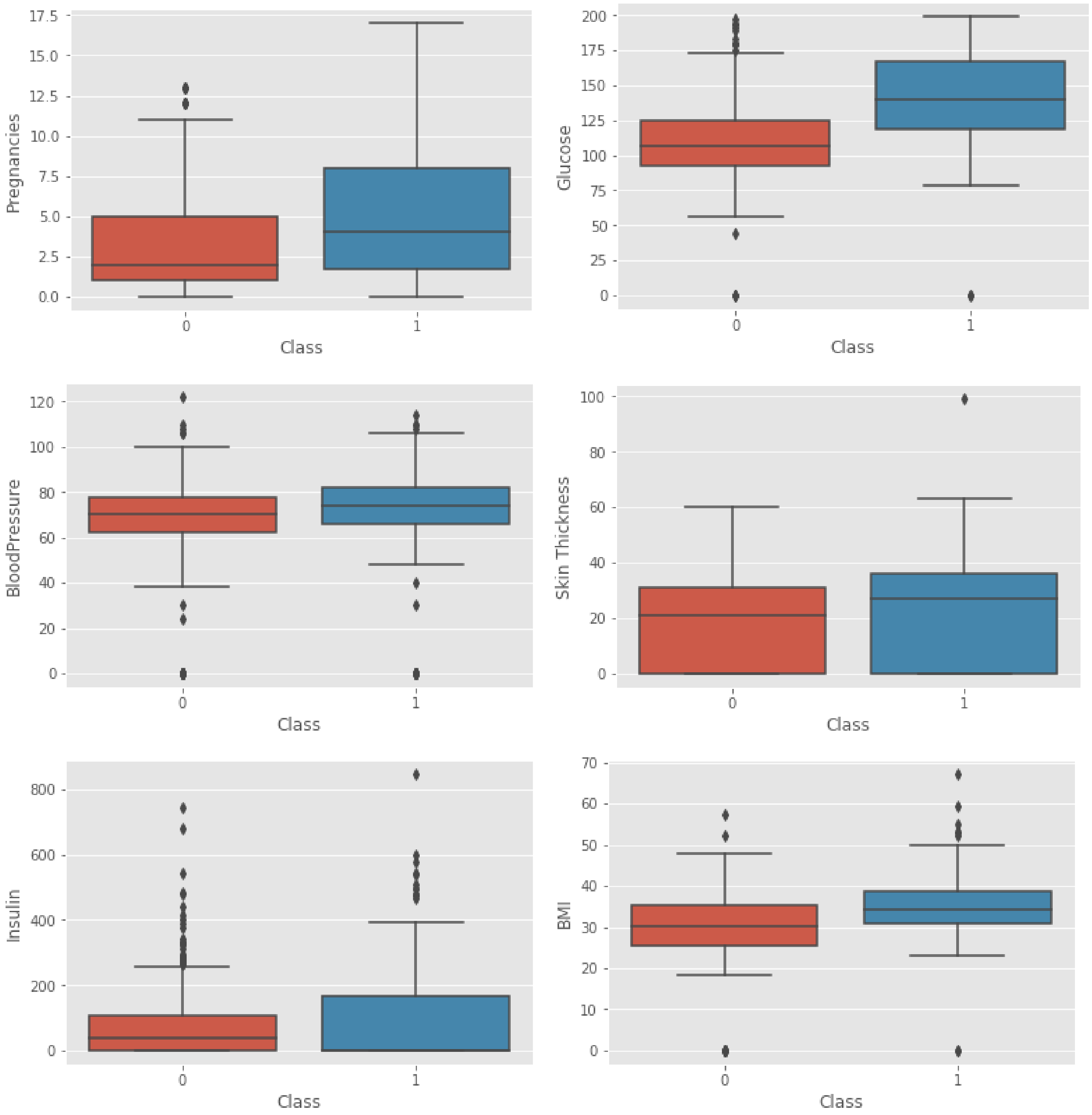

2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
DiabetesPedigreeFunction

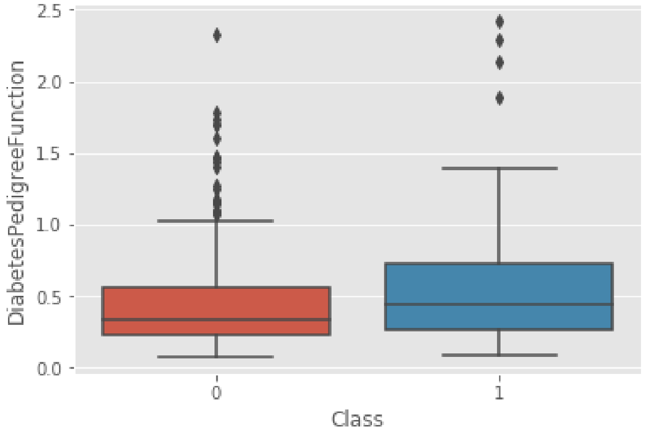

0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
10
20
30
40
50
60
70
80
Age

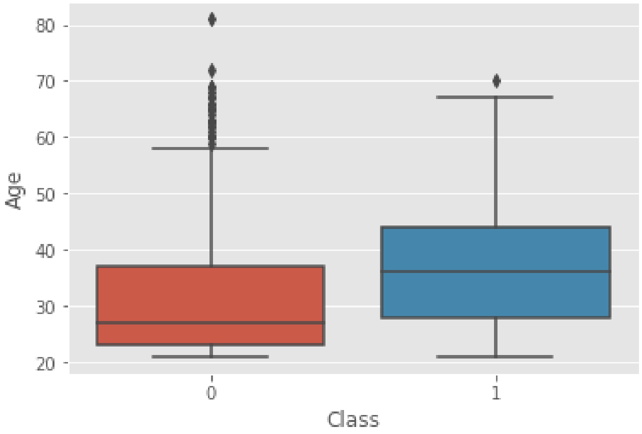

0.10
0.08
0.06
0.04
0.02
0.00
10
20
30
40
50
60
70
80
90
Age

BOX PLOTS

Red: Negative
Blue: Positive



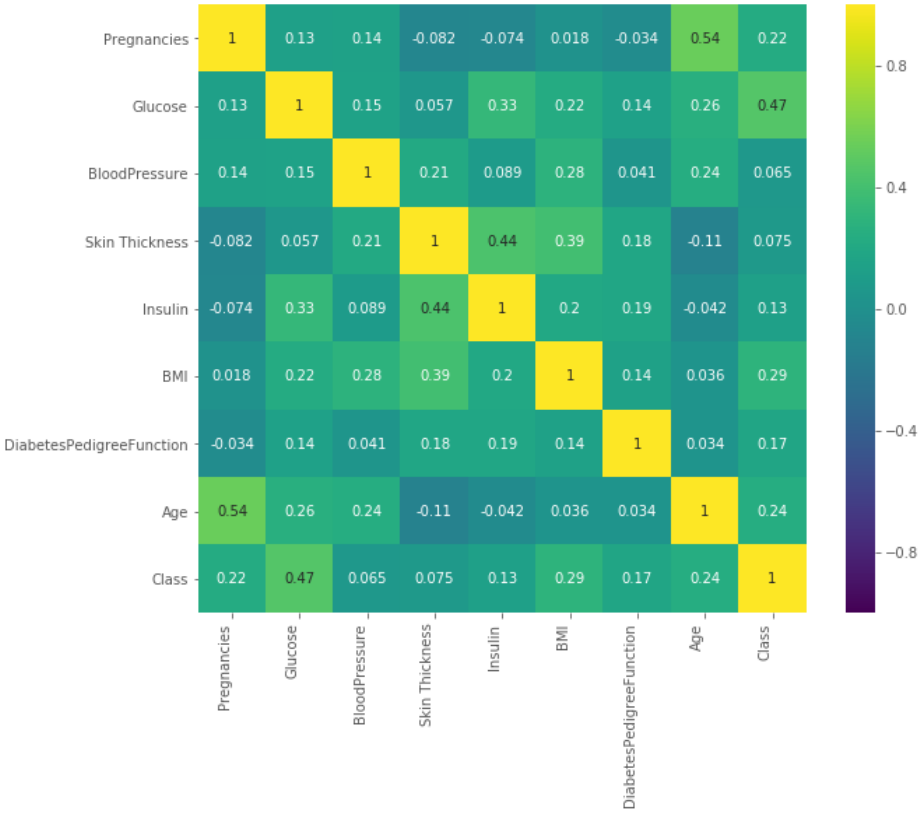




From the distributions and box plots above we can make a few comments on the underlying structure of each parameter. It is observable that higher pregnancy counts are more likely to be diagnosed with the disease however this can only be confirmed when the ration of men to women is known for both cases. It can be easily seen that higher glucose levels are seen more for positive cases as should be expected as can seen with the case of Age, where older people are more susceptible to getting the positive medical condition. Nothing for sure can be said about the blood pressure given the inaccuracies in the data and the fact that the distributions show that blood pressure is only slightly higher in positive cases. Similarly, BMI for positive cases peaks at a value a little higher than that of negative cases. All in all, we can see positive cases take higher values for each feature as compared to negative values and there is are visible bimodal graphs which peak at 0 which confirms our hypothesis that missing values were filled with zeros.

CORRELATION MATRIX

Next we will look at the correlation matrix which can be seen in the figure below.



From the figure above we can see that the following are highly correlated

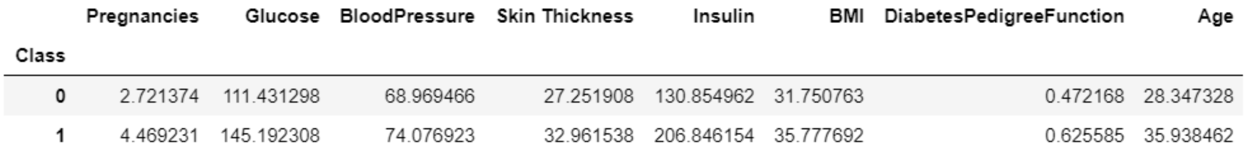
1. Age – Pregnancies
2. Diabetes – Glucose
3. Skin thickness – Insulin
4. Skin thickness – BMI
5. Glucose - Insulin

However, there are two peculiarities which can be seen is that even though Glucose and Diabetes are correlated, and Glucose and Insulin are correlated, Diabetes and Insulin aren't correlated with a correlation value of only 0.13

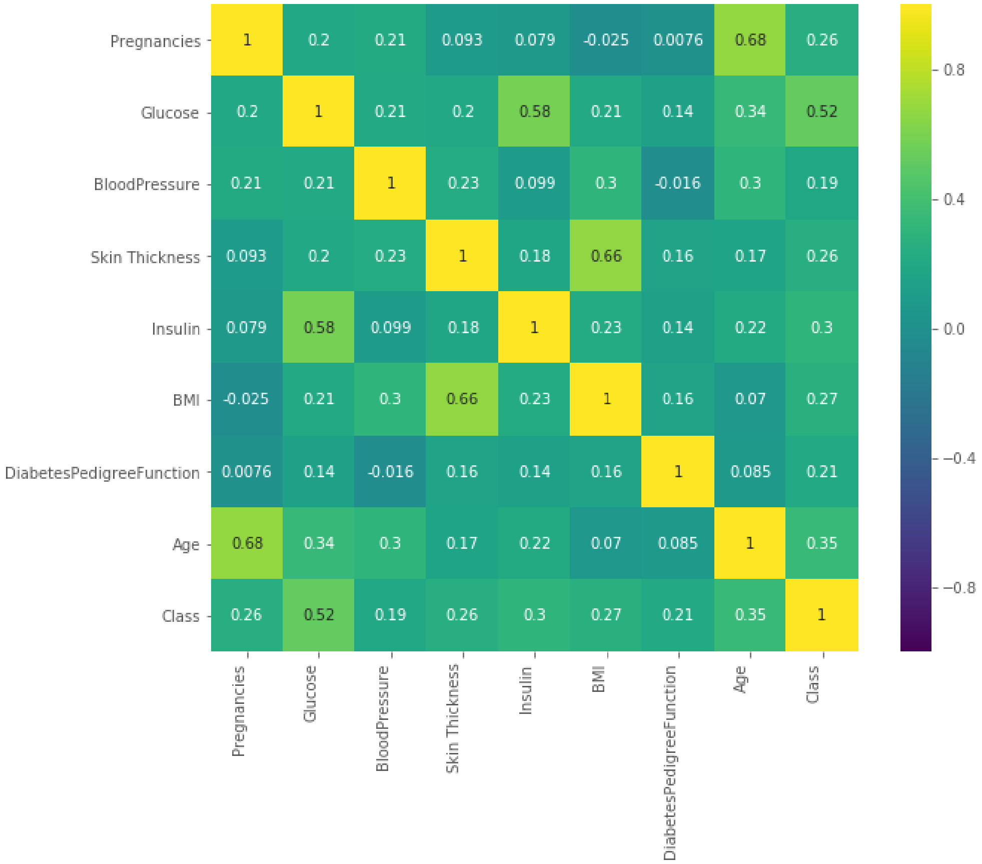
Similarly, BMI and Diabetes are correlated, and BMI and Skin thickness are correlated but Diabetes and Skin thickness aren't correlated with a value of 0.075.

We will try to look at these correlations after removing the 0 values, We will remove all the 0 values for Glucose, Skin Thickness, Blood Pressure, BMI and Insulin.

After removing the 0s, the mean values start to make sense all except the Insulin column.



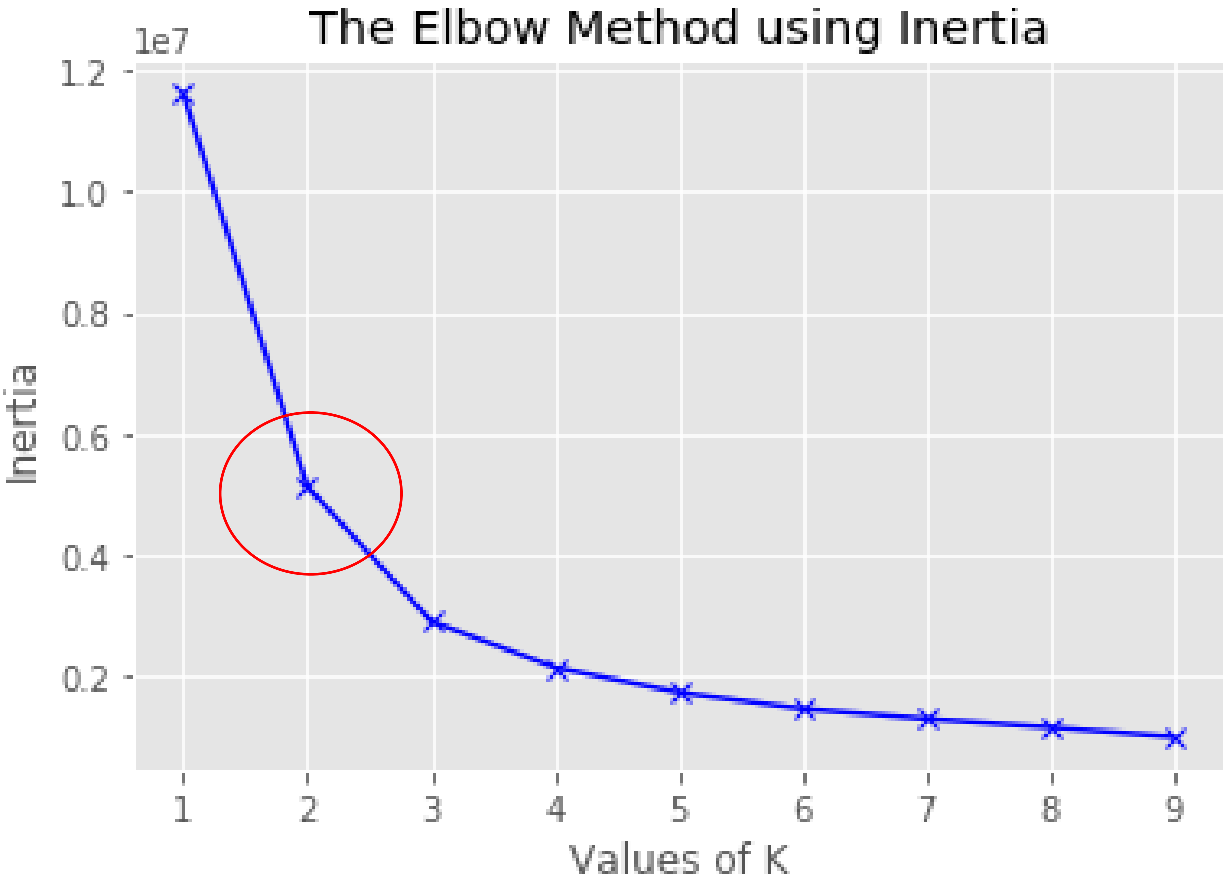
Class	Pregnancies	Glucose	BloodPressure	Skin Thickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	2.721374	111.431298	68.969466	27.251908	130.854962	31.750763	0.472168	28.347328
1	4.469231	145.192308	74.076923	32.961538	206.846154	35.777692	0.625585	35.938462



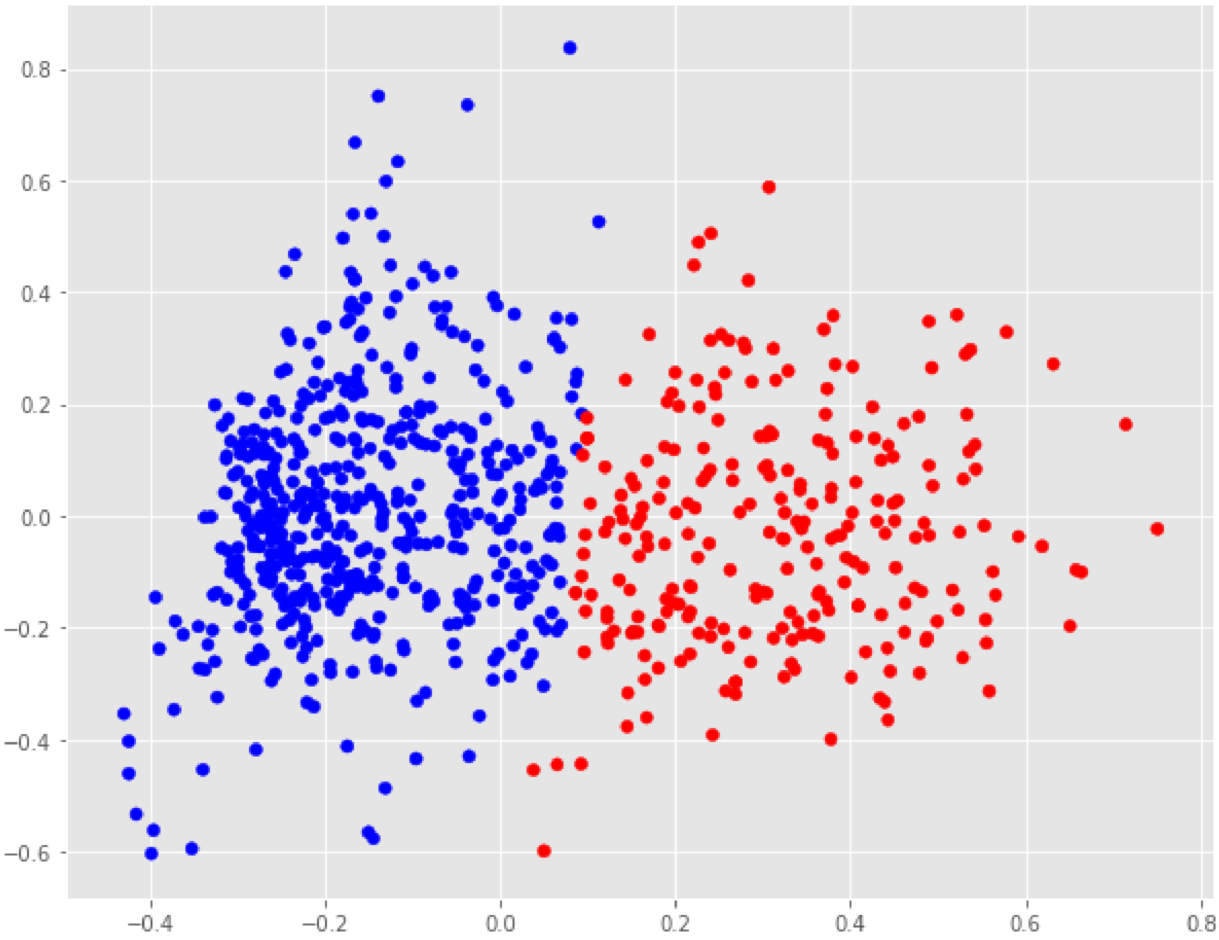
As for insulin, people who are mostly diabetic have low insulin levels which is why they can't convert the glucose to energy, however from the table above we can see that the mean value of insulin is higher in positive cases. One reason for that could be that the patients were tested after having been injected with insulin shots which has caused their values to shoot up. Other than that, another correlation that has become visible is the high correlation between Glucose and Age.

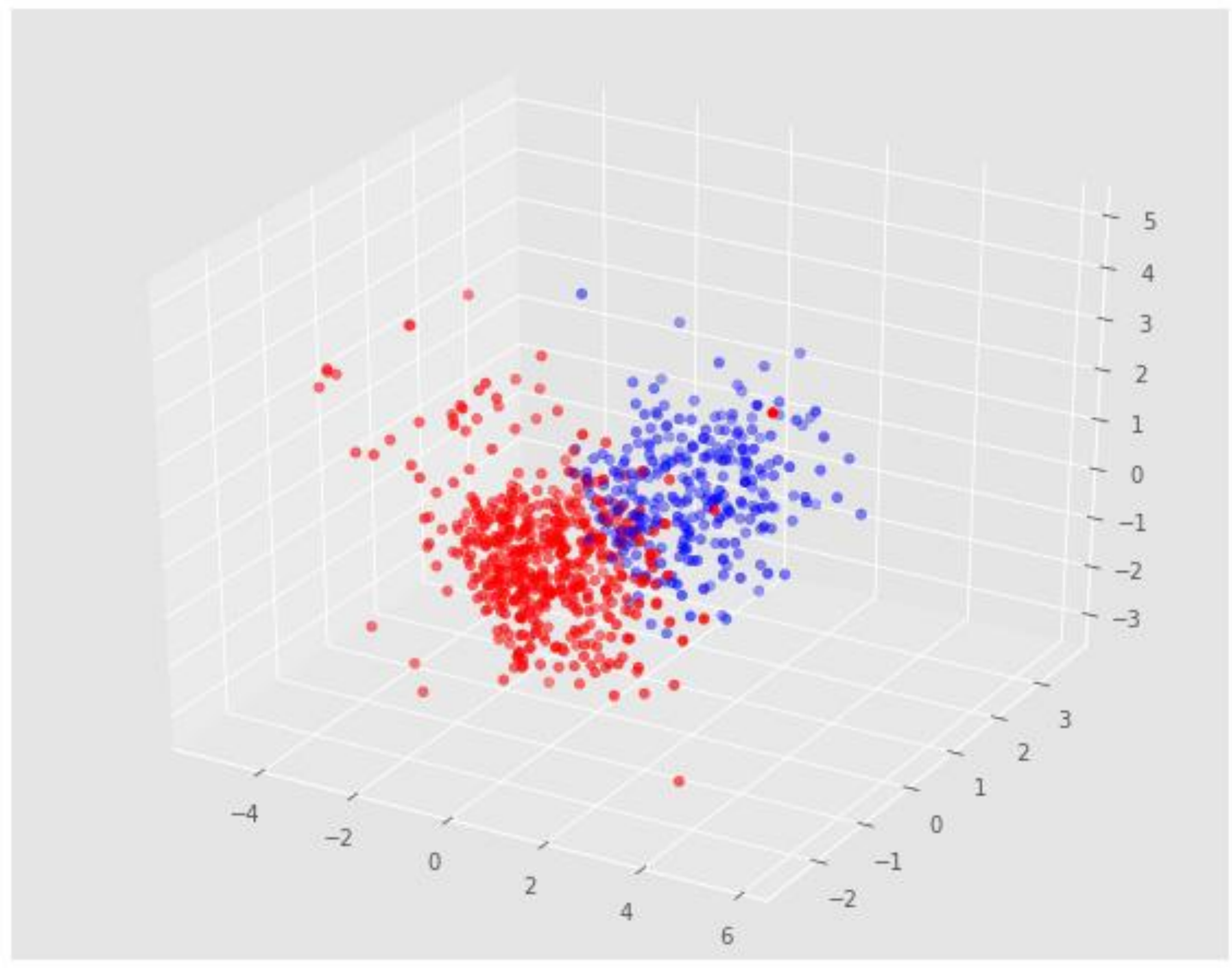
CLUSTERING ANALYSIS

This study then talks about the structure of the data using clustering analysis. This is to see whether visible clusters can be seen within our data and how much do they match with the diabetes status of an individual. For this analysis we will use **K Means Clustering** algorithm to look at the underlying structure. Before we actually go about using the K Means algorithm, we will first use a method called the Elbow method to see the optimal number of clusters that should be used. The **Elbow method** runs k-means clustering on the dataset for a range of values for **k** (say from 1-10) and then for each value of **k** computes an average score for all clusters. By default, the distortion score is computed, the sum of square distances from each point to its assigned center, the k which shows the highest drop, or where the shape of the elbow of the arm can be visualized shows the optimal number of clusters to be used.
The following graph showing the elbow method is shown below.



From the diagram we can see that the optimal number of clusters is equal to 2. Which is supportive of our idea as there are only 2 classes {positive and negative}. We then normalize our data so that each feature is given equal weightage according to their scales. We then run our K Means Algorithm for 2 clusters and observe the following clusters below.





From the 2D and 3D visualizations we can see that two very clear clusters being formed. These could be indicative of the diabetes status of an individual. To test our hypothesis, we look at the confusion matrix formed and see whether the labels given by the K Means algorithm matches with the status.

Before talking about the evaluation of the Algorithm, we first must define a few terms

Accuracy: Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations

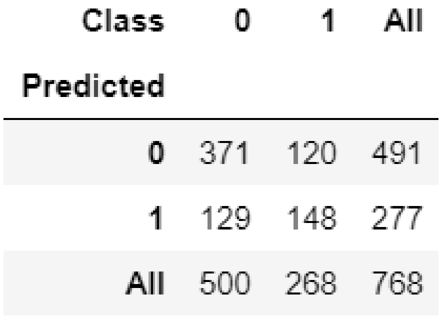
Recall: Recall is the ratio of correctly predicted positive observations to the all observations in actual class

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

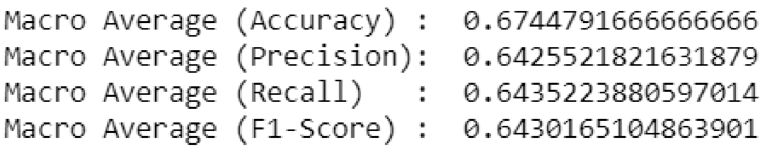
Macro Average: Macro-average method can be used when you want to know how the system performs overall across the sets of data.

Now that the terms have been discussed we can now look at the confusion matrix and the evaluation.

Confusion Matrix & Evaluation



Class	0	1	All
Predicted			
0	371	120	491
1	129	148	277
All	500	268	768



```
Macro Average (Accuracy) :  0.6744791666666666
Macro Average (Precision):  0.6425521821631879
Macro Average (Recall)   :  0.6435223880597014
Macro Average (F1-Score) :  0.6430165104863901
```

From the results above we can say that the clusters show somewhat matching to whether the status of the individual is diabetic or not. A random assignment for balanced classes would give a value around 0.5, however, in our case the classes are imbalanced, and the Macro Average Accuracy still is 0.17 higher with a value of 0.67. Thus, showing that these clusters are indicative of the whether the case is positive or negative.

PREDICTIVE MODELS

In this part of the study we will various machine learning algorithms to predict from our medical parameters whether an individual given his condition is diabetic or not. For this part we will use the following algorithms:

1. K – Nearest Neighbor Algorithm
2. Decision Trees
3. Logistic Regression
4. Support Vector Machines
5. Multi-Layer Perceptron
6. Random Forrest

We will not be using the Naïve Bayes Classifier as the Naive Bayes classifier assumes that the presence of a feature in a class is unrelated to any other feature. Even if these features depend on each other or upon the existence of the other features, however, that is not the case as all of these features are in some way dependent to each other as could be seen from the correlation matrix above.

Before we move to our first classification model, we first split our data into test and training data. Where the **Test Train Split is 20 – 80%**. The splitting is of a stratified nature which means that they instance in the training and test set are proportional to the class proportion.

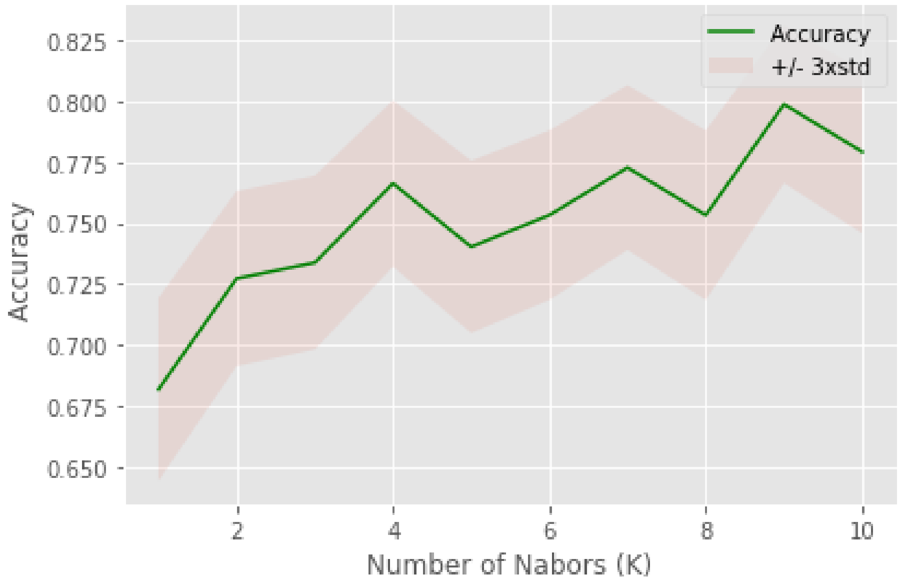
Train Data: 614

Test Data: 154

K-Nearest Neighbor

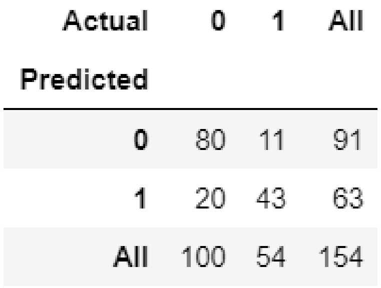
In k-NN, we began by normalizing the data between 0 and 1. k-NN uses the Euclidean distance, as its means of comparing examples to calculate the distance between two points. In order for all of the features to be of equal importance when calculating the distance, the features must have the same range of values. This is only achievable through normalization. Normalization is preferred over standardization in this case for k-NN as it is a distance-based classifier. We have already seen how well K Means (Unsupervised Distance Base) performed, now let's see how the supervised algorithm performs.

Before implementing the algorithm, we first have to see what the optimal value of k is. K in K means was the optimal number of clusters, here k is the k closest training examples in the feature space. We select the k training cases that have the smallest distance and look at their classification. The graph below shows which k-closest neighbors to look at during the classification task.

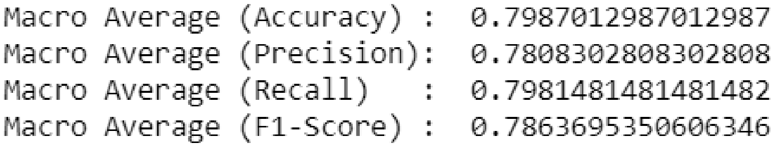


Here we can see that we will look at the 9 closest neighbors and according to the mode class will select the class for the testing sample. The green line in the diagram above shows the accuracy for each k used and the pink shade shows 3 times the standard deviation from that accuracy.

The confusion matrix and evaluation for the KNN model is shown below.



Actual	0	1	All
Predicted			
0	80	11	91
1	20	43	63
All	100	54	154



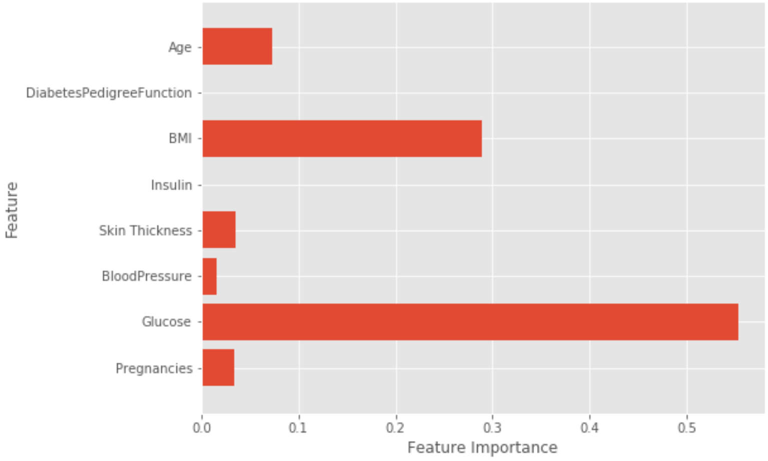
```
Macro Average (Accuracy) :  0.7987012987012987
Macro Average (Precision):  0.7808302808302808
Macro Average (Recall)   :  0.7981481481481482
Macro Average (F1-Score) :  0.7863695350606346
```

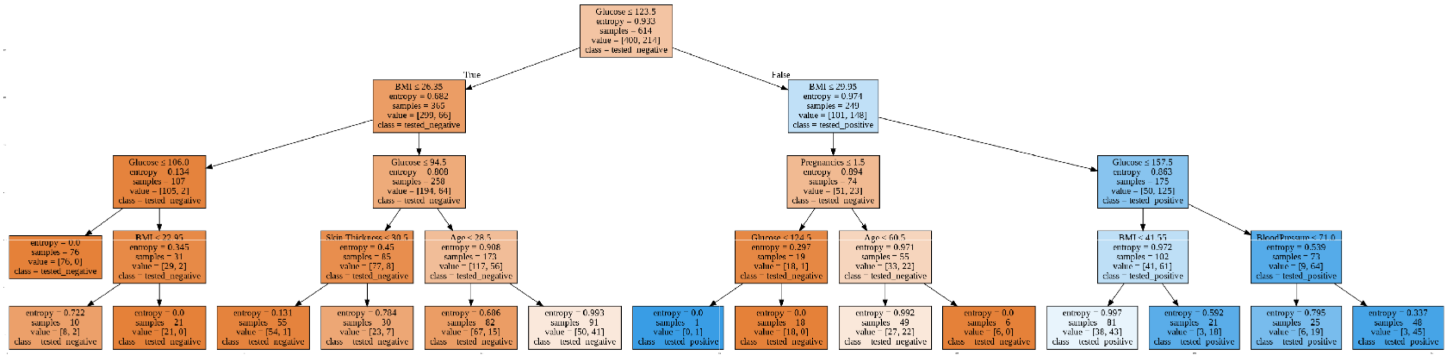
We can see that the supervised distance-based algorithm performs much better than what K Means did with a Macro Average Accuracy of 0.79

Decision Tree

The next model we will be using is the Decision Tree with a depth of 4 branches. For this classifier neither normalization nor standardization is required. The decision tree uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

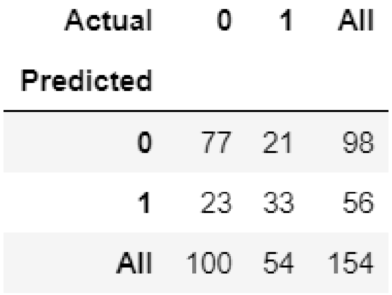
From the Decision Tree we are able to visualize what features it gives importance to and what conditional statements are produced based each features information gain.



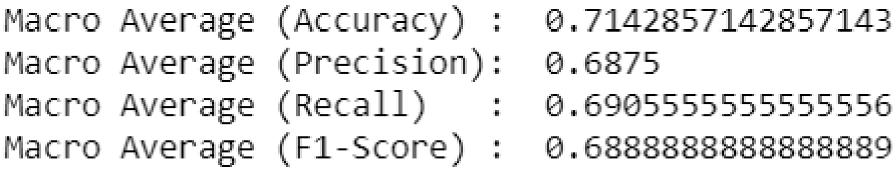

Glucose ≤ 123.5 entropy = 0.933 samples = 614 value = [400, 214] class = tested_negative
True
False
BMI ≤ 26.35 entropy = 0.682 samples = 365 value = [299, 66] class = tested_negative
BMI ≤ 29.95 entropy = 0.974 samples = 249 value = [101, 148] class = tested_positive
Glucose ≤ 106.0 entropy = 0.134 samples = 107 value = [105, 2] class = tested_negative
Glucose ≤ 94.5 entropy = 0.800 samples = 258 value = [194, 64] class = tested_negative
Pregnancies ≤ 1.5 entropy = 0.894 samples = 74 value = [51, 23] class = tested_negative
Glucose ≤ 157.5 entropy = 0.863 samples = 175 value = [50, 125] class = tested_positive
entropy = 0.0 samples = 76 value = [76, 0] class = tested_negative
BMI ≤ 22.95 entropy = 0.345 samples = 31 value = [29, 2] class = tested_negative
Skin Thickness ≤ 30.5 entropy = 0.45 samples = 85 value = [77, 8] class = tested_negative
Age ≤ 28.5 entropy = 0.908 samples = 173 value = [117, 56] class = tested_negative
Glucose ≤ 124.5 entropy = 0.297 samples = 19 value = [18, 1] class = tested_negative
Age ≤ 60.5 entropy = 0.971 samples = 55 value = [33, 22] class = tested_negative
BMI ≤ 41.55 entropy = 0.972 samples = 102 value = [41, 61] class = tested_positive
BloodPressure ≤ 71.0 entropy = 0.539 samples = 73 value = [9, 64] class = tested_positive
entropy = 0.722 samples = 10 value = [8, 2] class = tested_negative
entropy = 0.0 samples = 21 value = [21, 0] class = tested_negative
entropy = 0.131 samples = 55 value = [54, 1] class = tested_negative
entropy = 0.784 samples = 30 value = [23, 7] class = tested_negative
entropy = 0.686 samples = 82 value = [67, 15] class = tested_negative
entropy = 0.993 samples = 91 value = [50, 41] class = tested_negative
entropy = 0.0 samples = 1 value = [0, 1] class = tested_positive
entropy = 0.0 samples = 18 value = [18, 0] class = tested_negative
entropy = 0.992 samples = 49 value = [27, 22] class = tested_negative
entropy = 0.0 samples = 6 value = [6, 0] class = tested_negative
entropy = 0.997 samples = 81 value = [38, 43] class = tested_positive
entropy = 0.592 samples = 21 value = [3, 18] class = tested_positive
entropy = 0.795 samples = 25 value = [6, 19] class = tested_positive
entropy = 0.337 samples = 48 value = [3, 45] class = tested_positive

From the diagrams above we can see that the Decision tree gives most importance to the glucose levels and then BMI and age. We can also see from the tree decisions that all Glucose values less than 123.5 are classified as negative. And that the right side of the trees shows different cases of positive and negative instances.

The confusion matrix and evaluation for the Decision Tree model is shown below.



Actual	0	1	All
Predicted			
0	77	21	98
1	23	33	56
All	100	54	154



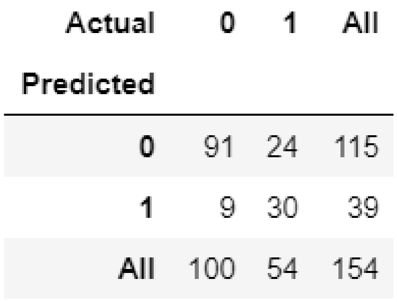
```
Macro Average (Accuracy) :  0.7142857142857143
Macro Average (Precision):  0.6875
Macro Average (Recall)   :  0.6905555555555556
Macro Average (F1-Score) :  0.6888888888888889
```

We can see that the Decision tree does just as good as the K Means Clustering but isn’t as good as the K Nearest Neighbor model.

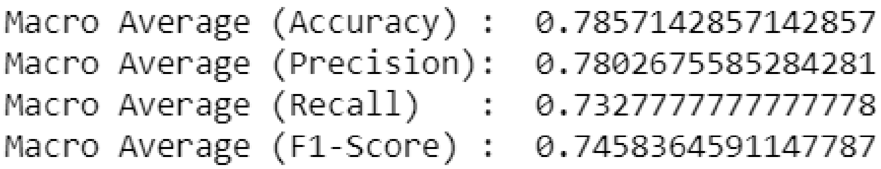
Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable to output the probability of a certain class.
In regression analysis, logistic regression is estimating the parameters of a logistic model (a form of binary regression).

The confusion matrix and evaluation for the Logistic Regression model is shown below.



Actual	0	1	All
Predicted			
0	91	24	115
1	9	30	39
All	100	54	154



```
Macro Average (Accuracy) :  0.7857142857142857
Macro Average (Precision):  0.7802675585284281
Macro Average (Recall)   :  0.7327777777777778
Macro Average (F1-Score) :  0.7458364591147787
```

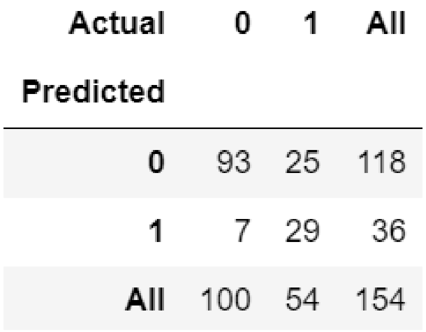
From the results above we can see that these results are much better than the Decision Tree model but the KNN model is still slightly better than this.

Support Vector Machines

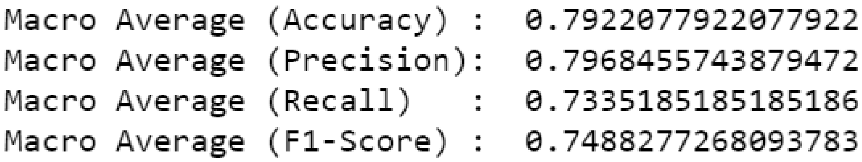
SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes. Support Vector Machine uses different kernelling methods to create an optimal hyperplane to separate classes such as

I. Radial Basis Function Kernel
II. Linear Kernel
III. Sigmoid Kernel

In our model we used the RBF Kernel as it yielded the best results. The confusion matrix and evaluation for the Support Vector Machines are shown below.



Actual	0	1	All
Predicted			
0	93	25	118
1	7	29	36
All	100	54	154

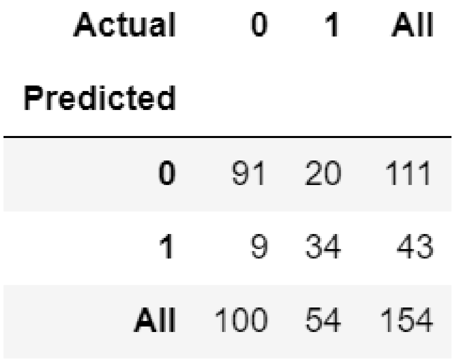


```
Macro Average (Accuracy) :  0.7922077922077922
Macro Average (Precision):  0.7968455743879472
Macro Average (Recall)   :  0.7335185185185186
Macro Average (F1-Score) :  0.7488277268093783
```

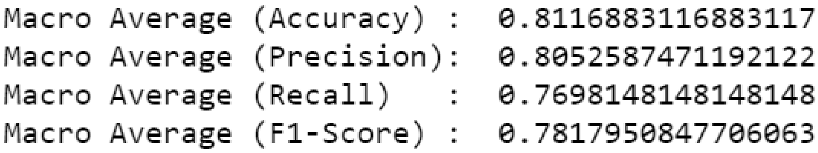
From the results above we can see that these results are much better than the Decision Tree model, slightly better than Logistic Regression and has results very similar to KNN.

Multi-Layer Perceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable. In this model we used 1000 iteration till it converged with a learning rate of 0.01. The confusion matrix and evaluation for the MLP are shown below.



Actual	0	1	All
Predicted			
0	91	20	111
1	9	34	43
All	100	54	154

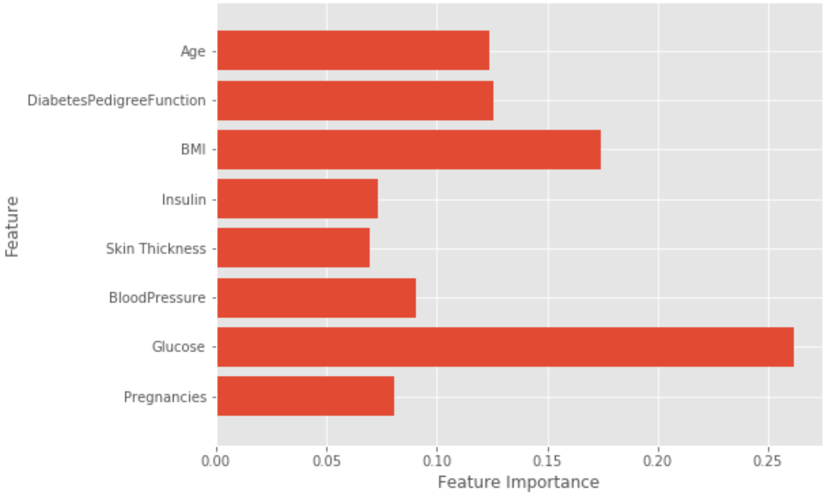


```
Macro Average (Accuracy) :  0.8116883116883117
Macro Average (Precision):  0.8052587471192122
Macro Average (Recall)   :  0.7698148148148148
Macro Average (F1-Score) :  0.7817950847706063
```

Random Forrest

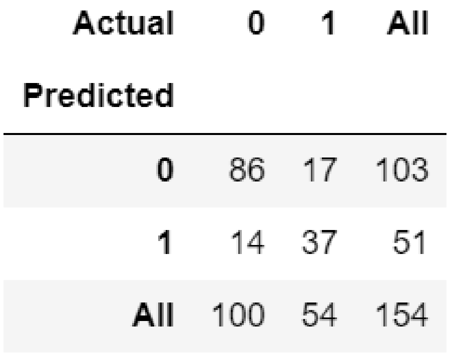
Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and then aggregates the votes from different decision trees to decide the final class of the test object.

Just like the decision trees we can view the features importance of all the medical parameters which can be seen below.

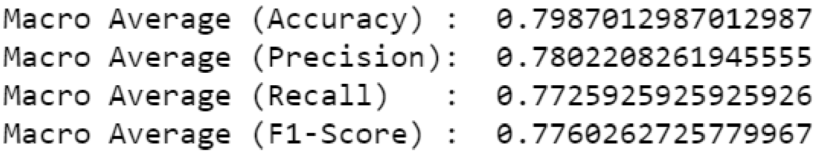


Unlike Decision trees we can see that the Random Forrest classifier has given a lot more importance to some of the other features.

The confusion matrix and evaluation for the Random Forrest are shown below.



Actual	0	1	All
Predicted			
0	86	17	103
1	14	37	51
All	100	54	154



```
Macro Average (Accuracy) :  0.7987012987012987
Macro Average (Precision):  0.780220826194555
Macro Average (Recall)   :  0.7725925925925926
Macro Average (F1-Score) :  0.776026272577967
```

Below is a table of algorithms which predicted best on the test data

Algorithm	Accuracy	F1-Score	Log Loss
MLP	0.811	0.782	6.5043
KNN	0.798	0.786	6.952
Support Vector Machine	0.792	0.748	7.176
Random Forrest	0.786	0.765	7.401
Logistic Regression	0.785	0.745	7.401
Decision Tree	0.714	0.688	9.868
K-Means Clustering	0.667	0.629	11.468

CONCLUSION

After running various models to predict on a test data set whether a certain individual is diabetic or not. It can be seen that the best results were given by the MLP model which uses a deep learning approach to solve the problem. The KNN (Distance approach model) and SVM also have pretty good results with high accuracies and f1-scores. Future work can include using **Synthetic Minority Oversampling Technique (SMOTE)** for class imbalance classification. Other than that, a new data set with standardized features if used can generate much better results, for example, glucose and insulin levels are checked after a person has fasted for 12 hours and that no insulin shots have been previously used. Ratio of males and females to give an idea what affect does the number of pregnancies have. There were a lot of limitations due to the data at hand however if this same approach is mimicked and some extra features such as nutritional behavior of an individual is provided then a more meaningful analysis can be obtained and maybe a recommendation system as to those people who are prediabetic what nutrition should they take and what should they avoid for a healthier lifestyle.