Transaction Fraud Detection

A project report

submitted in fully fulfilment of the requirements for the award of the degree of

Master of Techonology

in

Computer Science and Engineering

By

Name of Candidate: RAMJI KUMAR

Course: MTech

Semester: 1st Semester

Roll No: MT2024123

Name of Candidate: Ankit Sharma

Course: MTech

Semester: 1st Semester

Roll No: MT2024022

Under the Supervision of

Teaching Assistance (TA)

Abstract

In this project, we implemented custom machine learning classification algorithms from scratch, without relying on existing library functions, to demonstrate an in-depth understanding of core computational concepts. The algorithms implemented include Logistic Regression, Decision Tree, Support Vector Machine (SVM), and Random Forest. Each model was developed with a focus on mathematical principles

The models were tested and validated using a synthetic dataset divided into training and validation sets to ensure proper evaluation. A consistent framework was maintained for measuring model performance, with accuracy as the primary metric. The project not only highlights the computational mechanics behind these widely used algorithms but also provides insights into their comparative performance on a sample dataset.

TABLE OF CONTENTS

Objective

The primary objective of this project is to develop and implement core classification algorithms from scratch to gain a comprehensive understanding of their mathematical foundations and computational mechanisms. Specifically, the goals are

1. **Algorithm Development**: To design and implement popular machine learning classification models, including Decision Tree, and Random Forest, without using predefined library functions.

2. **Performance Evaluation**: To evaluate and compare the performance of these models on a dataset by computing accuracy and analysing their effectiveness for classification tasks.

3. **Practical Understanding**: To bridge the gap between theoretical concepts and practical implementation by coding algorithms manually and understanding their behaviours under different conditions.

5. **Future Scalability**: To lay the foundation for further exploration of machine learning algorithms, including scaling them for large datasets and integrating advanced techniques for feature selection, optimization, and real-world applications.

Benefits of Transaction Fraud Detection

The benefits of a transaction fraud detection algorithm are significant for financial institutions, e-commerce platforms, and other entities dealing with financial transactions. Some of the key benefits include

1. **Prevention of Financial Losses**:
    - **Early Detection**: Algorithms can quickly identify unusual or suspicious transaction patterns, often before they result in significant financial loss. This helps in preventing fraud from escalating.
    - **Mitigation of Losses**: By detecting fraudulent transactions early, institutions can mitigate potential losses and protect both customers and businesses.
2. **Enhanced Security**:
    - **Real-Time Monitoring**: Real-time fraud detection algorithms monitor transactions as they occur, allowing for immediate intervention. This reduces the risk of unauthorized transactions slipping through unnoticed.
    - **Anomaly Detection**: Advanced techniques like machine learning and statistical models can identify anomalies in transaction patterns that may not be apparent through traditional rules-based systems.
3. **Customer Trust and Confidence**:
    - **Protecting Customer Data**: Effective fraud detection algorithms protect sensitive customer information from being compromised, which enhances customer trust and confidence in the service.
    - **Customer Support**: Early detection allows for prompt customer support intervention, helping customers quickly resolve issues related to fraudulent transactions.
4. **Reduction in False Positives**:
    - **Optimized Accuracy**: Machine learning-based algorithms can learn from historical data, improving their accuracy over time and reducing false positives—transactions incorrectly flagged as fraudulent.

Description

A Transaction Fraud Detection system is designed to identify and mitigate fraudulent activities within financial transactions, particularly for businesses dealing with credit card transactions, online payments, and e-commerce. The goal of such a system is to ensure the security and integrity of transactions by detecting suspicious patterns and anomalies that may indicate fraudulent behaviour.

Key Components:

1. Dataset:

- **Input Data:** The dataset typically contains various features of transactions such as **transaction** (Cash_out,Cash_In,Payment,Debit,Transfer,Trasscation),**amount**,**transactionType**, **newBalance** , **OldBalance** , **userID**, Step, Amount etc.
- Output Label: The target variable in the dataset is often binary (**fraudulent** or **non-fraudulent**) based on past transaction records.

2. Methodologies:

- **Feature Engineering**: The process involves selecting, transforming, and creating new features from the raw data to enhance the model's predictive capability. This might include statistical features, aggregation of transaction data, or transformation for anomaly detection.
- **Machine Learning Models**: Various algorithms can be used for fraud detection:
    - **Decision Trees** and **Random Forests**: Effective for capturing complex patterns in data and reducing overfitting.

3. Workflow:

1. **Data Collection**: Gathering a comprehensive dataset that includes past transaction details, demographic data, and other relevant features.
2. **Data Preprocessing**: Cleaning the dataset by handling missing values, normalizing data, encoding categorical features, and scaling numerical features.
3. **Model Training**: Training the selected machine learning models using the preprocessed data. Models are evaluated using techniques such as cross-validation to tune hyperparameters and assess performance.
4. **Model Evaluation**: Assess the model's performance using metrics like accuracy, precision, recall, F1-score, and AUC-ROC. This helps in understanding the trade-off between detection rate and false positive rate.

Preprocessing

1**. Data Collection:**

- **Input Data:** The dataset typically contains various features of transactions such as **transaction** (Cash_out,Cash_In,Payment,Debit,Transfer,Trasscation),**amount**,**transactionType**, **newBalance** , **OldBalance** , **userID**, Step, Amount etc.
- Output Label: The target variable in the dataset is often binary (**fraudulent** or **non-fraudulent**) based on past transaction records.

2. **Data Cleaning**:

One-Hot Encoding:

- **Objective**: Convert categorical features (e.g., transaction type) into a format that is suitable for machine learning models.
- **Action**: Use pandas.get_dummies() to convert categorical columns into binary columns.
    - **Example**: If transaction types are represented as strings (e.g., "credit", "debit"), use one-hot encoding to convert them to binary columns like transaction_credit and transaction_debit.

3. Outlier Detection:

- **Objective**: Identify and remove outliers to prevent the model from being influenced by anomalies that do not represent real fraudulent patterns.
- **Action**: Use IQR (Interquartile Range) method to filter out outliers.
    -

\text{lower_bound} = Q1 - 1.5 \times IQR]
upper_bound=Q3+1.5×IQR\text{upper_bound} = Q3 + 1.5 \times IQRupper_bound=Q3+1.5×IQR

4 .Filtering: Remove rows where the feature values are outside these bounds.

Dataset Splitting:

- **Train-Test Split**:

- **Objective**: Split the dataset into training and testing sets to evaluate the performance of the model accurately.
- **Ratio**: A common ratio is 80% training and 20% testing.

3. Data Transformation:

5. **Feature Scaling**:

1 **Objective**: Normalize or standardize numerical features to bring all values onto the same scale, which improves the convergence speed and stability of machine learning models.

2 **Methods**:

1. **Normalization (Min-Max Scaling)**: Rescale features between 0 and 1.

1.

X' = \frac{X - min(X)}{max(X) - min(X)}]

2. **Standardization (Z-score Normalization)**: Rescale features to have a mean of 0 and a standard deviation of 1.

1.

X' = \frac{X - \text{mean}(X)}{\text{std}(X)}]

Model Design

Overview of Models:

- **Objective:** In a transaction fraud detection context, selecting the right model is crucial to accurately distinguish between legitimate and fraudulent transactions. The following models were selected based on their effectiveness in handling imbalanced datasets, their ability to capture complex patterns, and their computational efficiency.

1. Decision Tree:

- **Objective:** Model the decision boundaries between fraudulent and legitimate transactions using a tree structure.
- **Methodology:** A decision tree recursively splits the dataset based on feature values that maximize the information gain. The tree depth was controlled to avoid overfitting.

2. Random Forest:

- **Objective**: Ensemble learning method that constructs multiple decision trees and merges their predictions.
- **Methodology**: Each tree is trained on a bootstrapped sample of the dataset with random subsets of features. The final prediction is made by averaging the results from all trees.
- **Evaluation**:
- **Accuracy**: Achieved an accuracy of X.X% on the test dataset.
- **Confusion Matrix**: Included to evaluate the classification performance.

3. Model Selection Criteria:

- **Factors Considered:**
  - Accuracy: Ability to correctly classify fraudulent and legitimate transactions.
  - ROC Curve and AUC: To measure how well the model can distinguish between classes.
  - Precision, Recall, F1 Score: For understanding the trade-offs between sensitivity (recall) and specificity (precision).

- Computational Efficiency: Processing speed and memory usage, especially important when deploying models in real-time fraud detection systems.

Conclusion

This report provides a detailed examination of various models for transaction fraud detection, their methodology, and performance evaluation. By understanding the strengths and limitations of each model, one can make informed decisions regarding model selection and deployment, ensuring effective and reliable fraud detection in real-time systems.