Ashton Josh Kabou

INST414

Sprint 1: Data Science Lifecylce Option

Project: Predicting Synthetic Fraud in Financial Transaction

Problem statement:

In today's fastest economy and technology evolution, financial transactions reach significant numbers each year. Millions of Americans use debit and credit cards for their day to day expenses, such as groceries and utilities. This increase in transactions led to various types of fraud, such as account takeovers, credit card frauds, true identity theft, ACH fraud and synthetic fraud. For the scope of our project, we will focus on one of the fastest growing frauds: Synthetic fraud. We will navigate the impacts of synthetic fraud in financial institutions.

According to the Federal Reserve Bank, financial synthetic fraud is when someone uses a combination of real and false identification information to create a new account or to make transactions. The Department of Justice claims that synthetic fraud is one of the fastest growing frauds in the US. According to Transunion, synthetic fraud happens through 5 main mechanisms. The first one being credit repair agencies, who use credit profile number, which is a nine digit number that serves as SSN. These companies claim to provide a fresh start to their client, but often CPNs are false or created identities belonging to oppressed individuals, such as immigrants. The second major cause of synthetic fraud is loosening credit. As the economy grew and to reduce discrimination, US financial institutions loosen credit requirements facilitating synthetic fraud.

The third major cause of synthetic fraud is SSN randomization. Before 2011, SSN 4 first digits were associated with the SSN agency that issued it, but after 2011 the US started randomizing all digits of SSN making it hard to recognize a false SSN. The fourth major cause of synthetic fraud is data breaches. Transunion claims billions of record breaches happened over the last decade providing a significant amount of personal information to fraudsters. In August of 2024, a data breach occurred in the US and billions of personal identifiable information, such as social security numbers, names, and addresses were disclosed. "This breach allegedly exposed up to 2.9 billion records with highly sensitive personal data of up to 170M people in the US, UK, and Canada" (*National Public Data Breach: What You Need to Know*, 2024). This breach exposed enough personnel information to fraudsters, who might use them to commit synthetic financial fraud. With the recent data breach that occurred in 2024 we can expect that synthetic fraud in financial institutions will be higher in 2025.

Synthetic fraud is very dangerous because the individual whose information is being used, is not aware and might not report unauthorized transactions. Fraudsters profit from this situation by exploiting financial institutions by building a credible profile before committing important transactions then after closing or abandoning the account. This problem is worth tackling especially in developed countries, due to advanced technology. For example, one can have his credit cards, ID’s , and even credit information on his phone. In case the phone is lost or hacked, this information can be compromised. Despite the efforts of financial institutions, it is difficult to identify false information and frauds.

Review of similar solutions:

According to RCB BANK, financial institutions and businesses combat synthetic fraud by deploying machine learning algorithms and AI to analyze customers behavior and transactions. In addition, they use cross data and biometrics to uncover potential anomalies. Forbes claims that many banks use NLP to process documentation and detect fraud, but this operation must comply with privacy regulations of the region in which the bank resides.

Scope of Work:

1) Problem:
    - Identify fraud in financial institutions, especially synthetic fraud
    - Distinguish real users from fraudsters
2) Dataset components:
    - Transaction data : amount of transactions, timestamps, merchant details, transaction type
    - Card information: credit and debit card information, cards limitless, types, activation date
    - Merchant category codes: industry standard MCC, standard classification codes for business types
    - Fraud labels: Binary classification labels for transactions
    - User data: Demographic information on customers, account details

Methodology:

For my predictive analysis I will use a financial transaction dataset suited for analytics from Kaggle. Starting out with the dataset, I will inspect the dataset and handle missing values, duplicate records and data inconsistencies using Python with its library Pandas. While inspecting the dataset, I will identify fraud indicators and synthetic frauds already confirmed by the dataset. Moving forward with my predictive analysis, I am not yet set on all models I will implement, but plan on using regression, neural networks, and decision trees.

Data source, success metrics and evaluation criteria

As stated previously, I will be using a financial transaction dataset from Kaggle with a usability score of 10.00. This dataset is big enough and contains sufficient information for my analysis with components such as transaction data, card information, merchant category codes, fraud labels, and user data. Some files are CSV and others JSON. In order to have a reliable model, my success metrics will consist of : accuracy, precision, sensitivity, F1 score, AUC-ROC, false positive rate and false negative rate. To ensure my model performs well, I will evaluate how well it predicts unseen data, and differentiate fraud vs real transactions. Also, I will compare the cost of false positives vs false negatives.

Technical Skills Development Plan:

I plan on mastering decision trees and regression through this course. Since Deep learning is not taught in this course I will self learn neural networks.

Works Cited

Biggest Data Breaches of 2024. (2024, December 27). Proven Data. Retrieved February 22, 2025, from https://www.provendata.com/blog/2024-biggest-data-breaches/

Everything Financial institutions need to know about Synthetic Fraud. https://www.thomsonreuters.com/en-us/posts/investigation-fraud-and-risk/synthetic-identity-fraud/

Getting Value from NLP for Fraud Detection. (2023,May 10) (https://www.forbes.com/councils/forbesbusinesscouncil/2023/05/10/getting-value-from-nlp-for-fraud-detection/

Identity Thief sentenced for using a new form of fraud “Synthetic Identities”. (2017, April 28). Department of Justice. Retrieved February 23, 2025, from https://www.justice.gov/usao-ndga/pr/identity-thief-sentenced-using-new-form-fraud-synthetic-identities

National Public Data breach: What you need to know. (2024). Microsoft Support. Retrieved February 22, 2025, from *https://support.microsoft.com/en-us/topic/national-public-data-breach-what-you-need-to-know-843686f7-06e2-4e91-8a3f-ae30b7213535*

What's behind the rise in synthetic identity fraud. (2023, November 29) https://www.transunion.com/blog/money-2020-whats-behind-rise-synthetic-identity-fraud?atvy=%7B%22264995%22%3A%22Experience+B%22%7D

Synthetic Identity Fraud Defined. FedPayments Improvement. Retrieved February 22, 2025, from https://fedpaymentsimprovement.org/strategic-initiatives/payments-security/synthetic-identity-payments-fraud/synthetic-identity-fraud-defined

Synthetic Identity Fraud: The Fastest-Growing Financial Crime of 2025. RCB Bank. Retrieved February 23, 2025, from https://rcbbank.bank/learn-synthetic-identity-fraud-the-fastest-growing-financial-crime-of-2025/

Sprint 2:

Gather datasets identified in Sprint 1

I used the Financial Transaction Dataset:Analytics from Kaggle which contains 5 files(transactions_data, users_data, card_data, train_fraud_labels, mcc_codes). I used all 5 files and did not change data availability or scope.

Dictionary:

variables_description = {

"id_x": "Unique identifier for a transaction in the primary dataset.",

"date": "Timestamp indicating when the transaction occurred.",

"card_id": "Unique identifier for the card used in the transaction.",

"amount": "Transaction amount in monetary units.",

"transaction_id": "Identifier for the transaction as recorded in another dataset.",

"Fraud label": "Binary label indicating whether the transaction is fraudulent (1) or not (0).",

"errors_encoded": "Encoded representation of transaction error types.",

"mcc_encoded": "Encoded representation of the Merchant Category Code (MCC).",

"zip_encoded": "Encoded representation of the ZIP code associated with the transaction.", "merchant_state_encoded": "Encoded representation of the state where the merchant is located.",

"merchant_city_encoded": "Encoded representation of the city where the merchant is located.",

"merchant_id_encoded": "Encoded representation of the merchant's unique identifier.",

"use_chip_encoded": "Encoded flag indicating whether the transaction used a chip (1) or not (0).",

"year": "Year of the transaction derived from the date.",

"month": "Month of the transaction derived from the date.",

"day": "Day of the transaction derived from the date.",

"hour": "Hour of the transaction derived from the date.",

"minute": "Minute of the transaction derived from the date.",

"second": "Second of the transaction derived from the date.",

"id_y": "Secondary identifier for a transaction, possibly linked to other datasets.",

"current_age": "Current age of the client associated with the transaction.",

"retirement_age": "Expected retirement age of the client.",

"latitude": "Latitude coordinate of the transaction location.",

"longitude": "Longitude coordinate of the transaction location.",

"per_capita_income": "Per capita income for the client's location or region.",

"yearly_income": "Yearly income of the client associated with the transaction.",

"total_debt": "Total debt held by the client.",

"credit_score": "Credit score of the client.",

"num_credit_cards": "Number of credit cards the client owns.",

"gender_encoded": "Encoded representation of the client's gender.",

"id": "Unique identifier for a specific dataset related to the client or transaction.",

"client_id_y": "Alternative identifier for the client, used for merging datasets.",

"num_cards_issued": "Total number of cards issued to the client.",

"credit_limit": "Credit limit of the card used in the transaction.", "year_pin_last_changed": "Year when the PIN for the card was last changed.",

"card_number_encoded": "Encoded representation of the card number.",

"card_brand_encoded": "Encoded representation of the card's brand (e.g., Visa, MasterCard).",

"card_type_encoded": "Encoded representation of the card type (e.g., credit, debit).",

"cvv_encoded": "Encoded representation of the card's CVV (Card Verification Value).",

"has_chip_encoded": "Encoded flag indicating whether the card has a chip (1) or not (0).",

"card_on_dark_web_encoded": "Encoded flag indicating if the card number was found on the dark web.",

"expires_year": "Year when the card is set to expire.",

"expires_month": "Month when the card is set to expire.",

"acct_open_year": "Year the account associated with the card was opened.",

"acct_open_month": "Month the account associated with the card was opened."

}

Data cleaning and preprocessing

1) Handling missing values:

After merging transactions_data and train_fraud_label datasets, some transactions did not have fraud labels. Missing values in the fraud label column were addressed by separating labeled and unlabeled transactions. Rows without labels were removed from the labeled dataset to ensure that only transactions with complete information were used for training.

In the transaction dataset, some zip codes and merchant_city were missing, i could not drop the rows because it would imply dropping all online transactions since only online transactions were missing zip and merchant_city. I could not impute with mean or mode since it is not appropriate for my data. I encoded(label encoding) all missing values in zip and merchant_city.

2) **Handling Outliers:**

   I converted monetary values (amount, yearly_income, per_capita_income, total_debt, credit_limit) to numeric and removed dollar signs using regex. Doing so ensured all continuous variables were numeric for proper calculations and transformations, allowing detection and handling of extreme values in further steps.

3) **Handling Inconsistencies:**

   I encoded columns: errors,mcc,zip,merchant_state, merchant_city, use_chip into numeric representations. Label encoding made categorical variables compatible with ML algorithms.

   I dropped original columns after encoding to remove redundancy and feature duplication.

Data visualization

1) **Class Imbalance plot:** This plot visualizes the distribution of fraud (`1`) and non-fraud (`0`) transactions. It helped see imbalance between fraud and non fraud cases in the dataset, which I handled with SMOTE ad class weighting.
2) **Distribution of Transaction amounts plot:** This histogram shows the frequency of transaction amounts, alongside a density estimation curve. Analyzing transaction amounts helps identify trends or patterns in spending behavior for both fraudulent and non-fraudulent transactions.
3) **Feature Correlation Heatmap**: This heatmap shows the pairwise correlations between numerical features. The heatmap helped identify relationships between features, which helped feature selection and engineering.
4) **Transaction Frequency by Hour:**

This plot analyzes the frequency of transactions across different hours of the day, split by fraud and non-fraud labels. Patterns of fraud may vary by time of day.

Feature engineering and selection

1) **Creation of new features:**

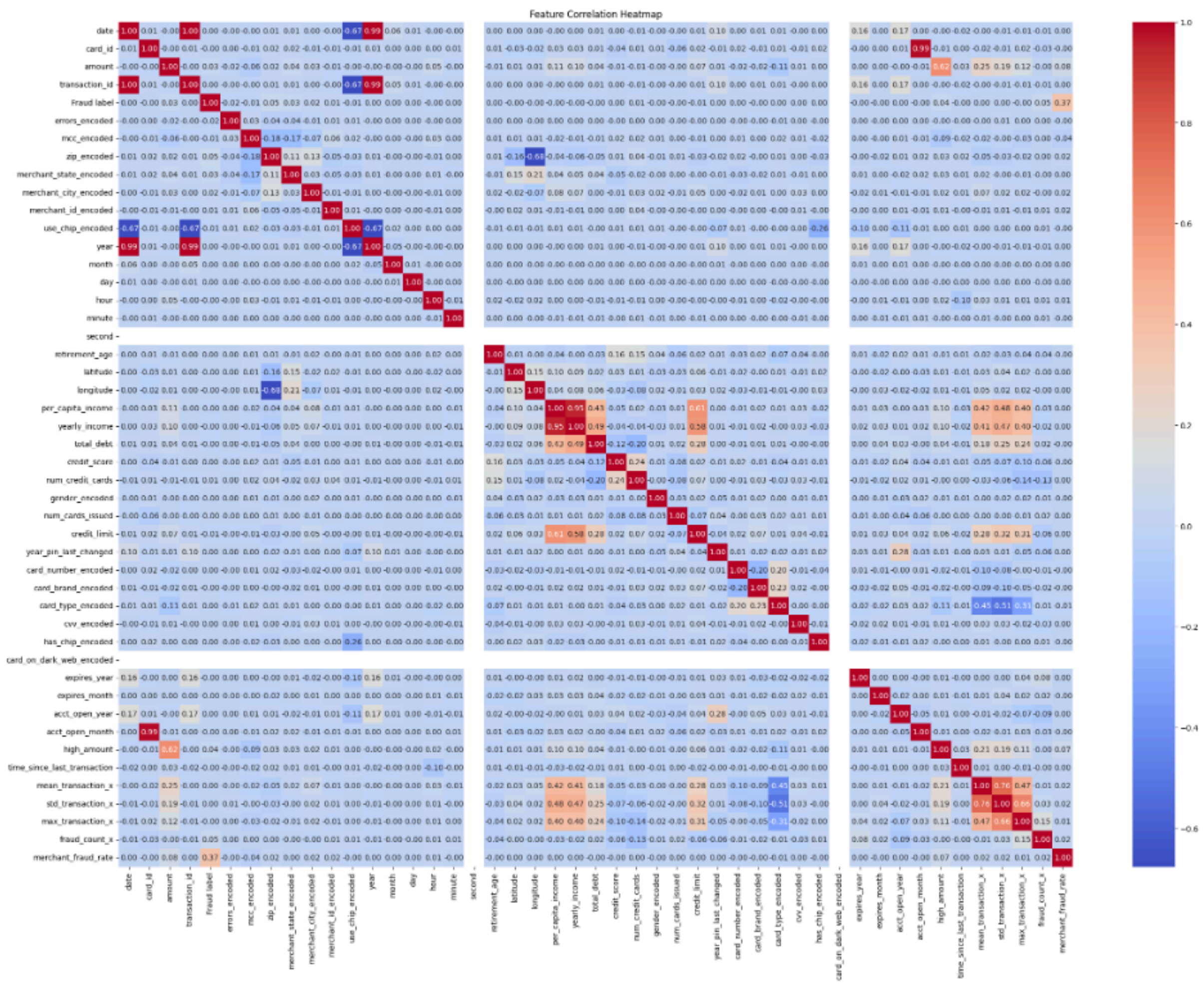
   a) **Flagging high-value transactions:** A new binary feature high amounts is created by identifying transactions that fall in the top 5% by amount. This could help the model identify potentially high-risk transactions, as larger transactions are often more scrutinized in fraud detection.
   b) **Time since last transaction**: A new feature, time_since_last_transaction, is calculated by sorting the data by transaction date and then finding the time difference between each transaction and the previous one. This feature could be useful in fraud detection, as fraudulent transactions may often occur within short intervals of legitimate ones.
   c) **Card activity statistics:** Transaction statistics (mean, standard deviation, and maximum) and the count of fraud occurrences are aggregated for each card_id and added as new features: mean_transaction, std_transaction, max_transaction, and fraud_count. This aggregation captures the card's general usage pattern, which can help identify suspicious behavior.
   d) **Merchant fraud likelihood:** The fraud rate is computed for each merchant (merchant_id_encoded), which represents the proportion of fraudulent transactions for that merchant. This feature helps capture patterns related to merchants that have a higher likelihood of being associated with fraudulent activity.

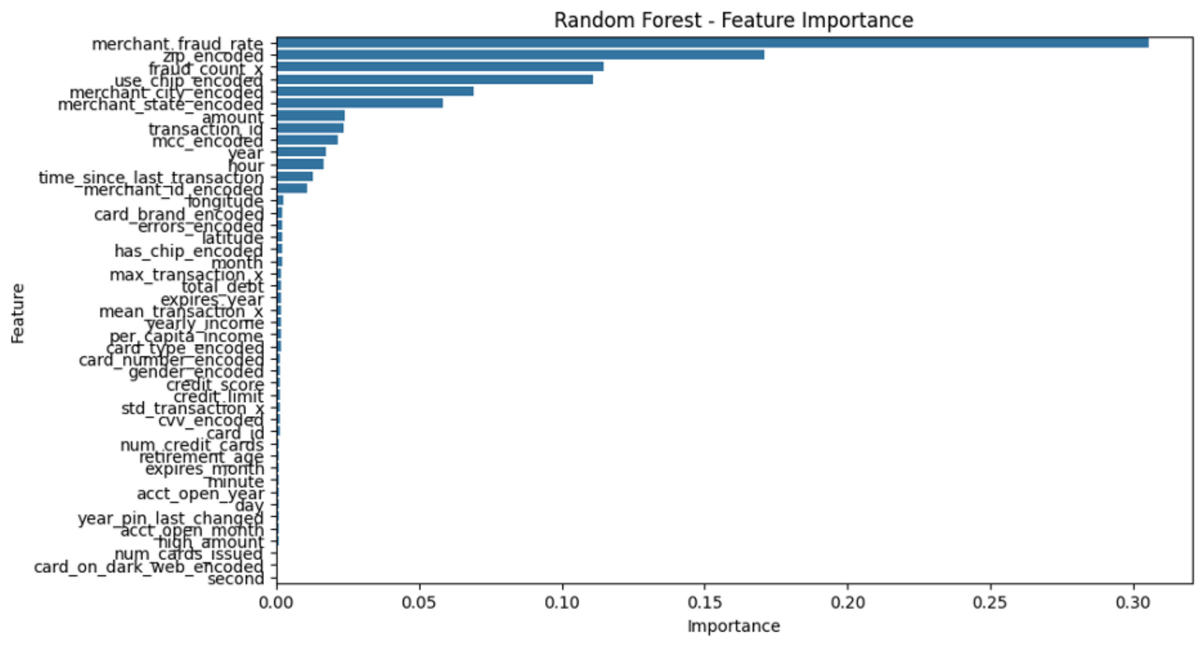
2) **Most relevant features:**

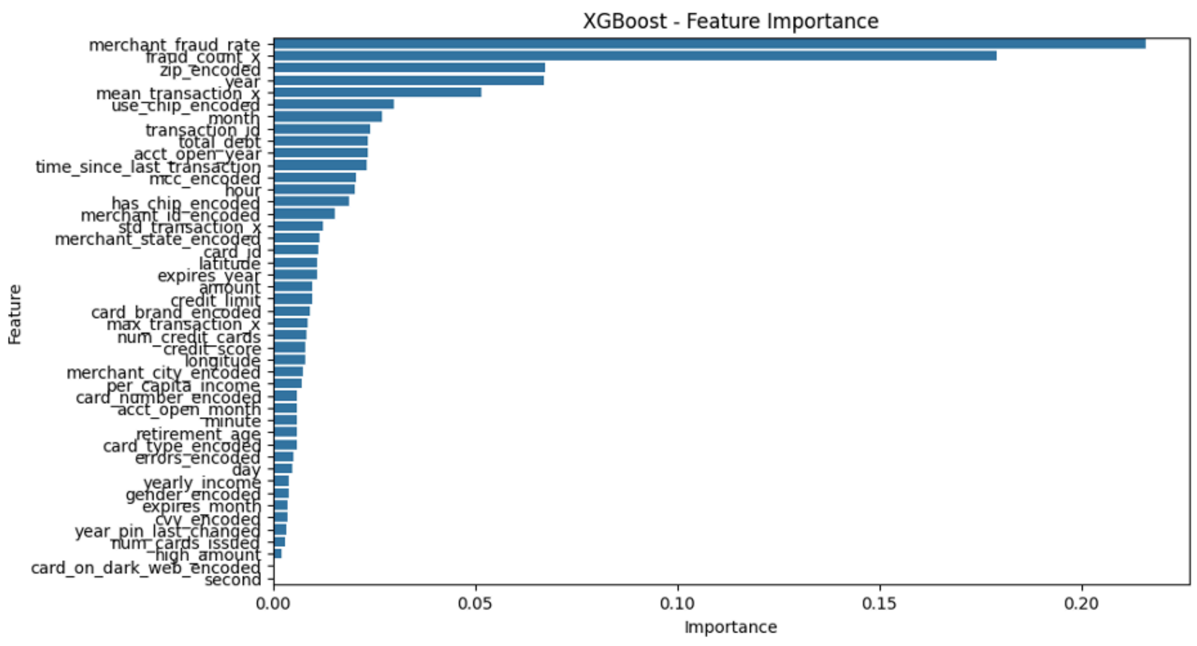
I found the most important features by using the heatmap and by training my models with all features and selecting the most important.

"merchant_fraud_rate", "fraud_count_x", "zip_encoded", "use_chip_encoded","mean_transaction_x", "transaction_id", "mcc_encoded", "merchant_city_encoded", "merchant_state_encoded", "time_since_last_transaction"

Graphs:


Feature Correlation Heatmap
date
card_id
amount
transaction_id
Fraud label
errors_encoded
mcc_encoded
zip_encoded
merchant_state_encoded
merchant_city_encoded
merchant_id_encoded
use_chip_encoded
year
month
day
hour
minute
second
retirement_age
latitude
longitude
per_capita_income
yearly_income
total_debt
credit_score
num_credit_cards
gender_encoded
num_cards_issued
credit_limit
year_pin_last_changed
card_number_encoded
card_brand_encoded
card_type_encoded
cvv_encoded
has_chip_encoded
card_on_dark_web_encoded
expires_year
expires_month
acct_open_year
acct_open_month
high_amount
time_since_last_transaction
mean_transaction_x
std_transaction_x
max_transaction_x
fraud_count_x
merchant_fraud_rate
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6


Random Forest - Feature Importance
Feature
Importance
merchant_fraud_rate
zip_encoded
fraud_count_x
use_chip_encoded
merchant_city_encoded
merchant_state_encoded
amount
transaction_id
mcc_encoded
year
hour
time_since_last_transaction
merchant_id_encoded
longitude
card_brand_encoded
errors_encoded
latitude
has_chip_encoded
month
max_transaction_x
total_debt
expires_year
mean_transaction_x
yearly_income
per_capita_income
card_type_encoded
card_number_encoded
gender_encoded
credit_score
credit_limit
std_transaction_x
cvv_encoded
card_id
num_credit_cards
retirement_age
expires_month
minute
acct_open_year
day
year_pin_last_changed
acct_open_month
high_amount
num_cards_issued
card_on_dark_web_encoded
second
0.00
0.05
0.10
0.15
0.20
0.25
0.30



3) Rational:

1. **merchant_fraud_rate**
   - **Importance**: If a merchant has a history of fraudulent transactions, it increases the likelihood that a new transaction from that merchant is also fraudulent.
   - **Impact on model**: A high fraud rate suggests that the merchant may be involved in suspicious activity.
2. **fraud_count_x**
   - **Importance**: Tracks the number of times fraud has been reported for a particular entity (e.g., a customer, merchant, or card).
   - **Impact on model**: Entities with a high fraud count are more likely to be involved in fraudulent activity again.
3. **zip_encoded**
   - **Importance**: Certain ZIP codes may have a higher prevalence of fraudulent transactions, especially in high-risk areas.
   - **Impact on model**: Helps capture regional fraud trends by encoding location-based risk.
4. **use_chip_encoded**
   - **Importance**: Chip-based transactions (EMV) are generally more secure than magnetic stripe transactions.
   - **Impact on model**: Transactions that **do not** use the chip may have a higher fraud risk, especially for in-person transactions.

5. **mean_transaction_x**
    - **Importance**: Represents the average transaction amount for a given entity. Fraudsters may have patterns of making unusually high or low transactions.
    - **Impact on model**: Transactions that deviate significantly from a user's average spending pattern can indicate fraud.
6. **transaction_id**
    - **Importance**: Can be used as an identifier to track recurring fraud attempts from the same source.
    - **Impact on model**: If certain transaction IDs exhibit a pattern linked to fraud, they become strong indicators.
7. **mcc_encoded (Merchant Category Code)**
    - **Importance**: Some categories of merchants (e.g., gambling, electronics, or high-value goods) have higher fraud rates.
    - **Impact on model**: Helps the model learn which business types are more likely to be associated with fraud.
8. **merchant_city_encoded**
    - **Importance**: Fraudulent activities might be more concentrated in certain cities.
    - **Impact on model**: Encoded city data helps the model identify geographic fraud patterns.
9. **merchant_state_encoded**
    - **Importance**: Like city-level encoding, some states might have more fraud cases due to regulatory loopholes or fraud hotspots.
    - **Impact on model**: Provides another geographical fraud signal.
10. **time_since_last_transaction**

- **Importance**: If a customer suddenly makes multiple transactions in a short time, it may indicate card theft or bot activity.
- **Impact on model**: Helps detect rapid-fire transactions, which are common in fraud schemes.

Model evaluation and comparison

Model Parameters:

- Random Forest

    - n_estimators=100
    - class_weight='balanced'
    - random_state=42

- XGBoost
    - n_estimators=100
    - learnign_rate=0.1
    - scale_pos_weight=len(y_train[y_train==0])/len(y_train[y_train==1])
    - eval_metric='logloss'
    - random_state=42

Evaluation Metrics Used:

- Accuracy: Measures overall correctness of the model.
- Precision: Fraction of correctly identified fraud cases among those predicted as fraud.
- Recall (Sensitivity): Fraction of actual fraud cases correctly identified.
- F1-Score: Harmonic mean of precision and recall, useful for imbalanced datasets.
- ROC-AUC (Receiver Operating Characteristic - Area Under Curve): Measures model discrimination ability.
- PR-AUC (Precision-Recall Area Under Curve): More relevant for imbalanced datasets, such as mine.

Model Performance Results:

Random Forest Test Results:

Accuracy: 0.9997

Class	Precision	Recall	F1-score	Support
0 (Non-Fraud)	1.00	1.00	1.00	1,780,327
1 (Fraud)	0.93	0.86	0.89	2,666

- **Macro Average:** Precision = 0.96, Recall = 0.93, F1-score = 0.95

- **Weighted Average:** Precision = 1.00, Recall = 1.00, F1-score = 1.00
- **Random Forest Mean Accuracy:**0.998
- **Random Forest ROC_AUC:**0.9853
- **Random Forest PR_AUC:**0.9172

XGBoost Test Results:

Accuracy: 0.9571

Class	Precision	Recall	F1-score	Support
0 (Non-Fraud)	1.00	0.96	0.98	1,780,327
1 (Fraud)	0.03	1.00	0.06	2,666

- **Macro Average:** Precision = 0.52, Recall = 0.98, F1-score = 0.52
- **Weighted Average:** Precision = 1.00, Recall = 0.96, F1-score = 0.98
- **XGBoost mean Accuracy:** 0.9784
- **XGBoost ROC-AUC**: 0.9984
- **XGBoost PR_AUC:**0.6554

Strength and weaknesses of models:

1) **Strength**
    a) **Random Forest:**
        - **High accuracy (99.97%)**: Effectively classifies both fraud and non-fraud cases.
        - **Balanced precision and recall**: A precision of 0.93 and recall of 0.86 for fraud cases means it detects fraud while keeping false positives relatively low.

- **Strong ROC-AUC (0.9853) and PR-AUC (0.9172)**: Indicates that the model is highly effective at distinguishing between fraud and non-fraud cases.

b) **XGBoost:**

- **Higher recall (1.00) for fraud cases**: Captures all fraudulent transactions, ensuring no fraud goes undetected.
- **Superior ROC-AUC (0.9984)**: Suggests the model is highly effective at ranking fraudulent transactions.

2) Weaknesses:

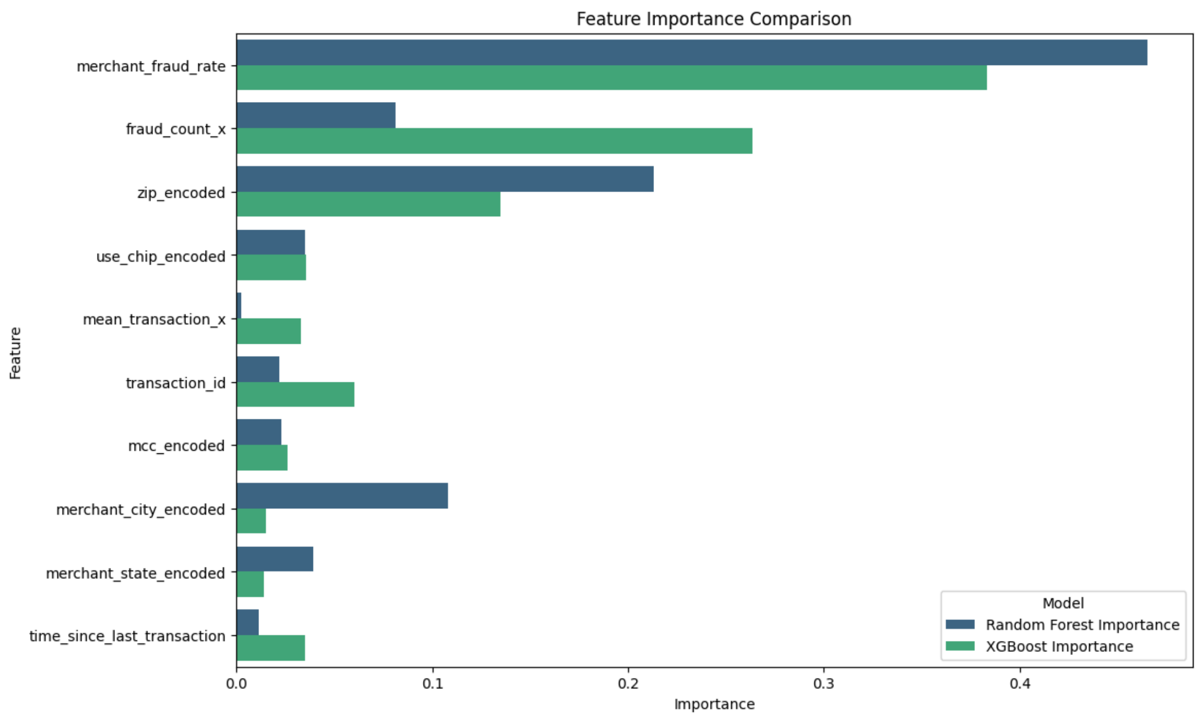
a) **Random Forest:**

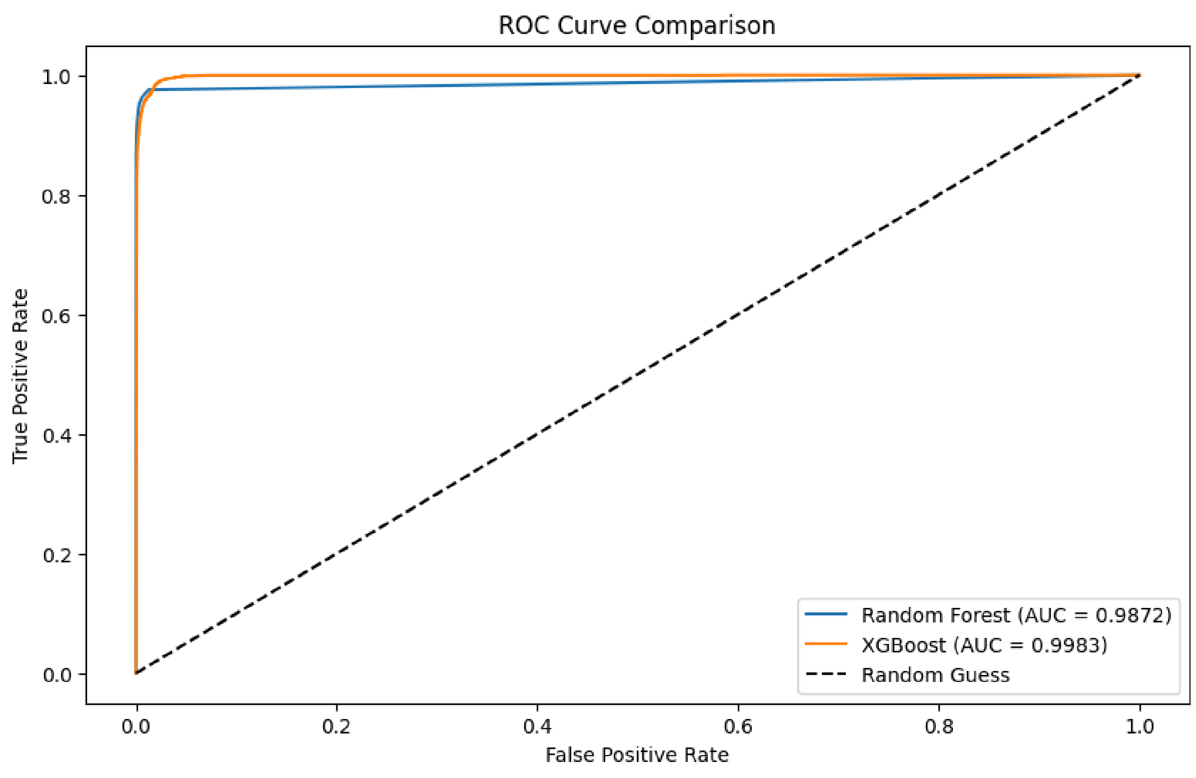
- **False negatives still exist**: While recall (0.86) is high, some fraudulent cases may go undetected.
- **Computationally expensive**: Training on large datasets may take longer compared to simpler models.

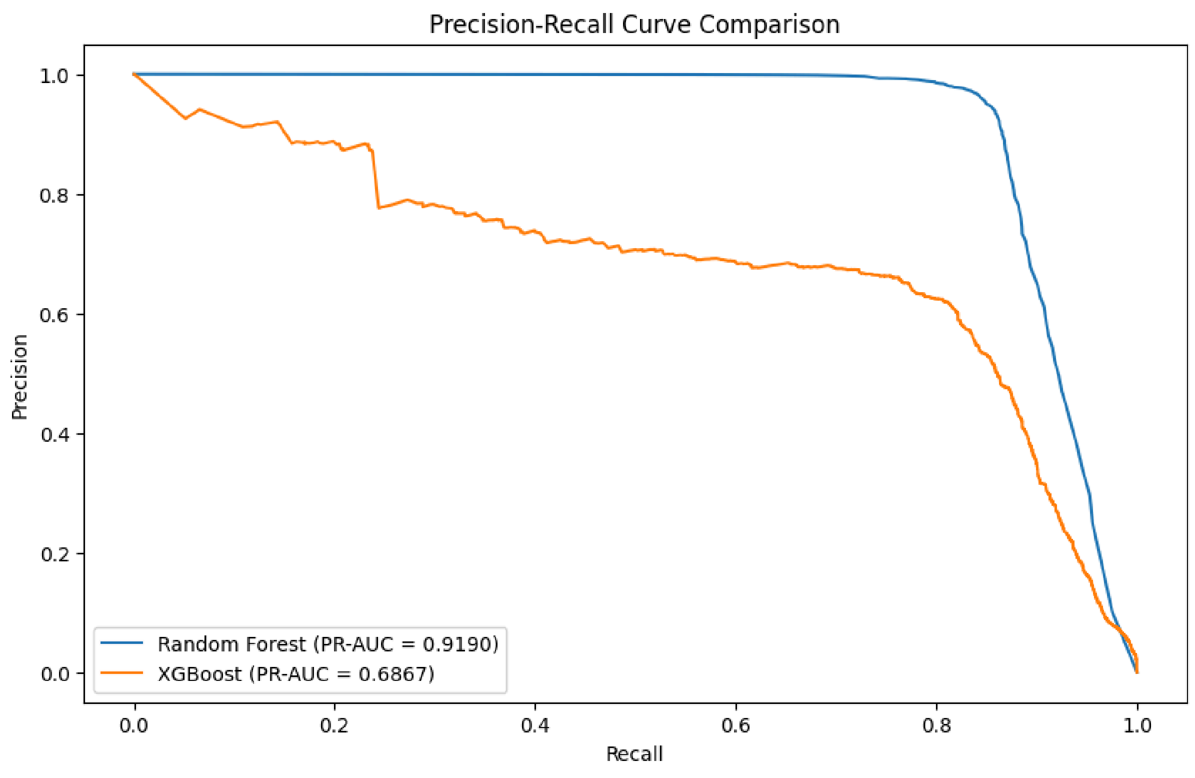
b) **XGBoost:**

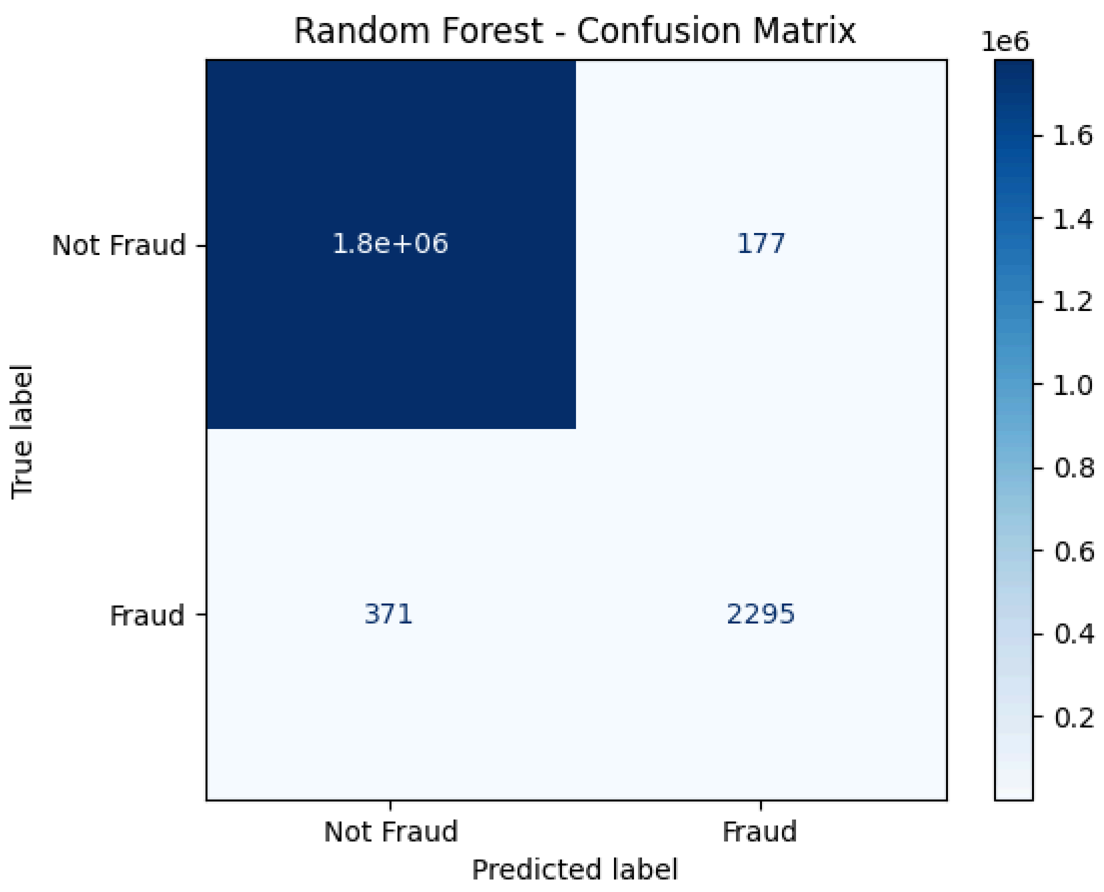
- **Extremely low precision (0.03) for fraud cases**: Predicts many false positives, which could lead to unnecessary fraud alerts.
- **Low PR-AUC (0.6554)**: Indicates that while the model detects fraud well, it struggles with precision.

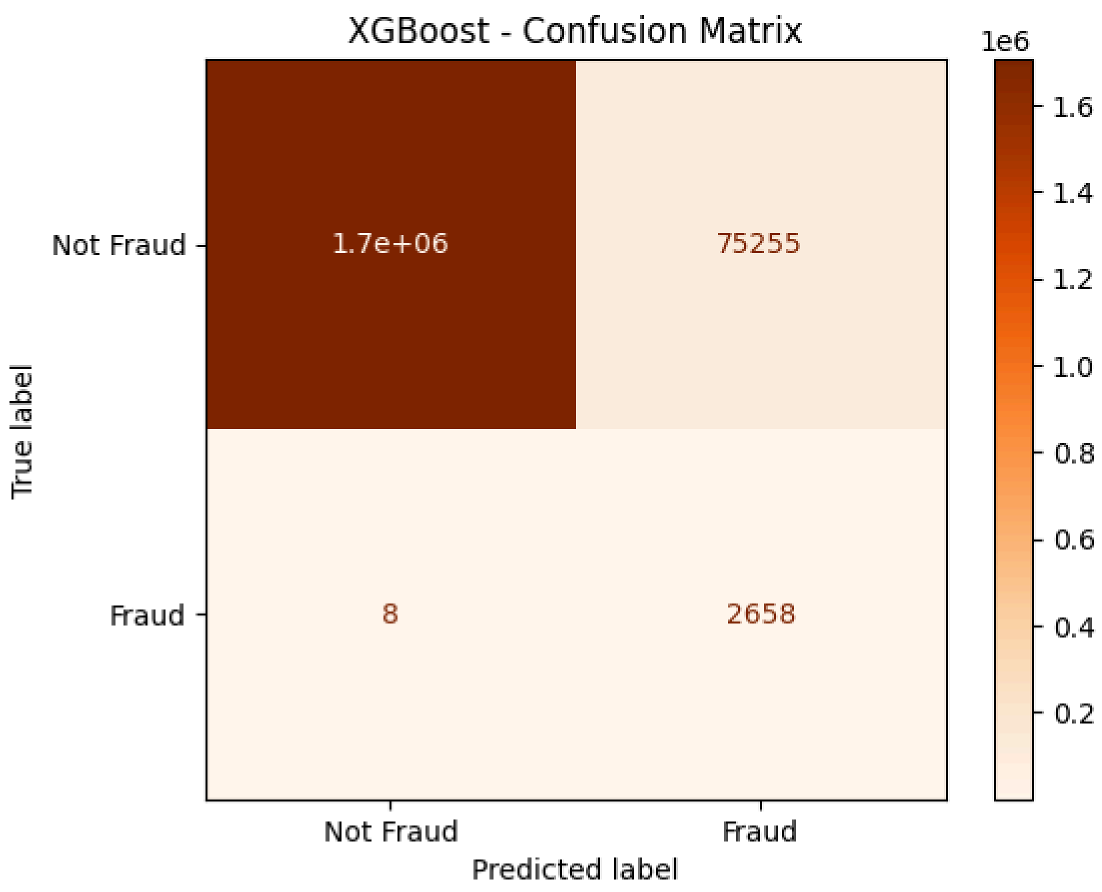
Comparaison graphs:




ROC Curve Comparison
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Random Forest (AUC = 0.9872)
XGBoost (AUC = 0.9983)
Random Guess


Precision-Recall Curve Comparison
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
Random Forest (PR-AUC = 0.9190)
XGBoost (PR-AUC = 0.6867)


Random Forest - Confusion Matrix
Not Fraud
Fraud
True label
1.8e+06
177
371
2295
Not Fraud
Fraud
Predicted label
1e6
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2


XGBoost - Confusion Matrix
True label
Predicted label
Not Fraud
Fraud
Not Fraud
Fraud
1.7e+06
75255
8
2658
1e6
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2