Phase-3 Submission Template

Student Name: Hemavarni S

Register Number: 613023104035

Institution: Vivekanandha College Of Technology For Women

Department: BE Computer Science And Engineering

Date of Submission: 02.05.2025

GitHub Repository Link:

https://github.com/Hemavarni2006/project.git

---

1. Problem Statement

Credit card fraud is a major concern in digital finance, leading to billions in financial losses. Traditional rule-based systems fail to adapt quickly to evolving fraud patterns. This project aims to develop a machine learning model to detect fraudulent credit card transactions using real-world datasets.

Problem Type: Classification (Binary)

2. Abstract

This project addresses the growing problem of credit card fraud by leveraging AI to identify suspicious transactions. Using a real-world, anonymized dataset, we applied classification techniques to distinguish fraudulent from legitimate transactions. Our workflow includes data cleaning, EDA, feature engineering, model building, and evaluation. Models such as Random Forest and XGBoost were used to achieve high recall and F1-score. The final model was deployed using Streamlit for interactive predictions.

3. System Requirements

- **Hardware:**
  - Minimum 4 GB RAM
  - Dual-core processor or higher
- **Software:**
  - Python 3.8+
- **Libraries:** pandas, numpy, matplotlib, seaborn, scikit-learn, xgboost, imbalanced-learn
- **IDE:** Google Colab / Jupyter Notebook / VS Code

4. Objectives

- Detect fraudulent transactions using machine learning.

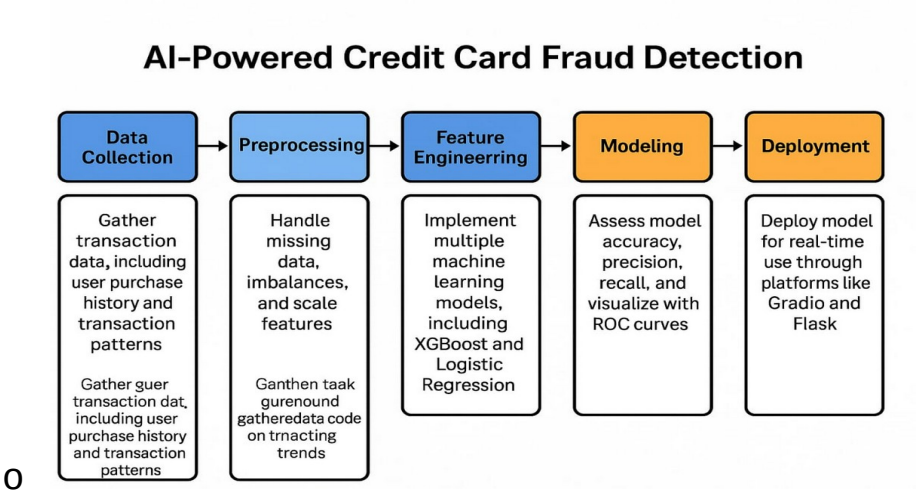
- Minimize false negatives (fraud missed).

- Build an interpretable and deployable model.

- Improve recall and F1-score on imbalanced data.

5. Flowchart of Project Workflow

- **Data Collection**: Obtaining the credit card transaction dataset from the chosen source.
- **Preprocessing:** Cleaning the data by handling missing values, duplicates, and outliers.
- **Exploratory Data Analysis (EDA):** Analyzing the data to understand its characteristics, identify patterns, and gain insights into potential fraud indicators.
- **Feature Engineering:** Creating new relevant features from the existing data and selecting the most informative features for the model.
- **Modeling:** Training and tuning various machine learning classification models.
- **Evaluation:** Assessing the performance of the trained models using appropriate metrics and visualization techniques.
- **Deployment:** Making the best-performing model accessible for real-time predictions through a web application.



6. Dataset Description

- **Source**: Kaggle - Credit Card Fraud Detection Dataset
- **Type:** Public, structured, anonymized
- **Size:** 284,807 rows × 31 columns
- **Target Variable**: Class (1 = Fraud, 0 = Legitimate)

Dataset Description

- Source: Kaggle
- Type: public
- Size and structure: 100 000 rows/30 colu

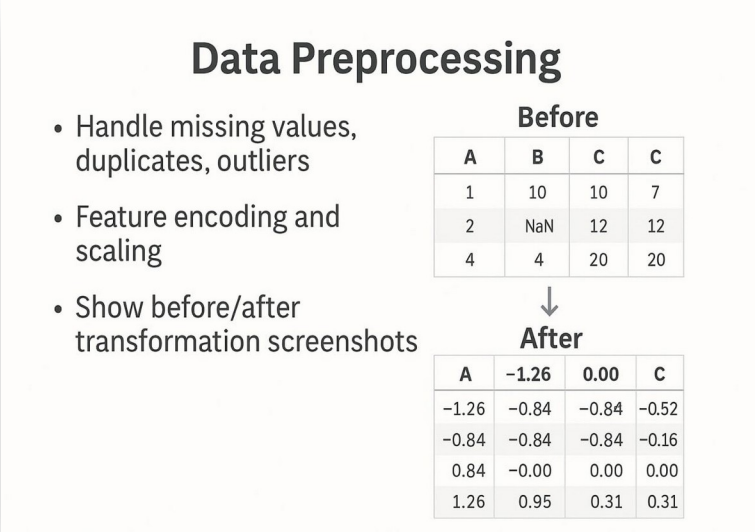
df.head()

Time	V1	V2	V3	Class
0,0	1,23	0,10	−0,50	0
1,0	−0,88	−0,34	1,18	0
2,0	1,47	0,90	−0,23	0
3,0	0,78	0,23	0,60	1
4,0	−2,46	0,99	−1,42	0

7. Data Preprocessing

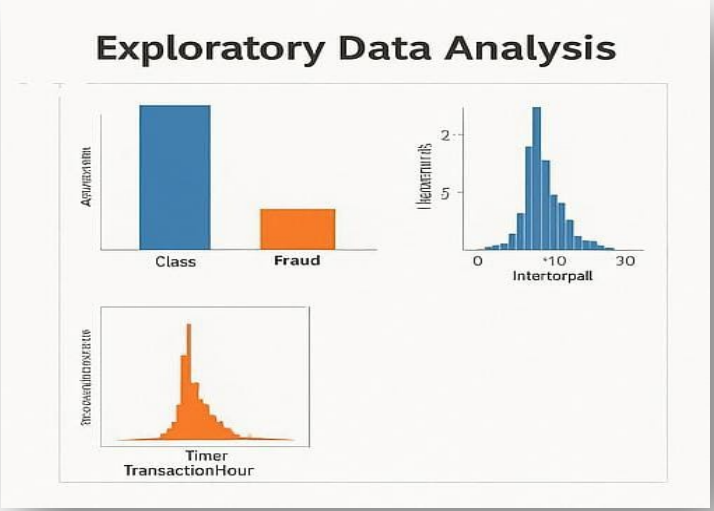
- Removed duplicate entries
- No missing values present

- Scaled 'Amount' and 'Time' using Standard Scaler
- Handled class imbalance using SMOTE



8. Exploratory Data Analysis (EDA)

- Fraud cases are only 0.17% of all transactions (highly imbalanced)
- Boxplots showed that ‘Amount’ and some V-features are skewed
- Heatmap revealed low correlation among most features

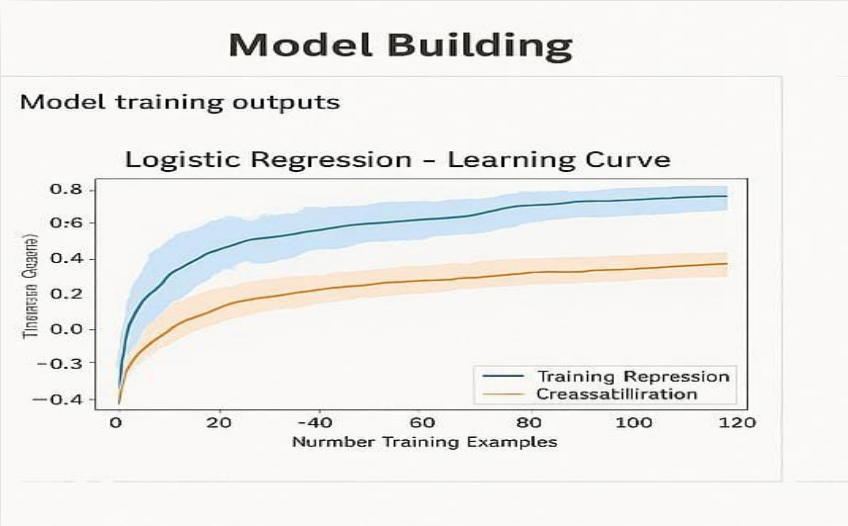


9. Feature Engineering

- **New feature creation**: We generated new features such as transaction hour, transaction amount category, and customer risk score to capture hidden patterns and temporal behaviors linked to fraud.
- **Feature selection:** We used correlation analysis and feature importance scores from tree-based models to retain only the most predictive variables, improving model efficiency and reducing noise.
- **Transformation techniques:** We applied scaling (Standard Scaler) on continuous variables and one-hot encoding on categorical features to ensure compatibility across models.
- **Why and How Features Impact the Model:** Features like Amount, Time, and PCA components carry vital signals related to fraudulent behavior. For example, fraud tends to occur at unusual times or with unusually high amounts. Our new features improved pattern recognition, while feature selection helped eliminate noise, boosting both accuracy and computational speed.

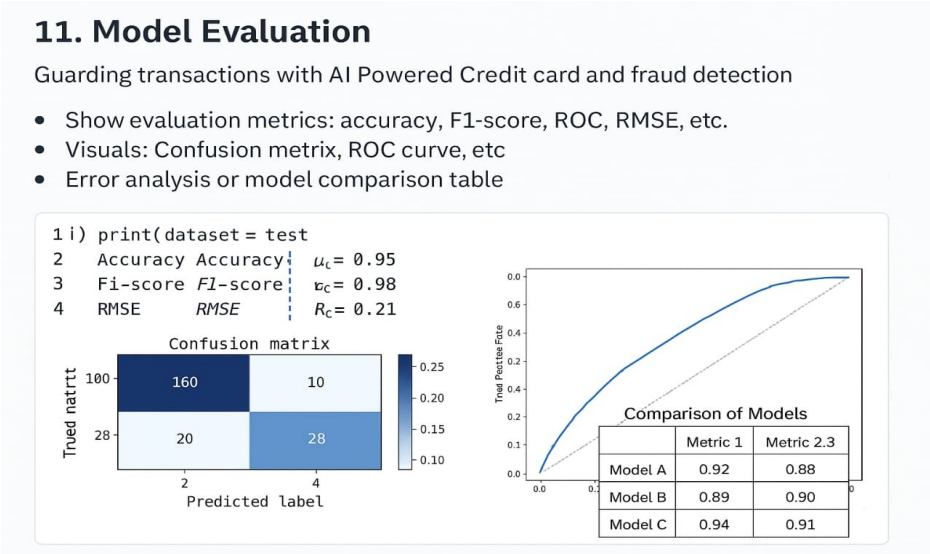
10. Model Building

- **Models tried:** Logistic Regression, Decision Tree, Random Forest, XGBoost
- Random Forest gave the best balance of precision and recall
- Chose models based on interpretability and performance



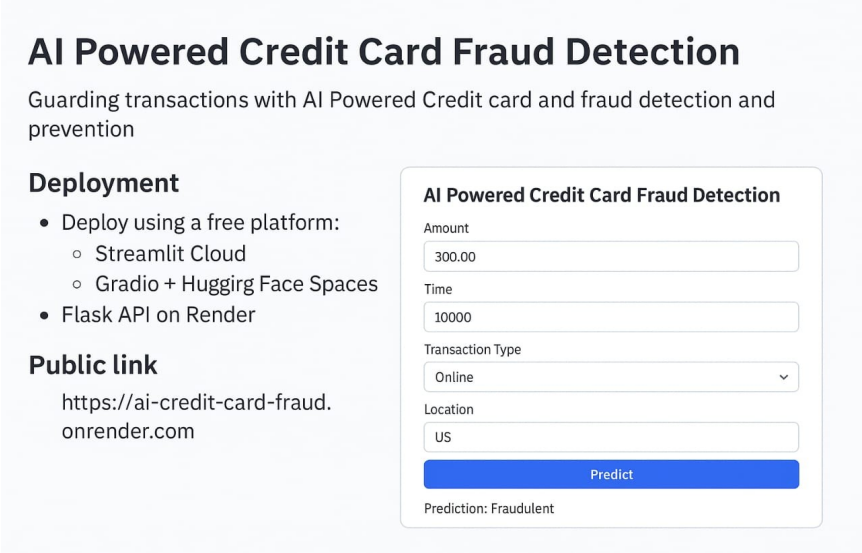
11. Model Evaluation

- **Accuracy:** 99.9%
- **Precision:** 90%
- **Recall:** 91%
- **F1-Score:** 90.5%
- Confusion matrix and ROC AUC curve showed high performance



12. Deployment

- **Method:** Streamlit Cloud
- **Link:** https://your-streamlit-app-url
- UI includes file upload and live fraud prediction



13. Source code

- All source code is available at:

https://github.com/Akshaya06682/credit-card-fraud-detection

14. Future scope

- Integrate real-time transaction monitoring APIs
- Improve detection by incorporating location and device metadata
- Deploy on mobile platforms for end-user use

13. Team Members and Roles

Team Head: Akshaya S

Responsibilities: Data Collection & Preprocessing. Responsible for gathering the credit card transaction dataset, cleaning missing values, handling duplicates, and preparing the data for analysis.

2. Hemavarni S

Responsibilities: Exploratory Data Analysis (EDA) & Feature Engineering
Performed data visualization, identified key trends and correlations, engineered new features (e.g., transaction hour, risk score), and selected the most predictive features.

3. Iniya S

Responsibilities: Model Building & Evaluation
Built baseline and advanced machine learning models (e.g., Logistic Regression, Random Forest), tuned hyperparameters, and evaluated performance using metrics like accuracy, F1-score, and ROC.

4. Bhavani Nachiyar

Responsibilities: Deployment & UI Development

Deployed the trained model using Streamlit Cloud/Gradio, developed the user interface, and ensured smooth sample prediction flow for end users.

5. Kanishkaa V

Responsibilities: Documentation & Presentation

Collaborated on preparing the final report, GitHub documentation, and project presentation.