GitHub link: https://github.com/senthil7788

TITTLE: GUARDING Transactions with AI-Powered Credit Card Fraud Detection and Prevent

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/senthil7788/Project-phase2

5. Data Preprocessing

Data Collection:

- Transaction ID, Cardholder details, Merchant details, Transaction amount, Timestamp, Transaction type, Device details, Geolocation

Data Cleaning:

- Handle missing values using imputation or dropping rows/columns with excessive missing values.

6. Exploratory Data Analysis (EDA)

Univariate Analysis:

- Mean, Median, Mode, Std Deviation, Min & Max

Visualizations:

- Histograms, Box Plots, Density Plots

Bivariate & Multivariate:

- Correlation matrix, Scatter plots, Grouped bar charts

Key Insights:

- G1 and G2 are the strongest indicators of G3

- More study time correlates with higher G3

- More failures/absences lead to lower scores

7. Feature Engineering

Transaction-Based Features:

- Transaction Amount Differences: Deviation from average transaction

- Why: Unusual amounts can indicate fraud

8. Model Building

Algorithms Used:

- Linear Regression, Random Forest Regressor

Rationale:

- Linear Regression: simple, interpretable

- Random Forest: handles non-linearity, robust

Train-Test Split:

- 80/20 split with reproducibility

Evaluation Metrics:

- MAE, RMSE, R^2 Score

9. Visualization of Results & Model Insights

Feature Importance:

- G1, G2 most important followed by study time and failures

Model Comparison:

- Random Forest better performance than Linear Regression

Residual Plots:

- Checked for bias

User Testing:

- Gradio interface for testing predictions

10. Tools and Technologies Used

- Python 3, Google Colab
- pandas, numpy, matplotlib, seaborn, plotly, scikit-learn, Gradio

11. Team Members and Contributions

- Data cleaning: B.THIRUPATHI - mean, median, mode imputation; KNN imputation for advanced cases
- EDA: M. SENTHIL KUMAR - addressed class imbalance, bias detection
- Feature Engineering: S.SATHISH KUMAR - average transaction amount, transaction frequency
- Model Development - algorithm selection, class imbalance handling, appropriate evaluation