Fraud Detection

Abstract

This project aims to detect fraudulent transactions using Spark's MLlib. The methodology includes data acquisition, preprocessing, model training, evaluation, and the use of various performance metrics to validate the model's effectiveness.

1. Introduction

Fraud detection is critical in the financial industry to protect customers and financial institutions from losses. This project focuses on developing an efficient method for detecting fraudulent transactions using machine learning techniques implemented in Spark's MLlib.

2. Data Acquisition and Preprocessing

2.1 Data Source

The dataset used in this project consists of transaction records with labeled fraud instances. The data is sourced from Kaggle.

2.2 Data Import

The data was imported using PySpark's read.csv function, allowing for efficient loading and processing of large CSV files.

2.3 Data Preprocessing

Preprocessing steps included calculating age, binning age, extracting the hour of transaction, and calculating the distance between transaction locations. Additional steps involved log transformation of transaction amounts and city populations. The dataset was then split into features and target variables.

3. Model Development

3.1 Model Architecture

Two machine learning models were used for fraud detection:

- Logistic Regression
- Random Forest

3.2 Model Training

The models were trained on the preprocessed training data. Hyperparameters were tuned, and various performance metrics were used to evaluate the models. The training process included the following steps:

1. Indexing and one-hot encoding of categorical variables.

2. Assembling numerical and categorical features into a single feature vector.
3. Standardizing numerical features.
4. Training the models using the processed features and labels.

4. Model Evaluation

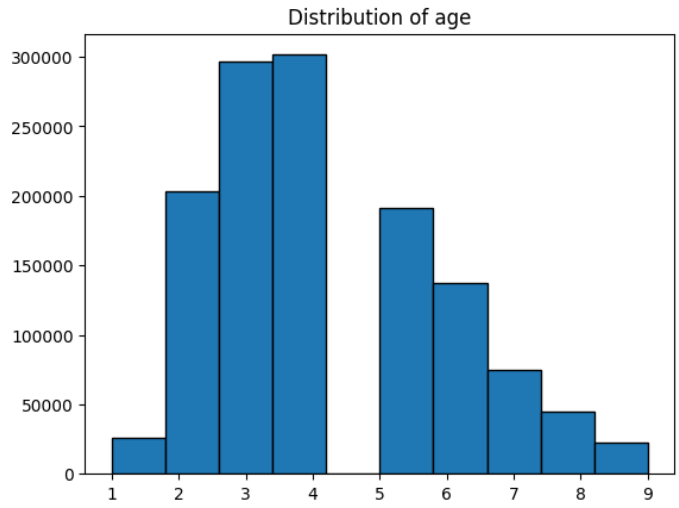
4.1 Accuracy and Loss Curves

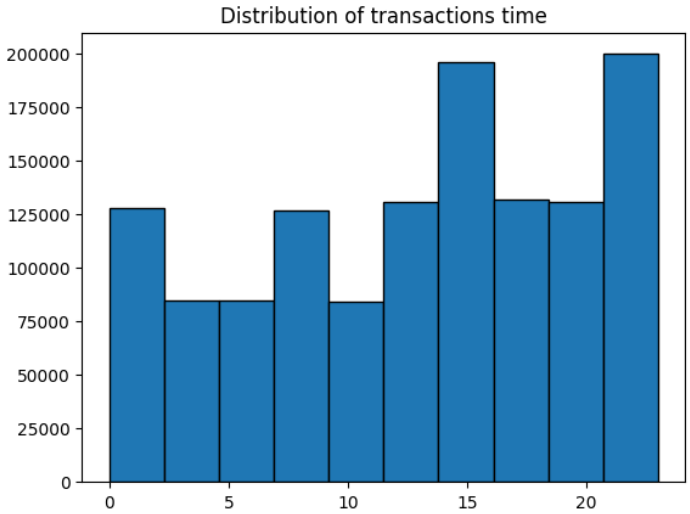
The training process was monitored using accuracy and loss curves for both training and validation datasets.

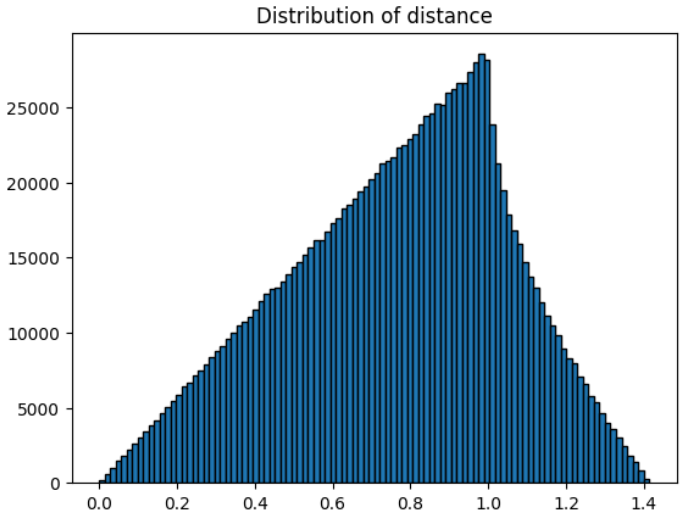
4.2 Performance Metrics

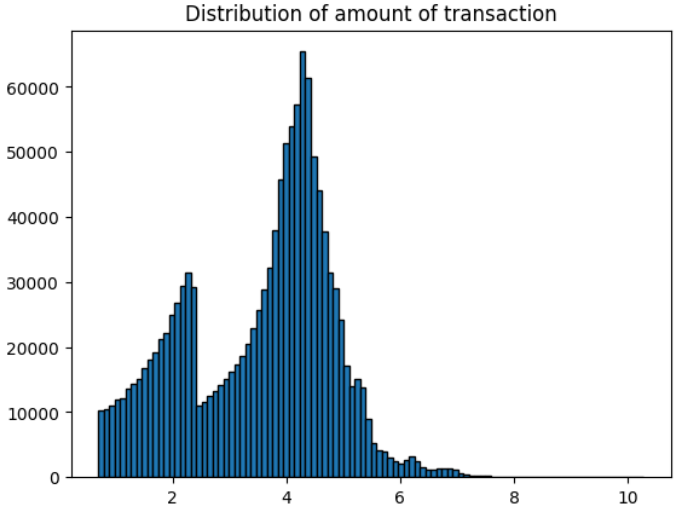
Performance metrics such as accuracy, precision, recall, and F1 score were calculated for both models to determine their effectiveness.

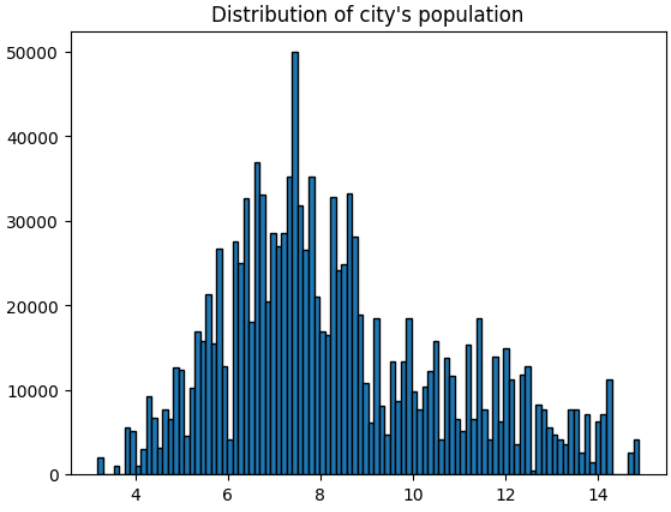
5. Graphs

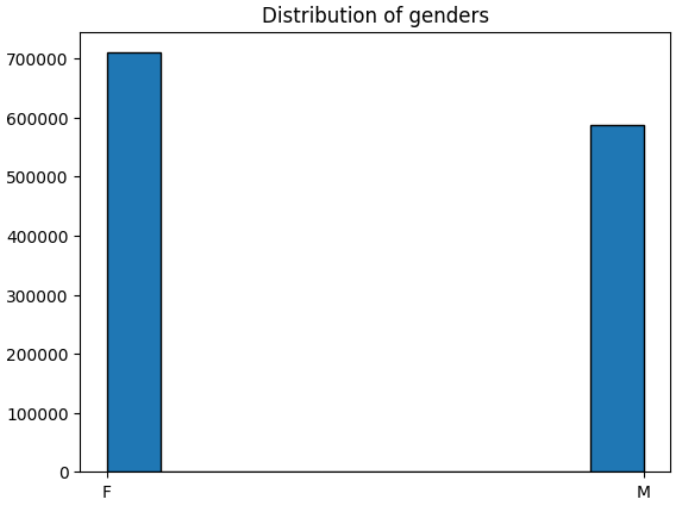





Distribution of distance
25000
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4


Distribution of amount of transaction
60000
50000
40000
30000
20000
10000
0
2
4
6
8
10


Distribution of city's population
50000
40000
30000
20000
10000
0
4
6
8
10
12
14



5. Performance Results

The following performance results were obtained:

Dummy Classifier

- **Accuracy**: 0.99
- **Precision**: 0.99
- **Recall**: 0.99
- **F1 Score**: 0.99

The dummy classifier serves as a baseline, always predicting the most frequent class, which in this case is non-fraudulent transactions. While it shows high accuracy, precision, recall, and F1 score, it is not indicative of the model's ability to detect fraud since it essentially predicts that no transactions are fraudulent.

Logistic Regression

- **Accuracy**: 0.84
- **Precision**: 0.35
- **Recall**: 0.35
- **F1 Score**: 0.35

Logistic Regression achieved an accuracy of 0.84, which indicates that the model correctly identified a high number of transactions. However, the precision and recall scores are significantly lower at 0.35, indicating that while the model is good at identifying non-fraudulent transactions, it struggles to accurately detect fraudulent ones. The low precision means a high number of false positives, and the low recall indicates a high number of false negatives.

Random Forest

- **Accuracy**: 0.97
- **Precision**: 0.60
- **Recall**: 0.60
- **F1 Score**: 0.60

Random Forest showed substantial improvement over Logistic Regression with an accuracy of 0.97. The precision and recall scores are both 0.60, indicating a more balanced performance in identifying both fraudulent and non-fraudulent transactions.

Given the higher precision and recall of the Random Forest model compared to Logistic Regression, it was chosen as the final model for fraud detection. The Random Forest model is more effective at reducing both false positives and false negatives, making it a better fit for detecting fraudulent transactions.

7. Future Scope

In future work, we plan to enhance the model's accuracy and explore advanced techniques for fraud detection. Additionally, we will develop an interface using **Streamlit** to provide a user-friendly platform for detecting fraudulent transactions in real-time. This interface will be linked with the backend model, creating a finished state ready for practical deployment in fraud detection systems.

8. Conclusion

This project successfully demonstrates the use of Spark's MLlib for fraud detection in financial transactions. The models achieved notable performance, highlighting the effectiveness of this approach.

References

- PySpark Documentation
- Public Dataset on Kaggle

Name	Roll No	Section
Abdul Moiz	22L-7468	BDS-4C
Nouveen Leghari	22L-7495	BDS-4C