ISE 5103 INTELLIGENT DATA ANALYTICS

Click Predictions

By

Team: Power Rangers

(Dheeraj Srivathsav & Vimlesh Bavadiya)

Executive Summary

In online advertisement business, the measure of performance is click-through rate (CTR). It is the ratio of number of clicks on the advertisement to the number of views of the advertisement. Higher the CTR, higher the profitability. The aim of this project is to predict the probabilities of the advertisement being clicked using various classification models. The data used for the project is obtained from the Kaggle competition Outbrain Click Prediction.

Outbrain released 2 billion page views and 16900000 clicks of 700 million unique users across 560 cities. The data has been anonymized. It contains several csv files which contains different attributes. The page views file is 29.7 GB compressed and 100 GB uncompressed. The data is too large to handle. A sample of the page views was provided. Different files were merged together to form one file with necessary attributes. The data from files.csv was not used as the data would provide only partial information to its sample nature and full file data was not feasible to use.

Even after the exclusion of the file data, the events and train data size was enormous and personal laptops were unable to handle the size. Reading of the data file itself took hours. The R package data.table was handy to read the data much faster. When merging the data together in one data frame, the size of memory was an issue. Another computer with better memory was used for the project.

The variables available are quite unique and we have found correlation between the features apply feature engineering. Missing ness is properly handled with few mean imputations and removal of columns at times (clearly explained later).

Problem Description

Outbrain is one of the largest content discovery platform. It is an online advertiser specializing in presenting sponsored website links. It uses behavioral targeting to recommend advertisements and pages as opposed to the related items widget. They recommend the content on major media properties including CNN, People and ESPN. The company pays the publishers to display the recommendations on their site. External company pays outbrain on a daily pay per click basis with links to third party content appearing as recommendations. Hence the profitability of outbrain increases with an increase in clicks or CTR. The dataset for this problem 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. Outbrain has provided data of 2 billion page views and 16.9 million clicks of 700 million unique users across 560 cities around the world. The data has been anonymized and distributed into several csv files. The task of the competition was to provide a submission file with ad id displayed in decreasing order of click probability for each display id.

For predicting the probability of the ad getting clicked models like logistic regression, random forest and boosted trees have been selected as they are already know and easy to apply. Based on the literature there have been several other models applied for predictive modelling on such data like Naïve Bayes, Support Vector Machines (SVM) and Factorization Machines (FM).

The huge dataset and limited computational power has been quite a challenge for the project. The data have been scattered and combining the data was the initial task. The default read function in R took hours just to read the complete csv files. Data.table package was used to increase the reading speed. Personal laptops with 8 GB RAM were initially used for merging files. But they were not enough for multiple merges when the file size exceeded 4 GB. Another laptop was then used with RAM of 32 GB to bypass the memory limitation. Merging of the files was successful with final size of the csv file near to 8 GB. The merging of the files was done as per following diagram.

Dataset and features:

The Outbrain dataset provided a total of eight datasets:

Page views describes features of all viewed pages, regardless of an advertisement being clicked.

Events consists of features of pages viewed when one displayed advertisement was clicked.

Promoted content provides advertisement features.

Clicks train/test provides examples with labels to be used for training and examples without labels to be used for the Outbrain competition.

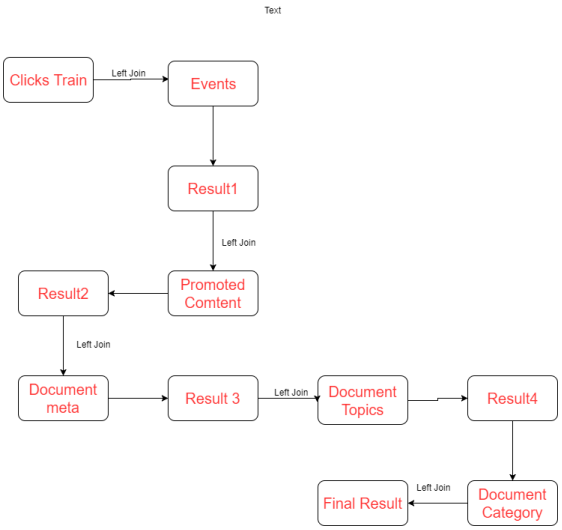
Documents meta describes documents' metadata.

Documents entities, documents topic, and documents categories provide mentioned entities (person, place, or location), topic, and taxonomy of categories of the documents, respectively.

TABLE I. Features provided by Outbrain (n = 19)

Users	uuid, geographic location
Documents	Document id, ad id, source id, publisher id, publish time, entity id & confidence level, topic id & confidence level, category id & confidence level, advertiser id, campaign id
Page View Event	display id, timestamp, platform, traffic source

Merging Data:



The **Train dataset and the Events dataset was left joined with the display_id**. This **Result was again left joined with the Promoted content** dataset with the **ad_id** column. After merging this the result two columns named document.x and document.y was renamed as document_source

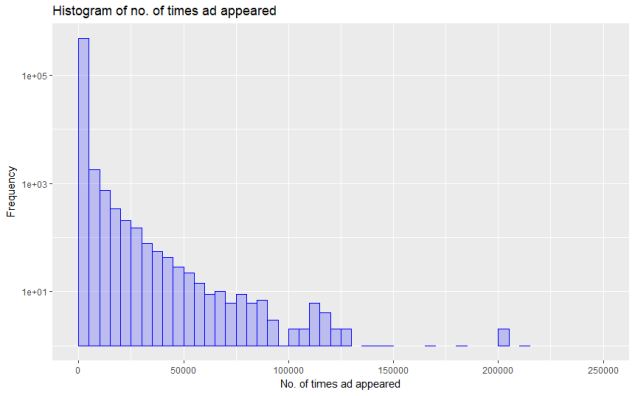
(The actual document in which the advertisement is present.) and document_landing (This the document to which it is directed once the advertisement is clicked). Now, this result is merged with document_meta dataset based on (document_id &document_source), (document_id&document_landing). Now, we get the corresponding columns from document_meta like publisher_id and source_id for both document _source and document_landing.In the similar manner this result is again left joined with the document_topics and document_category. Now, the final table consists of 22 attributes with 80 million records.

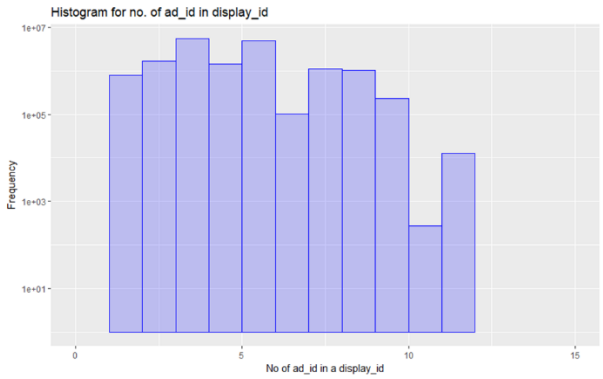
Attributes of Final Data set :

```
> names(sample.set)
 [1] "document_landid"           "document_sourceid"          "ad_id"                      "clicked"                   "platform"
 [6] "campaign_id"               "advertiser_id"              "source_sourceid"            "publisher_sourceid"        "source_landid"
[11] "publisher_landid"          "category_sourceid"          "confidence_sourcelevel"     "category_landid"           "confidence_landlevel"
[16] "topic_sourceid"            "confidencetopic_sourcelevel" "topic_landid"              "confidencetopic_landlevel" "country"
[21] "state"                     "dma"
```

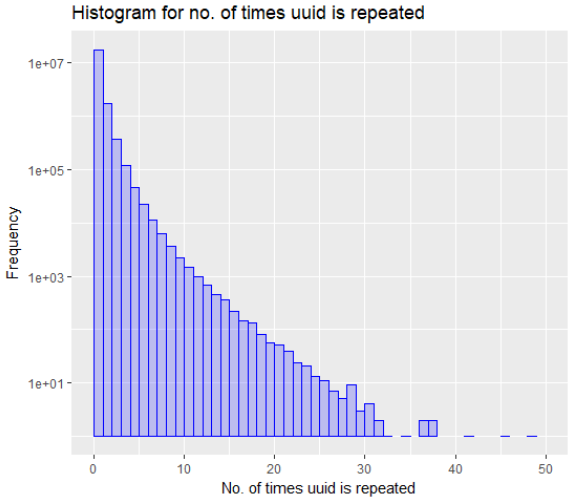
Exploratory Data Analysis :

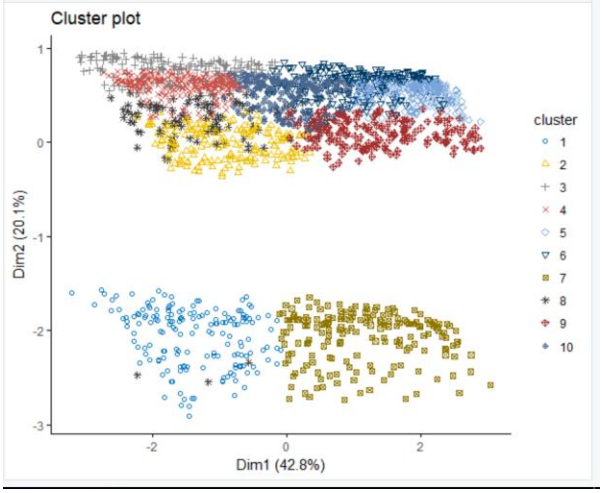
We spend a significant portion of our time on the project understanding and exploring the provided data. As we have the data in multiple csv files firstly we started to explore the csv files individually to understand the data before they were merge. The spread for each of the variables was investigated in the train and events dataset. The occurrence of adid in the train dataset can be seen in Figure A1. **86.93 % of the ads appear less than 50 times.** There are **several ads which have been repeated over 200000 times in the train data**. This can be considered an outlier. But if we assume that the proportional frequency of those ads in the test data is going to be the same then we need to have those outliers in the train data to model the occurrence of the ad with specific document and the probability that the ad will be clicked. The number of ad ids in a displayed can be seen from Figure A2 **which has a spread from 2 to 12 with maximum displays having 3 ads in it.**





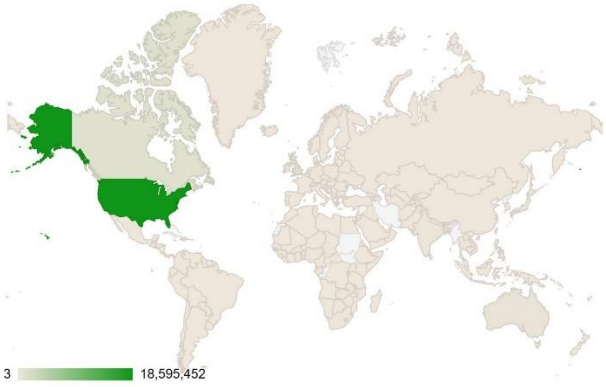
In the events data, the timestamp is given in milliseconds since 1970-01-01. The column platform consists of numeric data indicators of desktop=1, mobile=2 and tablet=3. It was observed that most of the sites were visited on desktop and mobile. From the **histogram of uuid it can be seen that 99.96 % of the users appear less than 10 times in the data.**

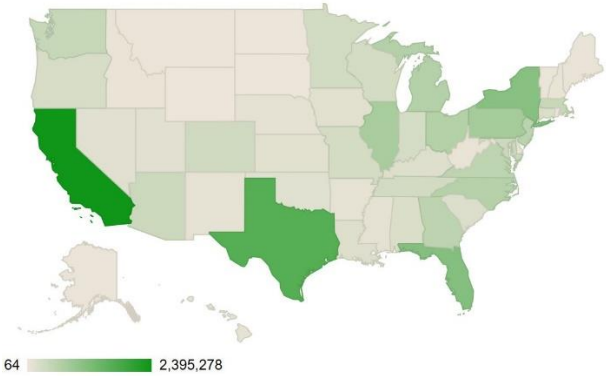




Clustering was one of the other methods used for exploration of the data. It can not only be used for data exploration but even for classification based on the clusters. 5000 and 20000 random samples of data were used. Gap, elbow and Silhouette statistical methods were used for finding out the optimum number of clusters. As can be seen from Figures in Appendix, different methods gave different clusters based on data used for clustering. From the figure above it can be seen that there are primarily 3 clusters but when k means was applied to the data with number of clusters as 10, it separated data into 10 clusters. Implying that k means is not a reliable measure for clustering.

No of unique topic in document categories is 300. The data was explored based on geographic location and 99 % of the data was represented by top 9 countries. The distribution can be seen in the figure given below but since most data is from USA the gradient color is not visible clearly for other countries except CANADA, the second highest. The gradient can be more clearly seen in the map of US. California and Texas are the biggest contributors of data.

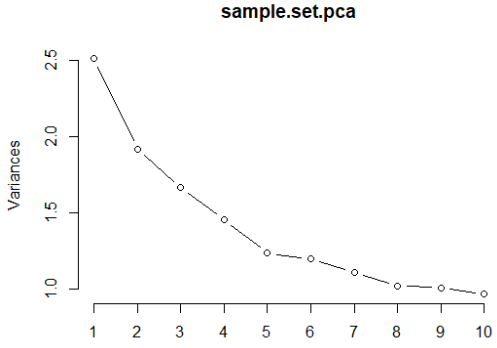


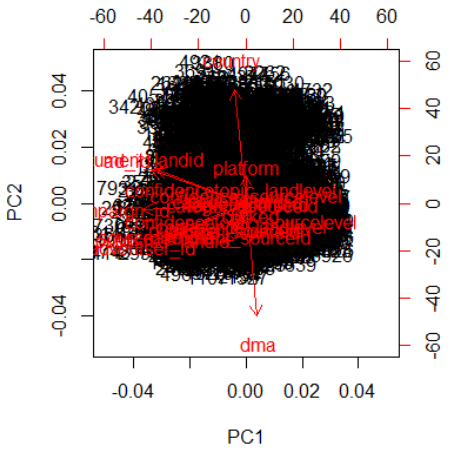


Missingness:

Missingness was explored in individual data and no missingness was found. Missingness has been introduced in the dataset duing merging of different files. The missingness is not completely at random. The missing values are missing attributes of the documents. Since attributes of documents are important feature for training the model, those rows have been removed. Also mean imputation is performed on the dma column and the missing values are replaced.

PCA:

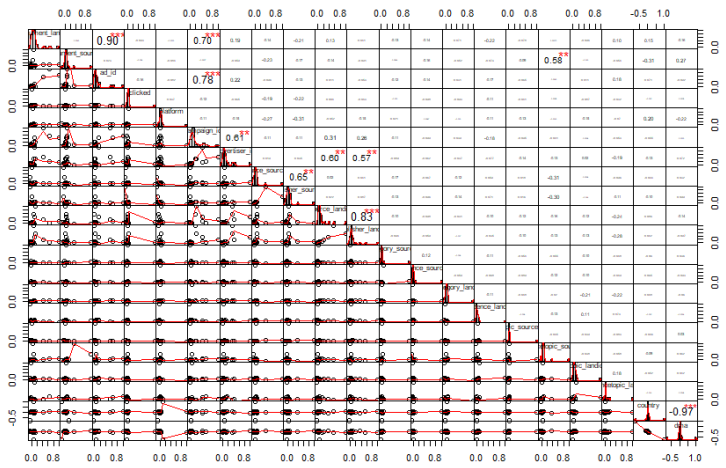




To further explore the data, a principal component analysis is performed. The first plot shows a line plot of 10 principal components and their associated variance that the principal components explain. We can see that there is a clear elbow at 5 which indicates that first five principal components explain a large portion of variance. The second plot shows the variance explained by the principal component 1 and 2. There is a lot of overlap between the data points and it is complex to visualize the data and hence can't be interpreted properly. But we can see that country, platform are in one direction and dma is in the opposite direction it mean that one helps for the user to click the advertisement and the other opposes the user to click the advertisement.

Correlation:

From the plot below we can see that the document_landing and the ad_id are highly correlated (0.90) which means that the information the advertisement is providing is almost the same with the information document is providing after the advertisement is being clicked. Apart from this country and dma are also highly correlated. we removed the highly correlated columns and ran the models which gave us better results (Table-4 in Results and validation)



Training and Test Dataset's:

As our data set was so huge our machines were not able to allocate the required vector memory space (even after using machine with 32 GB RAM) and hence we were not run the models with this data. So, we have sampled the data randomly into 10,000 records. This sampled data was then split into two sets: training set and testing set. The data was split in such a way that the clicked advertisements (which are in small proportion) are not neglected. The training data and testing data was created with 60% of non-clicked advertisement and 40% of clicked advertisement. The final dataset has 10,000 records and over 5000 attributes (after creating dummies). This data was used to run the models

Analysis Plan:

Feature Selection:

In our dataset most of the features are categorical and have high cardinality i.e same factor levels is repeated no.of times. This can be seen in the histograms and plots above. For example, Outbrain has collected the data from 231 different countries but most of the data (nearly 95 %) is from US and Canada. So, overcome the issue of high cardinality, we are grouping all the low frequency factor levels as one group. By doing so the information is barely lost and the no.of features are drastically reduced.For the same country example by reducing the high cardinality 231 countries are reduced to just 3 countries(US, CA, Others) . This method was considered for other features with high cardinality like state (reducing from 400 to 64), dma (212 to 73), publisher_id, source_id for both source and landing. If all values are of relatively equal significance, then we avoided those columns (topic_id, category_id) without loss of information. This was necessary to run the models with limited computer memory.

Once after reducing the no.of factor levels to properly handle the categorical data we have used one-hot encoding method. Consider the same country example with one-hot encoding, this one feature was split into three features: is_US, is_CA, is_Others. Each contained binary values (1 if

the particular platform was used and 0 otherwise). Most of the categorical data was reformatted in this way. And as a result, we have got over 5,000 features ready for model fitting.

Model Selection:

After applying the previously mentioned data transformations to produce a set of numerical features we have chosen three models for the analysis: Logistic Regression, Random forest and gradient boost decision trees. Logistic regression was our first choice of classifier due to its model simplicity. Since our data has binary-valued labels and we want to output the likelihood of an ad id being picked given a document view, logistic regression was the natural approach. Random Forest(RF) is a useful model when the relationship between the target variable and predictor variables is not linear. Unlike logistic regression, it doesn't require a specific data type as input which works in our favor given the copious amounts of categorical features present in the data set. Another advantage to using this particular model, is the small number of hyperparameters required to be tuned. Through our research and help from kernels we have included Gradient boost decision trees(GBDT) with Generalized boosted regression model, black boost and glm boost. Each of these models have advantage and disadvantage but are well-known classification models. In addition, all of the models used in the analysis incorporated some form of re-sampling method such as K-fold cross validation or boosting methods of the training data when building the models. This was to insure the most accurate and reliable results when comparing results.

Each model was built using training data, and then used on test data to calculate the predicted values in the form of decreasing order of probabilities corresponding to each display_id.

Results & Validations:

1.Only Selected Features without one-hot encoding:

Features used: Document_id, ad_id, Publisher_id, category_id, advertiser_id, campaign_id, platform, country, dma

Model	Package	CV method	Parameter	Range	Selected	AUC	Kappa	Accuracy	Logloss	Balanced Accuracy	Sensitivity	Specificity
Logistic Regression	stats	-	-	-	-	0.50625	0.0132	0.702	10.2926	0.5062	0.18	0.8325
Random Forest	randomForest	-	mtry= 50, ntrees= 500	-	-	0.6315344	0.0057	0.791	-	0.5019	0.02	0.9838
Generalized boosted regression Model	gbm	5 fold	ntrees	5000	156	0.622396	0	0.8	0.4846	0.5	0	1
Blackboost	mboost					0.63571	0	0.8	0.488129	0.5	0	1
glmboost	mboost					0.60621	-0.073	0.761	0.532237	0.49758	0.004292	0.99087
glmboost	mboost		mstop	5000		0.6446	0.0059	0.797	0.483	0.5019	0.01	0.9938

To get the basis idea of how models are performing we have selected only important features without applying one hot encoding (creating dummies) and applied to the models.

We are considering AUC as a major deciding factor to select the best model. From the above table **Random Forest and glm boost** (after cross validating) gave better results compared with other models.

2.Applying One Hot encoding and trying with different feature's:

From the analysis of above table glmboost gave us the best results. So, with the same technique (glm boost) and now with the data on which we have applied one hot encoding technique .We have performed on different features to get the better results.

Model	Package	CV method	Parameter	Range	Selected	AUC	Kappa	Accuracy	Logloss	Balanced Accuracy	Sensitivity	Specificity
With confidence level's and topics and categories dataset features												
glmboost						0.6437812	0.119	0.618	0.64135	0.5537	0.2325	0.875
blackboost						0.605454	0	0.6		0.5	0.66157	1
removing confidence levels												
glmboost						0.642875	0.1067	0.615	0.64119	0.5479	0.2125	0.8833
removing confidence levels and topics and categories												
glm boost						0.64299	0.1515	0.628	0.641259	0.5692	0.275	0.8633

From the table applying one hot encoding and considering all the features with glm boost gave us the best result with **AUC: 0.64378**

3.Applying Hyper Parameter Tuning:

With the total dataset applying hyper parameter tuning with different parameters values.

Model	Package	CV method	Parameter	Range	Selected	AUC	Kappa	Accuracy	Logloss	Balanced Accuracy	Sensitivity	Specificity
Random Forest	randomForest	-	mtry= 100, ntrees= 50	-	-	0.649975	0.1759	0.644	-	0.5792	0.255	0.9033
Random Forest	randomForest	-	mtry= 120, ntrees= 5000	-	-	0.6558	0.1475	0.63	-	0.5667	0.25	0.8833
glmboost	mboost		mstop	500		0.5179	0	0.6	0.677	0.5179	0	1

From the analysis of the above table Random Forest with **mtry=120, ntrees=5000** gave us the better result with **AUC: 0.6558**

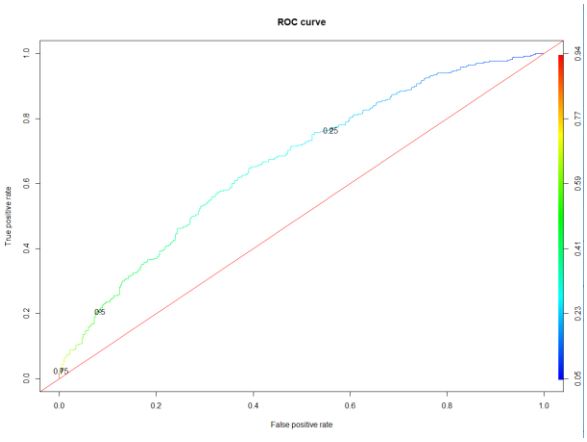
4.Applying Cross validation and Hyper Parameter Tuning:

Model	Package	CV method	Parameter	Range	Selected	AUC	Kappa	Accuracy	Logloss	Balanced Accuracy	Sensitivity	Specificity
Random Forest	randomForest	3 fold 5 repeats	mtry= 120, ntrees= 5000	-	-	0.66475	0.137	0.632	-	0.5608	0.207	0.9167

By cross validating with 3 folds and 5 repeats and mtry=120, ntrees=5000 we got the best **AUC = 0.66475**

Random forest: CV - 5 Folds,3 Repeats mtry 120 ntrees 5000 :

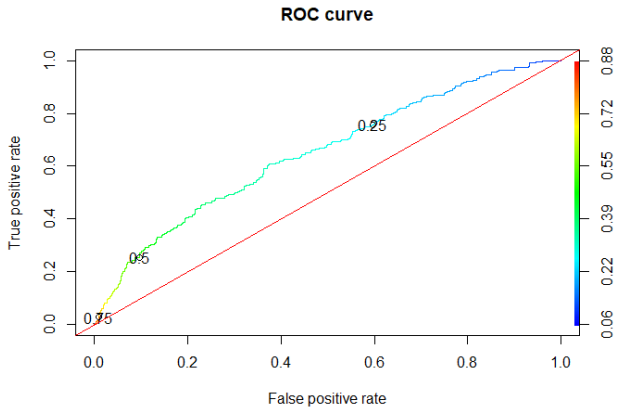
AUC: 0.66475




```
> Confusionmatrix=caret::confusionMatrix(predictedvalue, truevalue, positive="1")
> Confusionmatrix
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 550 318
         1  50  82

               Accuracy : 0.632
                 95% CI : (0.6013, 0.662)
    No Information Rate : 0.6
    P-Value [Acc > NIR] : 0.02064

                  Kappa : 0.137
 Mcnemar's Test P-Value : < 2e-16

            Sensitivity : 0.2050
            Specificity : 0.9167
         Pos Pred Value : 0.6212
         Neg Pred Value : 0.6336
             Prevalence : 0.4000
         Detection Rate : 0.0820
   Detection Prevalence : 0.1320
      Balanced Accuracy : 0.5608

       'Positive' Class : 1

> pred <- prediction(fittedvalue, truevalue)
> auc <- performance(pred, measure = "auc")
> auc1 <- auc@y.values[[1]]
> auc1
[1] 0.6647542
```

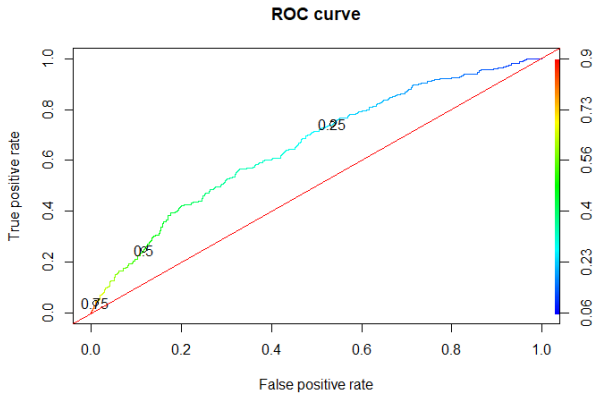
Random forest: mtry 100 ntrees 50 :

AUC : 0.649975

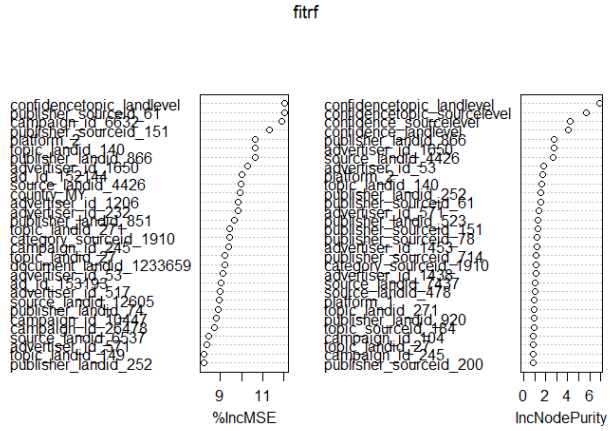


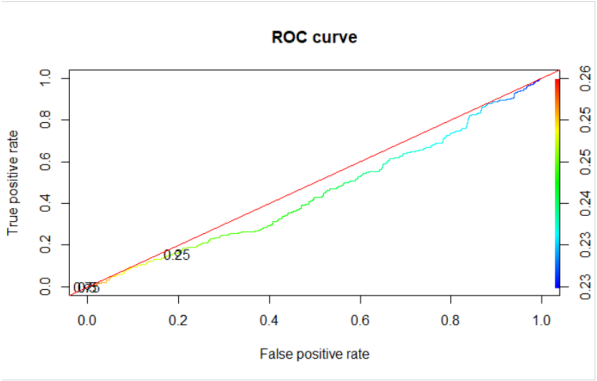
Random forest: mtry 120 ntrees 5000 :

AUC : 0.64558



RF-Variable Importance:





Logistic Regression: AUC-0.4487

Random forest gave us the better results when compared with the other models. It gave us the AUC =0.664558

Conclusion:

Issues:

There were several issues encountered through out this project that had to be overcome.

1.Firstly, each dataset is so huge that with general read.csv it was taking more than 20 min to just load the data. So, we started using fread function in Data. Table package. This was really quick and took less than 2 min to load 80 million data.

2. The major issue was understanding the features and finding the link linking(loops) within the data.For example,if the **advertisement on a particular document is clicked then it opens another document known as document_landing and clicking the advertisement displayed on the document_landing might again link us back to the original document.**Hence,Without proper data understanding we won't be able to know which csv files have to be joined. We have to know which Join to perform and how to resolve the problem of missing ness which was generated using merging.

3.With majority (almost all) of the columns being unique and categorical technique like logistic Regression did not work, as this technique considers document_id : 10 as less important and document : 1,00000 as more important but they both are equal. Due, to which we are getting very bad results. To resolve the above issue we have applied One-hot encoding technique (creating dummies) in which each factor level is converted into a separate column.

4.After Applying this technique we are getting 15000 attributes which is really huge and hence we have reduced the no.of factor levels by considering the factors which constitutes more than 90% of the data. By doing this we have drastically reduced the no.of attributes from 15000 to 3000.

5.After merging all the csv files the final dataset consists of 20 million records with 3000 attributes. Our machines computational capacity was not sufficient to run the models with this huge data. So have randomly sampled the data into 5,000 records. Then we have spitted the train and the test data in 60% of non - clicked advertisements and 40 % of clicked advertisements.

Future Work:

Results proved that the proper merging of the data and good feature selection is key for building a good model. The models that we created did not give us great results and would have been slightly more than the above average model. As the data is too large and and cross validation for any given model with the sampled data is taking more than 3 hours of time. There were models which took us the whole night to run (nearly 6-7 hours) on a 32 GB machine. In order to resolve this time complexity, we have to use distributed computing. There are several technologies and tools which work on distributed computing like Teradata, Hadoop, Amazon web Services (Redshift, Snowflake). The Top scores in kaggle have used Factorization machines which are known for doing well with on large datasets. So, in the future we can try using distributed computing and Factorization machines for better results.

Overall Conclusion:

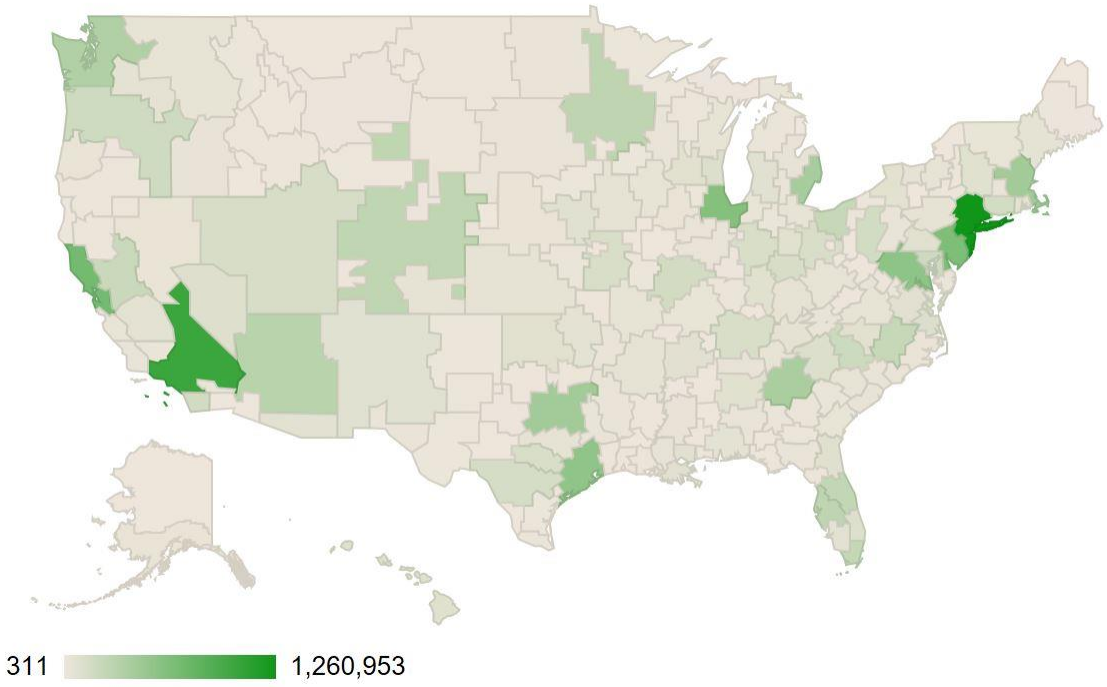
Given a very large dataset and number of attributes, lot of time was taken to solve the time and space complexities issues. Apart from that data understanding has taken so long time as there were inter linking(loops) within the data.For example,if the **advertisement on a particular document is clicked then it opens another document known as document_landing and clicking the advertisement displayed on the document_landing might again link us back to the original document.** It became quite complex to solve the problems of this kind. So, most of the time was taken for the data understanding.

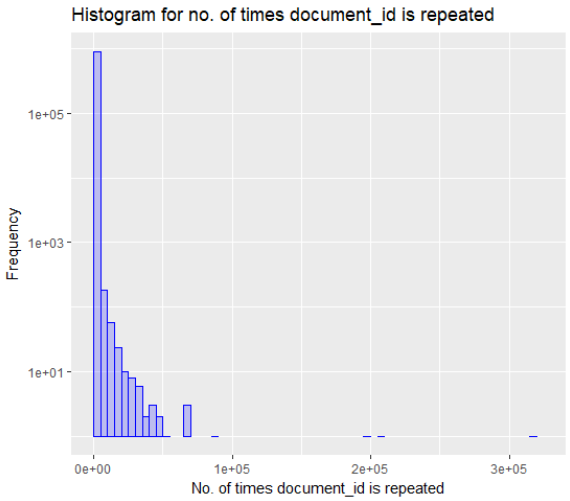
Our best model was random forest and could have got even better results if we had more time and computing resource to solve the time and space complexities. We have learnt that there is a high correlation between the category of document the user are presented with and the advertisement's they have clicked. For example, if the actual document is on business then the user is most likely to click on the business-related advertisement. The dataset really challenged us and we definitely have learnt a lot from this project.

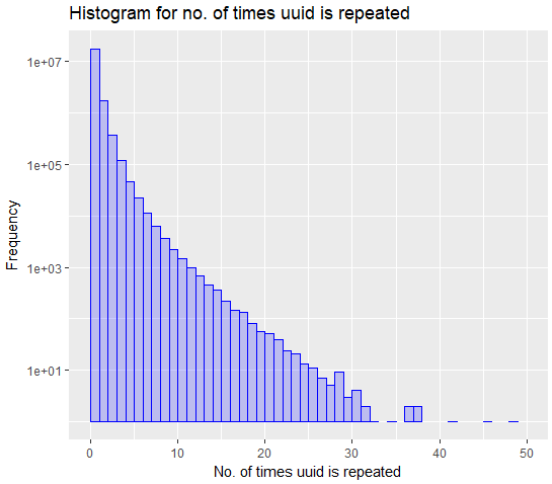
References:

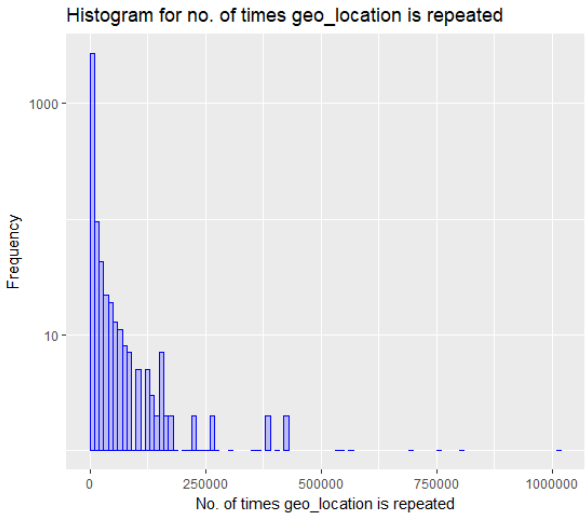
1. Factorization Machines Steffen Rendle Department of Reasoning for Intelligence. The Institute of Scientific and Industrial Research https://www.ismll.unihildesheim.de/pub/pdfs/Rendle2010FM.pdf
2. Hoachuck, J., Singh, S. and Yamada, L., Outbrain Click Prediction.
3. L. Breiman, "Random Forest," Machine Learning, vol. 45. Springer US, 2001, pp. 5–32.
4. C. D. Manning, P. Raghavan, and H. Schütze, "Evaluation in Information Retrieval," An introduction to information retrieval. New York: Cambridge University Press, 2008.

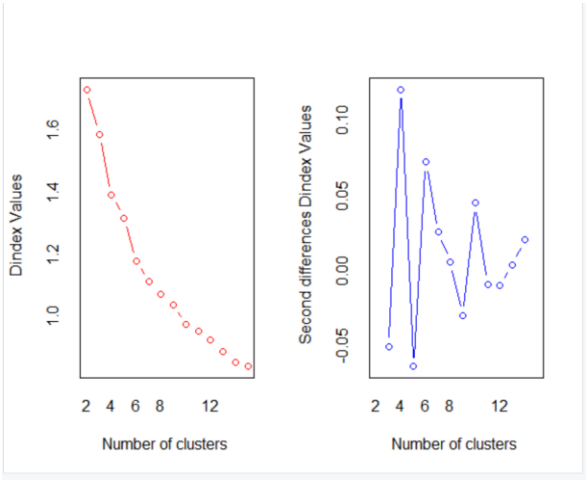
Appendix:


311
1,260,953


Histogram for no. of times document_id is repeated
Frequency
1e+05
1e+03
1e+01
0e+00
1e+05
2e+05
3e+05
No. of times document_id is repeated


Histogram for no. of times uuid is repeated
Frequency
1e+07
1e+05
1e+03
1e+01
0
10
20
30
40
50
No. of times uuid is repeated





For 5000 data