“From Data to Diagnosis: The Power of Machine Learning in Predicting Diabetes”

Introduction:

The main characteristic of diabetes that increases the risk of microvascular damage (retinopathy, nephropathy, and neuropathy) is the degree of hyperglycemia. It is linked to a lower life expectancy, considerable morbidity from certain diabetes-related microvascular complications, a higher risk of macrovascular problems such as peripheral vascular disease, ischemic heart disease, and stroke, and a lower quality of life [1]. The Health and Family Welfare Ministry of India produced a study report titled National Diabetes and Diabetic Retinopathy Study, which states that during the past four years, the country’s diabetes prevalence has stayed at 11.8% [2]. To study this analysis and learn more about the risk factors for diabetes, so created early detection prediction models.

Data Description:

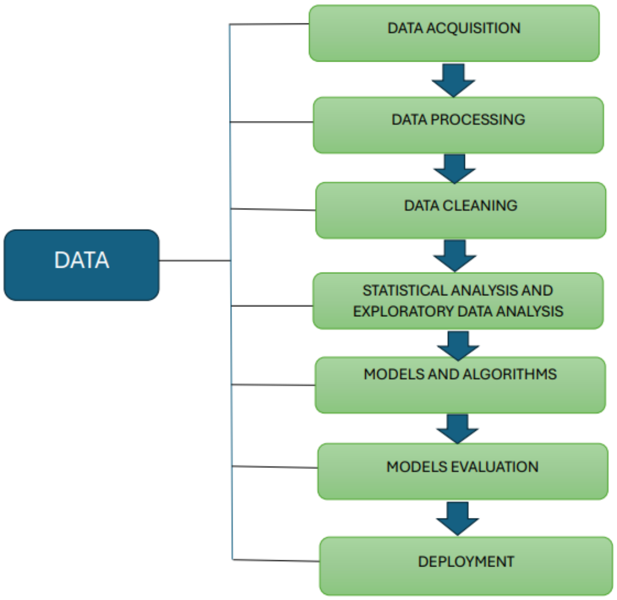
Nine properties total eight features and one target feature with 768 events make up the dataset. Among the features are:

1. Pregnancies: The total number of pregnancies the person has experienced.
2. Glucose: To express the glucose level in the blood.
3. Blood pressure: measured in millimeters of mercury (mm Hg).
4. Skin Thickness: The thickness of the skin folds in the triceps (mm).
5. Insulin: Serum insulin (mu U/ml) after two hours.
6. Body mass index, or BMI (weight in kg divided by height in m^2).
7. Diabetes Pedigree Function: The function assesses a person's genetic susceptibility to diabetes.
8. Age: The person’s age is expressed in years.
9. The target variable is a binary result that indicates if a person has diabetes (1) or not (0).

Methodology:

From data collection to implementation, this technique will walk you through every step assuring thorough understanding.

Figure 1: Flow Chart



The flow chart presents an organized approach to data analysis and machine learning, beginning with "Data Acquisition", which is the first step in the data collection process. After "data processing", which involves preparing and transforming the data, "data cleaning" is performed to eliminate errors or inconsistencies. The data is examined in "Statistical Analysis and Exploratory Data Analysis" to find trends and insights. "Models and Algorithms" is the next step in the process, where algorithms are used, and predictive models are created. After the models have been assessed in "Model Evaluation", the "Deployment" stage involves using effective models. This order ensures a comprehensive process from data gathering to implementation.

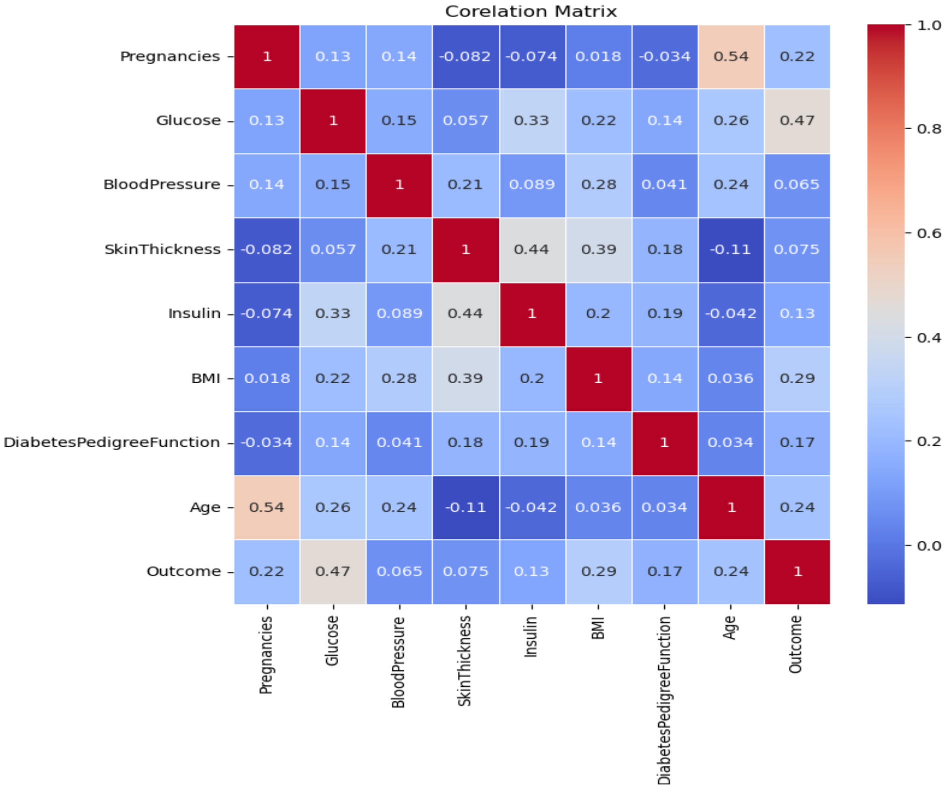
- **Data Acquisition:** The first step in the process is to obtain the Pima Indian Diabetes dataset, which is usually offered in CSV format. This dataset includes one target variable, outcome, and multiple medical predictors factors like Pregnancies, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age are among the predictor variables.
- **Data Preprocessing:** Data cleaning and preparation is the first step when the data is fed into a Python environment using libraries like Pandas.

  Missing Values: We looked for null and absent values, after figuring out that several features had meaningless zeros, so we changed every zero with the mean value, excluding the columns for outcomes and pregnancies to contrast the proportions and means of individuals with and without diabetes.

- **Exploratory Data Analysis:** Exploratory data analysis is equivalent to looking at your data for the first time to understand what's going on. It involves tasks like plotting graphs, basic statistics, and identifying any odd or intriguing patterns. At the core, it aids in improving your understanding of your data before conducting a more in-depth analysis.
  1. Statistical Analysis: The goal of statistical analysis is to measure and evaluate the importance of relationships between variables. Important variables, such as mean, median, standard deviation, and range, are used to comprehend the distribution form, central tendency, spread, shape, and use descriptive statistics. The outcomes of the statistical test highlight characteristics that show significant variations between the two groups, offering important data for predictive modeling.

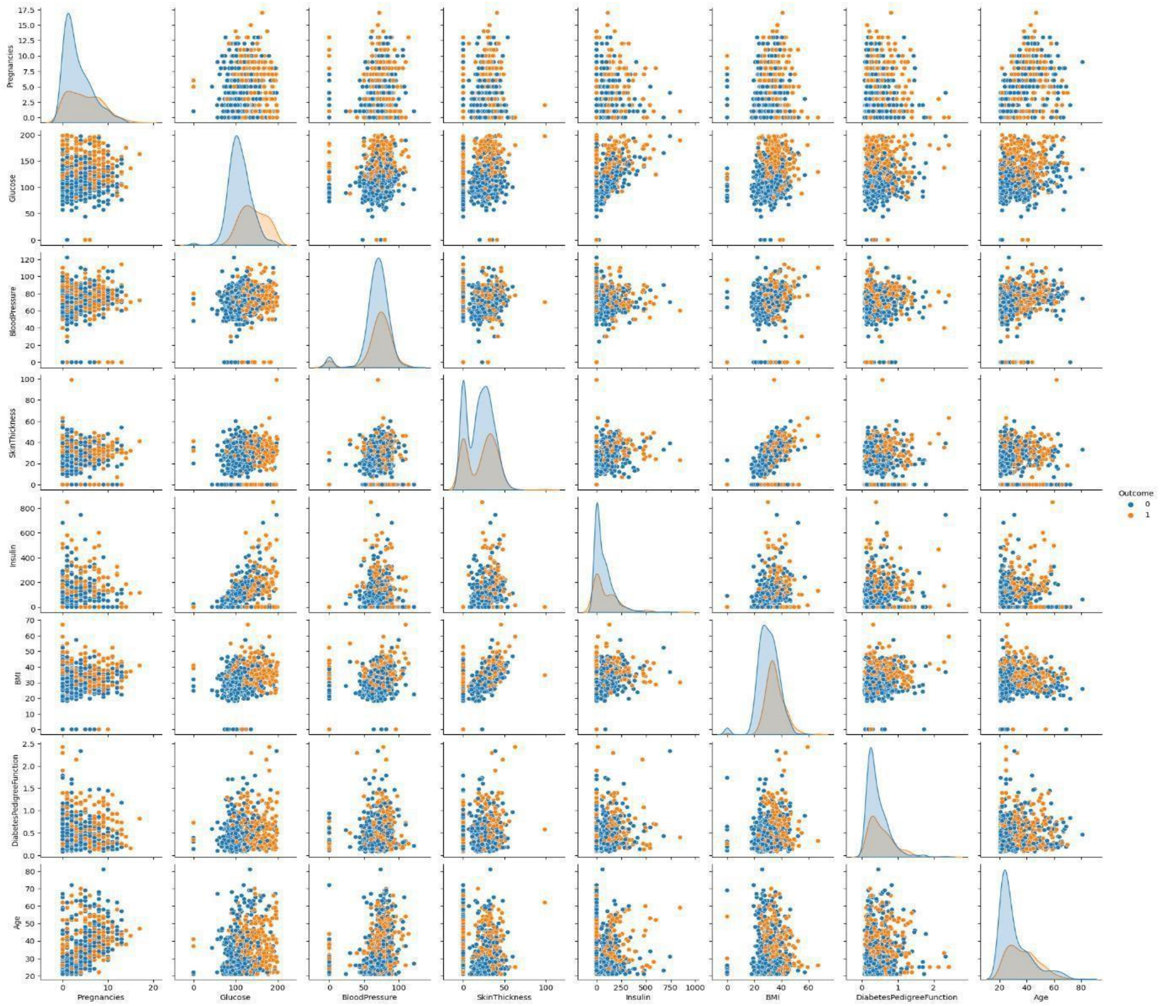
  2. Visualization: The (Figure 2 and 3) heatmap and scatterplot shows the numerical correlation coefficients, which range from -1 to 1 between various variables. Strong positive correlations are indicated by coefficients near 1, which means that when one variable rises, the other also tends to rise. One variable increases as the other drops when there is a significant negative connection, as shown by a coefficient near -1. No linear relationship is implied by a coefficient close to zero. Higher glucose levels may be linked to a higher risk of diabetes Outcome, for instance as indicated by the moderately positive correlation between Glucose and Outcome of 0.47. The correlation between Pregnancies and Age is 0.54, suggesting a moderately positive link and the possibility that the number of pregnancies would rise with age. However, the correlation between SkinThickness and Pregnanices is -0.82, indicating a very weak negative association.

Figure 2: Heatmap:



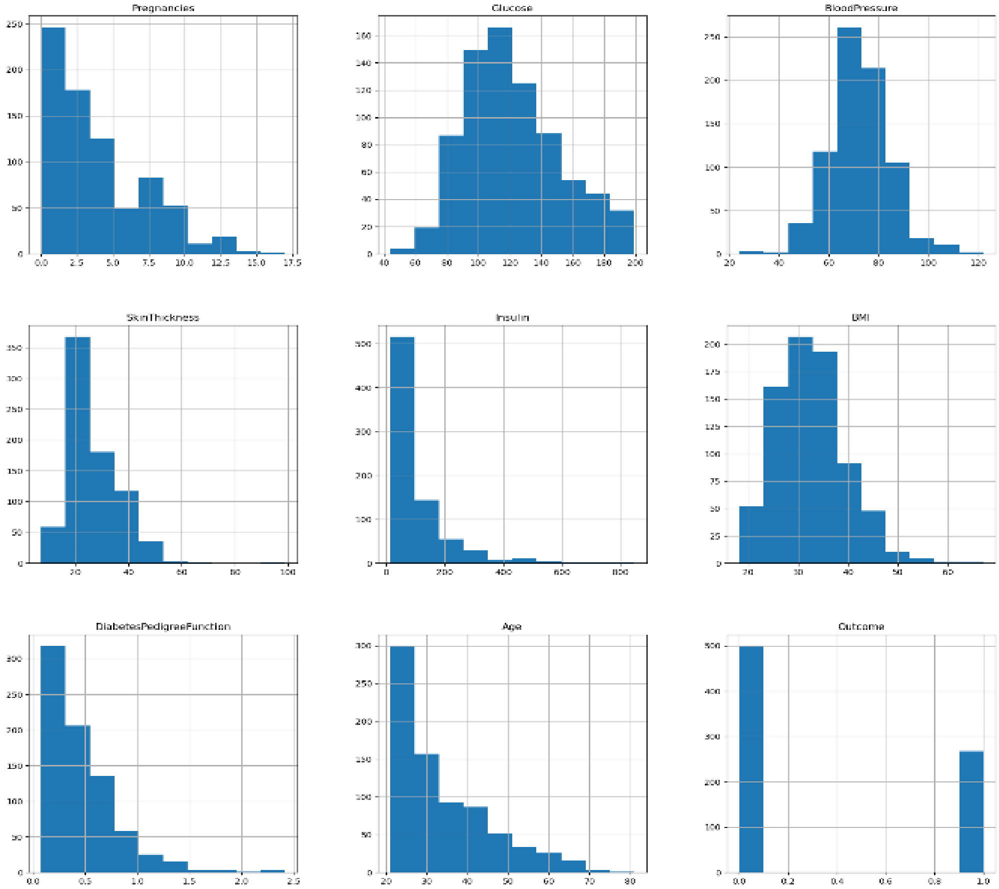
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1	0.13	0.14	-0.082	-0.074	0.018	-0.034	0.54	0.22
Glucose	0.13	1	0.15	0.057	0.33	0.22	0.14	0.26	0.47
BloodPressure	0.14	0.15	1	0.21	0.089	0.28	0.041	0.24	0.065
SkinThickness	-0.082	0.057	0.21	1	0.44	0.39	0.18	-0.11	0.075
Insulin	-0.074	0.33	0.089	0.44	1	0.2	0.19	-0.042	0.13
BMI	0.018	0.22	0.28	0.39	0.2	1	0.14	0.036	0.29
DiabetesPedigreeFunction	-0.034	0.14	0.041	0.18	0.19	0.14	1	0.034	0.17
Age	0.54	0.26	0.24	-0.11	-0.042	0.036	0.034	1	0.24
Outcome	0.22	0.47	0.065	0.075	0.13	0.29	0.17	0.24	1

Figure 3: Scatterplot:



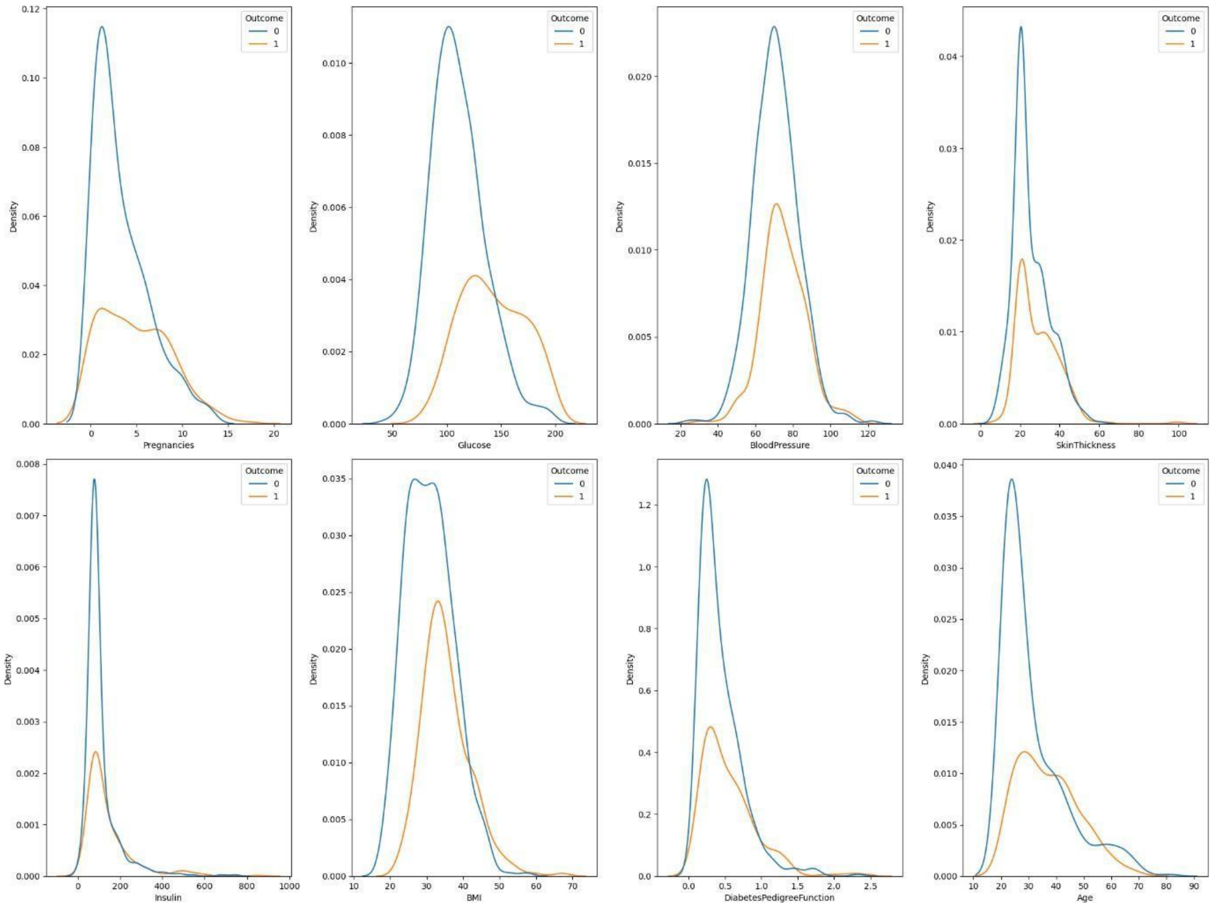
Histogram: In (Figure 4) Most people have fewer than five pregnancies, according to the Pregnancies histogram, with frequency falling as the number rises. Glucose levels peak at approximately 100, meaning that the most frequent blood sugar level is normal. The distribution of blood pressure is similarly bell-shaped, with a center of about 70-80 mmHg, because the levels of SkinThickness and Insulin are right-skewed, most issues have lower values and fewer have high interpreting. The BMI appears to be properly distributed, peaking between 30 and 35. The right-skewed DiabetesPedigreeFunction suggests that the majority of participants had fewer genetic sensitivities to diabetes. Age denotes a younger generation because frequency decreases with age. The outcome displays two different bars, one for each subject's percentage of diabetes 0, and 1.

Figure 4: Histogram



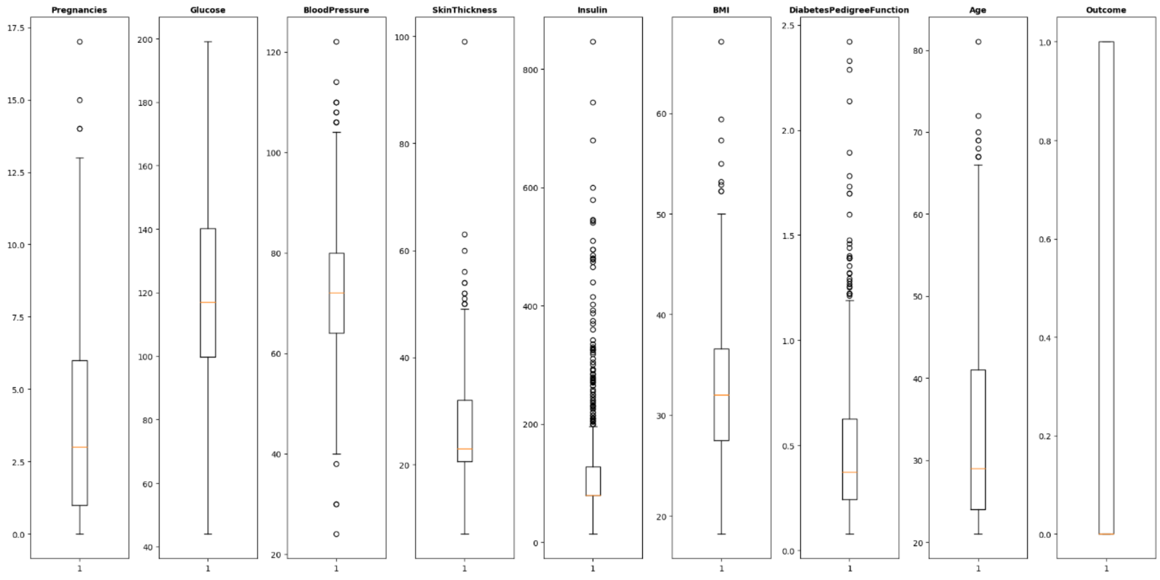
KDE plots: In (Figure 5) distribution is shown graphically in the image using a sequence of Kernel Density Estimation (KDE) plots. Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age are only a few of the variables that each plot relates to. The distribution of each variable for people without diabetes (Outcome 0) is shown by the blue line, and the distribution for people with diabetes (Outcome 1) is shown by the orange line. By comparing the distribution of each variable for the two outcomes, these plots enable the identification of potentially important factors that influence diabetes outcomes.

Figure 5: KDE plots.



Boxplots: In (Figure 6) Boxplots display the extreme median, quartiles, and outliers. Most of the results for pregnancies are few, however, there are a few significant outliers. The greater ranges of BloodPressure and Glucose suggest greater individual variability. Insulin and SkinThickness display many high outliers that are far from the median. The term BMI displays a compact box with a few high outliers, meaning that most people have a moderate BMI, and a few numbers have higher readings. A boxplot representing a binary variable is most likely misrepresented in the final plot (Outcome), DiabetesPedigreeFunction and Age demonstrate increasing spread with age.

Figure 6: Boxplots



- **Models and Algorithms:**

1. Data Preprocessing: By feature splitting, the data will be divided into X and Y. All the independent variables will be stored in X, and the only dependent feature in Y will be the outcome. Then StandardScaler is used to normalize the features to ensure that they have a mean of 0 and a standard deviation of 1. This is important because it prevents features that have larger scales from outweighing features that have smaller scales, which can lead many machine learning algorithms to perform poorly. The dataset is then divided into 70% and 30%, into training (X_train, y_train) and testing data (X_test, y_test), whereby the model is trained using training data and tested to see if it is appropriately predicting the outputs.

2. Model Selection: The dataset is used to predict binary outcomes (diabetes or no diabetes) from a set of diabetes predictors, so we can consider the following model selection.

1. Decision Tree
2. Logistic Regression
3. SVM (Support Vector Machine)
4. Random Forest
5. KNN (K-Nearest Neighbors)

3. Model Training:

Decision Tree: The DecisionTreeRegressor class must first be imported from sklearn. Tree. Use the instances fit () method and provide the training data (X_train, y_train) this technique trains the models.

Logistic Regression: Start by importing the LogisticRegression class from sklearn. linear_model. Use the fit () method with training data (X_train, y_train) to train the model. It then gains knowledge of the weight that reduces prediction errors for the target.

SVM (Support Vector Machine): The SVC class (Support Vector Classifier) is a kind of SVM used for classification tasks that is imported from sklearn.svm. To train the SVM model, use the fit () method with training data. The goal of this stage is to identify the hyperplane in your dataset that best divides the classes.

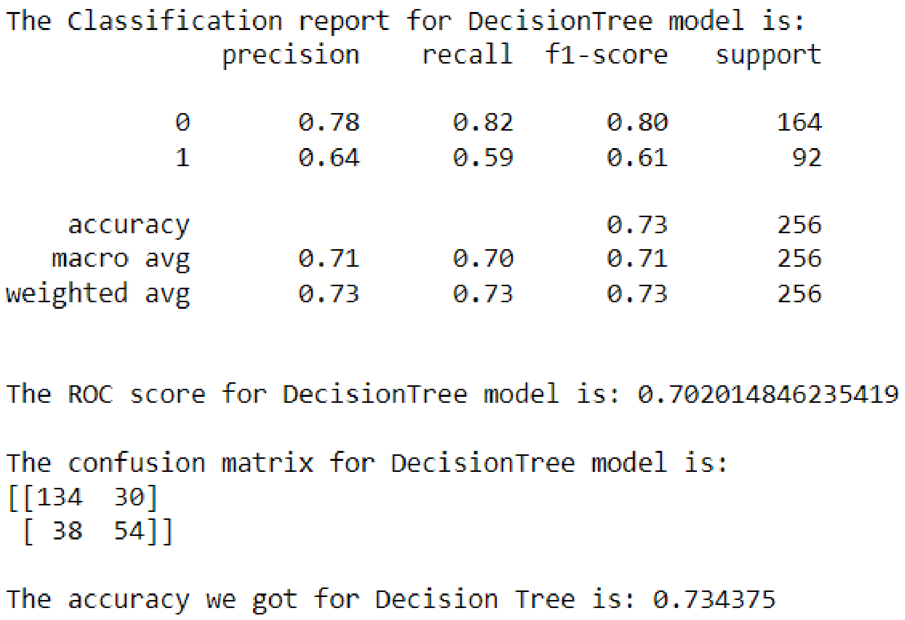
Random Forest: Make a model instance by importing the RandomForestClassifier from sklearn.ensemble. Call the fit function on your model instance, passing in training data. Through the construction of serval decision trees on different dataset subsamples and the averaging of their predictions, this technique trains the Random Forest model.

KNN (K-Nearest Neighbors): Start by importing KNeighborClassifier from sklearn.neighbors. Pass the training data to the KNN instance fit () function. This technique prepares the model for prediction by modifying the data.

- **Models Evaluation:**

Decision Tree: An overall accuracy of 73.43% (Figure 7) shows statistics reflecting the effectiveness of a Decision Tree model, including metrics like Precision, F1-score, and Recall.

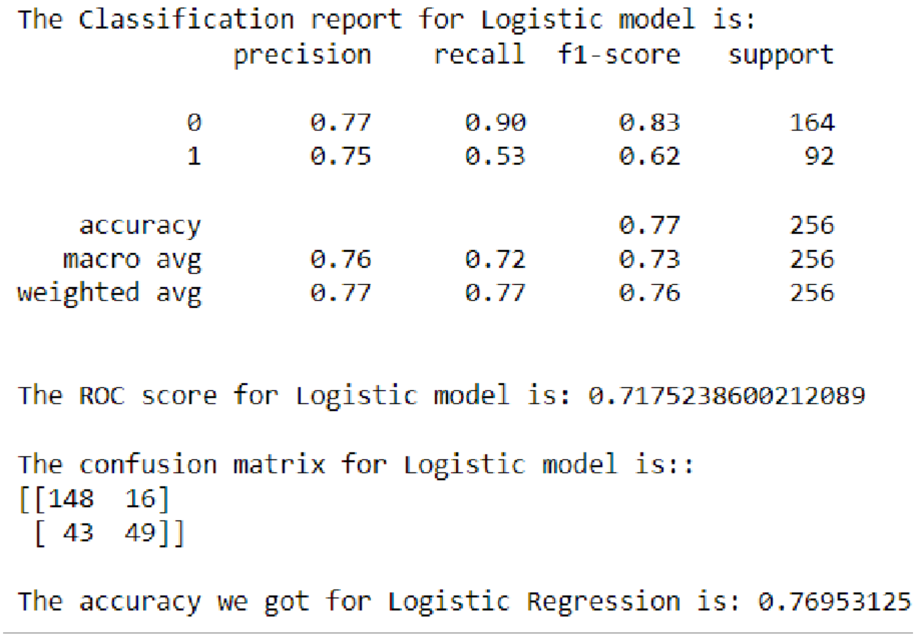
Figure 7: Classification Report for Decision Tree.



```
The Classification report for DecisionTree model is:
              precision    recall  f1-score   support

           0       0.78      0.82      0.80       164
           1       0.64      0.59      0.61        92

    accuracy                           0.73       256
   macro avg       0.71      0.70      0.71       256
weighted avg       0.73      0.73      0.73       256


The ROC score for DecisionTree model is: 0.702014846235419

The confusion matrix for DecisionTree model is:
[[134  30]
 [ 38  54]]

The accuracy we got for Decision Tree is: 0.734375
```

Logistic Regression: With a total accuracy of almost 77%, (Figure 8) summarizes the accuracy of a Logistic Regression model along with other measures like Precision, Recall, and F1-score that evaluate its performance on a classification report, with a respectable ROC score of 0.7175.

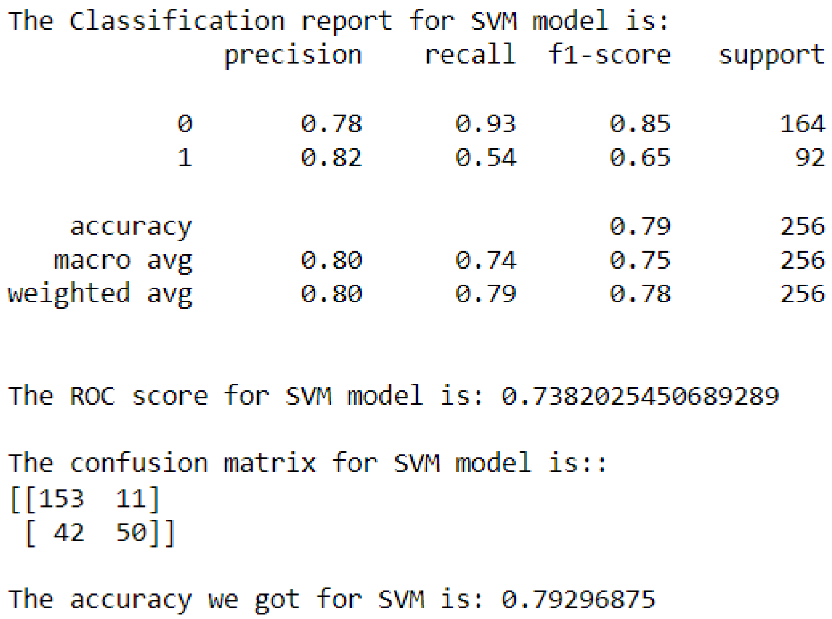
Figure 8: Classification Report for Logistic Regression.



```
The Classification report for Logistic model is:
              precision    recall  f1-score   support

           0       0.77      0.90      0.83       164
           1       0.75      0.53      0.62        92

    accuracy                           0.77       256
   macro avg       0.76      0.72      0.73       256
weighted avg       0.77      0.77      0.76       256


The ROC score for Logistic model is: 0.7175238600212089

The confusion matrix for Logistic model is::
[[148  16]
 [ 43  49]]

The accuracy we got for Logistic Regression is: 0.76953125
```

SVM (Support Vector Machine): (Figure 9) shows a performance for the SVM model used for classification. With Precision, Recall, and F1-score showing good performance on one class but modest performance on the other, the model has a comparatively high accuracy of 79.29%, given its ROC score of 0.738, the model appears to be reasonably well at differentiating between the two groups.

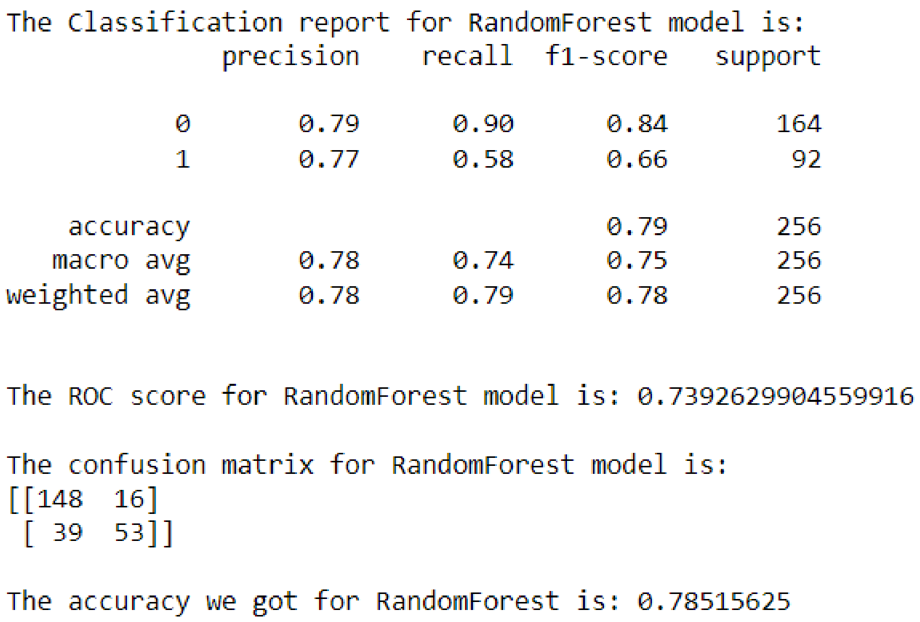
Figure 9: Classification Report for SVM.



```
The Classification report for SVM model is:
               precision    recall  f1-score   support

           0       0.78      0.93      0.85       164
           1       0.82      0.54      0.65        92

    accuracy                           0.79       256
   macro avg       0.80      0.74      0.75       256
weighted avg       0.80      0.79      0.78       256


The ROC score for SVM model is: 0.7382025450689289

The confusion matrix for SVM model is::
[[153  11]
 [ 42  50]]

The accuracy we got for SVM is: 0.79296875
```

Random Forest: An overview of a RandomForestClassifier performance is shown in (Figure 10), showcasing a 78.5% overall accuracy along with particulars about Precision, F1-score, and Recall for two classes. A sufficient capacity to discriminate between the classes beyond chance is indicated by the ROC score of 0.739.

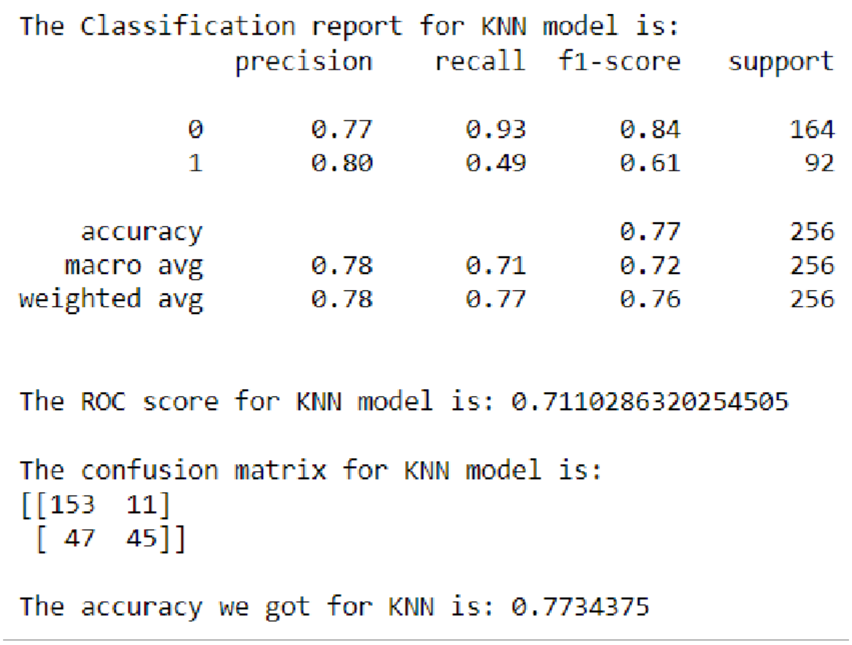
Figure 10: Classification report for Random Forest.



```
The Classification report for RandomForest model is:
               precision    recall  f1-score   support

           0       0.79      0.90      0.84       164
           1       0.77      0.58      0.66        92

    accuracy                           0.79       256
   macro avg       0.78      0.74      0.75       256
weighted avg       0.78      0.79      0.78       256


The ROC score for RandomForest model is: 0.7392629904559916

The confusion matrix for RandomForest model is:
[[148  16]
 [ 39  53]]

The accuracy we got for RandomForest is: 0.78515625
```

KNN (K-Nearest Neighbors): The (Figure 11) represents a KNN model classification report, which has an overall accuracy of 77.34%. It suggests that the model does well in recognizing the negative class (0) but less well in recognizing the positive class (1). With a ROC score of 0.711, the model appears to be able to discriminate between the two classes with some degree of accuracy.

Figure 11: Classification report for KNN.



```
The Classification report for KNN model is:
              precision    recall  f1-score   support

           0       0.77      0.93      0.84       164
           1       0.80      0.49      0.61        92

    accuracy                           0.77       256
   macro avg       0.78      0.71      0.72       256
weighted avg       0.78      0.77      0.76       256


The ROC score for KNN model is: 0.7110286320254505

The confusion matrix for KNN model is:
[[153  11]
 [ 47  45]]

The accuracy we got for KNN is: 0.7734375
```

- **Confusion Matrix:** Additional confusion matrix measures need to be run to calculate the model's accuracy score. The performance of a model can be seen with the use of this matrix. The (Figures 12, 13, 14, 15, and 16) displays the quantity of Actual Positive, Actual Negative, Predicted Positive, and Predicted Negative. In medical diagnostics, where a Predicted Negative might have very high consequences, this breakdown is very important for understanding the model behaviour.

Figure 12: Decision Tree.

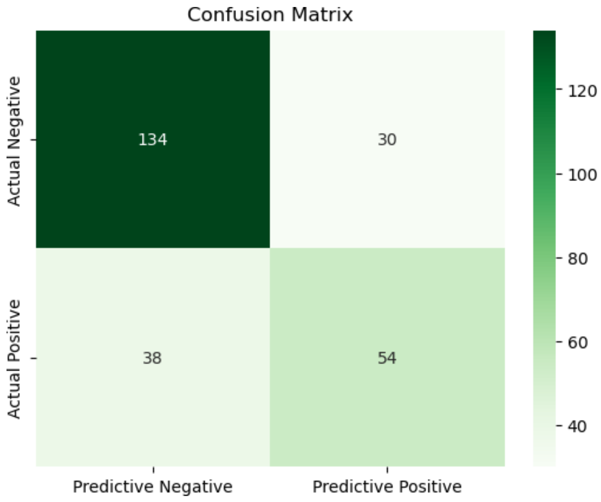


Figure 13: Logistic Regression.

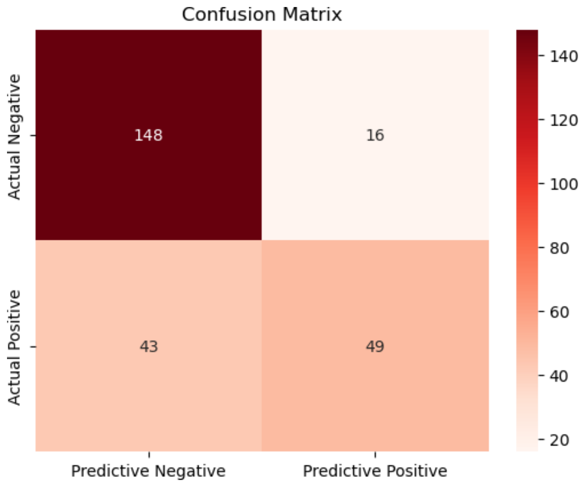
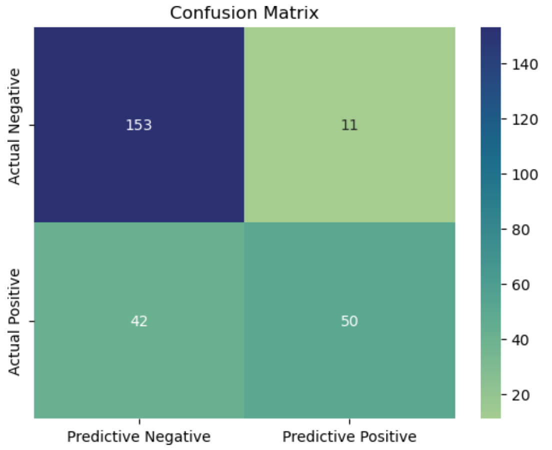


Figure 14: SVM.

Figure 15: Random Forest.



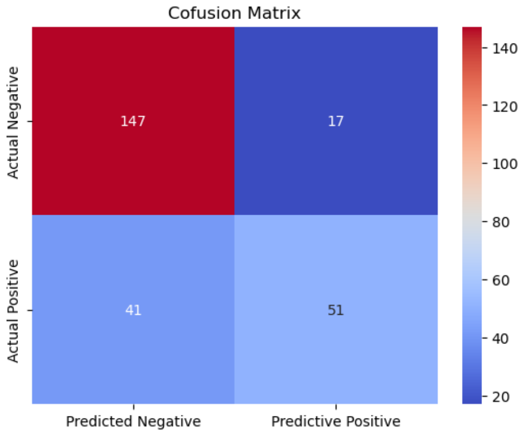
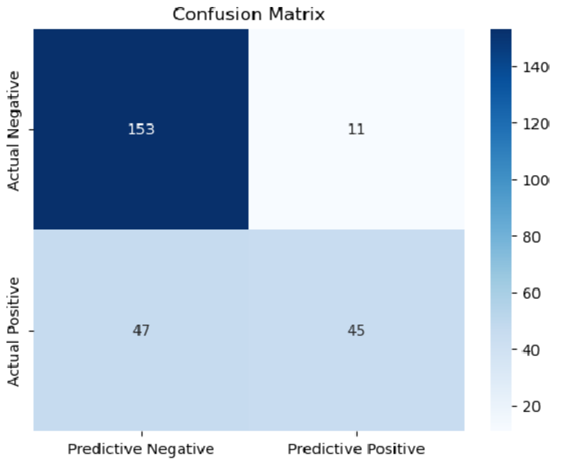
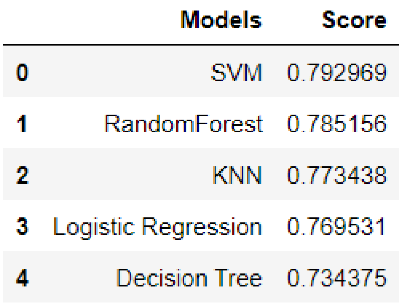


Figure 16: KNN.

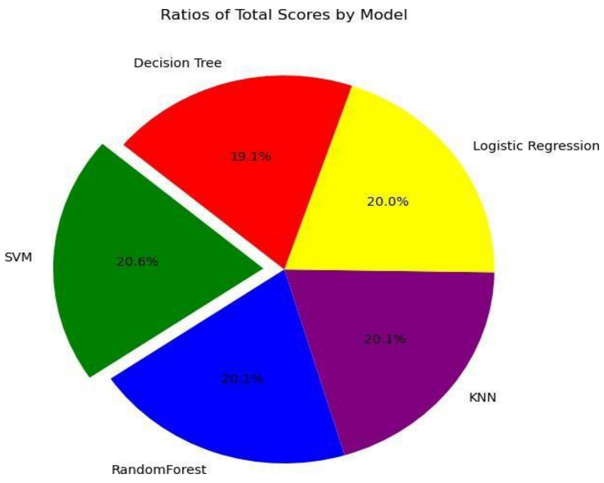


Result and Discussion:

With an accuracy of 79.29%, SVM stood out among the other algorithms, trailed by Random Forest, KNN, Logistic Regression, and Decision Tree. These scores offer a clear comparison of the accuracy with which each model has been able to determine whether a patient has diabetes or not. The distribution of overall performance ratings among the five predictive models used to forecast diabetes is shown in the above pie chart. With 20.6% of the total points, SVM is the top model. Random Forest, KNN, and Logistic Regression trials closely after, with the Decision Tree having the lowest share at 19.1%. The model's performances are seen to be balanced in this representation, with none exceeding the others.



	Models	Score
0	SVM	0.792969
1	RandomForest	0.785156
2	KNN	0.773438
3	Logistic Regression	0.769531
4	Decision Tree	0.734375



How each algorithm tackles the task of diabetes diagnosis and the reason why SVM is better than others will be the main topics of discussion while analyzing this result. One possible explanation for SVM efficiency could be its capacity to process non-linear data and identify the hyperplane that best divides the classes. Moreover, Random Forest and KNN outperformed Decision Tree and Logistic Regression, perhaps because they benefited from instance-based and ensemble methods, respectively, and were more resistant to errors. Although its poor efficiency, the Decision Tree may still be useful in situations where comprehending the decision-making process is essential due to its simplicity of use and understanding. The performance balance of the models highlights how crucial it is to choose the best model for the task by considering the particulars of the dataset and the problem itself.

Conclusion:

Overall, in this report, we provide the findings from our thorough study, which included statistical analysis, data exploration and visualization, data pre-processing, and model building. Our results provide valuable insight into the variables linked to diabetes and demonstrate the efficacy of machine learning methods in establishing the probability of developing the disease. The potential of machine learning as a tool for medical diagnostic decision support. Our model's accuracy in patient classification shows how technology can support early identification and prompt action, both of which are essential for managing chronic conditions like diabetes.

Finally, our report emphasizes how important it is to have reliable datasets. Enough and high-quality data are necessary for the creation of trustworthy predictive models. In addition to contributing to the growing body of knowledge in healthcare analytics, our study lays the groundwork for future studies that will improve diabetes management and preventive strategies.

References:

1. Setacci, C., de Donato, G., Setacci, F. and Chisci, E., 2009. Diabetic patients: epidemiology and global impact. *J Cardiovasc Surg (Torino)*, *50*(3), pp.263-73.

2. Lakhwani, K., Bhargava, S., Hiran, K.K., Bundele, M.M. and Somwanshi, D., 2020, December. Prediction of the onset of diabetes using artificial neural network and pima indians diabetes dataset. In *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)* (pp. 1-6). IEEE.