CREDIT CARD FRAUD TRANSACTION DETECTION

Domain: Data Science

By: Navina.M

NAVINA.M

Fraud Detection Project Documentation

1. Project Overview

1.1 Objective

The primary goal of this project is to develop a fraud detection model using a credit card transaction dataset to enhance security measures. The objective is to create a robust system capable of identifying and preventing fraudulent activities.

1.2 Dataset

The dataset is sourced from Kaggle and is available at https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/. It comprises credit card transactions, and our aim is to build a model that can effectively distinguish between legitimate and fraudulent transactions.

2. Exploratory Data Analysis (EDA)

2.1 Data Exploration

Conducted an extensive exploration of the dataset to understand its structure and characteristics. Key visualizations included:

- Count plots for transaction classes to assess class imbalances.
- Histograms for transaction amounts and times to identify distribution patterns.
- Box plots for transaction amounts based on class to detect potential outliers.

2.2 Findings

Identified the following key findings:

- The dataset exhibits a class imbalance, with a significantly higher number of legitimate transactions compared to fraudulent ones.
- Transaction amounts vary widely, and there are potential outliers.

3. Data Preprocessing

3.1 Cleaning

Performed the following cleaning task:

- Addressed outliers in the transaction amounts through a robust method.

4. Feature Engineering

4.1 Feature Selection

Identified relevant features for fraud detection based on insights from EDA. Selected features include:

- Transaction amount.
- Transaction time.
- Other relevant transaction features.

4.2 Transformation

Created new features or transformations to enhance model performance:

- Engineered features such as transaction amount percentiles to capture additional information.

5. Model Selection

5.1 Algorithms Considered

Explored various algorithms for fraud detection:

- Logistic Regression
- Decision Trees
- Random Forest

5.2 Chosen Algorithm

Selected Decision Tree Classifier due to its interpretability and capability to capture non-linear relationships within the data.

6. Model Training

6.1 Dataset Split

Split the dataset into training and testing sets using an 80-20 split ratio to ensure a robust evaluation.

6.2 Training

Trained the Decision Tree Classifier on the training set using default hyperparameters.

7. Model Evaluation

7.1 Performance Metrics

Evaluated the model's performance on the test set using the following metrics:

- Precision
- Recall
- F1 Score
- Accuracy

7.2 Results

Obtained promising results with an accuracy of 93.45%, indicating the model's effectiveness in identifying fraudulent transactions.

8. Hyperparameter Tuning

8.1 Fine-Tuning

Conducted hyperparameter tuning for the Decision Tree Classifier using grid search to optimize the model's performance. Explored parameters such as:

- Maximum depth
- Minimum samples split
- Minimum samples leaf

9. Prediction

9.1 Model Application

Applied the tuned Decision Tree Classifier to make predictions on the test dataset.

9.2 Analysis

Analyzed model predictions, reviewed misclassifications, and identified potential areas for improvement.

10. Documentation

10.1 Approach

Documented the overall approach, methodologies, and challenges faced during the project. Emphasized the importance of understanding the business context and the implications of false positives and false negatives.

10.2 Explanations

Provided clear explanations for choices made regarding algorithms, features, and evaluation metrics. Addressed the reasoning behind data preprocessing steps and the significance of specific features in fraud detection.

11. Code

11.1 Jupyter Notebook

Shared a Jupyter Notebook containing the entire project code. Included comments and markdown cells for clarity and ease of understanding.

11.2 Code Snippets

Included key code snippets for each step of the project, focusing on data preprocessing, feature engineering, model training, and evaluation.

12. Conclusion

12.1 Key Findings

12.1.1 Exploratory Data Analysis

- Identified class imbalance and potential outliers.
- Gained insights into the distribution of transaction amounts.

12.1.2 Data Preprocessing

- Addressed missing values, outliers, and standardized features.
- Ensured the dataset is ready for modeling.

12.1.3 Feature Engineering

- Selected relevant features and engineered new ones.
- Applied transformations for improved model performance.

12.1.4 Model Selection

- Explored various algorithms and chose Decision Tree Classifier.
- Emphasized interpretability and non-linear relationship capturing.

12.2 Future Improvements

12.2.1 Feature Importance Analysis

- Conduct deeper analysis of feature importance.

- Refine feature selection based on importance metrics.

12.2.2 Real-time Monitoring

- Explore the implementation of real-time monitoring for dynamic fraud detection.

12.3 Overall Project Impact

- Established a foundation for an effective fraud detection system.

- Demonstrated the effectiveness of the Decision Tree Classifier.

- Provided comprehensive documentation for knowledge transfer and future enhancements.

In summary, the fraud detection model presented in this project represents a significant step towards creating a secure and reliable system for credit card transaction monitoring. Ongoing efforts to refine and expand the model will contribute to staying ahead of emerging fraudulent tactics and maintaining the system's effectiveness over time.