FindDefault (Prediction of Credit Card fraud)

Problem Statement:

A credit card is one of the most used financial products to make online purchases and payments. Though the Credit cards can be a convenient way to manage your finances, they can also be risky. Credit card fraud is the unauthorized use of someone else's credit card or credit card information to make purchases or withdraw cash.

It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase.

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

We have to build a classification model to predict whether a transaction is fraudulent or not.

Data:

The dataset for this project can be accessed by clicking the link provided below.
creditcard.csv

Attached below screenshot for dataset head:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

Dataset having 'Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10','V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20', 'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount', 'Class' columns.

The following steps to solve this problem statement:

1) **Exploratory Data Analysis:** Analyze and understand the data to identify patterns, relationships, and trends in the data by using Descriptive Statistics and Visualizations.

2) **Data Cleaning:** This might include standardization, handling the missing values and outliers in the data.

3) **Dealing with Imbalanced data:** This data set is highly imbalanced. The data should be balanced using the appropriate methods before moving onto model building.

4) **Feature Engineering**: Create new features or transform the existing features for better performance of the ML Models.

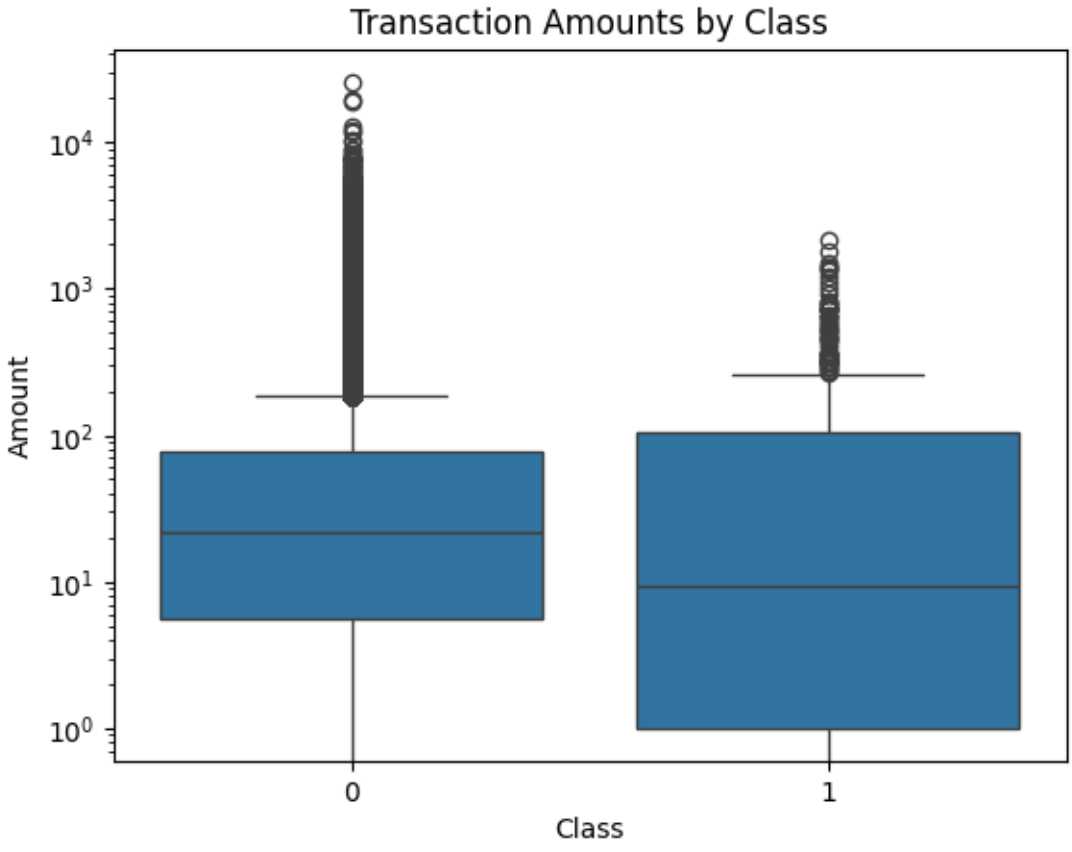
5) **Model Selection:** Choose the most appropriate model that can be used for this project.

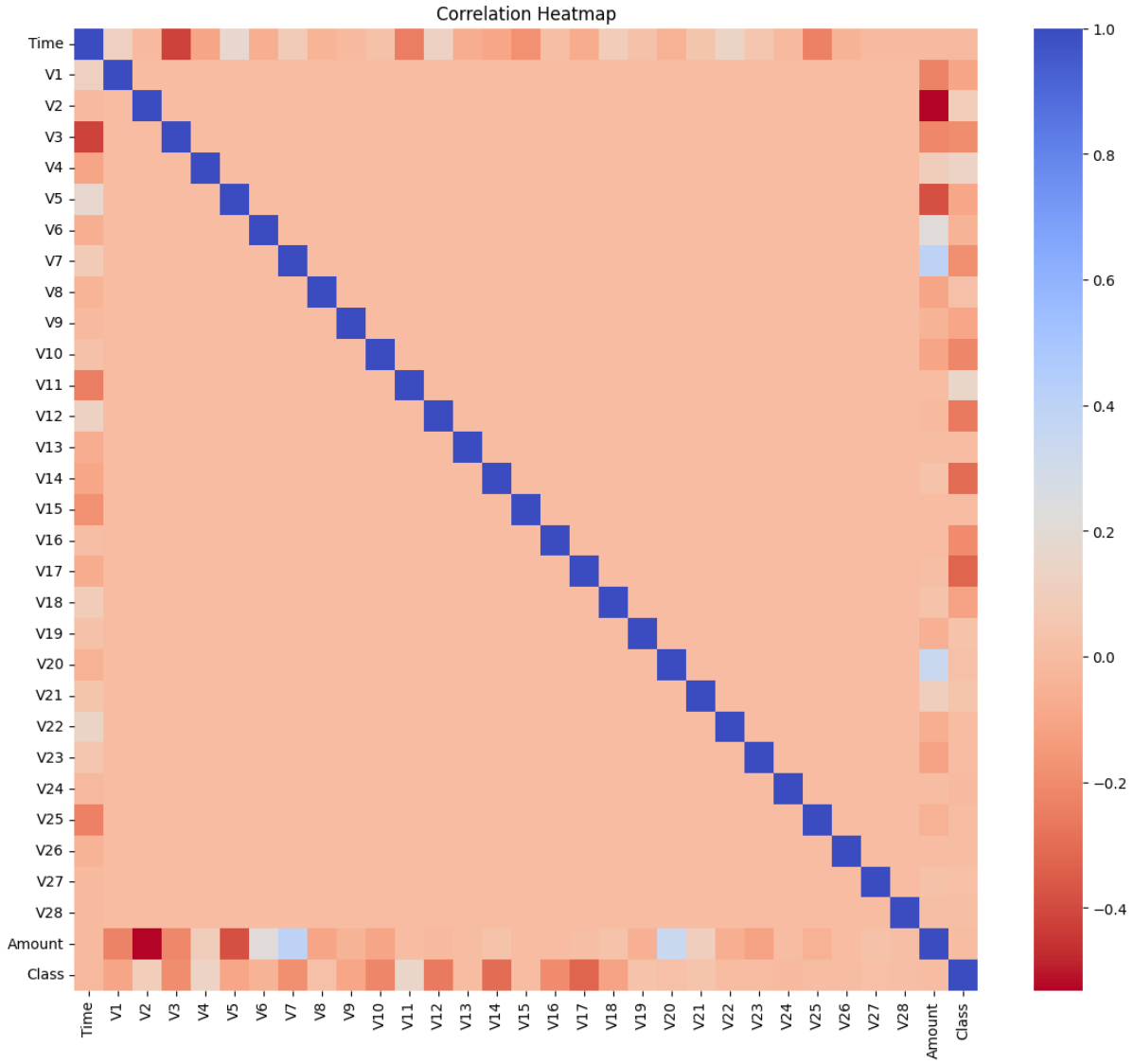
6) **Model Training:** Split the data into train & test sets and use the train set to estimate the best model parameters.

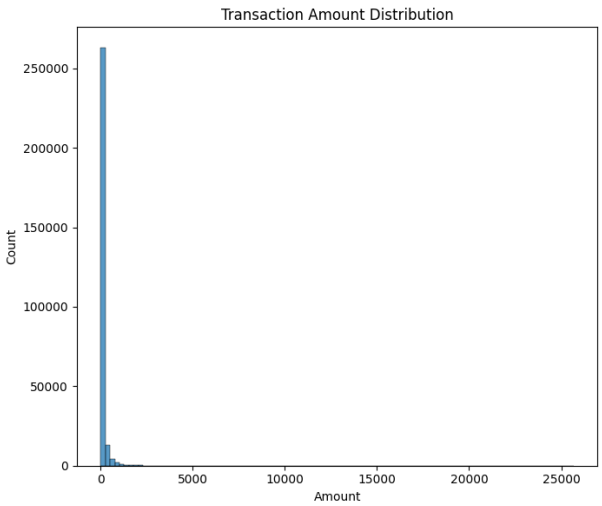
7) **Model Validation:** Evaluate the performance of the model on data that was not used during the training process. The goal is to estimate the model's ability to generalize to new, unseen data and to identify any issues with the model, such as overfitting.

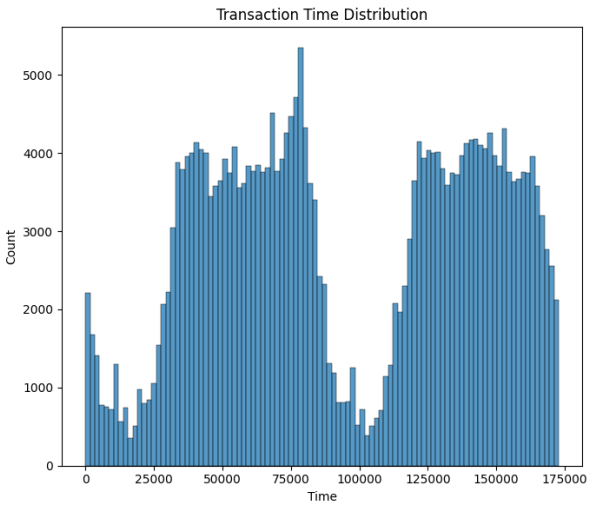
8) **Model Deployment:** Model deployment is the process of making a trained machine learning model available for use in a production environment.

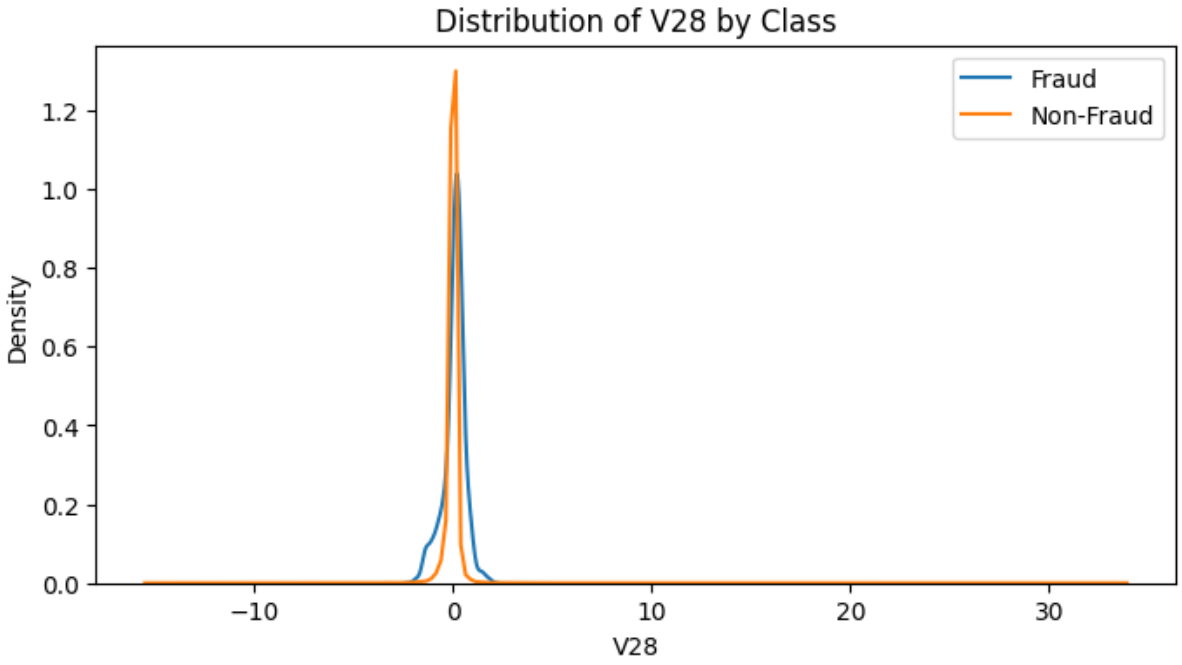
Visualization :

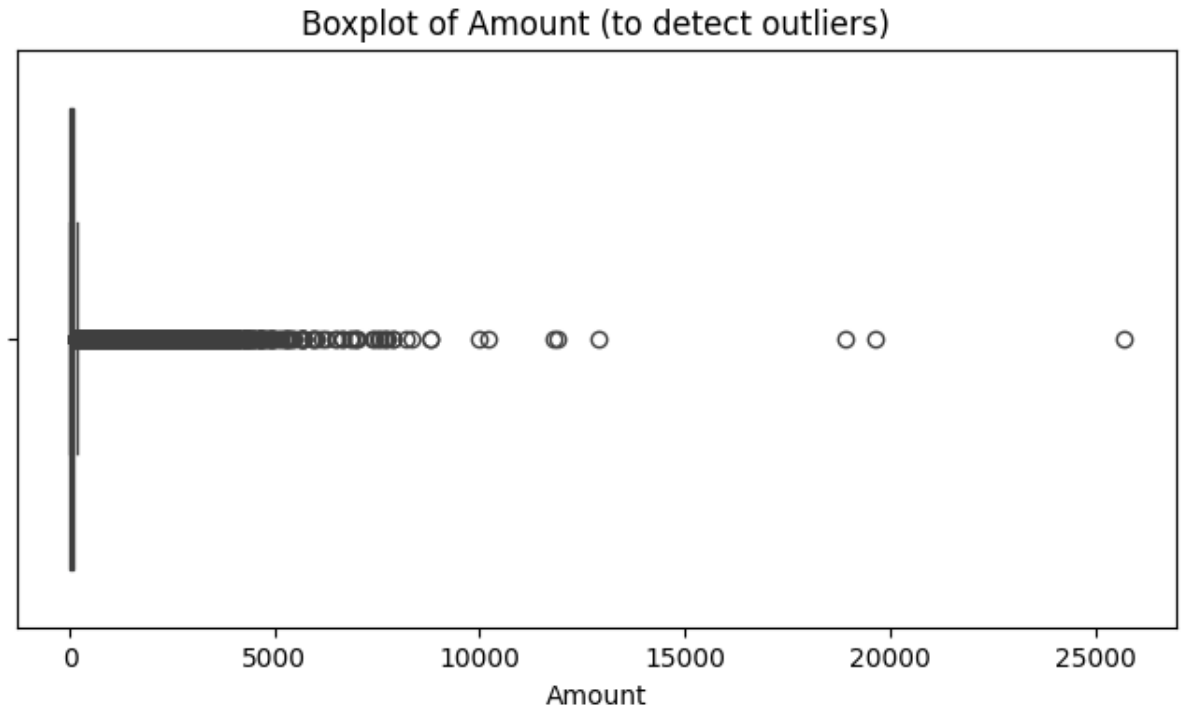



Correlation Heatmap
Time
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
V14
V15
V16
V17
V18
V19
V20
V21
V22
V23
V24
V25
V26
V27
V28
Amount
Class
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4


Transaction Amount Distribution
Count
Amount
0
50000
100000
150000
200000
250000
0
5000
10000
15000
20000
25000


Transaction Time Distribution
Count
Time
0
1000
2000
3000
4000
5000
0
25000
50000
75000
100000
125000
150000
175000





Design Choices:

- **Model Chosen**: Logistic Regression for interpretability and baseline benchmarking.
- **Data Balancing**: SMOTE to address heavy class imbalance.
- **Scaling**: StandardScaler for normalizing features.
- **Train-Test Split**: 80-20 stratified split to preserve fraud ratio.

Performance Evaluation:

Metric	Value
Accuracy	99.92%
ROC-AUC	0.96
Precision	0.83
Recall	0.65
F1-score	0.73

```
Logistic Regression
Accuracy: 0.9991748885221726
ROC-AUC: 0.9599608589918803
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.83      0.65      0.73        98

    accuracy                           1.00     56962
   macro avg       0.92      0.83      0.87     56962
weighted avg       1.00      1.00      1.00     56962
```

- **Precision (fraud class)**: 83% — When the model predicts a transaction as fraudulent, it's correct 83% of the time.
- **Recall (fraud class)**: 65% — The model correctly identifies 65% of actual fraud cases, meaning some fraud cases are still being missed.
- **F1-Score (fraud class)**: 73% — A good balance between precision and recall for the fraud class.

The model performs **exceptionally well on the majority class (non-fraud)**, with nearly perfect scores across all metrics, which is expected due to the class imbalance.

- **High accuracy and ROC-AUC** validate the model's strong predictive ability.
- **Good precision** shows low false positives — important to avoid unnecessarily blocking legitimate users.
- **Moderate recall** indicates the model is missing some fraud cases; improving recall would enhance fraud detection coverage.
- Logistic regression, being a simple and interpretable model, provides a strong baseline for this classification task.

By analyzing these metrics, we can better understand the trade-offs between catching fraudulent transactions and minimizing false positives, which is critical in real-world applications.

Accuracy is high due to class imbalance; precision/recall for class 1 (fraud) is more meaningful.

Future work:

- Consider using more advanced models (e.g., Random Forest, XGBoost) to boost recall.
- Apply **SMOTE** or other resampling techniques to further balance the dataset.
- Perform **hyperparameter tuning** and **feature selection/engineering** for additional gains.
- Monitor model performance over time if deployed in production, especially as fraud patterns evolve.

Conclusion:

We have logistic regression model achieved outstanding performance in detecting credit card fraud, with an overall **accuracy of 99.92%** and a **ROC-AUC score of 0.96**.

These metrics suggest that the model is highly effective in distinguishing between fraudulent and legitimate transactions.

However, due to the **highly imbalanced nature** of the dataset (where only 0.17% of transactions are fraudulent), accuracy alone is not sufficient to judge model performance. Therefore, we also evaluated **precision**, **recall**, and **F1-score**, particularly for the minority class (fraudulent transactions, class = 1):

In summary, while the 99.9% accuracy is encouraging, it must be complemented with these other metrics to ensure that the model not only performs well overall but is also reliable and practical for identifying fraudulent activities in a highly skewed dataset.