Predict a click

Click prediction data science challenge

Background

At the Marketplace team at trivago we have the goal to provide the right accommodations for our users and help our advertisers (online travel agencies, hotel chains and hoteliers) to run well performing campaigns on our website. The balance between the interests of users and advertisers is established by the Exposure Algorithms, which determine what hotels and advertisers we present to the user and at what position they will be shown. One important objective of the algorithm is to anticipate how likely users are to click on a specific hotel. In this challenge we ask you to predict how often a hotel will be clicked based on certain characteristics of the hotel.

Legal Notice

Please keep in mind that this case study is proprietary. That means that you are not allowed to share your code with anybody other than trivago. Especially you may not upload it to websites like Github or BitBucket. Please be fair.

Expectations

- The model should be implemented either in R, Python, or Scala
- The deliverable for this challenge are a submission file with predictions (described below) and the modelling code (R/Python/Scala) along with explanations on the thought process behind each step
- The bonus questions are optional but they might help to reflect on the topic. Feel free to also submit the answers to them

Datasets

1. **train_set.csv**: in the training dataset you will find a list of hotels and hotel characteristics. Every row has a unique identifier for the hotel - the hotel_id. The column n_clicks specifies the number of clicks the hotel has received in a specified time frame. This is the target variable you have to predict.
   Number of rows: 396487.
2. **test_set.csv**: in the test set you will find a list of new hotels with the same characteristics as in the training set and derived in the same time frame. In this dataset the target variable that you have to predict is missing.
   Number of rows: 132162.
3. **sample_submission.csv**: an example submission file. It should contain only the unique hotel_ids from the test set and the number of predicted clicks.

The training dataset provided is not clean and may be missing data or contain nonsensical values. Also, the datasets below contain potentially hashed and/or modified values.

Evaluation

The predictions will be evaluated by a normalized weighted mean square error:

$$\text{error} := \frac{1}{n} \frac{\sum_{i=0}^{n} w_i \cdot (\text{predictedClicks}_i - \text{observedClicks}_i)^2}{\sum_{i=0}^{n} w_i}$$

where

$$w_i := \log(\text{observedClicks}_i + 1) + 1$$

Column description

hotel_id: a number uniquely identifying each hotel
city_id: describes the city the hotel is located in

content_score: describes the quality of the content that is provided for the hotel on a scale from 0 (worst) to 100 (best)
n_images: number of images that are available for the given hotel
distance_to_center: distance (in meters) of the hotel to the nearest city center
avg_rating: average rating of the hotel on a scale from 0 (worst) to 100 (best)
n_reviews: number of reviews that are available for that hotel
avg_rank: average position the hotel had in the list
avg_price: average price in Euro of the hotel
avg_saving_percent: average saving users achieve on this hotel by using trivago, i.e. the relative difference between the cheapest and most expensive deal for the hotel
n_clicks: the number of clicks the hotel has received in a specific time frame (target variable, unique to the training set)

Bonus questions

1. Can you describe in your own words what the purpose of the evaluation metric above is? What alternative metrics would make sense in this context?
2. We mention the click prediction as one component of our Exposure Algorithms. What other components would you include to determine what advertiser or hotel to show to our users?
3. Which of the input variables have a high predictive power? What additional variables would you include to reduce the error further?
4. In addition to the model you used to calculate your results what are alternative models could you use for the prediction problem? What are trade-offs between the model you used and the alternatives?