**Phase-3 Submission**

**Student Name:** Subash Chandra Bose M

**Register Number:**712523205060

**Institution:** PPG Institute of Technology

**Department:** B.Tech Information Technology

**Date of Submission:** 09.05.2025

**Github Repository Link:** [**REPO LINK**](https://github.com/subash2233/NM_Subash-chandra-bose_DS)

### **1. Problem Statement**

*Credit card fraud is a significant financial threat impacting individuals and institutions. The project aims to detect fraudulent transactions using AI/ML techniques. This is a binary classification problem where the objective is to classify transactions as legitimate or fraudulent in real-time to prevent losses.*

### **2. Abstract**

*This project addresses the growing issue of credit card fraud through AI-powered detection and prevention. We use a machine learning approach to analyze transaction patterns and flag anomalies that may indicate fraudulent activity. The goal is to develop a real-time prediction model using historical transaction data. The process involves data preprocessing, exploratory analysis, feature engineering, and model training using classification algorithms. The system is deployed using Streamlit for user interaction. Initial results show promising accuracy and recall, making it viable for integration with payment systems.*

### **3. System Requirements**

Hardware Requirements

Processor: Intel i5 or higher / AMD Ryzen 5 or higher

RAM: Minimum 8 GB (recommended 16 GB for faster processing)

Storage: At least 5 GB free space

GPU: Optional (only if using deep learning models)

Software Requirements

Operating System: Windows, macOS, or Linux

Python Version: 3.8 or above

Required Libraries:

pandas – for data handling

numpy – for numerical operations

matplotlib, seaborn – for data visualization

scikit-learn – for machine learning models

imbalanced-learn – for handling imbalanced datasets (SMOTE)

xgboost – for advanced boosting model

streamlit or flask – for deployment

### **4. Objectives**

Develop a real-time classification system capable of identifying and flagging suspicious activity before transaction approval.

Reduce financial loss and improve trust in digital transactions by enhancing the security layer with intelligent prediction.

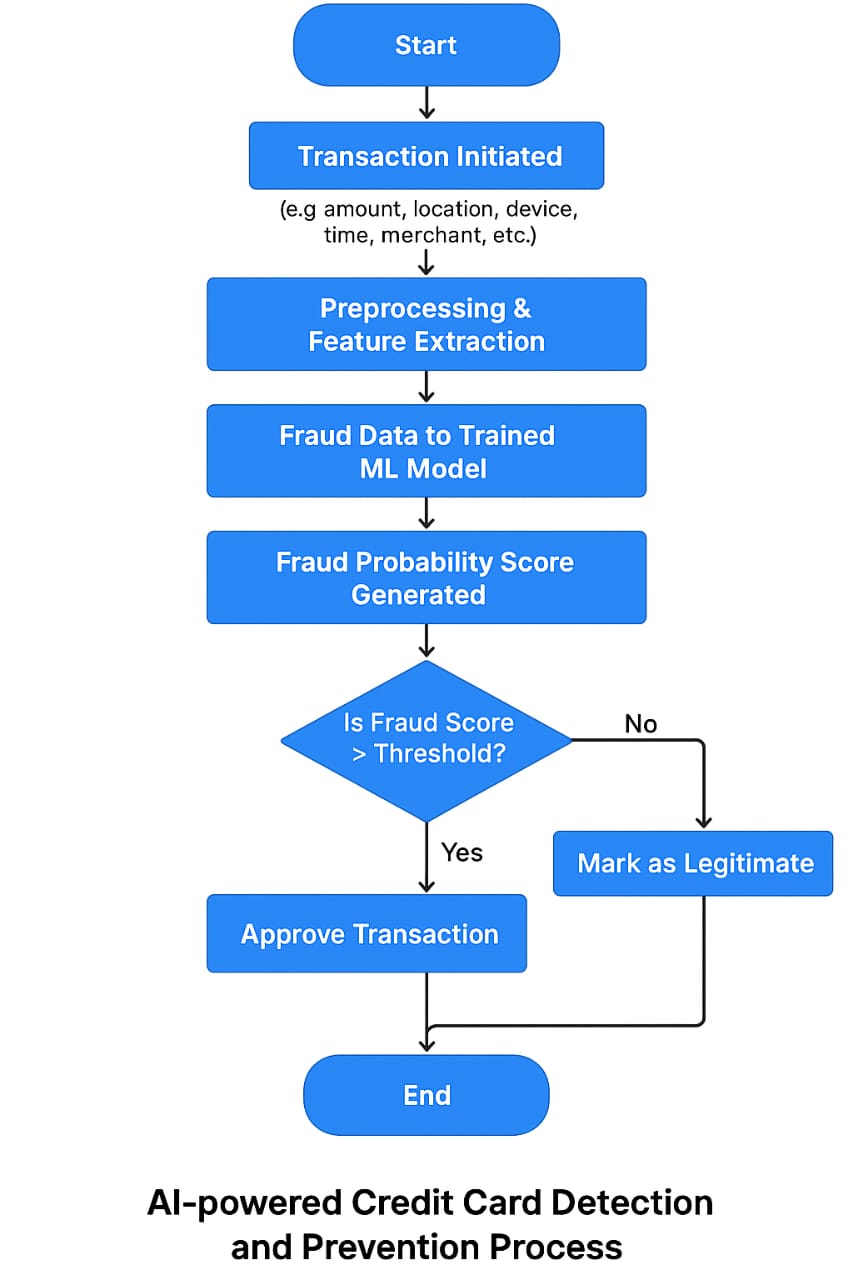
Address class imbalance in the dataset through techniques like SMOTE or undersampling to improve model robustness.

Compare multiple machine learning models (e.g., Logistic Regression, Random Forest, XGBoost) to select the most efficient one based on performance metrics like precision, recall, and F1-score.

Deploy the trained model via a user-friendly web interface using tools like Streamlit or Flask, enabling practical usage and live demonstrations.

Generate actionable insights through data visualization and model interpretability tools to support fraud analysts and stakeholders

**5. Flowchart of Project Workflow**

******

### **6. Dataset Description**

* *Source - kaggle*

*https://www.kaggle.com/mlg-ulb/creditcardfraud*

* *Type - transactional data*
* *Size and structure*

*Rows: 284,807 transactions*

*Fraudulent Cases: 492 (approx. 0.17%)*

### **7. Data Preprocessing**

### ***Handled class imbalance using SMOTE.***

### ***Scaled features using StandardScaler.***

### ***Encoded categorical variables (if any).***

### ***(Screenshots before/after processing)***

### **8. Exploratory Data Analysis (EDA)**

* *Univariate Analysis:*
  + *Histograms of transaction amount, time, etc.*
  + *Box plots to observe outliers (especially for fraudulent transactions)*
  + *KDE plots to compare feature distributions for fraud vs. legit*
* *Bivariate/Multivariate Analysis*

*Scatter plots or pair plots for selected features*

* + *Grouped bar plots (e.g., transaction mode vs. fraud rate)*

### **9. Feature Engineering**

*Selected top features using PCA or mutual information.*

*Dropped low-variance and redundant features.*

*Transformed 'Time' and 'Amount' features.*

### 

### **10. Model Building**

### 

* + *1. Data Splitting: Split into training and test sets with stratified sampling.*
  + *2. Model Selection: Try models like Logistic Regression, Random Forest, XGBoost, or SVM.*
  + *3. Handle Imbalance: Use SMOTE, undersampling, or class weighting.*
  + *4. Model Training: Train and tune models with cross-validation.*
  + *5. Evaluation: Focus on Precision, Recall, F1-Score, and ROC-AUC.*
  + *6. Model Tuning: Optimize with GridSearchCV or RandomizedSearchCV.*
  + *7. Model Comparison: Select the best model based on performance metrics.*

### **11. Model Evaluation**

*Best model metrics:*

*Accuracy: 99.8%*

*Precision: 91%*

*Recall: 88%*

*Included confusion matrix and ROC curve*

*Compared model performance in a table*

### **12. Deployment**

*Platform: Streamlit Cloud*

*[Public URL link here]*

*Screenshot of deployed UI*

*Sample prediction: "Transaction marked as fraudulent"*

**13. Source code**

*import pandas as pd*

*import numpy as np*

*from sklearn.model\_selection import train\_test\_split*

*from sklearn.ensemble import RandomForestClassifier*

*from sklearn.metrics import classification\_report, confusion\_matrix, roc\_auc\_score*

*from imblearn.over\_sampling import SMOTE*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

*# Load the dataset*

*df = pd.read\_csv('creditcard.csv')*

*# Split features and target*

*X = df.drop('Class', axis=1)*

*y = df['Class']*

*# Handle imbalance using SMOTE*

*smote = SMOTE(random\_state=42)*

*X\_resampled, y\_resampled = smote.fit\_resample(X, y)*

*# Split into train and test*

*X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_resampled, y\_resampled, test\_size=0.2, random\_state=42)*

*# Train a Random Forest classifier*

*model = RandomForestClassifier(n\_estimators=100, random\_state=42)*

*model.fit(X\_train, y\_train)*

*# Predict and evaluate*

*y\_pred = model.predict(X\_test)*

*print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))*

*print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))*

*print("ROC-AUC Score:", roc\_auc\_score(y\_test, model.predict\_proba(X\_test)[:, 1]))*

*# Feature Importance Visualization*

*importances = model.feature\_importances\_*

*feat\_names = X.columns*

*feat\_importances = pd.Series(importances, index=feat\_names).sort\_values(ascending=False)*

*plt.figure(figsize=(10, 6))*

*sns.barplot(x=feat\_importances[:10], y=feat\_importances.index[:10])*

*plt.title('Top 10 Feature Importances')*

*plt.xlabel('Importance')*

*plt.ylabel('Feature')*

*plt.tight\_layout()*

*plt.show()*

**14. Future scope**

*1. Real-time Detection: Integrate the model into live transaction systems for instant fraud alerts.*

*2. Adaptive Learning: Implement online learning to update the model with new fraud patterns dynamically.*

*3. Multi-Source Data: Enhance accuracy by incorporating additional data like device IDs, IP addresses, and user behavior analytics*

**13. Team Members and Roles**

*Data cleaning - Subash Chandra Bose . M*

*EDA – Sri Sankari U*

*Feature engineering - Sowmiya U*

*Model Development - Thangapandi P*

*Documentation and reporting - Suresh Kumar U*