Data Scientist II Technical Challenge

Task 2: Predictive Modeling

Cristina Sánchez Maíz | csmaiz@gmail.com | LinkedIn

Using the cleaned Customer Transactions dataset from Task 1:

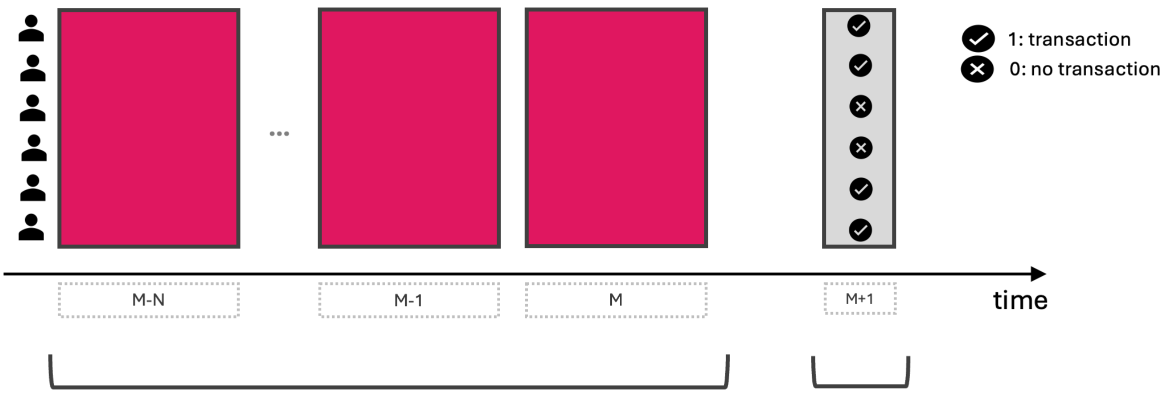
- Identify a target variable for prediction (e.g., predicting customer churn, transaction amount).
- Develop a predictive model using an appropriate machine learning algorithm.
- Evaluate the model's performance using relevant metrics (e.g., accuracy, precision,recall, RMSE).

Deliverables:

- Explanation of the chosen target variable and model.
- Model training and evaluation process.
- Performance metrics and interpretation.

Explanation of the chosen target variable and model.

*Given data until month M, the goal is to **predict** whether each customer will make a transaction in the following month M+1*



Input features are computed with data until month M

Target is computed with data from month M+1

Target variable

I have created a binary classification model where the target variable is stated as follows:

Given all data until month M,

- target=1 if the customer will make a transaction next month
- target=0 otherwise

Input features

For the input features, I used the information available not only for month M but for the N previous months, trying to capture the purchase pattern of each customer.
The features are:

- Number of transactions in month M, M-1, M-2, ..., M-N
- Total amount of the transactions in months M, M-1, M-2, ..., M-N
- Total number of months with transactions in the last R months

In Task 1, I discarded the customer features (customer_age and customer_income) so I do not have customer specific features to use in the predictive model.

Example.: For M=202405 (May 2024), N=3 and R=2, the input features are:

- Number of transactions in May (M), April (M-1), March (M-2) and February (M-3)
- Total amount of the transactions in months May (M), April (M-1), March (M-2) and February (M-3)
- Total number of months with transactions in the last R months (May and April). This is a variable that takes values from 0 to R.

Model training and evaluation process

Training algorithms

I used the scikit_learn library to create train and evaluate the models. As algorithms I applied Logistic Regression (LR), Random Forest (RF) and Extreme Gradient Boosting (XGBoost).

Evaluation metrics

Since the training datasets are balanced, **accuracy** is a good indicator of the overall model performance. It computes the proportion of correct predictions.

In the code, the classification_report function from scikit_learn is called. It computes also the precision and f1-score, between others.

Performance metrics and interpretation

Accuracy (% of samples correctly classified)

	202403	202404	202405	202404_05	202403_04_05
LR	50.00%	50.00%	40.00%	51.67%	50.00%
RF	36.67%	56.67%	46.67%	55.00%	53.33%
XGBOOST	40.00%	60.00%	56.67%	53.33%	62.22%

Column 202404_05 corresponds to the datasets 202404 and 202405 stacked so that the training has twice samples as individually.

Column 202403_04_05 corresponds to the datasets 202403, 202404 and 202405 stacked so that there are 3 times more training samples than individually.

Comments on performance

The accuracy is low but improves with increasing training samples stacking datasets.

Reasons:

- Few data: Training samples are
    - 80 for 202403, 202404 and 202405 columns
    - 160 for 202404_05 column
    - 240 for 202403_04_05 column
- No customer related information

Future work

- Data

- Consider adding more customers not only 100
- Add specific information about each customer

- ML algorithms

- Try another algorithms
- Hyperparameter Tuning