**Credit Card Fraud Detection Using Machine Learning**

**Introduction**

Fraud detection in financial transactions is a critical issue faced by banks and payment processors worldwide. Every day, millions of transactions are processed globally within seconds. The convenience of using a credit card - borrowing directly from the bank, avoiding the need to carry large amounts of cash, and making seamless payments - has revolutionized how we pay for goods and services.

However, this convenience comes with risk. Fraudsters are constantly developing new techniques to exploit credit card systems. Credit card fraud alone accounts for billions of dollars in annual losses. With the volume and velocity of modern transactions, manually detecting fraud is no longer feasible.

This project explores how machine learning can be leveraged to detect fraudulent transactions using a real - world, highly imbalanced credit card transaction dataset obtained from Kaggle. If developed correctly, a machine learning model can automatically scan vast amounts of transaction data to identify fraudulent activity efficiently and in real time.

The goal is to build a robust and efficient fraud detection model that maximizes the identification of fraudulent transactions while minimizing false positives. A high-performing model not only reduces financial losses but also improves the customer experience by avoiding unnecessary declines of legitimate purchases - enhancing customer satisfaction and loyalty to their financial institution.

**Project Motivation and Stakeholder Pitch**

Credit card fraud poses both financial and reputational risks to financial institutions. Unchecked fraud can damage customer relationships, erode trust in payment systems, and result in significant financial losses. The broader economic impact is also substantial.

As technology advances, so do the methods used by fraudsters - especially through creative social engineering schemes. Traditional rules-based fraud detection systems are often too rigid and slow to adapt to evolving fraud patterns.

This project proposes a scalable, automated machine learning solution capable of identifying and flagging potentially fraudulent transactions in real time. By implementing this model, key stakeholders - such as fraud analysts, compliance teams, and risk managers - can reduce investigation time, prevent more fraudulent charges, and maintain operational efficiency.

Ultimately, this solution supports the organization’s goals of improving fraud detection accuracy, reducing operational costs, and preserving customer trust.

**Data Overview**

The dataset used in this project, credit\_card\_transactions.csv, was obtained from the Kaggle public database and is intended for academic analysis and research. It contains a mix of legitimate and fraudulent credit card transactions. Each record includes features such as transaction amount, merchant category, transaction type, location, and timestamp data.

A major challenge in working with this dataset is its extreme class imbalance: fraudulent transactions account for less than 1% of all transactions. This skewed distribution requires the use of specialized preprocessing and modeling techniques to avoid a model that simply predicts the majority class.

The bar chart below clearly illustrates this imbalance between fraud and non-fraud transactions:

**Figure: Fraud vs. Non-Fraud Transactions**A blue rectangular bar graph

AI-generated content may be incorrect.

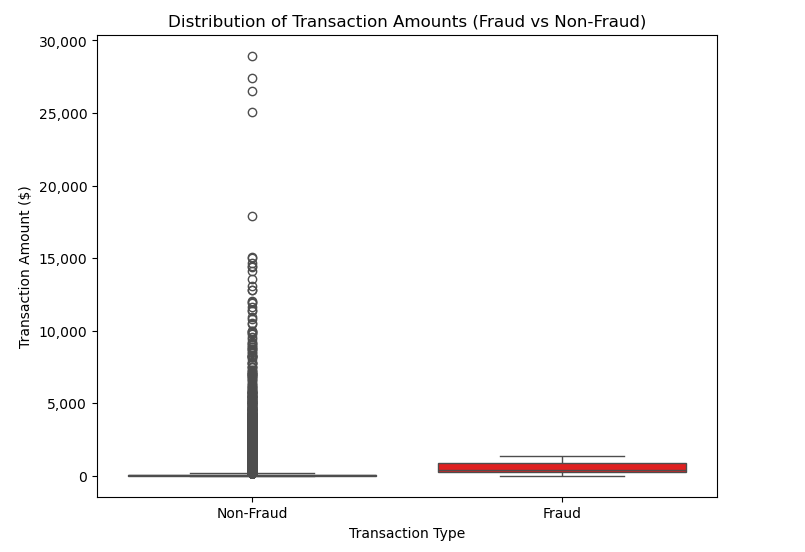
**Milestones 1–3 Summary**

**Exploratory Data Analysis (EDA)**

Milestone 1 focused on exploring the dataset to uncover important patterns and data characteristics. One of the most critical challenges identified in credit card fraud detection is class imbalance. As shown in the Figure: Fraud vs. Non-Fraud Transactions, the dataset is heavily skewed toward non-fraudulent activity. Key findings included:

* Severe class imbalance: Only 5,065 transactions (~0.57%) were labeled as fraud.
* Non-fraudulent transactions: 884,987 records (~99.43%) were legitimate.
* Higher fraud rates were observed for certain transaction types (e.g., international purchases).
* Time-based patterns suggested that fraudulent activity tends to cluster during specific periods.
* Transaction amount differences:
  + Non-fraudulent transaction amounts ranged from $1.00 to $28,948.90.
  + Fraudulent transactions ranged from $1.18 to $1,371.81.
* Fraudulent activity typically involved smaller transaction amounts.

These insights were supported by visualizations such as the Distribution of Transaction Amounts (Fraud vs. Non-Fraud) chart, which highlighted the contrast in transaction behavior between fraudulent and non-fraudulent records.



**Data Preparation**

Milestone 2 focused on preparing the data for machine learning by performing several essential preprocessing steps:

* Categorical features such as transaction type and merchant category were encoded using one-hot or label encoding.
* Timestamp features were decomposed into day, hour, and weekday to uncover time-based patterns.
* Unnecessary features (such as unique IDs or redundant columns) were dropped, and duplicate records were removed.
* Missing values were handled by replacing empty strings or invalid entries with NaN, and rows with missing critical values were dropped.
* Transaction amounts were standardized to ensure consistency by converting all values to float type.
* The dataset was split into training and testing sets using stratified sampling to maintain the original distribution of fraud vs. non-fraud cases in both sets.

This preparation ensured that the data was clean, consistent, and suitable for effective model training and evaluation.

**Milestone 3: Model Building and Evaluation**

Milestone 3 focused on developing a baseline classification model and evaluating its ability to detect fraud in an imbalanced dataset.

A Random Forest classifier was trained using the cleaned and preprocessed data. Key performance metrics on the test set included:

* Precision (Fraud): 0.95
* Recall (Fraud): 0.73
* F1-Score (Fraud): 0.83
* Precision-Recall AUC: 0.8943
* ROC AUC: 0.9931

The model performed well overall, achieving high precision and a strong F1-score. The Precision-Recall AUC of 0.8943 indicates the model effectively balances false positives and false negatives in the context of fraud detection. The ROC AUC confirms strong general classification ability.

However, the recall score of 0.73 suggests that while the model avoids many false alarms, it still misses a portion of actual fraud cases. Since missing fraud has higher business risk, this highlighted an opportunity for improvement in subsequent milestones.

**Milestone 4: Enhancements and Final Evaluation**

Milestone 4 focused on improving the baseline model and comparing it with an alternative algorithm.

**1. Hyperparameter Tuning (Random Forest)**

Using grid search with cross-validation, the Random Forest model was optimized. The best-tuned version achieved:

* Precision (Fraud): 0.78
* Recall (Fraud): 0.82
* F1-Score (Fraud): 0.80
* Precision-Recall AUC: 0.8347
* ROC AUC: 0.9954

This version of the model improved recall (fraud detection sensitivity) at the cost of slightly lower precision, indicating more false positives but fewer missed fraud cases.

**2. Threshold Adjustment**

The probability threshold was adjusted to maximize the F1-score (optimal threshold = 0.609), resulting in:

* Precision (Fraud): 0.88
* Recall (Fraud): 0.77
* F1-Score (Fraud): 0.82

Adjusting the threshold improved the balance between precision and recall, reducing false positives while maintaining strong fraud detection performance.

**3. XGBoost Model Comparison**

An XGBoost model was also trained and tuned for comparison. Results included:

* Precision (Fraud): 0.20
* Recall (Fraud): 0.97
* F1-Score (Fraud): 0.32
* Precision-Recall AUC: 0.8203
* ROC AUC: 0.9965

While XGBoost identified nearly all fraud cases (very high recall), its precision was unacceptably low, generating a large number of false positives—making it less practical for real-world deployment without additional calibration.

**Summary of Model Performance**

| **Model** | **Precision (Fraud)** | **Recall (Fraud)** | **F1-Score (Fraud)** | **ROC AUC** | **PR AUC** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| Milestone 3: Baseline RF | 0.95 | 0.73 | 0.83 | – | 0.8943 | Strong baseline with high precision |
| Milestone 4: Tuned RF | 0.78 | 0.82 | 0.80 | 0.9954 | 0.8347 | Improved recall, slightly lower precision |
| Milestone 4: Tuned RF + Thresh | 0.88 | 0.77 | 0.82 | – | – | Balanced trade-off via threshold tuning |
| Milestone 4: XGBoost | 0.20 | 0.97 | 0.32 | 0.9965 | 0.8203 | Very high recall, poor precision |

**Conclusion and Recommendations**

This project demonstrated that machine learning can be effectively used to detect credit card fraud, even in highly imbalanced datasets. The final tuned Random Forest model with threshold adjustment provided the best balance between detecting fraud and minimizing false positives—making it well-suited for deployment in a production setting.

**Key Takeaways:**

* Model selection and threshold tuning are critical in imbalanced classification problems.
* Random Forest remains a strong and interpretable choice for fraud detection.
* XGBoost, while powerful, requires careful calibration to be viable in practice due to its low precision.

**Recommended Next Steps:**

* Test the final model in a live A/B setting or shadow deployment environment.
* Explore ensemble techniques (e.g., stacking Random Forest and XGBoost).
* Consider implementing real-time scoring and continuous model monitoring.
* Collaborate with stakeholders to define acceptable risk thresholds for fraud alerts.