Online Fraud Transaction Detection Report

- **Author:** Arshia Goshtasbi
- **GitHub:** @Arshiagosh

Introduction

This report focuses on the development and evaluation of a decision tree-based algorithm for detecting fraudulent online transactions. Online fraud poses significant risks to financial institutions and security systems, making accurate fraud detection crucial. Decision trees offer an interpretable and powerful method for classifying data based on feature attributes, making them well-suited for fraud detection tasks.

Dataset

The dataset contains records of online transactions and includes features such as transaction type, amount, and transaction outcome (fraudulent or not). Before modeling, data preprocessing steps such as cleaning and feature engineering were performed to ensure data quality and suitability for modeling.

Data Cleaning

Data cleaning is a critical step in any data science project to ensure the quality and integrity of the dataset. In this project, the following steps were performed for data cleaning:

- Calculation of the number of null values in each column using `df.isnull().sum()`.
- Removal of rows with null values using `df.dropna()`.
- Removal of duplicate rows using `df.drop_duplicates()`.
- Dropping of columns deemed not useful for the analysis, such as `nameOrig` and `nameDest`, using `df.drop(columns=['nameOrig','nameDest'], inplace=True)`.

Modeling Approaches

Approach 1: Basic Modeling

In this approach:

1. The first 2000 rows of the dataset were used for training, and the next 2000 rows were used for testing.
2. No additional preprocessing was performed.

Results:

- **Entropy:** The model showed signs of underfitting and low performance.
- **Gini Index:** Similar performance was observed, indicating underfitting.

Approach 2: Preprocessing with One-Hot Encoding and Binning

In this approach:

1. One-hot encoding was applied to the 'type' column to convert categorical data into numerical format.
2. Binning was used to discretize continuous features into intervals.
3. The dataset was split into training and testing sets while maintaining a balanced distribution of fraud and non-fraud classes.

Results:

- **Entropy:** Improved performance compared to Approach 1, but still suboptimal.
- **Gini Index:** Showed slightly better results compared to entropy.

Gini index Vs Entropy

The choice of the impurity measure, entropy or Gini index, can affect the performance of the decision tree model. Entropy and Gini index are two different criteria used to determine the best split at each node of the tree.

Entropy measures the impurity or randomness of a set of examples. It calculates the expected amount of information needed to determine the class of a given example. Lower entropy values indicate more homogeneous subsets, which are preferred for splitting.

Gini Index measures the degree of probability of a particular variable being wrongly classified when it is randomly chosen. Lower Gini index values indicate more pure subsets, which are preferred for splitting.

In this project, the Gini index generally performed better than entropy, particularly in Approach 2, where preprocessing techniques were applied. The Gini index tends to isolate the majority class at each split, resulting in better performance for imbalanced datasets, which is common in fraud detection scenarios.

Approach 3: Transaction-Type-Based Modeling

In this approach:

1. Data was separated based on transaction types (e.g., TRANSFER, CASH_IN).
2. Binning and splitting were performed for each transaction type individually.

Results:

- Significant improvement in performance compared to previous approaches.
- Individual models for each transaction type yielded better results.
- The following metrics were evaluated:
  - Accuracy: The overall accuracy of the model's predictions.
  - Precision: The ratio of true positives to the sum of true positives and false positives.
  - Recall: The ratio of true positives to the sum of true positives and false negatives.
  - F1-Score: The harmonic mean of precision and recall, providing a balanced measure of the model's performance. (Most important in this project)

Sklearn and XGBoost Models

Various models from Sklearn and XGBoost were also tested with one-hot encoded data, but none performed well without separating the dataset based on the transaction type.

Conclusion

This report demonstrates the importance of data preprocessing and modeling strategies in online fraud detection. By leveraging decision trees and appropriate preprocessing techniques, significant improvements in detection accuracy can be achieved. The transaction-type-based modeling approach, in particular, showed promising results, indicating its effectiveness in handling diverse transaction behaviors.

Future Directions

- Further experimentation with advanced modeling techniques.
- Exploration of ensemble methods and deep learning approaches.
- Continuous monitoring and updating of models to adapt to evolving fraud patterns.

Note: The graph of the trees in the third approach is saved to the `./Graphs` directory. You can access it here.