

# **Credit Card Fraud Detection**

## **Milestone 3**

### **Project Report**

#### **Submitted By:**

**ASHVINI KUMAR**

#### **Project Duration:**

**04 November 2024 - 05 January 2025**

#### **Internship Program:**

**Infosys Springboard 5.0**

#### **Mentor's Name:**

**Muvendiran**

#### **Submission Date:**

**December 15, 2024**

# Credit Card Fraud Detection System

## Project Overview

This project focuses on detecting fraudulent credit card transactions by employing data preprocessing, thorough analysis, visualization, and machine learning techniques. By analyzing and modeling transaction data, the objective is to uncover patterns and insights that enhance fraud detection accuracy.

## Dataset

The dataset used is sourced from Kaggle's *Credit Card Fraud Detection* dataset. It includes transaction records with features such as Amount, Time, and anonymized attributes (V1 to V28), providing valuable data for fraud detection. The dataset comprises **284,807 transactions** and **31 columns**, including the Class feature that distinguishes fraudulent and non-fraudulent transactions.

## Project Structure

### 1. Data Reading & Preprocessing

- Dataset Loading:** Begin by loading the dataset and cleaning it by handling missing values and duplicates.
- Null Value Check:** To identify null values, use the following code snippet:

```
df.isnull().sum()
```

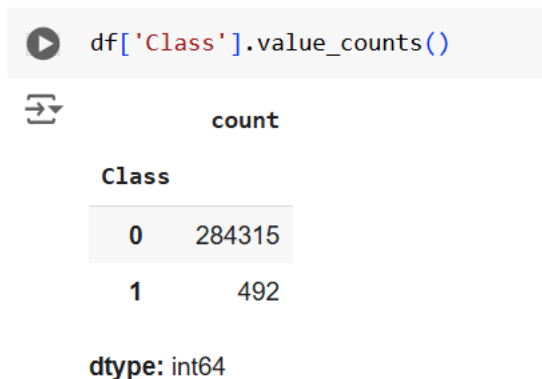
	0
Time	0
V1	0
V2	0
V3	0
V4	0
V5	0
V6	0
V7	0
V8	0
V9	0
V10	0
V11	0
...	...

This generates a count of null values in each column, assuming the DataFrame is named data.

- Descriptive Statistics:** Use the describe() method to generate a statistical summary, helping understand the range and distribution of numerical features.

## 2. Data Analysis

- The dataset exhibits a significant class imbalance with **284,315 non-fraudulent transactions** and only **492 fraudulent transactions**, posing challenges during model training.



```
df['Class'].value_counts()
```

	count
Class	
0	284315
1	492

dtype: int64

---

## 3. Feature Scaling

To enhance model performance:

- Scale the Amount feature using a robust scaler.
- Normalize the Time feature:

```
from sklearn.preprocessing import RobustScaler
```

```
new_df = df.copy()
```

```
new_df['Amount'] = RobustScaler().fit_transform(new_df['Amount'].to_numpy().reshape(-1, 1))
```

```
time = new_df['Time']
```

```
new_df['Time'] = (time - time.min()) / (time.max() - time.min())
```

---

## 4. Data Splitting

- Split the dataset into training, testing, and validation sets with the following sample sizes:
  - Training: **240,000 samples**
  - Testing: **22,000 samples**
  - Validation: **22,807 samples**

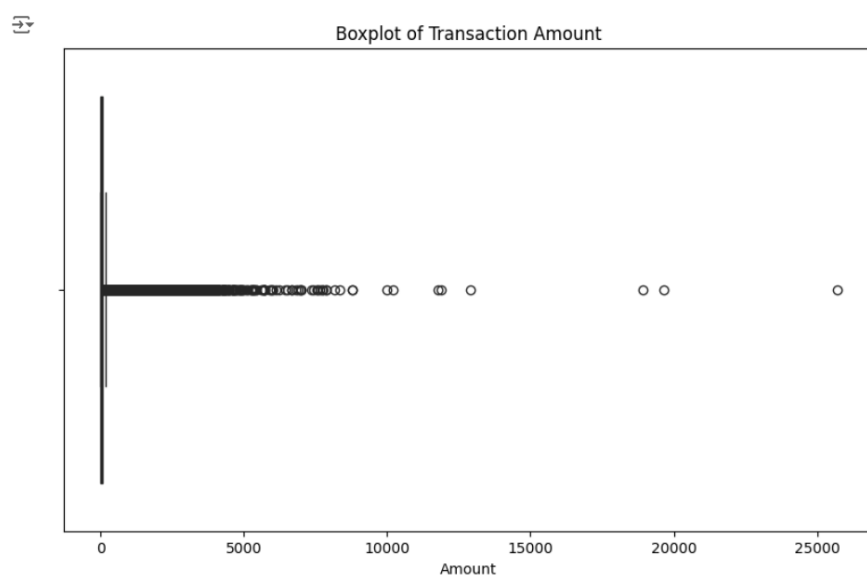
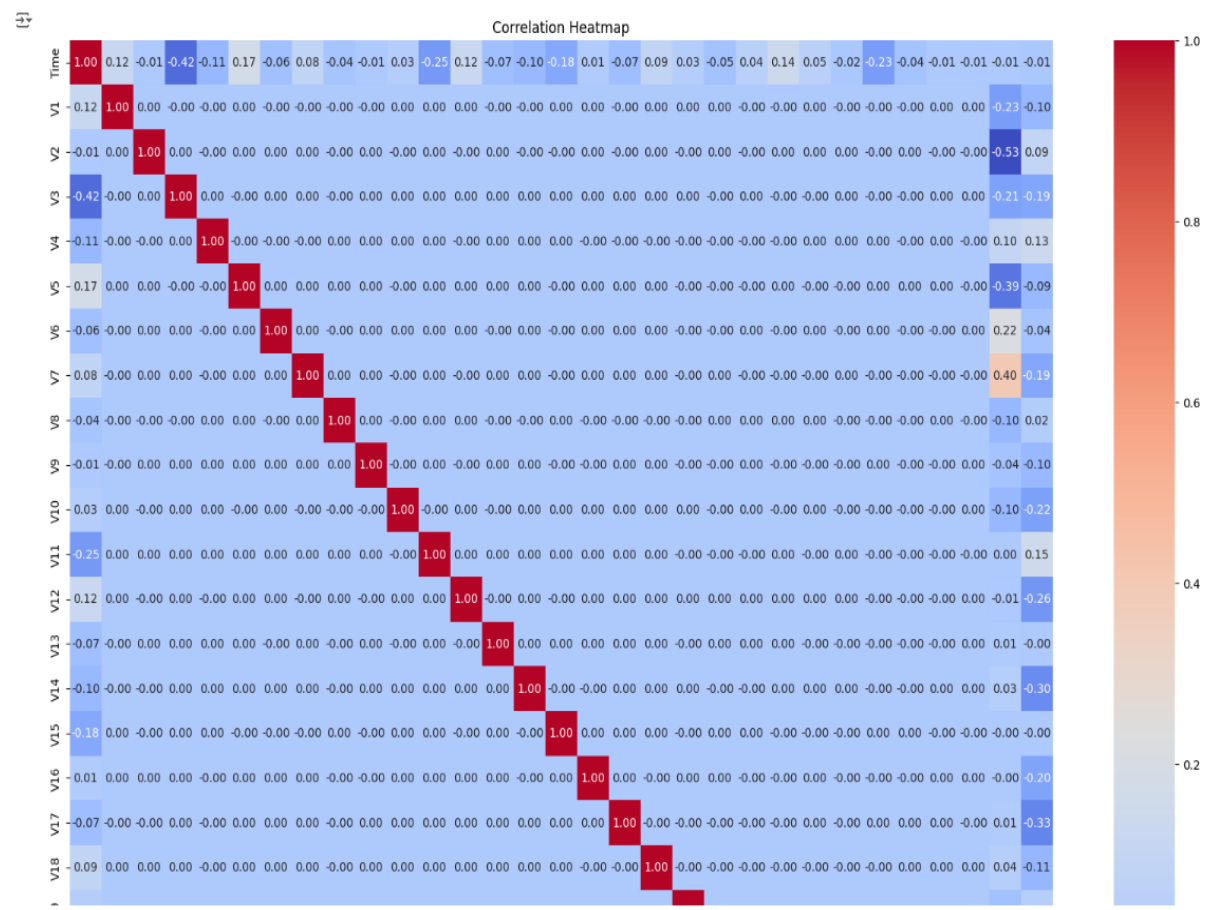
```
[ ] data_fraud = df[df['Class'] == 1]
    data_non_fraud = df[df['Class'] == 0]
```

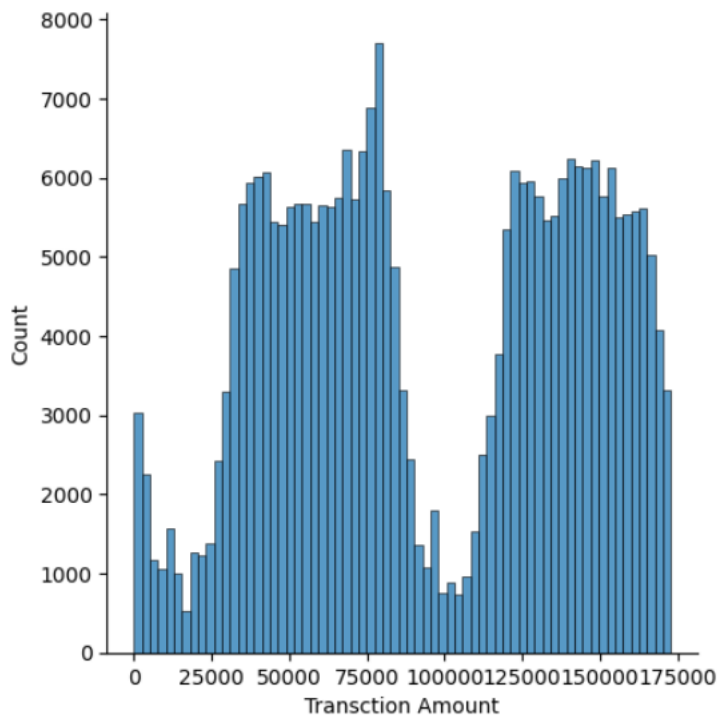
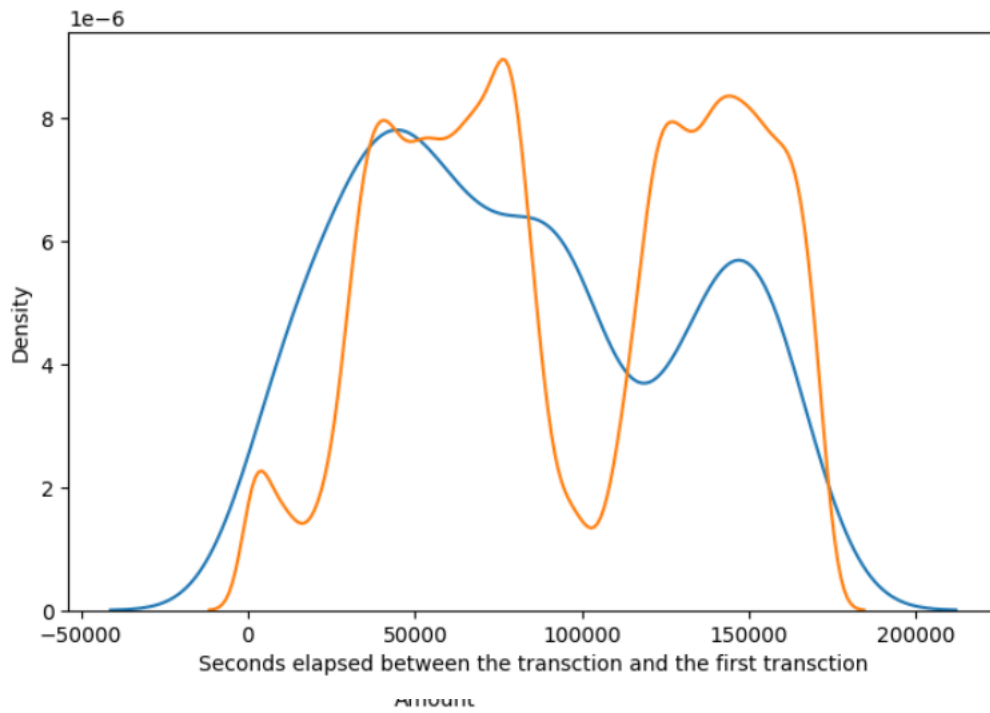
---

## 5. Data Visualization

Visualize the data using:

- **Pie Chart:** Display the proportion of fraudulent and non-fraudulent transactions.
- **Histogram:** Show transaction amount distribution for both categories.
- **Boxplot:** Highlight outliers and distribution patterns in transaction amounts.
- **Scatter Plot:** Examine the relationship between features.





## 6. Model Evaluation

### (a) Logistic Regression:

- Train and test the model using logistic regression.
- Evaluate performance using a classification report and accuracy score on the test dataset.



Accuracy: 0.9726484322881921

Confusion Matrix:

```
[[55322 1544]
 [  14   82]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.97	0.99	56866
1	0.05	0.85	0.10	96
accuracy			0.97	56962
macro avg	0.53	0.91	0.54	56962
weighted avg	1.00	0.97	0.98	56962

•

---

## Conclusion

Credit card fraud is a severe act of financial dishonesty. This project outlines common fraud methods, their detection techniques, and reviews advancements in the field. It demonstrates how machine learning enhances fraud detection through algorithms, pseudocode, implementation, and experimental results.

### Colab Link:

[https://colab.research.google.com/drive/14jOS1Jfo5raI9Ku0AzS\\_MvaKjBxUirXb?usp=sharing](https://colab.research.google.com/drive/14jOS1Jfo5raI9Ku0AzS_MvaKjBxUirXb?usp=sharing)

---