**Credit Card Fraud Detection using Machine Learning**

**Introduction**

In this project, we are required to make use of machine learning models to make predictions of fraudulent credit card transactions. The rate of fraudulent credit card transactions has been on a rise with the increase of digitalization and the advocacy for a cashless economy.

Credit card fraud detection is a necessity for all banking institutions to safeguard their clients’ funds. With machine learning, financial institutions can study trends and put into place proactive monitoring and fraud prevention techniques. Effective algorithms have helped reduce time spent reviewing transactions, costly chargebacks and fees and rejection of legitimate transactions.

**Data**

This dataset is taken from Kaggle website

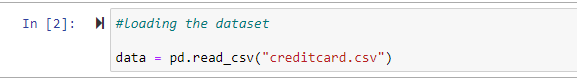
<https://www.kaggle.com/datasets/mlgulb/creditcardfraud>

It has a total of 284807 transactions with 492 fraudulent transactions and 284315 non-fraudulent transactions. The percentage distribution of fraudulent transactions and non-fraudulent transactions is 0.17% and 99.83% respectively.

**Data Cleaning and Preprocessing**

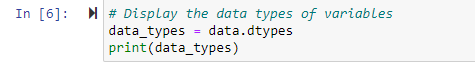
Python programming language has been used.

*Loading the Dataset*



*Exploratory Data Analysis (EDA)*

All our variables are continuous except for class which is categorical.



A look at the data for any missing values and duplicate rows showed no null values and 31 duplicated rows with 1081 transactions.

A screenshot of a computer code

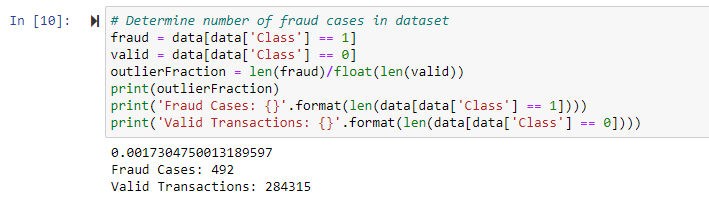
Description automatically generated

A computer code with black text

Description automatically generated

**Data Cleaning**

The data too looked highly unbalanced based on the number of fraudulent transactions. Let’s first apply our models without balancing it and if we don’t get a good accuracy then we can find a way to balance this dataset

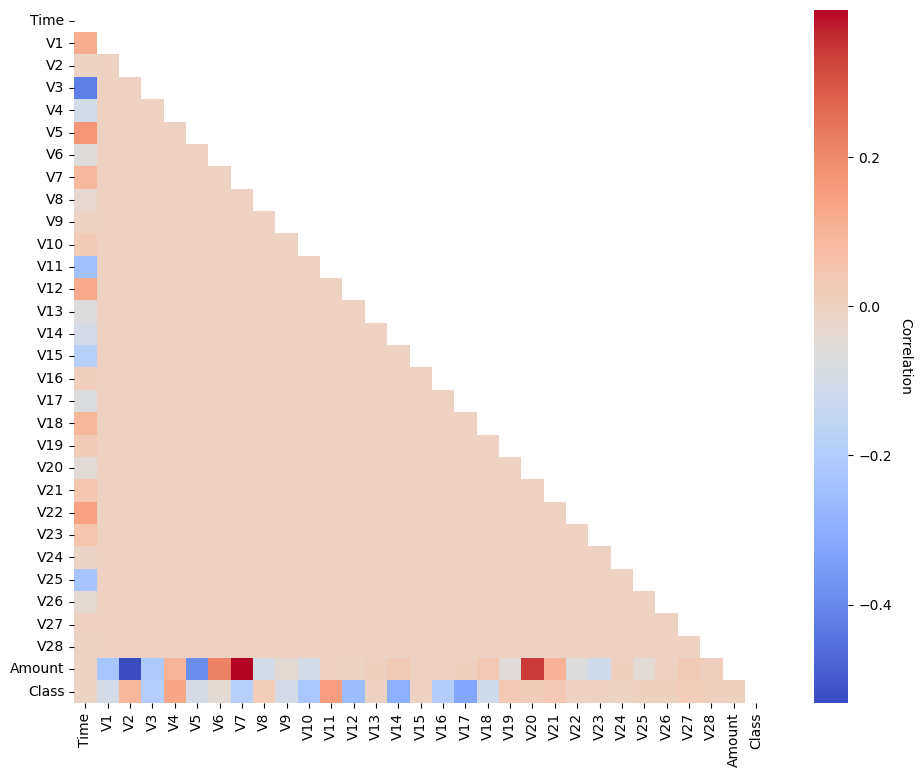
****

We decided to retain the duplicated rows. Dropping them could lead to distortion of the data leading to reduced accuracy during prediction.

**Feature Engineering**

The correlation matrix gives us a visual idea of how the features correlate with each other and helps us predict what variables are most relevant for our prediction.

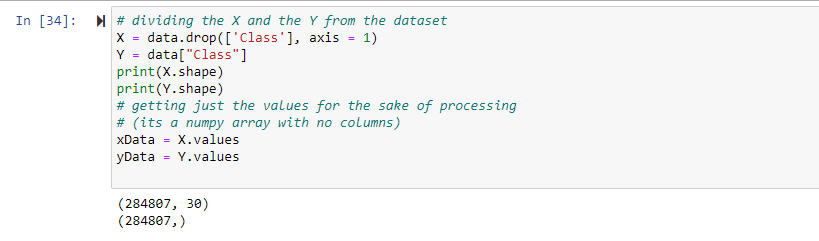
****

In the HeatMap we can clearly see that most of the features do not correlate to other features but there are some features that either has a positive or a negative correlation with each other. For example, V2 and V5 are highly negatively correlated with the feature called Amount. We also see some correlation with V20 and Amount. This gives us a deeper understanding of the Data available to us.

**Model Selection**

Training and Testing Data

We will be dividing the dataset into two main groups. One for training the model and the other for Testing our trained model’s performance.



**Algorithm Choice**

Different algorithms have been deployed on the data. The best performing algorithm based on its metrics will be selected for final deployment.

Some of the models deployed are Random Forest Classifier, Gradient boost Classifier, and XGBoost

*Random Forest Classifier*



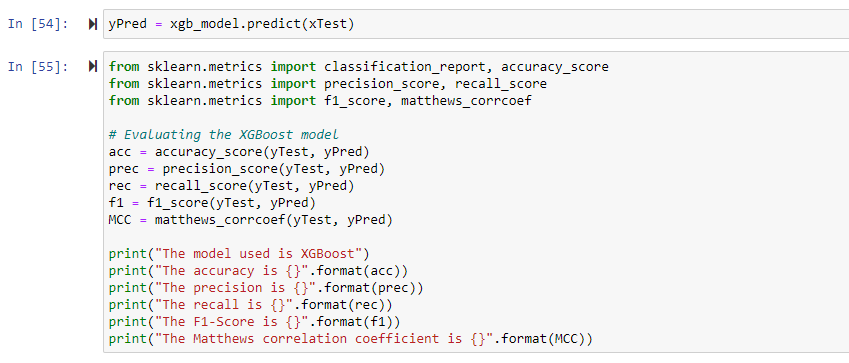
*Gradient Boost Classifier*



*Logistic regression model*

**

*Xgb model*

**

**Evaluation Metrics**

A Summary of each model's performance and any notable findings or insights.

**XGBoost**

A number on a white background

Description automatically generated

The XGBoost model demonstrated exceptional performance in credit card fraud detection, achieving high accuracy of 0.99, precision of 0.97, and a good recall rate of 0.78.

The F1-Score of 0.87 and Matthews correlation coefficient of 0.87 highlight the model's ability to balance precision and recall effectively. This model is a strong choice for fraud detection and can minimize false positives while capturing a substantial number of actual fraud cases. It's important to monitor and potentially fine-tune the model to maintain this high level of performance.

**The Logistic Regression Model**

**A number on a white background

Description automatically generated**

The Logistic Regression model achieved high accuracy of 0.99 and precision of 0.82, which is advantageous for reducing false alarms and correctly classifying fraudulent transactions. However, the model has a lower recall at 0.48, indicating that it may miss a significant number of actual fraudulent transactions.

The F1-Score at 0.61 and Matthews correlation coefficient of 0.63 suggest a balanced performance. Depending on your specific use case, you might consider fine-tuning the model or exploring other algorithms to improve recall if capturing more fraud cases is a priority.

**The Gradient Boost Classifier**

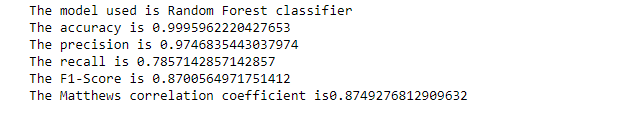
A white background with black text

Description automatically generated

The Gradient Boosting classifier achieved high accuracy of 0.99 but has a trade-off between precision at 0.73 and recall at 0.6. It may have a higher rate of false positives and could miss some actual fraudulent transactions.

The F1-Score of 0.66 and Matthews correlation coefficient 0.66 indicate a balanced performance.

**Random Forest Classifier**



The Random Forest classifier demonstrated impressive performance in credit card fraud detection, achieving high accuracy of 0.99 and precision of 0.97. However, there is a trade-off with recall at 0.78, indicating that some fraudulent transactions may go undetected.

The F1-Score of 0.87 and Matthews correlation coefficient of 0.87 show that the model provides a good balance between precision and recall, making it a strong choice for fraud detection.

**Conclusion**

The Random Forest classifier and XGBoost scored highly with the same metrics. Any of the models will be good for deployment.