Data Scientist II Technical Challenge

# Task 2: Predictive Modeling

Cristina Sánchez Maíz | csmaiz@gmail.com | LinkedIn

---

## Table of Contents



**Task:**
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

I have built a predicted model summarized in Figure 1.

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

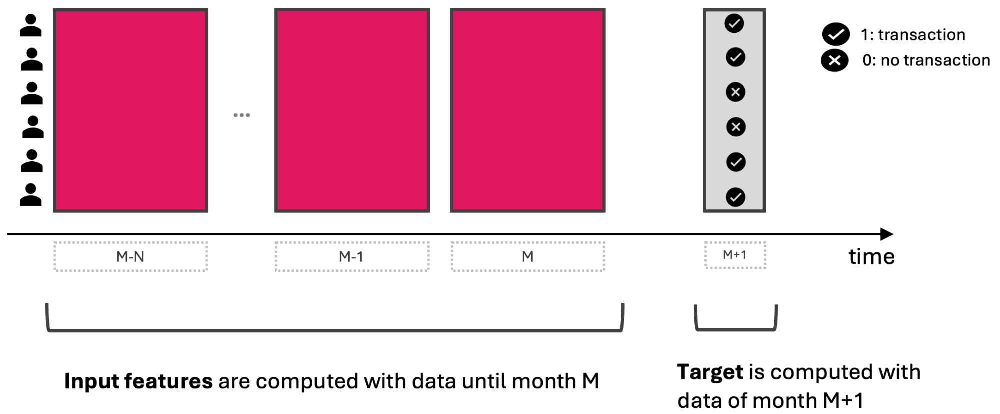



*Figure 1: Predictive model*

## 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.
These 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 consider these customer-specific features 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.

## 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 in month M+1
- target=0 if the customer won't make a transaction in month M+1

# Model training and evaluation process

## Training algorithms

I used the *scikit_learn* library to build and evaluate the models. As algorithms, I applied Logistic Regression (LR), Random Forest (RF) and Extreme Gradient Boosting (XGBoost) that are easy to apply for binary classification.

## 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.

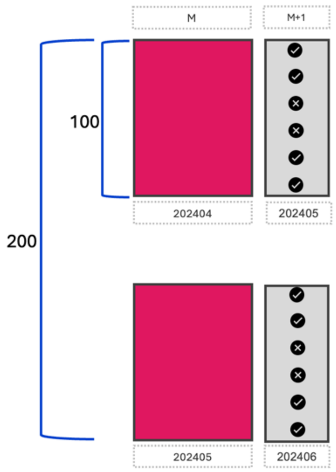
In the notebook, the *classification_report* function from *scikit_learn* is called. It provides additional metrics like precision and f1-score. For the sake of simplicity, in this report I only show accuracy.

# Performance metrics and interpretation

The dataset is splitted so that the 80% of the samples are used for training and the remaining 20% is used to assess the performance of the algoritms. Table 1 shows the accuracy for several dates and the three different algoritms.

| | M | | | | |
|---|---|---|---|---|---|
| | 202403 | 202404 | 202405 | 202404_05 | 202403_04_05 |
| **M+1** | 202404 | 202405 | 202406 | 202405/06 | 202404/05/06 |
| **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% |

*Table 1: Accuracy results*




Column 202404_05 corresponds to the datasets 202404 and 202405 stacked so that the training dataset twices the number of samples. Note that when M is 202404, M+1 is 202405; when M is 202405, M+1 is 202406. This is a trick to collect more training samples and try to improve the model performance.

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

*Figure 2: How to build the stacked dataset 2024_05*

# Comments on performance

The results shows that the accuracy is not very high. However, it increases as the training samples are incremented.

The following is a set of reasons to justify the accuracy results:

**Low training/test size:**

- Training samples are:
  - 80 for 202403, 202404 and 202405 columns
  - 160 for 202404_05 column
  - 240 for 202403_04_05 column

- Test is done with:
  - 20 samples for 202403, 202404 and 202405 columns
  - 40 samples for 202404_05 column
  - 60 samples for 202403_04_05 column

**No customer-related information**

Training dataset does not contain any specific information about each customer that can help to model their purshase profile.

# Future work

Suggestions to improve the prediction performance:

- Consider adding more customers not only 100.
- Add specific information about each customer. In order to understand the customer purchase pattern, data like:
  - personal information (age, income, location, ...)
  - traffic consumption (time/GB spent in shopping apps/webs, ...)
  - interests (social networks, shopping, traveling, electronics, ...)
  - customer journey within e-commerce platforms (items pending in shopping cart, ...)
- Try another ML algorithms and consider an exhaustive Hyperparameter Tuning.