ML Hackathon

E-commerce Advertisement Analysis

November 24, 2019

Crooked Three

Sriram G. (IMT2017018)

Ravi Kiran (IMT2017034)

S. Purvaj (IMT2017039)

Contents

1 Introduction

1.1 About the problem statement

In today's scenario e-commerce companies play an integral part in the economy and also there is a lot of analysis required for the companies to asses themselves an The problem is to predict whether an advertisement on a webpage has been clicked by a user or not based on system data, app/webpage data and advertisement data. This is generally used by large e commerce companies to track their sales and also see which websites are more influential and try to increase their sales by choosing partner websites to advertise customized products for the users.

1.2 References

- Analytics Vidhay : WNS wizard 2019 hackathon
- Medium
- Stackoverflow

2 Data

Drive Link for Data :
https://drive.google.com/open?id=1wxn1bqwKrXXeEeYc47kjkuQKauzppp3f

2.1 Exploration of data

The data for the problem is across three seperate csv files. One has information about the app, operating system, network used and time of the impression(time at which the data point was recorded) along with the prediction label. Lets call this file/table *impression_-info*. Second file has information about the user and device used by the user. Lets call this file/table *user_info*. The third file contains information on the item advertised and the its price. Lets call this file/table *item_info*.

The impression info consists of the data about the impression i.e. the advertisement in the partner website. The system log info consists of the information about the user's interaction with the actual website itself. Item data consists of the information item that the user has seen when he last logged into the actual website.

2.2 Data Analysis

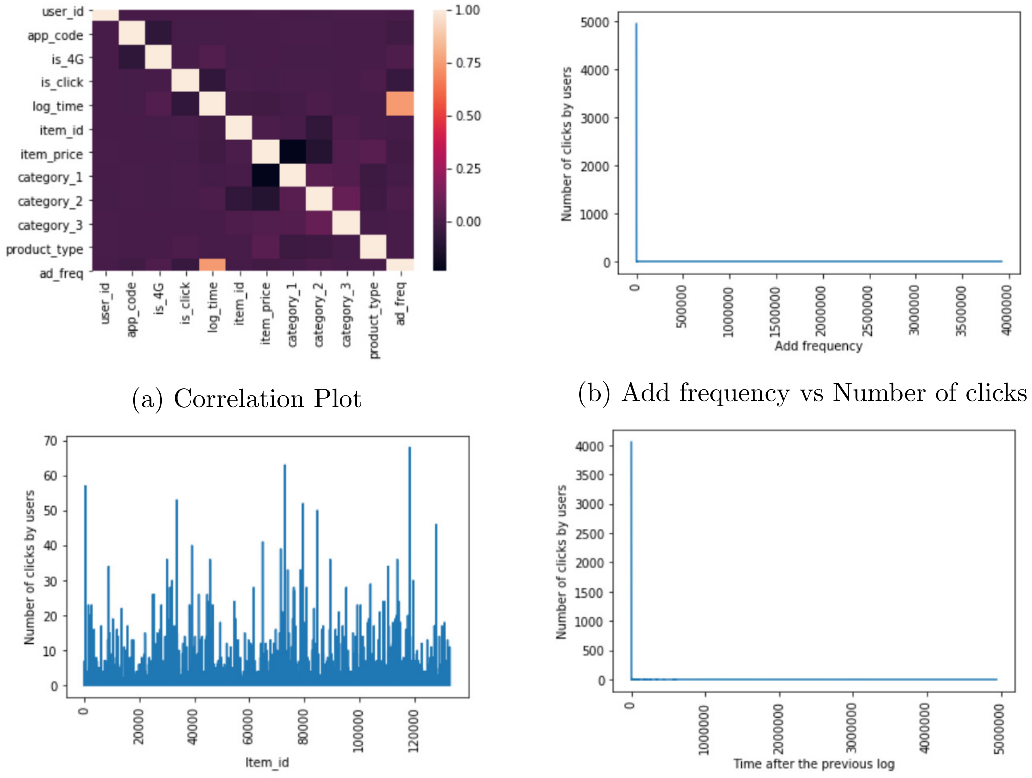
Firstly, there are no null values across all the three data files. Secondly, the three data files cannot be combined directly as they are of different sizes. The file which contains the prediction labels is governs how much training data is available which in our case is the *impression_info* file. We must join this file in an appropriate way so that we can extract suitable features from the whole dataset. Now, we have the complete dataset with about *** data points.

Another important observaion about the given data is that it is highly biased. Only 5% of the given data is positive while the remaining 95% data is negative.

Some of the plots that we used to understand the data better have been shown in the following page:

2.3 Data Pre-processing and Feature Engineering

The impression ids for a specific user are only after he logs into the system atleast ones. Therefore each impression that a user encounters between two successive logs is related only to the item that the user has seen in the previous log. We use this technique to join the data files in a sensible manner.



(a) Correlation Plot

(b) Add frequency vs Number of clicks

(c) Item id vs Number of clicks

(d) Item id vs Number of clicks

Figure 2.1: Data Analysis

2.3.1 Time duration between log time and impression time:

This feature is calculated as the time difference between the most recent log time(the most recent time instant at which the user was logged onto the app/website) and the impression time(the time instant at which the advertisement appears to the user) in seconds.

After looking at the data we found out that the data can be understood as below:

(T_1) **log entry of user 1**

(T_2) ++++++++++++++++++++++ impid_1 user_1

(T_3) ++++++++++++++++++++++ impid_2 user_1

(T_4) ++++++++++++++++++++++ impid_3 user_1

(T_5) ++++++++++++++++++++++ impid_4 user_1

(T_6) **log entry of user 2**

(T_7) ++++++++++++++++++++++ impid_5 user_2

(T_8) ++++++++++++++++++++++ impid_6 user_2

(T_9) ++++++++++++++++++++++ impid_7 user_1

(T_{10}) ++++++++++++++++++++++ impid_8 user_1

(T_{11}) **log entry of user 1**

(T_{12})	++++++++++++++++++++	impid_9 user_1
(T_{13})	++++++++++++++++++++	impid_10 user_1
(T_{14})	++++++++++++++++++++	impid_11 user_2

In the above example the user 1 logs at an instant time_1 and then there are many impression ids. Now each impression id for the user_1 between T(1) and T(10) will only be aboutthe item that he searched for, when he entered for the first time. We create a new column which contains the difference between the time at which the user logs into the system and the time at which the user encounters the impression id.

$(T_i)-(T_j)$ where $i > j$ and i $\epsilon\{1,6,11\}$ and j $\epsilon\{2,3,4,5,7,8,9,10,12,23,14\}$ depending on the user id.

2.3.2 Item_id from recent log

The second feature that was extracted was the details of the item that the user checked in the last log. This might useful for the situation when the user's probability of buying a specific item is based on his/her's previous history of buying products. Hence the user_id is not removed from the training data and is also used as a feature.

2.3.3 Add frequency

The third feature that was extracted was the details of the frequency at which the user was seeing the ads of that particular website.Now we finally end up 11 features for training the models.

2.3.4 Pre processing

Initial step was to normalize the data and to convert the categorical attributes into integers. Then we used PCA to reduce the dimensionality of the data.We tried using multiple values for components of the pca reducability. We have seen that the data is higly biased. Therefore we have used some techniques to generalize the model. We split the negative data(is_click=0) into 20 groups and trained these data groups along with the positive data in different models and used voting classifier for training.

For testing and prediction, we considered the outputs of the various models trained as described above. We considered the class that the majority of the models predicted as our prediction.

3 Model Selection and Building

3.1 Model-1: Logistic Regression

Logistic regression is one of the simple binary classifier that is based on a log loss . W is the weight matrix and x is the data matrix. B is the bias.

$$\text{Hypothesis : Z = WX+ b}$$
$$h(\theta(x)) = sigmoid(Z)$$

where sigmoid is

$$sigmoid(x) = \frac{1}{1+e^{-x}}$$

Theory

results:

Considered data of size 22000(estimated)
Accuracy** : 69.49%

3.2 Model-2: XGB Classifier

Theory

Results:

Considered data of size 22000(estimated)
Accuracy** : 98.49%

3.3 Model-3: Random Forest Classifier

Theory

Random Forest uses the idea of Decision Trees. It takes a random subset of features and builds the decision tree and then uses it for prediction.

Results:

Considered data of size 22000(estimated)
Accuracy** : 98.21%

3.4 Model-4: Voting Classifier

Theory

It is an ensemble of the above models and then predicts based on importance.

Results:

Considered data of size 22000(estimated)
Accuracy** : 98.48 %

4 Observations and Results

Model	PCA-6	PCA-7	without PCA
Logistic Regression	73.47%	73.47%	69.49%
Random Forest Classifier	97.45%	97.07%	98.21%
XGB Classifier	97.31%	97.2%	98.49%
Voting Classifier	97.31%	97.17%	98.48%

**Accuracy mentioned in the above is an average over a set of 20 different variables.