Valerio Ganci, 03.03.2024

Conversion Rate Forecast - Technical Report

1. Introduction

The goal of the case study is to forecast the conversion rate, i.e. the ratio between the bookings and the clickouts, in 2023.08.11 for each hotel-advertiser couple present in the metrics dataset. The dataset span 10 days between 2023.08.01 to 2023.08.10. An additional dataset, the hotels one, contains information about each hotel, that could help in the conversion rate prediction.

After the phases of data cleaning, EDA, and feature engineering, the best strategy to predict the target variable consists of a first XGBoost binary classification model and a subsequent XGBoost regression model. The classification model predicts if the conversion rate is going to be zero (0) or strictly higher (1). The regression model improves the conversion rate predictions for the hotel-advertiser couples that were predicted to have a strictly positive conversion rate.

2. Data Overview

1. The data consists of 2 datasets, called Hotels and Metrics. The Hotels dataset has 5 columns, being hotel_id, stars, n_reviews, rating, and city_id. The hotel_id is a unique identifier of a given accommodation; stars is the number of stars of the corresponding hotel; rating is the average rating that is given by users on the corresponding accommodation with a 0-10 scale; n_reviews is the number of reviews used to get the rating; city_id is a unique identifier of the city where the corresponding hotel is located. The Hotels dataset has 224 rows corresponding to 224 unique hotels.
2. The Metrics dataset has 5 columns, being ymd, hotel_id, advertiser_id, n_clickouts, n_bookings. The ymd column lists dates on a YYYYMMDD format; hotel_id is the same as in the Hotels dataset; advertiser_id is a unique identifier of advertiser; n_clickouts is the number of clicks from customers on a particular hotel item, from a particular advertiser, and on a particular day; n_bookings is the number of bookings made by customers on a particular hotel item, from a particular advertiser, and on a particular day.
3. Both datasets don't contain duplicate rows. Each day has a similar number of entries. The Metrics dataset doesn't contain any missing values. The Hotels dataset contains missing values in the n_reviews and rating columns for the hotel ID = 216.
4. The features of both datasets show distributions which deviate from a normal shape, with the presence of outliers. Particularly problematic are the highly right-skewed distributions of the target variables n_clickouts and n_bookings. The latter is sparse, meaning that most values are null (83% of bookings entries). For these features, we cannot consider the values higher than 0.75 percentile as outliers as they are an intrinsic characteristic of the dataset.

3. Data Cleaning & Feature Engineering

1. The Hotels and Metrics datasets were merged on their common hotel_id column. From the date column, 3 additional columns were created, respectively listing the day, the weekday (0 corresponding to Monday and 6 to Sunday), and if the day is a weekend day (1 if it is, 0 if it is not).
2. The n_reviews and rating missing values for the hotel having ID of 216 (which has 0 stars and is located in city ID = 3) were imputed from the median values of these features calculated in the subset of the hotels having 0 stars and city ID = 3.

3. The hotels, advertisers, and cities were given a new ID based on their total number of bookings. A higher n_booking value will correspond to a higher ID. This will help the training of our models.
4. To reduce the n_reviews distribution skewness, a log10 transformation was applied. The remaining outliers were set to a limit value calculated as the percentile(0.75) + 1.5 $\times$ iqr or as percentile(0.25) - 1.5 $\times$ iqr, where iqr is the interquartile range, i.e. percentile(0.75) - percentile(0.25).
5. The hotel having ID of 214 (0 stars, 480 reviews, 9.2 rating) is an outlier, having a number of clickouts (> 600) and bookings (> 5) that don't follow the trend of the hotels having the same hotel metrics. Therefore, the stars, reviews, and rating values for this hotel were imputed to follow the overall trend of the dataset.
6. As the n_bookings and n_clickouts features are right-skewed, so is the distribution of their ratio, the conversion rate. A log10 and a Box-Cox transformation did not improve the situation.
7. Since the future values of the conversion rate may depends on their past values, I created lagged features (of 1 and 2 days) of the conversion rate, bookings, and clickouts. An additional feature was created as the difference between the 1-day-lagged and the 2-day-lagged conversion rates.

4. EDA

1. Some hotels, advertisers, and cities have more clickouts and bookings than others.
2. The number of bookings is higher than $\sim$5 when the number of clickouts is higher than $\sim$200.
3. The top 5 hotels in terms of bookings have ID: 73, 36, 26, 33, 35.
4. The top 5 hotels in terms of clickouts have ID: 36, 73, 26, 214, 35.
5. The top 5 cities in terms of bookings have ID: 34, 3, 51, 67, 46.
6. The top 5 cities in terms of clickouts have ID: 34, 3, 46, 51, 27.
7. The top 1 city has $\sim$4 times the amount of clickouts and bookings then the second top one.
8. The top 5 advertisers in terms of bookings and clickouts have ID: 5, 39, 37, 24, 1.
9. The top 1 advertiser has $\sim$4 times the amount of clickouts and bookings then the second top one.
10. The total (all hotels and advertisers) daily bookings and clickouts are nearly constant in time, except for the last day (-25% bookings and -12% clickouts).

5. Model Creation & Evaluation

1. The merged dataset was split in 2 subsets, a train and a test one. The test subset contains only the last two days (2023.08.09 - 2023.08.10) and correspond to 20% of the entire dataset. The train subset contains the remaining days.
2. A first attempt was made by training an XGBoost regression model with the following predictors: 'new_hotel_id', 'new_advertiser_id', 'new_city_id', 'stars', 'log10_reviews', 'rating', 'day', 'weekday', 'is_weekend', 'conversion_rate_lag1', 'conversion_rate_lag2', 'conversion_rate_var', 'bookings_lag1', 'clickouts_lag1'. The model has a R^2 score of 0.026, meaning that the model can explain 2.6% of the target variable variance. The model therefore does not perform significantly better than a horizontal line fitting the data (a model that simply predicts the mean of the target variable). Removing some predictors in the training phase does not improve the situation. The two steps approach described in the intro is therefore used.

3. An additional column is created as a binary conversion rate, 0 if the conversion rate is null, 1 otherwise. Lagged features are created for the binary conversion rate.
4. A first Naive-Bayes classification model, created to predict if the conversion rate is going to be zero (0) or higher (1), has accuracy and recall of 80.4%, and a precision of 85.0% on the test dataset. The model was trained on the following predictors: 'new_hotel_id', 'new_advertiser_id', 'new_city_id', 'stars', 'log10_reviews', 'rating', 'day', 'weekday', 'is_weekend', 'conversion_rate_bin_lag1', 'conversion_rate_bin_lag2', 'conversion_rate_bin_var', 'bookings_lag1', 'clickouts_lag1'.
5. The best XGBoost classification model created, has accuracy and recall of 87.1%, and a precision of 85.7% on the test dataset. The model requires 3 predictors: hotel ID, advertiser ID, and city ID. When training the model on the predictors listed in point 4, the scores are slightly better ($+0.6\%$ in accuracy and recall, $+0.5\%$ in precision).
6. The best XGBoost regression model created, applied on the results obtained with the XGBoost classification model, has a R^2 score of 0.469 on the test dataset, i.e. the model can explain 46.9% of the target variable variance of the test dataset. In this case, the target variable was transformed with a yeo-johnson transformation to deal with the conversion rate distribution skewness. The model requires only two predictors: hotel ID, and advertiser ID. When the model is trained on the predictors listed in point 1, the score improves by 2.0%.
7. Both models are therefore time-independent.
8. Higher feature importance lays on the hotel ID, followed by the advertiser ID, and finally the city ID.

6. Insights and Next Step

Key Insights:

1. The number of bookings is $\gtrsim 5$ when the number of clickouts is $\gtrsim 200$
2. The top 5 hotels in terms of bookings have ID: 73, 36, 26, 33, 35
3. The top 5 hotels in terms of clickouts have ID: 36, 73, 26, 214, 35
4. The top 5 cities in terms of bookings have ID: 34, 3, 51, 67, 46
5. The top 5 cities in terms of clickouts have ID: 34, 3, 46, 51, 27
6. The top 1 city has ~ 4 times the amount of clickouts and bookings then the second top one
7. The top 5 advertisers in terms of bookings and clickouts have ID: 5, 39, 37, 24, 1
8. The top 1 advertiser has ~ 4 times the amount of clickouts and bookings then the second top one
9. The total (all hotels and advertisers) daily bookings and clickouts are nearly constant in time, except for the last day (-25% bookings and -12% clickouts)
10. The Naive-Bayes classification model, created to predict if the conversion rate is going to be zero (0) or higher (1), has accuracy and recall of 80.4%, and a precision of 85.0% on the test dataset
11. The best XGBoost classification model created, has accuracy and recall of 87.1%, and a precision of 85.7% on the test dataset. The model requires 3 predictors: hotel ID, advertiser ID, and city ID.
12. The best XGBoost regression model created, applied on the results obtained with the XGBoost classification model, has a R^2 score of 0.469 on the test dataset, i.e. the model can explain 46.9% of the target variable variance. The model requires only two predictors: hotel ID, and advertiser ID
13. Both models are therefore time-independent
14. Higher feature importance lays on the hotel ID, followed by the advertiser ID, and finally the city ID

Next Steps:

1. Apply oversampling techniques to reduce the class imbalance (null, non-null conversion rate)
2. Apply the regression step to the Naive-Bayes classification model results
3. Apply hyperparameter tuning to the models via GridSearchCV to improve the models' metrics
4. Test the performance of a Neural Network model
5. Increase the historical time span of the dataset
6. Retrieve additional predictors