Credit Card Fraud Detection Project

1. Introduction

This project aims to develop predictive models to identify fraudulent credit card transactions using a highly imbalanced dataset. The dataset contains transactions made by European cardholders in September 2013. Of the 284,807 transactions, only 492 are fraudulent, representing 0.172% of the total. The features are mostly the result of PCA transformation, with "Time" and "Amount" remaining in their original form.

2. Dataset Overview

- **Features**:
  - `V1` to `V28`: Principal components from PCA transformation.
  - `Time`: Seconds elapsed since the first transaction.
  - `Amount`: Transaction amount.
  - `Class`: Target variable (1: Fraud, 0: Non-fraud).
- **Key Observations**:
  - The dataset is highly imbalanced.
  - Fraudulent transactions are distributed evenly over time but occur for amounts less than $2500.

3. Design Choices

3.1 Data Preprocessing

- **Handling Missing Values**: No missing values were present.
- **Normalization**: Applied `StandardScaler` to normalize `Time` and `Amount`, essential for PCA and robust model training.
- **Feature Selection**: Used `SelectKBest` to identify top 10 features based on ANOVA F-scores.
- **Class Imbalance Treatment**:
  - **Under-sampling**: Balanced the classes by reducing majority class samples.
  - **Over-sampling**: Applied SMOTE to synthetically generate minority class samples.
  - **Hybrid**: Used SMOTETomek to combine both approaches.

3.2 Model Selection

We trained and evaluated the following models:

- Logistic Regression
- Support Vector Classifier (SVC)

- Decision Tree Classifier
- Random Forest Classifier
- Bagging Classifier
- Gradient Boosting Classifier
- XGBoost Classifier
- Stochastic Gradient Descent Classifier (SGD)

3.3 Hyperparameter Tuning

- Grid Search was used to optimize hyperparameters for Random Forest and XGBoost classifiers.

3.4 Performance Metrics

- **Confusion Matrix**: To understand classification errors.
- **Precision, Recall, and F1-Score**: To evaluate performance on imbalanced data.
- **ROC-AUC**: To assess the model's ability to distinguish between classes.

4. Performance Evaluation

- **XGBoost** emerged as the best-performing model:
  - **Precision**: High for both classes, ensuring accurate fraud detection without excessive false positives.
  - **Recall**: High for the minority class, indicating the model successfully captures most fraudulent transactions.
  - **F1-Score**: Balanced, reflecting overall robustness.
  - **AUROC**: Achieved a high score, indicating strong discriminatory power.
- **Impact of Feature Selection**:
  - Removing features like `Amount`, `V13`, `V15`, `V22`, and `V23` marginally improved performance.

5. Discussion of Future Work

5.1 Enhanced Feature Engineering

- Investigate feature importance further to extract new meaningful features.
- Explore advanced techniques like autoencoders to create more robust features.

5.2 Improving Model Generalization

- Use ensemble learning to combine the strengths of different models.
- Experiment with advanced deep learning models, such as neural networks, to capture non-linear relationships.

5.3 Addressing Data Imbalance

- Collect more fraud data to reduce dependence on synthetic balancing methods.
- Explore adaptive boosting algorithms that inherently address class imbalance.

5.4 Real-World Deployment

- Develop an API or service to integrate the model into real-time transaction systems.
- Include mechanisms to update the model periodically as new fraud patterns emerge.

5.5 Explainability and Interpretability

- Implement explainability tools like SHAP or LIME to interpret model predictions, building trust with stakeholders.

5.6 Scalability and Optimization

- Optimize the pipeline for faster processing on large-scale datasets.
- Explore distributed computing frameworks like Apache Spark for scalability.

6. Conclusion

The project successfully developed a robust model for fraud detection with XGBoost performing the best. Future efforts will focus on enhancing feature engineering, improving generalization, and ensuring scalability and interpretability for real-world applications.