Ads Recommendation

by AvmnuSng

Introduction

The *OLX* Group is a network of leading classifieds platforms in 40 markets, including the brands *OLX*, *Avito*, *letgo*, and *dubizzle*. The OLX Group connects local people to buy, sell or exchange used goods and services by making it fast and easy for anyone to post a listing through their mobile phone or on the web. Hundreds of millions of people in local markets around the world generate more than 1.7 billion monthly visits, 35 billion monthly page views and 54 million listings every month on the OLX Group's online marketplaces.

OLX is one of the largest e-commerce sites worldwide. OLX is designed to enable people to connect through buying and selling goods and services quickly, easily and for free. We try to create a richer world by bringing people together for win-win exchanges. Sellers win. Buyers win. Communities win. Earth wins.

In the context of Online Classifieds, buyers want to find ads that are relevant to them. When people come to browse our ads, they find themselves looking through a lot of products in which they may not be interested. We have categories where the users can find thousands of published products.

In this case, to help our users find those relevant ads we want you to build a recommender system to predict a maximum of *ten* ads, we should show in a specific category for a particular user.

Dataset

You are given three training datasets, `user_data.csv`, `ads_data.csv` and `user_messages.csv`. Each of this files is a comma separated file with useful information for this task:

- `user_data.csv` contains user behavior of the 30 days before 2017-06-17 00:00:00GMT. Each row is an interaction of the user with the platform. Each interaction could be a "view" or a "first chat" of a particular ad. Additionally to the timestamp of that event and the location of the user you will have information about the ad it interacted with. Information such as views, impressions, and messages that it received. Take care that this information change over time so you have the information of the ad at the time of the event. The definition of each attribute is as following:
  - `"event_time"` : The time of the event in milliseconds.
  - `"user_id"` : The id of the user.
  - `"event"` : The name of the event, it could be "view" for a viewed ad or "first_message" for the first message that ad receives from this user.
  - `"channel"` : The platform the user is using (android, ios, etc.).
  - `"user_lat"` : The user latitude at the moment of the event.
  - `"user_long"` : The user longitude at the time of the event.
  - `"origin"` : If the user came from the homepage, or it was exploring a category, or the user did a search.
  - `"ad_id"` : The id of the ad.
  - `"images_count"` : The number of images in the ad.
  - `"ad_impressions"` : The number of impressions of the ad at the time of the event.
  - `"ad_views"` : The number of views of the ad at the time of the event.

- `"ad_messages"` : The number of users that messaged this ad at the moment of the event.

- `ads_data.csv` contains the ads and their information. Each row provides information on each ad. Information such as ad title, description, location, time of creation, etc. The ads shown here have an enabled attribute telling if that ad can be displayed to the users in the following days. The definition of each attribute is as following:

  - `"ad_id"` : The id of the ad.
  - `"category_id"` : The category of the ad.
  - `"seller_id"` : The id of the user selling this item (for privacy reasons this id won't match the one from users_data).
  - `"creation_time"` : The time when the ad was created.
  - `"title"` : The title of the ad.
  - `"description"` : The description of the ad.
  - `"price"` : The price of the ad.
  - `"lat"` : The latitude of where the ad is located.
  - `"long"` : The longitude of where the ad is located.
  - `"source"` : The the channel through which the ad was posted (android, ios, etc.).
  - `"enabled"` : If the ad is enabled at the time of prediction.

- `user_messages.csv` contains for each user the ads s/he will chat in the seven days following 2017-06-17 00:00:00GMT.

We also provide a test set of **10507** users and categories pairs, in `user_messages_test.csv`. The goal is to predict a maximum of ten ads user will chat to in the next seven days for the given categories.

You can download the zip file provided **here**. The zip file contains all the training and test files.

Submission Details

You are required to upload the following three files:

- The output file, `ads_recommendation.csv` (max allowed size is `10MB`). The file should contain each `user_id` and `category` from the file `user_messages_test.csv` followed by a list of the recommended ads contained within square brackets and enclosed by double quotes.

  A valid output file has the following format:

```
user_id,category_id,ads
1,362,"[414182, 1068300]"
1,800,"[2872471, 2022462, 2572489, 2865634]"
1,806,"[2844639, 2876503, 2608868, 2866022]"
1,811,[2142756]
1,815,"[2591735, 2165051]"
1,881,"[2770058, 2459839]"
1,887,[2879400]
1,888,"[1279184, 2806977]"
2,362,"[1963811, 268407]"
2,800,"[2887222, 1716931, 2843966, 2853355, 2371107, 2398004, 2875717]"
```

  Note that the first line of the output file should contain the header, with `user_id`, `category_id` and `ads` as the column names separated by a comma.

- A *PDF* file (maximum allowed size is `4MB`) providing the findings and justification on the following topics:

  - Write a few lines about training dataset quality and any errors found in the training dataset.
  - Explain the data preprocessing steps.
  - Explain and justify the model you've chosen for the recommender system.

- The source code of your approach for this task. Upload a *zip* file (maximum allowed size is `5MB`) with all relevant files to reproduce your results. The submitted file must have a `README` file with a detailed description about how to run the model to recommends the ads list for the categories and generate the `ads_recommendation.csv`. Do not forget to include links to any external libraries or packages you use for the generation of your model.

There is no limit on execution time, but the code should generate the output file: `ads_recommendation.csv`.

Evaluation

Our users will check a particular category, and if they do not find a single interesting ad in the first ads, they will probably think nothing is interesting for them. So we want you to have high precision, in particular with the first ones. As we are interested in the precision of the first ads and the order in which they were shown, we will measure average precision. Average Precision at 10, or $\mathbf{AveP@10}$, is calculated as:

$$\mathbf{AveP@K} = \frac{\sum_{i=1}^{K} \mathbf{P(i)} \times \mathbf{rel(i)}}{\mathbf{Number\ of\ relevant\ items}}$$

Where $\mathbf{P(i)}$ means Precision at i and is calculated taking as the fraction of true positives for the first i items recommended. And $\mathbf{rel(i)}$ is 1 if the recommended ad i is relevant or not, K is the amount of recommended items. Finally, we will take the mean of the $\mathbf{AveP@10}$ for every combination, giving us the $\mathbf{MAP@10}$. For more information, please check Wikipedia.

Ranking

Before the end of the contest, all evaluation of the $\mathbf{AveP@10}$ value of your uploaded model will be performed on pre-selected 5271 users and categories pairs. At the end of the contest, your *last* uploaded file (i.e., the most recently uploaded file) will be used to calculate your final score and position on the leaderboard. Because of this, make sure that your final (very last) submission is the output file with the maximum score.

File Upload

Upload your files here:

ads_recommendation.csv:

Documentation:

Source Code:

Contest ends in a day

Max Score: 1000

Rate This Challenge:

Download problem statement

Download sample test cases

Suggest Edits

Collapse