Github link: https://github.com/Thirupathi5657

TITLE: GUARDING Transactions with AI-Powered Credit Card Fraud Detection and Prevention

1: Problem Statement:

Credit card fraud is a significant issue for financial institutions, merchants, and consumers globally. With the increasing volume of online and offline credit card transactions, the potential for fraudulent activity has also risen. Traditional fraud detection systems, relying on rule-based algorithms, often fall short in identifying new and sophisticated fraudulent schemes. This challenge is exacerbated by the vast number of transactions that must be processed quickly, the evolving nature of fraud tactics, and the need for real-time detection without negatively impacting legitimate user experiences.

To address this, there is a need for advanced, AI-powered credit card fraud detection and prevention systems that can adapt to emerging fraud tactics while minimizing false positives and optimizing the transaction experience for legitimate users.

2. Project Objectives

- Build a machine learning model that can reliably detect fraudulent transactions.
- Utilize supervised and unsupervised learning techniques to develop a classification model capable of differentiating between legitimate and fraudulent transactions.
- Train the model using labeled datasets with both fraudulent and non-fraudulent transactions.
- Implement anomaly detection techniques to identify emerging fraud patterns that have not yet been encountered in historical data.

3: Flowchart of the Project Workflow

| Start |

| Data Collection & Integration|

| - Collect transaction data |

| - Integrate external data sources (e.g., device info, geolocation) |

| Data Preprocessing |

| - Data cleaning & normalization|

| - Feature extraction & selection |

| Model Selection |

| - Choose appropriate ML models (e.g., supervised, unsupervised, RL)|

| - Select algorithms (e.g., decision trees, SVM, deep learning) |

| Model Training & Evaluation|

| - Split data into training & testing sets |

| - Train model on historical labeled data |

| - Evaluate model performance (precision, recall, F1-score) |

| - Hyperparameter tuning |

| - Cross-validation for robustness |

| - Adjust for false positives/negatives |

| Real-Time Fraud Detection |

| - Deploy model for real-time scoring of transactions |

| - Assign fraud risk score to each transaction |

| Action on Fraudulent Transactions |

| - Flag suspicious transactions |

| - Send alerts to customers or institutions |

| - Initiate verification process if necessary |

| Continuous Learning & Feedback Loop |

| - Monitor model performance (e.g., false positives, detection accuracy) |

| - Update model with new fraud patterns and data |

| - Retrain model periodically for continuous improvement |

| Compliance & Security |

| - Ensure privacy (GDPR, PCI-DSS) |

| - Data encryption and secure storage |

| End |

4. Data Description

● Dataset Name: Student Performance Data Set

● Source: UCI Machine Learning Repository

● Type of Data: Structured tabular data

● Records and Features: 395 student records and 33 features (numeric + categorical)

● Target Variable: G3 (final grade, numeric)

● Static or Dynamic: Static dataset

● Attributes Covered: Demographics (age, address, parents' education), academics (G1, G2, study time), and behavior (alcohol consumption, absences)

● Dataset Link: https://github.com/Thirupathi5657/Project-phase2-

5. Data Preprocessing

1. Data Collection

The first step in preprocessing is to gather the raw transaction data. This typically includes:

• Transaction Features:

  o Transaction ID

  o Cardholder details (user ID, card number, etc.)

  o Merchant details (merchant ID, merchant category, location, etc.)

  o Transaction amount

o Transaction time (timestamp)

o Transaction type (online, offline, etc.)

o Device details (device ID, IP address)

o Geolocation (latitude, longitude)

2. Data Cleaning

Data cleaning involves handling missing values, removing duplicates, and dealing with any inconsistencies or errors in the raw data.

Actions:

• Missing Values:

o Handle missing data points using techniques like imputation (mean, median, or mode) or dropping rows/columns with excessive missing values.

6. Exploratory Data Analysis (EDA)

● Univariate Analysis:

○ Mean, Median, Mode

○ Standard Deviation & Variance

○ Min & Max

○ Histograms, Box Plots, Density Plots

● Bivariate & Multivariate Analysis:

○ Correlation matrix

○ Scatter plots of G1 vs G3 and G2 vs G3

○ Grouped bar charts

● Key Insights:

○ G1 and G2 are the strongest indicators of G3

○ More study time correlates with higher G3

○ Students with more failures or absences tend to score lower

7. Feature Engineering

Transaction-Based Features

• Transaction Amount Differences:

o Amount vs. Average Transaction

o Formula: Transaction Amount - Average Transaction Amount

8. Model Building

● Algorithms Used:

○ Linear Regression

○ Random Forest Regressor

● Model Selection Rationale:

○ Linear Regression: interpretable and fast

○ Random Forest: robust to overfitting, handles mixed data types well

● Train-Test Split: 80% training, 20% testing

● Evaluation Metrics:

○ MAE, RMSE, R^2 Score

9. Visualization of Results & Model Insights

● Feature Importance: Bar plots from Random Forest

● Model Comparison: MAE, RMSE, and R^2 for both models

● Residual Plots: Prediction errors vs. actual grades

● User Testing: Integrated model into Gradio interface

10. Tools and Technologies Used

● Programming Language: Python 3

● Notebook Environment: Google Colab

● Key Libraries: pandas, numpy, matplotlib, seaborn, plotly, scikit-learn, Gradio

11. Team Members and Contributions

Data cleaning: (B.THIRUPATHI)

• Mean, median, or mode imputation for numerical features

• Mode imputation for categorical features

EDA: (M. SENTHIL KUMAR)

• Class imbalance awareness

• Bias detection

Feature engineering: (S.SATHISH KUMAR)

• Average transaction amount per user

• Transaction frequency

Model development:

• Algorithm selection, handling class imbalance

• Performance metrics analysis