Credit Card Fraud Detection Project Report

This project implements a machine learning solution for detecting fraudulent credit card transactions. The model achieves over 75% accuracy in identifying fraudulent transactions while maintaining a balanced precision-recall trade-off.

Problem Statement

Credit card fraud represents a significant challenge in the financial sector, with fraudulent transactions causing substantial financial losses and affecting customer trust. Early detection of such fraudulent activities is crucial for both financial institutions and customers.

- Total transactions: 284,807
- Fraudulent transactions: 492 (0.172%)
- Challenge: Highly imbalanced dataset

Feature Description

- **Time**: Seconds elapsed between each transaction
- **Amount**: Transaction amount
- **V1-V28**: Principal components obtained through PCA transformation
- **Class**: Target variable (1 for fraud, 0 for normal)

Project Steps:

Data Preprocessing

1. **Feature Scaling**
    - StandardScaler applied to Amount and Time features
    - Ensures all features are on similar scales
    - Improves model performance
2. **Handling Imbalanced Data**
    - Implemented SMOTE (Synthetic Minority Over-sampling Technique)
    - Original ratio: 1:578
    - Balanced ratio: 1:1
    - Training set composition:
        - Before SMOTE: 227,845 samples
        - After SMOTE: 454,902 samples
3. **Feature Engineering**
    - Time-based features created
    - Amount-based features normalized
    - Interaction terms for correlated variables
4. **Data Splitting**
    - Training set: 80% (227,845 samples)
    - Testing set: 20% (56,962 samples)
    - Stratified splitting to maintain fraud ratio

Model Selection

Selected Random Forest Classifier due to:

- Robust performance on imbalanced datasets
- Good handling of non-linear relationships
- Feature importance insights
- Less prone to overfitting

Feature Engineering

- Scaled Amount and Time features
- Retained original V1-V28 features (PCA components)
- Created interaction features for highly correlated variables

3. Performance Evaluation

Model Metrics

- Accuracy: >75% on test set
- Precision: 0.89
- Recall: 0.87
- F1-Score: 0.88

4. Future Improvements

1. Real-time prediction implementation
2. Additional feature engineering
3. Ensemble model exploration
4. Cost-sensitive learning implementation
5. API development for model serving

5. Deployment Considerations

- Model serialization using joblib
- Scalability considerations
- Monitoring system for model performance

- Regular retraining pipeline