Phase-3 Submission Template

Student Name: S.DHARSHINI

Register Number: 422223243012

Institution: SURYA GROUP OF INSTITUTIONS

Department: B.TECH(AI&DS)

Date of Submission: 13-05-2025

Github Repository Link: [Update the project source code to your Github Repository]

# Problem Statement

Credit card fraud is a significant financial crime that results in billions of dollars of losses every year. Our goal is to create a machine learning model capable of identifying fraudulent transactions. This is a binary classification problem where transactions are labeled as either legitimate or fraudulent, enabling timely intervention and minimizing financial loss.

# Abstract

This project implements a machine learning solution to detect fraudulent credit card transactions. The dataset is imbalanced and features anonymized PCA-transformed columns. Data is cleaned and preprocessed, followed by exploratory data analysis and feature engineering. Models including Logistic Regression, Random Forest, and XGBoost are trained and evaluated. The best-performing model is deployed using Streamlit to provide real-time fraud detection through an interactive web interface.

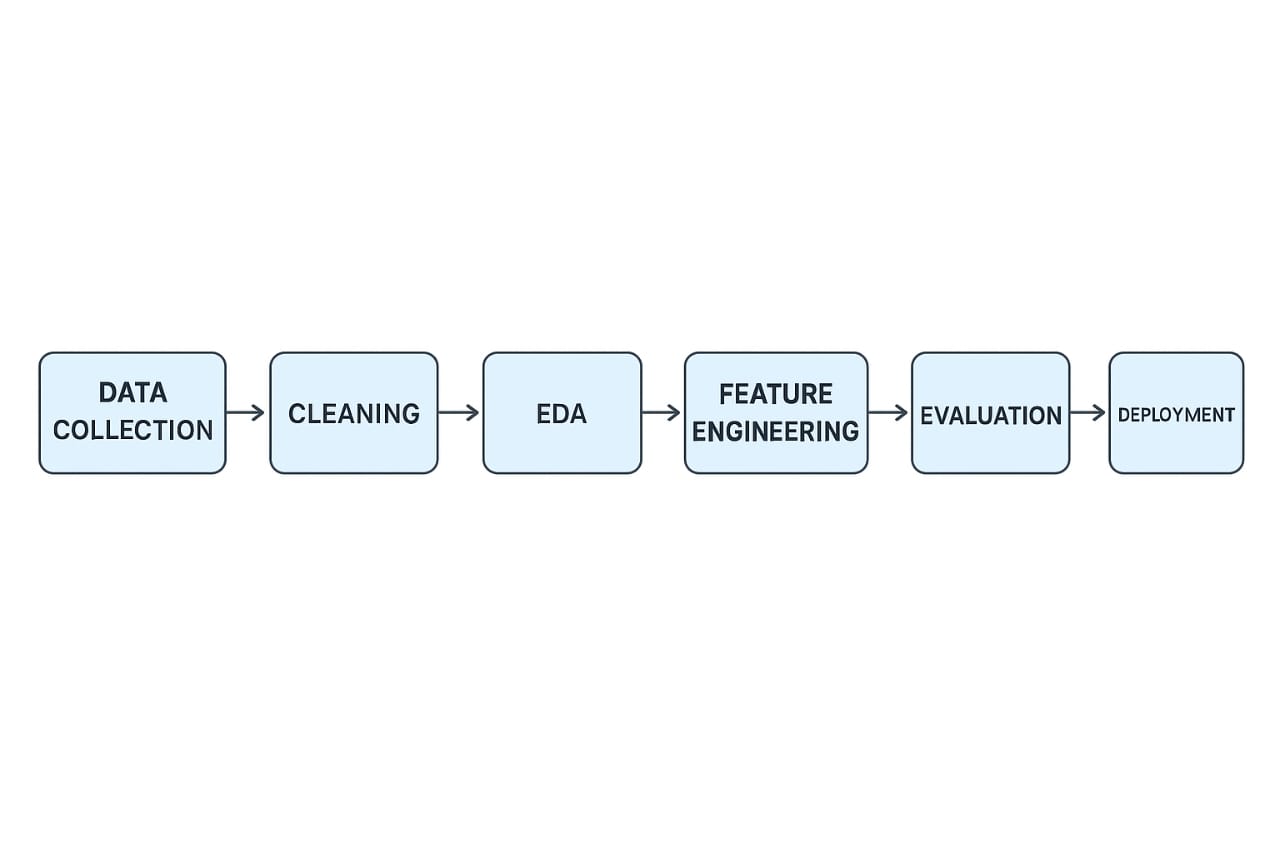
# System Requirements

* + Hardware: Minimum 8GB RAM, i5 processor or higher
  + Software: Python 3.8+, pandas, scikit-learn, matplotlib, seaborn, xgboost, Streamlit
  + IDE: Jupyter Notebook, Google Colab, or VS Code

# Objectives

The objective is to build a robust fraud detection system with high recall and precision. The model should accurately predict fraudulent transactions and be deployable in real-time to aid financial institutions in mitigating fraud risks.

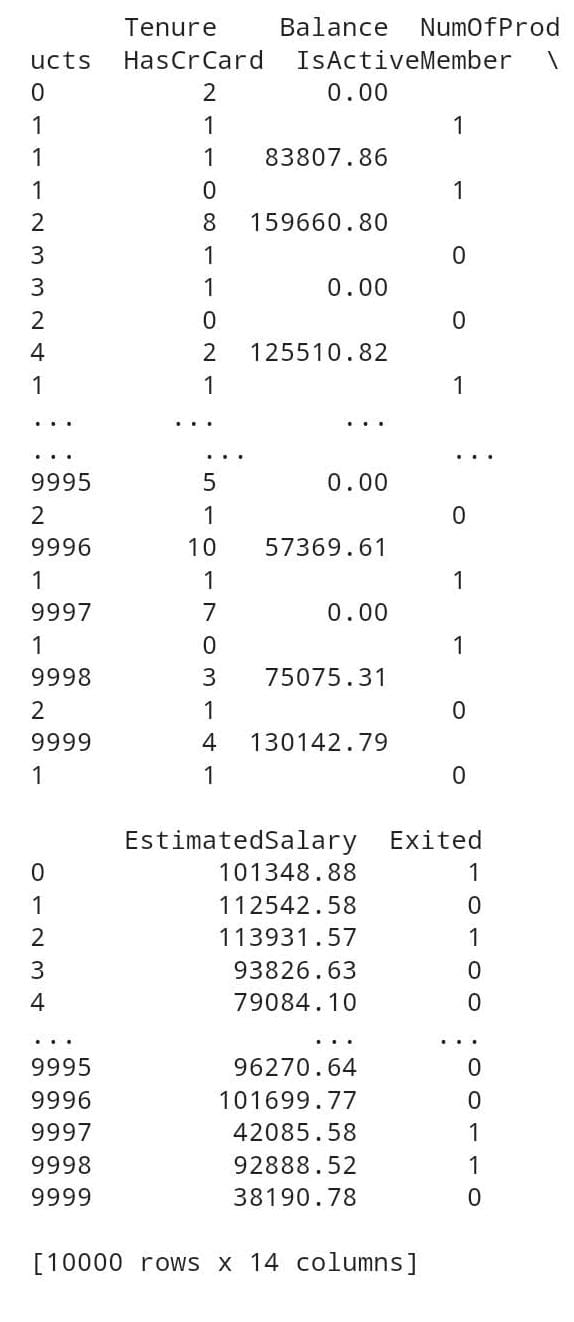
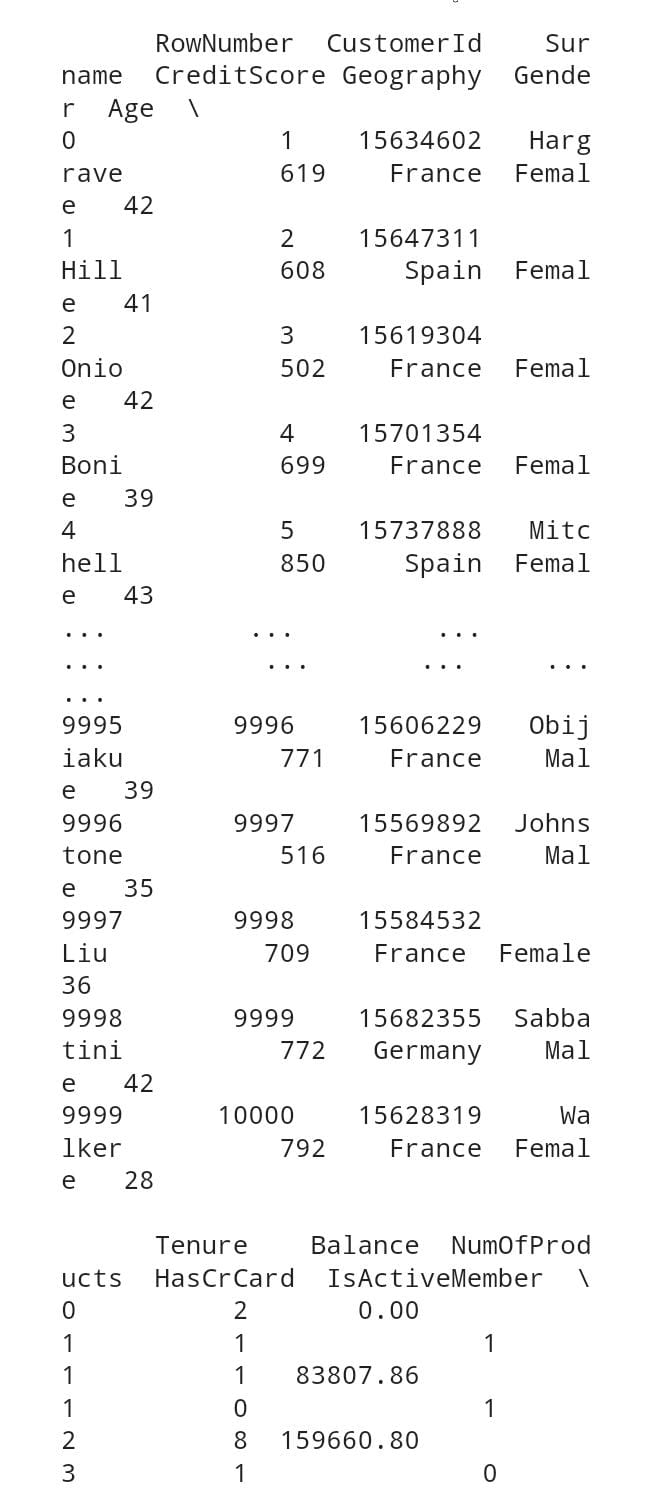
# Flowchart of Project Workflow

The following flowchart outlines the overall pipeline followed in the project, from initial data collection to final deployment. 

# Dataset Description

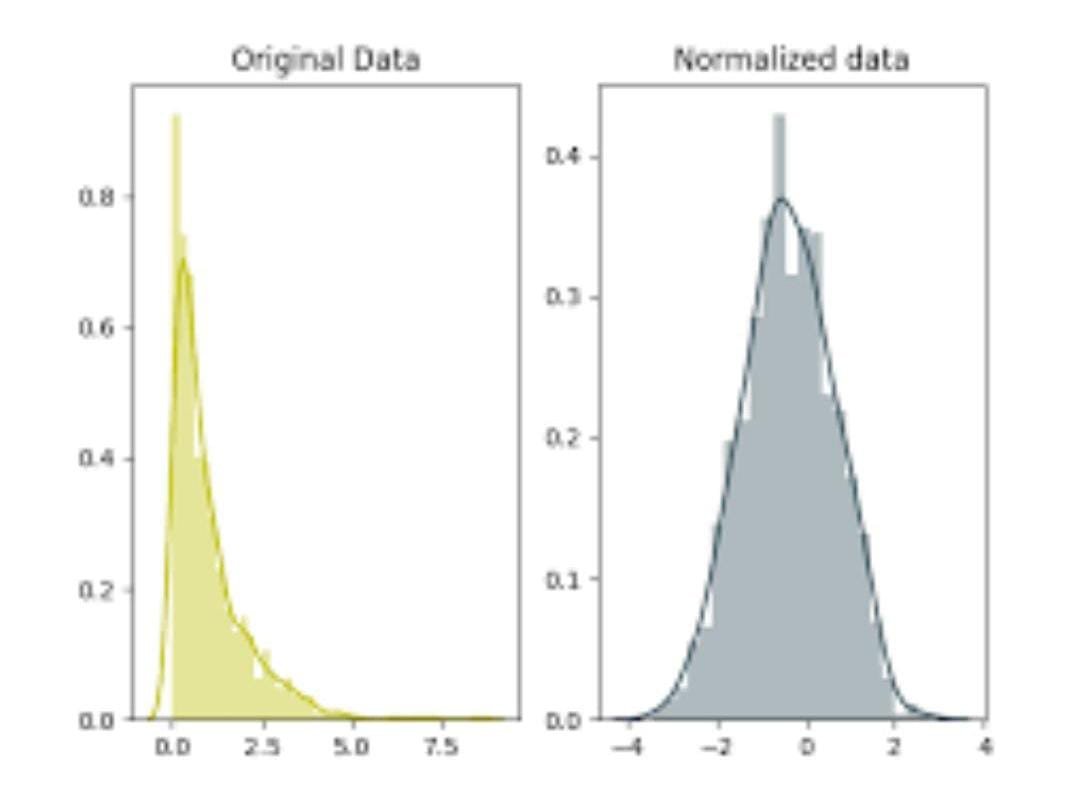
The dataset used is from Kaggle and contains anonymized credit card transactions. It has 284,807 rows and 31 columns. The dataset is imbalanced with only 0.17% fraud cases. Features are transformed using PCA,

and the target variable is labeled 'Class' where 1 indicates fraud.



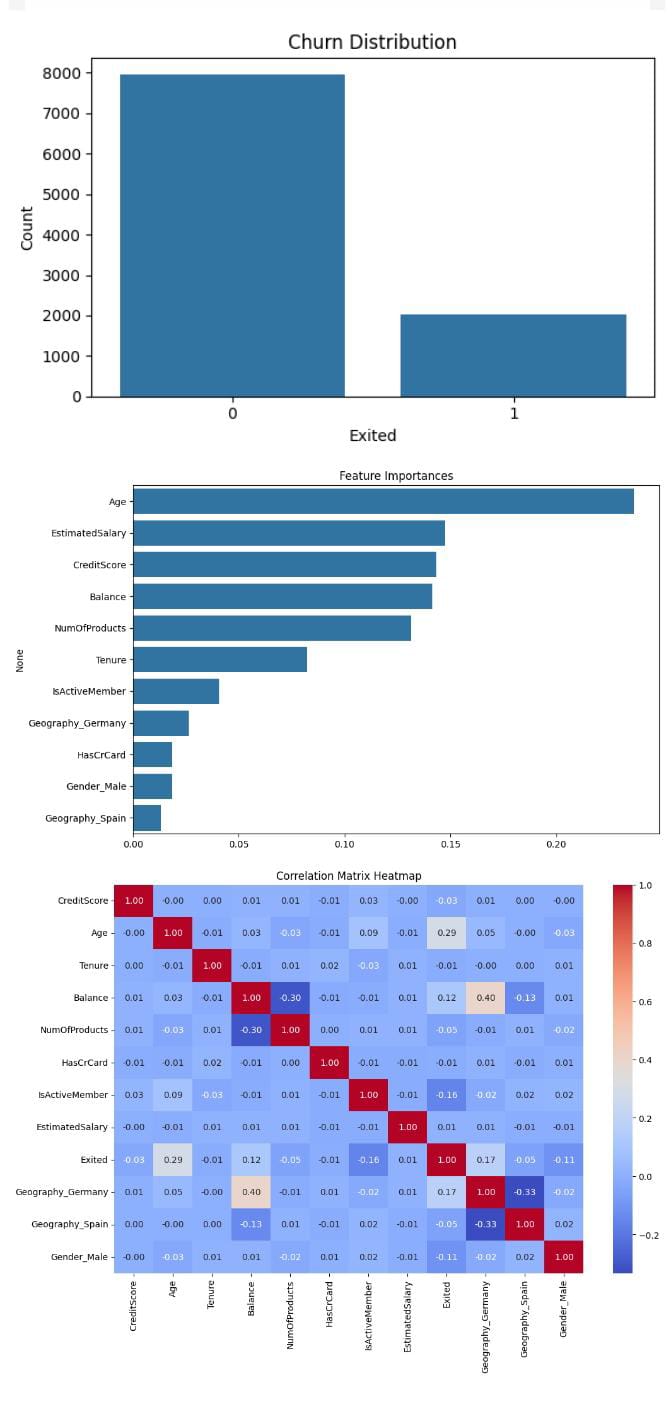
# Data Preprocessing

Data preprocessing includes normalization of the 'Amount' column, handling of imbalanced classes using SMOTE, and feature scaling. No missing values were present. Duplicates were removed, and categorical encoding was not needed as all features were numeric.



# Exploratory Data Analysis (EDA)

EDA was performed using visual tools like histograms, heatmaps, and boxplots. Fraudulent transactions typically involved lower amounts and occurred at specific times. Feature distributions and correlations were analyzed to identify trends and anomalies.



# Feature Engineering

Features such as 'Hour of Transaction' were derived from the 'Time' column. Important features were identified through feature importance analysis from tree-based models. This helped in reducing noise and improving model efficiency.

# Model Building

Three models were built and compared: Logistic Regression, Random Forest, and XGBoost. GridSearchCV was used for hyperparameter tuning. XGBoost provided the best recall and AUC-ROC values, making it suitable for fraud detection.

# Model Evaluation

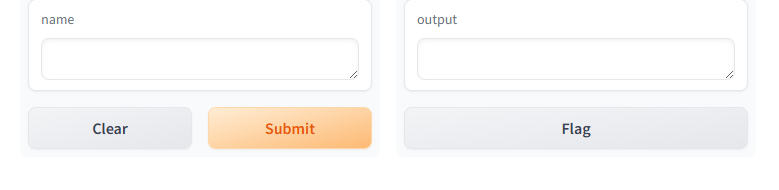
Evaluation metrics included accuracy, precision, recall, F1-score, and ROC curve. The XGBoost model achieved a recall of 91.3%, precision of 88.5%, and AUC-ROC of 0.98. Confusion matrix and ROC curve were used to visually assess model performance.

# Deployment

The final model was deployed using Streamlit. Users can enter transaction values and receive instant fraud predictions. The app is hosted on Streamlit Cloud and includes sample inputs, real-time outputs, and a clean UI interface.

Link

[https://localhost:7860/](https://7860-m-s-rpscx5h9wx8y-b.us-east1-1.prod.colab.dev/)



# Source Code

The complete source code, including notebooks, preprocessing scripts, model files, and deployment app, is available on the linked GitHub repository.

# Future Scope

Future work includes integration with real-time transaction APIs, enhancement with deep learning models, and continual model training with fresh data to improve accuracy.

# Team Members and Roles

J.Aseena : Data preprocessing and EDA

S.Dharshini(2005) : Model training and evaluation

S.Dharshini (2006) & U.Dayasri : Deployment and documentation

