Phase-2:

ProjectReport:

GitHublink : <https://github.com/SachinSarojini123/Credit-for-detection-and-prevenstion>

Project title :

Guarding Transactions withAI-poweredCredit CardFraud DetectionandPrevention

1.ProblemStatement

Credit card fraud is a growing threat in the digital economy. As transaction volumes increase, so does the complexity of identifying fraudulent activity. Traditional rule-based systemsare no longer sufficient to detect sophisticated fraud patterns.This project aims to develop a machine learning system that uses AI techniques to accurately detect and prevent fraudulent transactions in real time. The objective is to enhance transactionsecurity, minimizefinancial losses,and ensure customertrust.

2.ProjectObjectives

* Develop a machine learning model that detects fraudulent credit card transactions.- Analyze and rank important features influencing fraud detection.- Handle class imbalance effectively using techniques like SMOTE.- Provide visual insights and performance metrics.- Integrate a user interface using Gradio for real-time testing.- Ensure high recall without sacrificing too much precision.

3.FlowchartoftheProjectWorkflow



4.DataDescription

* Dataset Name: Credit Card Fraud Detection- Source: Kaggle- Type: Structured tabular data(anonymized)-RecordsandFeatures: 284,807 transactions, 30 features(V1–V28, Time, Amount)- Target Variable: Class (0 = legitimate, 1 = fraud)- Static or Dynamic: Static-Features: Principal components (V1-V28),Time, Amount,andClass

5.DataPreprocessing

* Checked for and handled missing values.- Standardized numerical fields using StandardScaler.- Applied SMOTE to address class imbalance (fraud ~0.17%).- Removedfeaturesthatdidn't contribute significantlyto prediction.

6.ExploratoryDataAnalysis(EDA)

* Univariate analysis: Distribution plots of Amount and Time.- Bivariate analysis: Correlation matrix; fraud patterns compared against legitimate.- Fraudulent transactions typically have smaller amounts.- PCA features (V1–V28) provide strong signal separation.

7.FeatureEngineering

* Normalized Amount and Time.- Created hour-of-day and transaction-rate features.- Reduced redundancy via correlation checks.- Engineered fraud likelihood indicators basedon outlierbehavior.

8.ModelBuilding

* Algorithms Used: - Logistic Regression - Random Forest - XGBoost Classifier- Rationale: - Logistic Regression: For baseline and interpretability. - Random Forest: For robust classification and feature ranking. - XGBoost: For optimized performance and fine-tuned detection.- Evaluation Metrics: - Precision, Recall, F1-score - ROC-AUC Score - Confusion Matrix- Train-Test Split: - 80% training, 20% testing - Stratified samplingto maintainclassproportions

9.VisualizationofResults&ModelInsights

* ROC-AUC curves show high detection accuracy for XGBoost.- Feature Importance: - Top predictorsincludedV14, V17, V10 (highlycorrelated with fraud).- Confusion matrix revealed strong recall, low false negatives.- Integrated model into Gradio UI for live fraud prediction.

10.ToolsandTechnologiesUsed

* Programming Language: Python- Environment: Google Colab- Libraries: - pandas, numpy - scikit-learn, XGBoost - seaborn, matplotlib - imbalanced-learn - Gradio

11.TeamMembersandContributions

ProjectLead

K.Harini-supervised all project stages, handled model development and reporting

TeamMembers

E.SachinSarojini-DataCleaningandPreprocessing

C.Logeshwari-ExploratoryDataAnalysisandFeatureEngineering

C.Sandhiya-InterfaceDevelopmentandVisualization