Autor: Sofie Meyer (Bc. Yegor Dubovoy)

Databázové systémy v chemii a Procedurální programování

Main goals:

1. Create a relation database for banking system with Microsoft SQL Server.
2. Import data from Kaggle dataset to new database.
3. Use this database in Matlab for data analysis and classification model implementation in order to predict fraudulent transactions.

Dataset:

https://www.kaggle.com/datasets/computingvictor/transactions-fraud-datasets

Creating database:

In order to start, I made an ER diagram using Crow's Foot notation to visualize the structure of the future database. The diagram included the main entities: Users, Cards, Transactions, Merchants, and MCC Codes. Each entity was designed with its relevant attributes and relationships:

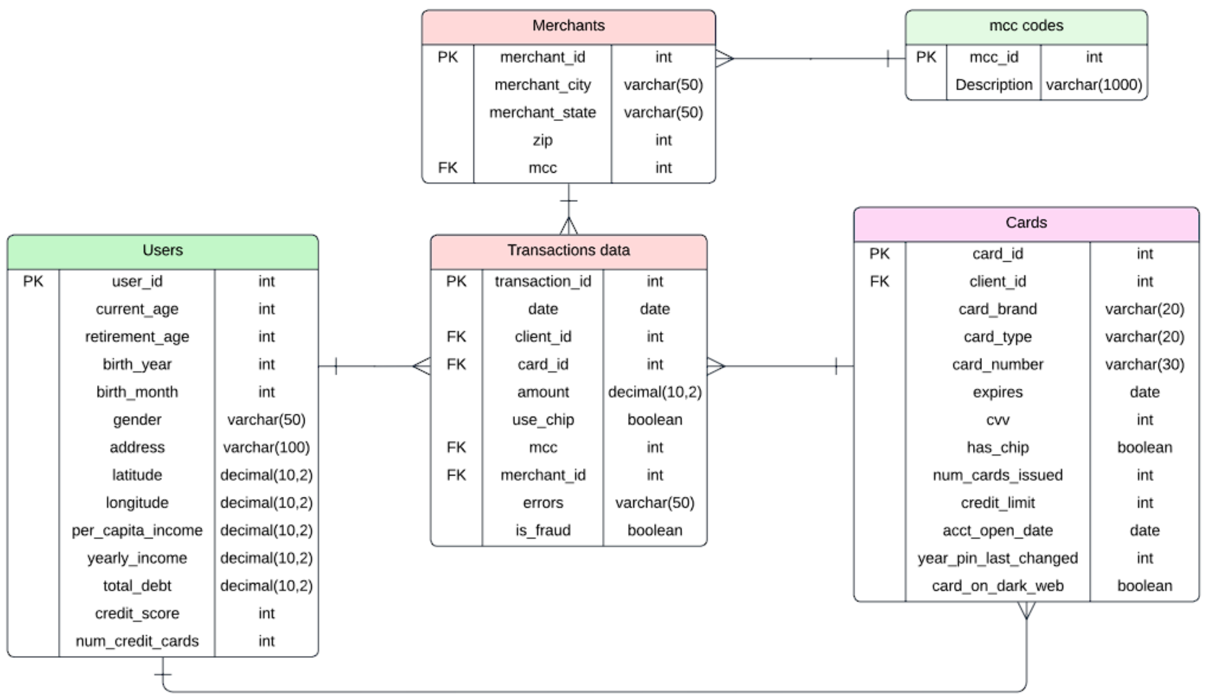
- **Users**: Attributes like user_id, current_age, yearly_income, and credit_score were defined to store key user details.
- **Cards**: Details such as card_id, card_brand, card_type, and card_on_dark_web were incorporated to track card information.
- **Transactions**: Attributes like transaction_id, amount, use_chip, and is_fraud were used to store transactional data and associated features.

- **Merchants**: Fields like merchant_id, merchant_city, and zip were designed to hold merchant-related details.
- **MCC Codes**: The mcc_id and description fields were included to classify transaction categories.

Relationships between the entities were explicitly represented. For example:

- Each transaction was linked to a user and a card (client_id and card_id as foreign keys).
- Transactions were also connected to merchants and MCC Codes (merchant_id and mcc).
- Merchants were related to MCC Codes to classify their type of business.

This ER diagram served as the foundation for building the database, helping to organize and normalize data efficiently while ensuring all relationships and dependencies were accounted for.



In order to create the server, **SQLEXPRESS** was utilized as the local server environment. This setup provided a robust and accessible platform for managing the database during the initial stages. After establishing the server and configuring the necessary database parameters, the development work was seamlessly transitioned to **DataGrip**.

Autor: Sofie Meyer (Bc. Yegor Dubovoy)

DataGrip was used for further development due to its advanced SQL editor, efficient database navigation, and powerful data visualization tools.

After that, the tables were created using a **table creation query** designed in alignment with the **ER diagram**. Each table was structured to reflect the relationships and constraints specified in the diagram, ensuring data consistency and integrity.

Once the database schema was established, the data was imported into the tables using the **DataGrip GUI interface**. This tool facilitated a smooth and user-friendly process for loading data into the database. Each table was populated with its respective dataset, adhering to the predefined schema

Data analysis in Matlab:

When the database setup was complete, the focus shifted to **MATLAB** to begin the data analysis process. MATLAB was chosen for its robust analytical capabilities, offering tools for efficient data extraction, processing, and visualization. The connection to the database was established using **ODBC**, allowing seamless querying and retrieval of the stored data for further analysis and modeling.

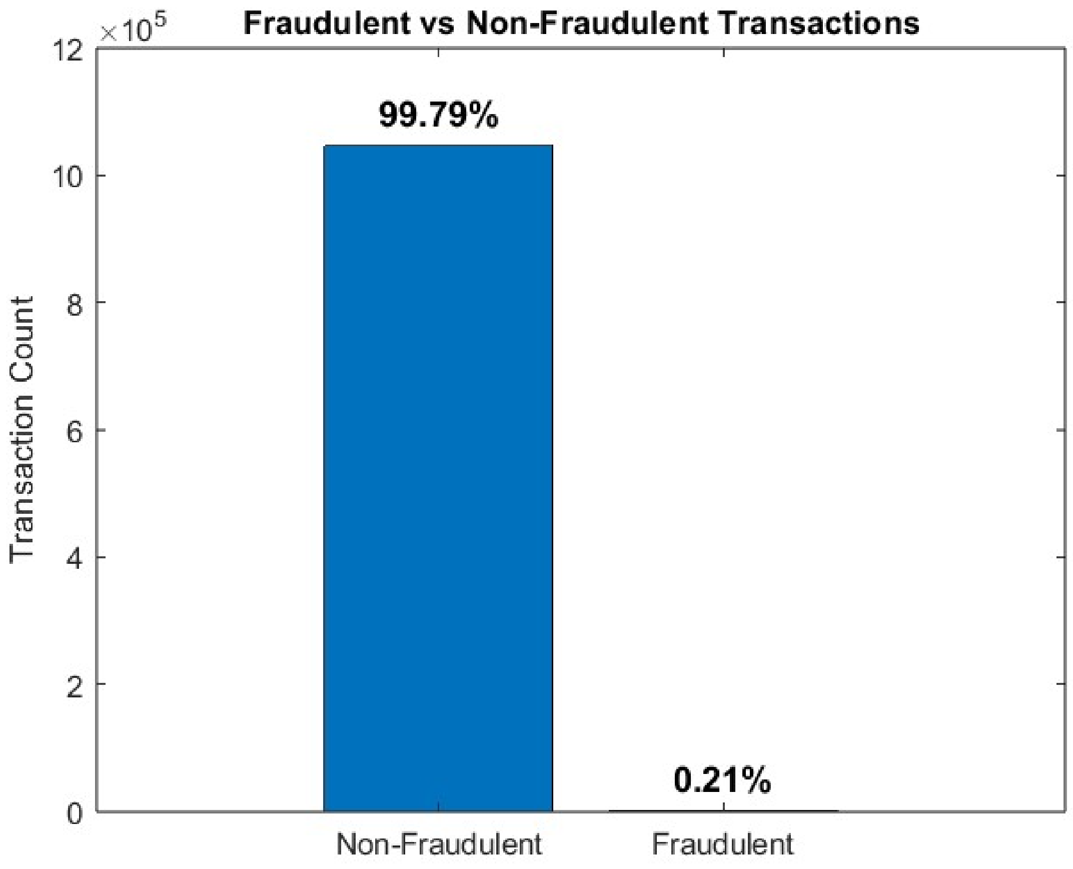


Figure 1. Fraudulent vs. Non-Fraudulent transactions

This bar chart provides a visualization of the distribution between fraudulent and non-fraudulent transactions in the dataset. As expected in real-world financial datasets, the majority of transactions are non-fraudulent, comprising **99.79%** of the total, while fraudulent transactions account for only **0.21%**. This stark imbalance highlights the challenges of fraud detection, where accurately identifying rare fraudulent cases is crucial while avoiding misclassifications in most legitimate transactions.

Autor: Sofie Meyer (Bc. Yegor Dubovoy)

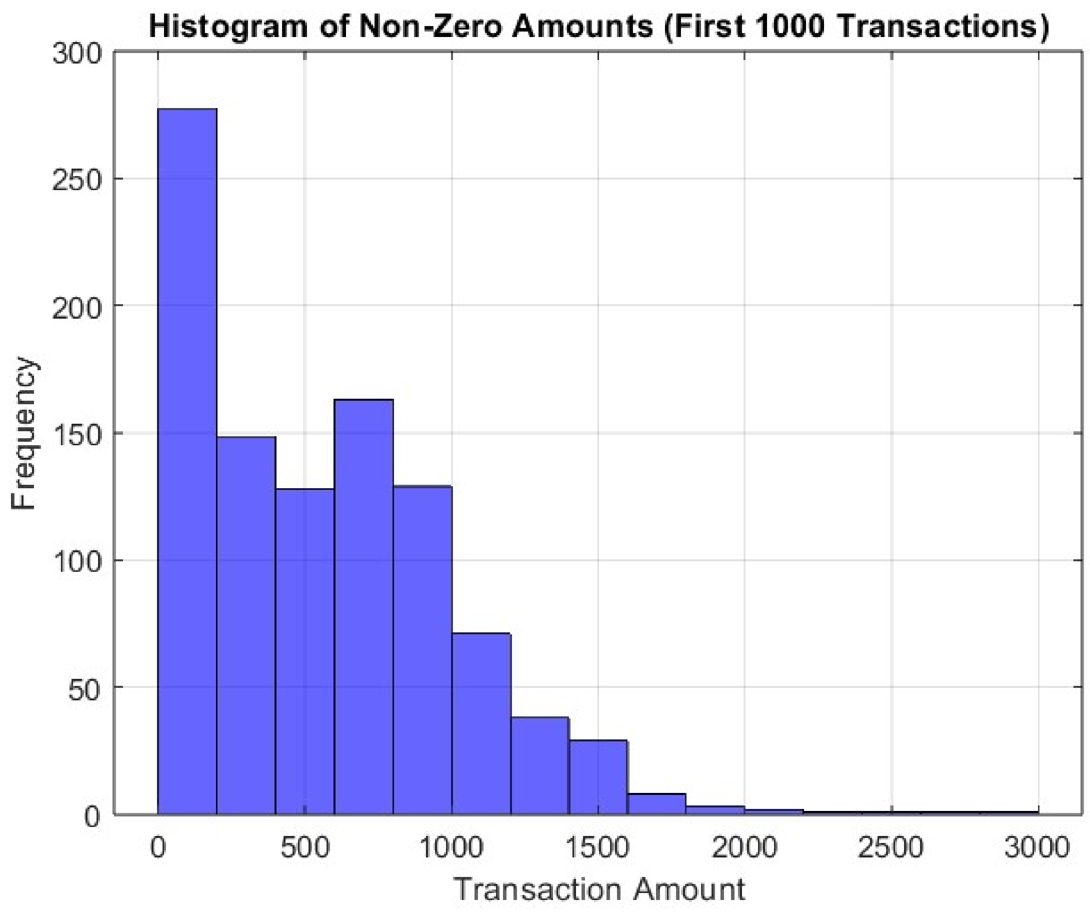


Figure 2. Histogram of transactions amount

This histogram illustrates the distribution of non-zero transaction amounts in the first 1,000 transactions. The majority of transactions are clustered around lower values, with a sharp decline as the transaction amount increases. This pattern is consistent with real-world scenarios, where small-to-medium-sized transactions dominate financial datasets, while larger transactions occur less frequently.

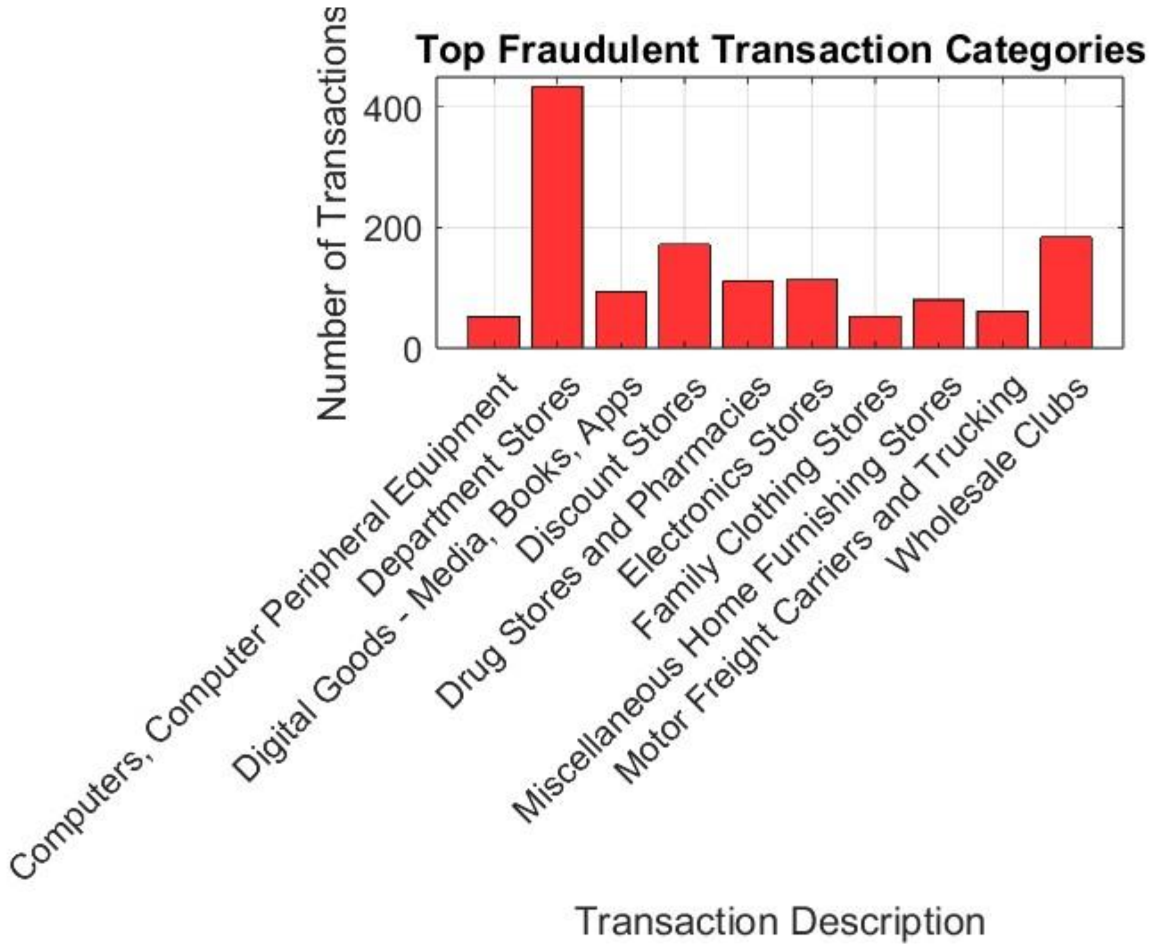


Figure 3. Fraudulent transactions by categories

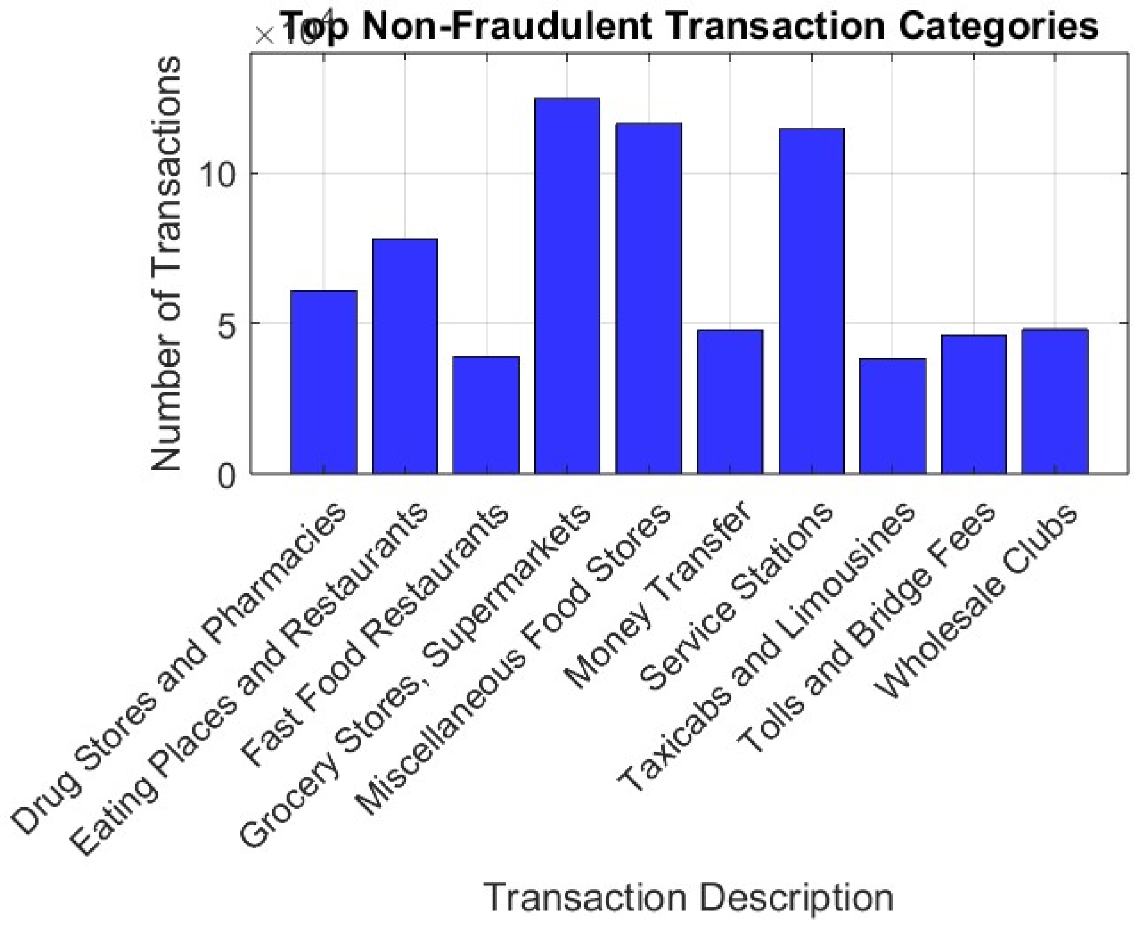


Figure 4. Non-Fraudulent transactions by categories

Based on the bar charts:

1. **Fraudulent Transactions by Categories**:
   - The most common fraudulent transaction category is **Department Stores**, which stands out significantly compared to other categories.
   - Other frequent categories include **Discount Stores**, **Drug Stores and Pharmacies**, and **Wholesale Clubs**.
   - Categories such as **Electronics Stores** and **Home Furnishing Stores** have lower counts but still show notable fraudulent activity.
2. **Non-Fraudulent Transactions by Categories**:

    - The distribution for non-fraudulent transactions is more balanced across categories.
    - **Money Transfer**, **Miscellaneous Food Stores**, and **Service Stations** dominate this group.
    - Other common categories include **Grocery Stores**, **Restaurants**, and **Taxis**.

Observations:

- Fraudulent transactions tend to cluster around specific retail categories, especially department stores and discount outlets.
- Non-fraudulent transactions reflect a broader range of typical consumer activities like grocery shopping, dining, and transportation.

Classification models implementation:

Models were developed using the **Matlab Classification Learner** interface, leveraging the features **'Amount'**, **'use_chip'**, and **'encoded_category'**, with **'is_fraud'** serving as the target variable. To ensure the models avoided overfitting, a **5 K-Folds Cross-Validation** approach was employed.

Models Created:

1. **Decision Tree**:
    - A straightforward model that divides data into branches based on feature thresholds.
    - Useful for interpretability and understanding key splitting criteria.
2. **Random Forest**:
    - An ensemble method built on multiple decision trees.
    - Handles imbalanced data better, especially when combined with class weighting.
    - Provides feature importance rankings to evaluate the impact of each predictor.
3. **RUSBoost Ensemble**:
    - Specifically designed for highly imbalanced datasets like this one.

  - Combines random under-sampling (RUS) with boosting to give more attention to minority class predictions.

Decision Tree model:

The **Decision Tree** model was built and evaluated using **Matlab's Classification Learner**. Below is the detailed information about the model parameters and results:

Model Configuration:

- **Preset**: Fine Tree
- **Maximum Number of Splits**: 100
- **Split Criterion**: Gini's Diversity Index

Misclassification Costs:

A **custom cost matrix** was used to penalize false negatives more heavily, reflecting the higher importance of identifying fraudulent transactions:

- Cost of predicting 0 (Non-Fraudulent) for true 1 (Fraudulent): 10
- Cost of predicting 1 (Fraudulent) for true 0 (Non-Fraudulent): 1

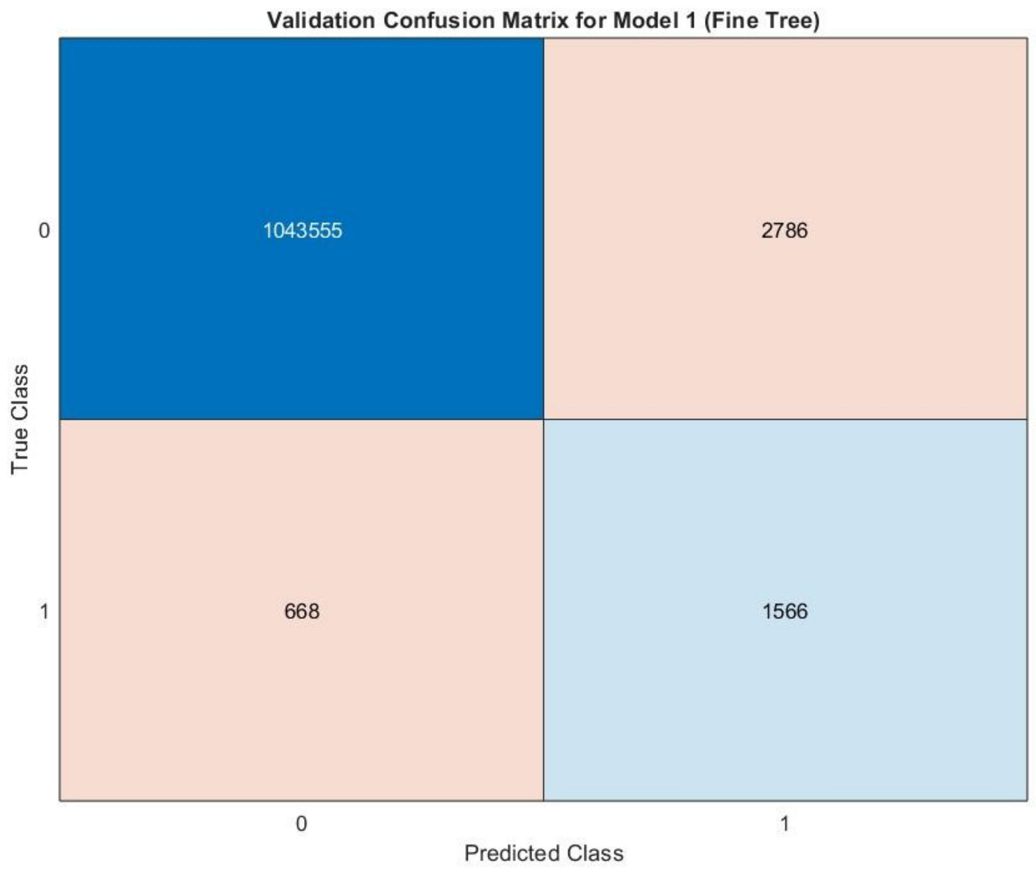


Figure 5. Validation Confusion Matrix for DecisionTree

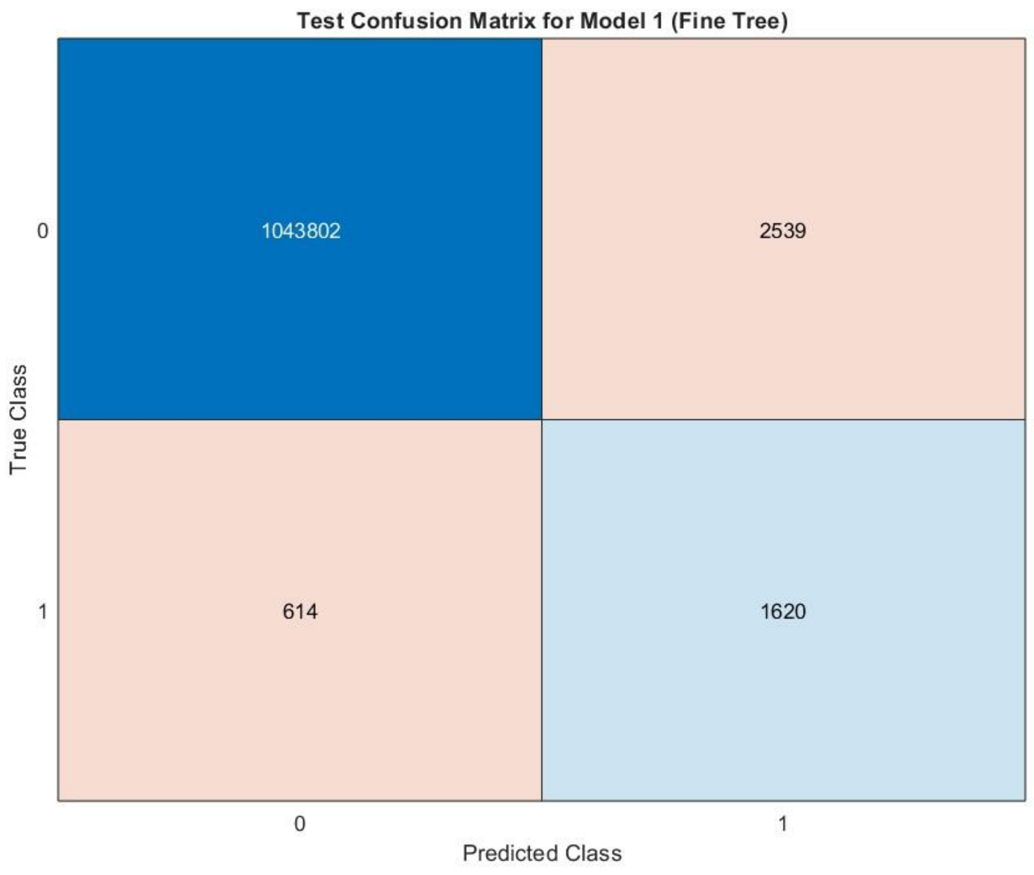


Figure 6. Test confusion matrix for DecisionTree

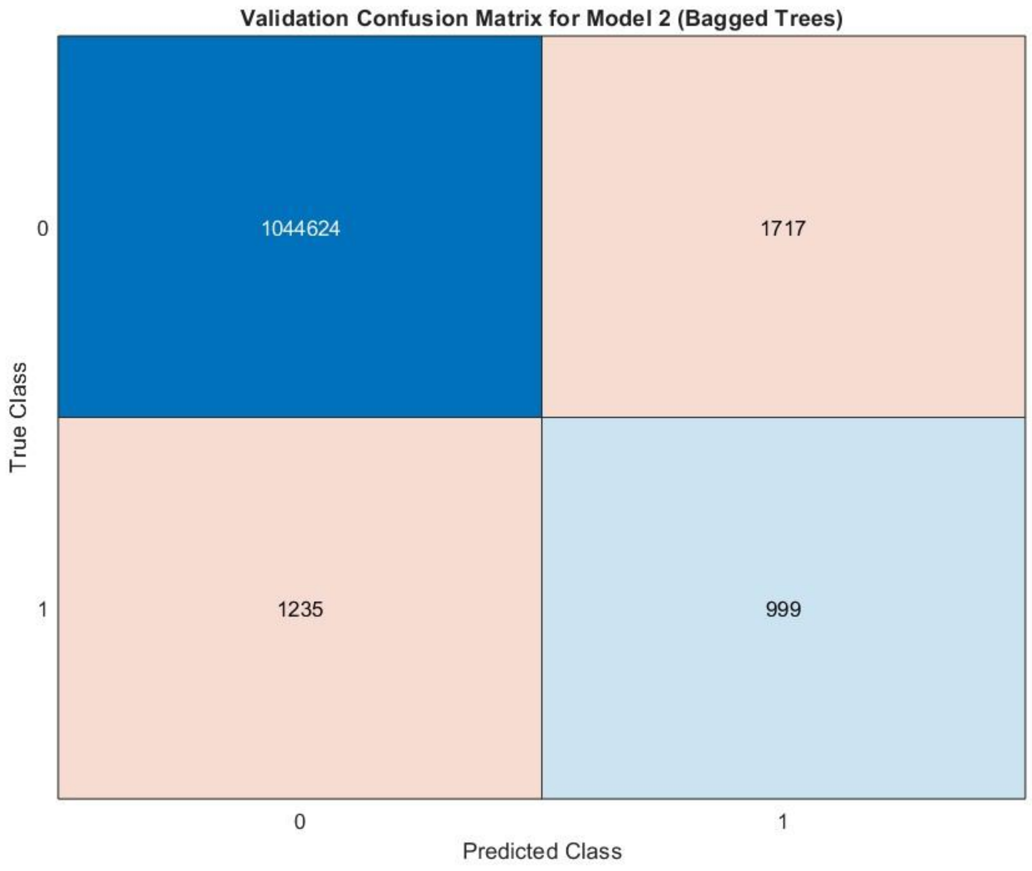
Observations:

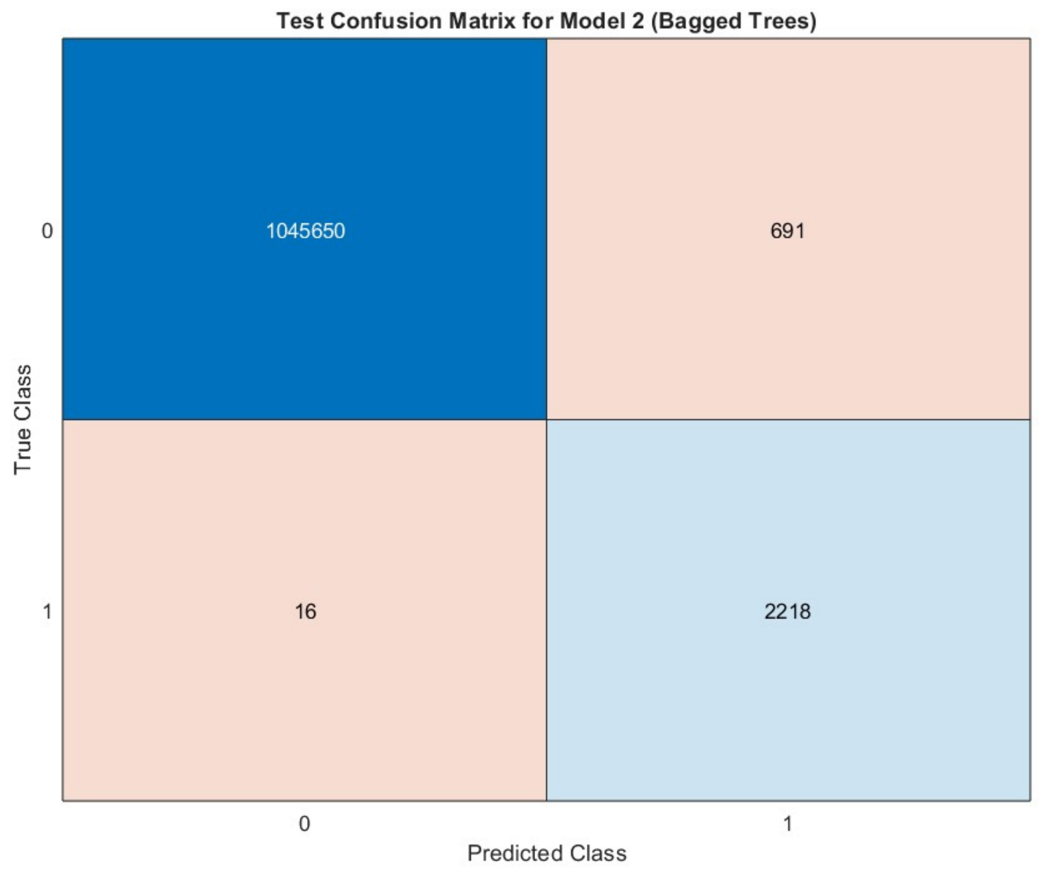
- The **True Negative rate (specificity)** is high, indicating the model is excellent at identifying non-fraudulent transactions.
- The **True Positive rate (sensitivity)** is lower, reflecting difficulty in detecting fraudulent transactions, a common challenge in imbalanced datasets.

False Positives and False Negatives are small compared to the total dataset size, which suggests the model is performing reasonably well for a highly imbalanced problem.

Random Forest model:

Autor: Sofie Meyer (Bc. Yegor Dubovoy)





Observations:

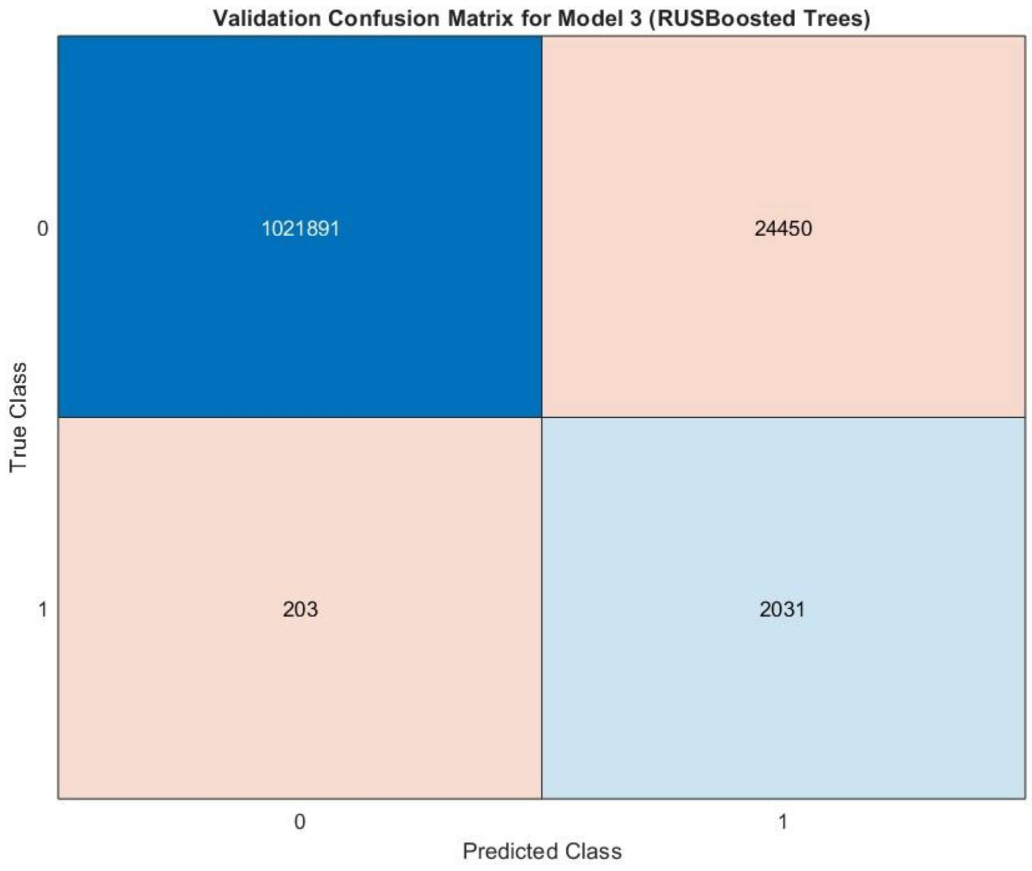
- **Accuracy on Non-Fraudulent Transactions:** Both validation and test sets show very high accuracy for classifying non-fraudulent transactions.
- **Improvement in Test Set:** The test set performance is significantly better, particularly in detecting fraudulent transactions, which is a key improvement over the validation results.
- **False Positives vs. False Negatives:** While the test set reduces FN significantly, FP also remains relatively low, balancing sensitivity and specificity.

RUSBoostingEnsemble model:

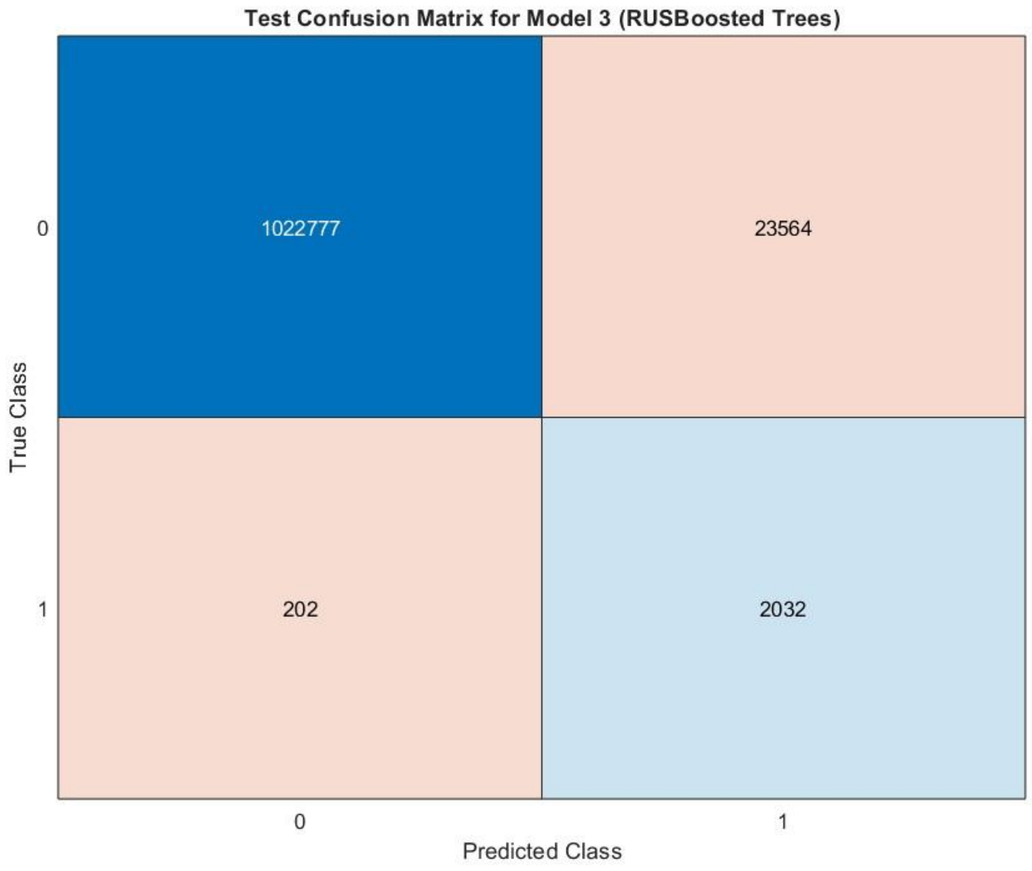
The model effectively identifies most fraudulent transactions (True Positives). The False Negative count is very low, which indicates that the model rarely misses fraudulent cases.

A relatively high number of False Positives indicates the model occasionally misclassifies legitimate transactions as fraudulent. This is expected due to the use of Random Under-Sampling to balance the dataset.

Autor: Sofie Meyer (Bc. Yegor Dubovoy)



Autor: Sofie Meyer (Bc. Yegor Dubovoy)


Test Confusion Matrix for Model 3 (RUSBoosted Trees)
True Class
Predicted Class