Phase-2

Student Name: Jayapratha A **Register Number:** 422723104047

Ins tu on: V.R.S.College of Engineering and Technology

Department: Computer Science Engineering **Date of Submission:**

10.05.2025 **Github Repository Link: h ps://github.com/Juiena-oss/Juiena.git**

1. Problem Statement

Credit card fraud is a major financial issue for banks, retailers, and consumers. The goal is to build a model that

detects fraudulent transactions based on historical transaction data.

- Problem Type: Binary Classification (Fraudulent vs. Non-Fraudulent)

- Why it Matters: Preventing fraud reduces financial losses and improves trust in financial systems. Real-time fraud

detection systems are essential for securing digital transactions.

2. Project Objectives

Technical Objective: Build and evaluate models to detect fraudulent transactions with high precision and recall.

- Model Goals:

- Minimize false negatives (missing fraud)

- Maintain interpretability (especially in high-risk domains)

- Handle class imbalance effectively

- The objective evolved post-EDA to focus more on handling data imbalance and model interpretability.

3. Flowchart of the Project Workflow

Data Collection → Data Preprocessing → EDA → Feature Engineering → Model Building → Evaluation → Results
Interpretation

4. Data Description

Dataset Name: Credit Card Fraud Detection

- Source: Kaggle (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud)

- Type: Structured, time-series

- Records: ~284,807 transactions

- Features: 30 (28 anonymized features + Time, Amount)

- Target: Class (0 = Non-Fraud, 1 = Fraud)

- Nature: Static dataset, highly imbalanced

5. Data Preprocessing

Missing Values: None detected

- Duplicates: Removed ~100 duplicate entries

- Outliers: Identified and treated using IQR on Amount

- Data Types: All numeric

- Encoding: Not required (already numeric)

- Scaling: StandardScaler applied to Amount and Time

- Imbalance: Will be handled during model training with SMOTE or class weights

6. Exploratory Data Analysis (EDA)

Univariate: Fraud cases are <0.2% of data. Amount distribution is skewed.

- Bivariate: Fraudulent transactions tend to have higher values in certain principal components (e.g., V14, V17)

- Multivariate: Correlation matrix shows strong patterns in a few components

- Insights:

- V14 and V17 show distinct distributions for fraud vs. non-fraud

- Feature selection or dimensionality reduction may be valuable

7. Feature Engineering

Created hour_of_day from Time

- Binned Amount into categories for analysis - PCA not applied as data already anonymized - SMOTE used to balance classes before training

8. Model Building

Models Used: Models: Logis c Regression, Random Forest, XGBoost - Split: 70/30

Train-Test split with stra fica on - Metrics:

- Accuracy
- Precision
- Recall
- F1-score
- AUC-ROC
- Why these models:

Logis c Regression for baseline & interpretability

- Random Forest/XGBoost for robustness and handling imbalance

9. Visualization of Results & Model Insights

Confusion Matrix: Shows effectiveness in capturing fraud - ROC Curve: AUC > 0.90 for best model - Feature Importance: V14, V17, V10 most important in fraud detection - Conclusion: XGBoost provided best performance with minimal overfitting

10. Tools and Technologies Used

Language: Python - IDE: Jupyter Notebook - Libraries: pandas, numpy, seaborn, matplotlib, scikit-learn, imbalanced-learn, XGBoost

- Visualization: seaborn, matplotlib, Plotly

11. Team Members and Contributions

Jayapratha.A: Data Cleaning, EDA -Kamali.V: Feature Engineering, SMOTE, Model Training Jayabharathi.N: Documentation, Visualizations