**Phase-2 Submission Template**

**Student Name:** P.Dowrish

**Register Number:** 410623243024

**Institution:** Dhaanish Ahmed College of Engineering

**Department:** Artificial intelligence and data science

**Date of Submission:** 05/05/2025

**Github Repository Link:**[**https://github.com/dowrish2005/dowrish.git**](https://github.com/dowrish2005/dowrish.git)

# 1. Problem Statement

* Credit card fraud has emerged as a critical issue in financial security, leading to significant monetary losses and compromised personal data. This project focuses on leveraging AI and machine learning to predict fraudulent transactions in real-time, ensuring safer transactions and financial integrity. The problem type is classification, where transactions are categorized as fraudulent or legitimate.
* The significance of solving this problem lies in its potential to enhance security protocols for financial institutions, reduce chargeback losses, and improve consumer trust in digital payments.

# 2. Project Objectives

* Develop a machine learning model that accurately classifies transactions as fraudulent or legitimate.
* Identify and rank the most influential features that contribute to fraud detection.
* Provide insights into how transaction patterns, device behaviors, and demographic attributes affect fraud likelihood.
* Ensure model interpretability and usability in real-world banking and financial environments.
* Incorporate a user-friendly interface using Gradio for fraud detection testing.

**3. Flowchart of the Project Workflow**

A diagram of a data analysis process

AI-generated content may be incorrect.

# 4. Data Description

* **Dataset Name:** Credit Card Fraud Detection Dataset
* **Source:** Kaggle / Public Financial Institutions
* **Type of Data:** Structured tabular data
* **Records and Features:** Over 284,807 transactions with 30 attributes (numeric + categorical)
* **Target Variable:** Fraudulent (1) vs. Legitimate (0) transactions
* **Static or Dynamic:** Static dataset with potential integration for real-time processing
* **Attributes Covered:** Transaction amount, time, location, authentication method, and cardholder behaviour
* **Dataset Link**: <https://www.kaggle.com/datasets/bhadramohit/credit-card-fraud-detection>

# 5. Data Preprocessing

* Verified dataset integrity: handled missing values and inconsistencies.
* Removed irrelevant features with very low variance.
* Checked and confirmed absence of duplicate transactions.
* Categorical features one-hot encoded for machine learning compatibility.
* Applied StandardScaler to numeric columns to normalize values.
* Used SMOTE to balance fraudulent vs. legitimate transactions due to class imbalance.

c

# 6. Exploratory Data Analysis (EDA)

* **Univariate Analysis:**
  + Histograms of transaction amounts to understand fraudulent patterns.
  + Boxplots for high-risk variables like device type, transaction location, and authentication failures.
  + Count plots for categorical features such as merchant type and transaction frequency.
* **Bivariate & Multivariate Analysis:**
  + Correlation matrix shows strong relationships between transaction time and fraud likelihood.
  + Scatter plots of transaction amounts vs. fraud risk confirm specific fraud-prone ranges.
  + Grouped bar charts highlight differences in fraud occurrence based on merchant categories and transaction volume.

# 7. Feature Engineering

* Created interaction features such as avg\_transaction\_per\_day and velocity\_features.
* Derived binary feature: high\_risk\_transaction = (yes/no) based on abnormal activity patterns.
* Removed highly correlated or redundant features to reduce multicollinearity.
* Performed label encoding for binary features like authentication\_verified.

# 8. Model Building

* **Algorithms Used:**
  + Logistic Regression: Baseline comparison.
  + Random Forest: Capturing non-linear fraud patterns.
  + XGBoost: Optimized model performance on imbalanced data.
* **Model Selection Rationale:**
  + Logistic Regression: Easy to interpret.
  + Random Forest: Handles categorical and numeric features well.
  + XGBoost: Effective at detecting complex fraud patterns.
* **Train-Test Split:**
  + 80% training, 20% testing.
  + Used stratified sampling for balanced fraud detection.
* **Evaluation Metrics:**
  + Precision, Recall, F1-score (important for fraud classification).
  + AUC-ROC Curve to evaluate fraud detection capability.

# 9. Visualization of Results & Model Insights

* **Feature Importance:**
  + Visualized using bar plots from Random Forest and XGBoost.
  + High transaction amounts and unusual time intervals ranked highest in fraud risk.
* **Model Comparison:**
  + Plotted precision, recall, and AUC-ROC curves for all models.
  + XGBoost significantly outperformed Logistic Regression in fraud identification.
* **Residual Plots:**
  + Checked misclassification rates against actual fraud cases to minimize false negatives.
* **User Testing:**
  + Integrated model into a Gradio interface for transaction fraud detection.

# 10. Tools and Technologies Used

* **Programming Language**: Python 3
* **Notebook Environment**: Google Colab
* **Key Libraries**:
  + pandas, numpy for data handling
  + matplotlib, seaborn, plotly for visualizations
  + scikit-learn for preprocessing and modeling
  + Gradio for interface deployment

# 11. Team Members and Contributions

○ Data cleaning: M.Dinesh

○ EDA: P.Dowrish

○ Feature engineering:A.Hafeez

## ○ Model development:F.Fayaz

○ Documentation and reporting:M.Dinesh