Detecting Credit Card Fraud with Machine Learning: Handling Imbalanced Data and Model Optimization

Ismat Ara Khan

Abstract

Credit card fraud is a major challenge due to its rarity and high cost. This study investigates multiple machine learning models to detect fraudulent transactions in a highly imbalanced dataset. Models evaluated include Logistic Regression, KNN, Balanced Random Forest, XGBoost, and LightGBM, with and without SMOTE oversampling. After hyperparameter tuning and threshold optimization, XGBoost achieved the best performance, detecting 85% of frauds with 91% precision and an F1-score of 0.88. While SMOTE improved recall for linear and distance-based models, tree-based ensemble methods handled imbalance effectively without oversampling. The results suggest that tuned XGBoost is a robust solution for practical fraud detection, with threshold adjustment recommended to balance detection performance and operational cost.

1 Introduction

Credit card fraud is a growing concern in the financial industry, leading to significant financial losses and undermining customer trust. Detecting fraudulent transactions is particularly challenging because they are extremely rare compared to legitimate transactions, creating a highly imbalanced dataset. Traditional detection methods often fail to identify subtle fraud patterns or generate excessive false alarms, which increases operational costs.

The goal of this project is to develop and evaluate machine learning models for detecting fraudulent transactions with high accuracy, recall, and precision. Special attention is given to class imbalance, and techniques such as SMOTE oversampling, class weighting, and threshold optimization are explored. Multiple models, including Logistic Regression, K-Nearest Neighbors (KNN), Balanced Random Forest, XGBoost, and LightGBM, are compared to determine the most effective approach for real-world deployment.

2 Data Overview

The dataset used in this study consists of 284,807 credit card transactions collected over a period of time and contains 31 columns, including 30 numerical features and the target variable *Class* [1].

2.1 Key Features

- V1 to V28: Anonymized principal components derived from transaction attributes to protect confidentiality.
- Time: Number of seconds elapsed between each transaction and the first transaction in the dataset.
- Amount: Transaction amount in dollars.
- Class: Target variable indicating whether a transaction is fraudulent (1) or non-fraudulent (0).

The dataset is highly imbalanced, with fraudulent transactions representing less than 0.2 percent of all transactions. This imbalance poses a significant challenge for model training and evaluation.

3 Methodology

The goal of this study is to develop models that accurately detect fraudulent transactions while handling extreme class imbalance. The methodology follows these key steps:

3.1 Data Preparation

- Split dataset into train 80% and test 20%, stratified by class.
- Scale features (StandardScaler) for models sensitive to feature magnitude (Logistic Regression, KNN).
- Apply SMOTE oversampling [2] on the training set for select models to balance classes.

3.2 Model Selection and Training

Five models were evaluated:

- **Logistic Regression**: A linear classification model that estimates class probabilities using a logistic function and serves as a strong baseline for binary classification problems [3].
- **K-Nearest Neighbors (KNN)**: A distance-based, non-parametric algorithm that classifies observations based on the majority class among their nearest neighbors [4].
- **Balanced Random Forest**: An ensemble tree-based method that improves standard Random Forest by balancing class distributions during bootstrap sampling, making it effective for imbalanced datasets [5].
- **XGBoost**: A scalable gradient boosting algorithm that builds sequential decision trees optimized with regularization to achieve high predictive performance [6].
- **LightGBM**: A gradient boosting framework that uses leaf-wise tree growth to improve training efficiency and performance on large-scale datasets [7].

Models were trained using default and tuned hyperparameters, with evaluation focused on precision, recall, F1-score, and ROC-AUC for the minority (fraud) class.

3.3 Model Optimization:

- Hyperparameter tuning was performed to improve performance.
- Threshold adjustment was applied to balance recall and precision, maximizing fraud detection while limiting false positives, which is particularly important for classification tasks involving highly imbalanced data [8, 9].
- Confusion matrices were used to visualize model performance.

This methodology ensures that the models effectively detect rare fraudulent transactions while handling class imbalance and providing practical deployment-ready performance.

4 Results and Discussion

The models were evaluated on precision, recall, F1-score, and ROC-AUC for detecting fraudulent transactions. Results are presented for models with and without SMOTE, followed by the optimized XGBoost performance.

Table 1 shows the comparison between models trained with SMOTE and without SMOTE shows that oversampling has a model-dependent impact. SMOTE generally increases recall for the fraud class, particularly for Logistic Regression and KNN, indicating improved ability to detect rare fraud cases; however, this often comes at the cost of lower precision, leading to more false positives. In contrast, tree-based ensemble models such as XGBoost and Balanced Random Forest demonstrate strong

Table 1: Model Performance Comparison With and Without SMOTE

Model	With SMOTE				Without SMOTE			
	ROC-AUC	Precision	Recall	F1	ROC-AUC	Precision	Recall	F1
XGBoost	0.980	0.802	0.827	0.814	0.939	0.895	0.786	0.837
Balanced Random Forest	0.979	0.661	0.857	0.747	0.977	0.431	0.888	0.580
Logistic Regression	0.971	0.104	0.898	0.187	0.961	0.831	0.602	0.698
KNN	0.948	0.535	0.847	0.656	0.944	0.959	0.724	0.826
LightGBM	0.947	0.618	0.857	0.718	0.710	0.400	0.429	0.414

performance even without SMOTE, as they inherently handle class imbalance through class weighting and split criteria. For these models, SMOTE provides only marginal recall gains while sometimes reducing precision and increasing computational cost. Notably, XGBoost maintains a strong balance between precision and recall in both settings, confirming its robustness. Overall, the results suggest that SMOTE is beneficial for linear and distance-based models but largely unnecessary for advanced tree-based models, where careful class weighting and threshold optimization are more effective.

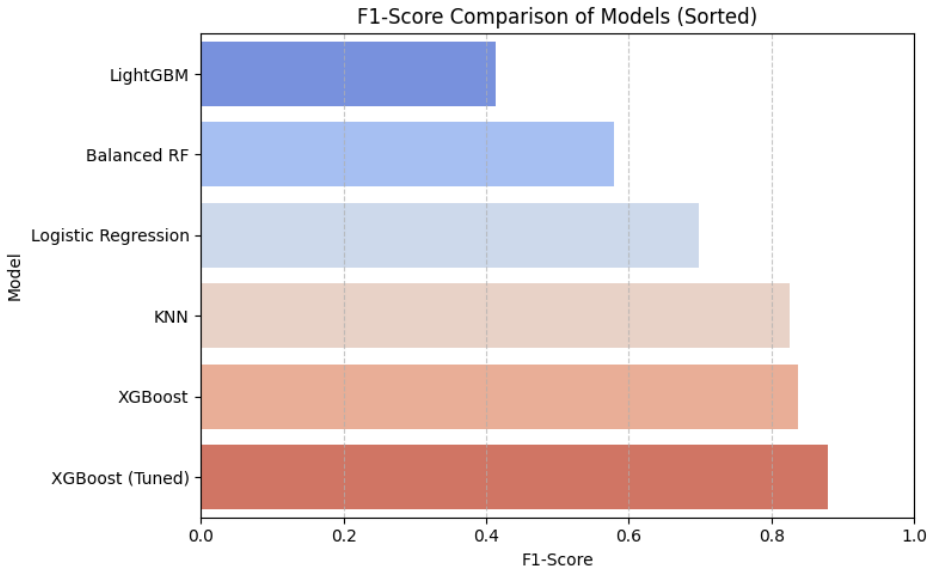

Figure 1: Horizontal bar plot of F1-score for different models, showing improvement after tuning XGBoost.

As illustrated in Figure 1, hyperparameter tuning and threshold optimization significantly improved the performance of the XGBoost model, particularly for the highly imbalanced fraud detection problem. Tuning key parameters such as the learning rate, maximum tree depth, number of estimators, and class imbalance weight allowed the model to capture complex, non-linear fraud patterns while mitigating overfitting. Additionally, adjusting the classification threshold away from the default value of 0.5 provided a more effective balance between precision and recall, prioritizing the detection of fraudulent transactions without generating excessive false positives [8, 9]. As shown in Figure 1, these optimizations increased the model's F1-score to 0.88 for the fraud class, outperforming other models and demonstrating its robustness and practical applicability in real-world fraud detection.

5 Conclusion

This study demonstrates the effectiveness of machine learning for detecting credit card fraud in a highly imbalanced dataset. XGBoost outperformed other models, achieving the best balance between precision and recall. While SMOTE improved recall for linear and distance-based models, it added little benefit for tree-based ensembles and increased computation. Hyperparameter tuning and threshold adjustment further enhanced XGBoost, producing a robust model capable of detecting most fraudulent

transactions with minimal false positives. These results indicate that a tuned XGBoost with threshold optimization provides a reliable, practical solution for real-world credit card fraud detection.

References

[1] Kaggle, "Credit card fraud detection dataset," 2016.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[3] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*. Wiley, 3 ed., 2013.

[4] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[6] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD*, pp. 785–794, 2016.

[7] G. e. a. Ke, "Lightgbm: A highly efficient gradient boosting decision tree," in *NeurIPS*, pp. 3146–3154, 2017.

[8] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[9] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PLOS ONE*, vol. 10, no. 3, p. e0118432, 2015.