TAKE HOME TEST

FRAUD MODEL

Gabriel Roger do Nascimento

Data Scientist SumUp

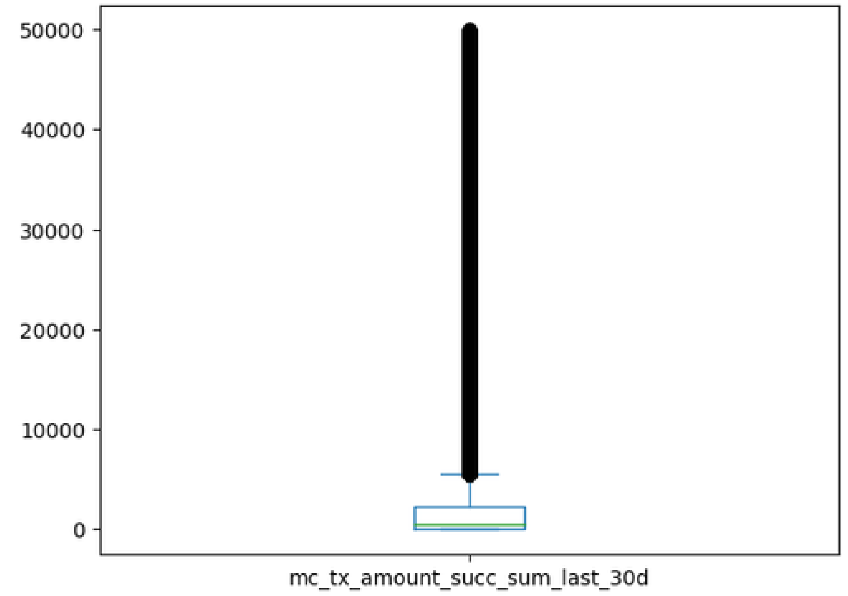
GOAL

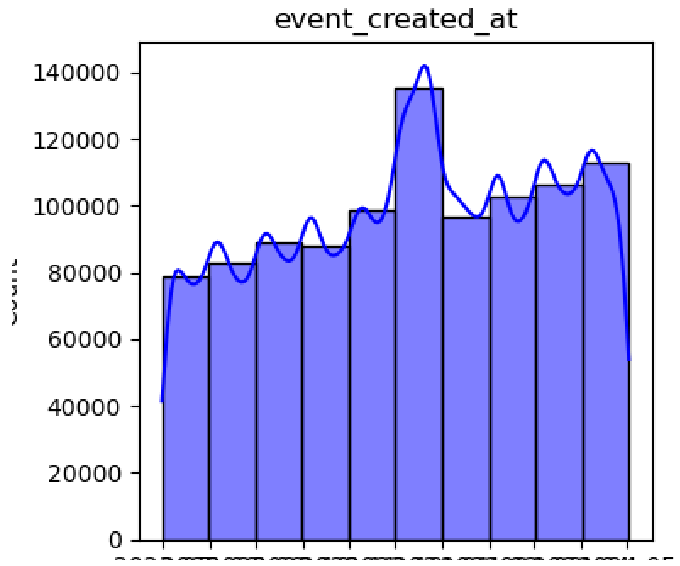
Develop a fraud detection model to identify suspicious transactions and protect our customers.

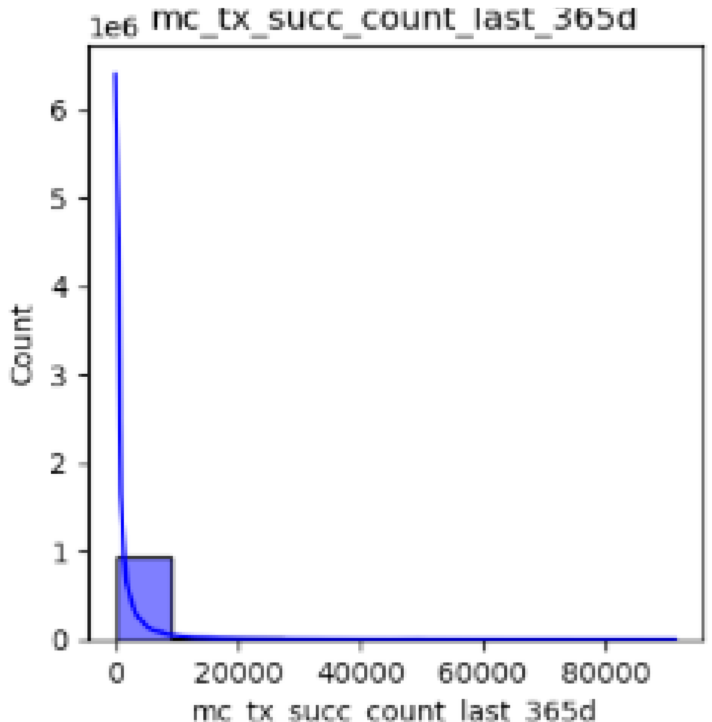
DATA

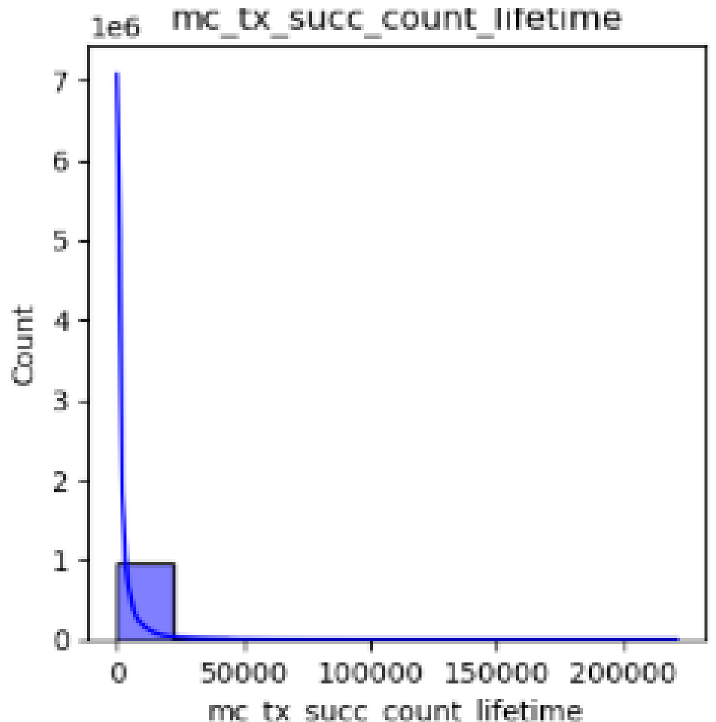
Period: 01/07/2023 to 01/05/2024

- **991.965** transactions after cleaned
- **808.513** Person's transactions
- **171.797** Mei's transactions
- **386** Legal entity's transactions
- **11269** Without Definition









PROBLEMS

DUPLICATED & NULL DATA

UNBALANCED

HIGH FEATURE DIMENSIONALITY

99%

Not Fraud

986956

0.005%

Fraud

5009

CLEANING & SELECTION

CLEAN DUPLICATES & NULL DATA

- **Drop Duplicates:** Remove any duplicate rows from the dataset.
- **Fill with -1:** Replace missing values (NaN, None, etc.) in the remaining features with the value -1.

BALANCING A DATASET

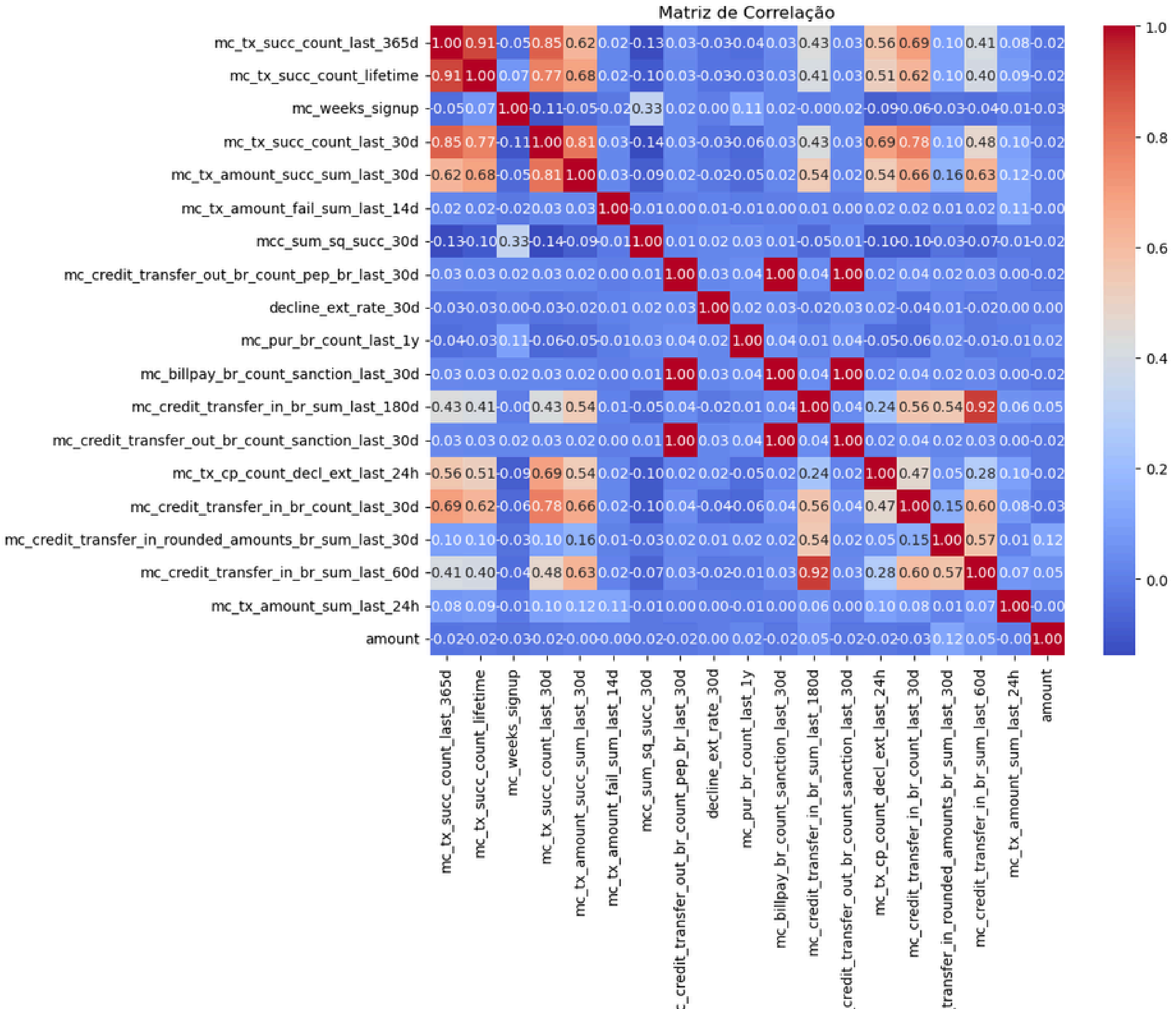
- **Smote:** Oversampling minority class with synthetic data (Just Train Dataset).

FEATURE SELECTION

- Random Forest to determine the importance of each feature.
- **SelectFromModel** with mean threshold to select features based on their **Random Forest** importance scores. This allows for an assessment of feature importance relative to one another.

SELECTED FEATURES

mc_tx_succ_count_last_365d - 11.33%
mc_tx_succ_count_lifetime - 10.68%
mc_weeks_signup - 6.10%
mc_tx_succ_count_last_30d - 2.35%
mc_tx_amount_succ_sum_last_30d - 0.54%
mc_tx_amount_fail_sum_last_14d - 0.53%
mcc_sum_sq_succ_30d - 0.50%
mc_credit_transfer_out_br_count_pep_br_last_30d - 0.41%
decline_ext_rate_30d - 0.39%
mc_pur_br_count_last_1y - 0.33%
mc_billpay_br_count_sanction_last_30d - 0.33
mc_credit_transfer_in_br_sum_last_180d - 0.29%
mc_credit_transfer_out_br_count_sanction_last_30d - 0.23%
mc_tx_cp_count_decl_ext_last_24h - 0.22%
mc_credit_transfer_in_br_count_last_30d - 0.19%
mc_credit_transfer_in_rounded_amounts_br_sum_l.. - 0.16%
mc_credit_transfer_in_br_sum_last_60d - 0.09%
mc_tx_amount_sum_last_24h - 0.07%



MODEL

- **Data Splitting:** Divide the dataset into training (70%), validation (15%), and test (15%) sets.
- **Data Normalization:** Apply StandardScaler to normalize the features.
- **Grid Search:** to compare the performance of Random Forest and XGBoost (with tuned hyperparameters).
- Evaluate models using the **ROC AUC** metric.
- Select **XGBoost** as the final model based on the best ROC AUC score.

METRICS

Metrics with Threshold of 0.7:

AUC Train: 0.9924325269652872

AUC Val: 0.9645852112619379

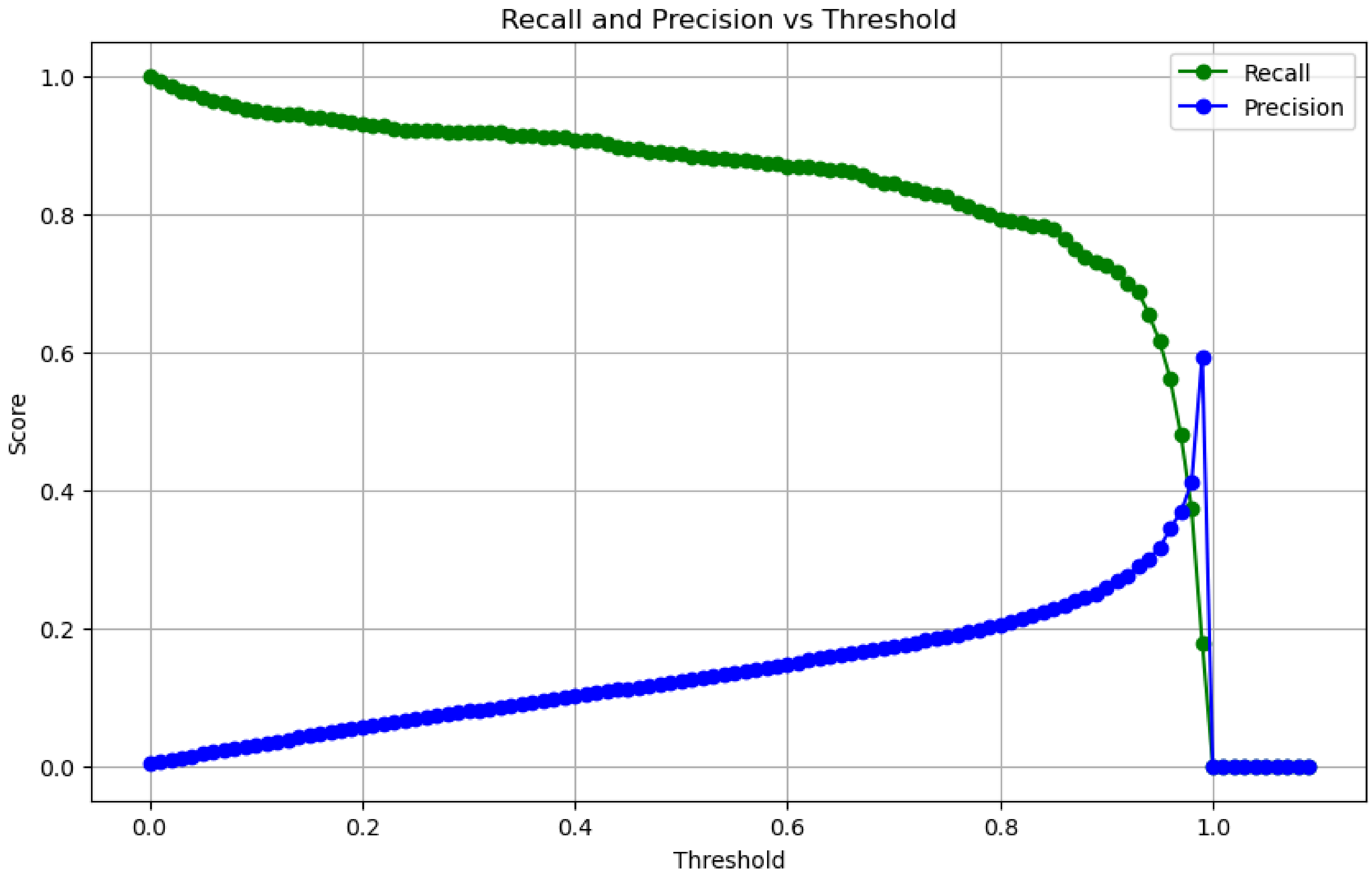
AUC Test: 0.9697022227997787

Precision: 0.1736510545056149

Recall: 0.844207723035952

F1-Score: 0.2880508859609269

METRICS



Recall and Precision vs Threshold
Recall
Precision
Score
Threshold
0.0
0.2
0.4
0.6
0.8
1.0

CONCLUSION

- **Recall Prioritization**: In banking, it's more critical to detect all potential fraud cases (even if it means more false positives) than to miss actual fraud cases.
- **False Positive Impact:** False positives (classifying a legitimate transaction as fraudulent) inconvenience customers (e.g., blocked cards, extra verification
  - .Customers prefer to prevent fraud, even if it means occasional inconveniences from false positives.
- **False Negative Impact:** False negatives (missing actual fraud) result in financial loss and damage to customer trust.)

API FLASK EXAMPLE:

```
import requests
import json

data = {
    "mc_tx_succ_count_last_365d": 202.0,
    "mc_tx_succ_count_lifetime": 0.0,
    "mc_weeks_signup": 189.7143,
    "mc_tx_succ_count_last_30d": 0.0,
    "mc_tx_amount_succ_sum_last_30d": 0.0,
    "mc_tx_amount_fail_sum_last_14d": 0.0,
    "mcc_sum_sq_succ_30d": 9.109,
    "mc_credit_transfer_out_br_count_pep_br_last_30d": 0.0,
    "decline_ext_rate_30d": 1.0,
    "mc_pur_br_count_last_1y": 20,
    "mc_billpay_br_count_sanction_last_30d": 12.2,
    "mc_credit_transfer_in_br_sum_last_180d": 2.4311,
    "mc_credit_transfer_out_br_count_sanction_last_30d": 15.0,
    "mc_tx_cp_count_decl_ext_last_24h": 0,
    "mc_credit_transfer_in_br_count_last_30d": 3.0,
    "mc_credit_transfer_in_rounded_amounts_br_sum_last_30d": 0.0,
    "mc_credit_transfer_in_br_sum_last_60d": 10,
    "mc_tx_amount_sum_last_24h": 20
}


url = 'http://localhost:5001/predict'
response = requests.post(url, json=data)

if response.status_code == 200:
    print(response.json())
else:
    print(response.status_code)
    print(response.text)
```

```
{'classe': 'Fraude', 'prob': 0.7159508466720581}
```

NEXT STEPS

- Adjustments to the threshold or model parameters may be necessary to refine the results.
- Benchmark more algorithms with different hyperparameters.
- Explore more possible features and better ways to balance and select the features.
- If possible, collect more data.
- Improve EDA analysis.
- Data related to the devices used for transactions could be a valuable area for future exploration. (Device Fingerprint)
- Explore seasonality on date transactions
    - Time series model based

THANKS