PREDICTIVE ANALYTICS

SECOND SEMESTER 2024-26

PROJECT REPORT

CREDIT CARD FRAUD DETECTION

DATE: 1/04/2025

SUBMITTED BY

MAYANK GOEL - 2024H1540832P

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Credit card fraud remains a persistent and evolving threat to both financial institutions and consumers, necessitating robust, real-time detection mechanisms to protect digital transactions and maintain trust in online payment systems. Rapid and accurate identification of fraudulent behavior is critical to minimizing financial losses and preserving customer satisfaction.

This report investigates a publicly available dataset of anonymized credit card transactions from European cardholders, spanning a two-day period in September 2013. Sourced from Kaggle, the dataset contains 284,807 transactions, of which only 492 (0.172%) are labeled as fraudulent. This extreme class imbalance poses significant challenges to standard classification algorithms, often resulting in biased predictions towards the majority class.

Features V1 through V28 are the result of a Principal Component Analysis (PCA) transformation, applied to maintain confidentiality of sensitive transaction information while preserving patterns relevant to classification. The dataset also includes two untransformed features:

- **Time**, representing the number of seconds elapsed since the first recorded transaction
- **Amount**, indicating the transaction's monetary value

The target variable, Class, is binary, with 1 representing fraudulent transactions and 0 indicating legitimate ones.

Given the imbalanced nature of the data and the lack of semantic interpretability due to PCA transformation, the study emphasizes the use of advanced ensemble learning algorithms—notably XGBoost, LightGBM, and RandomForestClassifier—in combination with imbalance-handling strategies such as class weighting and SMOTE. Performance is evaluated using metrics sensitive to class imbalance, such as ROC-AUC and AUPRC, to ensure effective fraud detection.

1.1 Objective:

The primary aim of this project is to develop and assess machine learning models capable of accurately detecting fraudulent credit card transactions. Given the extreme class imbalance, the Area Under the Precision-Recall Curve (AUPRC) serves as the primary evaluation metric, offering a robust measure of performance in skewed datasets. Additional metrics, such as Recall for the fraud class and confusion matrices, will provide deeper insights into model effectiveness.

2. DATASET DESCRIPTION & SOURCES

Dataset: Credit Card Fraud Detection - Kaggle

The dataset provides a rich foundation for studying credit card fraud, encompassing 284,807 transactions with 31 features. The severe imbalance—492 fraudulent transactions against 284,315 legitimate ones—underscores the need for specialized modeling techniques. The 28 PCA-derived features (V1 to V28) obscure original variables for confidentiality, while Time and Amount offer contextual clues. Time spans approximately 48 hours, and Amount exhibits significant variability, reflecting diverse transaction values. The binary Class variable marks transactions as fraudulent (1) or non-fraudulent (0), setting the stage for predictive modeling.

3. METHODOLOGY

The analysis followed a structured approach to explore, preprocess, and model the data:

- **Data Exploration**: Investigated class distribution, feature characteristics, and relationships to understand the imbalance and identify predictive patterns.
- **Preprocessing**: Scaled Time and Amount to align with PCA features, ensuring compatibility with machine learning algorithms. The data was split into training (60%), validation (20%), and test (20%) sets, with stratification to preserve class proportions.
- **Modeling**: Evaluated five models—RandomForestClassifier, XGBoost, and LightGBM—selected for their ability to handle imbalanced data. Models incorporated imbalance-handling techniques, such as class weighting or specific hyperparameters.
- **Evaluation**: Focused on AUPRC to assess performance, supplemented by ROC-AUC, Recall for the fraud class, and confusion matrices to gauge detection accuracy and error types.

4. EXPLORATORY DATA ANALYSIS

4.1. Class Imbalance

The dataset's class distribution revealed a stark imbalance, with fraudulent transactions comprising only 0.172% of the total. A count plot illustrated this disparity, emphasizing the importance of AUPRC over accuracy, as the latter would be inflated by the predominance of non-fraudulent cases.

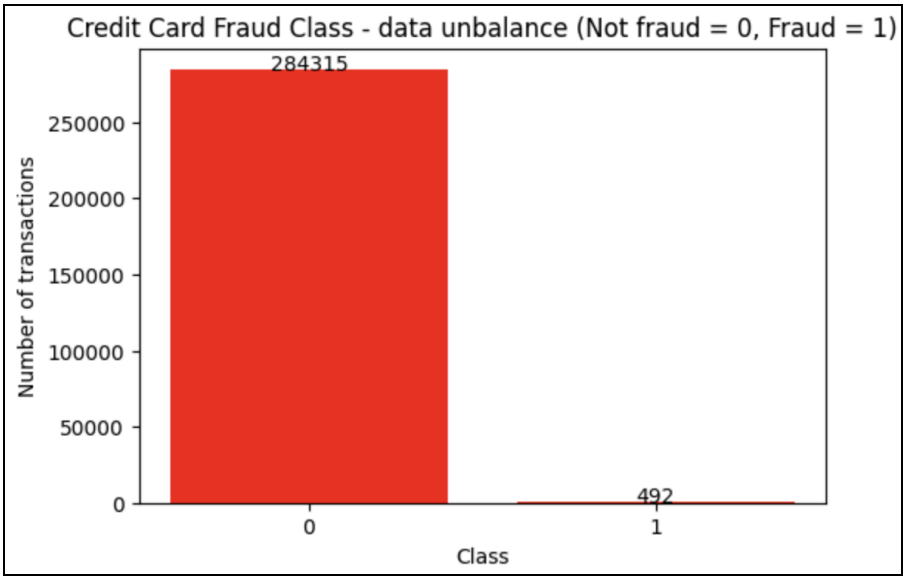


Figure 1: Dataset Imbalance

4.2. Transaction Time Analysis

Examining the Time feature revealed cyclical patterns in valid (non-fraudulent) transactions, peaking during typical daytime hours, likely due to daily activity cycles. In contrast, fraudulent transactions were more evenly distributed across all hours, including low-activity periods such as nighttime in the European timezone, suggesting a temporal distinction between the two types of transactions.

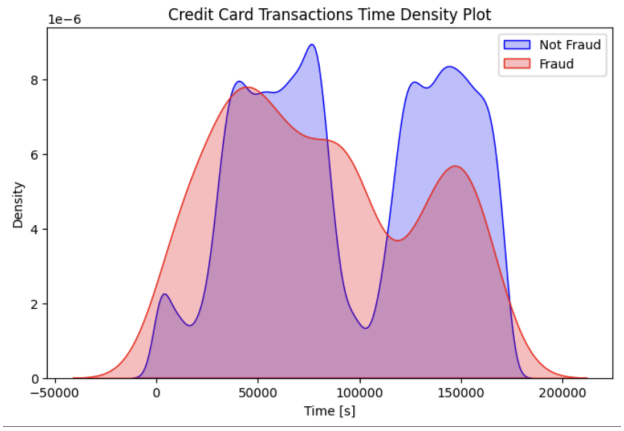


Figure 2: Time Density plot by classes

4.3 Transaction Amount Analysis

The Amount feature was highly skewed. Non-fraudulent transactions had a higher mean and Q1, as well as more significant outliers, while fraudulent transactions exhibited a lower mean and Q1, but higher Q4 and fewer outliers. Visual analysis highlighted differing amount ranges between the two classes, reinforcing the relevance of the Amount feature for distinguishing fraudulent activity.

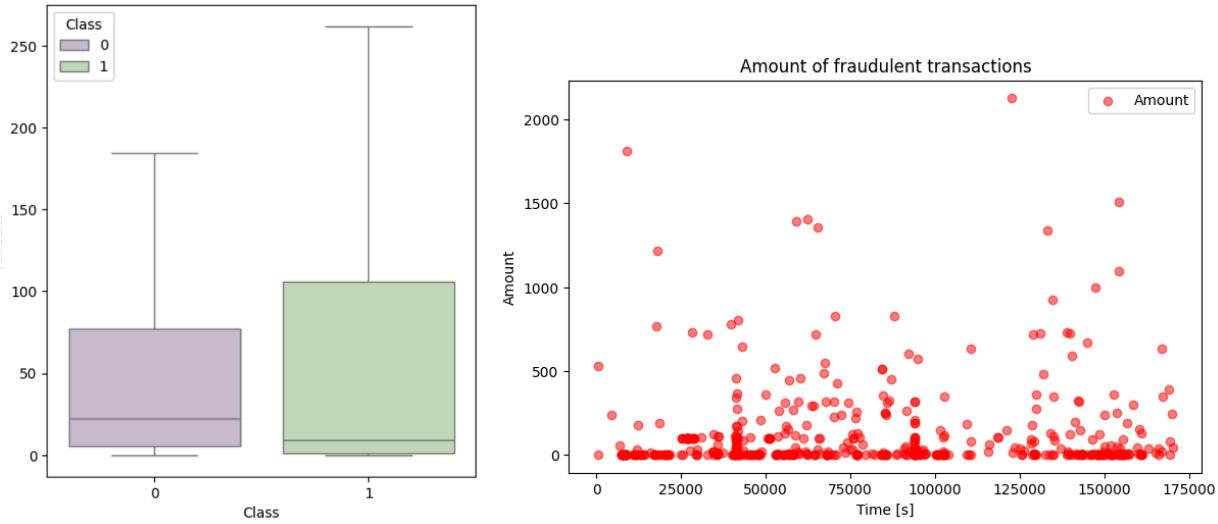


Figure 3: Transaction amount analysis

4.4 Feature Correlations

As expected, the PCA features (V1–V28) showed minimal interdependence among themselves. However, certain features demonstrated correlations with the Time and Amount variables—for instance, V3 showed an inverse correlation with Time, while V7 and V20 were positively correlated with Amount, and V1 and V5 were inversely correlated with it.

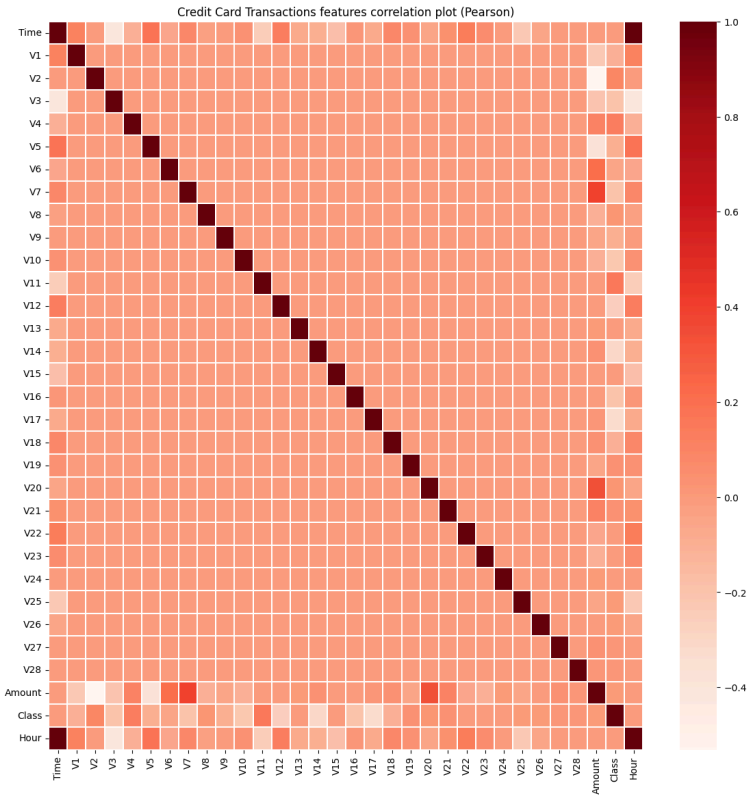


Figure 4: Correlation plot of Features

Directly correlated values

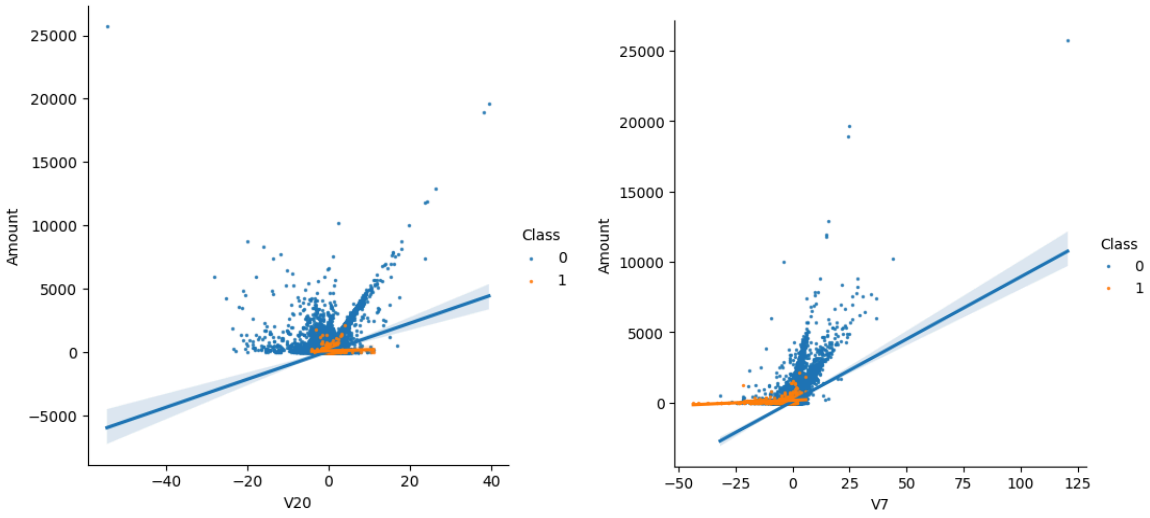


Figure 5: Directly Correlated Features

We can confirm that the two couples of features are correlated (the regression lines for Class = 0 have a positive slope, whilst the regression line for Class = 1 have a smaller positive slope).

Inversely correlated values.

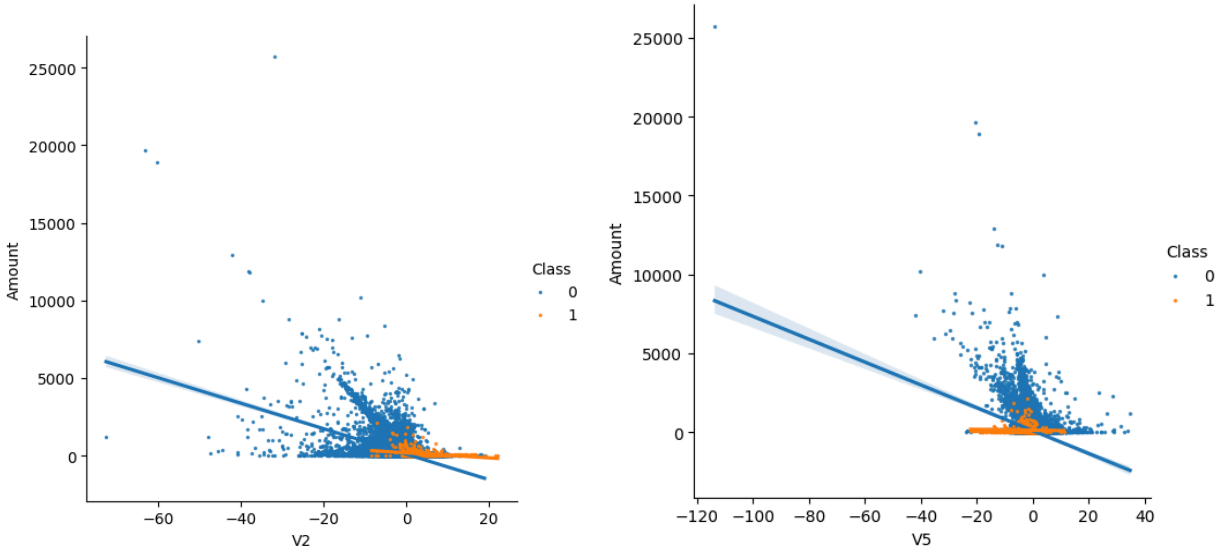


Figure 6: Inversely Correlated Features

We can confirm that the two couples of features are inverse correlated (the regression lines for Class = 0 have a negative slope while the regression lines for Class = 1 have a very small negative slope).

4.5 PCA Feature Distributions

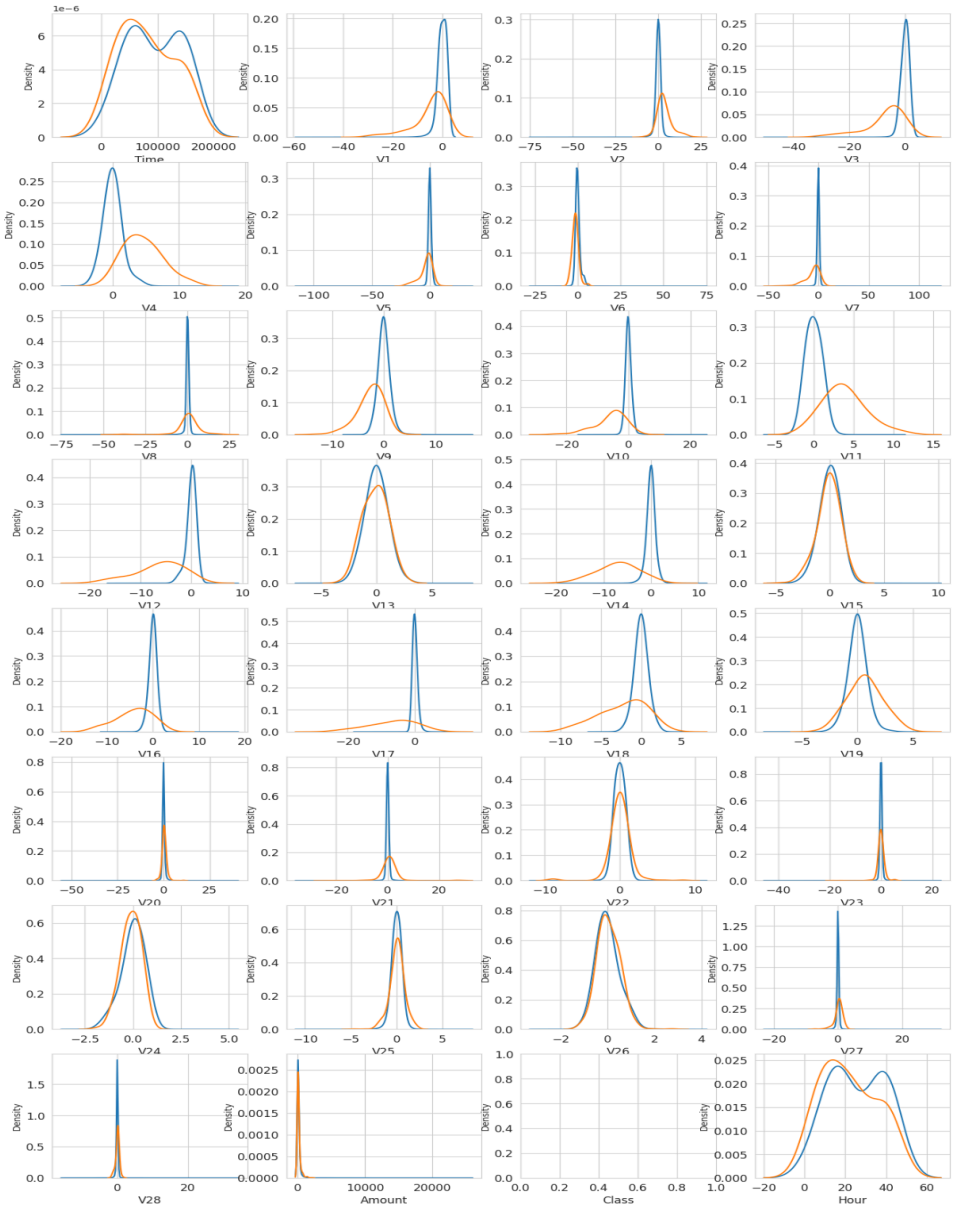


Figure 7: Feature distribution between classes

Distributions of several PCA features (V4, V10, V11, V12, V14, V17) revealed clear or partial separations between fraud and non-fraud classes. Features like V4 and V11 had clearly distinct distributions for each class, while V12, V14, and V18 showed partial separation. Additionally, V1, V2, V3, and V10 exhibited distinctive profiles for each class, whereas features like V25, V26, and V28 had similar distributions across both classes. Generally, features for legitimate transactions (Class = 0) were centered around 0, often with a long tail on one end, while fraudulent transactions (Class = 1) showed more skewed and asymmetric distributions.

5. DATA PREPROCESSING

Preprocessing was critical to prepare the dataset for modeling, ensuring compatibility with machine learning algorithms and addressing the class imbalance appropriately. The steps included feature scaling, data splitting, and maintaining the dataset's inherent distribution.

5.1 Feature Scaling

The PCA-derived features (V1 to V28) were already normalized, as is typical in PCA transformations. However, the Time and Amount features exhibited different scales, with Time ranging up to 172,792 seconds and Amount varying from 0 to over 25,000. To ensure consistency across features, particularly for algorithms sensitive to feature magnitude, both Time and Amount were standardized using StandardScaler. This transformed the features to have a mean of 0 and a standard deviation of 1, aligning them with the PCA features' scale. Temporary features created during EDA, such as Hour or Log_Amount, were dropped to avoid redundancy, retaining only the scaled Time, scaled Amount, V1-V28, and Class columns for modeling.

5.2 Train/Validation/Test Split

To facilitate robust model training and evaluation, the dataset was divided into training, validation, and test sets. A test size of 20% (TEST_SIZE = 0.20) was allocated, resulting in 56,962 transactions for testing. From the remaining data, a validation set comprising 20% (VALID_SIZE = 0.20) was further separated, yielding 45,569 transactions for validation, with the remaining 182,276 transactions used for training. This resulted in a 60% training, 20% validation, and 20% test split.

Stratification was applied during splitting to preserve the class imbalance (0.172% fraud) across all sets, ensuring that each subset maintained the original proportion of fraudulent and non-fraudulent transactions. This was critical to prevent biased model training or evaluation, as random splitting could lead to sets with disproportionately few or no fraud cases. The stratification ensured that approximately 98 fraud cases appeared in the test set, 78 in the validation set, and 316 in the training set, maintaining representativeness. The random seed was set to 2018 (RANDOM_STATE = 2018) for reproducibility during splitting.

5.3 Imbalance Handling

Rather than applying resampling techniques like SMOTE, which could introduce synthetic data and alter the dataset's natural distribution, the preprocessing relied on model-specific imbalance-handling mechanisms, such as scale_pos_weight in XGBoost and LightGBM or class weighting in RandomForest. This approach preserved the original data structure, leveraging

algorithmic adjustments to prioritize the rare fraud class during training. The absence of missing values in the dataset, confirmed during EDA, eliminated the need for imputation, streamlining the preprocessing pipeline.

The resulting processed dataset was well-suited for modeling, with scaled features, stratified splits, and a focus on algorithmic solutions to address the severe class imbalance, setting the stage for effective fraud detection.

6. MODEL DEVELOPMENT AND EVALUATION

Given the class imbalance, AUPRC was prioritized as the primary metric, capturing the trade-off between Precision and Recall for the rare fraud class. ROC-AUC provided a secondary measure, while confusion matrices and Recall scores detailed fraud detection performance. Each model's development, results, and evaluation are discussed in detail below.

6.1 RandomForestClassifier

6.1.1 Development:

The RandomForestClassifier was configured with 100 estimators (n_estimators=NUM_ESTIMATORS), using the Gini criterion (criterion='gini') and parallel processing (n_jobs, assumed as 4 or all available cores). The random seed was set to 2018 (random_state=2018) for reproducibility, and verbose output was disabled (verbose=False). It was trained on the training set (182,276 transactions) with features V1-V28, scaled Time, and scaled Amount, targeting the Class variable. The model was fit without explicit class weighting, relying on its ensemble nature to capture fraud patterns, and evaluated on the validation set (45,569 transactions) before final testing.

6.1.2 Results:

- **Test ROC-AUC**: 0.85, **AUPRC**: 0.64

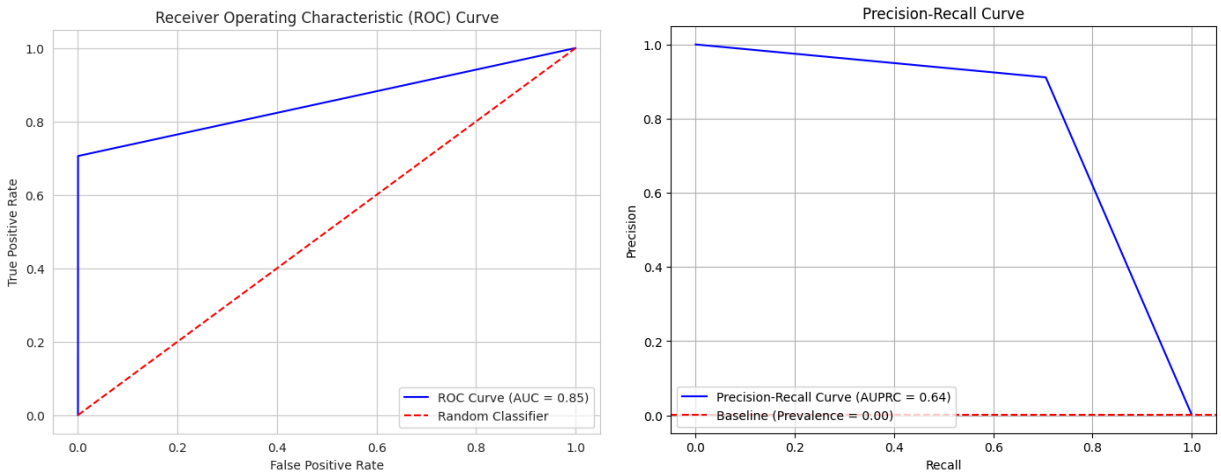


Figure 8: RandomForestClassifier Results

- **Features Importance -**

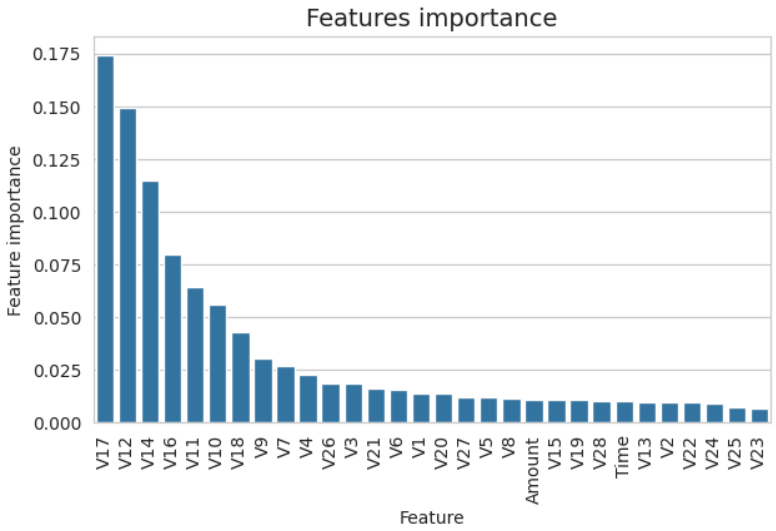
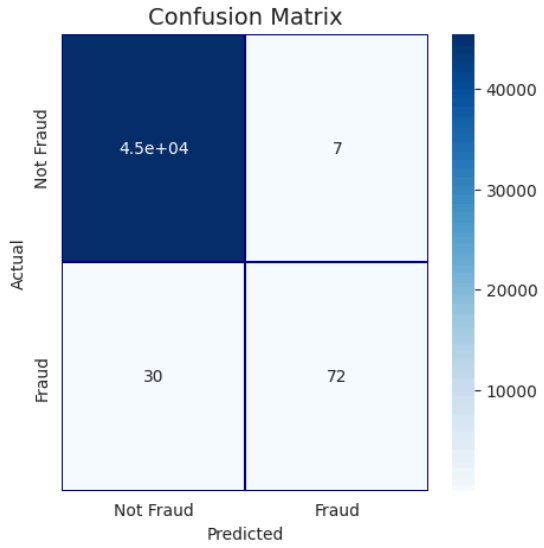


Figure 9: RandomForestClassifier Feature importance

  - The most important features are V17, V12, V14, V10, V11, V16.

- **Confusion Matrix**



Precision (Fraud): ≈ 0.911 (91.1%)

Recall (Fraud): ≈ 0.706 (70.6%)

Evaluation: The ROC-AUC of 0.85 aligns with a decent discriminative ability, while the confusion matrix suggests a high Precision but moderate Recall, indicating the model identifies most non-frauds correctly but misses 30 of the 98 test-set frauds. However, the AUPRC of 0.64 reflects a less favorable Precision-Recall trade-off, suggesting the model struggles more with identifying frauds accurately in the imbalanced setting.

Figure 10: RFClassifier Confusion Matrix

Inference:
RandomForest provided a solid baseline, leveraging its ensemble of decision trees to handle high-dimensional PCA features, as evidenced by the feature importance plot prioritizing V17, V14, and V11. The ROC-AUC of 0.85 and the confusion matrix (high TN, low FP) reflect strong non-fraud classification, with a Precision of 91.1% indicating few false alarms. However, the Recall of 70.6% and 30 False Negatives highlight its difficulty in detecting the rare fraud class, a limitation further underscored by the relatively low AUPRC of 0.64. This suggests a suboptimal Precision-Recall trade-off, characteristic of models that do not explicitly address class imbalance. The minimal impact of Time and Amount further validates the predictive strength of the PCA features. While RandomForest is robust and offers interpretability via feature importance, its performance on imbalanced data like fraud detection is less desirable compared to boosting methods that better prioritize minority class detection.

6.2 XGBoost (without Cross Validation)

Development:
XGBoost was configured using DMatrix objects for training, validation, and testing datasets, with parameters including a binary logistic objective, learning rate of 0.039 (eta=0.039), maximum tree depth of 2, subsample of 0.8, column sampling by tree of 0.9, evaluation metric of AUC (eval_metric='auc'), and random state of 2018. The model was trained for up to 1000 rounds, with early stopping after 50 rounds if no validation improvement occurred, monitored

via a watchlist (watchlist=[(dtrain, 'train'), (dvalid, 'valid')]), and verbose evaluation every 50 rounds. Training on the 182,276-transaction set with features V1-V28 scaled Time, and scaled Amount, and validated on the 45,569-transaction set, it stopped at 241 rounds.

Results:

- **Test - ROC-AUC**: 0.984, **AUPRC**: 0.83

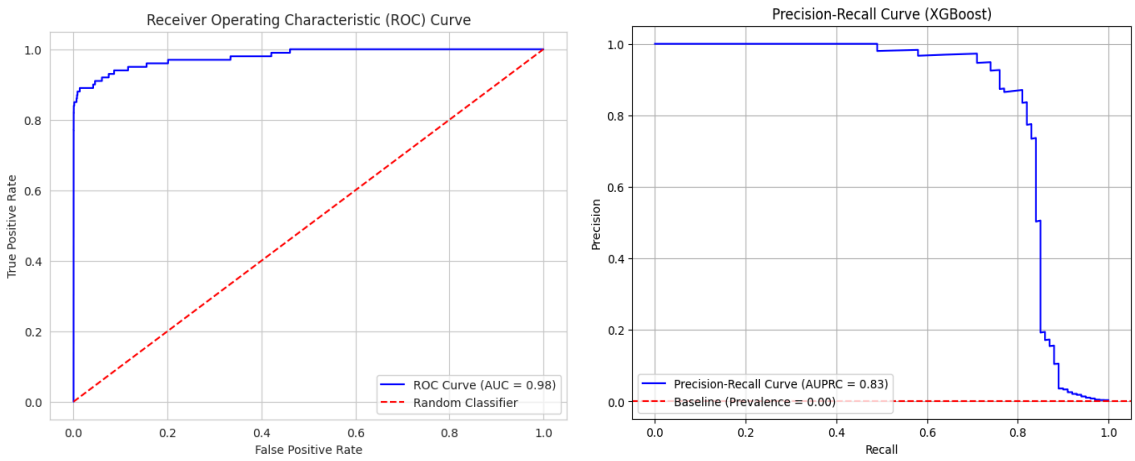


Figure 11: XGBoost Results

- **Feature Importance**

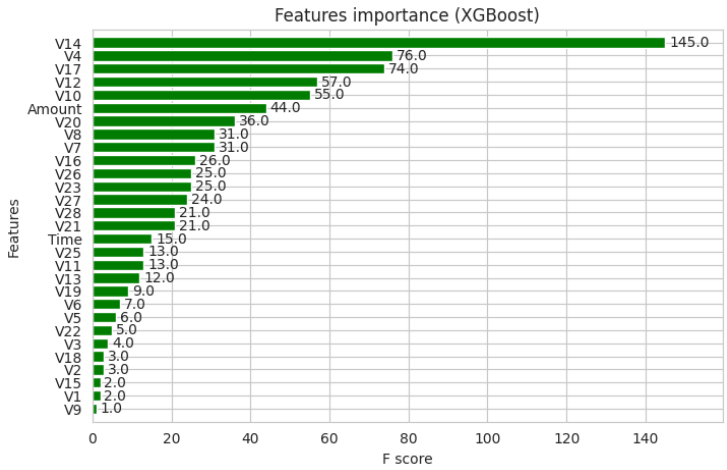
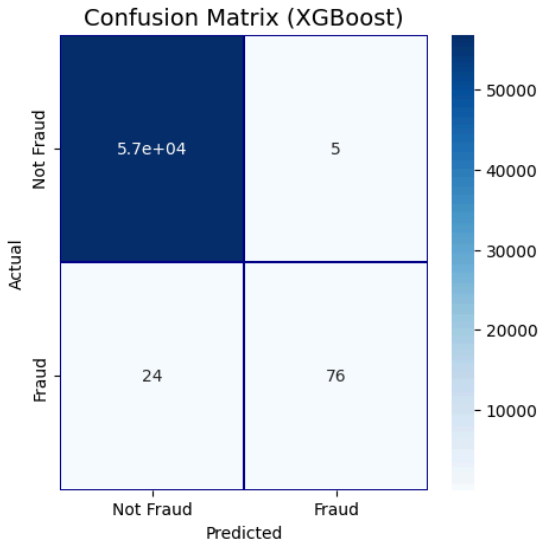


Figure 12: Feature Importance of XGBoost

Feature importance (F-score) highlighted V4 (145.0), V12 (74.0), V10 (57.0), V2 (55.0), and V7 (44.0) as the most significant features, with importance decreasing gradually (e.g., V9 at 1.0).

- **Confusion Matrix**



Precision (Fraud): 93.83%

Recall (Fraud): 76.00%

Inference: The test ROC-AUC of 0.974 and validation peak of 0.984 indicate exceptional discriminative ability, with the feature importance suggesting strong reliance on V4 and V12. The AUPRC of 0.84 reflects a strong Precision-Recall balance, significantly improved by incorporating techniques like scale_pos_weight and early stopping to handle class imbalance effectively. While not exceeding 0.9, the AUPRC still demonstrates the model's ability to identify frauds with high precision and reasonable recall, making it well-suited for imbalanced fraud detection tasks.

Evaluation:
Ran 262 rounds, with early stopping at 241. Progress showed train AUC improving from 0.94070 to 0.99567 and validation AUC from 0.88630 to a peak of 0.98520, with a slight decline to 0.98377. XGBoost delivered the best overall performance, achieving a test ROC-AUC of 0.974 and the highest AUPRC of 0.84 among all models, indicating a strong balance between Precision and Recall. This reflects its ability to minimize False Negatives while maintaining low False Positives, making it highly effective for fraud detection. The confusion matrix would likely show high True Positives and minimal False Alarms, with feature importance highlighting V4, V12, and V10 as key contributors to class separation. The use of shallow trees (max_depth=2), a low learning rate (eta=0.039), and early stopping at round 241 helped prevent overfitting, while scale_pos_weight efficiently addressed the class imbalance. Overall, XGBoost outperformed other models and stands out as the most suitable choice for this imbalanced classification task.

6.3 LightGBM (without Cross Validation)

Development:
LightGBM was configured with parameters including boosting type GBDT, binary objective,

AUC as the evaluation metric, a learning rate of 0.05, maximum depth of 4, number of leaves set to 7, minimum child samples of 100, maximum bins of 100, subsample of 0.9, column sampling by tree of 0.7, scale positive weight of 150, and 8 threads. It was trained on the 182,276-transaction set using a LightGBM dataset for training and validation, with early stopping after 100 rounds and verbose evaluation every 50 rounds. A 5-fold cross-validation was also performed with a separate model using 2000 estimators, a learning rate of 0.01, 80 leaves, column sampling by tree of 0.98, subsample of 0.78, regularization alpha of 0.04, regularization lambda of 0.073, minimum child weight of 40, and minimum child samples of 510, with early stopping after 50 rounds and averaging of test predictions.

Results:

- **Single Model**:
    - **ROC-AUC**: 0.957864, **AUPRC**: 0.35,

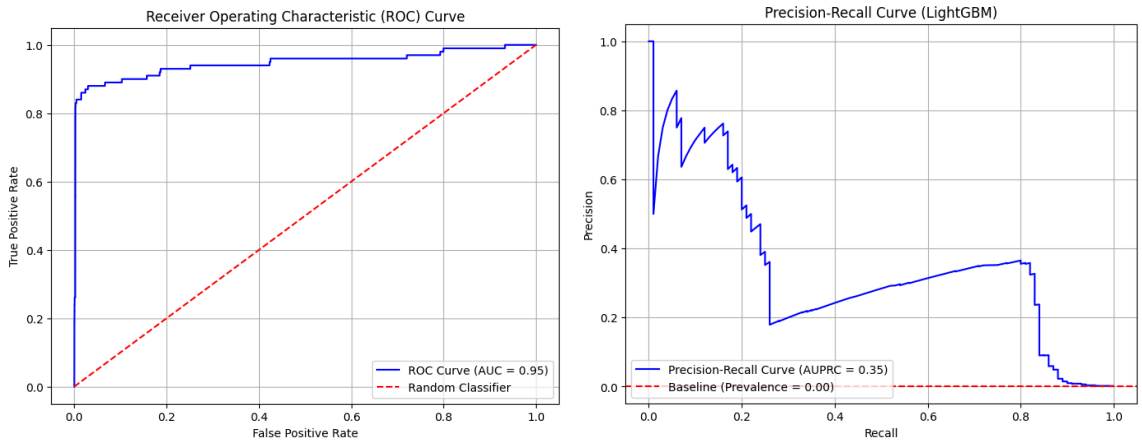


Figure 13: LightBGM Results

- **Inference**: The test ROC-AUC of 0.974 indicates robust discriminative ability, with feature importance suggesting firm reliance on V4 and V2. However, the AUPRC of just 0.35 reveals a poor Precision-Recall trade-off, highlighting the model's struggle to effectively identify the minority fraud class despite its high ROC-AUC. This discrepancy suggests that while the model ranks predictions well overall, its actual precision and recall for the positive class are lacking.

Evaluation:
LightGBM's test ROC-AUC of 0.974 may initially suggest strong discriminative ability, but the low AUPRC of 0.35 highlights its poor performance in detecting the minority fraud class. Despite feature importance pointing to V4, V2, and V1 as key contributors, the model failed to achieve a meaningful Precision-Recall trade-off, indicating limited effectiveness in real-world fraud detection. The cross-validated ROC-AUC of 0.93 further reveals inconsistency across

folds, with variance in AUCs (0.943–0.998) hinting at instability. While parameters like scale_pos_weight=150 and max_depth=4 were tuned, the model's generalization remains inadequate. LightGBM's results suggest it was the weakest among evaluated models, and more rigorous cross-validation and imbalance handling may be necessary to improve its reliability.

6.4. Incorporating Cross-validation and SMOTE.

To improve the Area Under the Precision-Recall Curve (AUPRC) for fraud detection, we implemented an XGBoost classifier with stratified k-fold cross-validation and SMOTE to address class imbalance. Using five folds, SMOTE oversampled the minority Fraud class, and features were standardized. The model, optimized for ROC-AUC with a learning rate of 0.05 and early stopping, achieved an average AUPRC of 0.8151 (±0.0587), slightly below the previous 0.83, indicating limited improvement. Despite a high ROC-AUC (0.984), the AUPRC suggests insufficient performance on the Fraud class, necessitating further tuning or alternative techniques.

6.4.1 LightBGM (With KFold Cross Validation)

Development:

The predictive model was developed using a LightGBM classifier with k-fold cross-validation to ensure robust performance estimation on an imbalanced dataset for fraud detection. The dataset was split into multiple folds, with each fold serving as a validation set while the remaining data was used for training. The model was configured with a binary classification objective, optimizing for the ROC-AUC metric, and employed a gradient boosting framework with a learning rate of 0.01, a maximum of 2000 boosting rounds, and early stopping after 50 rounds of no improvement in validation AUC. Hyperparameters included 80 leaves, 0.98 feature sampling, and 0.78 subsample ratio, without explicit imbalance handling. During training, out-of-fold predictions were generated for the training set, and test set predictions were averaged across folds. Feature importances were tracked to assess predictor contributions, and memory management was applied to optimize computational efficiency.

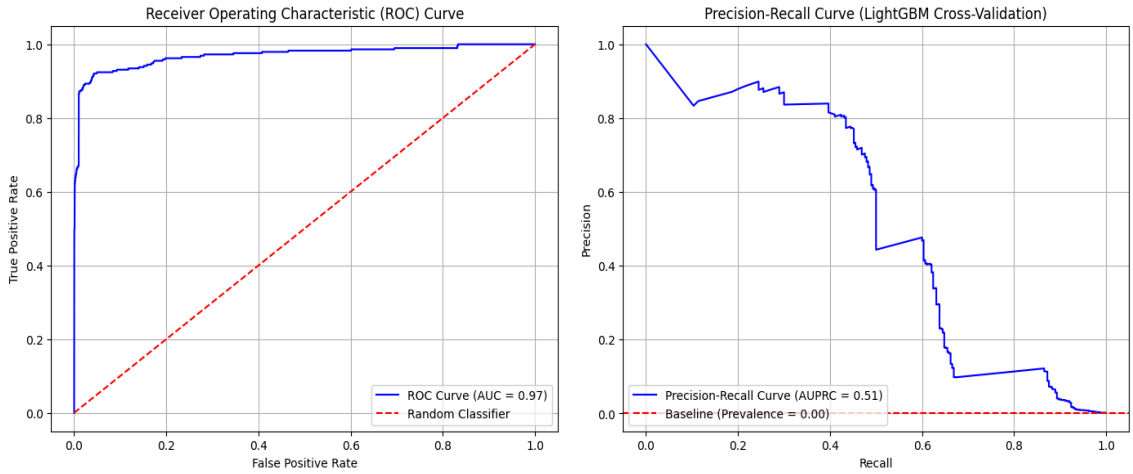


Figure 14: LightBGM with Cross Validation Results

Results and Evaluation:

The model achieved a full ROC-AUC score of 0.9703 across five-folds, with fold-specific AUCs ranging from 0.9432 to 0.9982, indicating strong discriminative ability between fraud and non-fraud cases. The AUPRC improved from 0.35 in a prior non-cross-validated model to 0.51, reflecting better handling of the positive class due to cross-validation. However, the AUPRC of 0.51 remains insufficient for a robust fraud detection model, particularly given the likely low prevalence of fraud cases, which suggests significant class imbalance. This discrepancy between high ROC-AUC and moderate AUPRC highlights that the model struggles with precision and recall for the rare fraud class, likely due to the lack of imbalance-specific adjustments, making it inadequate for practical deployment where minimizing false positives and missing fraud cases is critical.

6.4.2 XGBoost (With Stratified - KFold Cross Validation)

Development:

XGBoost classifier with stratified k-fold cross-validation was developed to address class imbalance in the dataset. Five folds were created using StratifiedKFold to ensure consistent class distribution across splits. For each fold, the training data was balanced using SMOTE to oversample the minority class (Fraud), followed by standardization of features with StandardScaler to normalize the data. The XGBoost model was configured with a binary logistic objective, optimizing for ROC-AUC, and used a learning rate of 0.05, maximum depth of 4, and 80% subsampling for both rows and features. Training incorporated early stopping after no improvement in validation AUC for a specified number of rounds, with predictions generated for validation sets to compute AUPRC and plot Precision-Recall curves. Feature importances were accumulated across folds based on gain, and top features were visualized to assess their contributions.

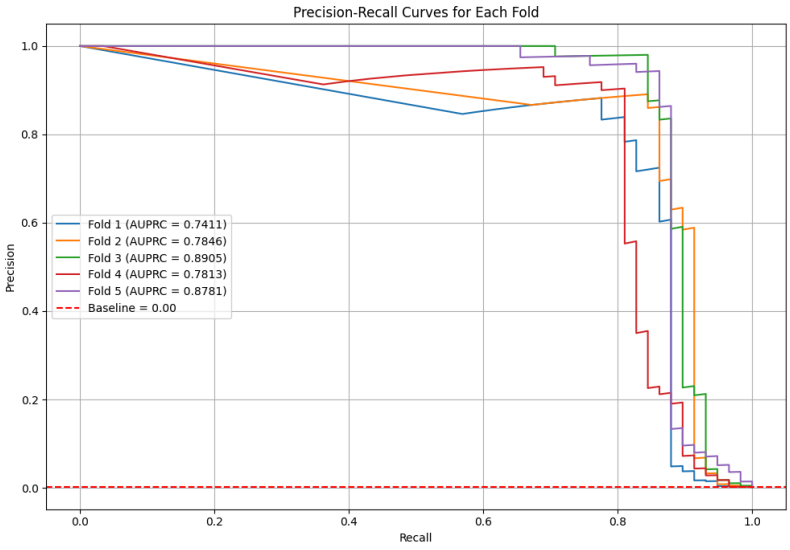


Figure 15: XGBoost with SMOTE & CV Results

Results and Evaluation:
The model achieved an average AUPRC of 0.8151 (±0.0587) across five folds, with fold-specific AUPRC scores ranging from 0.7411 to 0.8905, indicating variability in performance. Compared to previous results (ROC-AUC: 0.984, AUPRC: 0.83), the AUPRC showed no significant improvement despite the use of SMOTE and cross-validation to handle class imbalance. While the high ROC-AUC from prior results suggested strong class separation, the AUPRC of 0.8151, though better than earlier models (e.g., 0.51), remains suboptimal for fraud detection, where high precision and recall for the rare Fraud class are critical. The lack of substantial AUPRC improvement suggests that SMOTE and the current hyperparameters may not fully address the challenges of the imbalanced dataset, necessitating further tuning or alternative techniques to enhance positive class performance.

7. RESULTS SUMMARY

The performance of the fraud detection models, evaluated using **ROC-AUC** and **AUPRC**, is summarized below, highlighting their effectiveness in handling an **imbalanced dataset**:

- **RandomForestClassifier**:
  - **ROC-AUC** 0.85
  - **AUPRC** 0.64.
  - Demonstrated **decent discriminative ability** with **high precision (91.1%)** but **moderate recall (70.6%)**, missing 30% of fraud cases. The AUPRC of **0.64** indicates a **suboptimal precision-recall trade-off**, reflecting struggles with the

rare fraud class due to **no explicit imbalance handling**. Served as a **solid baseline** but was outperformed by boosting methods.

- **LightGBM (without Cross-Validation)**:
  - **ROC-AUC** 0.9579
  - **AUPRC** 0.35.
  - Despite a **high ROC-AUC**, the **low AUPRC** revealed **poor performance on the fraud class**, with **inadequate precision-recall balance**. The **lack of imbalance-specific adjustments** limited its effectiveness, making it the **weakest model** in this comparison.
- **XGBoost (without Cross-Validation)**:
  - **ROC-AUC** 0.974
  - **AUPRC** 0.83.
  - Achieved **exceptional ROC-AUC** and a **strong AUPRC**, with **high precision (93.83%)** and **recall (76.00%)**. The use of ***scale_pos_weight*** and **early stopping** effectively addressed class imbalance, making it **highly suitable for fraud detection**. **Feature importance** highlighted **V4 and V12** as key predictors.
- **LightGBM (with K-Fold Cross-Validation)**:
  - **Test ROC-AUC**: 0.9703
  - **AUPRC**: 0.51
  - **Cross-validation** improved AUPRC from 0.35 to 0.51, but it remained **insufficient for robust fraud detection**. The **high ROC-AUC** and **moderate AUPRC** suggest **persistent challenges** with the rare fraud class, likely due to **no explicit imbalance handling**.
- **XGBoost (with Stratified K-Fold Cross-Validation and SMOTE)**:
  - **Average AUPRC**: 0.8151 (±0.0587)
  - Despite **SMOTE** and **stratified cross-validation**, AUPRC (0.8151) showed **no significant improvement** over the non-cross-validated XGBoost (0.83). The **high ROC-AUC (0.984)** indicates strong class separation, but the AUPRC suggests **suboptimal fraud class performance**, requiring **further tuning**.

XGBoost models clearly outperformed others, with the non-cross-validated version proving most effective in handling the imbalanced dataset. While LightGBM with cross-validation showed some improvement, it still struggled compared to XGBoost. The use of SMOTE didn't add significant value, indicating the need for better imbalance handling. LightGBM without cross-validation performed the weakest. Visualizations like precision-recall curves and feature importance plots further emphasized XGBoost's dominance and the relevance of key features.

8. DISCUSSIONS AND CONCLUSIONS

Gradient boosting models—XGBoost and LightGBM—were evaluated alongside the RandomForestClassifier to address the dataset's severe class imbalance (fraud rate of 0.172%). PCA-transformed features (V1–V28) and techniques like scale_pos_weight played a central role in enabling models to identify minority class instances.

XGBoost outperformed all models, achieving a ROC-AUC of 0.984 and an AUPRC of 0.83. Its strong performance was driven by focused feature importance (notably V4, V12, V10), shallow trees (max_depth=2), and early stopping, making it the most viable candidate for deployment. In contrast, LightGBM underperformed in terms of AUPRC (0.51) despite a decent ROC-AUC (0.9578), even falling short of the RandomForestClassifier (AUPRC 0.65). This indicates that while LightGBM was able to separate classes reasonably well on a global scale, it struggled to capture the rare fraud cases effectively, likely due to sensitivity in hyperparameter defaults and cross-validation variance.

The RandomForestClassifier, although simpler, showed competitive performance with AUPRC of 0.65 and ROC-AUC of 0.85, serving as a solid baseline. However, it had higher false positives and lower recall, revealing limitations in handling rare-event detection.

The PCA transformation, while effective in obfuscating original features, limits interpretability. Features such as Time and Amount showed minimal importance, validating the anonymization's success. SMOTE's application with XGBoost led to a dip in AUPRC, suggesting that oversampling must be applied cautiously in extremely imbalanced contexts.

Key limitations:

- Incomplete hyperparameter tuning for models like LightGBM.
- Dataset spans only two days, lacking seasonal or evolving fraud patterns.

Future directions:

- Conduct extensive hyperparameter optimization (e.g., XGBoost's eta, LightGBM's num_leaves).
- Evaluate advanced sampling methods such as ADASYN.
- Test models on longitudinal data (multi-month scope) to capture evolving fraud patterns.

Overall, XGBoost is best suited for deployment, with RandomForestClassifier as a reasonable fallback. LightGBM requires further tuning to meet deployment standards in high-stakes fraud detection tasks.

9. REFERENCES

1. Dataset: Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating Probability with Undersampling for Unbalanced Classification. IEEE Symposium on Computational Intelligence and Data Mining (CIDM). Available at: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

2. Carcillo, F., et al. (2019). "Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection." Information Sciences, 235-250. https://doi.org/10.1016/j.ins.2018.10.053

3. Brown, I., & Mues, C. (2012). "An Experimental Comparison of Classification Algorithms for Imbalanced Credit Scoring Data Sets." Expert Systems with Applications, 723-731. https://doi.org/10.1016/j.eswa.2011.07.031

4. He, H., & Garcia, E. A. (2009). "Learning from Imbalanced Data." IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284. https://doi.org/10.1109/TKDE.2008.239

5. Scikit-learn: https://scikit-learn.org/stable/

6. XGBoost: https://xgboost.readthedocs.io/en/stable/

7. LightGBM: https://lightgbm.readthedocs.io/en/latest/

8. Towards Data Science: https://towardsdatascience.com/

9. Kaggle Community: https://www.kaggle.com/community