Credit Card Fraud Detection

Name: Supriya K

Problem Statement:

Credit card fraud is a major concern in today's financial world. It involves the unauthorized use of a card or card information for fraudulent transactions, leading to financial losses for both consumers and businesses. The goal of this project is to develop a machine learning model that accurately detects fraudulent transactions in real-time.

In this context, I analyzed a dataset containing credit card transactions made by European cardholders over two days. This dataset consists of 284,807 transactions, with only 492 marked as fraudulent, making the dataset highly imbalanced.

Exploratory Data Analysis (EDA) and Preprocessing:

- Checked for missing values and I found none.
- Identified and removed 1,081 duplicate rows.
- Scaled the 'Amount' column using StandardScaler for better model performance.
- Converted the 'Time' column to the correct datetime format.
- Visualized the distribution of the target variable (fraudulent vs. non-fraudulent transactions) using bar and pie plots.
- Dropped the 'Time' column as it was unnecessary for model training.

Dealing with Imbalanced Data:

I applied **SMOTE** to generate synthetic samples of fraudulent transactions, balancing the dataset and improving model performance.

SMOTE (Synthetic Minority Over-sampling Technique) is a over- sampling technique which create synthetic examples to oversample the minority class. In this dataset fraud transaction are very less hence it create synthetic examples to balance with normal transactions.

Model Selection and Training:

Several machine learning models were trained to identify fraudulent transactions. I used an 80:20 split for training and testing the data. The models evaluated are:

- **Logistic Regression:** It is used for binary classification where the aim is to predict the discrete outcome 0 or 1. In my logistic regression model the accuracy, precision, recall, F1 scores are as follows:
    - Accuracy: 94.4%
    - Precision: 94.5%
    - Recall: 94.4 %
    - F1 Score: 94.4%

- **Decision Tree:** It is intuitive and versatile machine learning model which breaks down data by making decisions based on feature values and create a tree like structure. In my decision tree model the accuracy, precision, recall, F1 scores are as follows:
    - Accuracy: 99.8%
    - Precision: 99.8%
    - Recall: 99.8%
    - F1 Score: 99.8%

- **Random Forest:** It is an ensemble learning method, which builds multiple decision trees and merges their result to improve the predictive accuracy and control overfitting. In my random forest model, the accuracy, precision, recall, F1 score as follows:
    - Accuracy: 99.9%
    - Precision: 99.9%
    - Recall: 99.9%
    - F1 Score: 99.9%

- **XGBoost with GridSearchCV (Best Model):** Extreme Gradient Boosting is an advanced and powerful implementation of gradient boosting algorithms designed for speed and performance. XGBoost builds an ensemble of decision trees sequentially, where each new tree attempts to correct the errors of the previous trees. In my XGBoost model the accuracy, precision, Recall, F1 scores are as follows:
    - Accuracy: 99.97%
    - Precision: 99.97%
    - Recall: 99.97%
    - F1 Score: 99.97%

XGBoost performed the best, achieving high accuracy and F1 scores.

Model Evaluation:

For each model, confusion matrices and ROC curves were generated to assess performance. Among the evaluated models, **XGBoost** emerged as the most effective for detecting fraudulent transactions.

Hyperparameter Tuning:

Hyperparameter tuning is the process of optimizing the parameters that control the learning process of a machine learning model, as opposed to model parameters that are learned from the data during training. Hyperparameters are crucial for the performance and effectiveness of models like XG Boost. The goal is to find the best set of hyperparameters to maximize the model's predictive performance on unseen data.

For the XGBoost model, I used Grid Search hyperparameter tuning method followed by the classification report.

For the logistic Regression model, I used Random search hyperparameter tuning method followed by the classification report.

Cross-Validation:

To further validate the model's performance, I used **stratified K-Fold cross-validation**, which provided a robust evaluation of the XGBoost model across different folds of the dataset.

Model Deployment:

The final XGBoost model was saved using joblib as **Credit_card_fraud_detection_model**. This model can be loaded later for predicting whether new transactions are fraudulent.

Model Prediction and Testing:

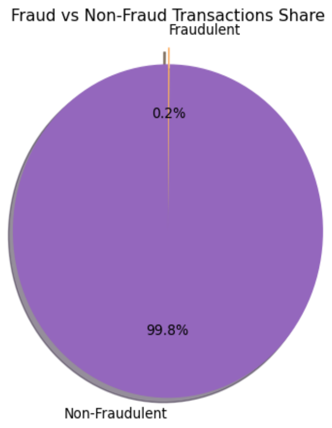
For checking the model, I loaded the credit_card_fraud_detection_model and checked the model's prediction on the actual data, which was given where, 0 means Non-Fraudulent Transaction and 1 means Fraudulent Transaction. **I found that model is a Non-Fraudulent Transaction**.

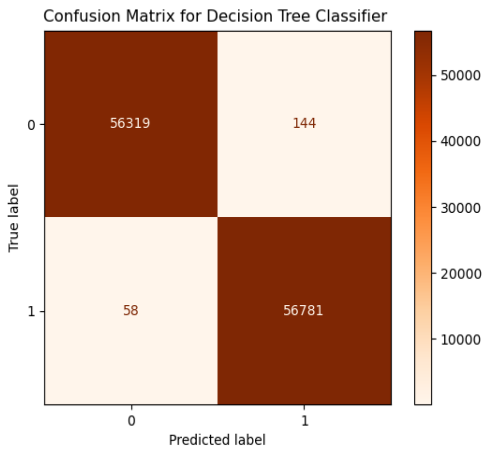
Cost-Benefit Analysis:

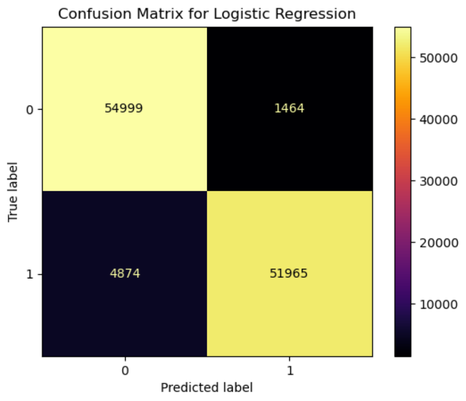
For credit card companies, a high recall is crucial, especially for larger transactions. Missing a fraudulent transaction could result in significant financial losses. The XGBoost model, with its high recall, is well-suited for detecting high-value fraud transactions.

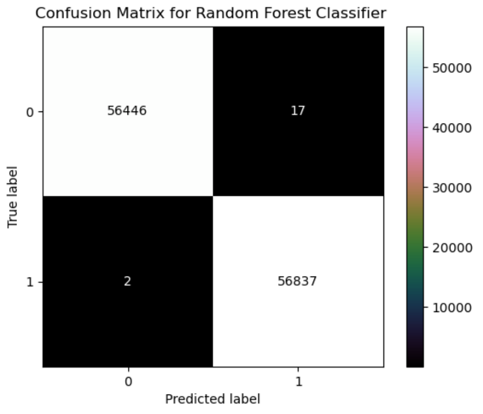
Visualizations Generated:

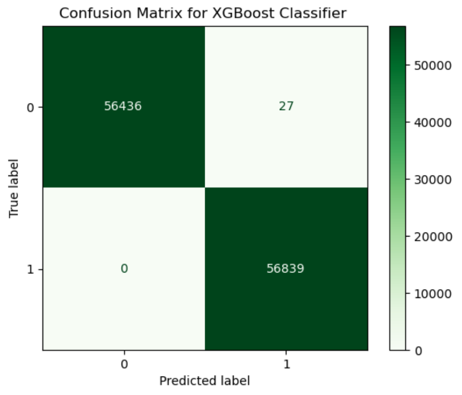
I have included the images of the visualizations such as confusion matrices, ROC curves, correlation matrix, pair-plot generated for my model for a better documentation.

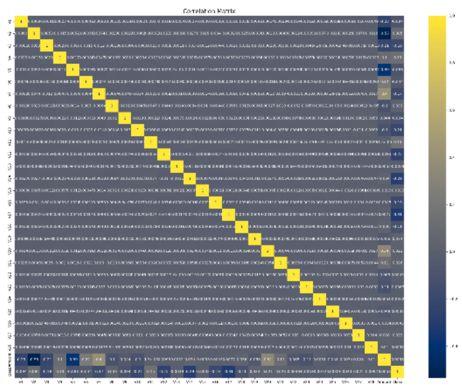


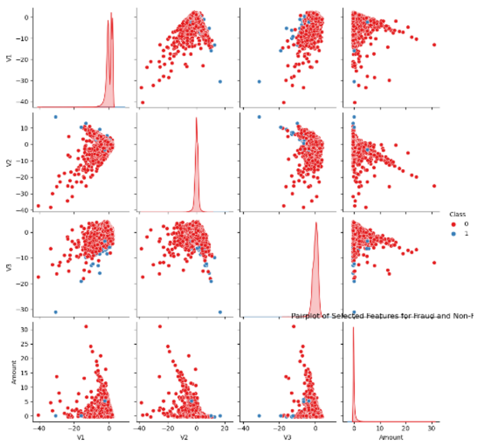


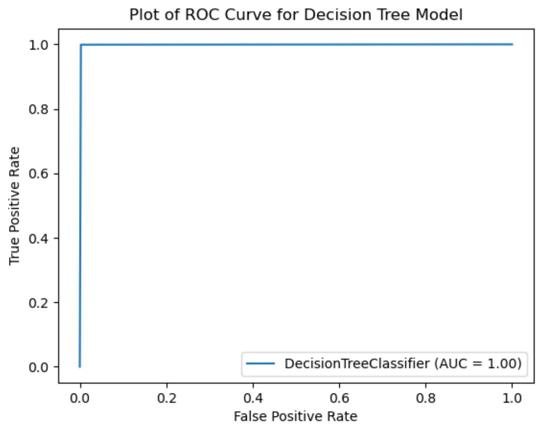


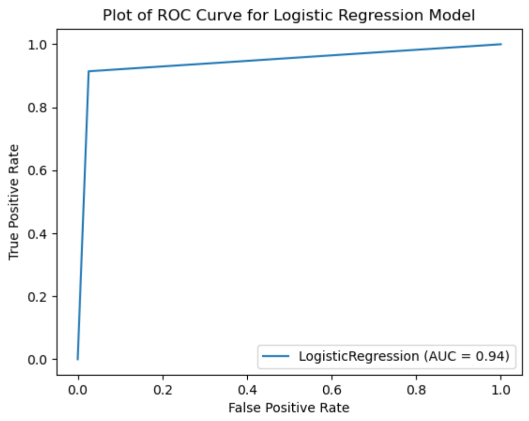


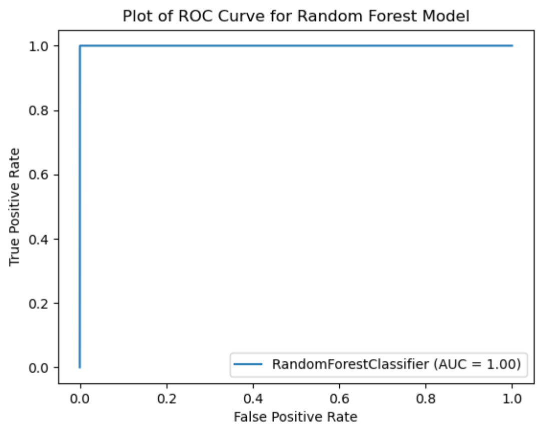

Confusion Matrix for XGBoost Classifier
56436
27
0
56839
True label
Predicted label
0
1
0
1
50000
40000
30000
20000
10000
0

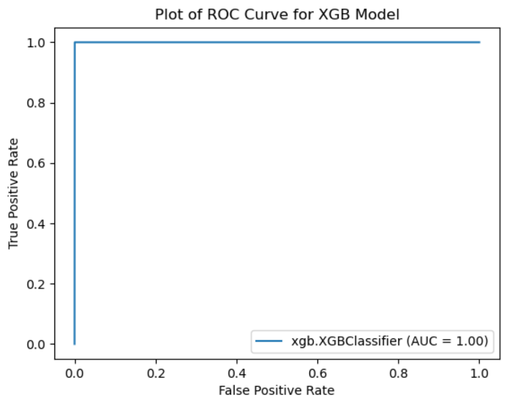

Correlation Matrix

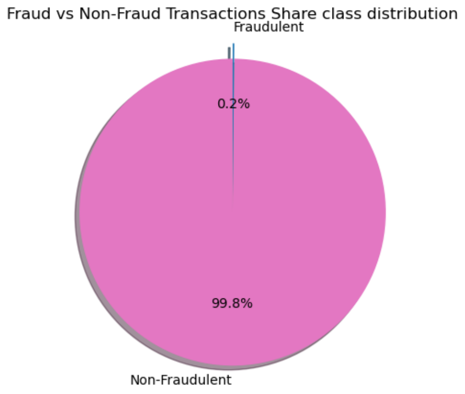

Pairplot of Selected Features for Fraud and Non-F
Class
0
1
V1
V2
V3
Amount

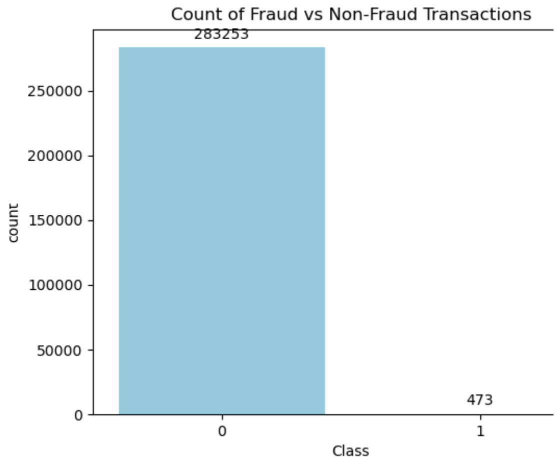

Plot of ROC Curve for Decision Tree Model
DecisionTreeClassifier (AUC = 1.00)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

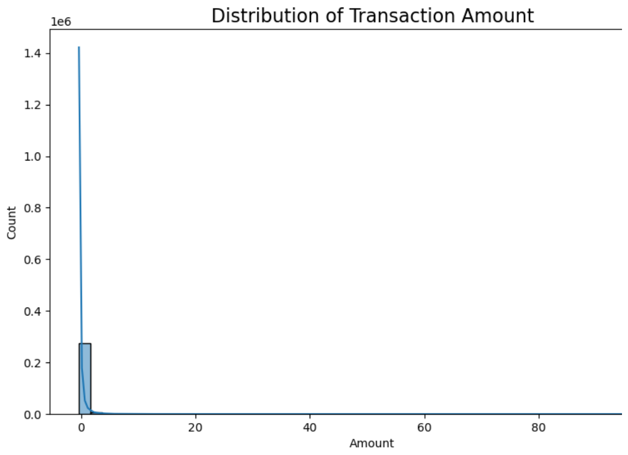

Plot of ROC Curve for Logistic Regression Model
LogisticRegression (AUC = 0.94)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Plot of ROC Curve for Random Forest Model
RandomForestClassifier (AUC = 1.00)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Plot of ROC Curve for XGB Model
True Positive Rate
False Positive Rate
xgb.XGBClassifier (AUC = 1.00)
0.0
0.2
0.4
0.6
0.8
1.0


Fraud vs Non-Fraud Transactions Share class distribution
Fraudulent
0.2%
99.8%
Non-Fraudulent


Count of Fraud vs Non-Fraud Transactions
283253
473
count
Class
0
1
0
50000
100000
150000
200000
250000


Distribution of Transaction Amount
1e6
Count
Amount
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0
20
40
60
80