DSC 550 – Final Term Project

Fraud Detection for Small E-Commerce Platforms

Introduction to the Problem

Financial fraud is a major challenge for businesses and financial institutions, leading to significant monetary losses and reputational damage. Fraudulent transactions often account for a small percentage of total transactions, making them difficult to detect using traditional methods. Our goal is to build an effective fraud detection model that accurately identifies fraudulent transactions while minimizing false positives.

Why This Problem is Important

- Financial Impact: Fraudulent transactions cost businesses billions of dollars annually.
- Customer Trust & Security: Failure to detect fraud can lead to customer dissatisfaction and loss of trust in financial institutions.
- Regulatory Compliance: Many industries require stringent fraud detection mechanisms to comply with financial regulations.
- Data-Driven Decision Making: An optimized fraud detection system can reduce manual investigation efforts and enhance operational efficiency.

Pitch to Stakeholders

"Imagine a scenario where fraudsters exploit vulnerabilities in our transaction system, costing our company millions and eroding customer trust. Our data-driven approach to fraud detection will enable us to proactively identify and prevent fraudulent transactions, reducing financial losses and improving customer confidence. By leveraging advanced machine learning techniques, we can significantly enhance our fraud detection capabilities, ensuring compliance and security. Investing in this project means protecting our bottom line and strengthening our reputation as a secure financial platform."

Data Source

Data was obtained through Kaggle datasets. The dataset was preprocessed to extract meaningful features, such as transaction volume per merchant, deviation from average transaction amounts, and transaction binning to classify amounts into categories.

https://www.kaggle.com/datasets/bhadramohit/credit-card-fraud-detection

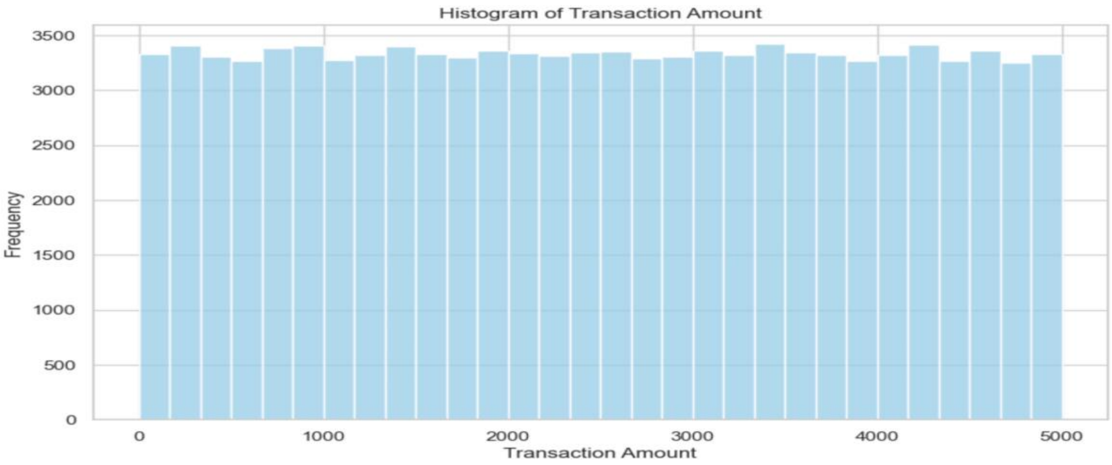
Exploratory Data Analysis (EDA)

The dataset comprises 100,000 transactions generated to simulate real-world credit card activity. Each entry includes the following features:

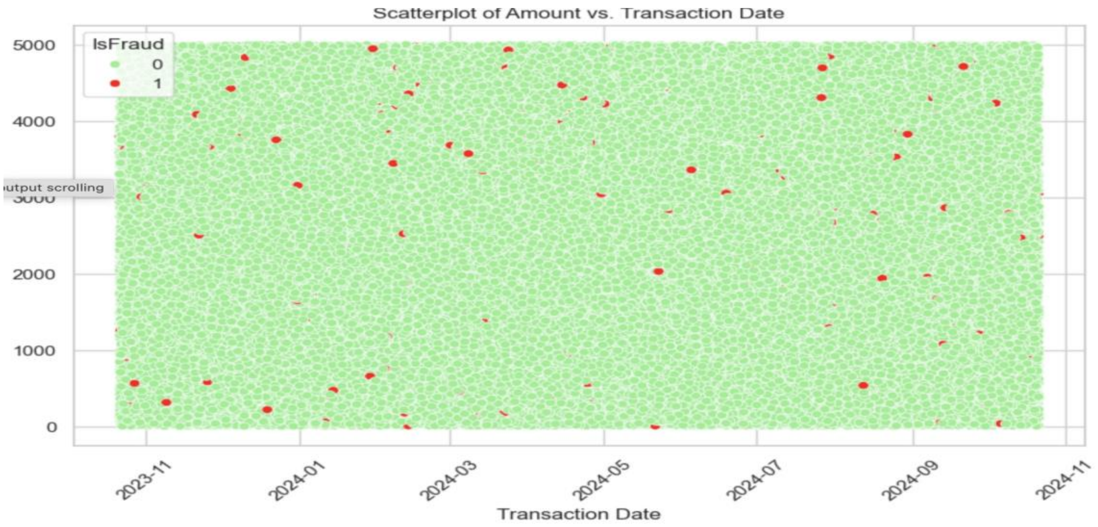
- Transaction ID: Unique identifier for each transaction
- Transaction Date: Timestamp of transaction

- Amount: Transaction amount
- Merchant ID: Unique identifier for the merchant
- TransactionType: Nature of the transaction (e.g., refund)
- Location: Geographical location of the transaction
- Is Fraud: Target variable (1 = Fraud, 0 = Legitimate)

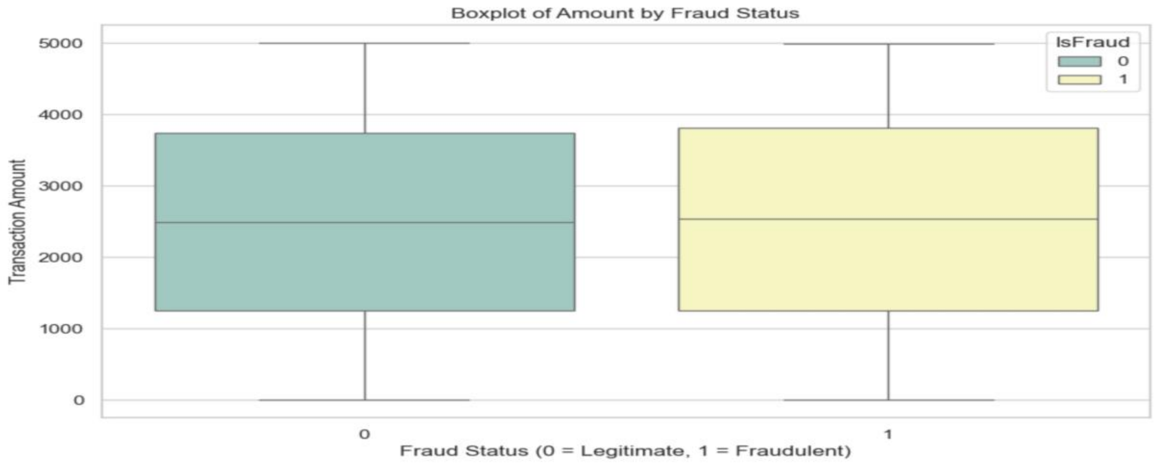
Feature Correlations & Visuals



The histogram shows the distribution of transaction amounts in the dataset. Relatively equal height of the bars across the range suggests uniform distribution with no significant skewness. It does not indicate any specific ranges of transaction amounts that are disproportionately associated with fraud.



The scatterplot shows the relationship between the transaction date and the transaction amount. Fraudulent transactions (Red points) are sparsely and randomly distributed, with no apparent clustering at specific dates. They also span a wide range of amounts, indicating that fraud is not limited to transactions of a particular amount and can occur at any value.



The boxplot compares the distribution of transaction amounts between legitimate and fraudulent transactions. The slight differences in medians and IQRs could be informative for identifying patterns in fraud detection.

Data preparation

Feature Engineering & Transformations

- Extracted Temporal Features
- Created Merchant Behavior Features
- Merchant Avg Amount: Average transaction amount per merchant.
- Deviation From Merchant Avg: Difference between transaction amount and merchant's average.
- merchant_transaction_count: Number of transactions per merchant.
- Binned Transaction Amounts into categories like Low, Medium, High, Very High.
- Converted Categorical Variables (Transaction Type, Location) into dummy variables.
- Handled Missing Values: No arbitrary dropping; filled using mean/median imputation.

Model building and evaluation

- Baseline Model: Random Forest Classifier - We selected Random Forest due to its ability to handle categorical and numerical features, robustness to overfitting, and feature importance insights.
- Handling Imbalanced Data
- SMOTE (Synthetic Minority Oversampling Technique) was used to generate synthetic fraud samples.
- Random under sampling was applied to reduce the number of legitimate transactions.
- Model Training & Cross-Validation

Performance Metrics

Metric	**Score**
Accuracy	0.99
Precision	0.00 (for fraud class)
Recall	0.00 (for fraud class)
F1-Score	0.00 (for fraud class)
AUC-ROC Score	0.471

Confusion Matrix

19751	49
200	0

The model achieves high accuracy (99%) but fails to detect fraud (zero recall for fraud class).

Conclusion

Key Findings from Analysis & Model Building

- The dataset is highly imbalanced (99% legitimate transactions, 1% fraud), making it difficult for the model to correctly identify fraudulent transactions.
- The Random Forest model achieved a high accuracy (99%), but this is misleading because it predicts almost all transactions as legitimate.
- The AUC-ROC score (0.471) suggests that the model performs no better than random guessing for fraud detection.
- Feature Importance Analysis highlighted that merchant-related features (e.g., Merchant Avg Amount, Deviation from Merchant Avg, Transaction Amount) are the most influential in predicting fraud.

Is This Model Ready for Deployment?

No, the model is not ready for deployment.

- A fraud detection model must prioritize recall to minimize missed fraud cases.
- This model's inability to detect fraudulent transactions makes it ineffective in a real-world scenario.
- A high accuracy score does not mean good fraud detection—the model is overfitting to the majority class (legitimate transactions).

Recommendations for Improvement

- Address Class Imbalance - Consider cost-sensitive learning, where misclassifying fraud cases is penalized more heavily.

- Try Alternative Models - Boosting algorithms (XGBoost, LightGBM, AdaBoost) may handle imbalanced datasets better Or Anomaly detection methods (Isolation Forest, Autoencoders) could be useful, as fraud cases are rare and unusual.
- Feature Engineering Enhancements - Explore external data sources (e.g., historical fraud reports, user demographics) for better fraud indicators.

Potential Challenges & Future Opportunities

- Imbalanced Data & Low Fraud Cases – Even with resampling techniques, fraudulent transactions are rare and difficult to model.
- Real-Time Fraud Detection – A practical fraud detection system must detect fraud in real time, requiring low-latency and high-speed processing.
- Concept Drift – Fraudsters change tactics over time, meaning the model needs continuous updates and retraining.
- False Positives vs. False Negatives Trade-off – Striking a balance is crucial: too many false positives lead to legitimate transactions being blocked, while too many false negatives allow fraud to go undetected.
- Explainability & Trust – Financial institutions require fraud detection models to be interpretable and justify why a transaction is flagged.

This project provided valuable insights into fraud detection challenges, but the current model is not effective for deployment. Future improvements should focus on handling class imbalance, exploring better algorithms, and enhancing feature engineering to build a fraud detection system that is both accurate and reliable.