Table of Contents

Credit Card Fraud Detection Report

1. Introduction

1.1 Project Overview

This report presents a comprehensive analysis and modeling process for the detection of fraudulent transactions in credit card data. The primary objective is to identify underlying patterns and distinguish legitimate transactions from fraudulent ones through machine learning techniques. The dataset used contains anonymized features representing real-world credit card transactions recorded over a two-day period.

The approach to solving this binary classification problem combines **exploratory data analysis (EDA)**, **feature engineering**, and the development of **robust supervised machine learning models**. Special attention is given to the **class imbalance challenge**, as fraudulent transactions constitute only a very small portion of the total dataset.

1.2 Methodology Overview

To achieve high accuracy and generalizability in fraud detection, the following steps were implemented throughout the project:

- **Extraction of temporal features** (e.g., time-based behavior patterns from the `Time` variable)
- **Feature engineering**, such as converting `Time` into `Hours` and `Days`
- **Scaling and normalization** of numerical features
- **Addressing class imbalance** through under-sampling and SMOTE (Synthetic Minority Oversampling Technique)
- **Training and evaluation of multiple classifiers**, including:
    - Logistic Regression
    - Random Forest
    - XGBoost

Each model was evaluated using multiple performance metrics, including:

- **ROC AUC**
- **Precision-Recall AUC**
- **F1-Score**
- **Confusion Matrix**
- **Optimal threshold selection**

Visualizations are incorporated throughout to provide a deeper understanding of the data distribution, model performance, and decision boundaries. The ultimate goal is to support the creation of an effective fraud detection system that can generalize well to unseen transaction data and contribute to reducing financial loss.

2. Dataset Description

2.1 Summary About Data

The dataset used in this project **Credit Card Fraud Detection** competition. It contains transactions made by European cardholders over a two-day period in September 2013. The dataset is **highly imbalanced**, with only **492 fraudulent transactions out of 284,807 total**, making up approximately **0.172%** of the data.

Each transaction in the dataset is labeled as either **fraudulent (1)** or **legitimate (0)** in the `Class` column.

Key characteristics of the dataset:

- **Number of Records**: 284,807
- **Number of Fraudulent Records**: 492
- **Class Imbalance Ratio**: ~1 fraud in every 580 transactions
- **Number of Features**: 31 (including the label column)

The dataset was split into three subsets for modeling purposes:

- **Training Set**
- **Validation Set**
- **Test Set**

This split ensures that model performance can be assessed objectively, reducing the risk of overfitting and allowing fair evaluation on unseen data.

2.2 Feature Description

Feature Name	Description
Time	Seconds elapsed between each transaction
V1–V28	Result of PCA transformation on original features to ensure anonymity
Amount	The monetary value of the transaction
Class	Target variable (0 = legitimate, 1 = fraud)

Additional engineered features were created during the preprocessing stage:

- **Hours**: Extracted from `Time` to analyze behavior by time of day
- **Days**: Derived from `Time` to identify weekly patterns

These derived features provided valuable temporal insights and improved model accuracy by allowing the algorithm to learn behavioral trends linked to time.

3. Exploratory Data Analysis and Visualizations

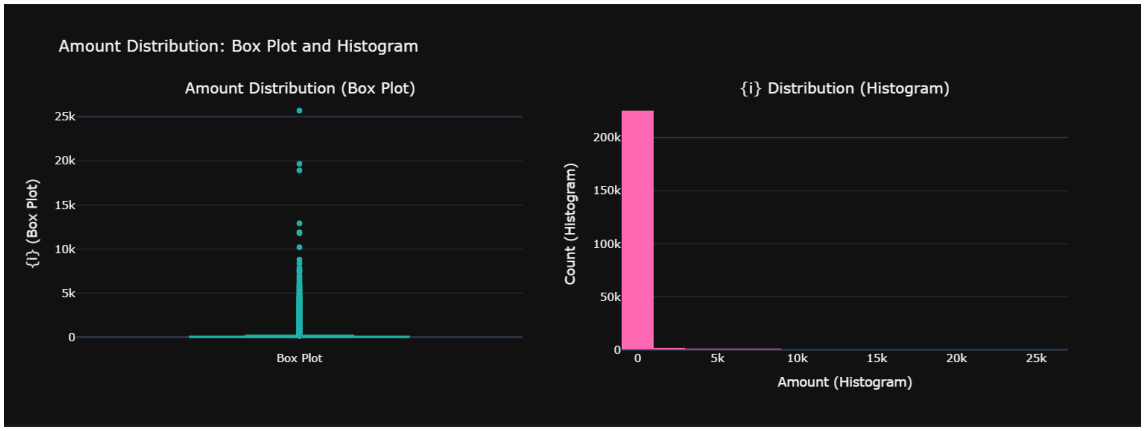
3.1 Amount, Hours, and Day Columns

Box Plot Interpretation

The box plot of the `Amount` feature reveals that the majority of transactions are tightly clustered around relatively low values. However, several extreme outliers are present, representing a small number of transactions with significantly higher amounts. This pattern suggests a **right-skewed distribution**, which is commonly observed in financial transaction data, where most purchases or transfers are small, while large-value transactions are rare but possible.

Histogram Interpretation

The histogram further supports the skewed nature of the data. Most transactions fall into lower value ranges, while the frequency of transactions decreases sharply as the amount increases. This distribution is consistent with typical consumer spending behavior, where low-value transactions dominate, and high-value transactions are infrequent.

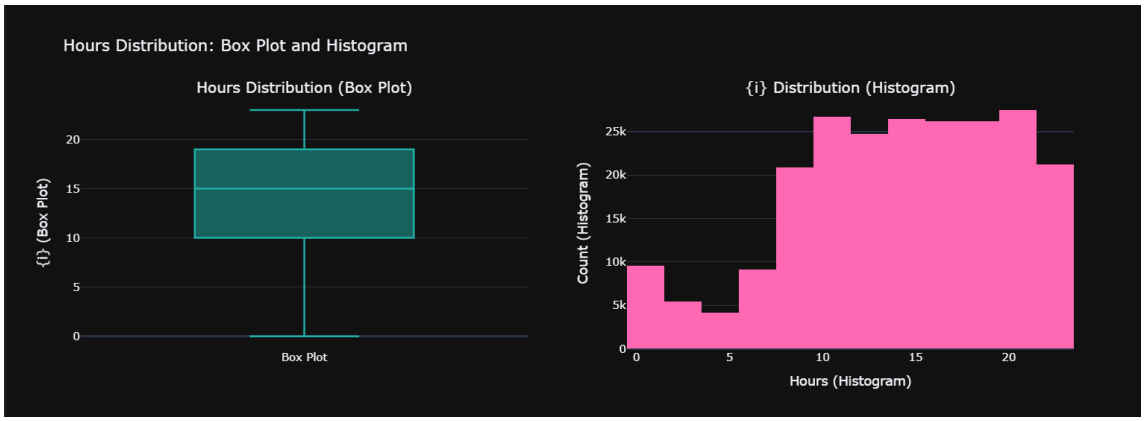


Box Plot Interpretation

The box plot for the `Hours` feature (derived from the original `Time` feature) shows a relatively wide interquartile range, spanning from approximately 5 to 20 hours. The median transaction time is around 15 to 16 hours. The absence of extreme outliers suggests a balanced spread of transactions across different times of day, without any abrupt surges or gaps.

Histogram Interpretation

The histogram illustrates a relatively uniform distribution of transaction frequency across the 24-hour day. Certain hours appear to have slightly higher transaction counts, potentially indicating **peak activity hours**, but overall, the data suggests that transactions occur consistently throughout the day. This indicates a **non-restrictive usage pattern**, typical in digital transaction systems that operate continuously.

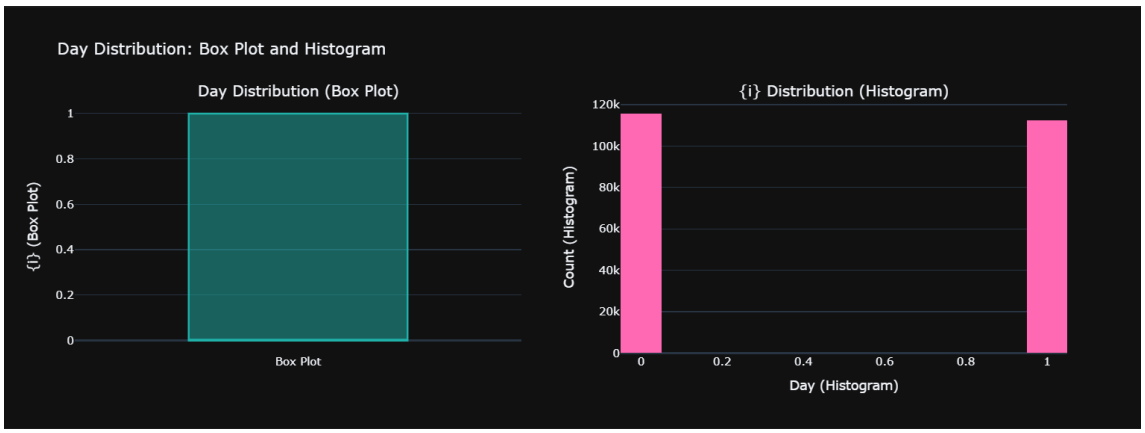


Box Plot Interpretation

The box plot for the `Days` feature (also derived from the `Time` variable) indicates that transactions are evenly distributed between two distinct day categories, labeled as 0 and 1. This distribution implies that the dataset captures transactions occurring over at least two separate days.

Histogram Interpretation

The histogram confirms an almost equal distribution of transactions across the two days. This may correspond to either two consecutive days in the dataset or a categorization such as weekday vs. weekend. The uniformity in distribution suggests **no major variation in transaction volume between the two periods**, implying that transaction behavior remains stable over time.



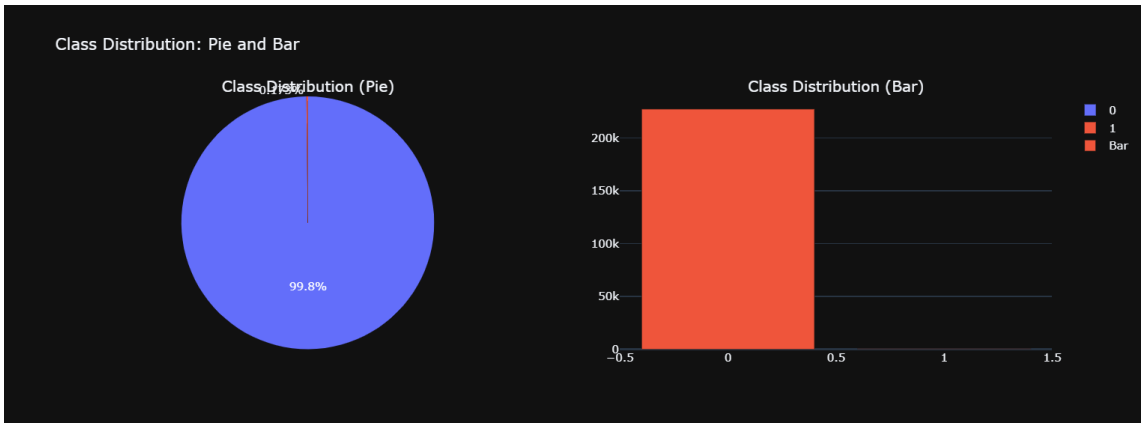
3.2 Label Distribution

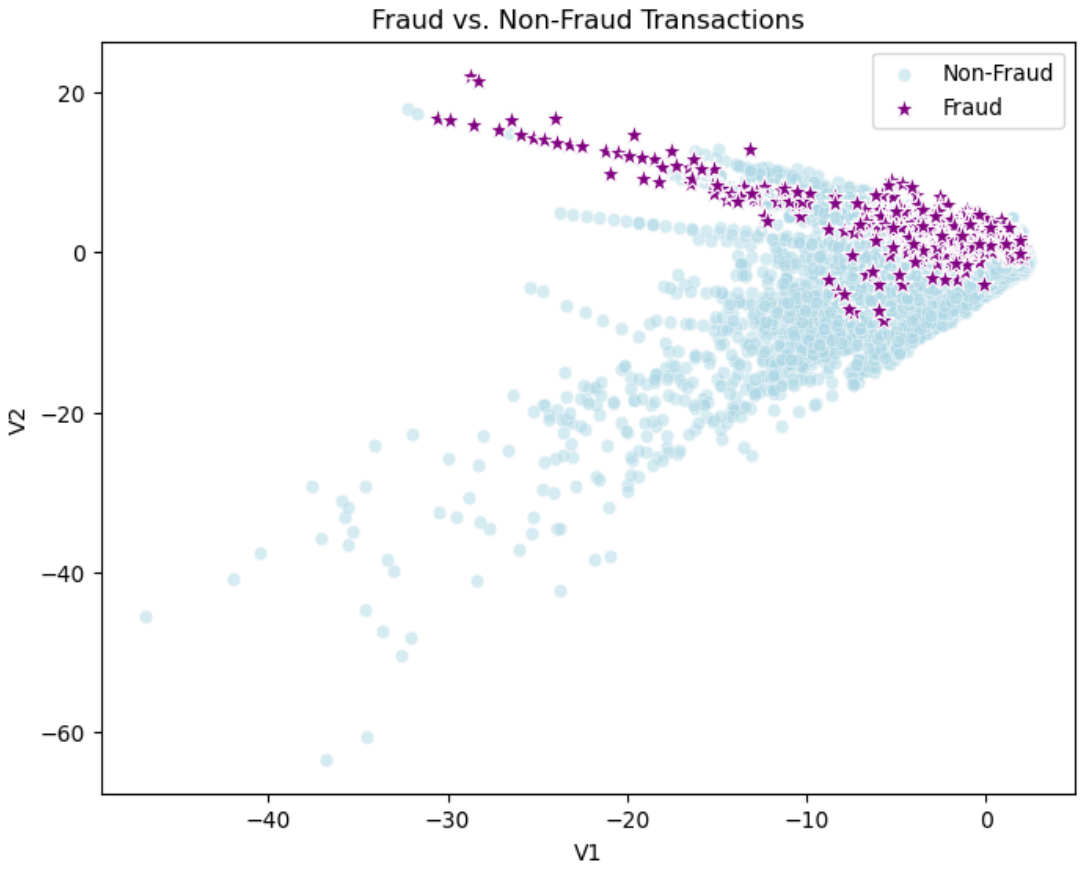
Pie Chart Interpretation

The pie chart visualization of the target variable (`Class`) clearly demonstrates a **severe class imbalance** within the dataset. Non-fraudulent transactions (class `0`) account for approximately **99.8%** of the total observations, while fraudulent transactions (class `1`) make up only **0.2%**. This pronounced imbalance is a common challenge in fraud detection tasks and significantly impacts the performance of standard classification algorithms. Without addressing this imbalance, models may default to predicting only the majority class, achieving high accuracy but failing to detect actual fraud.

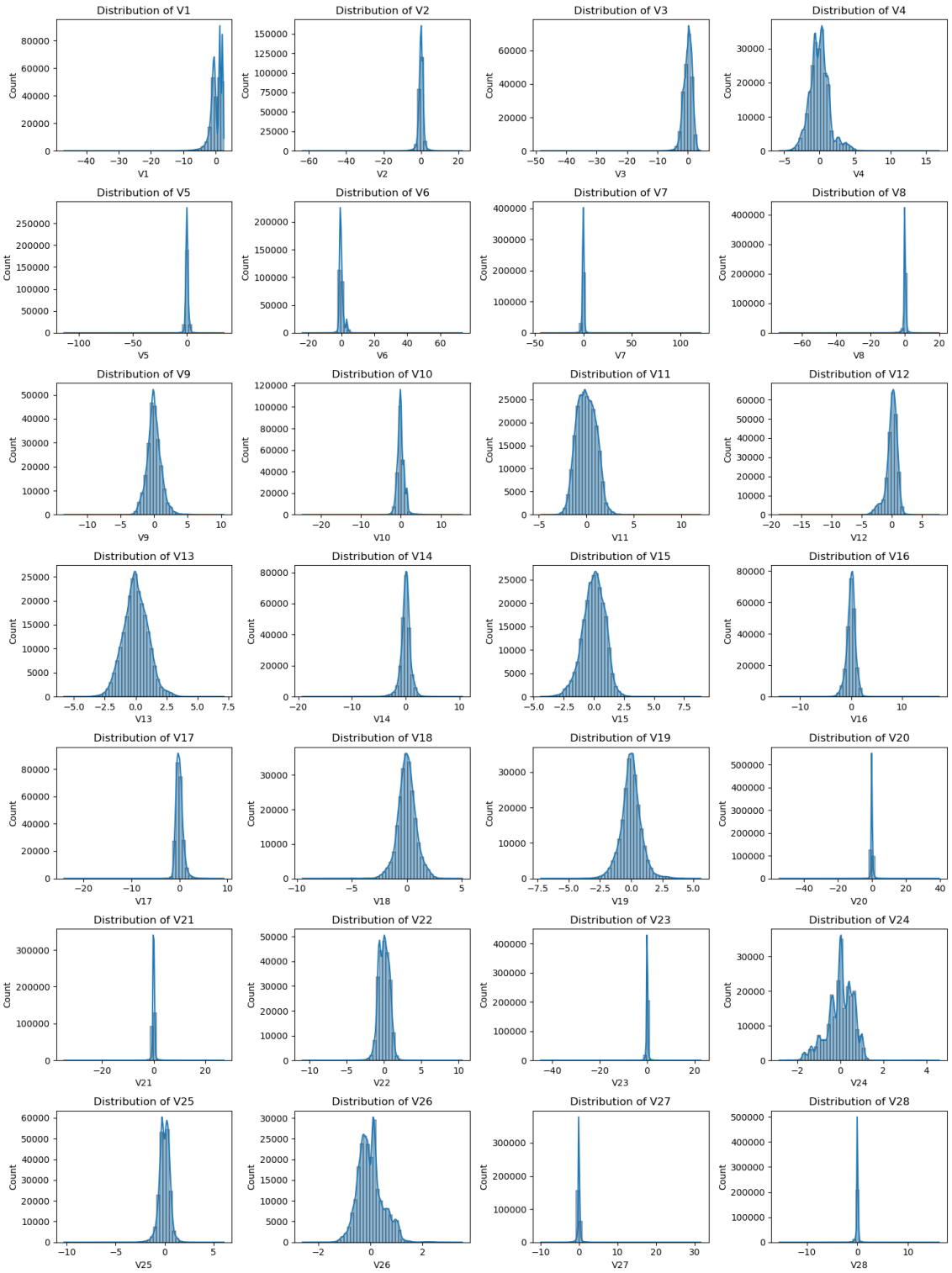
Bar Chart Interpretation

The bar chart further highlights the disproportionate nature of the dataset. The bar corresponding to the fraudulent class is almost negligible compared to that of the non-fraudulent class. This visualization reinforces the necessity of applying **resampling techniques** (such as **SMOTE**, **undersampling**, or **SMOTETomek**) to balance the classes during training. Additionally, it suggests that traditional accuracy metrics may not be suitable for model evaluation, and **precision, recall, F1-score, and AUC** should be prioritized instead.





3.3 Unknown Columns



Histogram Interpretation

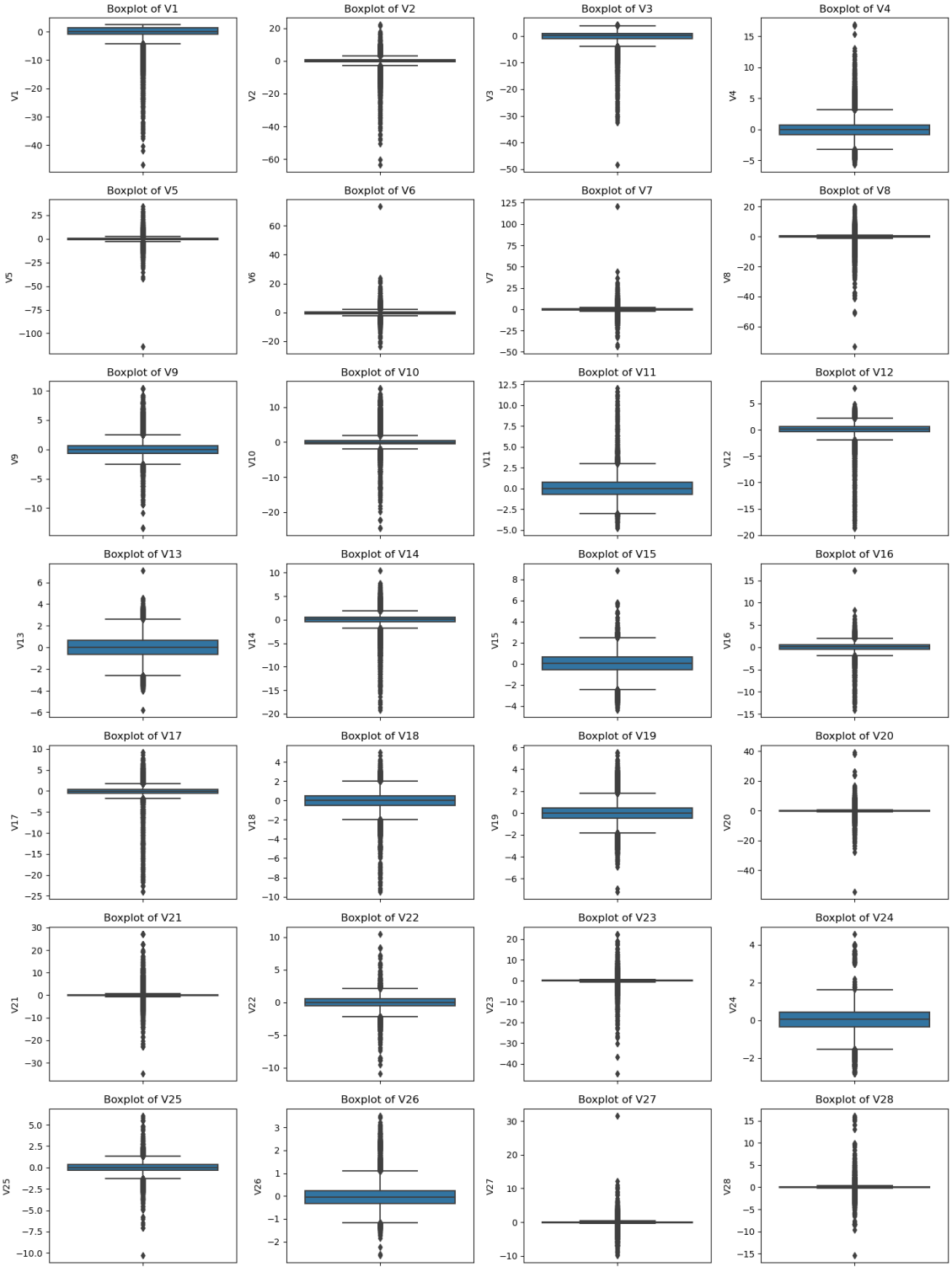
An in-depth inspection of the histograms for the dataset's numerical features reveals a variety of distribution patterns:

- Several features display a **Gaussian-like (normal) distribution**, centered around zero, which suggests that the data has either been standardized or is naturally centered.

- A number of features exhibit **strong skewness**, particularly **right-skewed distributions**, indicating the presence of outliers or rare but significant transaction patterns.
- Many features have a **strong peak near zero**, implying that most of their values lie within a narrow range, possibly resulting in low variance.
- Conversely, some features show **wider spreads**, which may contribute more information and variability to the model during training.

Key Observations

- The overall feature set appears to be **well-scaled**, with the majority of feature values centered near zero, aligning with the use of **standardization** techniques during preprocessing.
- Features that exhibit **long tails** may represent **rare but critical behaviors** in transactional data, potentially indicative of fraudulent activity.
- These varying distribution shapes emphasize the importance of applying **feature selection** (e.g., removing low-variance features) and possibly **non-linear transformations** to enhance model performance.
- Understanding these distribution patterns aids in the design of more effective preprocessing and modeling strategies tailored to the nature of the data.



Box Plot Interpretation

The box plots generated for the dataset's features reveal important characteristics that influence model behavior and preprocessing strategies:

- Several features exhibit a **wide interquartile range (IQR)**, indicating a high degree of variability within the transaction data. This variability may highlight potential distinguishing patterns useful for classification.
- **Numerous outliers** are visible across many features. These outliers likely represent rare or extreme transaction behaviors, such as unusually large amounts

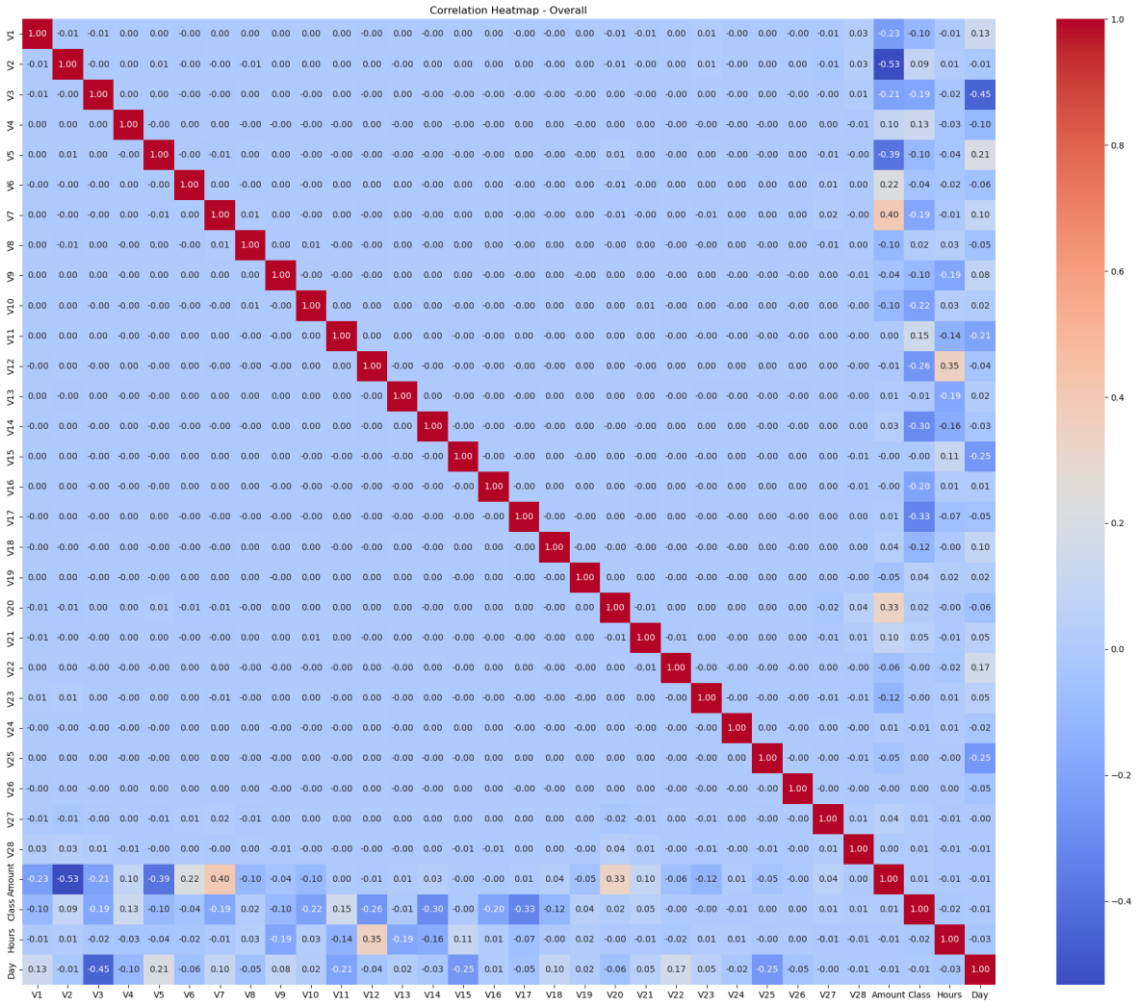
or uncommon feature combinations, which are highly relevant in the context of fraud detection.

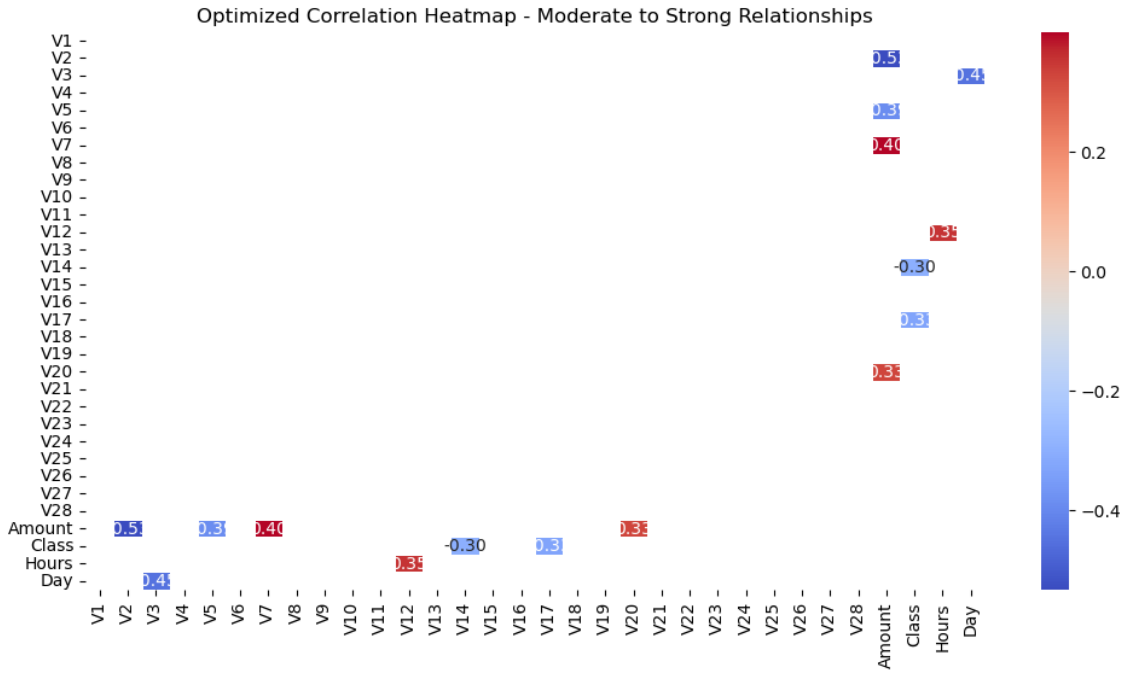
- While some features show a **symmetrical distribution** around their medians, others are **skewed**, reflecting uneven data distribution and potential biases.

Key Observations

- The **presence of outliers** across multiple features suggests the need for thoughtful preprocessing, such as **clipping, transformation, or robust scaling**, to prevent them from negatively impacting model performance.
- The variation in **IQR width** among features implies that certain variables carry more **informative spread**, possibly making them stronger indicators of fraudulent behavior.
- Features with **centralized box plots** indicate tightly clustered values, while **long-tailed distributions** point to irregular but possibly critical data points.
- These insights are vital for guiding decisions around **feature scaling, normalization**, and designing **fraud detection models** that can handle extreme and rare cases without overfitting.

3.4 Correlation Matrix





Key Insights

The correlation heatmap provides valuable insights into the relationships between various features in the dataset:

- Several features show **moderate to strong correlations**, indicating potential redundancy or shared informational content. These relationships could influence how features interact within machine learning models.
- A few features display **strong positive or negative correlations** with the target variable (`Class`), suggesting they may hold predictive power in distinguishing between fraudulent and non-fraudulent transactions.
- The presence of **negative correlations** highlights **inverse relationships** between feature pairs, which could be particularly useful for identifying anomalous behaviors associated with fraud.

Recommended Actions

Feature Selection and Engineering

- Highly correlated features may introduce **multicollinearity**, especially in linear models, potentially leading to overfitting or misinterpretation. Techniques such as **Principal Component Analysis (PCA)** can help reduce dimensionality while preserving variance.
- Features that demonstrate strong correlation with the target variable should be prioritized during **model training and feature engineering**, as they may improve fraud detection performance.

Data Preprocessing

- Features that exhibit **near-zero correlation** with the target can be considered for **removal**, as they are less likely to contribute meaningful information and may act as noise.

- Features with **extreme correlation values** might benefit from **normalization or transformation**, particularly if their scale differs from others, to ensure consistent behavior across models.

Model Optimization

- Tree-based models such as **Random Forest** and **XGBoost** can be used to analyze feature importance and validate whether highly correlated features actually enhance performance.
- Regularization techniques, including **L1 (Lasso)** and **L2 (Ridge)** penalties, can help mitigate the effects of multicollinearity in linear models by penalizing less important or redundant features.

Anomaly Detection Strategy

- Features showing **distinct or unexpected correlation patterns** can inform the design of **custom rule-based fraud detection mechanisms**.
- Further investigation into outliers or unusual pairwise correlations may uncover hidden behaviors or systematic fraud tactics not captured by standard models.

---

4. Modeling

4.1 Pipeline Design

The modeling pipeline is structured to ensure systematic handling of preprocessing, resampling, model training, and evaluation. The main stages are:

- **Feature Engineering**: The "Time" feature is converted into two new features — `Hours` and `Days` — to better capture temporal patterns.
- **Scaling and Feature Selection**: Standardization is applied using `StandardScaler`, and features with variance below a threshold (`0.01`) are removed via `VarianceThreshold`, ensuring only informative features are retained.
- **Model Training**: Multiple models are trained using the resampled training set. Each model is evaluated separately on train, validation, and test data.
- **Evaluation**: Models are assessed based on accuracy, F1-score, ROC AUC, PR AUC, and confusion matrices. Threshold optimization based on F1-score ensures appropriate decision boundaries.

4.2 Model Selection and Configuration

Three classification models were selected for evaluation due to their relevance and diversity in handling imbalanced binary classification:

- **Logistic Regression**
  - Solver: `liblinear` (supports small datasets and binary classification)
  - Random state: 42
  -

- **Random Forest Classifier**
  - Estimators: 100
  - Random state: 42
- **XGBoost Classifier**
  - `use_label_encoder`: False
  - `eval_metric`: "logloss"
  - Random state: 42

All models are instantiated using `get_models()` from the `models.py` module, which encapsulates model configuration for cleaner code management.

4.3 Evaluation Metrics

To ensure comprehensive performance analysis, the following metrics were used:

- **Accuracy**: Measures the overall correctness of the model.
- **F1 Score**: Harmonic mean of precision and recall; crucial for imbalanced datasets.
- **ROC AUC**: Measures the ability to distinguish between classes.
- **PR AUC**: Focuses on precision vs. recall trade-off, more suitable for imbalanced data.
- **Confusion Matrix**: Provides a visual representation of classification performance.
- **Optimal Threshold**: Selected based on maximizing the F1-score on validation data.

The evaluation is performed via the `full_model_evaluation()` function, which also saves plots for:

- Precision-Recall Curve
- ROC Curve
- Precision/Recall vs. Threshold
- Confusion Matrix

All plots are automatically saved in model-specific directories for traceability

4.4 Performance Summary

The models were evaluated on train, validation. Performance highlights

Model	Accuracy	F1-score	PR AUC	ROC AUC	Optimal threshold
Logistic regression	0.9994	0.80	0.77	0.983	0.2485
Random forest	0.9995	0.851	0.87	0.988	0.1528
XGBoost	0.9996	0.871	0.85	0.986	0.3203

Observations:

- XGBoost generally performs best across all metrics due to its robustness and ability to handle imbalance.
- Logistic Regression, while simple, provides a strong baseline and is interpretable.
- Random Forest offers a balance between performance and interpretability, and is more resilient to overfitting than Logistic Regression.

---

5. conclusion

In this project, we developed and evaluated a robust machine learning pipeline for credit card fraud detection, addressing the critical challenges posed by class imbalance and real-world data complexity. The pipeline incorporated essential preprocessing steps, including feature engineering, scaling, variance filtering. This ensured that the training data was well-balanced and informative, improving model generalization.

Three classification models — Logistic Regression, Random Forest, and XGBoost — were trained and evaluated using multiple performance metrics. Among them, XGBoost consistently delivered superior results across key indicators such as F1-score, ROC AUC, and PR AUC, making it the most effective model for this task. However, Random Forest also performed competitively while offering better interpretability than XGBoost.

The evaluation framework, which included threshold optimization and detailed visualizations, provided deep insights into each model's strengths and weaknesses. This comprehensive approach ensures that the final model can effectively identify fraudulent transactions while minimizing false positives — a crucial consideration in real-world fraud detection systems.

Overall, the results demonstrate the effectiveness of combining thoughtful preprocessing, resampling, and ensemble learning techniques in building high-performing fraud detection systems. Future work could explore techniques such as anomaly detection, cost-sensitive learning, or deploying the model in a real-time pipeli.