PRML MAJOR PROJECT

PREPARED BY

Abhaymani Singh(B21EE001)

Garvit Gangwal(B21EE019)

Gaurav Naval(B21EE020)

1. Project Overview

We are given the **Diabetes Dataset** and in this Major Project, we analyze the dataset using different concepts and implement an end-to-end machine learning pipeline for the task given in the project.

2. Preprocessing

We start with the preprocessing of the **Diabetes Dataset** which is the first and foremost step while implementing any machine learning pipeline.

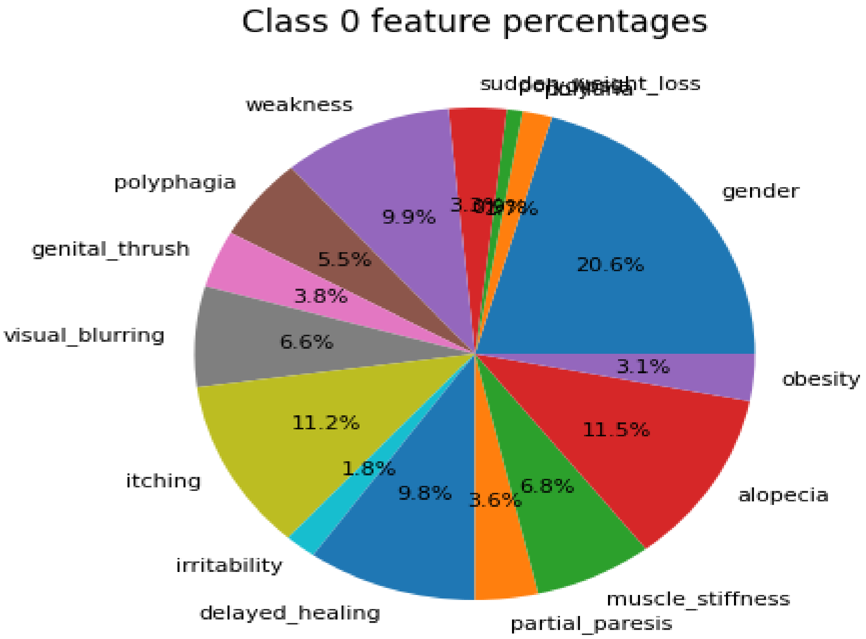
1) We checked for the **Null Values** and found that there were None of them.
2) All the features of the data type "**int64**" except gender which is of data type "**object**".
3) Hence we **label encode** the gender feature.

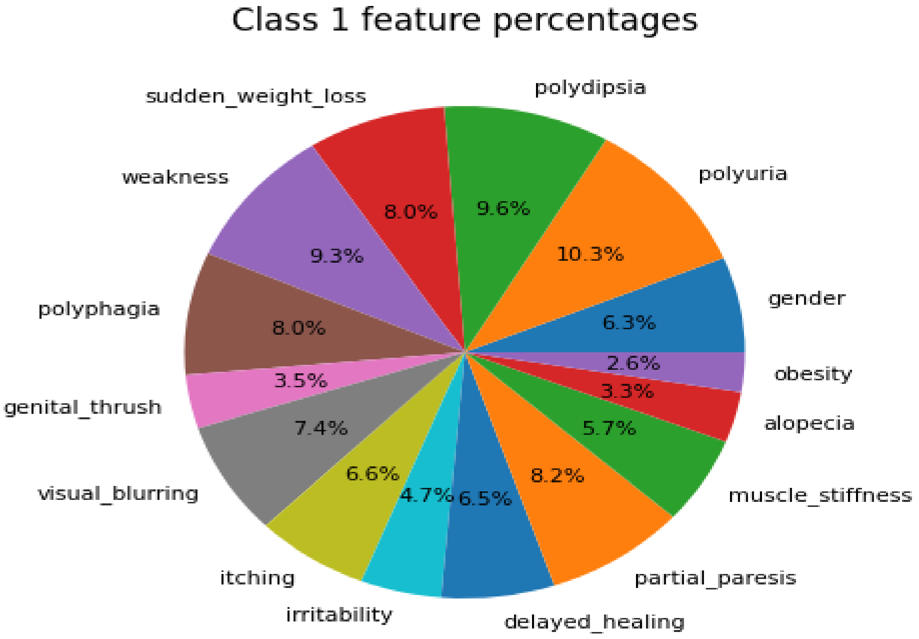
3. Visualisation(Dataset)

The goal of visualization is to make a complex dataset more understandable.

1) **A histogram** with all the numerical features is plotted in which the **class label** is present on the x-axis and the number of people on the y axis except for age in which age number is present on the X axis.
2) **Box plots** for all the numerical features are plotted.
3) Correlation matrix is plotted and the **correlation matrix heatmap** is plotted.
4) **Pair plots** and **Bar plots** of all the features are plotted too.
5) **Pie charts** for each **class label** are plotted indicating how much each feature contributes to that particular class.
6) Data set is splitted into **training** and **testing** dataset using tts.

Below pie charts shows the contribution of each feature to a particular class label :





4. Machine Learning Algorithms

Through this Minor project we have used various ML algorithm for analysis which include the following:

1) **Decision Tree Classifier(DTC)** : A decision tree classifier is a machine learning algorithm that can be used for both **classification** and **regression** tasks. It works by recursively splitting the data into subsets based on the most informative feature at each step, until the data is fully partitioned into homogenous groups.
2) **Gaussian Naive Bayes(GNB)** : Gaussian Naive Bayes (GNB) is a probabilistic classification algorithm that makes predictions by using **Bayes' theorem**, which calculates the probability of a hypothesis given the evidence. It is based on the assumption that the features are **independent** and **normally distributed**.
3) **K Nearest Neighbour(KNN)** : The k-nearest neighbour (KNN) algorithm is a type of instance-based or lazy learning algorithm used for **classification** and **regression** tasks. The algorithm works by finding the k closest training examples to the test instance in the feature space and using their labels or values to predict the label or value of the test instance.
4) **Random Forest Classifier(RFC)** : Random forest classifier is a popular **ensemble learning** algorithm for classification problems. It is composed of multiple decision trees, each trained on a random subset of the training data and a random subset of the features. The final prediction is made by aggregating the predictions of all the trees.
5) **Support Vector Machine(SVM)**: SVM works by finding the **optimal hyperplane** that maximally separates the data into two classes, or by approximating a function that predicts a **continuous output.**

6) **Logistic Regression(LR) :** Logistic Regression is a supervised machine learning algorithm used for **binary classification** problems, where the target variable takes only two values, typically 0 or 1. The algorithm models the probability of the target variable being in one of the two classes as a function of the predictor variables.

7) **Feature Selection(FS)** : Feature selection is a process of selecting a subset of relevant features (predictor variables) from a larger set of features to be used in building a predictive model. The goal of feature selection is to improve the model's performance, reduce its complexity, and enhance its **interpretability**.

8) **Multilayer Perceptron(MLP) :** A multilayer perceptron (MLP) is a type of artificial neural network that consists of many layers of linked nodes, each with its own non-linear activation function. MLPs are trained using a supervised learning technique, such as backpropagation, to modify the weights of the node connections and minimise the gap between the network's expected and intended output.

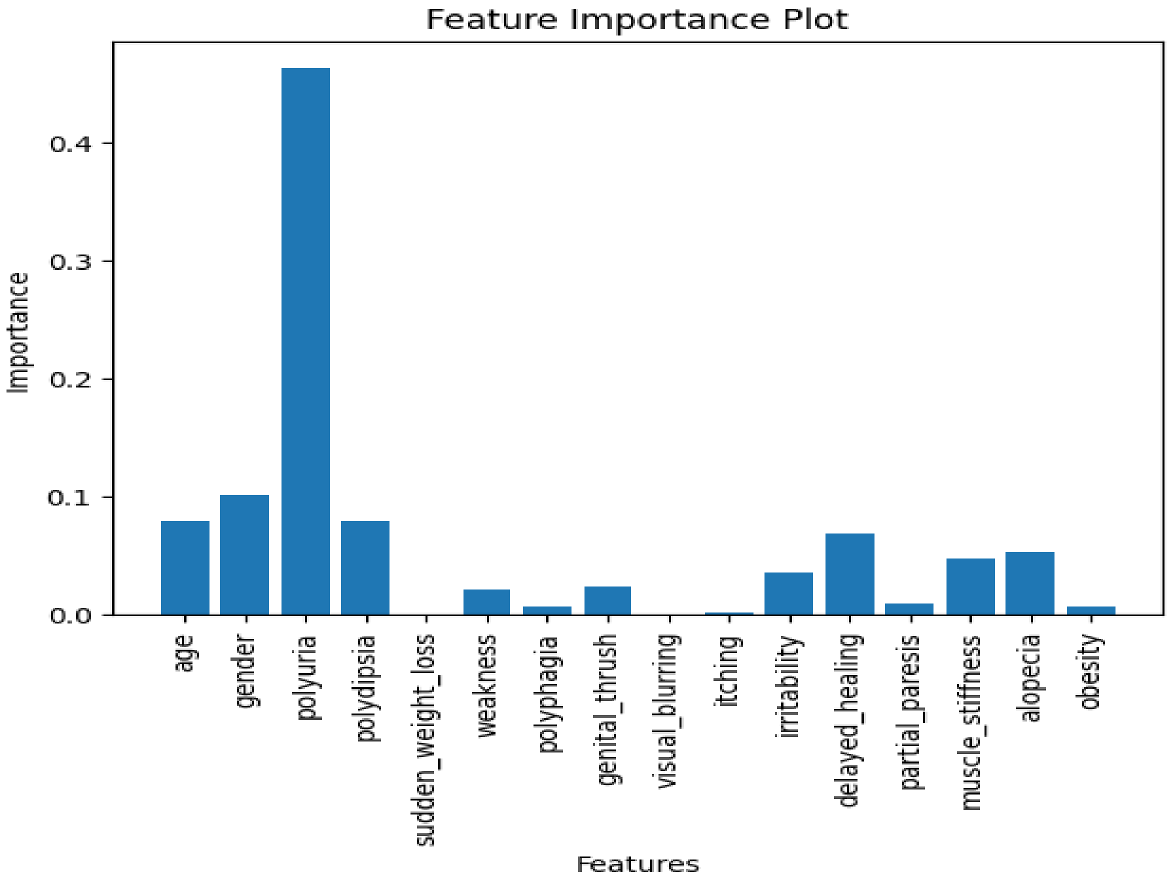
5. Application of ML Algorithms

1) Decision Tree Classifier(DTC)

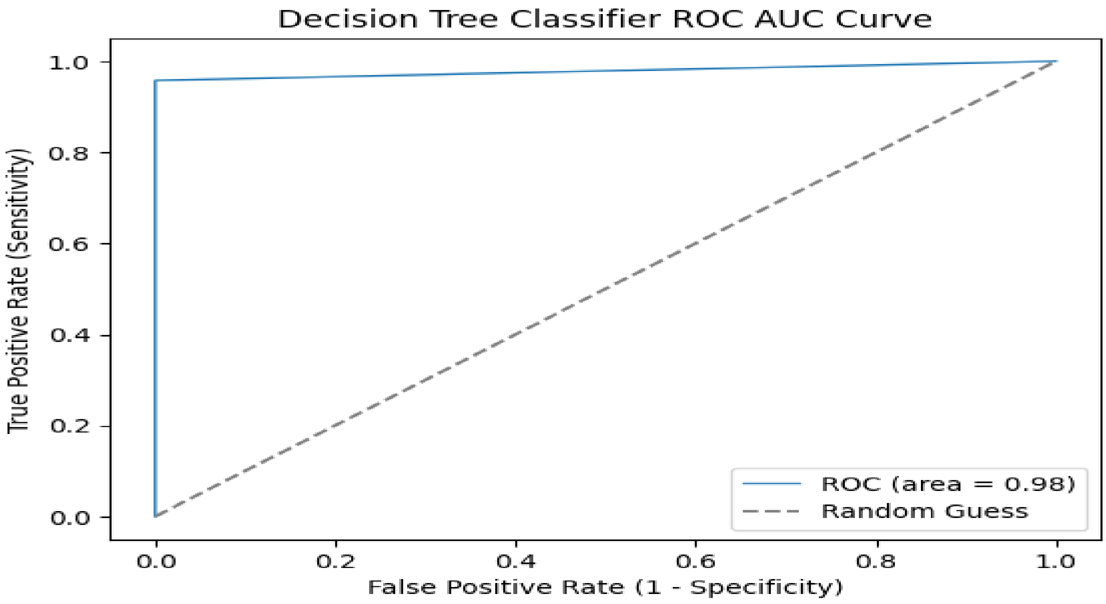
- We optimise the **hyperparameters** max depth and min_sample_splits of the decision tree classifier by performing grid search.
- The optimal hyperparameters are as follows :
    - **Max depth** : 8
    - **Min_sample_splits** : 2

- The performance of DTC is evaluated on the basis of optimal hyperparameters using metrics like accuracy score , f1 score, recall and precision.
- Following are the values of the performing metrics :
  - **Accuracy score**: 95.1923076923077
  - **F1 Score** : 96.35036496350364
  - **Precision** : 100.0
  - **Recall** : 92.95774647887323
- Feature importance plot and ROC-AUC Curve is plotted.
  - **Area** : 0.98
- We obtain the confusion matrix too.
- The plotted curves are drawn below :

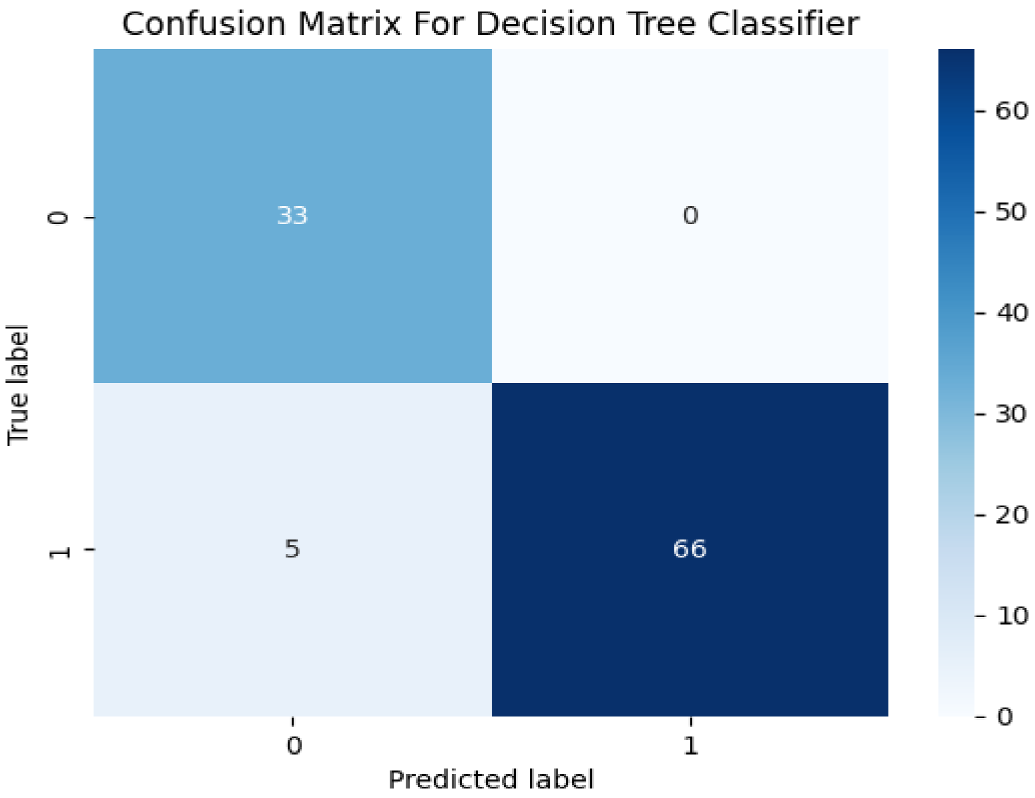
Feature Importance curve:



ROC-AUC CURVE :



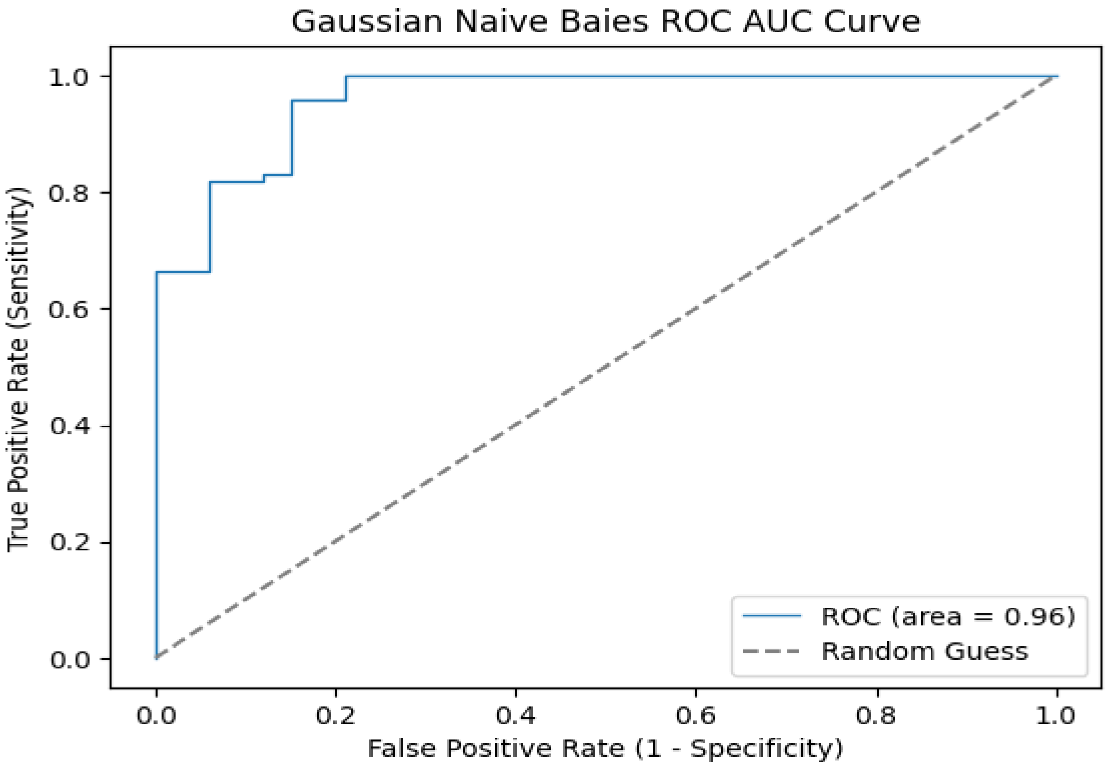
CONFUSION MATRIX :



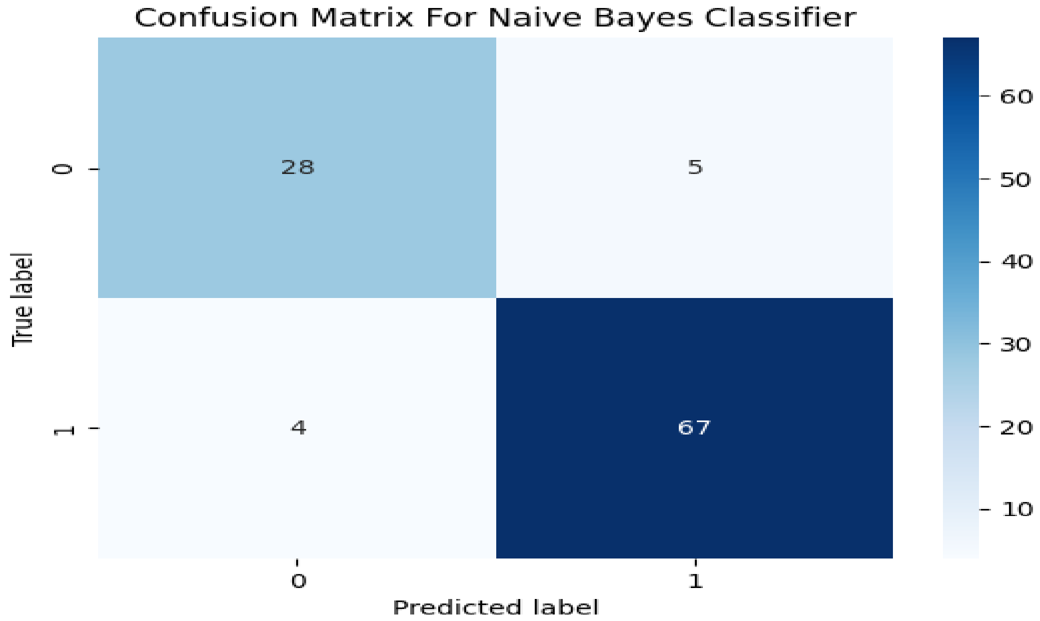
2) Naive Bayes Classifier(GNB)

- We initialise a GNB classifier and fit it into the training data.
- The performance is then evaluated on the basis of various performance metrics like accuracy, precision,recall etc.
- The values are as follows :
  - **Accuracy score** : 91.34615384615384
  - **F1 Score** : 93.70629370629372
  - **Precision** : 93.05555555555556
  - **Recall** : 94.36619718309859
- We obtain the ROC-AUC curve :
  - **Area** = 0.96
- The confusion matrix is also obtained.

ROC-AUC Curve :



Confusion Matrix :

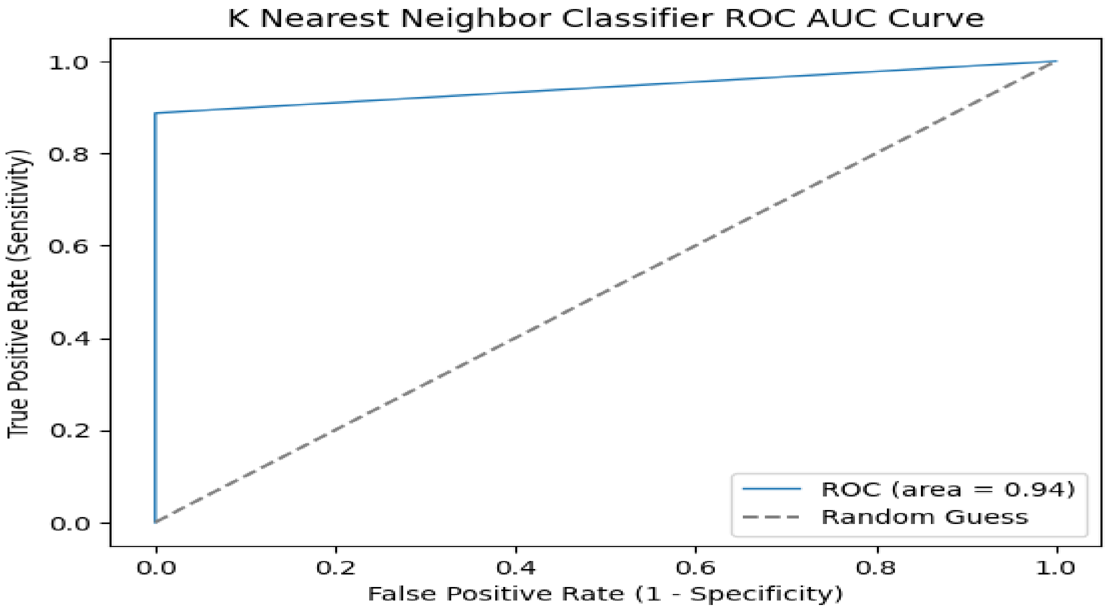


3)K Nearest Neighbour(KNN)

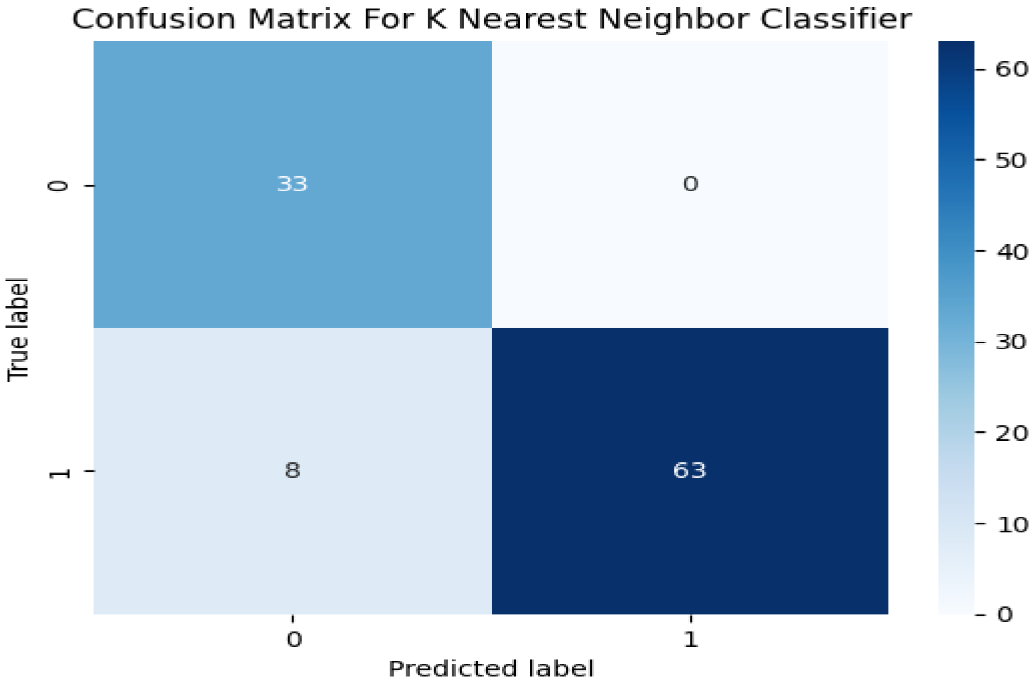
- The optimal value of the parameters of KNN is obtained using Grid search.
- The optimal values are as follows :
  - **Algorithm** : brute
  - **Weights** : uniform
  - **N_neighbours** : 1
  - **P** : 1
- Performance metrics are calculated based on the above optimal values :
  - **Accuracy score** : 91.34615384615384
  - **F1 Score** : 93.70629370629372
  - **Precision** : 93.05555555555556
  - **Recall**   : 94.36619718309859
- ROC-AUC Curve is plotted :
  - **Area** = 0.94

- Confusion matrix is also obtained.

ROC AUC CURVE :



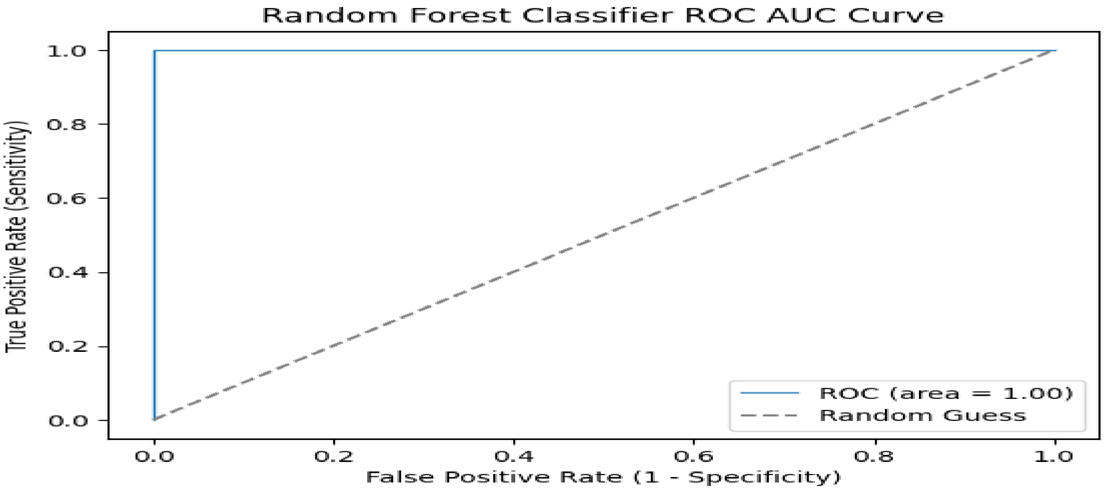
The confusion matrix :



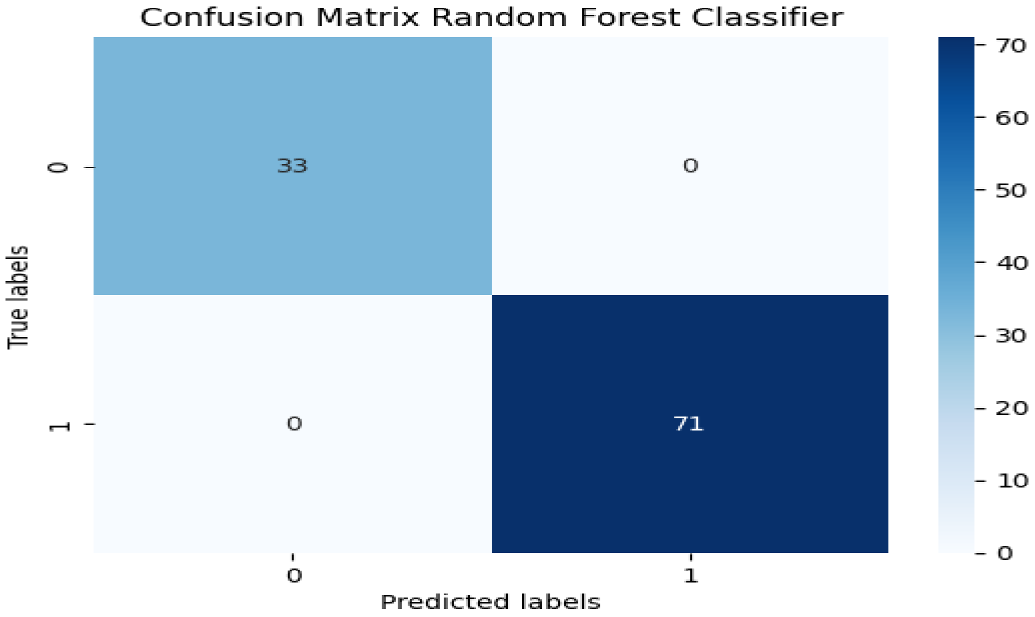
4)Random Forest Classifier

- We perform **hyperparameter tuning** on random forest classifiers using grid search.
- The hyperparameters used are as follows :
    - **Max Depth**
    - **Min_Sample_Split**
    - **N_estimators**
- Different values of these hyperparameters are varied and a best set of values is evaluated using GridSearchCV.
- The best hyperparameter values are as follows :
    - **Max Depth** : 10
    - **Min_Sample_Split** : 2
    - **N_estimators** : 50
- The value of the evaluation metrics are as follows :
    - **Accuracy score** : 100.0
    - **F1 Score** : 100.0
    - **Precision** : 100.0
    - **Recall** : 100.0
- The ROC curve and the confusion matrix is also plotted.

ROC CURVE :



CONFUSION MATRIX :



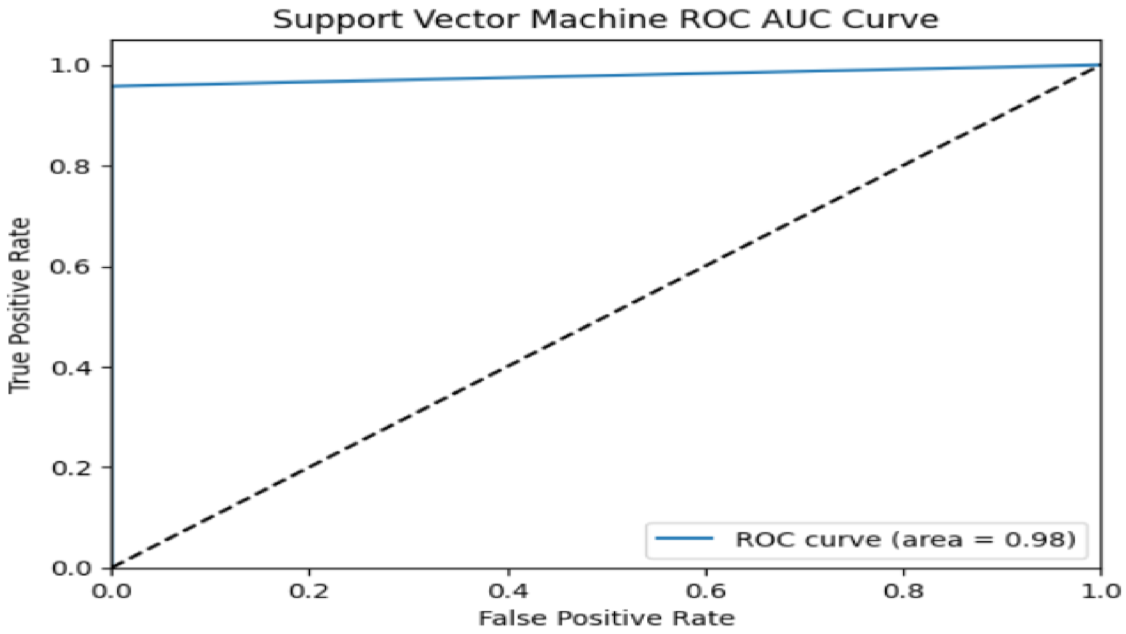
5)Support Vector Machine(SVM)

- We perform hyperparameter tuning for a Support Vector Machine (SVM) model using GridSearchCV.
- We consider different values of hyperparameters ‘c’, ‘gamma’ and ‘karnel’.
  - **C : Regularisation parameter**
  - **Gamma : Kernel coefficient**
  - **Kernel : Type of kernel to be used in SVM model**
- A new SVM model is then created using the best hyperparameters found and fitted to the training data
- The best hyperparameters are as follows :
  - **C** : 100
  - **Gamma** : 0.1
  - **Kernel** : ‘rbf’
- Evaluation metrics are calculated which are as follows :
  - **Accuracy score** : 97.11538461538461

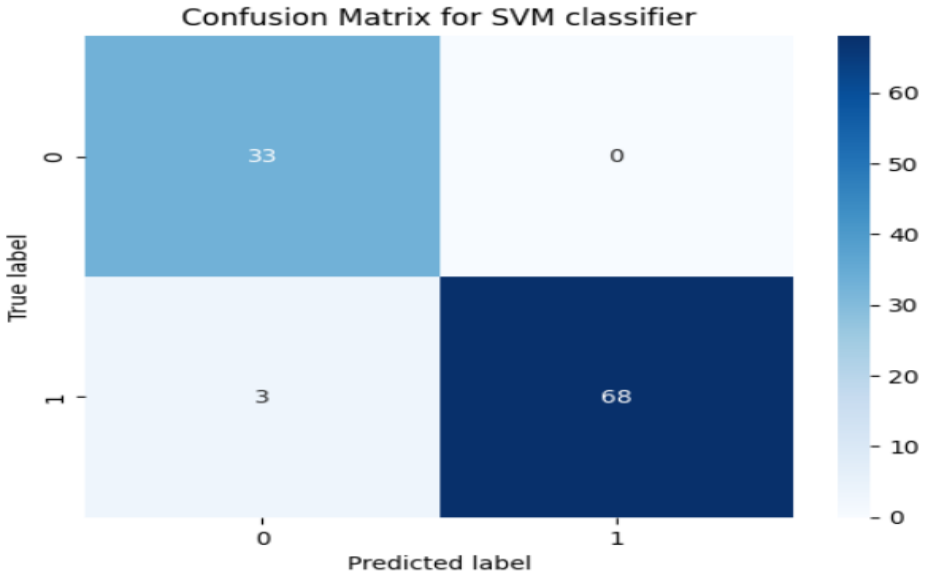
- **F1 Score** : 97.84172661870502
- **Precision** : 100.0
- **Recall** : 95.77464788732394

- The ROC curve and the confusion matrix is also plotted.

ROC CURVE :



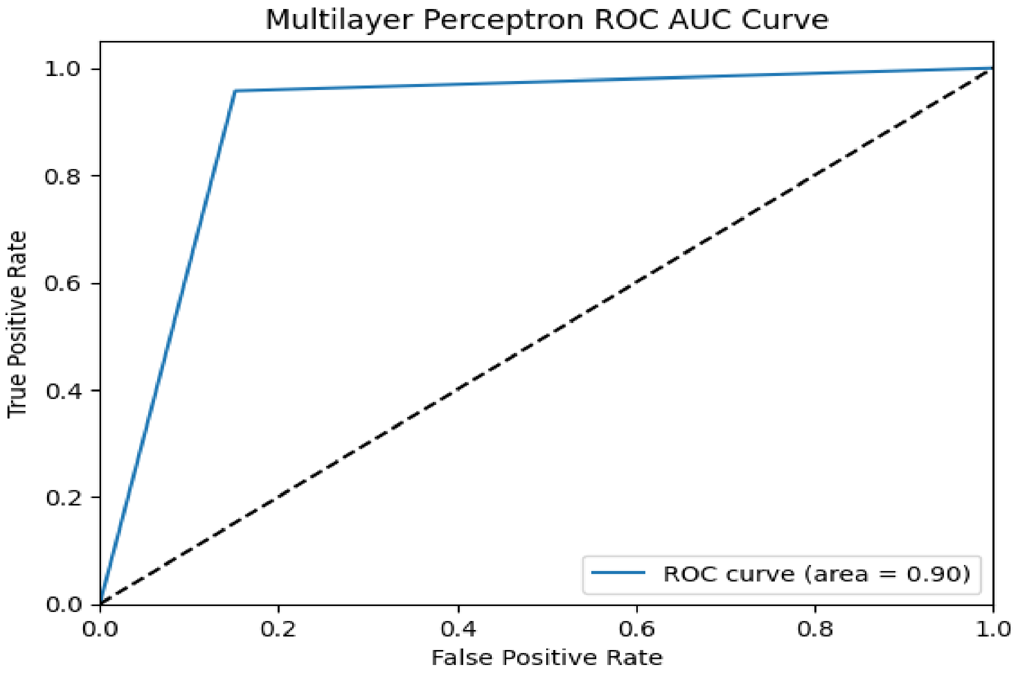
CONFUSION MATRIX :



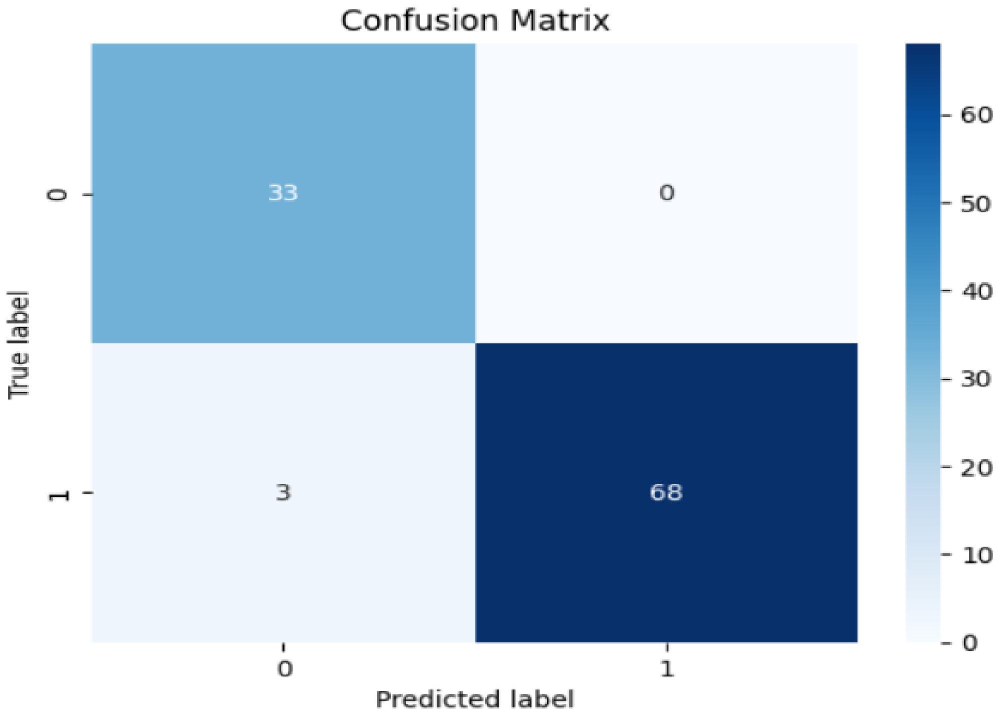
6)Multi-Layer Perceptron

- We perform hyperparameter tuning for a multi layer perceptron using grid search.
- The hyperparameters considered are as follows :
  - **Hidden_layer_sizes**
  - **Activation**
  - **Solver**
  - **Alpha**
  - **Learning rate**
- The best set values calculated using grid search are as follows :
  - **Hidden_layer_sizes** : (100,100)
  - **Activation** : ‘relu’
  - **Solver** : ‘adam’
  - **Alpha** : 0.001
  - **Learning rate** : ‘constant’
- The value of the evaluation metrics are as follows :
  - **Accuracy score** : 92.3076923076923
  - **F1 Score** : 94.44444444444444
  - **Precision** : 93.15068493150685
  - **Recall** : 95.77464788732394
- The ROC-AUC curve and the confusion matrix is plotted too for reference.

ROC CURVE :



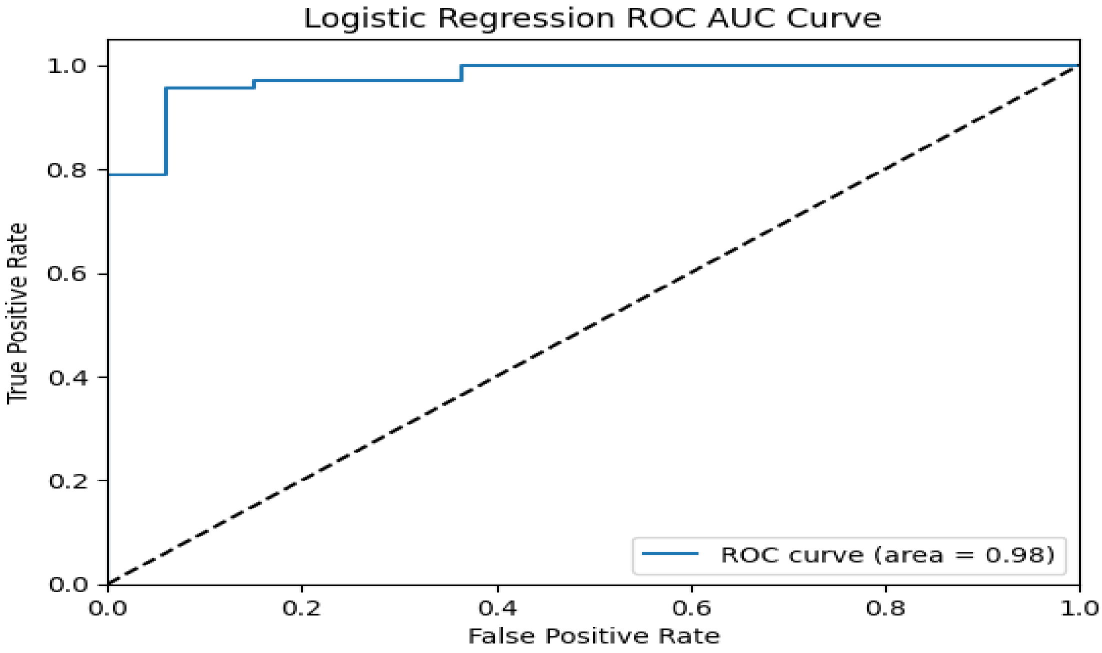
CONFUSION MATRIX :



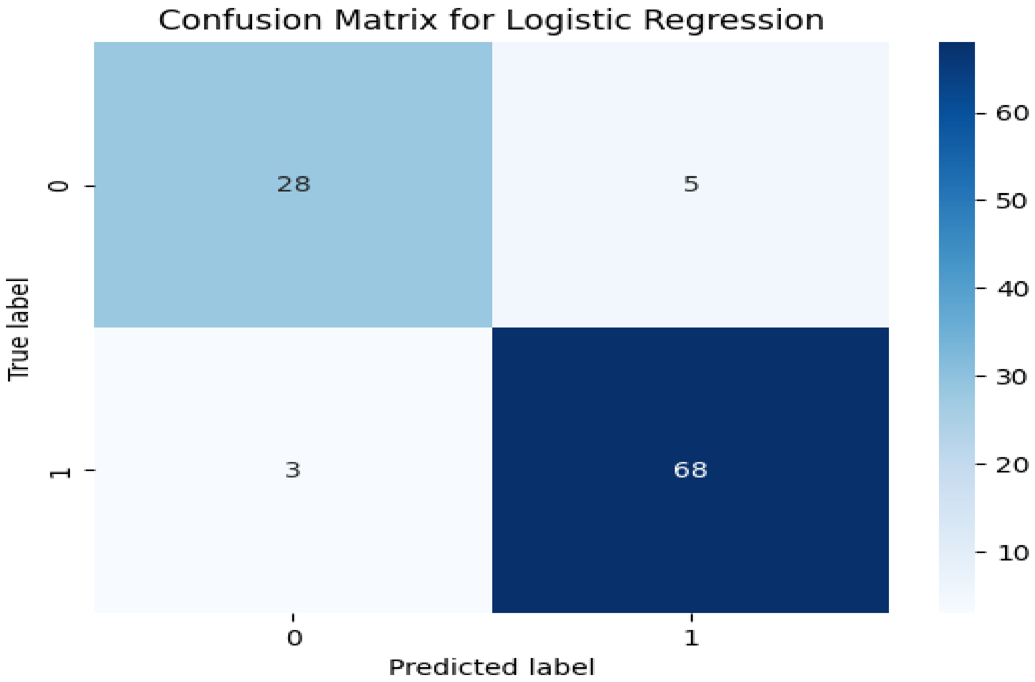
7) Logistic Regression

- We perform hyperparameter tuning for a logistic regression model using grid search.
- The hyperparameters are varied and the best set of values is calculated :
  - **Max_iter** : 100
  - **Penalty** : 12
  - **Solver** : ‘lbfgs’
  - **C** : 10
- The metrics are calculated and there values are as follows :
  - **Accuracy score** : 92.3076923076923
  - **F1 Score** : 94.44444444444444
  - **Precision** : 93.15068493150685
  - **Recall** : 95.77464788732394
- The ROC-AUC curve and the confusion matrix is also plotted for reference.

ROC-AUC CURVE :



CONFUSION MATRIX :



Feature Selection :

At last we performed feature selection using the SequentialFeatureSelector() and we used the Random Forest Classifier for this (because earlier we got the highest accuracy for it). We have taken the number of features to select to be 12 and calculated the evaluation metrics for Decision Tree classifier and Random Forest classifier.

For **decision tree** :

Accuracy score : 97.115384

F1 score : 97.841726

Precision score : 100

Recall score : 95.774647

For r**andom forest** :

Accuracy score : 99.038461

F1 score : 99.290780

Precision score : 100

Recall score : 98.591549

As we can see that the evaluation metrics increased for decision tree classifier which implies that selecting the best 12 features helps in reducing overfitting and noise in the data. On the other hand we can see that for the random forest classifier evaluation metrics decreased (initially all metrics were 100%) which implies that our model works perfectly for all the features. If we reduce some features, it will lead to a loss of information in case of random forest classifier.

We also checked the accuracies for number of features equal to 5 and 10 and we observed that they give less accuracy.

Contributions :

The project was completed through the collaborative efforts of the team members. **Abhaymani Singh** contributed significantly to the report writing process and implemented RFC. **Garvit Gangwal** and **Gaurav Naval** were primarily responsible for the coding aspects of the project and implementation of the machine learning models.