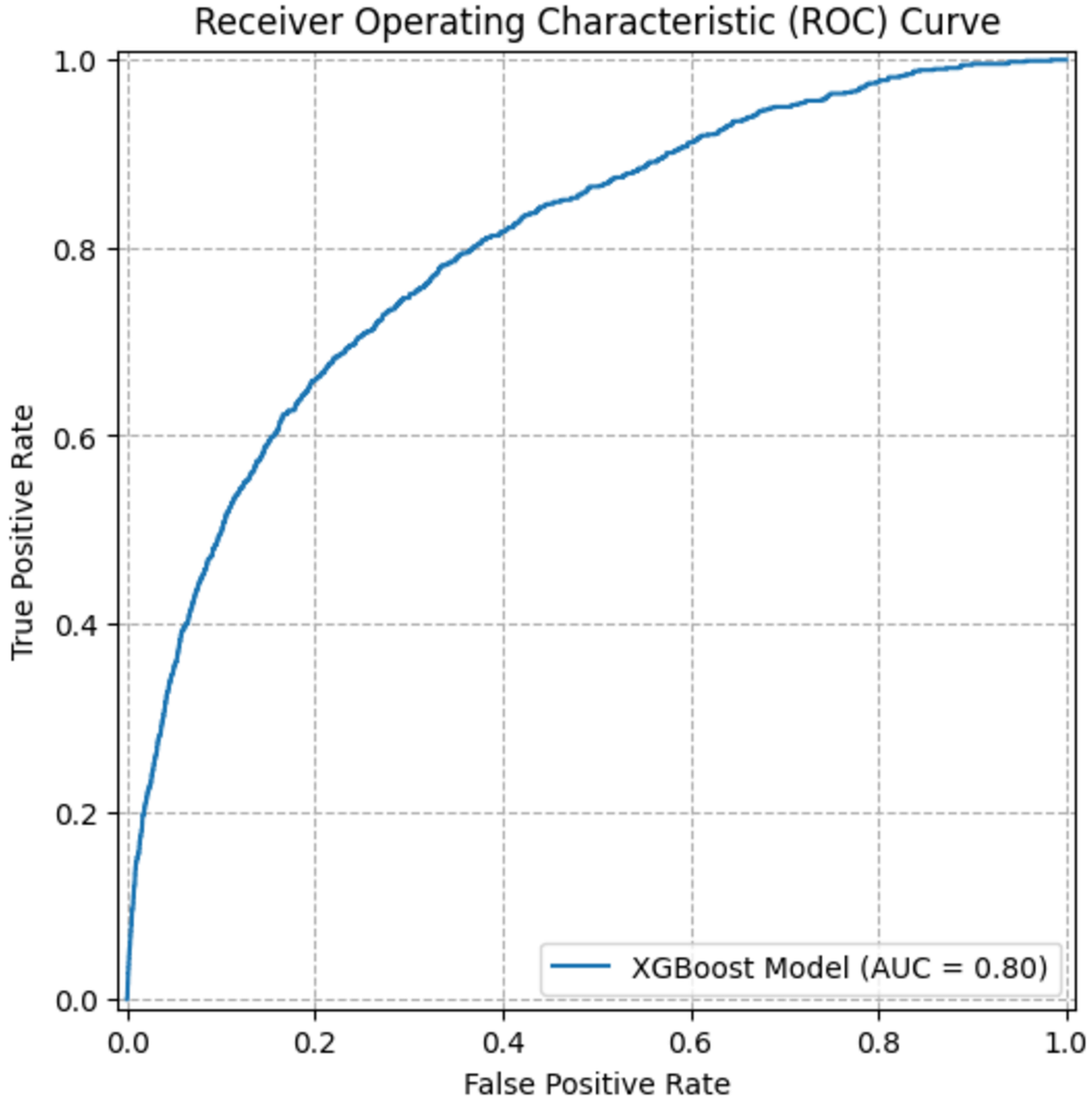
CBU

Default Prediction



Presented by: Scorpion

CBU

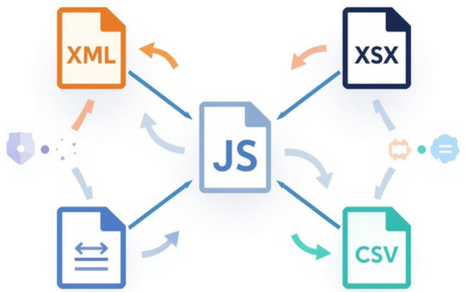
Task Overview

In this hackathon, Our main task is finding credit score of our customers whether they get default or not, using 6 provided files (in 5 different file formats) and then getting final result by merging all of them into results.csv file which should contain three columns: customer_id, prob and default.

CBU

Process

01



Member A

Demograhooics.csv/application_metadata.csv/loan_details.xlsx

02

Member B

financial_ration.jsonl/credit_history.parquet

03

Member C

georgaphic_data.xml

CBU

Data Cleaning

1. Fix data types
   - Convert currency/text numbers to float
   - Convert dates using pd.to_datetime()
2. Handle missing values
   - Impute numeric columns (median)
   - Impute categorical columns (most frequent)
3. Outlier handling
   - Capping extreme values (winsorization)
   - Removing impossible values (like negative income)

CBU

Model Selection & Evaluation

XGBoost was chosen because:

- Handles imbalanced data well (scale_pos_weight)
- Strong performance on structured financial data
- Captures non-linear patterns
- Provides feature importance
- More accurate than baseline models

For evaluation:

- Used ROC Curve & AUC
- Achieved AUC = 0.80, meaning good classification quality

Final results

Model Evaluation Metrics:
AUC-ROC: 0.8018
Accuracy: 0.9494
Precision: 0.5455
Recall: 0.0522
F1-Score: 0.0953

customer_id,prob,default
10000,0.031565856,0
10001,0.022534482,0
10002,0.07838209,0
10003,0.015420932,0
10004,0.029605506,0
10005,0.01120704,0
10006,0.03653438,0
10007,0.033919238,0
10008,0.024215585,0
10009,0.0064135683,0
10010,0.05958303,0
10011,0.0101528475,0
10012,0.0062175966,0
10013,0.013632494,0
10014,0.009375506,0
10015,0.13037239,0
10016,0.25038546,0
10017,0.009496485,0
10018,0.060634818,0
10019,0.0029714068,0
10020,0.00691052,0
10021,0.119932845,0
10022,0.09173962,0
10023,0.017920887,0