Github Link: https://github.com/Ganesh13

Project Title: Guarding transactions with AI-powered credit card fraud detection and prevention

PHASE-2

1. Problem Statement

Credit card fraud is a significant concern in the digital age, where millions of transactions occur daily. Detecting fraudulent transactions in real time is challenging due to the subtlety and evolving tactics of fraudsters. Traditional rule- based systems are insufficient in handling the complexity and scale of modern financial transactions. Scalable and adaptive system to identify and prevent fraudulent activities effectively

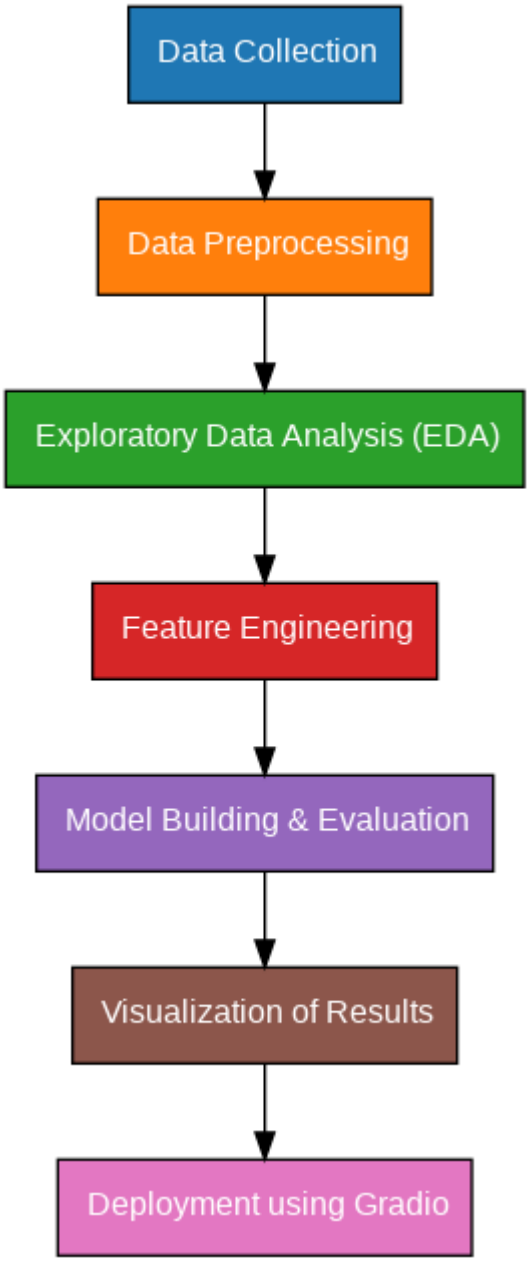
Lack of contextual understanding: Traditional systems often lack contextual understanding of transactions, leading to incorrect assumptions about legitimate the risk of false positive.

Regulatory Compliance: Financial institutions must comply with regulatory requirements, such as PCI-DSS, to ensure the security of credit card transaction and protect sensitive.

2. Project Objectives

- Develop an Advanced Machine Learning Model: Design and develop a sophisticated machine learning model that can analyze large datasets and identify complex patterns of fraudulent behavior.
- .Improve Accuracy and Reduce False Positives: Improve the accuracy of fraud detection and reduce the number of false positives, minimizing customer inconvenience and potential loss of business.
- Provide Real-Time Detection and Prevention: Develop a system that can detect and prevent fraudulent transactions in real-time, minimizing losses for financial institutions and merchants.
- Enhance Customer Experience: Provide a seamless and secure payment experience for customers, minimizing the risk of false positives and preventing fraudulent transactions.
- Adapt to Evolving Fraud Patterns: Develop a system that can adapt to evolving fraud patterns, ensuring that it remains effective in detecting and preventing fraudulent activity.
- Integrate with Existing Systems: Integrate the AI-powered fraud detection system with existing systems and infrastructure, ensuring seamless operation and minimal disruption to business processes.

3. Flowchart of the Project Workflow



4. Data Description

- **Source**: Typically sourced from anonymized credit card transactions (e.g., Kaggle's Credit card fraud detection dataset).
- **ATTRIBUTES:**
- **Time: Second elapsed between this transaction and the first.**
- **Amount: Transaction amount**
- **V1-V28: Result of PCA transformation (original features anonymized)**

- **Class: Target variable (0 = legitimate,1 = fraud)**
- Dataset: https://www.kaggle.com/mlg-ulb/creditcardfraud

5. Data Preprocessing

- Missing Values Handling: Check and handle any missing values
- Data Normalization: Scale Amount and Time since other features are already scaled.
- Handling Imbalanced Data: Use techniques like SMOTE or undersampling.
- Splitting Data: Train-test split (e.g., 70%-30%).
- Feature Selection: Based on correlation and importance.
- Outlier Detection: Optional, but can help reduce noise. ●

6. Exploratory Data Analysis (EDA)

- **Goal: Understand the dataset and identify patterns, anomalies, and relationships.**
  - ○Common steps:
  - ○ Lode data: Read transaction dataset (e.g., .csv).
  - ○ Summarize data: Use .info(), .describe (), and value counts.
  - ○ Class imbalance: Typically, fraud is <1% of all transactions.
  - ○ Correlation analysis: Identify relationships between features.
  - ○ Visualization: Histograms for distributions Box plots for outliers Time series plots for outliers Time series plots (fraud over time) Heatmaps (correlation)

7. Feature Engineering

- Goal: Create and transform variables to improve model performance.
- TECHNIQUES:
- Time-based features: Hour of transaction, day of week.
- User behavior: Average transaction amount, frequency.
- Geolocation features: Country, IP mismatch.
- Transaction risk flags: Large amount, new device, new location.
- Encoding: One-hot encoding for categorical variable.

8. Model Building

- Model to try;
- Logistic regression (baseline)
- Random forest/ XGBoost (strong tree-based models)
- Neural Network (for deeper patterns)
- Isolation forest / autoencoder (for anomaly detection)

Handling class imbalance:
SMOTE (synthetic minority oversampling technique)
Class weighting
Anomaly detection approaches

Metrics:

Precision, Recall, F1-score (focus on Recall for fraud detection)

ROC-AUC

Confusion Matrix

9. Visualization of Results & Model Insights

●ROC Curve

Precision-Recall curve

Confusion Matrix heatmap

Feature importance:

Tree-based feature importance

SHAP (Shapley Additive exPlanations) values for interpretability

Fraud insights:

Temporal patterns

High-risk transaction segments

Risk score distribution

10. Tools and Technologies Used

Language & Libraries:

Python (Pandas, Numpy, Matplotlib, Seaborn)

Scikit-learn, XGBoost, TensorFlow/Keras

SHAP (for explainable AI)

Platforms:

Jupyter Notebook / Google colab

Cloud: AWS/GCP/Azure for scaling

Databases: SQL or NoSQL for storing transaction

11. Team Members and Contributions

- *M.Ganesh- Data cleaning*
- *Harshini -EDA*
- *G.Ramya- Feature engineering*
- *Priyanka sahoo Model development*
- *Sowmiya Documentation and reporting*