Vishal Changrani

Capstone Report – Kaggle Competition: Outbrain Click Prediction

Mentor: Ankit Jain

Springboard Data Science Intensive

03/21/2017

Table of Contents

Problem definition

As my capstone project I attempted the Outbrain Click Prediction competition hosted by Kaggle

https://www.kaggle.com/c/outbrain-click-prediction

Outbrain provides personal content recommendation across thousands of sites. The recommendation is in the form of a table referred to as a 'display' showing two to twelve different content options (ads) on different publisher pages. In this competition the task was to predict which of the ads is the user most likely to click on. The submission was in the form of a list of document ids ordered in descending order of likelihood of being clicked for each display in the test data. Hence it was **supervised classification problem**.

Following is an example of an Outbrain display shown on a page of CNN.com,

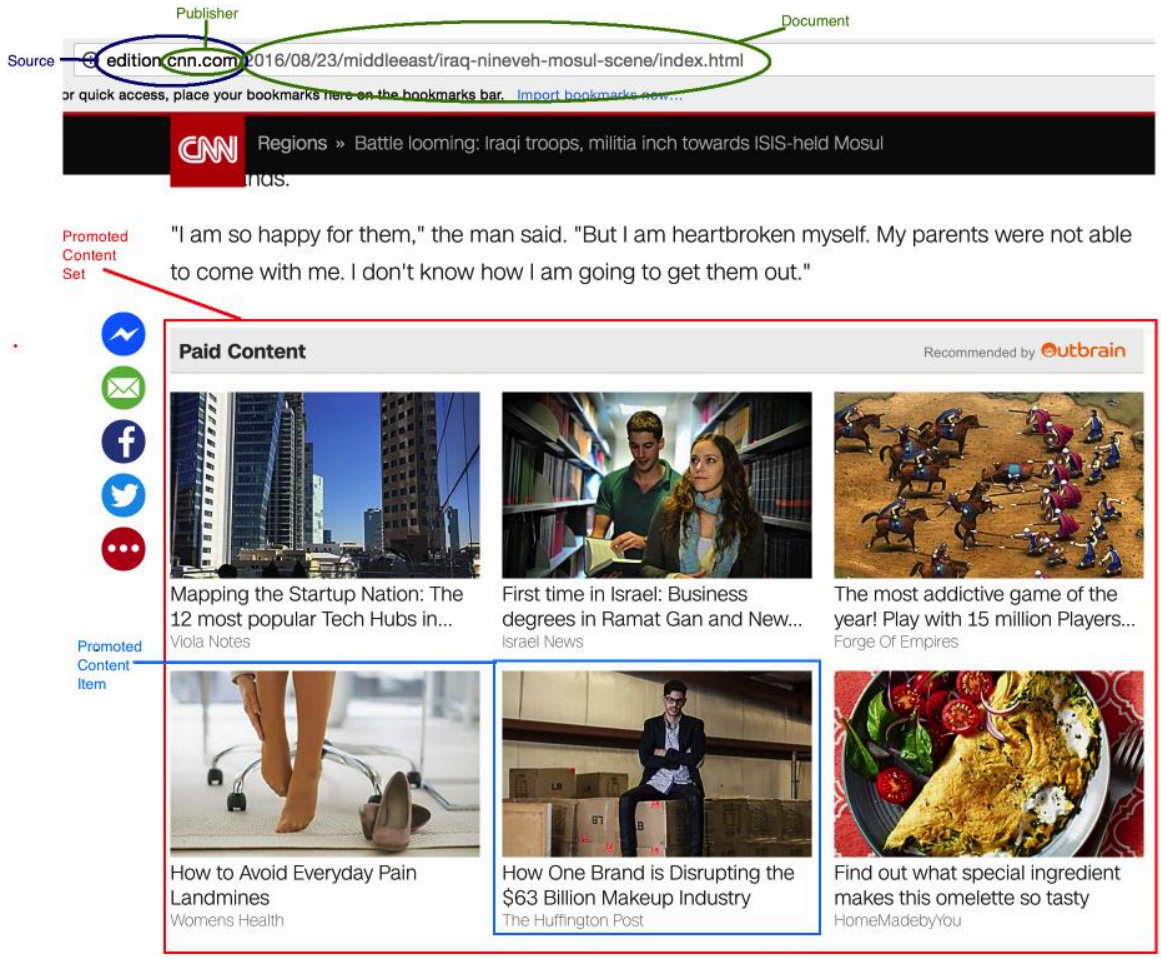


Figure 1Example of an Outbrain recommendation

Evaluation metric

Submission were evaluated using the Mean Average Precision @12(MAP@12):

$$MAP@12 = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{k=1}^{min(12,n)} P(k)$$

where |U| is the number of display_ids, P(k) is the precision at cutoff k and n is the number of predicted ad_ids.

From what I learnt, as the name suggest MAP is the mean precision calculated over the average precision (AP) of each query. The AP at k of each query in turn is a percentage of correct items among first k recommendations. Since here the maximum number of ads is twelve, k = 12.

So for e.g. if MAP@12 was 0.5 across 3 queries then it would mean that for each query on an average every second ad of the 2 to 12 ads was correctly classified.

The Kaggle leaderboard for this competition had a maximum MAP@12 of 0.70145.

Structure of data

Following is an Entity Relationship Diagram to understand how the problem data provided was organized,

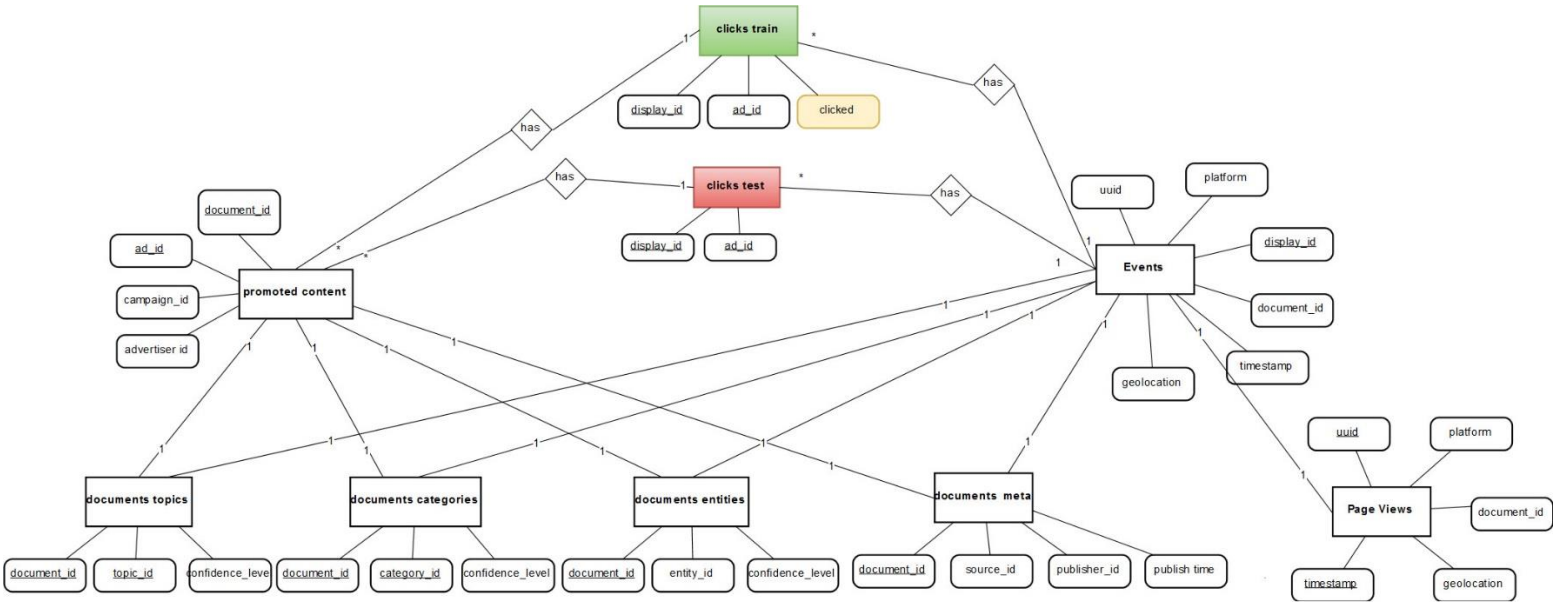


Figure 2 Data structure

- Clicks_train was the training data for which the dependent variable – 'clicked' was provided.
- Clicks_test was the test data for which the clicked probability needed to be predicted.
- There was meta data associated with both the source document on which the promoted content is shown and the target document to which the promoted content points.
- The meta data included the possible topics, categories and entities the document may belong to. One document could belong to more than one topic, more than one category and more than one entity. The relationship was quantified using a measure called 'confidence_level' which ranged from 0 to 1 depending on how confident Outbrain was in classifying the document with that topic, category or entity.
- User attributes such as platform (pc, smart phone or tablet), location (region and country), UUID and time of access was also provided.

Data Exploration

Data Cleansing

Not much of data cleansing was needed. Simple cleansing such as making the user platform id coherent (1, 2 or 3) and dropping 5 entries due to missing platform id was performed on the events dataframe.

Following are some of the descriptive statistics I observed in the data,

1. Number of ads

Number of ads shown per display in the training and test data,

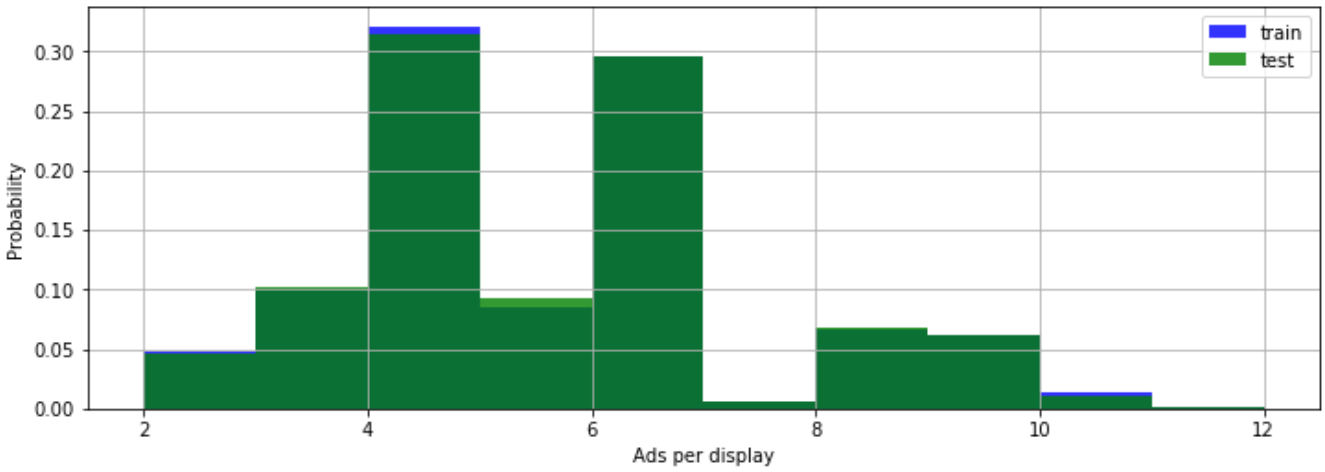


Figure 3 Norm Distribution of Ads per display

- Between 2 to 12 ads were shown each time
- Displays with 4 or 6 ads were at least thrice more likely than displays of any other number of ads

2. CTR

Click through rate (CTR) is a common measure of performance in such advertisement related problems. Here I calculated CTR as the ratio of number of times an ad was clicked to the number of times if was shown. The distribution of CTR that was observed for the training data is as below,

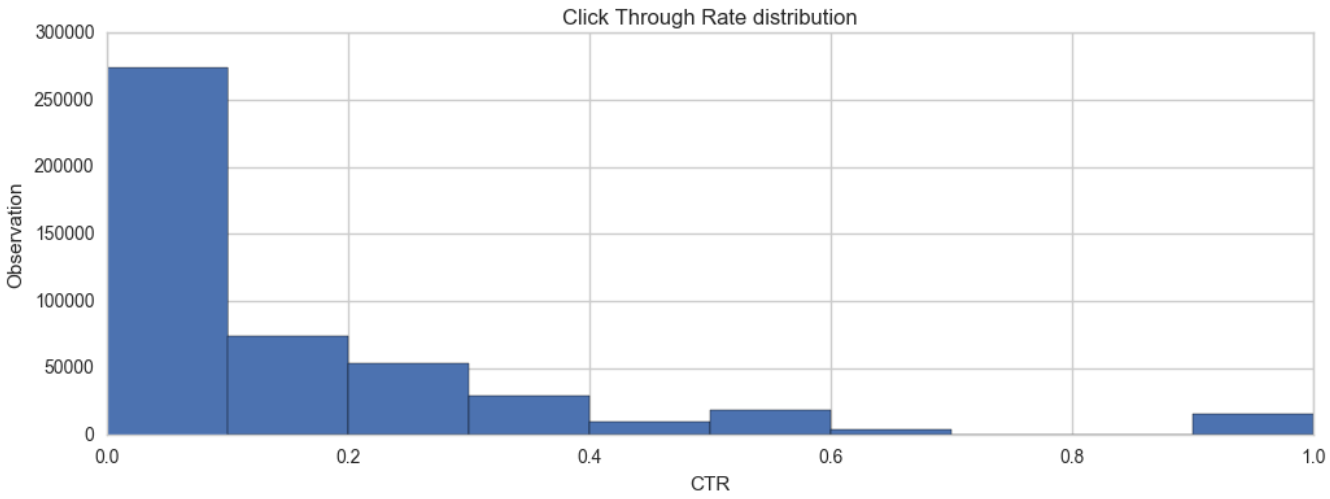


Figure 4 CTR Histogram

- The CTR distribution was right skewed.
- It was calculated to be around 14% on average.
- The low CTR can be attributed to the fact that for each display only one of the x number of ads displayed were clicked.
- 53% ads were clicked at least once or in other words 47% of ads were never clicked. Hence any ad had an approximately 50-50 chance of being clicked on average.
- Ad repetitions:
    - Each Ad was repeated on an average 182 times
    - The range of ad repetition was from 1 to 211824 (inclusive).
    - The median number of times an ad was repeated was 5.
    - The large disparity in the number of times an ad was repeated indicated that the ad selection logic may be skewed one way or the other.
- The correlation between the number of times an ad was repeated to its CTR was negligible (0.02) indicating that ad repetition didn't affect its probability of being clicked one way or the other. The correlation for ads that were clicked at least once and their CTR was negative but negligible as well (-0.04).

3. Clicks distribution

The training data referred to all the ads that were clicked and also when they were clicked in the 15 day period.

Following is the distribution of the displays (groups of ads) for the 15 day period.

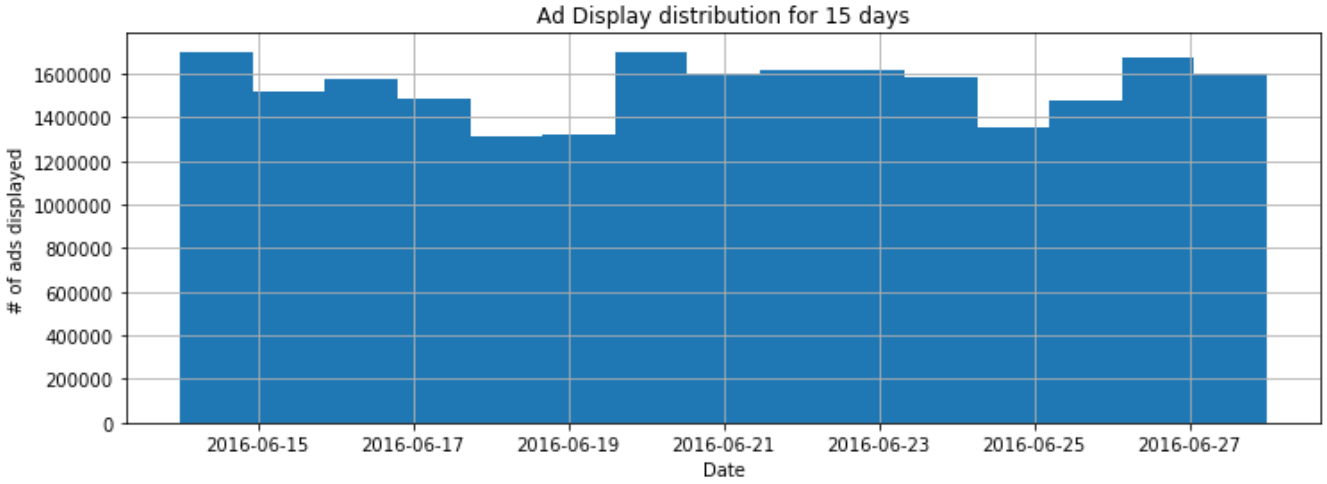


Figure 5 Ad distribution in 15 days

- This shows that the number of ads clicked daily remained fairly constant throughout the 15-day period

And following is the aggregated distribution of the displays throughout the day,

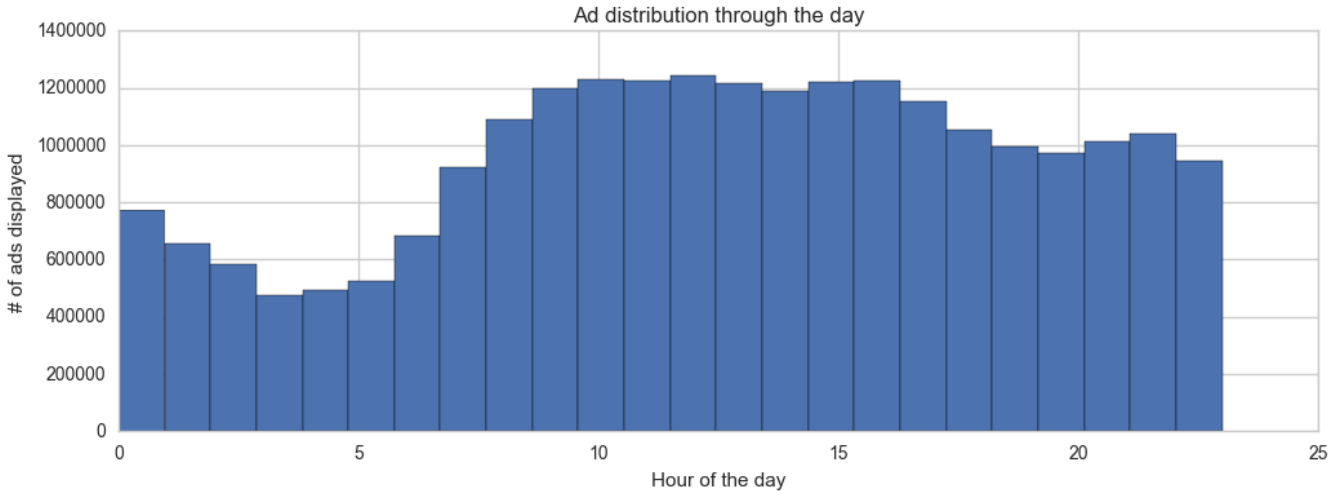
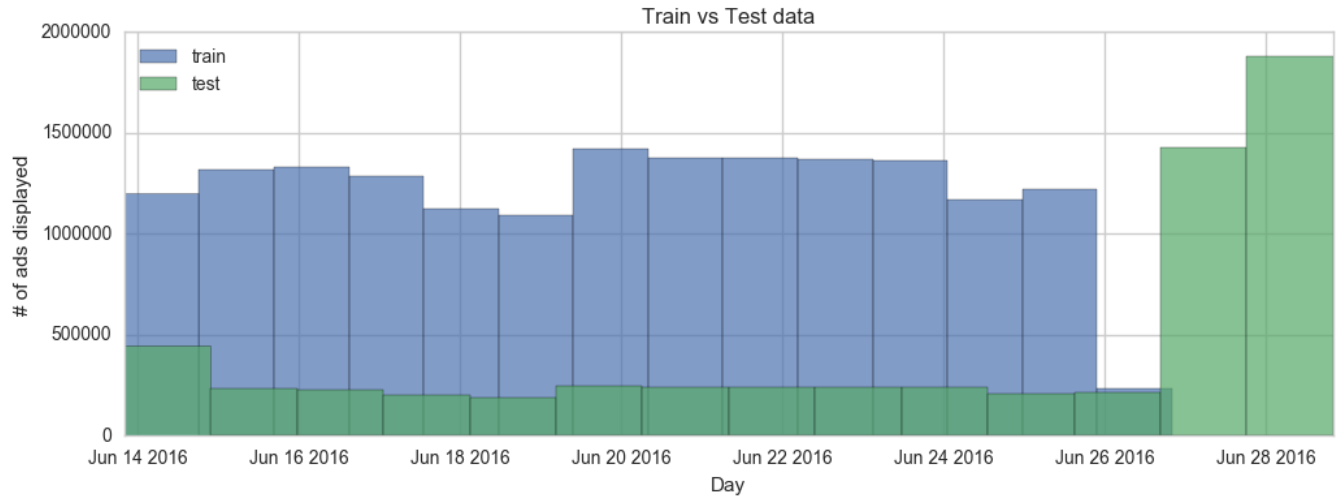


Figure 6 Ad distribution during the day

- It appears that activity picked up after 8:00am in the morning till around 5:00pm in the evening which are the regular business hours and then around 6:00 pm to 11:00pm in the night.

  Following is the how the test data compares with the training data distribution through the 15-day period,



- The prediction needed to be done for the complete 15-day period however for the last two days – 27th and 28th June there was no training data available.

4. Source of user traffic

Looking at where the user traffic originated from, the distribution of the users by country was as below,

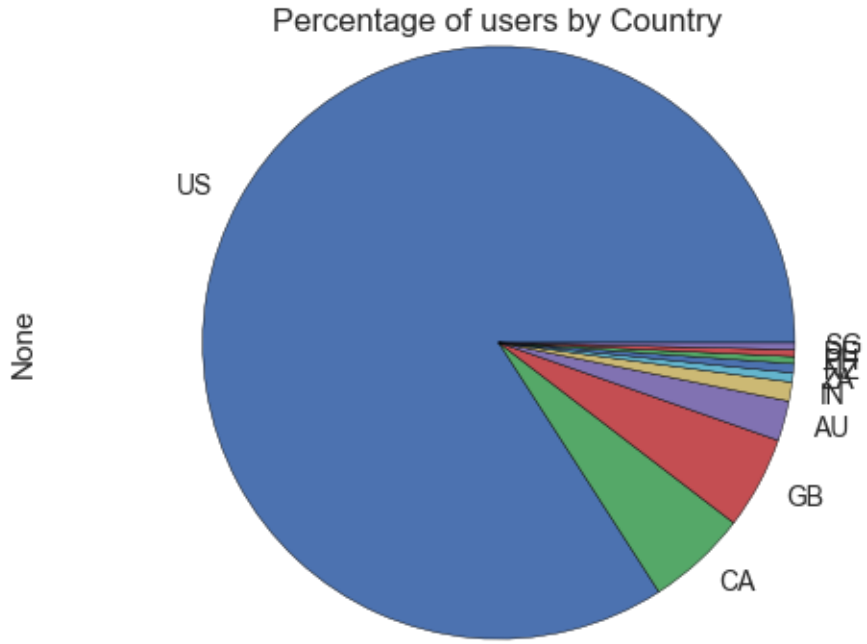


Figure 7 Percentage of users by country

Within the US, the user traffic distribution was as below,

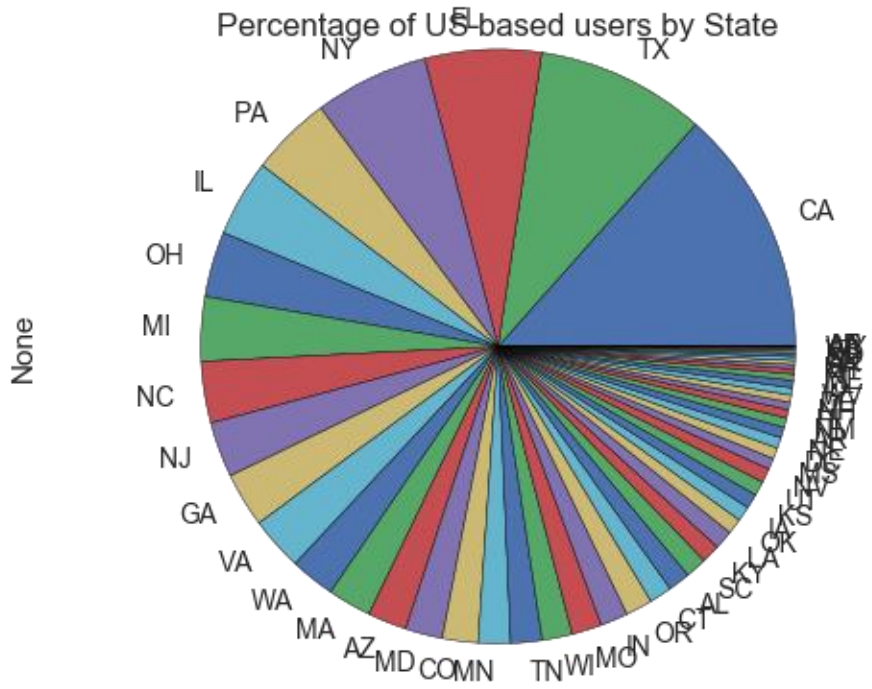


Figure 8 Percentage of user traffic by state

- Hence it was evident that most of the user traffic originated in the US.
- And then within the US, user requests mainly came from California.

The distribution below is the distribution of ads during the day only for the US such that all times have been normalized to EST such that 8:00 AM in NY is also 8:00 AM in Atlanta and 8:00 AM in California by subtracting 2 hours and 3 hours etc. respectively.

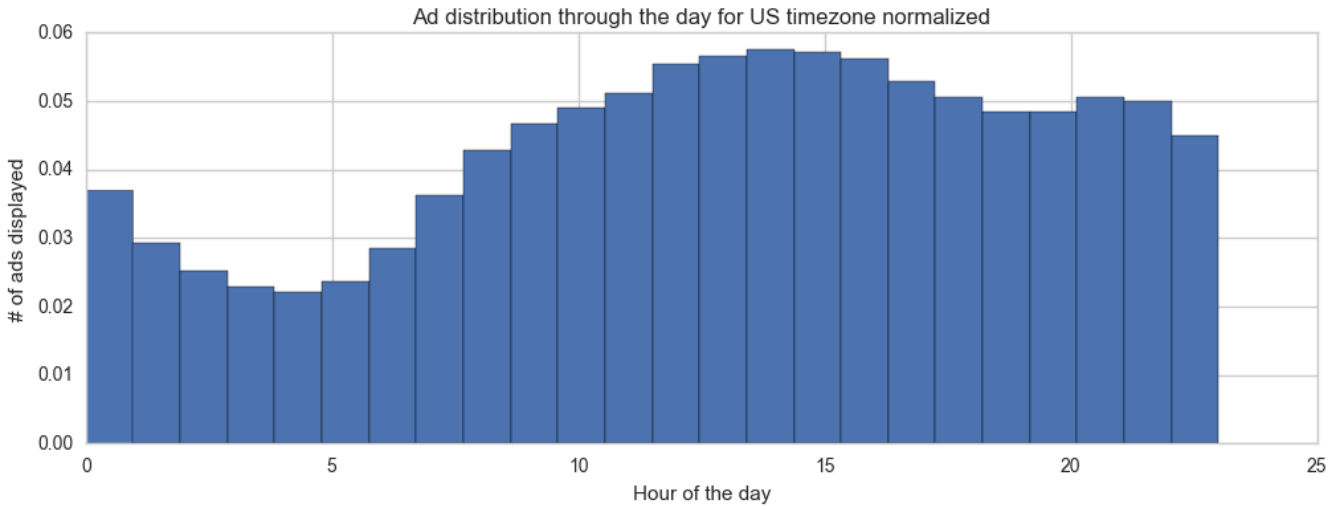
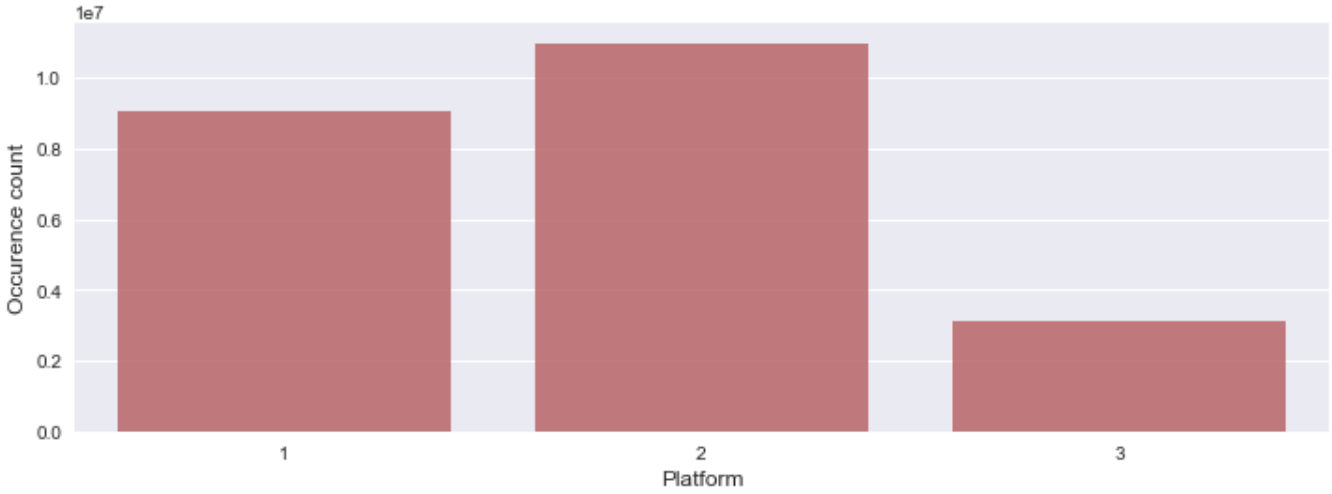


Figure 9 Ad distribution during the day (US only)

- This just re-iterated the earlier understanding that user traffic was more during normal business hours and during the evenings.
- Furthermore, it was observed that most of the user traffic originated from mobile phones. (1 – PC, 2 – Mobile, 3 – tablet)



5. Topics, Categories and Entities

- There are 300 unique topics, 97 unique categories and 1,326,009 unique entities that the documents belong to.
- However, not all documents have an associated topic or category or entity.

Data Volume

The sheer amount of data for this problem was overwhelming. I kept running into out-of-memory errors during data exploration and model evaluation. Hence I selectively choose only part of the data for further exploration and model building.

```
# File sizes
page_views_sample.csv          454.35MB
documents_meta.csv             89.38MB
documents_categories.csv       118.02MB
events.csv                     1208.55MB
clicks_test.csv                506.95MB
promoted_content.csv           13.89MB
documents_topics.csv           339.47MB
documents_entities.csv         324.1MB
sample_submission.csv          273.14MB
clicks_train.csv               1486.73MB
```

Inferential statistics

Feature selection

1. Topic, Category and Entity

Topic, Category and Entity provided meta information about both the document on which the ad was shown (source document) and about the document to which the ad pointed to (target document). However, as noted earlier there were a large number of Topics, Categories and Entities and since this was all nominal data it had to be converted to dummy variables for each topic, category and entity resulting in a large feature set if all of them were to be considered. Furthermore, each of the topics, categories and entities themselves were not associated with a lot of documents leading to this feature set being sparse as seen in observations below,

```
Number of topics that appear more than 1% times: 19
Number of topics that appear more than 5% times: 1
Number of topics that appear more than 10% times: 0
Number of topics that appear more than 20% times: 0
Number of topics that appear more than 50% times: 0
Number of topics that appear more than 100% times: 0


Number of categories that appear more than 1% times: 28
Number of categories that appear more than 5% times: 4
Number of categories that appear more than 10% times: 1
Number of categories that appear more than 20% times: 0
Number of categories that appear more than 50% times: 0
Number of categories that appear more than 100% times: 0


Number of categories that appear more than 0.5% times: 6
Number of categories that appear more than 1% times: 3
Number of categories that appear more than 5% times: 1
Number of categories that appear more than 10% times: 0
Number of categories that appear more than 20% times: 0
Number of categories that appear more than 50% times: 0
Number of categories that appear more than 100% times: 0
```

Hence for simplification,

- I considered a document belonging to a single topic, category and entity for it had the maximum confidence level and dropping the other document topic, category and entity associations.
- I considered only the top five most frequent category, topic and entity for model building.

To further simply and reduce the number of features and more importantly to reduce multicollinearity I performed a Chi-Squared Test of Independence between the Topics, Categories and the Entities. My rationale was that documents of the same categories may have the same topics and the same entities e.g. documents with category sports may have topics such as football, baseball and entities such as players, score etc.

So the null hypothesis' in this case were:

i. There is no association between topics and categories

ii. There is no association between categories and entities.

And the alternate hypothesis' were:

i. Topics and Categories are associated (related) with each other
ii. Categories and Entities are associated (related) with each other.

The P-value for both the chi-square test, topics and categories and categories and entities came out to be 0 disproving the null hypothesis and indicating that there is indeed a relationship between the topics and the categories and categories and the entities suggesting that all three were related. Hence I just chose only categories as features.

2. Display size

The likelihood of any ad being clicked would depend on the total number of ads it is shown with in a display. For e.g. if an ad is shown in a display along with one other ad it is more likely to be clicked than if it is shown with five other ads. Hence I chose the size of the display as a feature for the model.

3. Regularized CTR

CTR represents popularity of the ad so it would serve as an excellent feature in determining future probability of an being clicked. However, CTR for ads that were shown less number of times may be high as compared to those that were shown more number of times. For e.g., an ad that was shown four times and was clicked three times will have a CTR of 0.7 while an ad that was shown a thousand times and clicked six hundred times would have a CTR of 0.6 but one may argue that the second ad had a better track record since it was seen more times. Hence a regularization was needed to give an advantage to the ads that were shown more times.

The regularized CTR was calculated as,

reg_ctr = number of times an ad was clicked / (number of times an ad was shown + regularization factor)

I arbitrarily set the regularization factor to 10.

Classifier selection

I chose Random Forest as the classifier because,

1. It is based on Decision Tree and easy to comprehend.
2. It would provide me feature selection for free.
3. It is an ensemble method which creates several decision trees and then combines their prediction. Hence I thought it would not overfit and generalize well.
4. It has inbuilt support for cross validation.
5. Not all features had a linear relationship with the click probability hence logistic regression may not have been a good fit.

Although I tried to choose all the different hyper-parameters of Random forest such as min_sample_size, number of estimators etc. using Grid Search I had no luck in getting the optimal values of all of the parameters using Grid Search since it would run forever even on a 16 core Azure VM. Hence I had to mostly choose parameters such that the classifier ran in modest amount of time.

The final parameters I passed to Random forest were,

- **Criterion**: gini (derived via grid searh)
- **n_estimators:** 3
- **max_features**: n_features (all features)
- **n_jobs**: 4 (utilizing 4 cores)
- **min_samples_leaf**: 0.10 (I added pruning only due to hardware limitations. Without pruning the depth of the tree would be large and the classifier wouldn't finish running)

Conclusion

- I created the model using the Random Forest classifier and fitted it on the training data with features: display_size, reg_ctr, platform, source document and target document values for dummy variables of the top five most frequent categories. In all thirteen features.
- I got a MAPK score of **0.61415** by using the model over the test data.
- Strangely though Random Forest reported the reg_ctr as the only feature with importance value as 1 and rest all features had values 0.
- The high importance of reg_ctr indicates that CTR value is one of the most important feature in ad prediction.
- The sparsity of categories, topics and entities may not qualify them as good features.
- This was my first attempt at a data science problem and I realized that finding good features is as difficult as tuning the model.
- I learnt how to approach a data science problem, how to explore data and make sense out of it and how to tune a model.
- Finally, I also learnt that how much fun, interesting and rewarding data science can be demanding a lot of perseverance at the same time.

References

http://hamelg.blogspot.com/2015/11/python-for-data-analysis-part-25-chi.html

https://www.kaggle.com/c/outbrain-click-prediction/discussion

https://makarandtapaswi.wordpress.com/2012/07/02/intuition-behind-average-precision-and-map/

http://fastml.com/what-you-wanted-to-know-about-mean-average-precision/

https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/

Appendix A: Detail data description from Kaggle

Source: https://www.kaggle.com/c/outbrain-click-prediction/data

Data Introduction

The dataset for this challenge contains a sample of users' page views and clicks, as observed on multiple publisher sites in the United States between 14-June-2016 and 28-June-2016. Each viewed page or clicked recommendation is further accompanied by some semantic attributes of those documents. For full details, see data specifications below.

The dataset contains numerous sets of content recommendations served to a specific user in a specific context. Each context (i.e. a set of recommendations) is given a display_id. In each such set, the user has clicked on at least one recommendation. The identities of the clicked recommendations in the test set are not revealed. Your task is to rank the recommendations in each group by decreasing predicted likelihood of being clicked.

As a warning, this is a very large relational dataset. While most of the tables are small enough to fit in memory, the page views log (page_views.csv) is over 2 billion rows and 100GB uncompressed. We have also uploaded a sample version of this file with the first 10,000,000 rows. The MD5 checksum of page_views.csv.zip is 3742c116bab4030e0a7ea1c0be623bd9.

Data Fields

Each user in the dataset is represented by a unique id (uuid). A person can view a document (document_id), which is simply a web page with content (e.g. a news article). On each document, a set of ads (ad_id) are displayed. Each ad belongs to a campaign (campaign_id) run by an advertiser (advertiser_id). You are also provided metadata about the document, such as which entities are mentioned, a taxonomy of categories, the topics mentioned, and the publisher.

File Descriptions

page_views.csv is a the log of users visiting documents. To save disk space, the timestamps in the entire dataset are relative to the first time in the dataset. If you wish to recover the actual epoch time of the visit, add 1465876799998 to the timestamp.

- uuid
- document_id
- timestamp (ms since 1970-01-01 - 1465876799998)
- platform (desktop = 1, mobile = 2, tablet =3)
- geo_location (country>state>DMA)
- traffic_source (internal = 1, search = 2, social = 3)

clicks_train.csv is the training set, showing which of a set of ads was clicked.

- display_id
- ad_id
- clicked (1 if clicked, 0 otherwise)

clicks_test.csv is the same as clicks_train.csv, except it does not have the clicked ad. This is the file you should use to predict. Each display_id has only one clicked ad. Note that test set contains display_ids from the entire dataset timeframe. Additionally, the public/private sampling for the competition is uniformly random, not based on time. These sampling choices were intentional, in spite of the possibility that participants can look ahead in time.

sample_submission.csv shows the correct submission format.

events.csv provides information on the display_id context. It covers both the train and test set.

- display_id
- uuid
- document_id
- timestamp
- platform
- geo_location

promoted_content.csv provides details on the ads.

- ad_id
- document_id
- campaign_id
- advertiser_id

documents_meta.csv provides details on the documents.

- document_id
- source_id (the part of the site on which the document is displayed, e.g. edition.cnn.com)
- publisher_id
- publish_time

documents_topics.csv, documents_entities.csv, and documents_categories.csv all provide information about the content in a document, as well as Outbrain's confidence in each respective relationship. For example, an entity_id can represent a person, organization, or location. The rows in documents_entities.csv give the confidence that the given entity was referred to in the document.