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

**Element 010**

Name: Raghunath Muddu
SID: 2139797

# Introduction:

Diabetes can have a significant impact on an individual's overall health and wellness. we will use various supervised machine learning classification models on a diabetes-related dataset to predict the probability of a person getting Diabetes within next five year. This research will involve implementing multiple methods, such as Logistic Regression, Support Vector Machines, Naive Bayes Classifier, K Nearest Neighbor, Decision Tree and random forest. We will also use of techniques K-fold cross-validation to know which models performs most effectively.

# Dataset description:

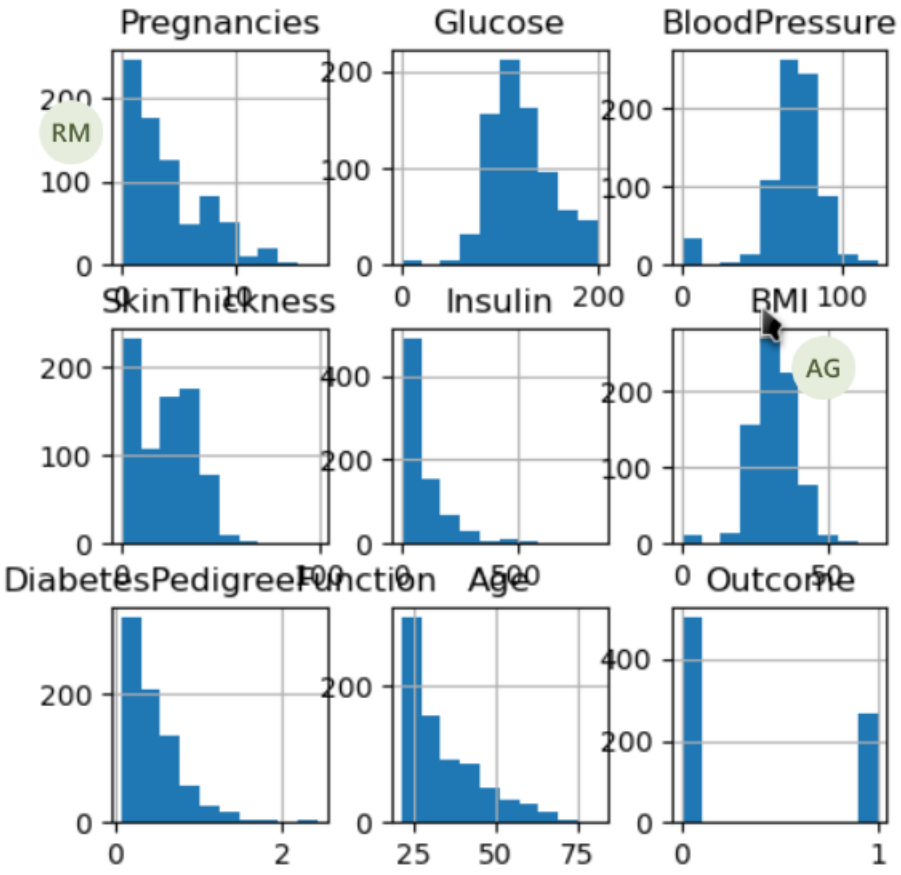
The dataset has 768 instances, with 8 unique and there are multiple indepndent variables and one outcome. The prediction variables are Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, age The dataset includes the following features of following datatypes.

- Pregnancies: No of times Pregnant
- Blood Pressure: Diastolic Blood Pressure (mm hg)
- Insulin: 2-Hour serum insulin (mu U/M1)
- Glucose: Plasma Glucose Concentration 2 hours in Oral Glucose tolerance test
- Skin Thickness: Triceps Skin fold Thickness (mm)
- BMI: Body mass index (weight/kg, height/m^2)
- Diabetes Pedigree Function: indicates the function which scores likelihood of Diabetes based on Family history.
- Outcome: Binary classification (0 or 1)
  500/768 are 0.
  268/768 are 1.

**Summary of dataset**:

| | 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 |

**Distplot** :




From the above table we can see that the min value columns of Pregnancies, Glucose, Blood pressure, Skin thickness, Insulin, BMI is Zero, as the min values can't be Zero, Hence we need to take the mean of every column and replace in the value of Zero

# Data cleaning

We have to make sure there are no Duplicate rows, if there are any Duplicate rows, we need to remove the duplicates from the dataset and Even before and after dropping duplicate, if the dataset still has the same shape which indicates there is no Duplicates in the dataset.

## Check the Null Values

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

There are no Null Values.

## Number of zeroes in dataset:

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

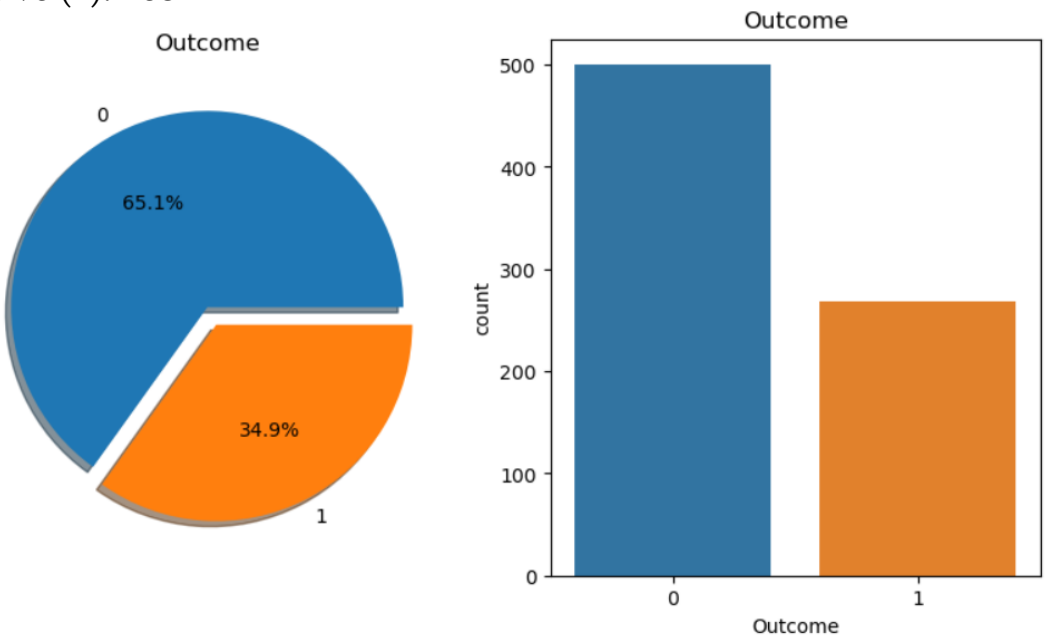
Replace no.of zero values with mean of that columns

| | 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 |

After replacing the mean with the zero's, numerical have changed.

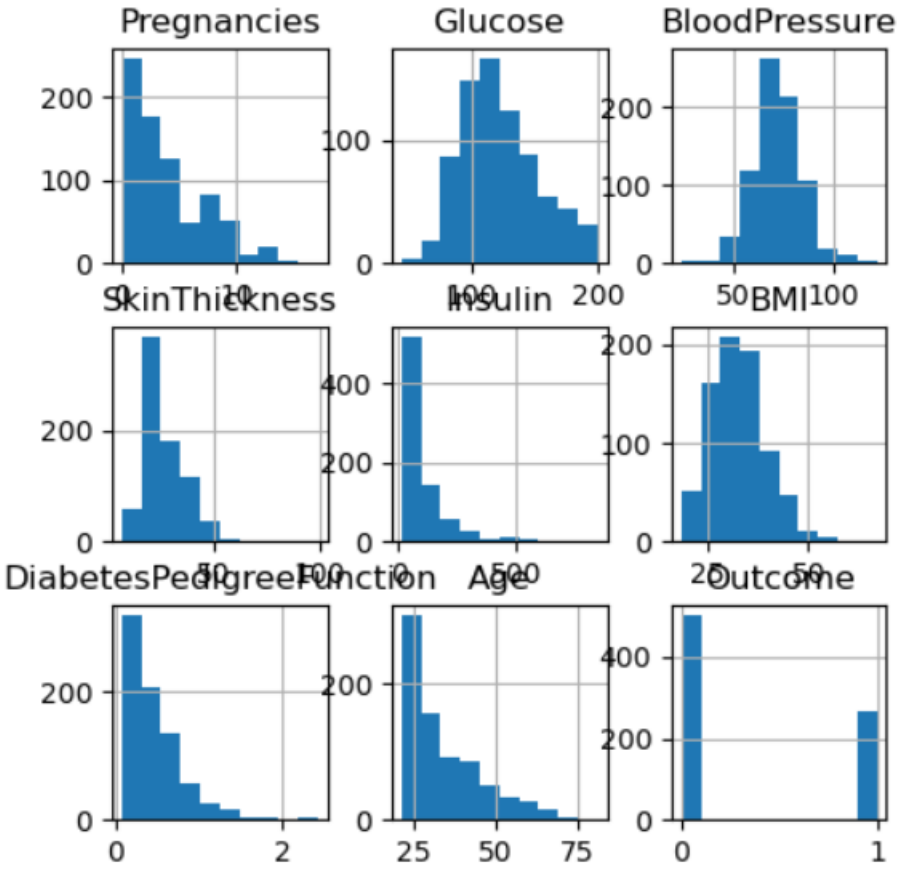
# Data visualisation

Count plot for outcome variable:
Negative (0): 500
Positive (1): 268

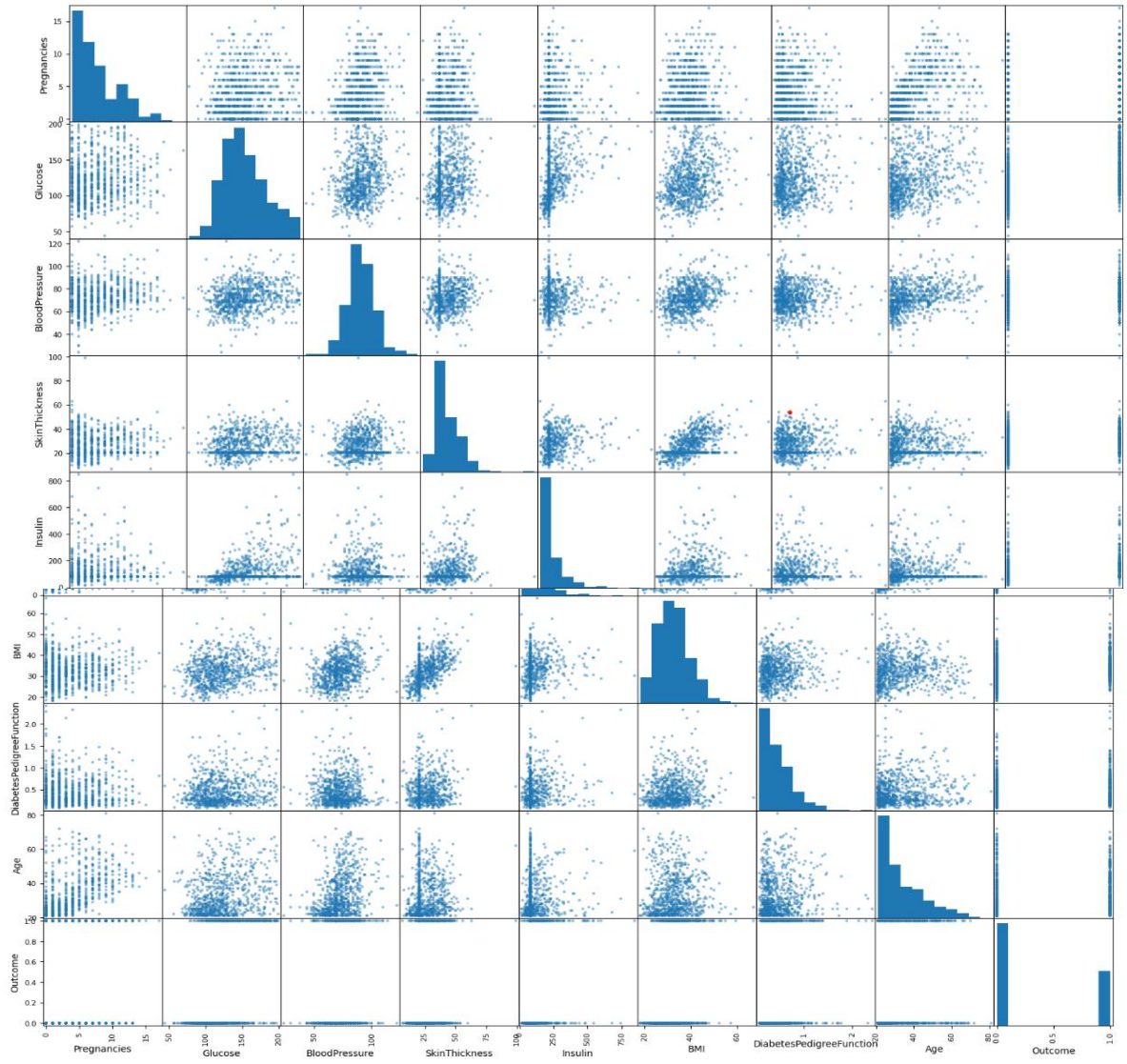



The dataset consists of 768 people, with 268 having diabetes (labelled as positive with a value of 1) and 500 not having diabetes ( labelled as negative with a value of 0). The outcome column represents the presence or absence of diabetes. The countplot indicates that the dataset is imbalanced since the number of patients without diabetes is greater than those with diabetes.

Histograms for all Variables: Histograms are more commonly used Graphs to display the numeric data.



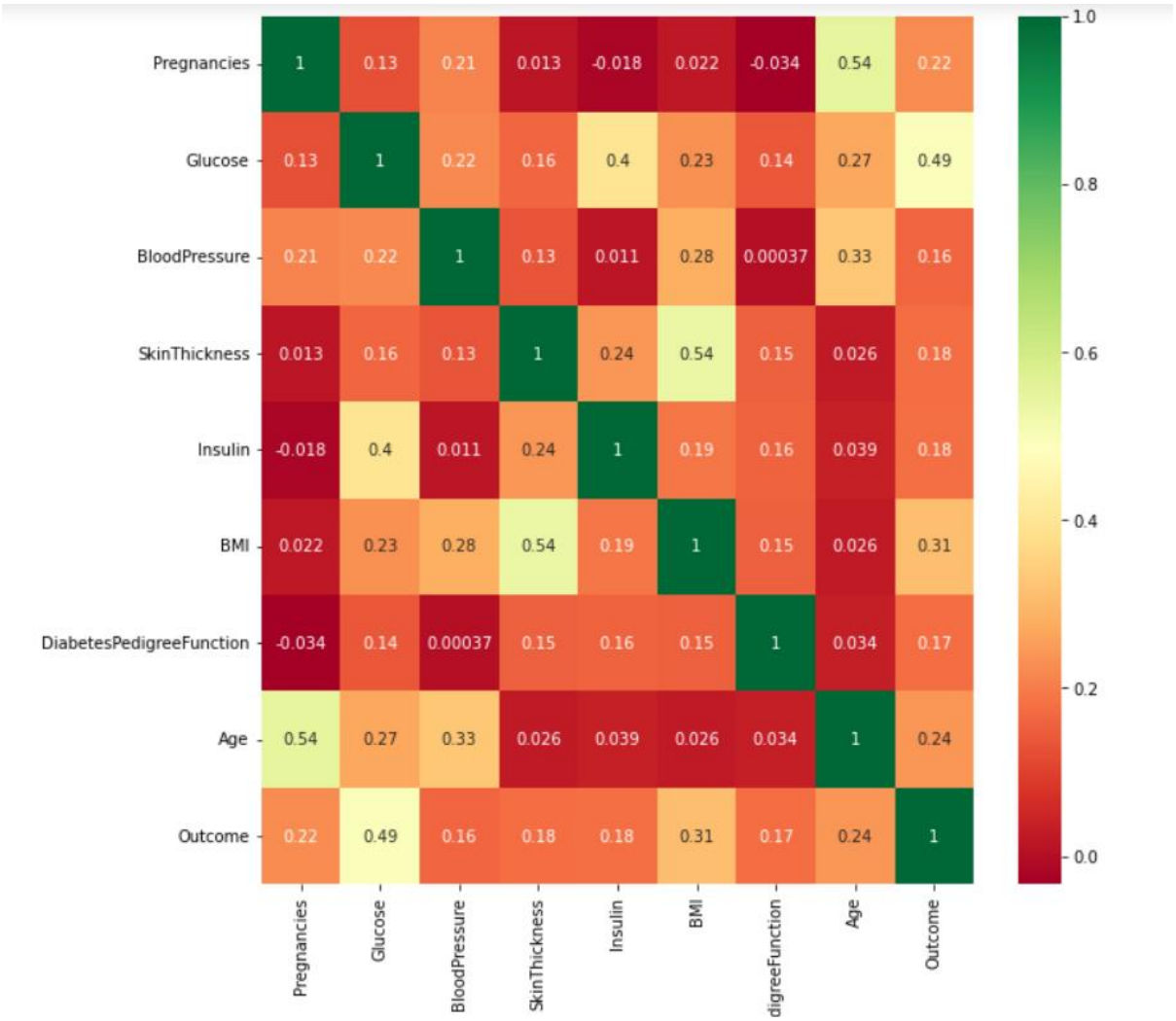
Scatter plot: In scatter plot the graph plots a values of two variable along with two axes and it also makes to understand the correlation between the two variables



Pair plot: va[riables pairplot is the best way to create the scatterplots between all the variables


- For Glucose variable it is observed that chances of getting diabetic increases as Glucose goes above 130.

Correlation matrix：




From the correlation heatmap, we can see that there is a high correlation between the outcome and (pregnancies, Glucose, BMI, Age, Insulin), we can still select this feature to accept input from the user and predict the outcome.

**There is no high correlation, there is medium correlation between**

# Data preprocessing:

- The dataset will be split into X and y for pre-processing using the train-test split function from the sklearn library's model selection function. A test size of 30% will be used.
- X will contain all of the features, such as Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, and Age.
- Y will contain the dependent variable or outcome.
- The data will be scaled using standard scaler before being fed to the models
- There are no categorical columns in the dataset, so there is no need for encoding.

# Model building:

- Different classification models such as logistic regression, K-nearest neighbour classifier, Naïve-Bayes, Support Vector Machine, and decision tree with default parameters will be built using the sklearn library.
- Once the models are trained, the predict function will be used to predict the results. • To further evaluate the models, different evaluation metrics will be employed.

# 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.

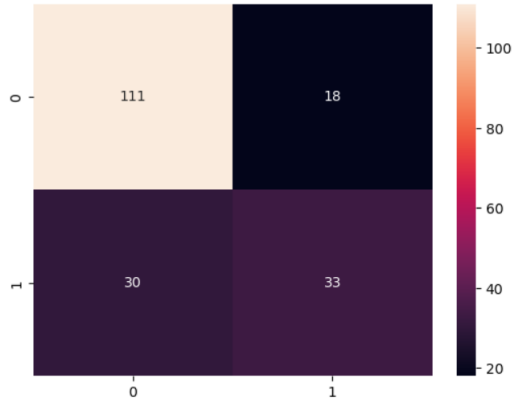
## Accuracy score:

```
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%
```

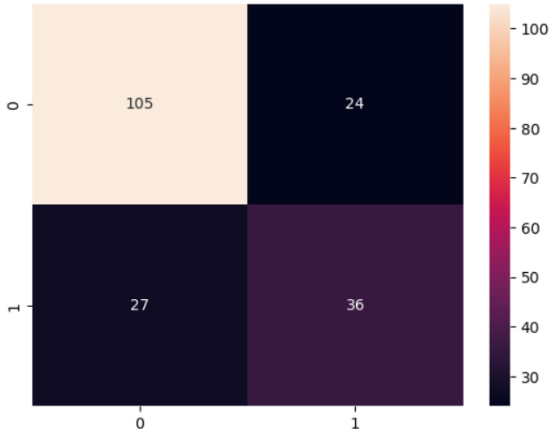
**SVM shows the highest accuracy along with logistic regression.**

# Confusion matrix:

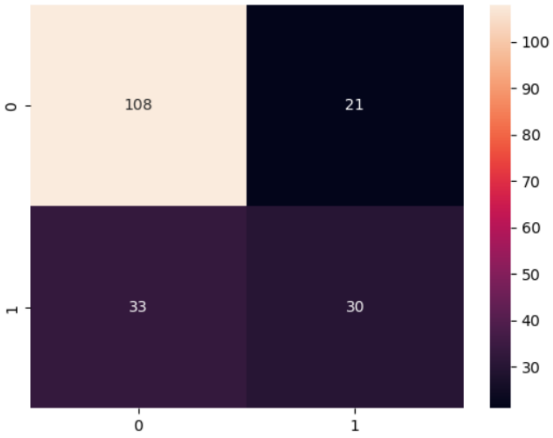
confusion matrix of Logistic Regression:



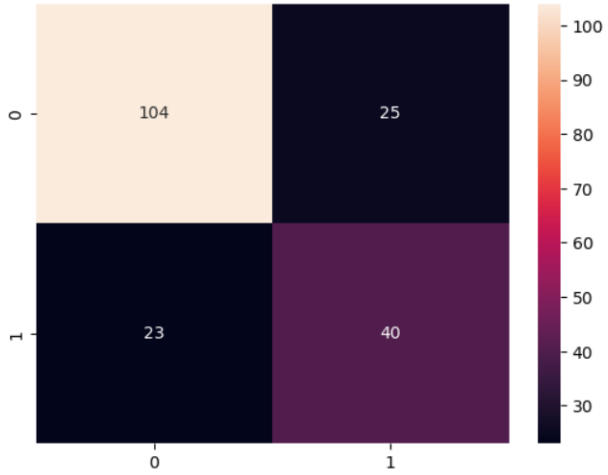

confusion matrix of K nearest Neighbour



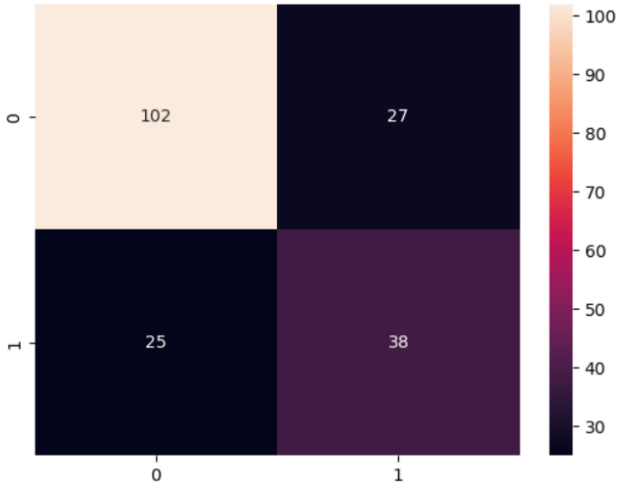

confusion matrix of Support Vector Machine



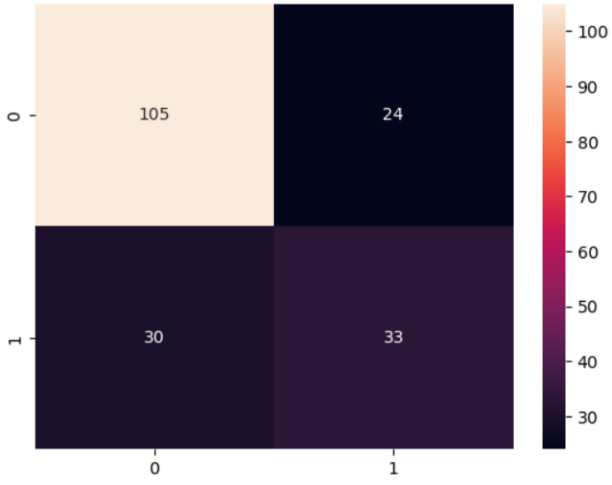
confusion matrix of Naïve Bayes Classifier




confusion matrix of Decision tree




confusion matrix of Random forest



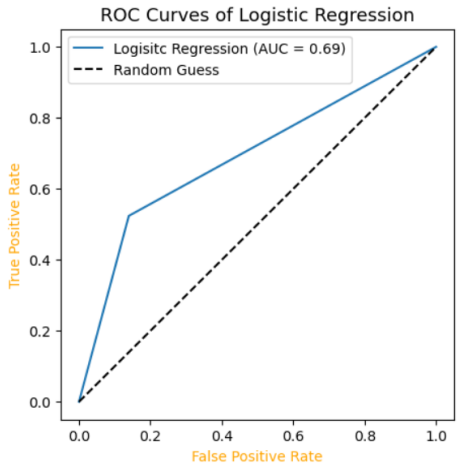
## Classification report:

```
Classsisfication report for Logisitic Regression
:              precision    recall f1-score   support

           0       0.79      0.86      0.82       129
           1       0.65      0.52      0.58        63

    accuracy                           0.75       192
   macro avg       0.72      0.69      0.70       192
weighted avg       0.74      0.75      0.74       192

Classsisfication report for K Nearest Neighbour
:              precision    recall f1-score   support

           0       0.80      0.81      0.80       129
           1       0.60      0.57      0.59        63

    accuracy                           0.73       192
   macro avg       0.70      0.69      0.69       192
weighted avg       0.73      0.73      0.73       192

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

           0       0.77      0.84      0.80       129
           1       0.59      0.48      0.53        63

    accuracy                           0.72       192
   macro avg       0.68      0.66      0.66       192
weighted avg       0.71      0.72      0.71       192

Classsisfication report for Decision Tree
:              precision    recall f1-score   support

           0       0.80      0.79      0.80       129
           1       0.58      0.60      0.59        63

    accuracy                           0.73       192
   macro avg       0.69      0.70      0.70       192
weighted avg       0.73      0.73      0.73       192

Classsisfication report for Naive-Bayes
:              precision    recall f1-score   support

           0       0.82      0.81      0.81       129
           1       0.62      0.63      0.62        63

    accuracy                           0.75       192
   macro avg       0.72      0.72      0.72       192
weighted avg       0.75      0.75      0.75       192
```

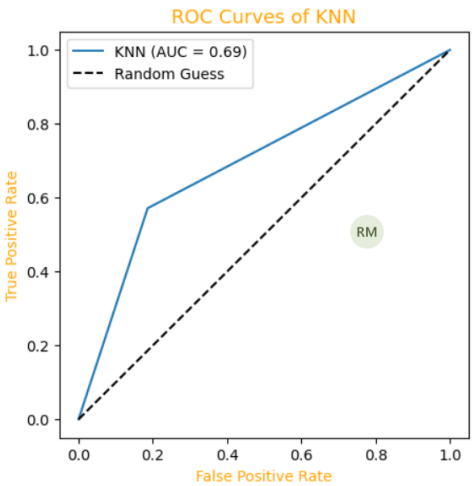
```
Classsisfication report for Random Forest
:              precision    recall  f1-score   support

           0       0.78      0.81      0.80       129
           1       0.58      0.52      0.55        63

    accuracy                           0.72       192
   macro avg       0.68      0.67      0.67       192
weighted avg       0.71      0.72      0.71       192
```

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

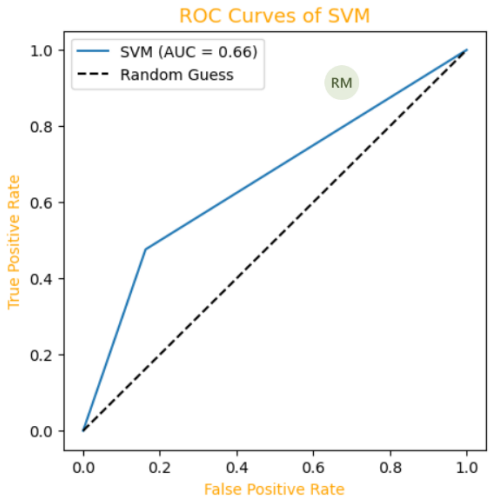
ROC curves:



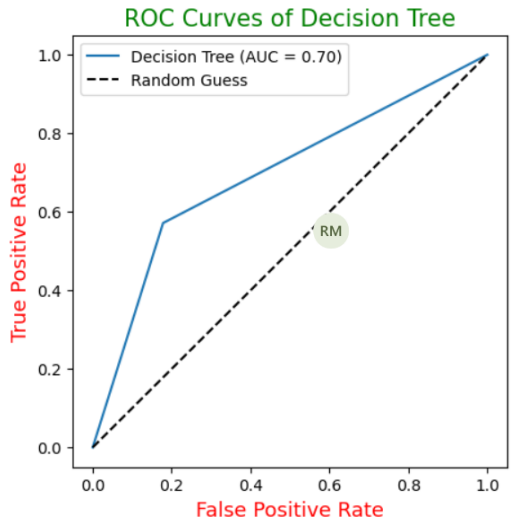

ROC Score: 0.69



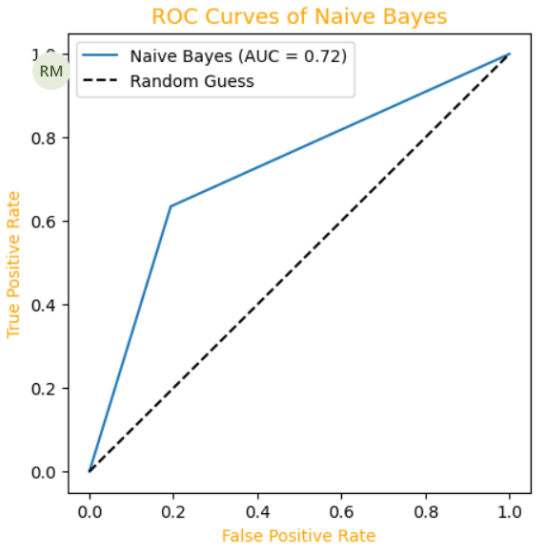

ROC Score: 0.69



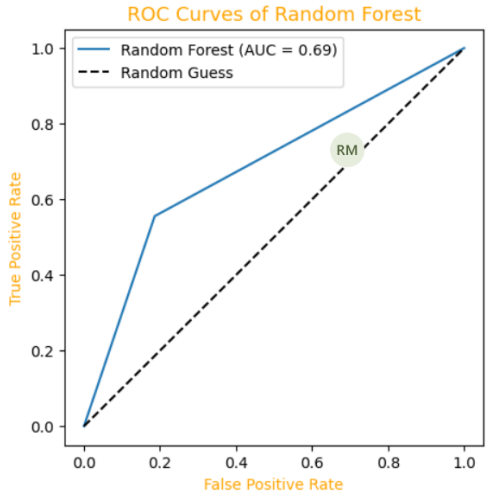

ROC Score: 0.66




ROC Score: 0.70




ROC Score: 0.72




ROC Score: 0.69

Naïve Bayes getting the highest ROC score.

## 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
```

Logistic regression and SVM got highest specificity.

KNN and SVM got highest Sensitivity.

## K Fold Cross validation using 10 folds:

```
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: 75.78
```

# Conclusion:

1. There were no NULL values present in the dataset however there ever many Zero's present, so we considered them as NULL values and imputed them
2. SVM shows the highest accuracy.
3. After SVM random forest and Logistic Regression have highest accuracy.
4. SVM shows highest precision, recall and f1-score.
5. Support Vector machine and KNN shows the highest ROC scores.
6. Logistic Regression and SVM has the highest specificity, KNN and Random Forest shows highest Sensitivity.
7. Logistic Regression, SVM and random forest shows the highest average accuracies while doing K Fold cross validation.

**From overall evaluation metrices Support Vector machines and Logistic Regression shows best evaluation results.**