OLX- Code and the Curious Ads Recommendation

Harsh Kara

July 2017

1 Data-set Quality

The data-set quality as such was nice, there were appropriate number of entries. These were not to large thus helped in faster generation of the results.

2 Data Prepossessing

- Firstly I converted the date and time given into date-time objects, and extracted the month and the day from it.

- Then I made separate data-frame **ads_cat, ads_price, ads_enabled**.These are used later to find the category, price, and whether ad is enabled or not. This work like hash to find the information given the add_id

- Also I have mapped the the event to integer 0 and 1.Also the origin being a categorical feature is also mapped to integer values, assigning the integer according to their relevance.

3 Model for Recommendation

For recommending I have first sorted the user_data given according to the following features in order. **by=['month','day','origin','ad_messages','ad_views','ad_impressions'] ,ascending=[False,False,True,False,False,False]** After this for each user(uid),

and category(cid) given in the test data, I have done the following steps.

- First found the ads corresponding to the user uid.

- Then found the category corresponding to the ads found in previous step.

- Then Removed all the rows where add id is different from cid. This gives add relevant for user with the category we are interested in

- Then I removed the adds which are not to be enabled at the time of prediction.

- Then I used the remaining ads which are not also in the sorted order only, and chose the top 10 ads as final result

Apart from many, one optimization I did was that there were some users in test which are not encountered in the user_data matrix. For those I have pre-computed the top 10 ads taking whole ads in place of the step 1 and then applied all the remaining steps. Since many user are not present in the user_data so pre-computing this saves a lot of time.