Ebpl-DS Credit Card Fraud Detection

Student Name : [MOORTHY M]

Register Number : [620623104026]

Institution : [GANESH COLLEGE OF ENGINEERING]

Department : [B.E COMPUTER SCIENCE AND ENGINEERING]

Date of Submission : [05/05/2025]

GitHub Repository Link

[Insert your GitHub link]

1. Problem Statement

Credit card fraud has become a pervasive threat in the digital era, costing billions annually.
With increasing online transactions, the need for real-time, accurate fraud detection is critical.
The project aims to develop an AI-powered system that can detect and prevent fraudulent credit card transactions using machine learning.

- Type of problem: Classification
- Why it matters: Effective fraud detection protects users, builds trust in digital finance, and saves financial institutions significant losses.

2. Project Objectives

- Build machine learning models that can accurately classify fraudulent vs. legitimate transactions.
- Achieve a balance between high recall (detecting fraud) and precision (avoiding false positives).
- Ensure real-world applicability with models that can scale and adapt to dynamic transaction patterns.

- Incorporate domain insights and anomaly detection for intelligent fraud prevention.

3. Flowchart of the Project Workflow

Suggested workflow:
1. Data Collection
2. Data Preprocessing
3. Exploratory Data Analysis
4. Feature Engineering
5. Model Selection and Training
6. Model Evaluation
7. Visualization & Insights
8. Deployment Preparation

(Insert a diagram here if editing manually)

4. Data Description

- Dataset name: Credit Card Fraud Detection Dataset
- Source: Kaggle (European cardholders' transactions, anonymized)
- Type of data: Structured, tabular, time-series
- Records and features: ~284,807 transactions with 30 features
- Static or dynamic: Static
- Target variable: Class (0 = Legitimate, 1 = Fraud)

5. Data Preprocessing

- Handled missing values (none in this dataset)
- Verified and removed duplicate records
- Analyzed outliers using boxplots and IQR method
- Converted Time and Amount as needed
- Normalized numerical features using StandardScaler
- Ensured class balance awareness (handled imbalanced data using SMOTE/undersampling)

6. Exploratory Data Analysis (EDA)

- Univariate Analysis: Class imbalance visualization, Distribution of Amount and Time
- Bivariate/Multivariate Analysis: Correlation heatmap, Fraud patterns
- Insights: Fraud transactions typically have lower amounts; time-based trends found

7. Feature Engineering

- Created hour_of_day from Time
- Standardized Amount and derived log_amount

- Used PCA-transformed features from the dataset (V1-V28)
- Performed class balancing using SMOTE
- Retained high variance/correlation features

8. Model Building

- Models: Logistic Regression, Random Forest Classifier
- Justification: Logistic Regression for baseline; Random Forest for robustness
- Data Split: 80/20 with stratification
- Metrics: Accuracy, Precision, Recall, F1-score, AUC-ROC

9. Visualization of Results & Model Insights

- Confusion matrix, ROC curve, PR curve
- Feature importance from Random Forest
- Visual summary of performance comparison

10. Tools and Technologies Used

- Language: Python
- IDE: Jupyter Notebook / Google Colab
- Libraries: pandas, numpy, seaborn, matplotlib, scikit-learn, imbalanced-learn
- Visualization: seaborn, matplotlib, plotly

11. Team Members and Contributions

Name	Contribution
[SABARI M]	Data preprocessing, EDA
[MOORTHY M]	Feature engineering, model training
[RAJA A]	Model evaluation, visualization
[PONSELVAN M]	Report writing, documentation
[PRAVEEN M].	visualization