Report – Model Building

Angel Thu Do

November 2025

Executive Summary

This document describes the model built to predict the binary response variable "target" influenced by account quality, seasonal effect, and macroeconomics effects over time.

In order to choose the best model with the most accuracy, five different models are built and trained: Logistic Regression, Random Forest, Histogram-based Gradient Boosting (HistGradientBoosting), LightGBM, and XGBoost.

In this report, XGBoost is the best model, which achieves the highest validation AUC (0.7148).

Modeling Approach

1. Problem Type

- Binary classification: Predict the binary response variable target.

2. Data Handling

- Data Merge: 6 CSV files merged into a single dataset.
- Temporal Split:
    - Build set (Date_ym < 2022-05) → train (80%) / validation (20%)
    - Hold-out set (Date_ym >= 2022-05) → test set → final evaluation
- Segmented evaluation: Performance analyzed per segment S and over time (Date_ym).

3. Feature Engineering

- Predictor features: P1–P19 (account/macro variables)
- Temporal features: year, month, quarter, day_of_year
- Seasonal features: is_jan, is_feb, is_jan_feb
- Cyclical encoding: month_sin, month_cos to capture seasonal cycles

- Time trend: months_since_start to capture temporal evolution

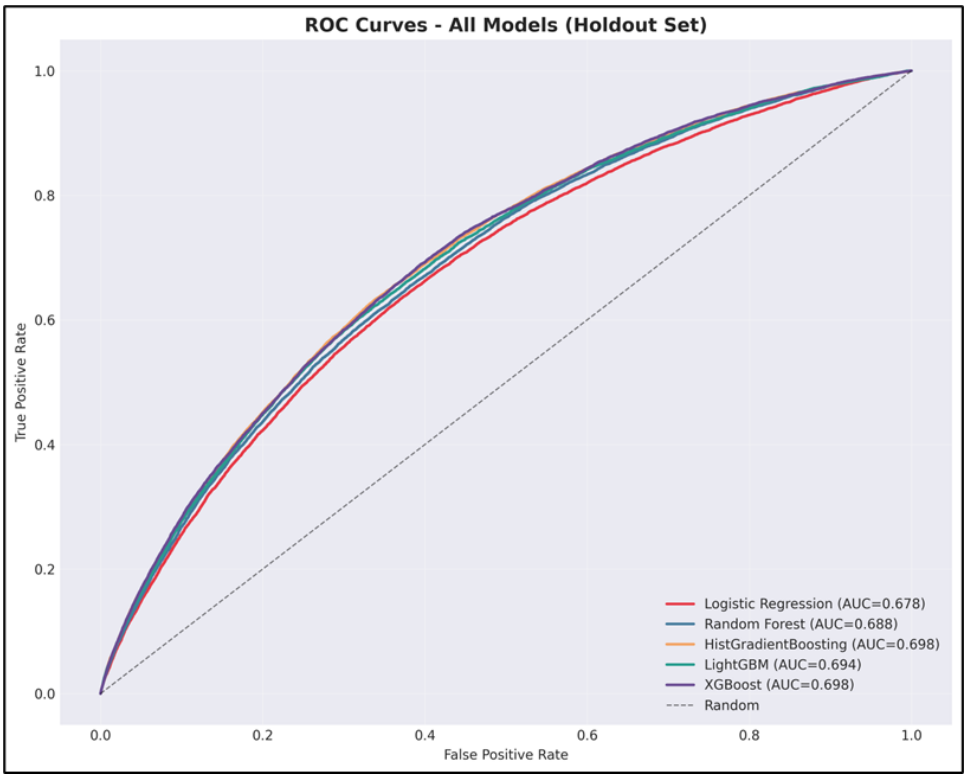
4. Modeling Pipeline

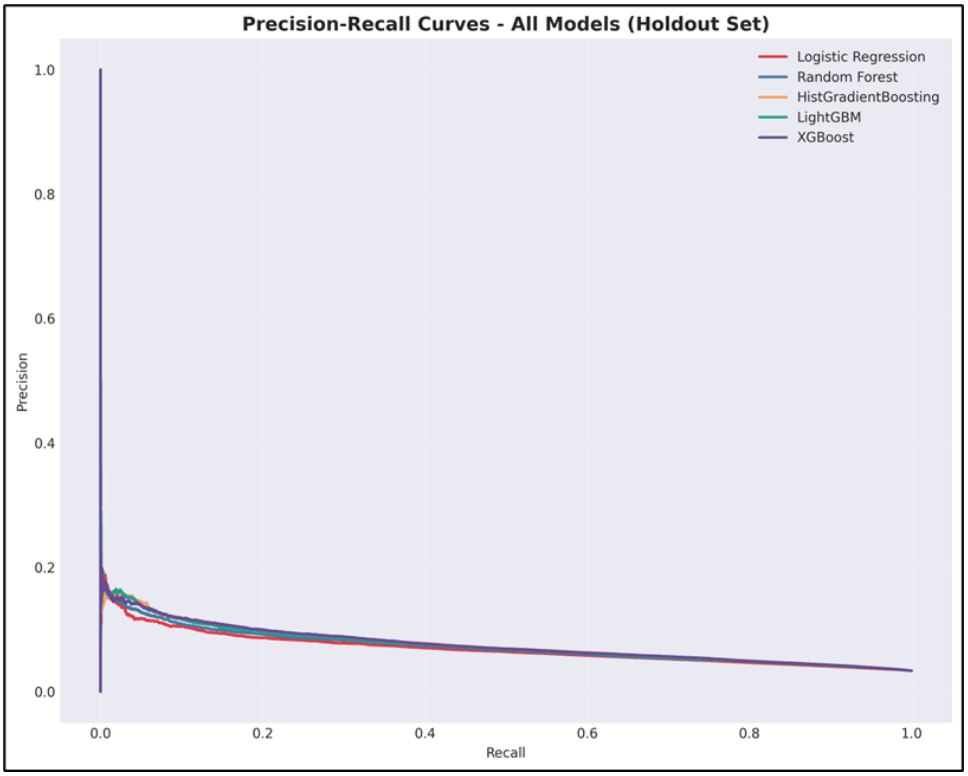
1. Preprocessing: Scale features if needed (Logistic Regression uses StandardScaler).
2. Model Training: Five models trained independently:
    - Logistic Regression
    - Random Forest
    - HistGradientBoosting
    - LightGBM
    - XGBoost
3. Validation: Early stopping for boosting models to prevent overfitting.
4. Metrics Computed:
    - AUC (Train, Validation, Holdout)
    - Accuracy (Train, Validation, Holdout)
5. Best Model Selection: Based on highest validation AUC.

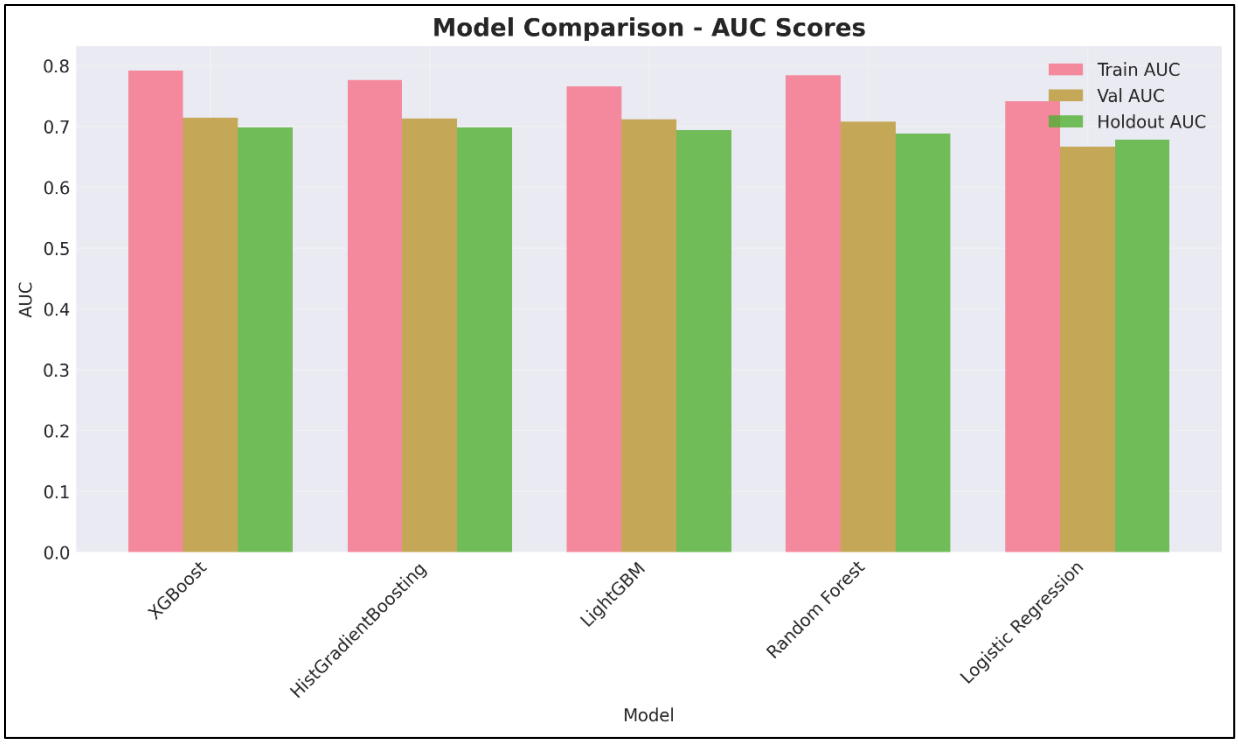
5. Evaluation Approach

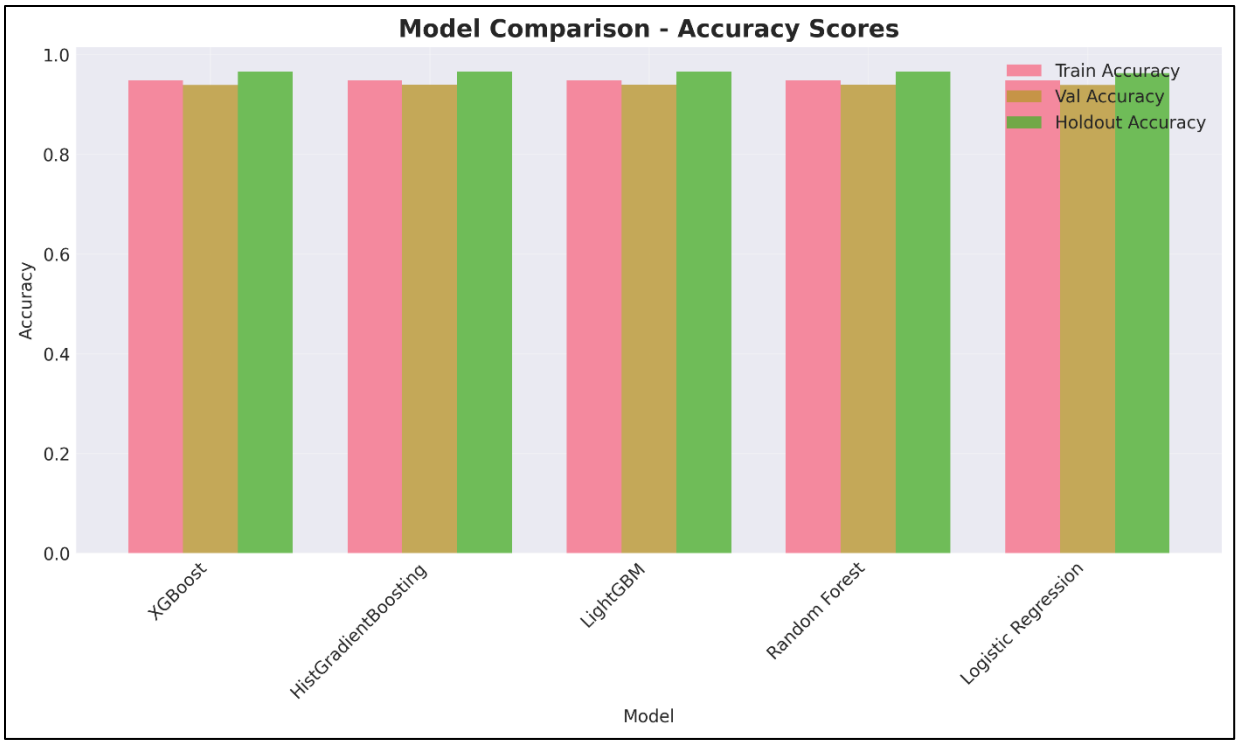
- Hold-out set evaluation: Measure performance on unseen data.
- Temporal evaluation: Track AUC and Accuracy over months.
- Segment × Time evaluation: Check model stability across customer segments and time.
- Visualization: Plots for comparison (bar plots, heatmaps, ROC, precision-recall, dashboard).

Charts


ROC Curves - All Models (Holdout Set)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Logistic Regression (AUC=0.678)
Random Forest (AUC=0.688)
HistGradientBoosting (AUC=0.698)
LightGBM (AUC=0.694)
XGBoost (AUC=0.698)
Random

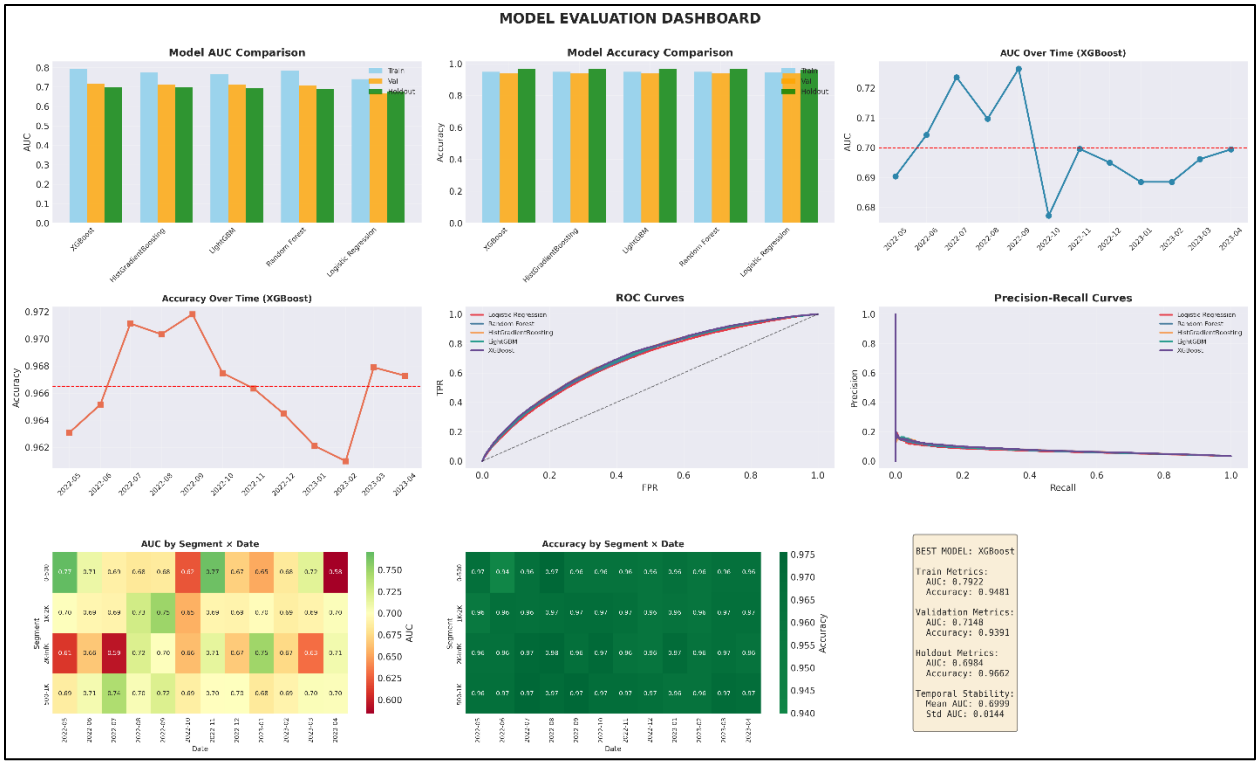

Precision-Recall Curves - All Models (Holdout Set)
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
Logistic Regression
Random Forest
HistGradientBoosting
LightGBM
XGBoost

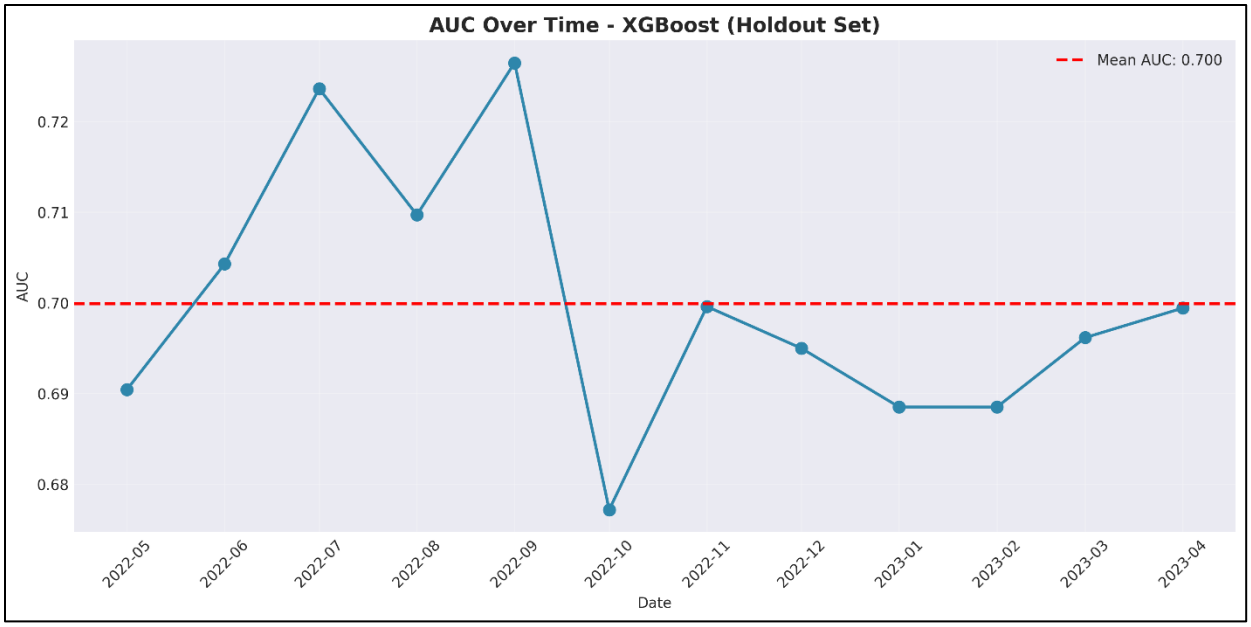

Model Comparison - AUC Scores
Train AUC
Val AUC
Holdout AUC
AUC
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
XGBoost
HistGradientBoosting
LightGBM
Random Forest
Logistic Regression
Model

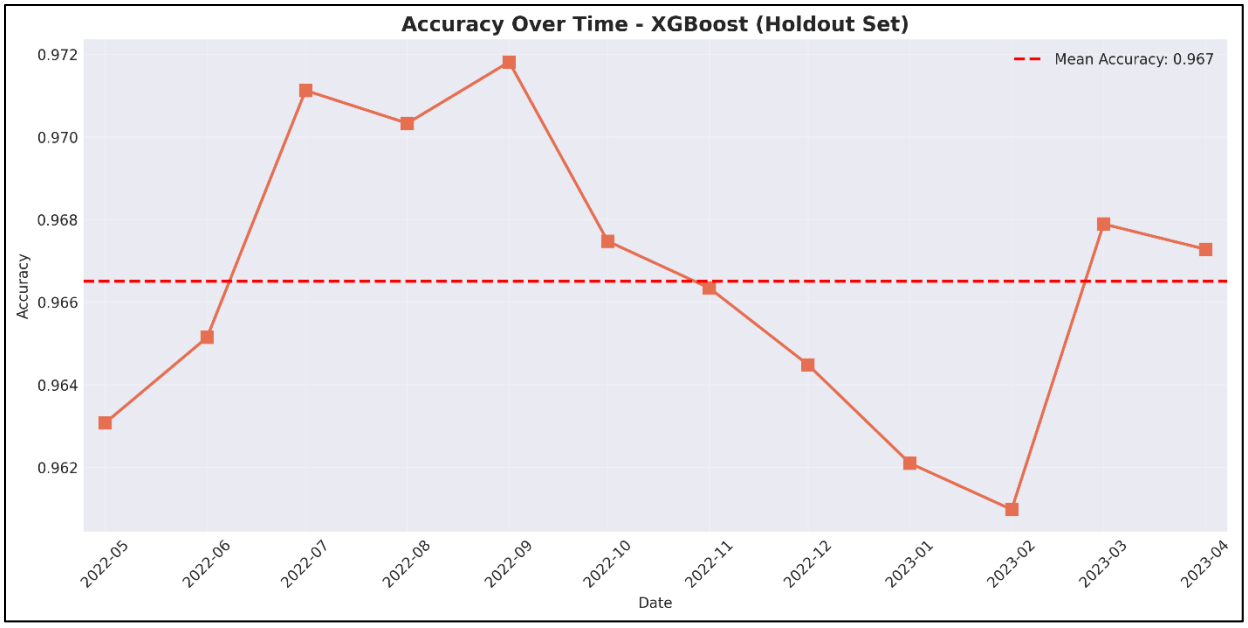

Model Comparison - Accuracy Scores
Train Accuracy
Val Accuracy
Holdout Accuracy
Accuracy
0.0
0.2
0.4
0.6
0.8
1.0
XGBoost
HistGradientBoosting
LightGBM
Random Forest
Logistic Regression
Model

```
================================================================================================================
MODEL PERFORMANCE METRICS
================================================================================================================
                     Model  Train_AUC  Val_AUC  Holdout_AUC  Train_Accuracy  Val_Accuracy  Holdout_Accuracy
                   XGBoost   0.792163 0.714811     0.698428        0.948081      0.939128          0.966159
HistGradientBoosting       0.776566 0.713444     0.698321        0.948009      0.939188          0.966241
                  LightGBM   0.766401 0.711470     0.694144        0.947968      0.939201          0.966269
             Random Forest   0.783889 0.707824     0.687978        0.947970      0.939192          0.966269
       Logistic Regression   0.741349 0.666598     0.677898        0.947918      0.938411          0.962778
================================================================================================================
```

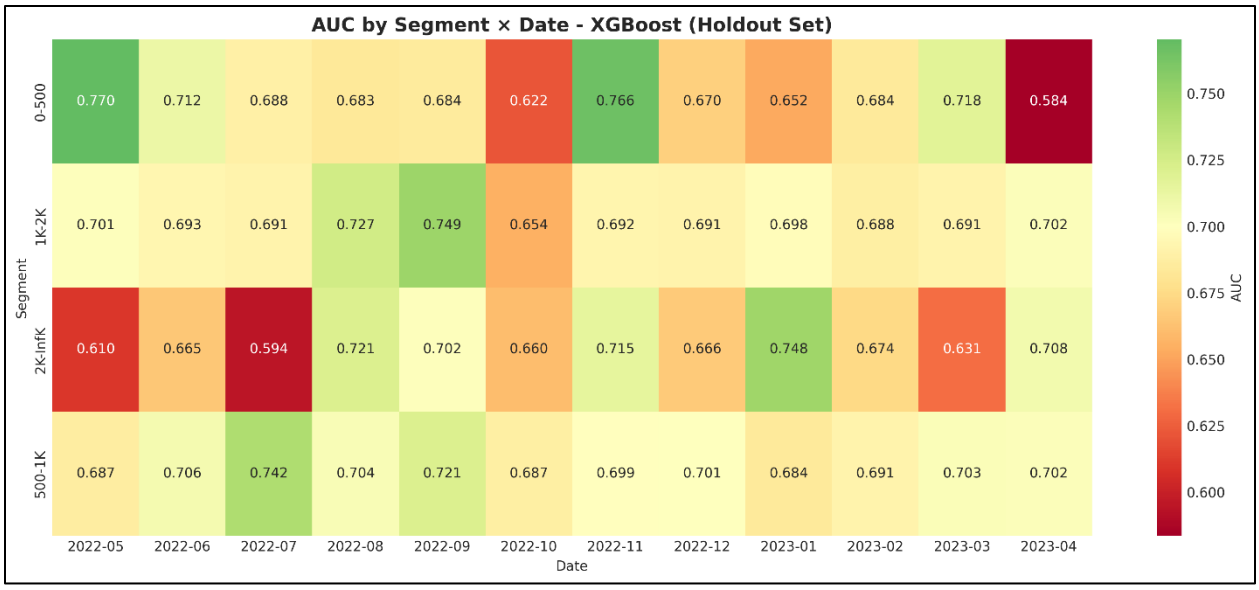
- For model comparison, there is not much different between models.
- Since the data is imbalance, validation AUC is preferred to evaluate model performance.
- Following charts and table, XGBoost achieves the highest validation AUC (0.7148) compared to other models.
- Thus, XGBoost is shown as the best model in this report.

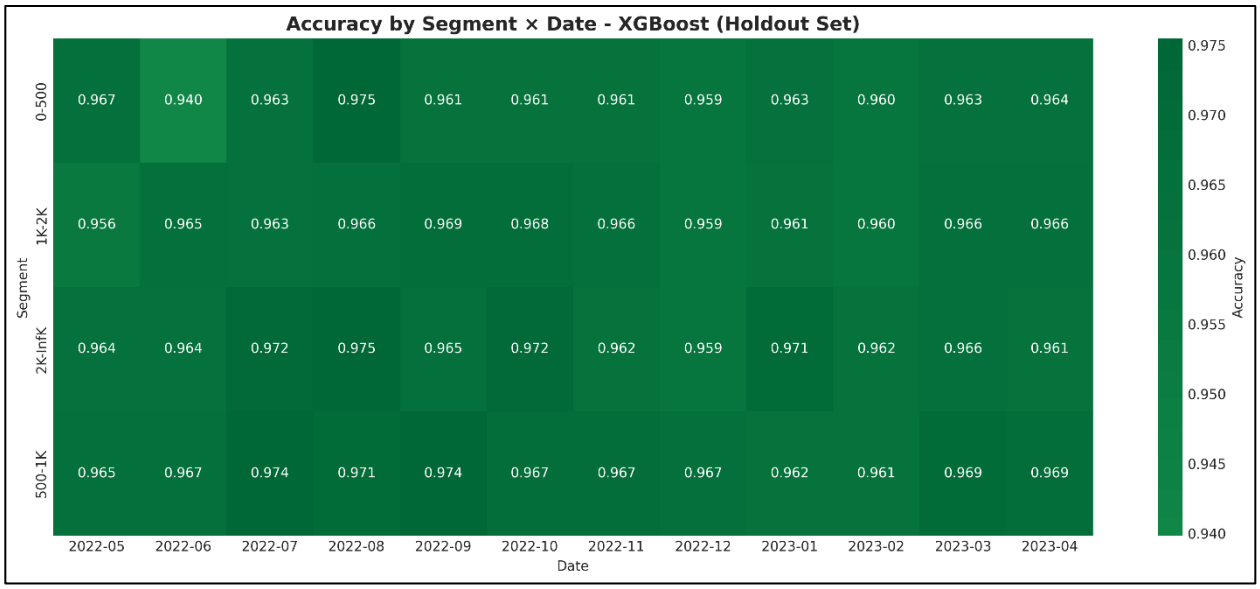






- In evaluation, following the best model XGBoost, the highest AUC (by Segment x Date_ym) is 0.77 with 0-500 in November 2022, while the lowest AUC (by Segment x Date_ym) is 0.58 with 0-500 in April 2023.
- Due to the imbalance dataset, accuracy metrics is overfitting (by Segment x Date_ym).





- In evaluation, following the best model XGBoost, the highest AUC (by Segment x Date_ym) is 0.77 with 0-500 in November 2022, while the lowest AUC (by Segment x Date_ym) is 0.58 with 0-500 in April 2023.
- Due to the imbalance dataset, accuracy metrics is overfitting (by Segment x Date_ym).

Conclusion

This approach is a multi-model, time-aware binary classification pipeline. It combines:

- Gradient boosting models for high predictive performance
- Linear model baseline for interpretability
- Temporal and segment evaluation for stability and reliability

It's a structured pipeline: data loading and merging → feature engineering → train/validation/test split → model definitions (5 models: LG, RF, HGB, LGBM, XGB) → model training → best model selection → hold-out set evaluation → metrics table → temporal evaluation (by Date_ym) → Segment x Date_ym evaluation → visualization (individual plots) & dashboard → final summary.

Supplementary

PPT slides is enclosed with this report.

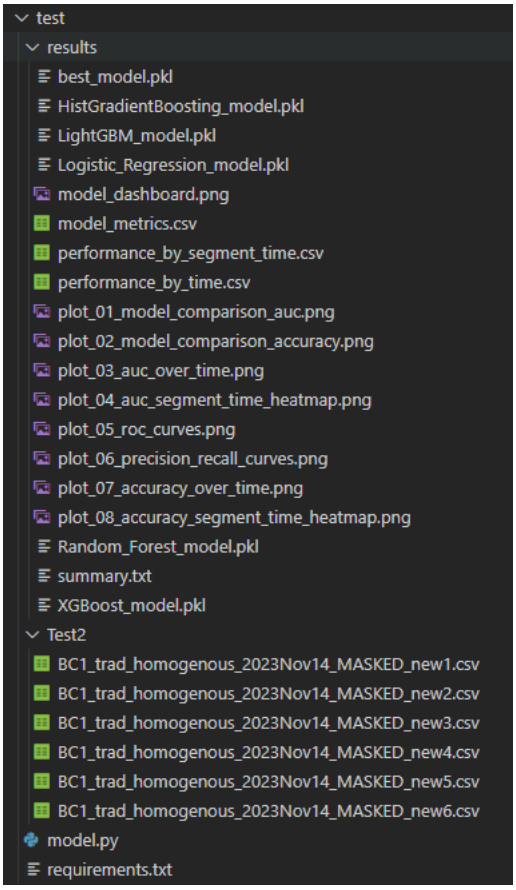
To run the code, please download the "test" zip folder including: model.py, requirement.txt.

All results are saved in the "results" folder.

Following command to create conda environment, install packages, and run python file.

```
conda create-n test python=3.10.13
conda activate test (or source activate test)
cd /your filepath/./.
pip install -r requirementx.txt
python model.py --data-dir ./Test2 --out-dir ./results
```

Below is the folder structure.


test
results
best_model.pkl
HistGradientBoosting_model.pkl
LightGBM_model.pkl
Logistic_Regression_model.pkl
model_dashboard.png
model_metrics.csv
performance_by_segment_time.csv
performance_by_time.csv
plot_01_model_comparison_auc.png
plot_02_model_comparison_accuracy.png
plot_03_auc_over_time.png
plot_04_auc_segment_time_heatmap.png
plot_05_roc_curves.png
plot_06_precision_recall_curves.png
plot_07_accuracy_over_time.png
plot_08_accuracy_segment_time_heatmap.png
Random_Forest_model.pkl
summary.txt
XGBoost_model.pkl
Test2
BC1_trad_homogenous_2023Nov14_MASKED_new1.csv
BC1_trad_homogenous_2023Nov14_MASKED_new2.csv
BC1_trad_homogenous_2023Nov14_MASKED_new3.csv
BC1_trad_homogenous_2023Nov14_MASKED_new4.csv
BC1_trad_homogenous_2023Nov14_MASKED_new5.csv
BC1_trad_homogenous_2023Nov14_MASKED_new6.csv
model.py
requirements.txt