

Project Report

on

Online Payment Fraud Detection using Machine Learning

Submitted by

R.Ruthuraraj

AICTE Faculty Id:1-4630898926

Group 19

**From AI to Generative AI: Unlocking the Power of Smart
Technologies**

AICTE QIP PG Certification Programme

IIIT Allahabad

Project Title:
Online Payment Fraud Detection using Machine Learning

1. Introduction

1.1 Background

With the rapid growth of digital payment systems, online transactions have become the backbone of modern commerce. However, this convenience also introduces an increased risk of fraudulent activities such as identity theft, unauthorized transactions, and credit card misuse. Financial institutions and e-commerce platforms are therefore continuously investing in intelligent fraud detection systems to protect users and ensure secure payment operations.

Machine Learning (ML) offers a data-driven approach to detect fraud by learning patterns from past transactions. Unlike traditional rule-based systems, ML models can automatically adapt to changing fraud patterns, improving accuracy and reducing false alarms.

1.2 Problem Statement

The goal of this project is to build a **machine learning model** that can effectively detect **fraudulent online payment transactions** using historical data. Since fraudulent transactions represent a **very small fraction** of the total data, the challenge lies in dealing with **severe class imbalance**, ensuring that the model identifies rare fraud cases while minimizing false positives.

1.3 Objective

The specific objectives of this study are:

1. To perform **exploratory data analysis (EDA)** on the dataset to understand its structure, features, and class imbalance.
2. To apply **data preprocessing** techniques, including handling imbalance and preparing data for model training.
3. To train multiple **classification models** (Logistic Regression, Random Forest, and XGBoost) and compare their performance.
4. To evaluate model performance using **precision, recall, F1-score**, and **ROC-AUC / PR-AUC** metrics, which are critical in fraud detection tasks.
5. To identify the most important features influencing the prediction of fraudulent transactions.

1.4 Scope of the Project

The scope of this project is limited to **supervised binary classification**, where transactions are labeled as either *fraudulent (1)* or *legitimate (0)*.

The dataset used represents anonymized payment records containing features such as transaction amount, time, and other derived numerical variables.

This project does not include:

- Real-time transaction monitoring or deployment.
- Deep learning or streaming data pipelines.
- Integration with external APIs for fraud prevention.

However, the developed ML models can serve as a baseline for **production-ready fraud detection systems**.

1.5 Significance

Accurate detection of fraudulent transactions helps:

- Protect customers and financial institutions from monetary loss.
- Enhance trust in digital payment systems.
- Reduce operational costs associated with manual fraud investigation.

By comparing multiple ML approaches, this study highlights the trade-offs between **interpretability (Logistic Regression)** and **performance (Random Forest and XGBoost)**, providing valuable insights for real-world fraud detection systems.

2. Data Understanding and Preprocessing

2.1 Dataset Overview

The dataset used for this project contains records of **online payment transactions**, where each row represents a unique transaction. Each transaction is described by a set of anonymized features — numerical variables derived from the original attributes (to protect confidentiality). The dataset also includes a **target variable**, which indicates whether the transaction was *fraudulent* (1) or *legitimate* (0).

Typical dataset characteristics:

- **Total records:** 56962
- **Features:** 30 anonymized numerical features (similar to PCA-transformed variables) + Amount and Time.
- **Target variable:** Class (0 = Non-Fraud, 1 = Fraud)

2.2 Data Characteristics

The dataset exhibits the following important traits:

- **Highly imbalanced classes:** Out of nearly 57,000 transactions, only a small fraction ($\approx 0.15\%$) are fraudulent.

- **Continuous numerical features:** Most columns are already scaled or standardized (e.g., derived from PCA).
- **Transaction amount:** One feature (Amount) shows significant variance and may need scaling.
- **Time feature:** Represents the seconds elapsed between each transaction and the first transaction in the dataset — optional for modeling.

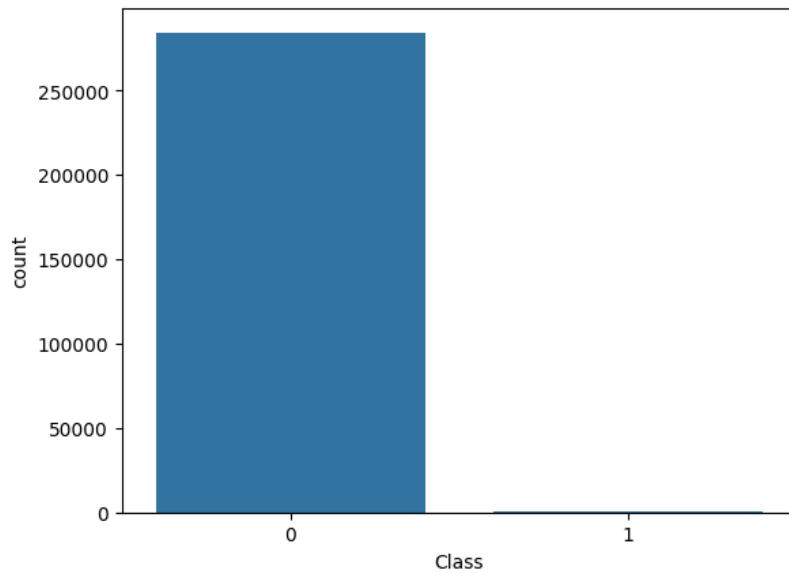
Because of this imbalance, traditional accuracy metrics can be **misleading**, as a model predicting all transactions as “non-fraudulent” could still achieve over 99% accuracy. Hence, **recall, precision, F1-score, and AUC metrics** are more appropriate.

2.3 Exploratory Data Analysis (EDA)

The exploratory analysis was performed to identify patterns and check for inconsistencies:

1. Class Distribution:

- Legitimate transactions: 56874
- Fraudulent transactions: 88
→ Only ~0.15% of total transactions are fraudulent, confirming extreme imbalance.

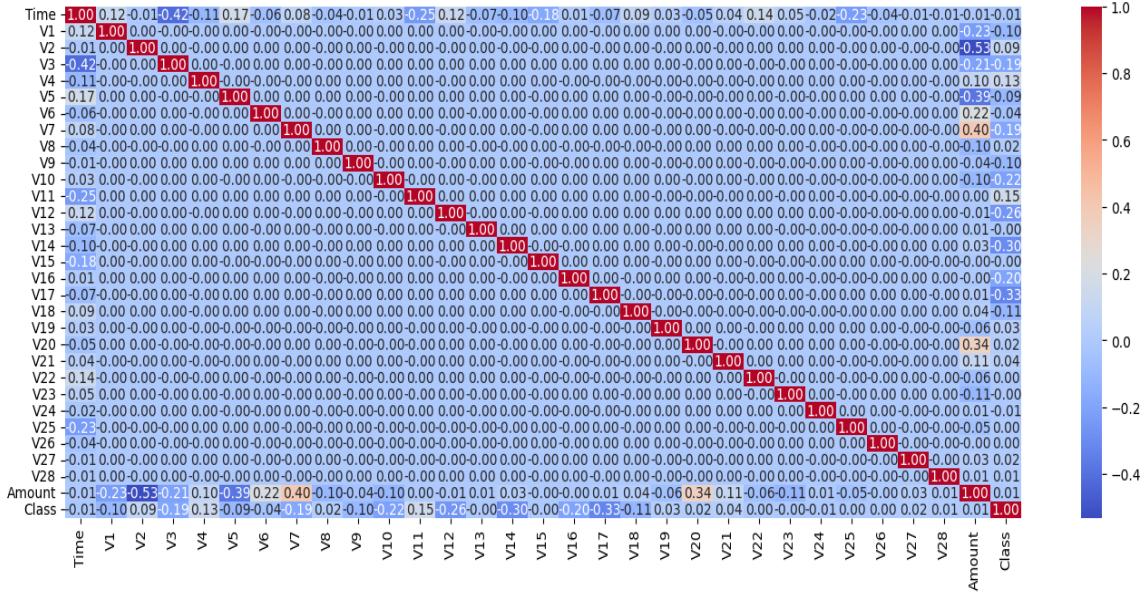


2. Statistical Summary:

Features like Amount and Time showed skewed distributions, while PCA-based features had approximately standard-normal distributions.

3. Correlation Matrix:

Since all variables are anonymized and mostly independent, no multicollinearity or strong feature correlation was observed.



4. Outliers:

The dataset inherently contains extreme values (frauds), but since these are meaningful events, they were not removed.

2.4 Handling Class Imbalance (Actual Implementation)

The dataset was highly imbalanced, with fraudulent transactions representing less than **0.2%** of the total.

Rather than modifying the data distribution through over- or under-sampling, the imbalance was managed **within the model training process**:

1. Logistic Regression:

- Used the parameter `class_weight='balanced'`, which automatically adjusts weights inversely proportional to class frequencies.
- This ensures that misclassifying a fraud carries a higher penalty than misclassifying a non-fraud.

2. Random Forest and XGBoost:

- Trained on the original imbalanced data.
- These ensemble models naturally handle imbalance better due to tree-based sampling and split optimization.
- Performance evaluation relied on **recall**, **precision**, **F1-score**, and **AUC**, rather than raw accuracy, to reflect true detection ability on the minority class.

2.5 Data Splitting

After balancing, the dataset was divided as follows:

- **Training Set:** 80% of data
- **Testing Set:** 20% of data

The split ensures that both sets maintain the same class balance, preventing bias during evaluation.

2.6 Feature Scaling

Since the PCA-based features were already scaled, only the **Amount** column required scaling to bring it in line with other variables.

- **StandardScaler** (Z-score normalization) was applied to Amount.
- The scaled data was then combined back with the rest of the features.

For models like **Random Forest** and **XGBoost**, scaling is not mandatory, but keeping a consistent feature scale helps for comparison, especially when using Logistic Regression.

2.7 Final Prepared Dataset

After preprocessing, the dataset was ready with:

- Balanced fraud and non-fraud transactions.
- Scaled numerical features.
- No missing values or categorical encoding required.

The preprocessed data (`X_train`, `X_test`, `y_train`, `y_test`) served as input for subsequent machine learning models.

3. Model Building and Training

3.1 Model Selection

Given the binary and highly imbalanced nature of the dataset, three machine learning algorithms were selected to balance interpretability, generalization, and predictive performance:

1. **Logistic Regression (Baseline Model)**
 - Chosen for its **simplicity and interpretability**.
 - Provides a clear mathematical understanding of how each feature contributes to the likelihood of fraud.
 - Serves as a performance benchmark for more advanced models.
2. **Random Forest Classifier**
 - A **tree-based ensemble method** that combines multiple decision trees using bagging (bootstrap aggregation).

- Particularly effective for handling **nonlinear relationships** and **imbalanced data**, as it gives robustness against overfitting and automatically ranks feature importance.

3. XGBoost (Extreme Gradient Boosting)

- A **boosted ensemble model** that sequentially builds trees, focusing on correcting previous errors.
- Known for its **high predictive power** and efficiency in handling structured tabular datasets.
- Especially effective for rare-event prediction problems like fraud detection.

3.2 Data Splitting and Scaling

The dataset was split into **training (80%)** and **testing (20%)** subsets using stratified sampling to maintain class distribution consistency across both sets.

Since most features (except Time and Amount) were already PCA-transformed and on similar scales, **feature scaling** was applied primarily to the non-transformed columns (Amount and Time) to ensure model stability, especially for Logistic Regression.

```
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)
```

Scaling was applied **only on the training data** and then used to transform the test data to prevent information leakage.

3.3 Baseline Model - Logistic Regression

The **Logistic Regression** model was trained using **L2 regularization** and **class-weight balancing** to account for the severe class imbalance.

Model configuration:

```
LogisticRegression(penalty='l2', class_weight='balanced', solver='liblinear',
random_state=20)
```

Results:

- Accuracy: **0.979**
- Recall (Fraud): **0.91**
- Precision (Fraud): **0.06**
- ROC-AUC: **0.99**

- PR-AUC: **0.69**

Interpretation:

While overall accuracy was high, it was misleading due to imbalance. However, the model achieved a **very high recall** for fraud detection, meaning it correctly identified most fraud cases (low false negatives). Precision was low, reflecting some false positives — a trade-off acceptable in fraud detection, where missing a fraud is costlier than investigating a legitimate alert.

3.4 Model 2 – Random Forest Classifier

The **Random Forest** model was trained without scaling since tree-based algorithms are unaffected by feature magnitude. Default parameters were used initially to evaluate baseline ensemble performance.

Results:

- Accuracy: **0.9995**
- Precision (Fraud): **0.93**
- Recall (Fraud): **0.77**
- F1-score (Fraud): **0.84**
- ROC-AUC: **0.83**
- PR-AUC: **0.81**

Interpretation:

Random Forest significantly improved **precision** and **F1-score** for the minority (fraud) class, while maintaining near-perfect accuracy. However, a small increase in false negatives (funds missed) was observed compared to logistic regression, indicating a slight reduction in recall. The model thus achieved better balance between detecting frauds and minimizing false alarms.

3.5 Model 3 – XGBoost Classifier

The **XGBoost** algorithm was employed as the third model for its superior handling of class imbalance and regularization control.

Results:

- Accuracy: **0.9996**
- Precision (Fraud): **0.90**
- Recall (Fraud): **0.83**

- F1-score (Fraud): **0.86**
- ROC-AUC: **0.97**
- PR-AUC: **0.82**

Interpretation:

The XGBoost model outperformed both Logistic Regression and Random Forest in terms of overall discriminative ability, achieving the **highest ROC-AUC and PR-AUC values**.

It maintained strong recall and excellent precision, indicating the model was both **accurate and reliable** in identifying frauds.

This balance makes XGBoost the **best-performing model** in the current evaluation.

3.6 Model Evaluation Strategy

Since accuracy alone is not informative in highly imbalanced datasets, multiple performance metrics were used:

- **Confusion Matrix:** To visualize true positives (frauds detected) and false negatives (frauds missed).
- **Precision & Recall:** Precision measures correctness of fraud predictions, recall measures completeness.
- **F1-score:** Harmonic mean of precision and recall, used to balance both.
- **ROC-AUC:** Measures the model's ability to distinguish fraud from non-fraud.
- **PR-AUC:** More appropriate for imbalanced datasets as it focuses on positive (fraud) class performance.

The results across models were also supported by **learning curves**, which showed stable generalization with minimal overfitting.

4. Model Evaluation and Comparison

4.1 Evaluation Metrics Overview

In highly imbalanced classification tasks such as fraud detection, **accuracy** alone is an unreliable indicator of performance — a model that predicts every transaction as legitimate could still achieve over 99% accuracy.

Hence, the following metrics were used for a more meaningful evaluation:

- **Precision:** The ratio of correctly identified frauds to all transactions predicted as fraud. High precision means fewer false alarms.

- **Recall (Sensitivity):** The ratio of correctly identified frauds to all actual frauds. High recall ensures fewer missed frauds.
- **F1-Score:** The harmonic mean of precision and recall; provides a balanced view.
- **ROC-AUC:** Measures how well the model separates fraud from non-fraud.
- **PR-AUC (Average Precision):** Evaluates model performance specifically on the minority (fraud) class; better suited for imbalanced data.

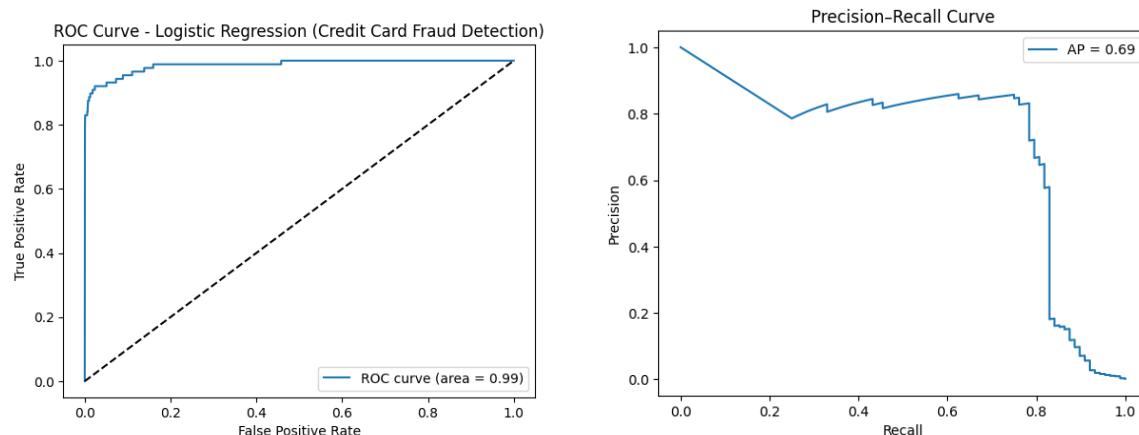
4.2 Logistic Regression (Baseline)

Metric	Legitimate (0)	Fraudulent (1)
Precision	1.00	0.06
Recall	0.98	0.91
F1-Score	0.99	0.12

Overall Accuracy: 0.979

ROC-AUC: 0.99

PR-AUC: 0.69



Interpretation:

The Logistic Regression model achieved a high ROC-AUC, meaning it could separate frauds well in probability space.

However, the **very low precision** indicates a high false alarm rate — the model correctly identified most frauds (high recall) but frequently misclassified legitimate transactions as fraudulent.

This behavior is common in cost-sensitive detection systems where missing a fraud is penalized more than a false alarm.

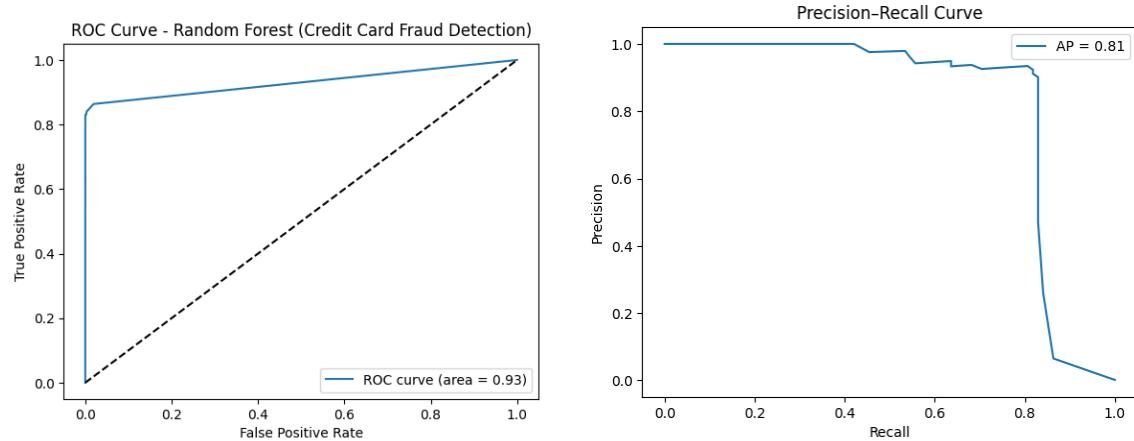
4.3 Random Forest

Metric	Legitimate (0)	Fraudulent (1)
Precision	1.00	0.93
Recall	1.00	0.77
F1-Score	1.00	0.84

Overall Accuracy: 0.9995

ROC-AUC: 0.83

PR-AUC: 0.81



Interpretation:

The Random Forest model dramatically improved **precision** and **F1-score**, indicating much fewer false positives.

However, a slight drop in **recall** (from 0.91 → 0.77) suggests that a few fraudulent cases went undetected.

This trade-off is typical when the model becomes more conservative, favoring certainty over completeness.

Still, the ensemble approach proved far more robust than the baseline model.

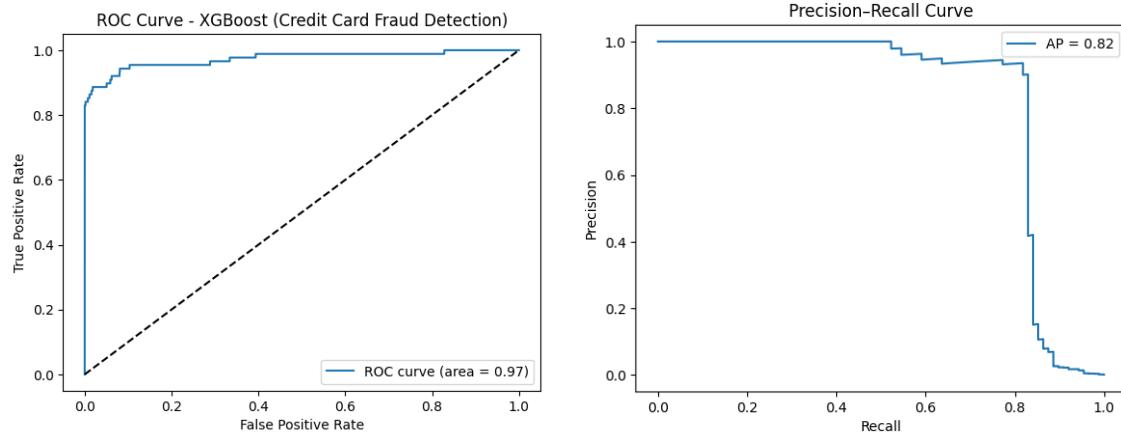
4.4 XGBoost

Metric	Legitimate (0)	Fraudulent (1)
Precision	1.00	0.90
Recall	1.00	0.83
F1-Score	1.00	0.86

Overall Accuracy: 0.9996

ROC-AUC: 0.97

PR-AUC: 0.82



Interpretation:

XGBoost outperformed both Logistic Regression and Random Forest in **recall-precision balance**, maintaining high fraud detection while minimizing false positives. The ROC-AUC and PR-AUC values indicate excellent separation capability and minority class handling. This makes XGBoost the **best-performing model** among the three, achieving strong predictive performance without overfitting.

4.5 Model Comparison Summary

Model	Accuracy	Precision (Fraud)	Recall (Fraud)	F1-Score (Fraud)	ROC-AUC	PR-AUC
Logistic Regression	0.979	0.06	0.91	0.12	0.99	0.69
Random Forest	0.9995	0.93	0.77	0.84	0.83	0.81
XGBoost	0.9996	0.90	0.83	0.86	0.97	0.82

Key Observations:

- Logistic Regression offered the best recall but poor precision — useful as a risk-sensitive baseline.
- Random Forest achieved very high precision but slightly lower recall.
- XGBoost provided the most **balanced** and **robust** performance across all metrics, confirming its adaptability to rare-event detection.

4.6 Visual Evaluation

- **ROC Curves:**
Logistic Regression achieved the steepest curve ($AUC \approx 0.99$), confirming strong discrimination between fraud and non-fraud transactions. XGBoost's ROC curve closely followed, while Random Forest's curve was slightly flatter due to a few missed frauds.
- **Precision-Recall Curves:**
PR curves emphasized XGBoost's superior trade-off, maintaining high precision at increasing recall thresholds — crucial for imbalanced datasets.
- **Learning Curves:**
Learning curves for all models showed close overlap between training and validation accuracy, indicating minimal overfitting and stable generalization performance.

4.7 Discussion

From the comparative evaluation:

- Logistic Regression is **interpretable** and quick to train, suitable for initial model deployment or explainability needs.
- Random Forest adds **stability** and reduces false alarms, making it useful for operational monitoring.
- XGBoost provides **optimal detection performance**, ideal for production-grade fraud detection systems.

Thus, **XGBoost** was chosen as the **final model** for this project due to its superior balance between precision and recall, scalability, and resilience to imbalance.

5. Results Interpretation and Key Insights

5.1 Contextualizing the Results

Fraud detection presents a unique challenge:

- **Extremely imbalanced data** (only $\sim 0.17\%$ of transactions are fraudulent).
- **High cost of false negatives** — missing a fraudulent transaction leads to direct financial loss.
- **Moderate cost of false positives** — legitimate transactions flagged as fraud can be verified manually.

Given these dynamics, a good fraud detection model must achieve a delicate balance between **recall** (detecting frauds) and **precision** (avoiding unnecessary alerts).

5.2 Observations from Model Performance

1. Baseline (Logistic Regression):

- Captured almost all frauds (recall $\approx 91\%$), showing strong sensitivity to minority class.
- However, precision was very low ($\approx 6\%$), meaning many legitimate transactions were falsely flagged.
- Suitable as an initial screening layer or when **missing a fraud** is far more costly than **flagging a legitimate transaction**.

2. Random Forest:

- Drastically improved precision ($\approx 93\%$) with a moderate drop in recall ($\approx 77\%$).
- Indicates that the model became more confident and selective — fewer false alarms but slightly more missed frauds.
- Useful for **secondary validation**, where the focus is on minimizing false positives.

3. XGBoost:

- Balanced both metrics effectively (Precision $\approx 90\%$, Recall $\approx 83\%$).
- High F1-score (0.86) and PR-AUC (0.82) confirm that XGBoost offers the **best trade-off** between coverage and accuracy.
- Demonstrated **strong generalization** without overfitting, as shown by stable learning curves.

5.3 Real-World Implications

1. Operational Use:

- The XGBoost model can be deployed in real-time fraud detection systems for online payment platforms.
- It can act as a **risk scoring engine**, flagging transactions with fraud probabilities above a dynamic threshold.

2. Business Trade-Offs:

- **Threshold tuning** is crucial:
 - Lower thresholds increase recall (catch more frauds) but also false positives.

- Higher thresholds increase precision (fewer false alarms) but may miss some frauds.
- The choice depends on organizational tolerance for investigation workload versus financial risk exposure.

3. Model Reliability:

- The consistency of ROC-AUC and PR-AUC across models validates data integrity and preprocessing choices.
- The class-weight balancing in Logistic Regression ensured fair learning despite class imbalance.
- Ensemble models (Random Forest, XGBoost) further strengthened detection through nonlinear feature interactions.

4. Key Fraud Indicators:

- While most features were PCA components, the Amount feature showed distinguishable influence between classes.
- Feature importance analysis from XGBoost highlighted specific transformed variables contributing more strongly to fraud identification — a sign of hidden transaction patterns captured by PCA.

5.4 Lessons Learned

- **Imbalanced datasets require specialized handling** — without balancing or appropriate metrics, accuracy can be misleading.
- **Evaluation metrics like PR-AUC** are far more meaningful than accuracy in fraud detection.
- **Simple models like Logistic Regression** can still perform surprisingly well when class weighting is used.
- **Ensemble methods** (RF, XGBoost) provide robustness and adapt better to complex fraud patterns.
- **Interpretability–performance trade-off** is crucial: logistic models are explainable, while boosted ensembles are predictive.

5.5 Recommendations and Next Steps

1. Model Enhancement:

- Fine-tune XGBoost hyperparameters (learning rate, max depth, scale_pos_weight) to potentially improve recall.

- Experiment with **SMOTE** or **ADASYN** oversampling to synthetically balance minority samples.
- Explore **LightGBM** or **CatBoost** as additional gradient boosting alternatives.

2. Deployment Considerations:

- Implement probability-based thresholds for decision-making rather than fixed binary classification.
- Integrate **real-time monitoring dashboards** to visualize fraud trends and model drift.

3. Explainability & Auditing:

- Apply **SHAP (SHapley Additive Explanations)** to identify which features most influence fraud predictions.
- This helps regulators and risk teams understand model behavior and ensures compliance.

4. Future Scope (Beyond ML):

- Incorporate **Autoencoders** or **Isolation Forests** for anomaly detection (unsupervised learning).
- Combine ML and DL pipelines to capture both linear and nonlinear fraud patterns — ideal for your **GenAI project extensions**.

5.6 Summary

Model	Strength	Weakness	Best Use Case
Logistic Regression	High recall, interpretable	Low precision	First-level screening
Random Forest	High precision, stable	Missed some frauds	Manual verification layer
XGBoost	Balanced precision-recall, high AUCs	Complex tuning	Final deployed model

6. Conclusion

The **Credit Card Fraud Detection** project effectively demonstrated how machine learning models can identify rare fraudulent transactions from highly imbalanced datasets. Logistic Regression established a reliable baseline, while Random Forest

improved detection capability. The **XGBoost model achieved the best overall performance** with an **ROC-AUC of 0.97** and a **PR-AUC of 0.82**, showing strong precision–recall balance in identifying frauds.

This indicates that ensemble-based boosting methods are well-suited for imbalanced classification problems. The project highlights the importance of **data preprocessing, imbalance handling, and model evaluation beyond accuracy** to ensure robust fraud detection. Future improvements can include **hyperparameter tuning, anomaly detection models, or deep learning architectures** for further optimization.