Phase-2 submission template

Guarding transactions with AI-powered credit card fraud detection and prevention

Student Name: [R.KANISH KUMAR]

Register Number: [620123106047]

Institution: [AVS ENGINEERING COLLEGE]

Department: [ECE]

Date of Submission: [10/5/2025]

Github Repository Link: [https://github.com/kanishkumar-rk/AI-CreditCard-Fraud-Detection]

Problem Statement

The rise of digital payments has significantly increased the number of credit card transactions, leading to a surge in fraudulent activities. Traditional rule-based fraud detection systems are often slow and lack adaptability to new fraud patterns. With the increasing complexity of fraudulent strategies, it becomes essential to develop an intelligent system to detect and prevent credit card fraud in real-time.

This is a classification problem where the goal is to identify whether a transaction is fraudulent or legitimate based on historical transaction data.

Solving this problem helps protect financial institutions and users from significant financial losses and improves the trust and security of digital payment systems.

1. Project Objectives

The main objective of the project is to develop a machine learning-based system that can detect fraudulent credit card transactions in real-time.

● Key technical objectives:

○ Build models with high precision and recall to minimize false positives and false negatives.

○ Ensure the model performs well even with imbalanced datasets.

○ Achieve real-world applicability through interpretability and efficiency.

● The project goal evolved after initial data exploration revealed a significant class imbalance and highlighted the need for advanced sampling techniques and robust evaluation metrics.

2. Flowchart of the Project Workflow

[Data Collection] → [Data Preprocessing] → [Exploratory Data Analysis] → [Feature Engineering] → [Model Selection & Training] → [Evaluation & Visualization] → [Deployment/Reporting]

4. Data Description

● Dataset Name: Credit Card Fraud Detection Dataset

● Origin: Kaggle (https://www.kaggle.com/mlg-ulb/creditcardfraud)

● Type: Structured data

● Number of Records: 284,807 transactions

● Features: 30 (including Time, Amount, anonymized PCA components)

● Static dataset

● Target Variable: 'Class' (0 = Legitimate, 1 = Fraudulent)

5. Data Preprocessing

● Missing values: None present in dataset.

● Duplicate records: Checked and none found.

● Outlier detection: Detected in 'Amount'; used scaling techniques.

● Data types: Ensured numeric types. ● Categorical encoding: Not needed as dataset is numerical.

● Normalization: Scaled 'Amount' and 'Time' features using standard scaler.

● Documented each transformation using Python code with markdown cells in Jupyter Notebook.

6. Exploratory Data Analysis (EDA)

● Univariate Analysis:

  ○ Histograms and boxplots used to analyze 'Amount', 'Time' and PCA components.

● Bivariate/Multivariate Analysis:

  ○ Correlation matrix showed relationships between PCA features.

  ○ Countplots revealed significant class imbalance (fraud cases ~0.17%).

  ○ Scatterplots helped understand separation of fraud vs. Non-fraud data points.

● Insights Summary:

  ○ Majority of data is legitimate, requiring resampling techniques.

  ○ Some PCA features showed strong separation between classes.

7. Feature Engineering

● Created feature 'Hour' from 'Time' to capture transaction timing.

● No categorical features to encode.

● Considered using Synthetic Minority Over-sampling Technique (SMOTE) for handling class imbalance.

● Applied standard scaling to improve model performance.

● Dimensionality already reduced using PCA in source dataset.

8. Model Building

- Implemented Logistic Regression and Random Forest models.
- Logistic Regression: Interpretable and performs well on linearly separable data.
- Random Forest: Handles non-linear patterns, robust to overfitting.
- Data split: 80% training, 20% testing using stratified sampling.
- Metrics used:
  - Accuracy, Precision, Recall, F1-score due to class imbalance.
  - AUC-ROC for better evaluation.

9. Visualization of Results & Model Insights

- Confusion Matrix: Visualized true positives, false positives, etc.
- ROC Curve: Compared model performance.
- Feature Importance: Plotted top contributing PCA components using Random Forest.
- Insights:
  - Random Forest performed better in recall.
  - Important features contributed significantly to fraud detection.

10. Tools and Technologies Used

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

11. Team Members and Contributions● [Enter Team Member 1 R.KANISH KUMAR]: Data Cleaning and Preprocessing

● [Enter Team Member 2 S.HARISH]: Exploratory Data Analysis and Visualization

● [Enter Team Member 3 M.JEEVAN]: Feature Engineering and Model Development

● [Enter Team Member 4 S.JAYAVEL]: Documentation and Reporting