

# **Diabetes Prediction Analysis**

## **FINAL PROJECT REPORT**

**Student Name: Varun Kumar Kumaravel**

[kumaravel.v@northeastern.edu](mailto:kumaravel.v@northeastern.edu)

**Percentage of Effort Contributed: 100%**

**Signature of Student 1: Varun Kumar Kumaravel**

**Submission Date: 27<sup>th</sup> April 2023**

# CONTENTS

- 1. Problem Setting:.....
- 2. Problem Definition: .....
- 3. Data Sources: .....
- 4. Data Description: .....
- 5. Project Motive:.....
- 6. Data Mining Tasks:.....
- 7. Data Exploration .....
- 8. Interesting Visualizations.....
- 9. Data Processing.....
- 10. Model Exploration and Selection: .....
- 11. Model Performance Evaluation and Interpretation.....
- 12. Project Impact: .....

## **IE 7275: Data Mining in Engineering**

### **1. Problem Setting:**

The high prevalence of obesity in children in our community has become a major public health concern. Despite efforts to promote healthy eating and physical activity, many children continue to consume a diet high in calories and low in nutrients, and engage in minimal physical activity. This has led to an increase in chronic diseases such as diabetes and hypertension. There is a need to develop effective interventions to promote healthy lifestyle behaviors among children and prevent the onset of chronic diseases. The problem setting for this project is to use Machine learning models to predict whether an individual is at risk of developing diabetes or has already developed diabetes based on certain features or risk factors. This analysis is important because early detection and intervention can prevent or delay the onset of diabetes and its complications. By analyzing the dataset, we can gain insights into the factors that are most strongly associated with the development of diabetes, as well as the relative importance of different features in predicting the diabetes risk. The project entails leveraging a dataset of patient information and performing a comprehensive analysis of the data, including exploratory data analysis, feature selection, and experimentation with diverse model types such as tree-based and neural network models.

### **2. Problem Definition:**

The diabetes prediction dataset typically includes features such as age, gender, BMI, blood pressure and glucose levels, among others. The Machine Learning model is provided with outcome variable and respective predictors. The final output is a binary classification for the outcome variable. The primary goal of the project is to overcome the challenge of identifying the most pertinent features from a vast array of patient information, and subsequently build a robust predictive model that accurately assesses the risk of diabetes based on these selected features. With the developed model, it is possible to predict and classify a patient's likelihood of developing diabetes in the future, thereby providing crucial insights for proactive healthcare management.

### **3. Data Sources:**

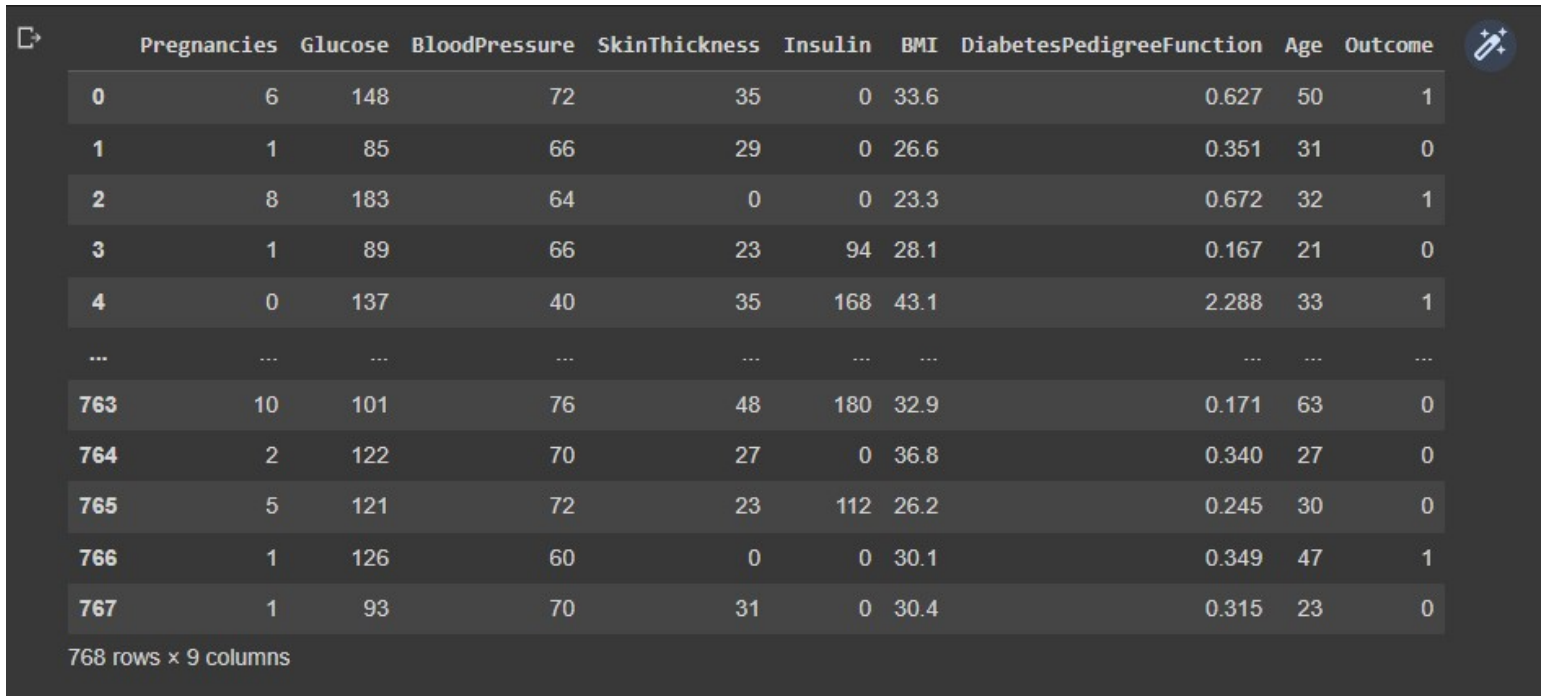
The dataset source for the Diabetes Prediction has been obtained from the website

<https://www.kaggle.com/code/ahmedelsaied3210/diabetics-prediction/input>

#### 4. Data Description:

The dataset contains a total of 768 records or instances, with each instance representing a single patient. There are no missing values in this version of the dataset. The dataset was originally collected as a part of the Diabetes Prevention Program (DPP) study conducted by the National Institute of Diabetes and Kidney Diseases (NIDDK).

The input parameters are pregnancies, glucose, blood pressure, skin thickness, Insulin, BMI, Diabetes Pedigree function, age, and outcome.



	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
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows x 9 columns

Attribute	Description	Dtype
Pregnancies	This column represents the number of times the individual has been pregnant	Int64
Glucose	This column represents the plasma glucose concentration of the individual, measured in milligrams per deciliter (mg/dL)	Int64
BloodPressure	This column represents the diastolic blood pressure of the individual measured in millimeters of mercury (mm Hg)	Int64
SkinThickness	This column represents the thickness of the triceps skin fold of the individual, measured in millimeters (mm)	Int64
Insulin	This column represents the 2-hour serum insulin levels of the individual, measured in microunits per milliliter (mu U/ml)	Int64
BMI	This column represents the body mass index of the individual, calculated as weight in kilograms divided by height in meters squared ( $\text{kg/m}^2$ )	Float
DiabetesPedigreeFunction	This column represents a function that scores the likelihood of an individual developing diabetes based on their family history	Float
Age	This column represents the age of the individual in years	Int64
Outcome	This column represents whether or not the individual has been diagnosed with diabetes	Int64

## 5. Project Motive:

According to the Centers for Disease Control and Prevention (CDC), as of 2021, approximately 34.2 million people in the United States have diabetes, which represents 10.5% of the total US population. This includes both diagnosed and undiagnosed cases.

The project aims to use the diabetes prediction dataset to select the most relevant features and develop a model that can accurately predict the risk of diabetes based on these features. The model can be used by healthcare professionals to identify individuals who are at high risk of developing diabetes and provide them with appropriate interventions and treatments. Our project involves evaluating 7 diverse Machine Learning models, namely Naïve Bayes, K Nearest Neighbors, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and K Means Clustering. Our aim is to identify the model that delivers consistent and reliable outcomes. After obtaining the prediction outcomes from these models, we will conduct a comprehensive performance analysis to determine

the most effective model. Additionally, we will subject the top-performing model to a cross-validation process to ascertain its repeatability and evaluate its predictive accuracy. This analysis will enable us to identify the most accurate and reliable model.

## 6. Data Mining Tasks:

### Data Exploration:

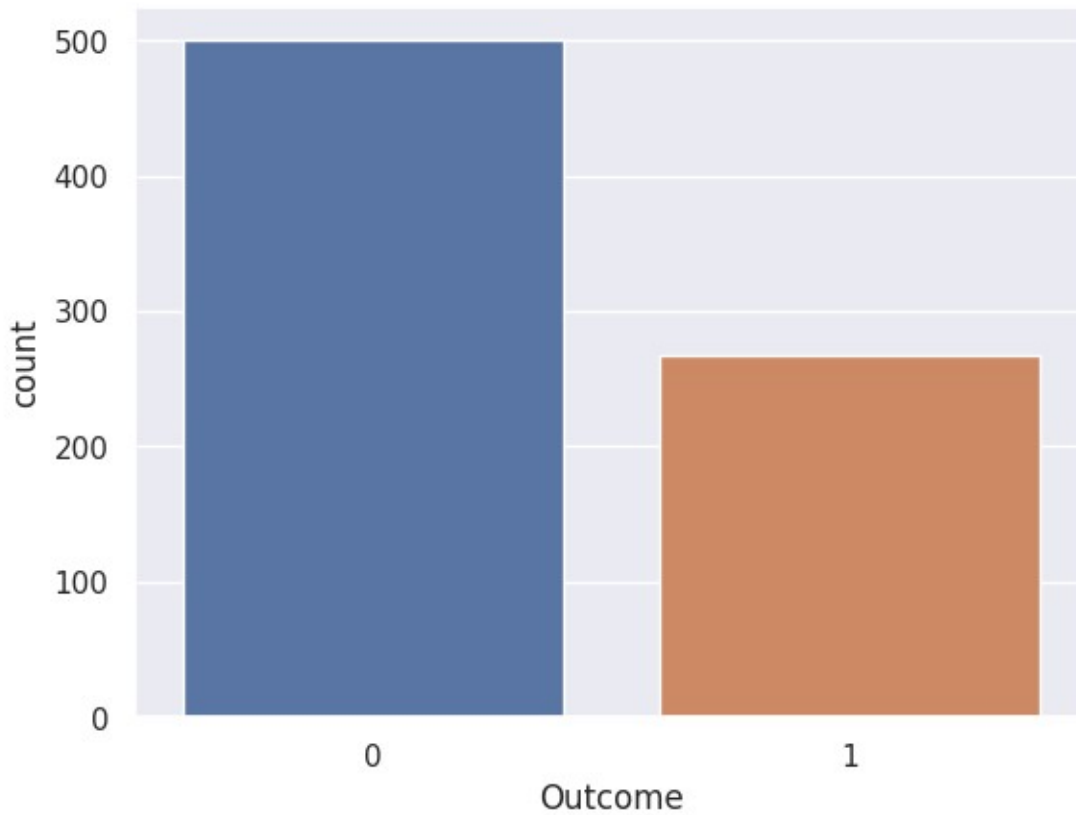
First of all, when I uploaded the dataset in Google Colab and started with the data cleaning process, The dataset was well-organized and had a high degree of data quality, making it easy to understand and work with. Below I will attach the count of null values:

```
Pregnancies      0
Glucose           0
BloodPressure     0
SkinThickness     0
Insulin           0
BMI               0
DiabetesPedigreeFunction  0
Age              0
Outcome           0
dtype: int64
```

*Figure 1*

There are no null values in the dataset. But some features like Blood Pressure, Insulin, Glucose, BMI have zero values which represent missing data. In the outcome column, 0 represents diabetes negative and 1 represents diabetes positive.

Below is the graphical representation of outcome count Plot:



*Figure 2: Graphical Representation of outcome Count Plot*

From the above graph, it is clear that there are more diabetes negative, but still there is considerable amount of diabetes positive, which means the count plot tells us that the dataset is imbalanced.

The below Histplot represents the distribution of the attributes in strokes dataset.

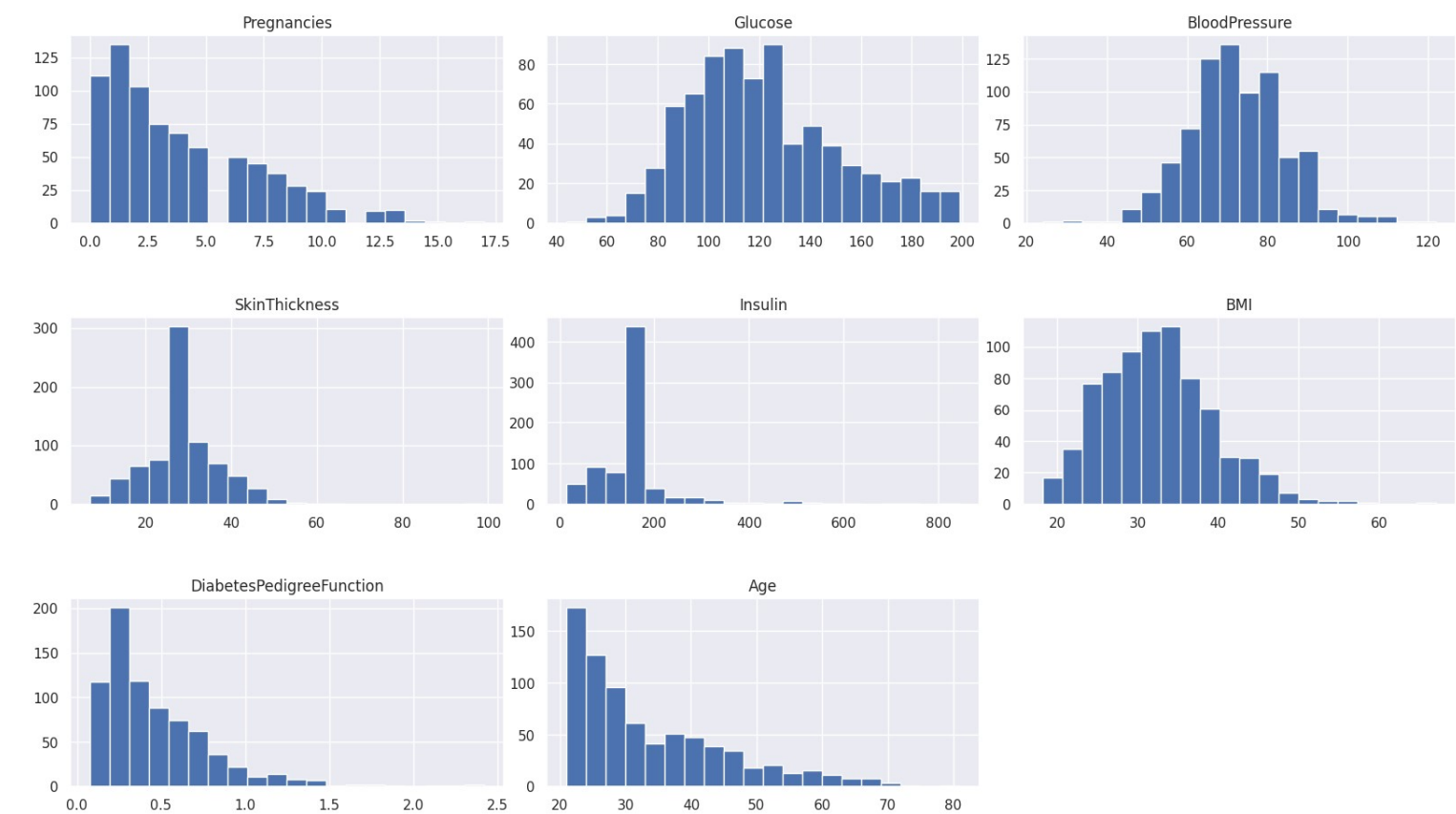
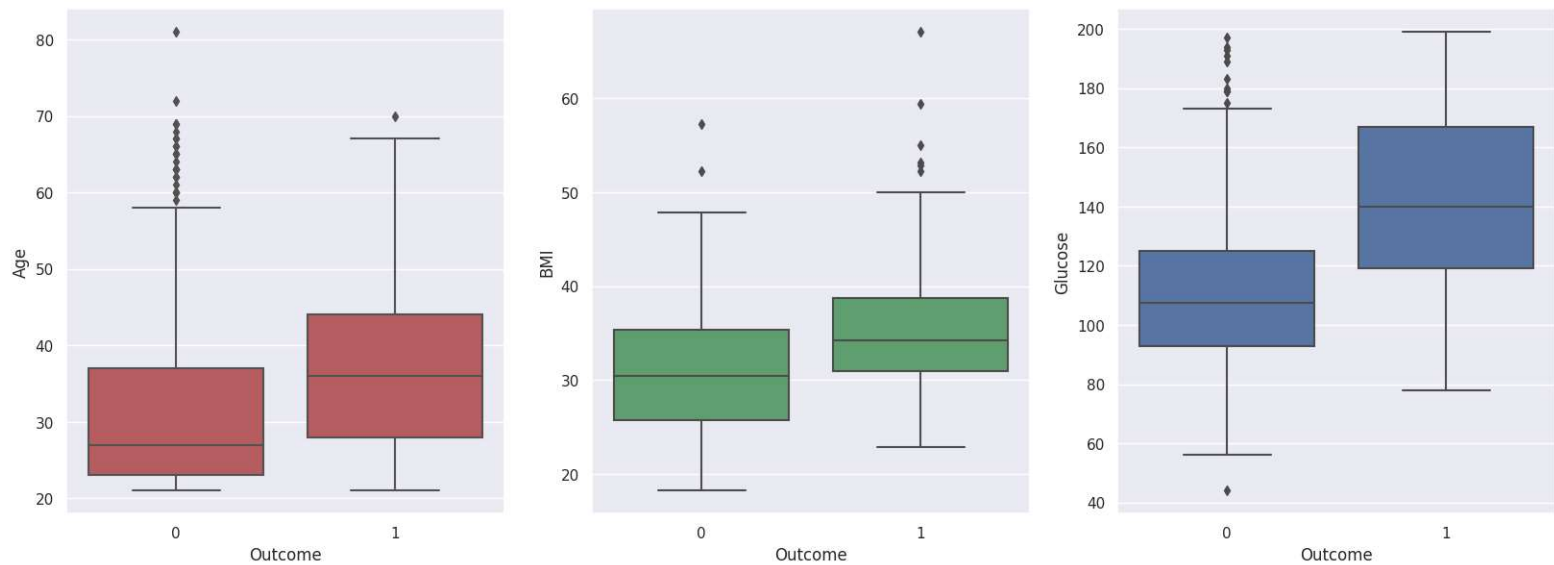


Fig: Histogram of each feature

Interesting Visualizations:





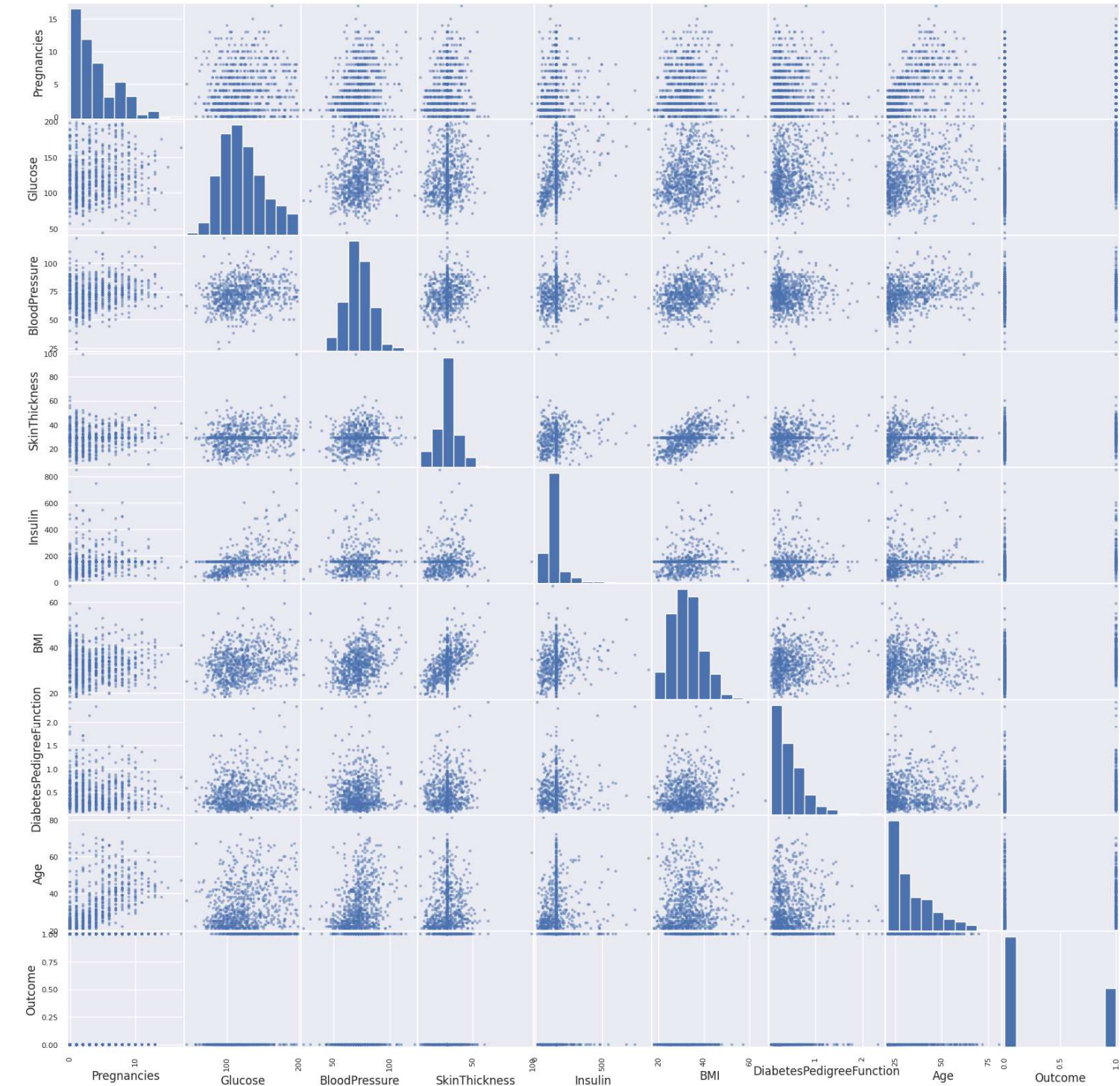


Fig: Scatter Matrix

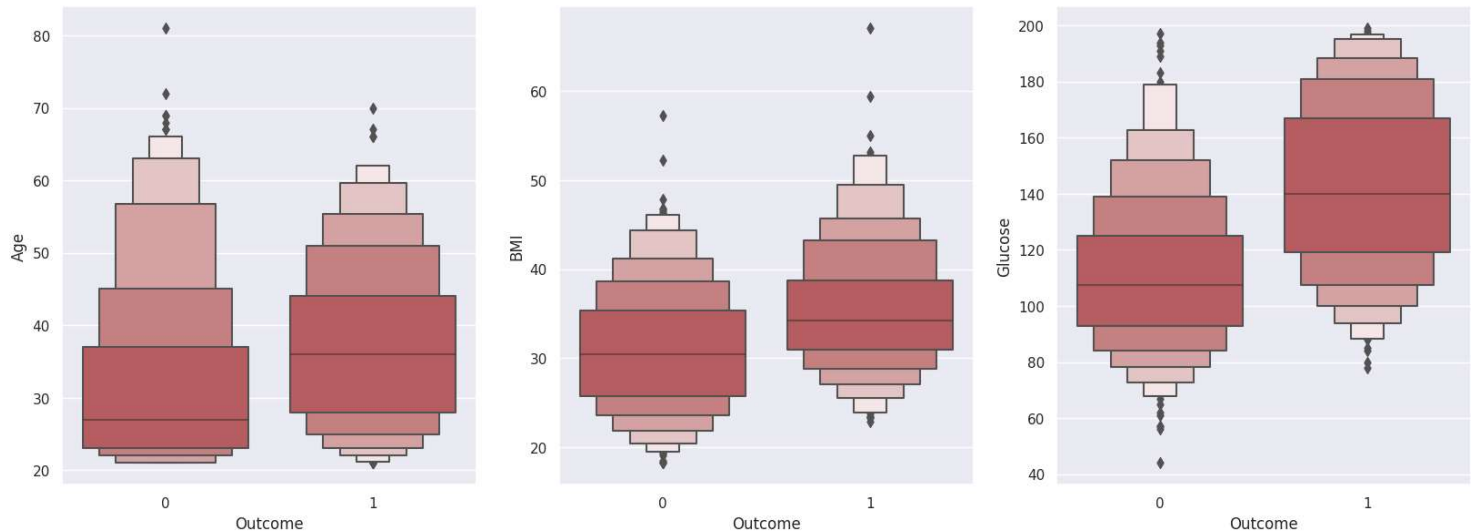


Fig: Pair Plot

Looking at the pairplot and scatter matrix, I can infer that:

- Age and BMI have a relatively strong positive linear relationship. This may indicate that as age increases, so does BMI, which is an important factor in the development of diabetes.
- Glucose and Outcome have a strong positive relationship. This is expected, as glucose levels are a key diagnostic indicator of diabetes.

- Blood pressure and skin thickness have a weak positive relationship. This suggests that these features may be less important in predicting diabetes compared to other features.
- There may be some outliers in the dataset for certain features, which could potentially affect model performance.



*Fig: Boxenplot*

Looking at the box plots and Boxen Plots for the diabetes prediction dataset, observations are provided below:

- Age is relatively normally distributed, with a few outliers above the upper whisker.
- BMI is slightly right-skewed, with some outliers above the upper whisker.
- Blood pressure is approximately normally distributed, with a few outliers above the upper whisker.
- Glucose is right-skewed, with several outliers above the upper whisker. This is expected, as high glucose levels are a key diagnostic indicator of diabetes.
- Skin thickness and insulin have many values at the minimum value of zero, indicating missing or incomplete data. This may need to be addressed before using these features in a model.



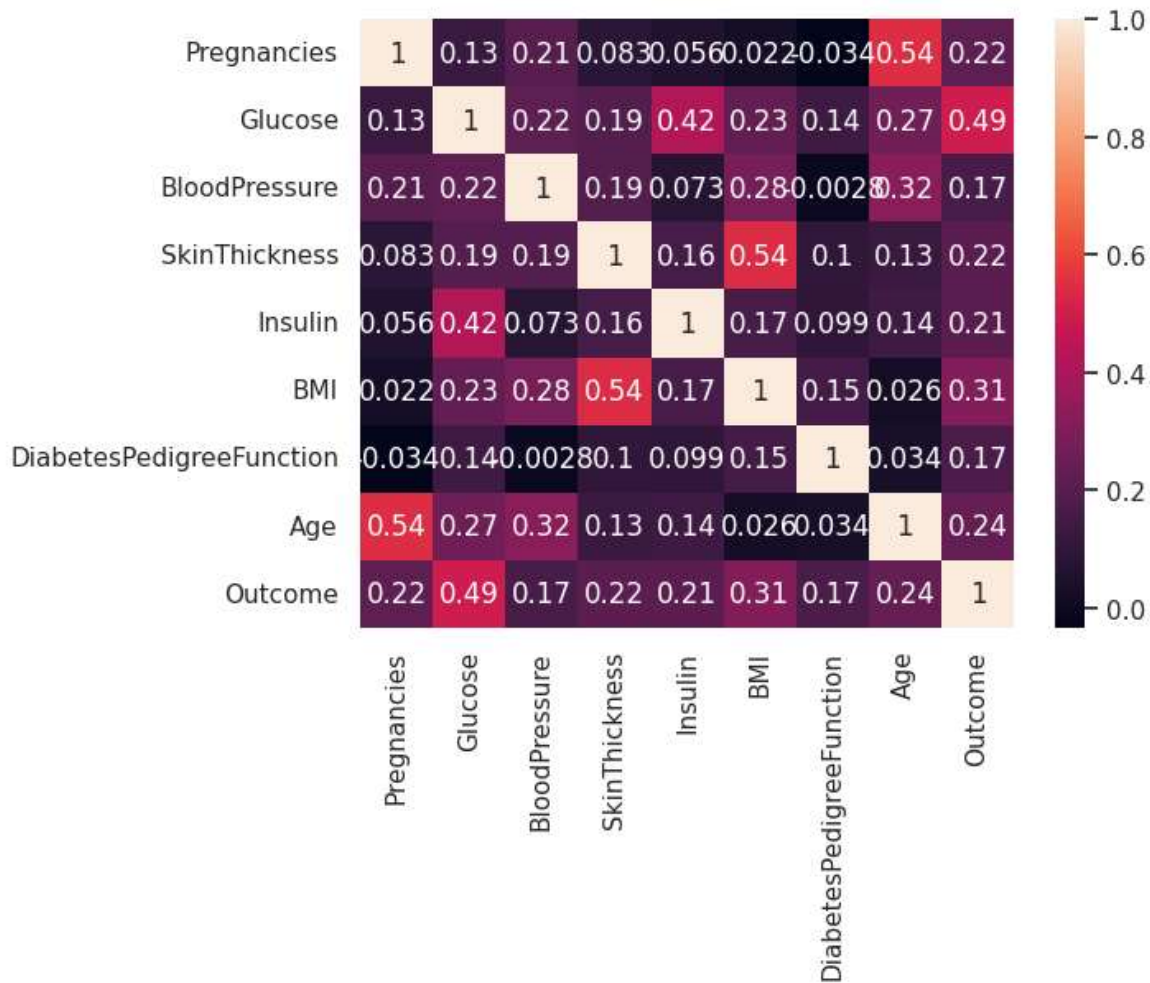


Fig: Correlation Heatmap

From the correlation heatmap, we can see that there is high correlation between Outcome and [Glucose, Age, Insulin, BMI]. We can select these features to accept input from the user and predict the outcome.

### DATA PROCESSING:

Now, assigning all the values 0 with NAN values. Once, it is replaced 0 with NAN, the number of NULL values has increased drastically. This can be seen below:

```

Pregnancies      0
Glucose           5
BloodPressure     35
SkinThickness     227
Insulin           374
BMI              11
DiabetesPedigreeFunction  0
Age              0
Outcome          0
dtype: int64

```

It is clear that there are more than 650 null values in the dataset after replacing '0' with NAN. The next step is replacing the NAN values with mean values. By using the `fillna()` function, along with `mean`, the NAN values are filled with the mean value of the dataset.

Now, the feature scaling is performed using `MinMaxScaler` to make the range of 0 as min and 1 as maximum. As we are splitting the dataset as X and Y, we are using the `'train_test_split()'` function from the Scikit-learn to split the dataset 'X' and its corresponding target variable 'Y' into training and test datasets. The `test_size` parameter is set to 0.20, which means that 20% of the data will be used for testing, and the remaining 80% will be used for training. The `'random_state'` parameter is set to 42 to ensure that the results are reproducible, and the `'stratify'` parameter is used to ensure that the split is representative of the distribution of the target variable 'Outcome'.

The number of samples in the training dataset will be 80% of the total number of samples, and the number of samples in the test dataset will be 20% of the total number of samples.

## MODEL EXPLORATION AND SELECTION:

In model exploration and selection, the goal is to identify a machine learning model that can accurately predict whether a patient has diabetes or not based on various input features (Glucose, Insulin, BMI, Age).

I have displayed the total number of samples in whole, training, and testing dataset below:

```

#displaying the total number of samples in the whole, training and testing dataset
print(f'Total # of sample in whole dataset: {len(X)}')
print(f'Total # of sample in train dataset: {len(X_train)}')
print(f'Total # of sample in test dataset: {len(X_test)}')

Total # of sample in whole dataset: 768
Total # of sample in train dataset: 614
Total # of sample in test dataset: 154

```

Then, I have defined a dictionary of Machine Learning models namely Naive Bayes classifier, Logistic Regression, Random Forest Classifier, Support Vector Machine Classifier, Decision Tree Classifier, K Nearest

Neighbors classifier, Gradient Boosting Classifier, Stochastic Gradient Classifier, and Neural Networks. Each model has its own advantages and disadvantages, and choosing the right model depends on the specific requirements of the project. Here are some of the models under consideration:

- **Random Forest Classifier**

**Advantages:**

- It can handle missing values and can also work with unbalanced datasets
- Has a low risk of overfitting.
- Easy to use, and the results are interpretable.

**Disadvantages:**

- It can be computationally expensive, especially for large datasets with a large number of trees
- Not suitable for problems with a small number of training examples

- **Logistic Regression**

**Advantages:**

- It is easy to implement and simple that can quickly provide predictions
- Computationally efficient
- It can provide probability estimates, which can be useful in certain applications like fraud detection

**Disadvantages:**

- Not suitable for problems with a large number of classes
- It may suffer from overfitting when the number of features is much larger than the number of observations
- It is sensitive to outliers and missing values in the data.

- **Support Vector Machines (SVMs)**

**Advantages:**

- It is effective in high-dimensional spaces
- It works well with both linearly separable and non-linearly separable datasets.
- It is robust to overfitting.

**Disadvantages:**

Computationally expensive and requires a lot of memory for training large datasets

- Results of SVM can be difficult to explain and interpret

- **K-Nearest Neighbors (KNN)**

**Advantages:**

- It is simple and easy to understand and implement
- Used for both classification and regression problems
- It can work well with both linear and non-linear decision boundaries

**Disadvantages:**

- It requires a lot of memory to store the entire dataset for fast querying during prediction
- It is sensitive to K where K is choice of number of neighbors and the distance metric is used to calculate the similarity between samples

- **Gradient Boosting Classifier**

**Advantages:**

- It can handle different types of input features
- Work well with both linear and non-linear decision boundaries
- It can achieve high accuracy with little tuning

**Disadvantages:**

- It can be computationally expensive, especially for large datasets
- Difficult to interpret the results
- It may be sensitive to imbalanced data and may require additional techniques to handle class imbalance

**MODEL PERFORMANCE EVALUATION:**

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, classification_report

#Define the names of the models to be evaluated
modelName = ["GaussianNB", 'BernoulliNB', 'LogisticRegression', 'RandomForestClassifier', 'SupportVectorMachine', 'DecisionTreeClassifier', 'KNeighborsClassifier', 'GradientBoostingClassifier', 'Stochastic Gradient Descent', 'Neural Nets']

#Create empty lists to hold train and test scores
trainScores = []
testScores = []

#Create an empty dictionary to store the results for each model
results = {}

#loop through each model name and fit the model to the training data
for m in models:
    model = models[m]
    model.fit(X_train, y_train)
```

```

# Calculate and store the train and test scores
train_score = model.score(X_train, y_train)
trainScores.append(train_score*100)

test_score = model.score(X_test, y_test)
testScores.append(test_score*100)

# Generate predictions using the test data and calculate performance metrics
y_predictions = model.predict(X_test)
conf_matrix = confusion_matrix(y_predictions, y_test)

tn = conf_matrix[0,0]
fp = conf_matrix[0,1]
tp = conf_matrix[1,1]
fn = conf_matrix[1,0]
accuracy = accuracy_score(y_test, y_predictions)
precision = precision_score(y_test, y_predictions)
recall = recall_score(y_test, y_predictions)
f1score = f1_score(y_test, y_predictions)
specificity = tn / (tn + fp)

# Generate classification report and store results for this model in the results dictionary
report = classification_report(y_test, y_predictions, output_dict=True)

results[m] = {
    'Train Score': train_score,
    'Test Score': test_score,
    'Accuracy': accuracy,
    'Precision': precision,
    'Recall': recall,
    'F1 Score': f1score,
    'Specificity': specificity,
    'Classification Report': report
}

# Remove the current model from the list of models to be evaluated
modelName.remove(m)

# Print the results for this model
print("-----")
print(f'{m}')
print(f'Train score of trained model: {train_score*100}')
print(f'Test score of trained model: {test_score*100}')
print(f'Confusion Matrix: \n{conf_matrix}\n')
print(f'Accuracy : {accuracy}')

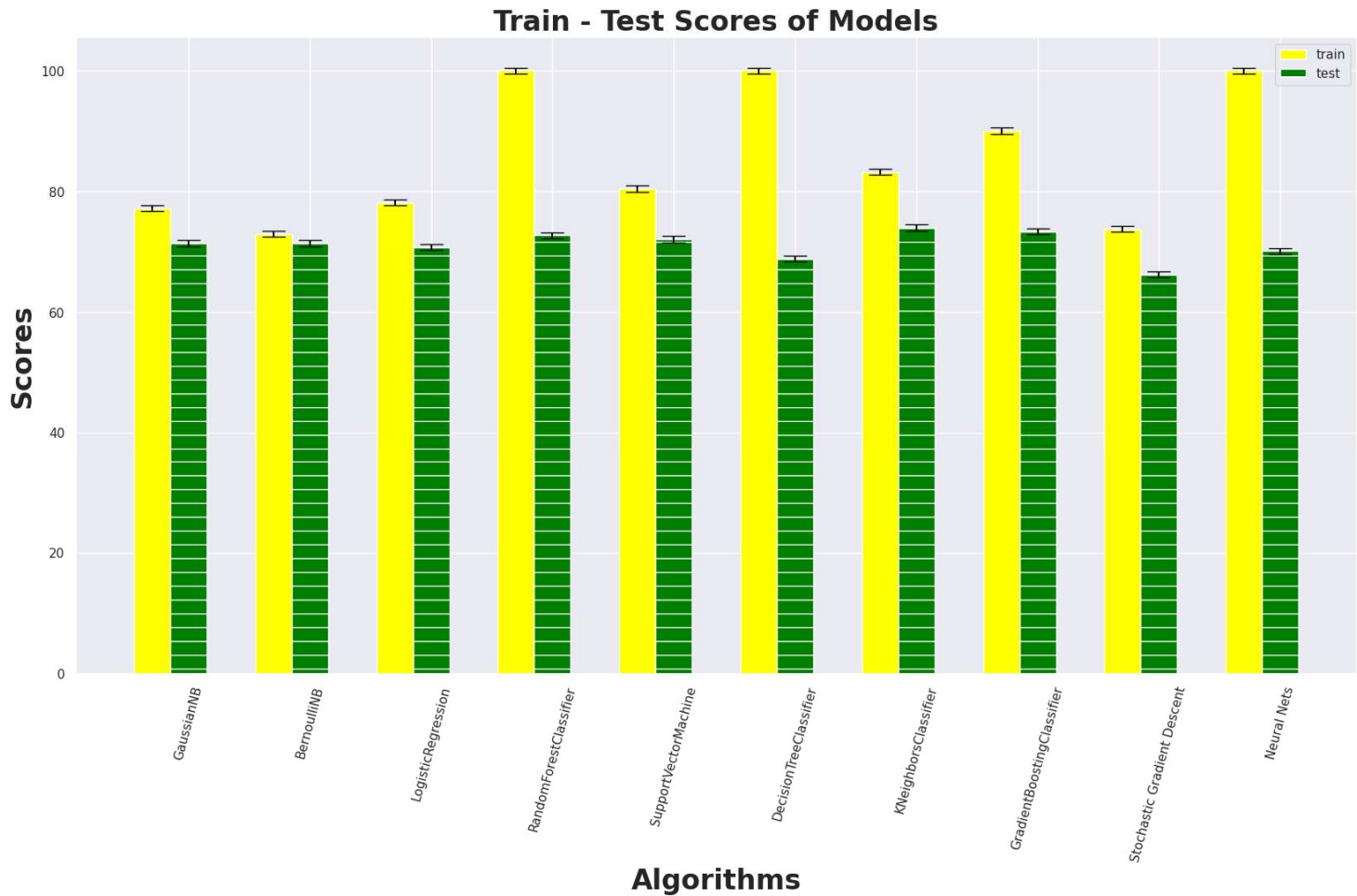
```



```

print(f'Precision: {precision}')
print(f'Recall    : {recall}')
print(f'F1 score  : {f1score}')
print(f'Specificity : {specificity}')
print(f'Classification Report: \n{classification_report(y_test, y_predictions)}\n')
print("")

```



Fig

**Accuracy Scores** of all the models are provided below:

Accuracy of GaussianNB: 71.42857142857143

Accuracy of BernoulliNB -----> 71.42857142857143

Accuracy of LogisticRegression -----> 70.77922077922078

Accuracy of RandomForestClassifier -----> 75.97402597402598

## IE7275: Data Mining in Engineering

Accuracy of SupportVectorMachine -----> 72.07792207792207

Accuracy of DecisionTreeClassifier -----> 68.181818181817

Accuracy of KNeighborsClassifier -----> 74.02597402597402

Accuracy of GradientBoostingClassifier -----> 73.37662337662337

Accuracy of Stochastic Gradient Descent -----> 66.23376623376623

Accuracy of Neural Nets -----> 70.12987012987013

From the above accuracy scores, it is clear that the accuracy score of Random Forest Classifier is the highest.

We evaluated all the models above according to their accuracies. Best algorithm is Random Forest with 75.324%. So, we will make k-Fold Cross Validation and Hyper-Parameter Optimization for Random Forest algorithm.

Below is the training, testing, accuracy and confusion matrix of Random Forest Classifier as follows:

Train score of trained model: 1.0

Test score of trained model: 0. 7532467532467533

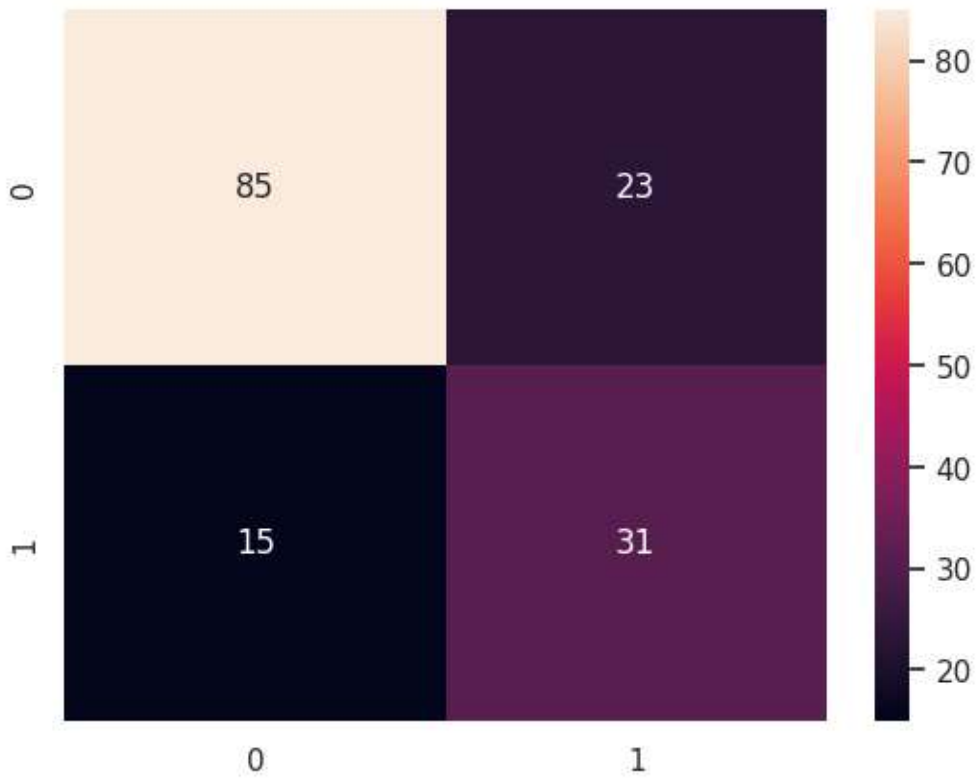
Accuracy: 75.32467532467533

Confusion Matrix:

```
[[85 23]
```

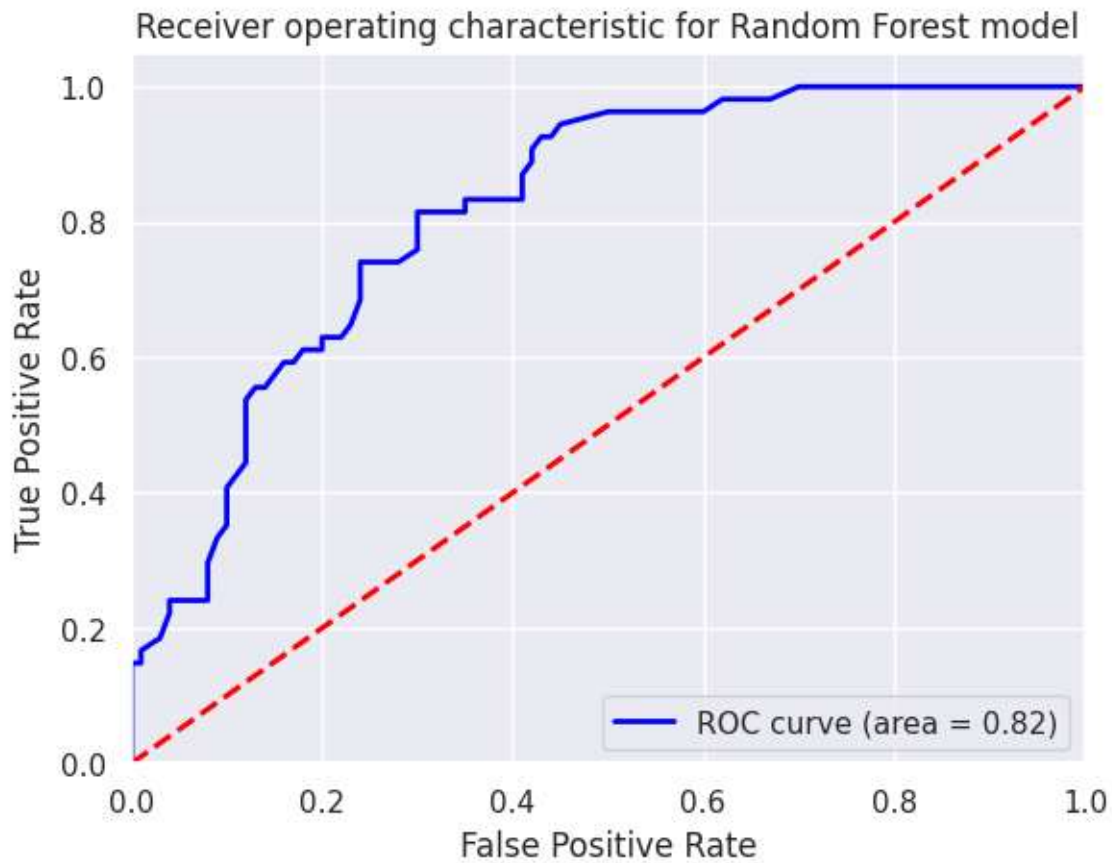
```
[15 31]]
```

Below I have provided the correlation heatmap of Confusion Matrix as follows:



*Fig: Confusion Matrix*

This confusion matrix indicates that the Random Forest Classifier predicted 85 true positives (correctly predicting patients with diabetes), 23 false negatives (incorrectly predicting patients without diabetes), 15 false positives (incorrectly predicting patients with diabetes), and 31 true negatives (correctly predicting patients without diabetes).

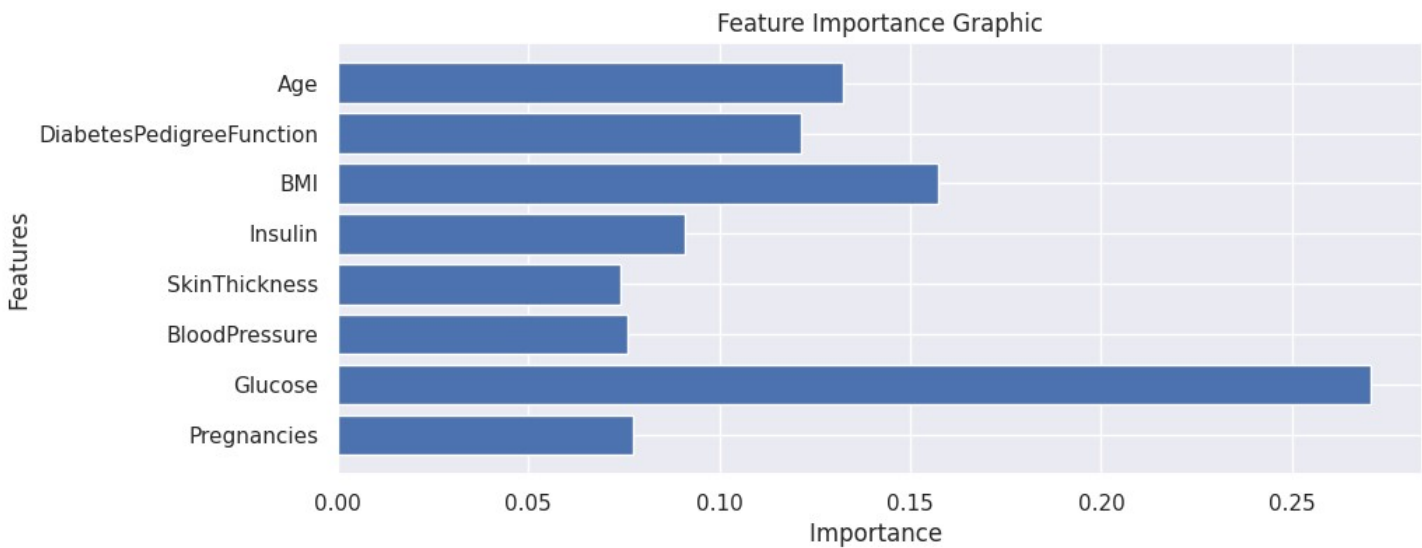


*Fig: ROC Curve for Random Forest Model*

The ROC curve with an area 0.82 indicates that the Random Forest Classifier has a good discriminatory power in distinguishing between patients with and without diabetes.

Features selection for Random Forest Classifier (selecting the most important features from the dataset)

Old Shape: (768,8) New Shape: (768,3)



*Fig: Feature Importance Plot*

By analyzing the feature importance plot, we can infer the following:

- Glucose level is the most important feature for diabetes prediction, as it has the highest feature importance score. This implies that glucose level is a crucial factor in determining whether a person has diabetes or not.
- Body mass index (BMI) is the second most important feature, which indicates that higher BMI values are associated with an increased risk of diabetes.
- Age, Diabetes Pedigree Function, and Insulin are also important factors, indicating that these factors may have a significant impact on diabetes prediction.
- Skin thickness, blood pressure, and pregnancies have relatively lower feature importance scores, suggesting that they may not be as crucial in predicting diabetes compared to other features.

### **Project Impact:**

- **Early Detection:** Early detection of diabetes through a prediction model can lead to earlier intervention and better management of the condition. This can result in improved health outcomes and reduced healthcare costs.
- **Personalized Treatment:** A diabetes prediction model can help healthcare providers personalize treatment plans for patients based on their individual risk factors, resulting in more targeted and effective interventions.
- **Health Education:** A diabetes prediction model can raise awareness of diabetes risk factors and

encourage individuals to adopt healthier lifestyles to prevent or manage the condition.

- **Resource Allocation:** By identifying high-risk individuals, a diabetes prediction model can help healthcare providers allocate resources and interventions more efficiently and effectively.
- **Research:** A diabetes prediction dataset can also be used for research purposes, such as identifying new risk factors, developing more accurate prediction models, and improving the understanding of the disease.