Principles of Data Mining and Machine Learning (2022 MOD007892 TRI2 F01CAM)

Element 010 Task 2

Name: Aatiqa Nawaz

Student Id: 2146625

Contents

Introduction:

Diabetes is a disease which happens with increase in glucose content in blood. We will use different machine learning models in this study to check which machine learning algorithms works best. We will do different type of evaluations on our data using evaluation metrics. Furthermore, it's thought that one in two persons with diabetes go untreated, increasing their risk of developing complications like kidney failure, blindness, and amputations. In order to prevent and control its rising prevalence, diabetes is a serious worldwide health problem that needs immediate attention. Early detection and management of diabetes are crucial for preventing or delaying the onset of complications and improving patients' quality of life. However, the diagnosis of diabetes can be challenging, as it requires the consideration of various risk factors such as age, family history, and medical history. Machine learning algorithms have emerged as a promising tool for predicting the risk of diabetes. These algorithms can analyse large and complex datasets to identify patterns and make accurate predictions based on medical predictor variables. By utilizing machine learning algorithms, healthcare providers can develop predictive models that can aid in early intervention strategies and improve the overall management of diabetes, leading to better health outcomes for patients.

Problem Statement:

Predict whether a patient has diabetes or not by using a dataset of patients. Develop an effective model that is the best one for the given classification task.

Dataset description:

Dataset of Patient includes these variables like pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes, Pedigree Function, Age and Outcome. The dataset has 768 instances, with 8 unique, independent properties for each row. 'Outcome', a dependent variable in the dataset, indicates whether or not a person has diabetes. The dataset includes the following features of following datatypes:

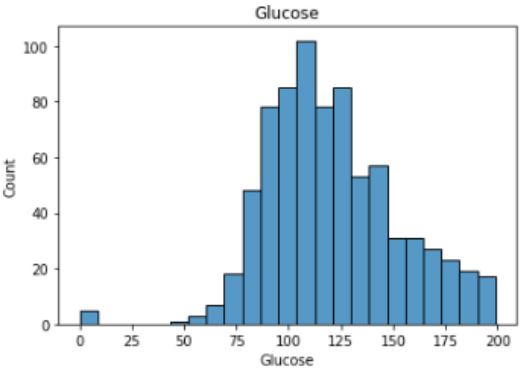
```
Pregnancies                 int64
Glucose                     int64
BloodPressure               int64
SkinThickness               int64
Insulin                     int64
BMI                       float64
DiabetesPedigreeFunction  float64
Age                         int64
Outcome                     int64
```

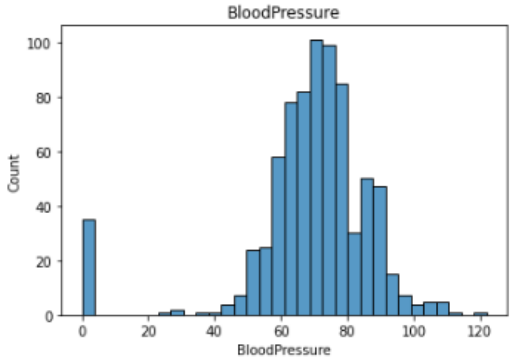
- **Description of the dataset:**

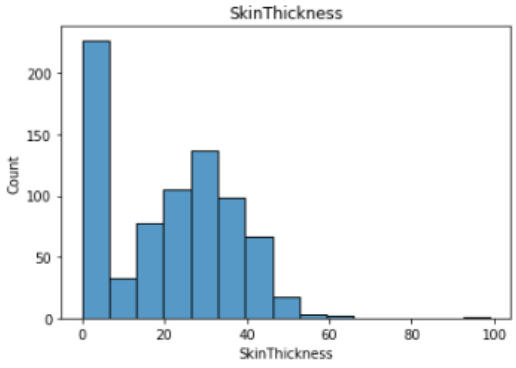
The method described will help to see how data has been spread for numerical values. We can clearly see the minimum values mean values, different percentile values, and maximum values.

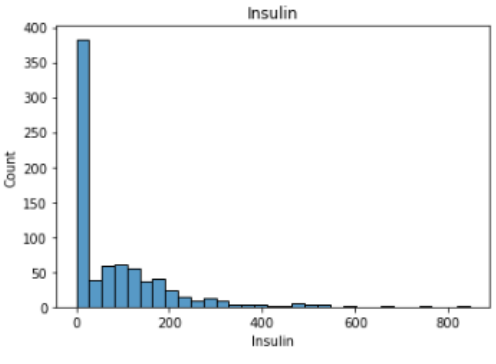
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

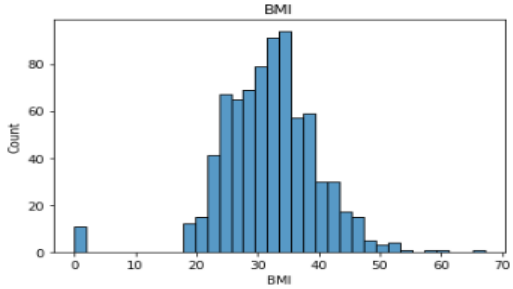
- **Histplot of all graphs:**











We can check that many values are lying at 0 for these columns, according to our description as well. We will replace this values with mean of the columns.

- **Data cleaning**

Drop the Duplicates: check if there are any duplicate rows or not, if these exist then we should remove from the data frame.

Before drop and after drop of the duplicates the data set has the same shape which means there were no duplicates in the dataset.

The next step is to check the Null Values using is null function of pandas. This results in No null values in dataset.

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
```

Number of zeroes in dataset:

```
No. of zero values in Glucose 5
No. of zero values in BloodPressure 35
No. of zero values in Insulin 374
No. of zero values in BMI 11
No. of zero values in DiabetesPedigreeFunction 0
No. of zero values in SkinThickness 227
```

We are replacing the 0 values with mean of dataset for Glucose, Blood Pressure, Insulin, BMI and Skin Thickness.

```
No. of zero values in Glucose 0
No. of zero values in Blood Pressure 0
No. of zero values in Insuin 0
No. of zero values in Skin Thickness 0
No. of zero values in BMI 0
```

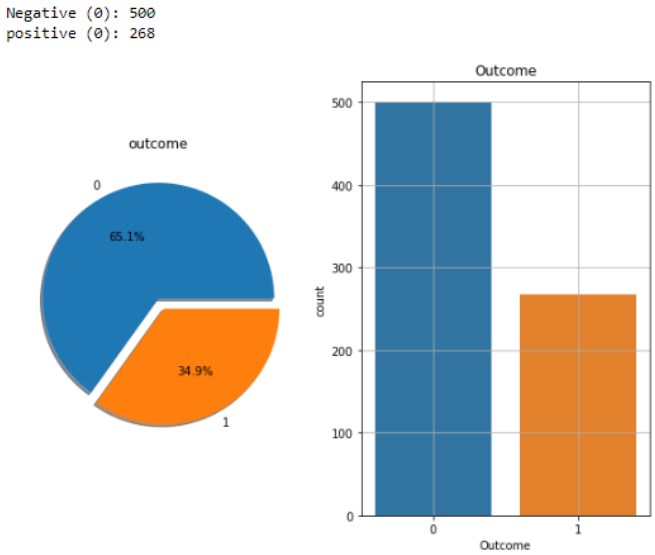
Description of dataset after imputing

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	121.681605	72.254807	26.606479	118.660163	32.450805	0.471876	33.240885	0.348958
std	3.369578	30.436016	12.115932	9.631241	93.080358	6.875374	0.331329	11.760232	0.476951
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	0.078000	21.000000	0.000000
25%	1.000000	99.750000	64.000000	20.536458	79.799479	27.500000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	79.799479	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Minimum values for numerical columns have changed after imputation.

- **Data visualizatios:**

Count plot for outcome variable

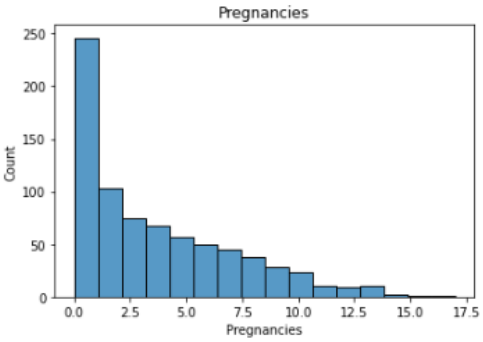


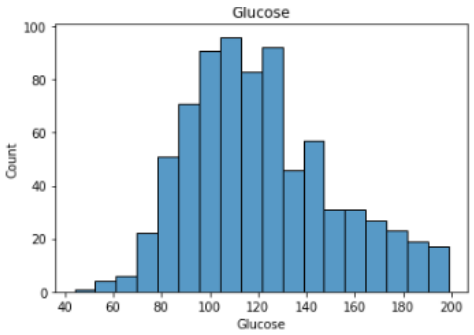
Count plot for outcome variable

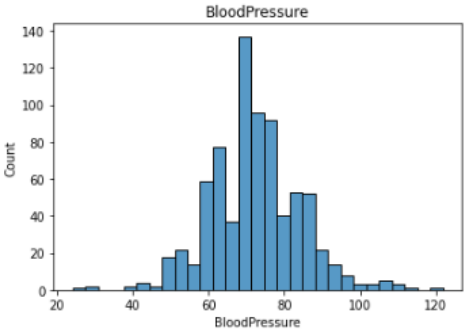
There are total of 768 records present in our dataset out of which 500 people are negative and rest 268 are having diabetes. 0 stands for negative and 1 stands for positive in diabetes.

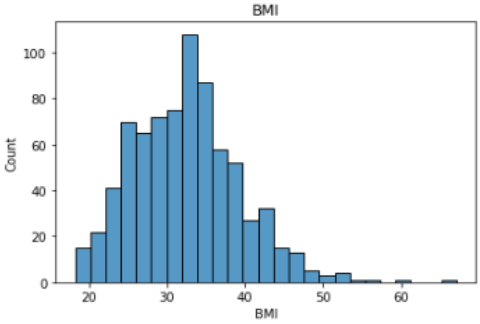
- **Histograms:**

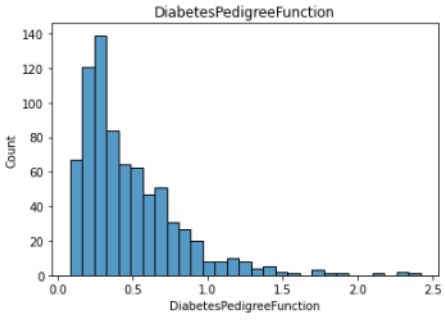
Histograms are one of the most common graphs used to display numeric data.

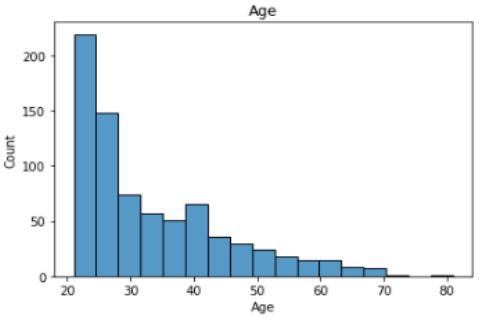


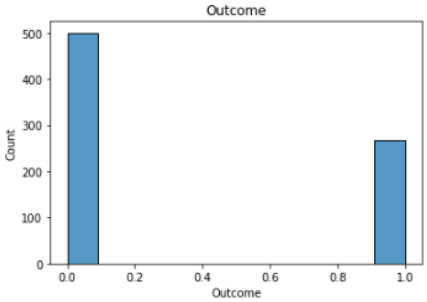










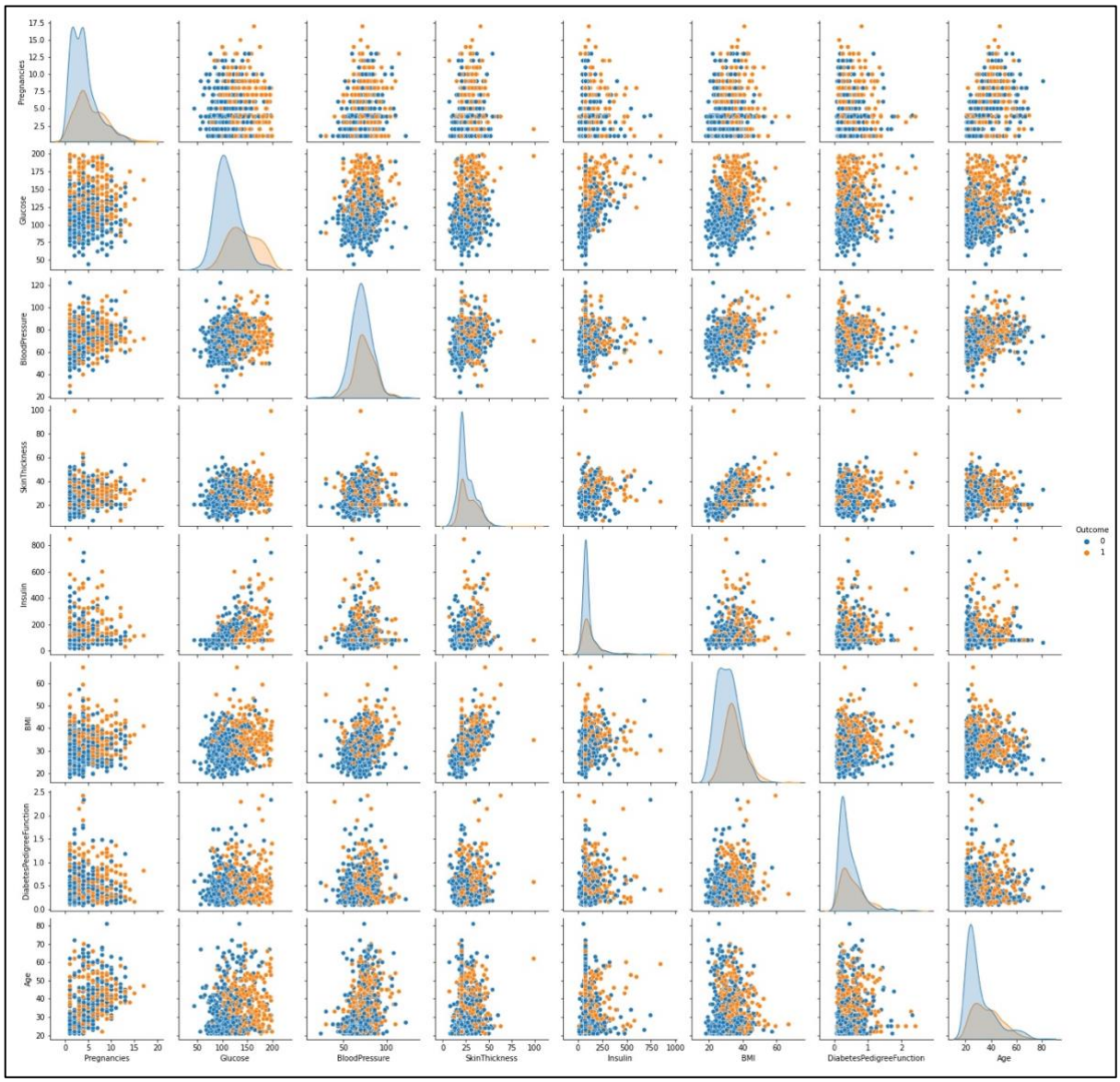


Distribution of features have changed after imputation.
Highest number of people have blood pressure 70 and glucose 100.

- **Analysing relationships between variables:**

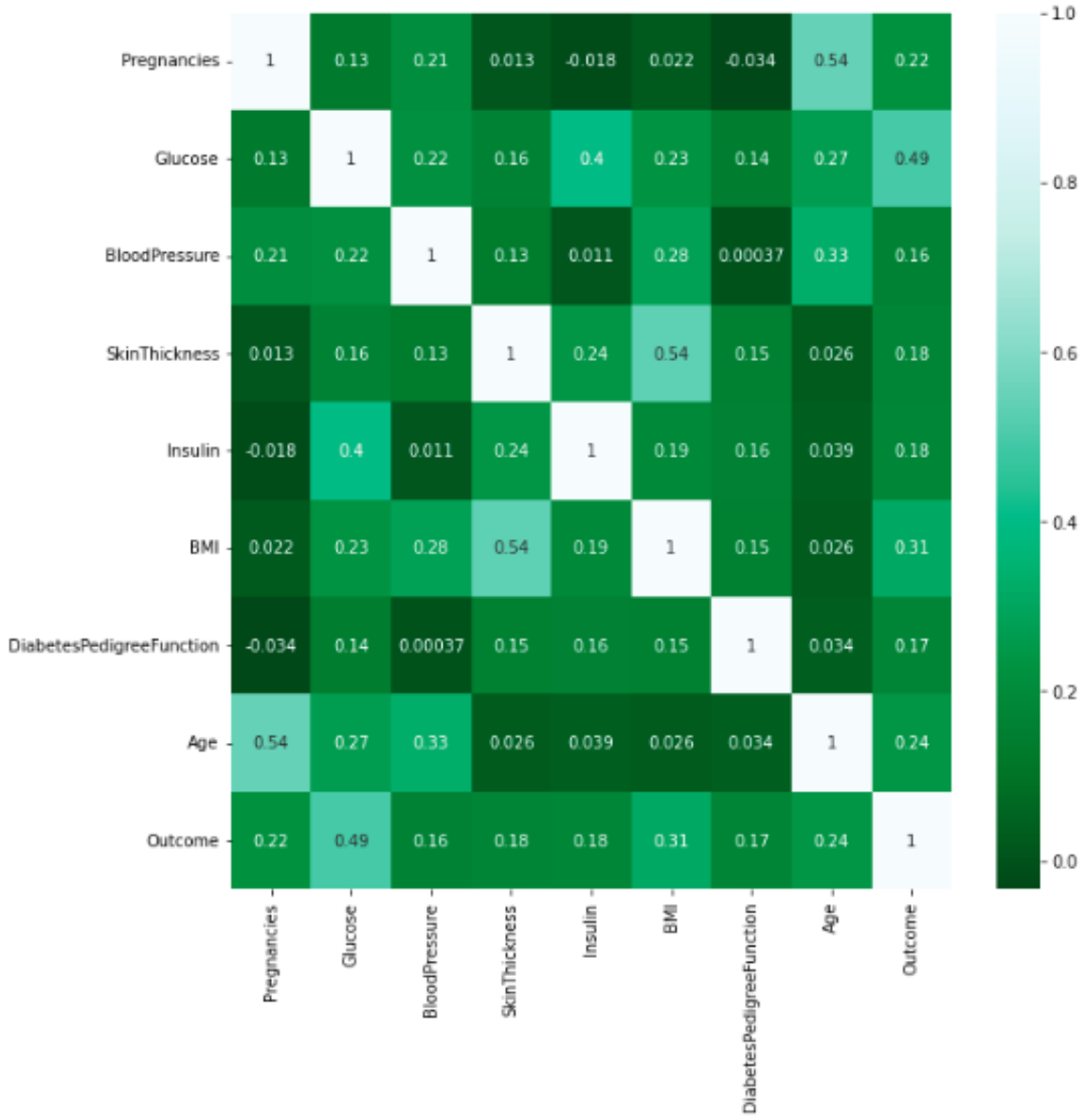
a. Pair plot:

Seaborn Pair plot is a great way to create scatterplots between all of your variables.



For the Glucose variable it is observed that chances of getting diabetes increase as Glucose goes above certain value.

b. Correlation matrix:



We can see outcome is medium correlated with glucose.

The strength of a relationship between two variables is measured by correlation analysis. The correlation coefficient, which indicates how much one variable changes when the other one does, is evaluated by the correlation analysis. You can determine the linear relationship between two variables using correlation analysis.

I get to learn how dependent certain feature variables are on specific target variables when we correlate them with the feature variables. According to the correlation heatmap, there is a strong association between the outcome and [pregnancy, glucose, body mass index, age, and insulin].

▪ Data pre-processing:

1. Using the train-test split function found in the sklearn library's model selection function, the dataset will be divided into X and Y for pre-processing. The test size will be 30%. X will contain the highlights in general, for example, Pregnancies, Glucose, Pulse, Skin Thickness, Insulin, BMI, Diabetes Family Capability, and Age. The outcome, or dependent variable, will be in Y. The information will be scaled utilizing standard scaler prior to being utilized for AI models. Encoding is not required because the dataset does not contain categorical columns.

2. Using the train-test split function from the sklearn library's model selection function, the dataset will be divided into X and Y for pre-processing. The test size will be 30%.

3. Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, and Age will all be included in X. The outcome, or dependent variable, will be in Y. Prior to being fed to the models, the data will be scaled using a standard scaler. Encoding is not required because the dataset does not contain categorical columns.

▪ Model Training:

1. Different classification models such as logistic regression, K-nearest neighbour classifier, Naïve-Bayes, Support Vector Machine, decision tree and random forest will be trained using the sklearn library. Once the models are trained, the predict function will be called to predict the results for every model.

2. To evaluate the models, different evaluation metrics will be evaluated and compared..

▪ Model Evaluation and Comparison:

To evaluate the models, different evaluation metrics such as precision, accuracy score, recall, F1-score, specificity and sensitivity and AUC-RUC curve will be employed to determine the best model. Below are the results for different evaluation metrics for classifiers with default parameters.

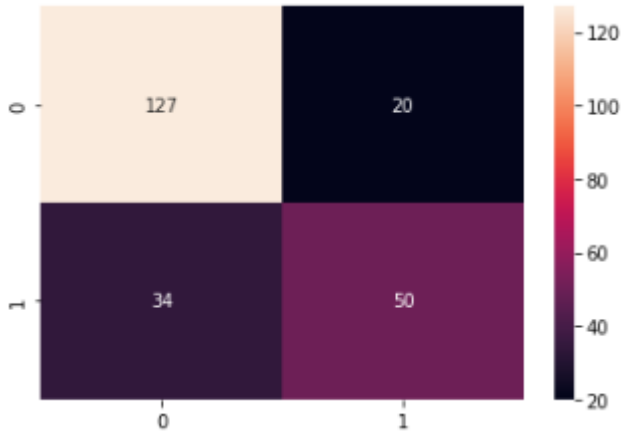
a. Accuracy Score

```
Accuracy score of logistic Regression: 76.62%
Accuracy score of KNN: 75.76%
Accuracy score of Naive-bayes: 72.73%
Accuracy score of Support Vector Machines: 79.65%
Accuracy score of Decision Tree: 71.00%
Accuracy score of Random Forest: 76.19%
```

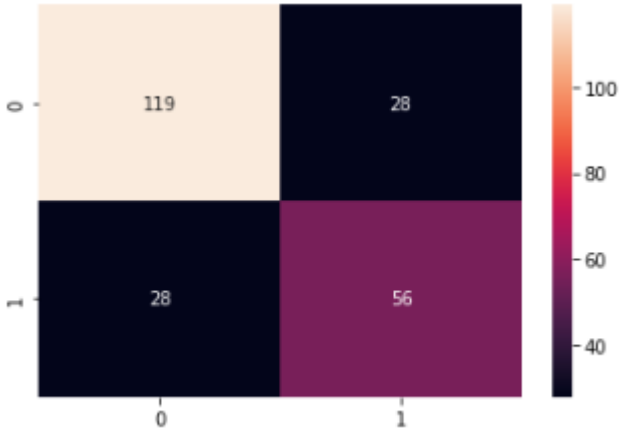
SVM shows highest accuracy of 76%

b. Confusion matrix:

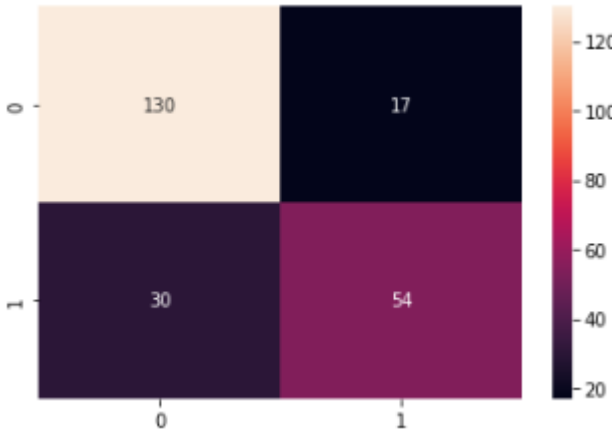
1. Confusion Matrix of "Logistic Regression"



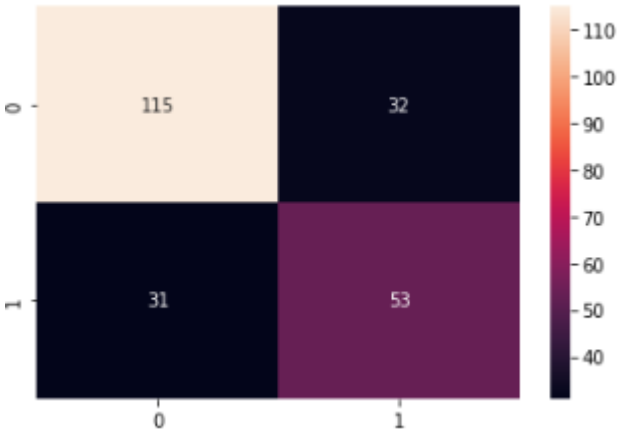
2. Confusion Matrix of "K Nearest Neighbour"



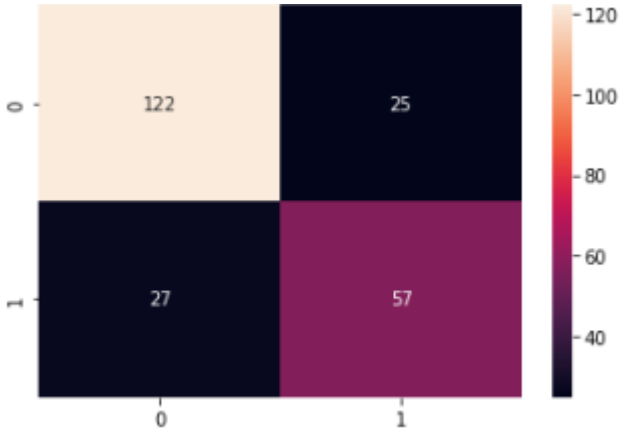
3. Confusion Matrix of "Support Vector Machines"



4. Confusion Matrix of "Naive-Bayes Classifier":



5. Confusion Matrix of "Decision Tree"



c. Classification report:

1. Classification report for Logistic regression:

Classification report for Logistic Regression :	precision	recall	f1-score	support
0	0.79	0.86	0.82	147
1	0.71	0.60	0.65	84
accuracy			0.77	231
macro avg	0.75	0.73	0.74	231
weighted avg	0.76	0.77	0.76	231

2. Classification report for K Nearest Neighbour:

Classification report for K Nearest Neighbour :	precision	recall	f1-score	support
0	0.81	0.81	0.81	147
1	0.67	0.67	0.67	84
accuracy			0.76	231
macro avg	0.74	0.74	0.74	231
weighted avg	0.76	0.76	0.76	231

3. Classification report for Support Vector Machine

```
Classification report for Support Vector Machine
:              precision    recall  f1-score   support

           0       0.81      0.88      0.85       147
           1       0.76      0.64      0.70        84

    accuracy                           0.80       231
   macro avg       0.79      0.76      0.77       231
weighted avg       0.79      0.80      0.79       231
```

4. Classification report for Decision Tree

```
Classification report for Decision Tree
:              precision    recall  f1-score   support

           0       0.78      0.76      0.77       147
           1       0.60      0.62      0.61        84

    accuracy                           0.71       231
   macro avg       0.69      0.69      0.69       231
weighted avg       0.71      0.71      0.71       231
```

5. Classification report for Naive-Bayes

```
Classification report for Naive-Bayes
:              precision    recall  f1-score   support

           0       0.79      0.78      0.78       147
           1       0.62      0.63      0.63        84

    accuracy                           0.73       231
   macro avg       0.71      0.71      0.71       231
weighted avg       0.73      0.73      0.73       231
```

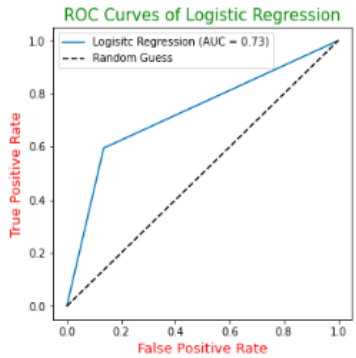
6. Classification report for Random Forest

```
Classification report for Random Forest
:              precision    recall  f1-score   support

           0       0.81      0.82      0.81       147
           1       0.67      0.67      0.67        84

    accuracy                           0.76       231
   macro avg       0.74      0.74      0.74       231
weighted avg       0.76      0.76      0.76       231
```

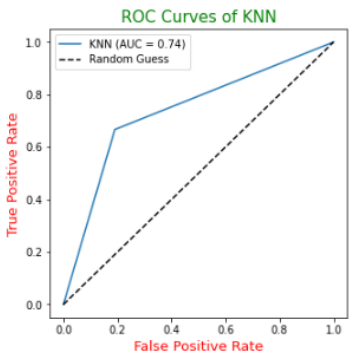
SVM, random forest, and logistic regression shows best precision and recall.

d. ROC curves:

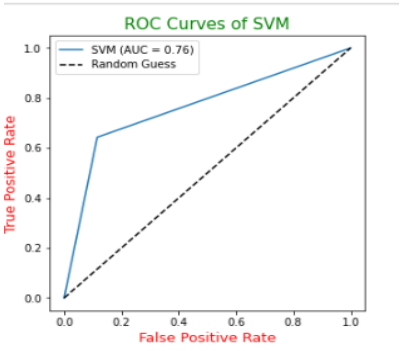
ROC curve is one the important evaluation metrics that should be used to check the performance of a classification model. It is also called relative operating characteristic curve, because it is a comparison of two main characteristics (TPR and FPR). It is plotted between sensitivity (aka recall aka True Positive Rate) and specificity (False positive Rate FPF = 1).
ROC (Receiver Operation characteristic) Curve tells us abouts how well the model can distinguish between two things (e.g. if a patient has a disease or no).
Area under Curve (AUC) help us to choose the best model amongst the models for which we have plotted the ROC curves.



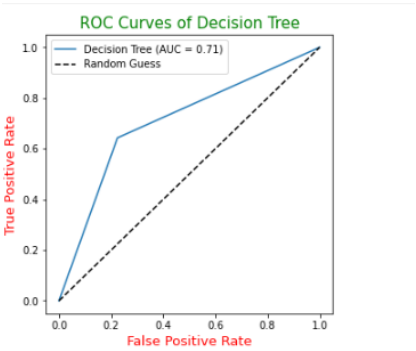
Roc score logistic regression = 0.73



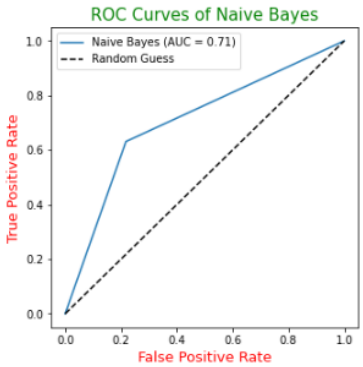
Roc score KNN= 0.74



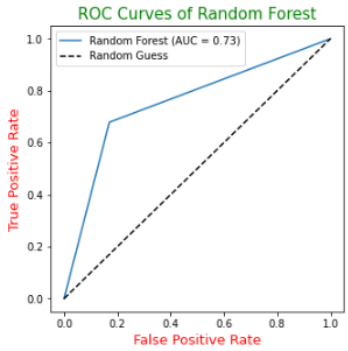
Roc score SVM= 0.74



Roc score Decision Tree= 0.71



Roc score Naive Bayes= 0.71



Roc Score Random Forest = 0.73

Support Vector machine and Naïve Bayes show the highest ROC scores.

e. Specificity and sensitivity

Logistic Regression: Specificity = 0.864, Sensitivity = 0.595
k-NN: Specificity = 0.810, Sensitivity = 0.667
Naive Bayes: Specificity = 0.782, Sensitivity = 0.631
SVM: Specificity = 0.884, Sensitivity = 0.643
Decision Tree: Specificity = 0.762, Sensitivity = 0.619
Random Forest: Specificity = 0.816, Sensitivity = 0.667

When compared we can see that Logistic regression and SVM shows highest Specificity.
Random forest and KNN shows highest sensitivity.
SVM shows overall best results with Specificity = 0.884, Sensitivity = 0.643

- # Comparative Analysis:

In the field of healthcare, ML algorithms are often used to predict and diagnose disease based on a patient's medical history and other relevant factors. In this context, the classification of patients as having or not having a particular disease is a critical task, and the accuracy of the classification model is of utmost importance. In this scenario, I have worked with a diabetes dataset to classify patients as diabetic or non-diabetic using different classification algorithms. After evaluating the performance of various models, I have found that SVM performs better than KNN, Naive Bayes, Random Forest, Decision Tree and Logistic Regression models in

terms of accuracy. The superior performance of SVM can be attributed to its ability to handle non-linear data and find optimal boundaries between different classes. The diabetes dataset may have complex relationships between different features, and SVM can capture these complex relationships to classify patients accurately. In contrast, KNN and Naive Bayes models are relatively simpler algorithms that may struggle with complex datasets such as the diabetes dataset. Decision Tree and Random Forest models can handle non-linear relationships, but they may not always provide optimal classification boundaries. Logistic Regression is a linear model and may not be suitable for non-linear data. Overall, the analysis of the diabetes dataset has shown that SVM is a suitable model for this problem, and it outperforms other commonly used classification models in terms of accuracy. This highlights the importance of choosing the right classification algorithm for a particular problem and dataset to achieve the best possible results.

- **K Fold Cross validation using 10 folds**

K-fold cross-validation is used to estimate the performance of a machine learning model on u nseen data, and to determine if the model is overfitting or underfitting the training data. It is a common technique used to select the best hyperparameters of a model, such as the regularizat ion parameter in logistic regression or the number of hidden layers in a neural network.

Average performance of Logistic Regression is: 76.43
Average performance of K Nearest Neighbour is: 71.35
Average performance of Naive-Bayes is: 74.48
Average performance of SVM is: 76.05
Average performance of Decision Tree is: 68.89
Average performance of Random Forest is: 75.

- SVM and logistic regression are showing highest average accuracy.

- **Conclusion:**

1. SVM shows the highest accuracy.
2. After SVM, Logistic Regression and Random Forest have highest accuracy.
3. SVM shows highest precision, recall and f1-score.
4. Random forest and logistic regression also shows good results.

5. Support Vector machine and Naive Bayes shows the highest ROC scores.
6. Logistic Regression has the highest specificity, KNN and Random Forest show highest Sensitivity.
7. Logistic Regression and Random Forest show the highest average accuracies.

By evaluating all models ,we know that Support Vector Machines and Logistic Regression show best evaluation results.

GitHub: https://github.com/aqtiqanawaz95/diabetes-mega-case-study/blob/main/2146625%20Diabetes%20Notebook.ipynb

Article: https://medium.com/@aatiqa.nawaz1995/introduction-to-support-vector-machine-ml-algorithm-bb754934a5f3