Mastercard Fraud Detection Hackathon Report

Tymoteusz Muter, Jan Sala, Stanisław Gajda, Oskar Pietrzyk

May 2025

1. Project Overview

In the Mastercard Fraud Detection Hackathon, the challenge was to develop a model to detect fraudulent transactions based on a fully synthetic but realistic dataset. The solution mimics real-world practices in fraud detection and follows a complete ML pipeline.

2. Dataset

The dataset includes:

- **Transaction data** with time, location, payment method, amount, and device info.
- **User data** with demographics and financial behavior.
- **Merchant data** with location and risk metadata.

The data is imbalanced (fraud rate $\approx$ 8%), so AUC was preferred over accuracy for evaluation.

3. Feature Engineering

New features were constructed using:

- **Geospatial** features: distances between user and merchant.
- **Temporal** features: time difference and session lengths.
- **Behavioral** features: speed, normalized transaction amount (e.g., amount-to-average ratio).

4. Feature Selection

To identify the most relevant predictors for fraud detection, we applied a diverse set of feature selection techniques on a standardized training set:

- **ANOVA F-test (SelectKBest)**: Selected features with the strongest linear relationship to the target.

- **Mutual Information**: Captured non-linear dependencies between features and the fraud label.
- **Recursive Feature Elimination (RFE)**: Used Logistic Regression to iteratively remove the least informative features.
- **Random Forest Importance**: Ranked features based on impurity reduction across decision trees.

From each method, the top 10 features were extracted. These were then aggregated and deduplicated into a unified final feature set for modeling. This hybrid strategy ensured both robustness and diversity, capturing linear, non-linear, and interaction-based relationships within the data.

5. Model Training and Evaluation

The following models were tested:

Table 1: Performance metrics for selected models

Model	Test AUC	Train Acc.	Test Acc.	F1 Score
XGBoost	0.5819	0.9156	0.9143	0.1772
LightGBM	0.5827	0.9156	0.9143	0.1773
CatBoost	0.5854	0.9156	0.9143	0.1790
Neural Network	0.5829	0.9156	0.9143	0.1778
Logistic Regression	0.5844	0.9156	0.9143	0.1795
Random Forest	0.5812	0.9156	0.9143	0.1770

AUC was selected as the main metric due to the class imbalance. Accuracy was misleading as predicting all zeros results in high accuracy but no true fraud detection.

To address the data imbalance, the `scale_pos_weight` parameter was applied to all tree-based models (XGBoost, LightGBM, CatBoost, Random Forest), adjusting for the ratio between negative and positive classes. For the model explanation and analysis, CatBoost was selected as the best-performing model, as it achieved a slightly higher F1 score and AUC than the others.

6. Feature Importance and Interpretability

- **SHAP analysis:** Summary plots highlighted key drivers of fraud predictions.
- **Partial Dependence and ICE plots:** Visualized marginal feature effects.
- **Permutation importance:** Quantified feature contribution to predictive performance.
- **Feature Importance:** Visualized Feature importance

These tools provided insights into model behavior and helped ensure transparency in decision-making.

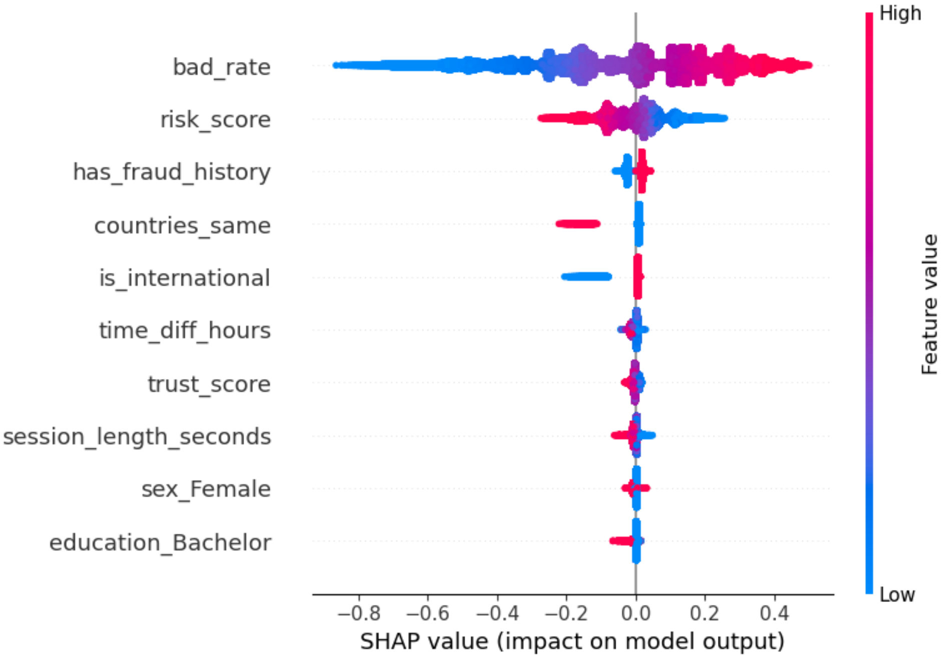

Figure 1: SHAP values for the top 10 features

7. Conclusion

The model shows moderate predictive performance (AUC ≈ 0.58–0.59), constrained by dataset signal. Nevertheless, it captures some fraud patterns via location, timing, and behavioral features. Interpretability was a critical part of understanding model decisions. Future improvements may include ensemble voting, better calibration, and real-time scoring pipelines.