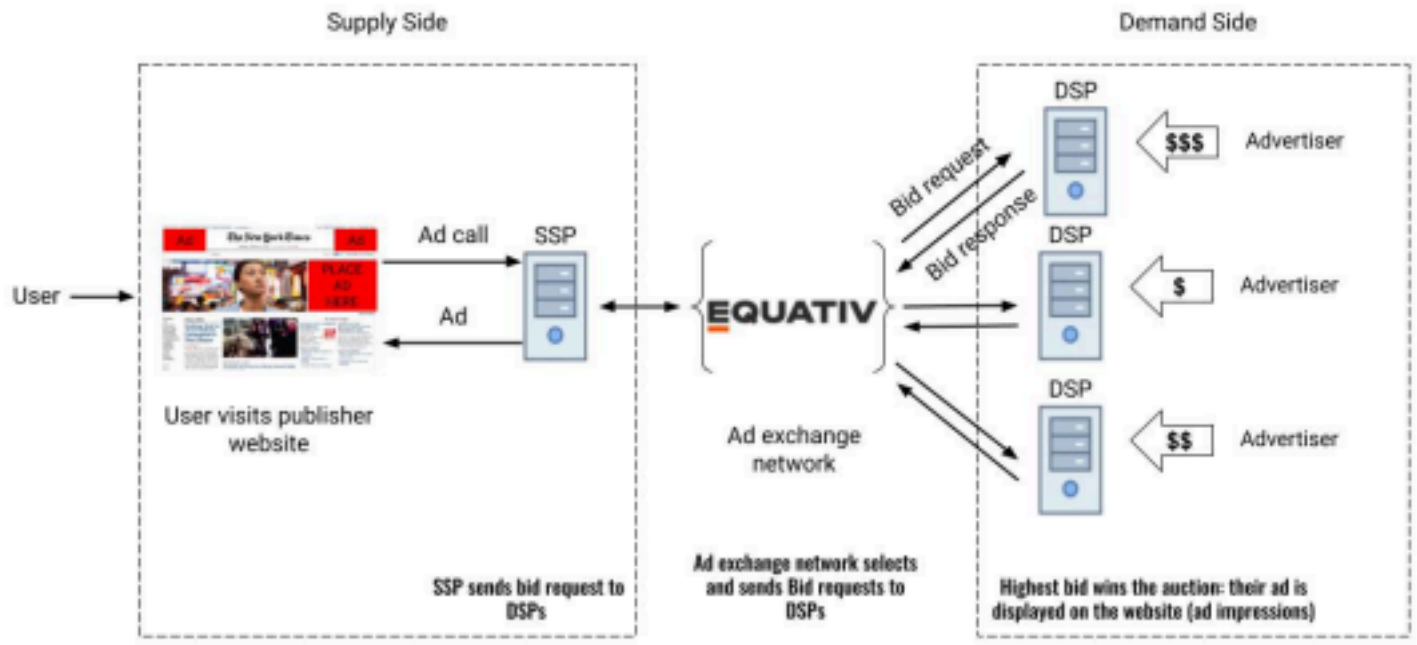
Technical Test - Data Scientist

Introduction



Equativ is a platform helping publishers or ad networks to sell their ad inventory using real time bidding.

The selling process, which is fully automatic, is described below :

Fig. 1. Equativ in the online advertising Ecosystem

From left to right, a process called auction is triggered when a user starts to interact with an online service through either visiting a web-page, searching an item, or checking the social media in a publisher website.

In the case that the web-page has an ad placement available, the publisher sends an ad request to Equativ (adserver and ssp).
The bid request related to the requested ads is forwarded from Ad Exchange to the DSPs which represent the advertisers.
The auction is set up to gather the bids of the DSPs: **a DSP may answer or not to our bid request with an offered price.**

The DSP with the highest bid wins the auction and its ad script is forwarded to the publisher to be embedded in the page viewed by the user.

For each Ad impression, Equativ would take a portion of the bidding price. We have to attract the highest bidding price from the DSPs in order to optimize the revenue of the publisher and ours.

Problem Statement

Context

One of the challenges of the DS team is the optimization of the number of DSP bid requests.

The first step for organizing an auction is to send the bid requests to the DSPs. This request “invites” a potential buyer to participate in the bidding process by responding with an offered_price. It contains data about the auction.

Equativ uses a huge computational power to deal with 10^11 auctions per day. Sending a bid request to every DSP would be a waste of resources as many of them are only interested in specific auctions. Besides, some DSPs are capable of handling a small amount of bid requests and only respond to auctions that are interesting to them.

Most of the bid requests are not answered by the buyers, this is what we call a “no bid” and it is a waste of resources.

As a result, our goal is to send only the most valuable bid requests to a DSP among those available.

The DS team has developed a model to classify a bid request with the following binary target :

- *is_revenue_opportunity* : true if the DSP decided to bid and returned a valid offered_price to the bid request

This value is not known when the decision to send a bid request is taken. However, it can be collected in the auction logs. It constitutes the "target" of the provided dataset.

Objective

The candidate should propose a metric to score the bid request computable **before it is sent to a DSP**.

Then, a machine learning model should be trained using the provided dataset.

You are free to use every feature provided in the anonymized dataset. However, some of them might not be informative.

Once you are happy with your model, compute its predictions on the test dataset and save them.

Dataset

Here are two download links on Google Cloud Platform containing a dataset curated by the DS team :

https://storage.googleapis.com/equativ-open-data/dataset.zip

https://storage.googleapis.com/equativ-open-data/test_dataset.zip

dataset.csv is a tabular dataset composed of 38 features and 1 target in csv format. It has been anonymized. You can use it to train and validate your model.

test_dataset.csv is composed of the same data but without the target. We will use your predictions on this test_dataset to evaluate your model.

Every row of the dataset is an example of a bid request sent to a DSP.

The name of the columns have been replaced for confidentiality, however we have separated the features in 3 groups.

- *user_LETTER* : information relative to the user browsing the internet
- *context_LETTER* : contextual information relative to the content of the publisher
- *buyer_LETTER* : information about the DSP the bid request was sent to

If the DSP responded with a correct bid, *is_revenue_opportunity* is True.

#	Name	Type
0	buyer_A	boolean
1	buyer_B	category
2	context_A	category
3	context_B	category
4	context_C	category
5	context_D	category
6	context_E	category
7	context_F	category

8	context_G	float
9	context_H	category
10	context_I	category
11	context_J	boolean
12	context_K	boolean
13	context_L	category
14	context_M	category
15	context_N	category
16	context_O	boolean
17	context_P	boolean
18	context_Q	boolean
19	context_R	boolean
20	context_S	category
21	context_T	category
22	context_U	category
23	context_V	category
24	context_W	category
25	context_X	category
26	context_Y	category
27	context_Z	category
28	context_ZA	category
29	user_A	category
30	user_B	boolean
31	user_C	category
32	user_D	category
33	user_E	category

34	user_F	category

35	user_G	category
36	user_H	category
37	user_I	category

Target	Type
is_revenue_opp b	

Sample weight : None

Expected Deliverables

We expect

- the source code, preferably in the form of a Jupyter notebook
- a requirements.txt allowing the recruiter to run your code and reproduce your results - a csv file containing the predictions of your best model computed with the test dataset

The notebook should include a description of your methodology, your model and areas for improvement.

Please send the deliverables by replying to the email received.

Don’t hesitate to contact us if you have any questions, we will not penalize you for asking questions.

Good luck !

***