Project Report

Fraud Detection in Credit Card Transactions

1. Introduction

Credit card fraud is a significant issue in the financial industry, leading to billions of dollars in losses each year. This project aims to build a machine learning model capable of detecting fraudulent credit card transactions. The primary objective is to accurately distinguish between legitimate and fraudulent transactions using various machine learning techniques. By leveraging historical transaction data, this model will assist financial institutions in minimizing fraud-related losses and maintaining customer trust.

2. Data Understanding and Preprocessing

Dataset Overview: The dataset used in this project contains records of credit card transactions. It consists of 363 rows and 9 columns, including the target variable `Is Fraudulent` that indicates whether a transaction is fraudulent ('yes') or legitimate ('no').

Columns:

- **Transaction ID:** A unique identifier for each transaction.
- **Customer ID:** A unique identifier for the customer making the transaction.
- **Transaction Date:** The date and time when the transaction occurred.
- **Transaction Amount:** The amount involved in the transaction.
- **Merchant:** The name of the merchant where the transaction took place.
- **Location:** The geographical location of the transaction.
- **Transaction Type:** The type of transaction (e.g., online purchase, in-store).
- **Card Type:** The type of credit card used (e.g., Visa, MasterCard).
- **Is Fraudulent:** A binary label indicating whether the transaction is fraudulent ('yes') or legitimate ('no').

Data Characteristics:

- **Total Rows:** 363
- **Total Columns:** 9
- **Class Distribution:**
  - Fraudulent Transactions: 183 (50.41%)
  - Legitimate Transactions: 180 (49.59%)
  - **Imbalance Ratio:** 1.0167 (Balanced dataset)

Initial Data Inspection:

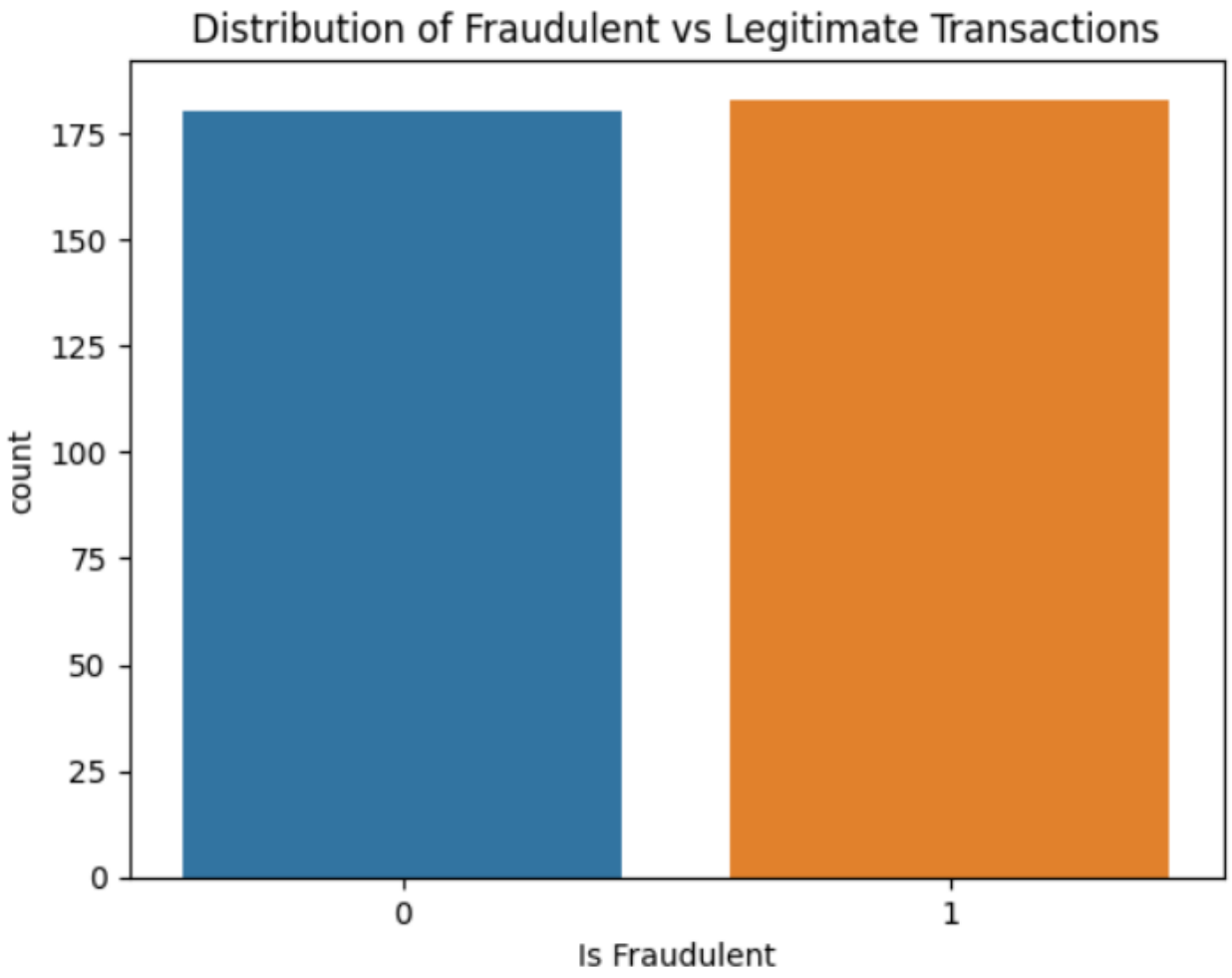
- **Null Values:** There were no missing values in the dataset, ensuring data completeness.
- **Duplicated Data:** No duplicate rows were found, indicating data integrity.

Preprocessing Steps:

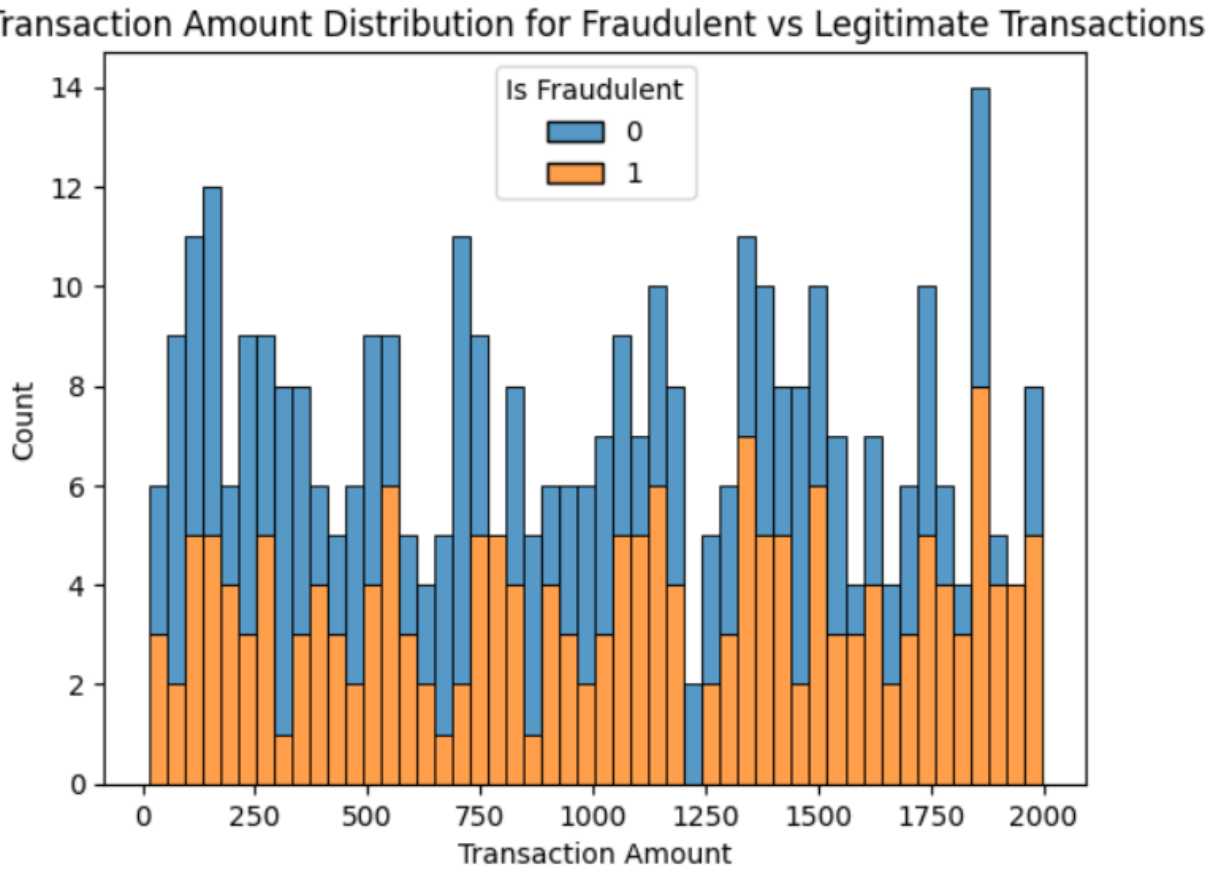
1. **Date Standardization:** The `Transaction Date` column was standardized to ensure consistent datetime formatting. This step was crucial for extracting time-based features like `Transaction Hour`.
2. **Encoding Categorical Variables:** Categorical features such as `Merchant`, `Location`, `Transaction Type`, and `Card Type` were transformed into numerical values using `LabelEncoder`. This was essential for the machine learning algorithms to process these features effectively.
3. **Data Balancing Check:** Since the dataset was already balanced with an almost equal distribution of fraudulent and legitimate transactions, no additional balancing techniques like SMOTE or undersampling were applied.

Visualization:

- **Class Distribution Plot:** A graph plot was created to visualize the distribution of fraudulent vs. legitimate transactions, confirming a balanced dataset.



- **Transaction Amount Distribution:** A histogram was plotted to show the distribution of `Transaction Amount` for both fraudulent and legitimate transactions. This visualization highlighted the range and frequency of transaction amounts, revealing any significant differences between fraudulent and legitimate transactions.



3. Feature Engineering

Feature engineering was performed to create new variables that might help in improving the model's ability to detect fraudulent transactions.

New Features Created:

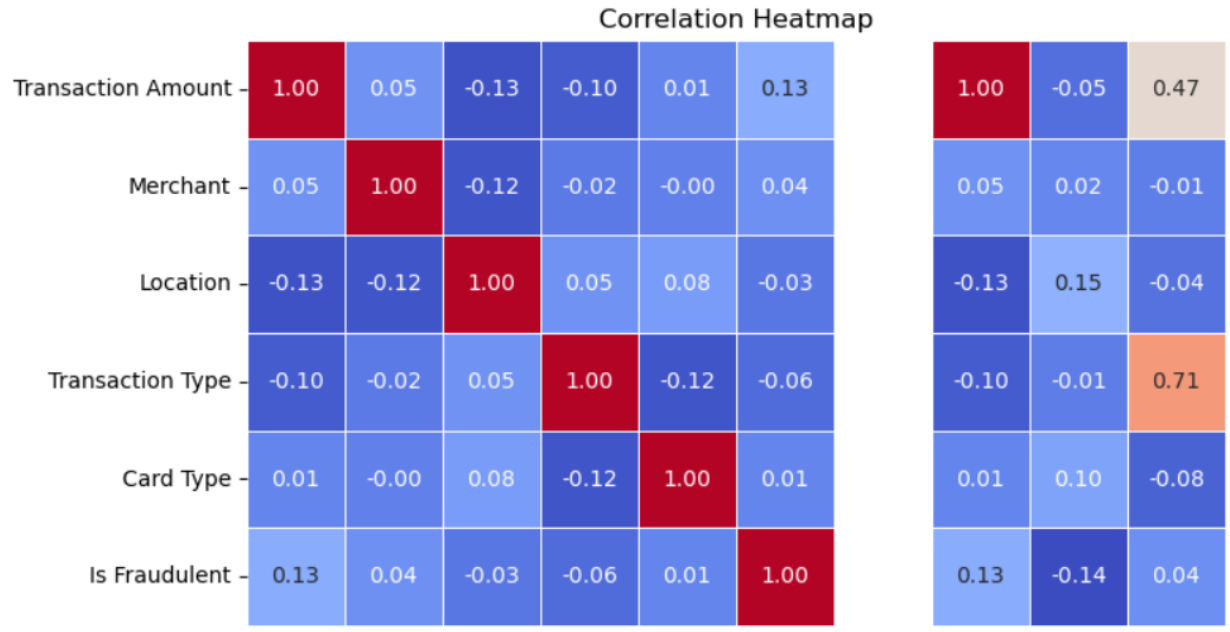
- **Transaction Frequency:** This feature represents the number of transactions made by each customer, indicating transaction activity levels. Higher frequencies might signal potential fraudulent activity.
- **Average Transaction Amount:** The average amount spent by a customer per transaction. It helps in identifying unusual spending patterns.
- **Transaction Hour:** Extracted from the `Transaction Date` to capture the time of day when the transaction occurred. Certain fraud patterns may correlate with specific hours of the day.

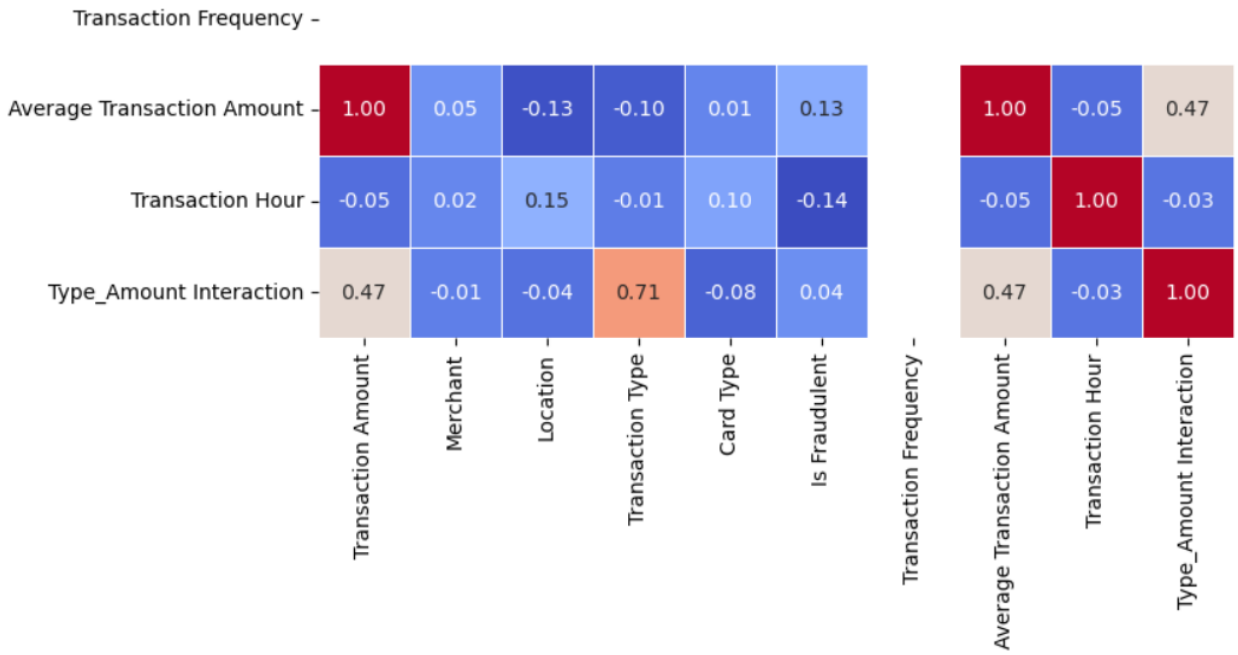
- **Type_Amount Interaction:** A feature representing the interaction between `Transaction Type` and `Transaction Amount`, which may help in identifying unusual spending behaviors specific to certain transaction types.

Updated Feature List: After feature engineering, the dataset included the following features:

- Transaction ID
- Customer ID
- Transaction Date
- Transaction Amount
- Merchant
- Location
- Transaction Type
- Card Type
- Is Fraudulent
- Transaction Frequency
- Average Transaction Amount
- Transaction Hour
- Type_Amount Interaction

Correlation Analysis: A heatmap was plotted to analyze the correlation among the features. This analysis was crucial for identifying potential multicollinearity, which could negatively impact model performance. High correlations between certain features may indicate redundancy, while low correlations can suggest unique contributions to the model.





4. Model Building

In this phase, several machine learning algorithms were implemented to build and evaluate a fraud detection model. The dataset was split into training and testing sets to validate model performance.

Model 1: Logistic Regression

- **Description:** Logistic Regression is a simple and interpretable model often used as a baseline for binary classification problems. It estimates the probability of a binary outcome based on the linear relationship between the dependent variable and one or more independent variables.
- **Performance:**
  - Precision (0, 1): 0.61, 0.57
  - Recall (0, 1): 0.55, 0.62
  - F1-score (0, 1): 0.58, 0.59
  - Accuracy: 0.59
  - AUC-ROC: 0.5957

Model 2: Random Forest Classifier

- **Description:** Random Forest is an ensemble learning method that builds multiple decision trees and merges them to get a more accurate and stable prediction. It is less prone to overfitting and can handle non-linear relationships well.
- **Performance:**
  - Precision (0, 1): 0.55, 0.53

- Recall (0, 1): 0.57, 0.51
- F1-score (0, 1): 0.56, 0.52
- Accuracy: 0.54
- AUC-ROC: 0.5256

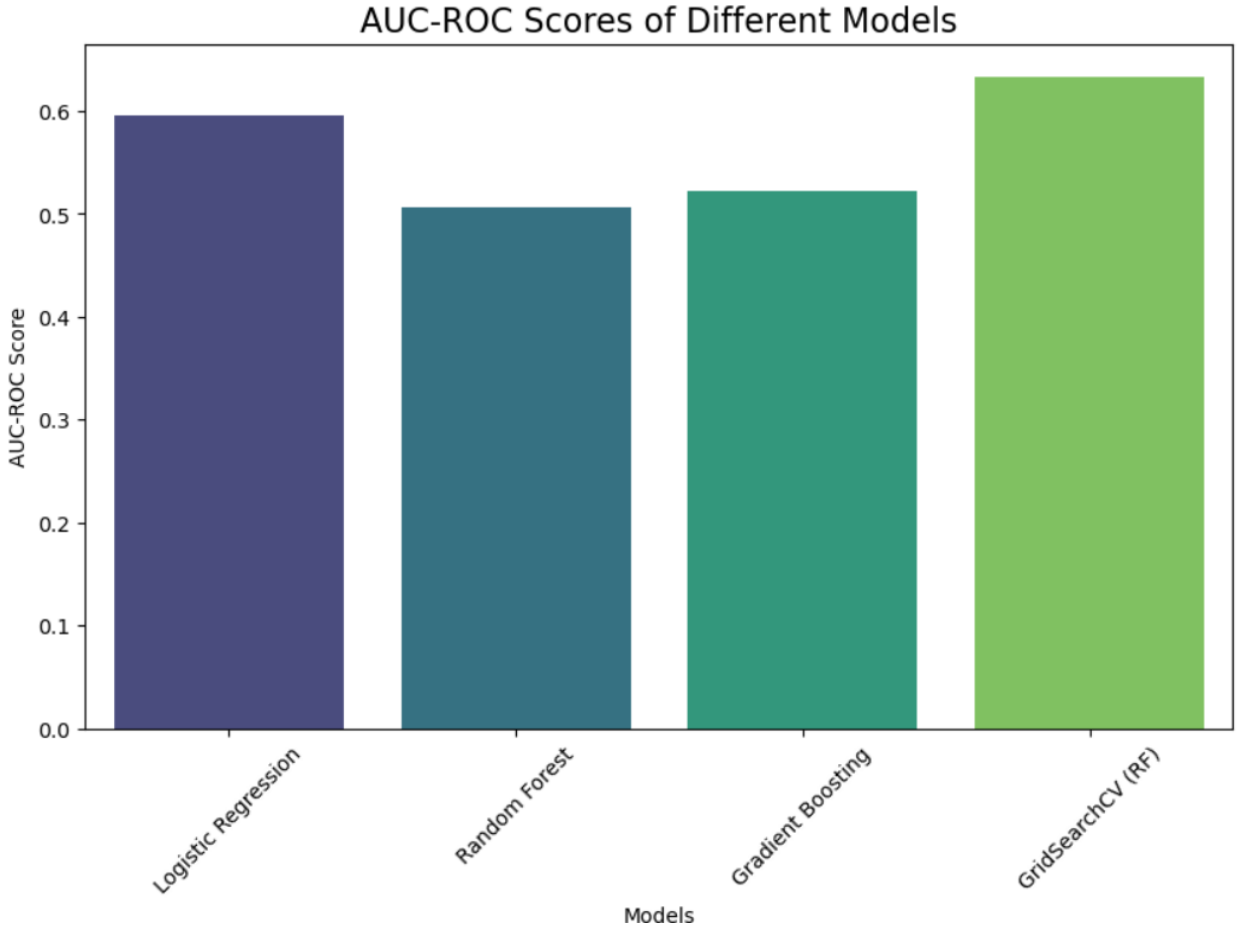
Model 3: Gradient Boosting Classifier

- **Description:** Gradient Boosting builds models sequentially, with each new model correcting the errors of the previous ones. It is a powerful algorithm known for its ability to handle a variety of data distributions and interactions.
- **Performance:**
  - Precision (0, 1): 0.61, 0.58
  - Recall (0, 1): 0.61, 0.58
  - F1-score (0, 1): 0.61, 0.58
  - Accuracy: 0.60
  - AUC-ROC: 0.5192

Model 4: Tuned Random Forest Classifier (with Grid Search)

- **Description:** Grid Search was used to fine-tune the hyperparameters of the Random Forest model. This involved systematically searching through parameter combinations to find the ones that yield the best performance.
- **Best Parameters:**
  - `bootstrap`: True
  - `max_depth`: 30
  - `max_features`: log2
  - `min_samples_leaf`: 1
  - `min_samples_split`: 10
  - `n_estimators`: 50
- **Best AUC-ROC:** 0.6342

A bar graph was created to visualize the AUC-ROC scores for all models, helping in the selection of the best-performing model for deployment.



5. Fraud Detection System Design

Based on the model performance, the tuned Random Forest model was selected for deployment due to its superior AUC-ROC score. The following design considerations were made for the real-time fraud detection system:

System Components:

- **Real-Time Processing:** The fraud detection model was integrated into a pipeline to flag transactions in real-time based on the predicted probability of fraud. This allows the financial institution to take immediate action on suspicious transactions.
- **Continuous Learning:** To ensure the model remains effective over time, it was designed to periodically retrain on new transaction data. This enables the system to adapt to evolving fraud patterns.
- **Integration:** The fraud detection model was designed to integrate seamlessly with the existing transaction processing infrastructure. This integration allows for quick deployment without significant changes to the underlying system.

Recommendations:

- **Data Expansion:** Incorporate additional features such as customer demographics or device information to further improve model performance.
- **Anomaly Detection:** Complement the current supervised learning approach with unsupervised learning techniques to identify novel fraud patterns.
- **Regular Model Updates:** Periodically retrain the model with new data to capture evolving fraud trends and enhance detection accuracy.

6. Conclusion and Recommendations

This project successfully demonstrated the use of machine learning to detect fraudulent transactions in a balanced credit card dataset. By leveraging various machine learning techniques and carefully engineered features, a robust fraud detection model was developed. The key findings and recommendations include:

- **Performance Analysis:** The tuned Random Forest model achieved the highest AUC-ROC score, making it the best choice for deployment.
- **Future Work:** Explore advanced techniques like deep learning and anomaly detection to further enhance detection capabilities.
- **Model Maintenance:** Regularly update the model with new transaction data to maintain its effectiveness in detecting emerging fraud patterns.