Project-2

Fraud Detection

Prepared by: Dushyant Kumar

Role: Machine Learning Intern

Abstract

In this project, we created a machine-learning model to determine whether a transaction is fraudulent or legitimate. To do this, we used a simulated dataset that mimics real-life fraud situations. The model we used, Random Forest, performed exceptionally well and achieved 100% accuracy, showing that it can reliably identify fraud. This report breaks down the process, the results, and what we learned along the way.

Introduction

Fraudulent transactions are a serious issue for banks and customers. They cause financial losses and undermine trust in financial systems. The faster we can identify fraud, the better we can prevent losses and keep systems secure. In this project, we used machine learning to spot fraudulent transactions based on a dataset designed to reflect common fraud patterns.

Objective

The main goal was to create a system that can look at a transaction and decide if it's fraudulent or legitimate using machine learning.

The Dataset

The dataset we used was a mix of legitimate and fraudulent transactions. It included the following details:

- **TRANSACTION_ID**: A unique number for each transaction.
- **TX_DATETIME**: The date and time the transaction happened.
- **CUSTOMER_ID**: A unique number for each customer.
- **TERMINAL_ID**: A unique number for each terminal or merchant.

- **TX_AMOUNT**: The amount of the transaction.
- **TX_FRAUD**: A label where 0 means the transaction is legitimate, and 1 means it's fraudulent.

Fraudulent Cases

1. Transactions with amounts over 220 were flagged as fraudulent.
2. Each day, two terminals were randomly chosen, and all transactions from those terminals for the next 28 days were marked fraudulent.
3. Three customers were randomly picked each day, and one-third of their transactions for the next 14 days were marked fraudulent after multiplying the amounts by 5.

Procedure

1. **Data Analysis**: First, we explored the dataset to understand its structure and looked for patterns.
2. **Data Preprocessing**: This involved fixing missing data, converting date columns to the right format, and adding new features to make fraud detection easier.
3. **Model Training**: We tried out several machine learning models, including Logistic Regression and Random Forest.
4. **Hyperparameter Tuning**: For Random Forest, we adjusted the settings (using GridSearchCV) to make it as accurate as possible.
5. **Model Evaluation**: We compared the models based on metrics like accuracy, precision, recall, and F1-score to figure out which one worked best.
6. **Feature Analysis**: We checked which features (like transaction amount or terminal ID) played the biggest role in detecting fraud.
7. **Testing**: Finally, we tested the model on new transaction data to see how well it could classify them.

Results

1. **Accuracy**: The Random Forest model achieved 100% accuracy, which means it didn't make a single mistake.
2. **Detailed Analysis**:
    - Fraud correctly identified: 2861 transactions.
    - Legitimate transactions correctly identified: 347,970 transactions.
    - Legitimate transactions wrongly flagged as fraud: 0.
    - Fraud missed by the model: 0.

3. **Important Features**: The top features that helped the model identify fraud were the transaction amount, the terminal ID, and customer spending habits.

Discussion

The Random Forest model worked perfectly on this dataset, which is exciting! However, this dataset was simulated and simpler than real-world data. Fraud in real life is much more complex and often harder to spot. So, while the results here are great, the model might not perform as well on real-world data without further adjustments. In the future, we could test it on actual financial datasets and add features that capture more subtle fraud patterns.

Conclusion

This project successfully built a machine learning model that can detect fraud with extremely high accuracy. The feature analysis gave us valuable insights into what drives fraud detection, like transaction amounts and spending patterns. Moving forward, we could focus on applying this model to real-world cases or even integrating it into systems for real-time fraud detection.