FRAUD DETECTION IN BANKING SYSTEMS

DSC680

Madison Christiansen

BACKGROUND

This project aims to develop a fraud detection system for a mobile transaction bank. The banking industry is a main target for different types of financial fraud. Different type of fraud includes but are not limited to identify theft, account takeover, and credit card fraud. These fraudulent activities can result in significant financial losses for the customers and the banks. Due to the growing landscape of online banking and mobile money transferring, ensuring a secure environment for both the customer and bank is essential. The main business problem is the constant threat of these fraud activities occurring within the mobile money sphere.

The first forms of digital banking can be traced back the 1960s, and in the 1980s banks started offering services that allowed customers to access their account through their home computers. Digital banking has become very popular since with allowing customers to access accounts quickly and streamlining operations. Although with this easy access this also means that there could be an increase in banking fraud. Cybercriminals and hackers use many methods to commit digital banking frauds such as phishing, malware, and social engineering tactics.

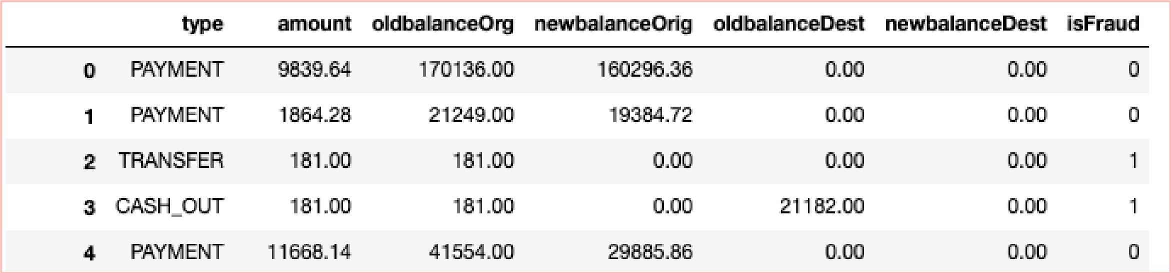
Mobile banking specifically has become popular because of the ease and convenience as most people have a smart phone. To combat the risk of fraud mobile banking apps, need to have strong authentication measures, such as biometric verification. Fraud risk is increased by weak passwords and lack of strong authentication protocols.

MEHTODS

Due to the lack of public dataset on financial services and specially the mobile money domain. PaySim is a simulated dataset from Kaggle based on a sample of real transactions from one month of financial logs from a mobile money service implemented in an African country. Many ethical considerations need to be taken when handling people private banking information. Being transparent and gaining informed consent from those who would be giving their data. Ensuring fairness and models are built to not discriminate against and specific groups.

The dataset shows a column of fraudulent transactions and those that are not. This will allow for a model to be built around what can pose as fraud in the future.

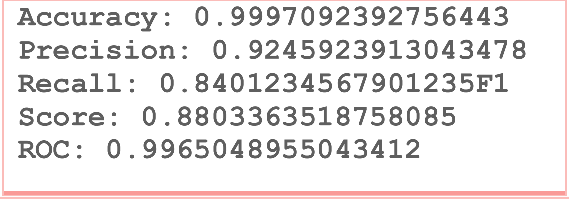
Data collection and processing involves using the PaySim dataset from Kaggle. Cleaning the data by dealing with missing values, outliers, and inconsistencies. Using feature engineering for creating or detecting features that can help identify fraud patterns. Examples: behavior patterns, past fraud, transaction amounts, frequency, historical statistics. Splitting the data into training and test sets. The training set will be used to train the model, and the test set will be used to evaluate the model's performance. Choosing an appropriate machine learning algorithms for fraud detection. For example, logistic regression, decision trees, random forest, gradient boosting, and neutral networks. Hyperparameter tuning to optimize the models hyperparameters using the validation set. Using learning rates, neutral network, and tree depth architectures to improve the model. Using the appropriate metrics to evaluate the model. Common tests include accuracy, precision, recall, F1, and ROC curve. Using past fraud detections as a threshold to determine the false positives and false negatives. Adjust thresholds and model as needed. With mobile banking evolving quick, a model needs to be evolving as well.



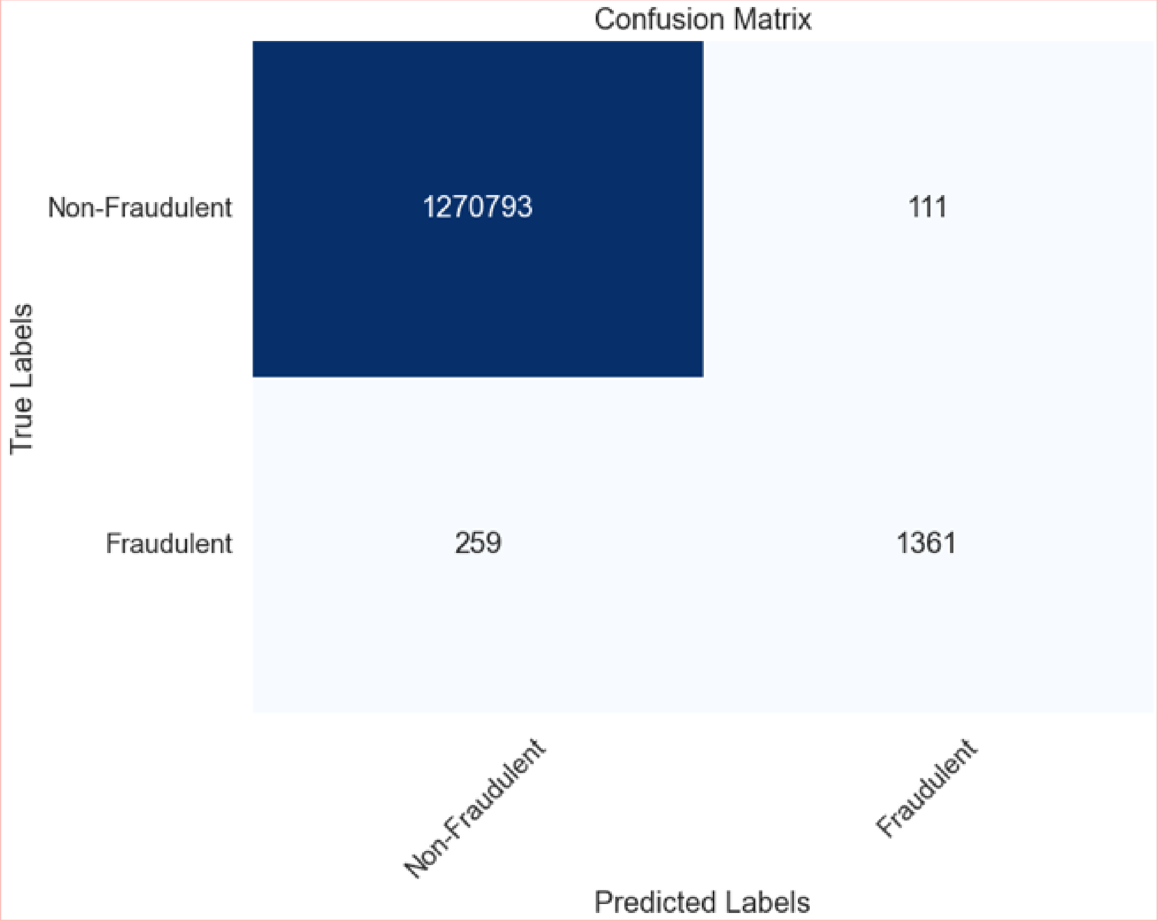
	type	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud
0	PAYMENT	9839.64	170136.00	160296.36	0.00	0.00	0
1	PAYMENT	1864.28	21249.00	19384.72	0.00	0.00	0
2	TRANSFER	181.00	181.00	0.00	0.00	0.00	1
3	CASH_OUT	181.00	181.00	0.00	21182.00	0.00	1
4	PAYMENT	11668.14	41554.00	29885.86	0.00	0.00	0

ANALYSIS

The evaluation metrics of the model indicates a strong performance in detecting fraud transactions. Based on the accuracy being 99% the model correctly classifies many transactions. The precision score of 0.925 means that the model predicts a transaction as fraud, and it is correct about 9.5% of the time. The recall score of 0.840 indicates that the model identified 84% of actual fraud transactions. The F1 score of 0.880 is a harmonic mean of the precision and recall, this provides a balanced measure of a model's performance. The ROC AUC score of 0.997 is high as well. This evaluates the model's ability to distinguish between fraud and non-fraud across different threshold values.



```
Accuracy: 0.9997092392756443
Precision: 0.9245923913043478
Recall: 0.8401234567901235F1
Score: 0.8803363518758085
ROC: 0.9965048955043412
```



CONCLUSION

This model demonstrates a strong capacity to detect fraud effectively (see Appendix A for full model code). With this being an ever-evolving system there could also be evolving fraud tactics that can bypass detection systems. Monitoring and evaluation of the model's performance on new data is crucial to ensure effectiveness in real world scenarios.

As technology continues to advance quickly and fraud tactics become more advanced, fraud detection systems need to meet the evolution. There are several options for fraud detection to be implemented in banking systems. Some options would be, AI, Advanced

Machine Learning, Behavioral Biometrics, Real-time Detection, Cross-Industry Collaboration, Continuous Adaptation.

Deploying the fraud detection system into the production environment will be beneficial to many banking systems. Understanding the audience and their needs to create a platform that could allow for ease of use. Constant monitoring and maintenance system to track system performance and scheduling regular model updates. Reposting metrics and tracking to see the systems effectiveness. Training and awareness programs for those who are involved in the systems development and customers to keep them informed with it being an ever-evolving landscape. Contingency planning is also needed, if there were to be an issue there needs to be a protocol in place to protect the customers and their information.

QUESTIONS

1. Is the model capable of adaptive learning?

   Adaptive learning refers to a machine learning model's ability to update its parameters or adapt its predictions as new data becomes available. The model will be periodically retrained using the entire dataset or a large batch of new data. As new data becomes available, we will discover that certain features become relevant in predicting fraud. Creating a feedback loop for model performance monitoring. Continuously collect feedback on the model's predictions, especially for false positives and false negatives.

2. What steps are being taken to stay ahead of emerging fraud threats?

   To stay ahead of emerging fraud threats, we can implement fraud detection systems that continuously monitor transactions, user behavior, and other relevant data sources. Along with this we can use AI algorithms to detect unusual patterns or anomalies that may indicate fraud. Staying up to date with relevant regulations and compliance requirements related to fraud prevention and data protection. Ensuring that fraud prevention measures are aligned with legal and regulatory standards.

3. How frequently are false positives occurring?

False positives occur when the system incorrectly flags a legitimate transaction or activity as fraudulent. We caught 111 in our model

4. What is the process for investigation and responding to alerts?

   When responding to alerts we walk through a few steps, alert prioritization, triage, classification, and then investigation of these alerts.

5. Are there any bias in the model?

   To assess bias in the model you collect all the demographic info, use fairness metrics, consider consent, and create visuals to see if there are disparities.

6. Is the dataset balanced?
7. How will you handle an imbalanced dataset?

   The dataset might be imbalanced due to the number of fraud vs non fraud found. What we did to minimize this risk is feature engineering, evaluation metrics to test, and threshold adjustment if needed.

8. Where did the data come from for the model?

   The data came from Kaggle which is a public dataset source.

9. What strategies are in place to monitor and update the model?

   Some effective strategies for monitoring and updating machine learning models: monitoring the distribution of input data to detect data drift. Sudden changes in data patterns can impact model performance. Set up automated alerts for significant drops in performance that may require attention. Establish feedback loops to collect user feedback and monitor model behavior in real-world scenarios.

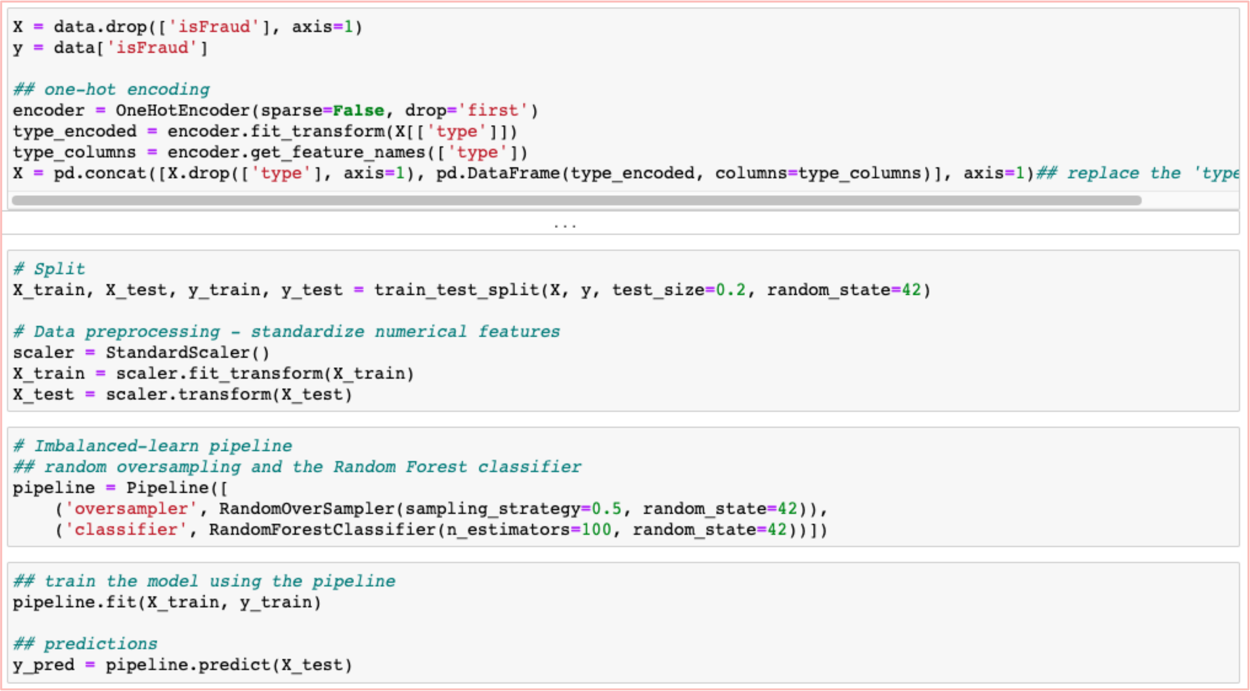
10. What are the considerations when deploying and maintaining the model in a production?

    Deploying and maintaining machine learning models in a production environment involves several critical considerations to ensure that the model performs well, remains reliable, and meets business objectives.

APPENDIX A

Model Code

This machine learning model specifically a binary classification model. Binary classification models are designed to classify data into one of two classes, and in this case the two classes are fraud and non-fraud.



```
X = data.drop(['isFraud'], axis=1)
y = data['isFraud']

## one-hot encoding
encoder = OneHotEncoder(sparse=False, drop='first')
type_encoded = encoder.fit_transform(X[['type']])
type_columns = encoder.get_feature_names(['type'])
X = pd.concat([X.drop(['type'], axis=1), pd.DataFrame(type_encoded, columns=type_columns)], axis=1)## replace the 'type
```

...

```
# Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Data preprocessing - standardize numerical features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
# Imbalanced-learn pipeline
## random oversampling and the Random Forest classifier
pipeline = Pipeline([
    ('oversampler', RandomOverSampler(sampling_strategy=0.5, random_state=42)),
    ('classifier', RandomForestClassifier(n_estimators=100, random_state=42))])
```

```
## train the model using the pipeline
pipeline.fit(X_train, y_train)

## predictions
y_pred = pipeline.predict(X_test)
```

REFERENCES

Ivey, A. (2023, April 20). *A brief history of digital banking*. Cointelegraph. https://cointelegraph.com/news/a-brief-history-of-digital-

banking#:~:text=Online%20banking%20portals%20were%20developed,people%20due%2 0to%20its%20convenience.

International, F. (2023, July 17). *5 reasons behind the increase in digital banking fraud*. Fraud.com. https://www.fraud.com/post/increase-in-digital-banking-fraud#:~:text=Online%20banking%20allows%20customers%20to,that%20can%20steal%2 0login%20credentials.

Ethical considerations in Data Collection - njhealthmatters.org. (n.d.). https://www.njhealthmatters.org/content/sites/njhc/Ethical_Considerations_in_Data_Collec tion.pdf

Machine learning for fraud detection. Ravelin. (n.d.). https://www.ravelin.com/insights/machine- learning-for-fraud-detection

A-to-Z core banking platform for both online and traditional banks. (2023, August 17). SEPA Cyber Technologies. https://sepa-cyber.com/core-banking-system/

What Code Do Banking Systems Run On? (2023, May 27). Medium. https://seattlewebsitedesign.medium.com/what-code-do-banking-systems-run-on-b62f1a50b574