CLICK-THROUGH RATE PREDICTION

NOT SO NAÏVE BAYES:

PRASHANTHI CHANDRASEKARAN

SANDEEP RAMESH

SOURABH

SUSHIDHAR JAYARAMAN

VJAYA VASAVI SEENIVASAN

Table of Contents

PROBLEM STATEMENT:

Click-through rate (CTR) is a very important metric for evaluating ad performance in online advertising. As a result, click prediction systems are essential and widely used for sponsored search and real-time bidding. In our dataset, we have provided 11 days' worth of Avazu data to build and test prediction models.

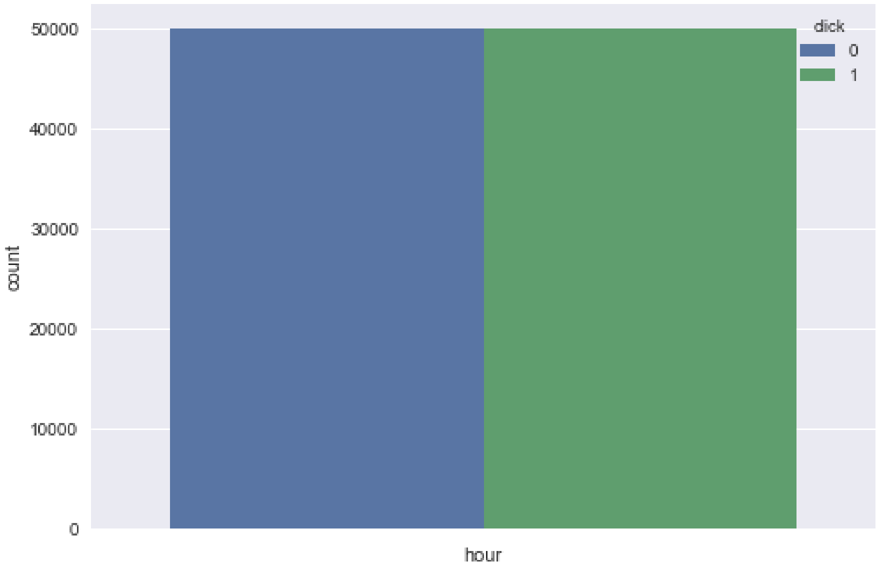
DATA FIELDS:

ATRRIBUTES	DESCRIPTION
Id	ad identifier
Click	0/1 for non-click/click
Hour	format is YYMMDDHH
banner_pos	Takes values between 0 and 6. It depicts the placement of ad on the screen
site_id, site_domain, site category	The attributes concerning the information about the site
app_id, app_domain, app_category	The attributes concerning the information about the app
device_id, device_ip, device_model, device_type, device_conn_type	The meta data of device
C1	anonymized categorical variable
C14-C21	anonymized categorical variables

The datatypes include: int64, object and all the attributes are non-null.

DATA COLLECTION:

The source of the original data is from https://www.kaggle.com/c/avazu-ctr-prediction/data .The data size was 5GB with 40 million records which was very large to be compatible for analysis with the available resources. Considering the occurrences of memory constraints, random sampling was done using SQLite DB to fetch 300K of records with equal number of clicks and non-clicks to avoid the bias in the observations and the analysis.240K of the records are assigned for training and 60K for test to perform the analysis. Our sampled data consists of 2 days worth of Avazu data out of 11 days to build and test prediction models.



DATA PREPARATION:

Label Encoding:

The data was label encoded by two methods:

Method1: we used the Label Encoder package from Sklearn to do automatic encoding of categorical features.

Method2: Each feature was label encoded based on the probability of clicks manually. This was done to have a healthy comparison of the results for various classifiers.

Ultimately Method 2 was used because it gave better results. Thus feature engineering of categorical features helped to increase the accuracy of various algorithms.

Scaling:

The data was standardised using Standard Scalar package for some of the features with mean equal to 0 and standard deviation equal to 1.

Data type conversion:

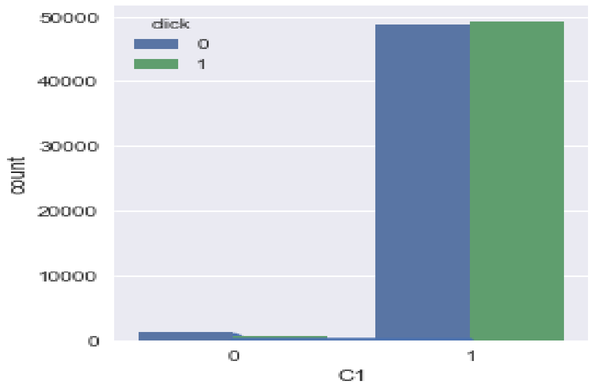
The attributes including site_id, site_domain, site category, app_id, app_domain, app_category etc were grouped based on the first character of the alpha numeric string to reduce the number of categories. The attributes including C1, banner_pos, C14-C21 etc were converted as categorical variables from int64 for memory optimization while training the model.

Feature Engineering:

Listed below are some of the features on how the feature engineering was done.

- C1:

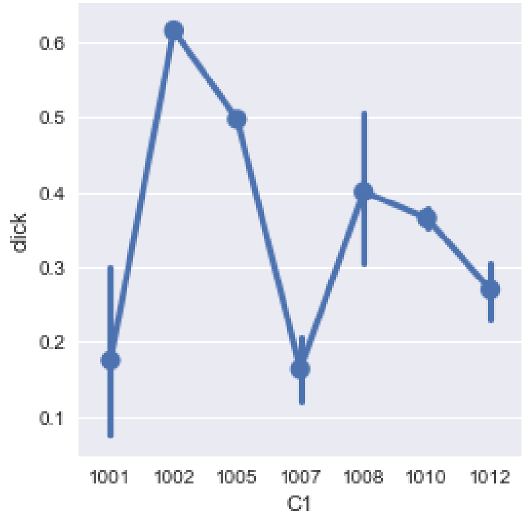
Initially, there were 7 categories for C1 and based on the count of the clicks, it has been reduced to two categories. We inferred that counts cannot be used to categorize the features as the counts of one of the category of C1 could be more but the probability could be less, hence we considered the probability of clicks for categorising C1 as below:



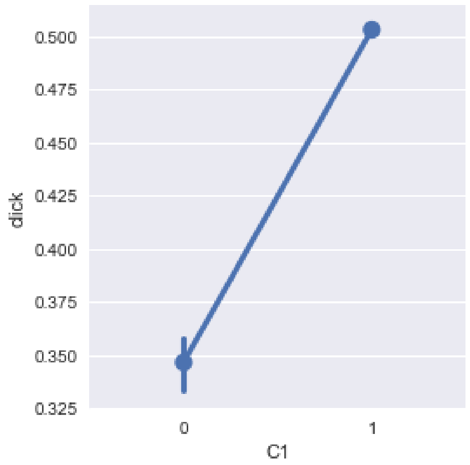
Categorising based on the probability of clicks:

```
1    98013
0     1987
Name: C1, dtype: int64
```

Initial Plot:

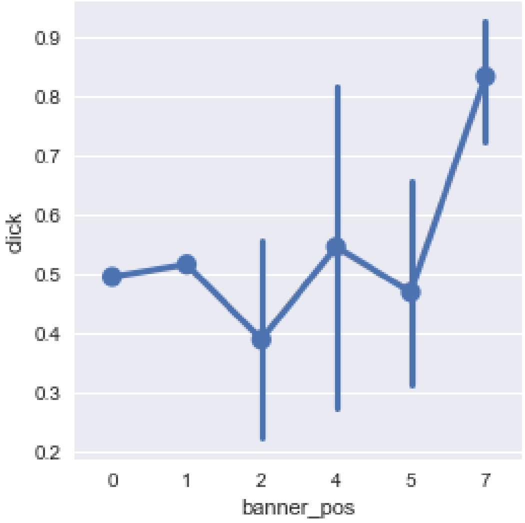


After Feature Engineering:

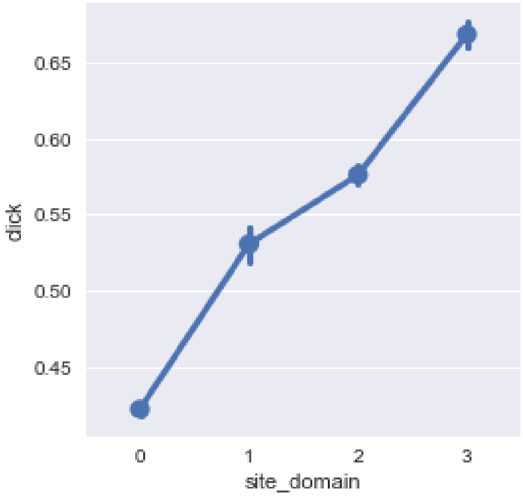


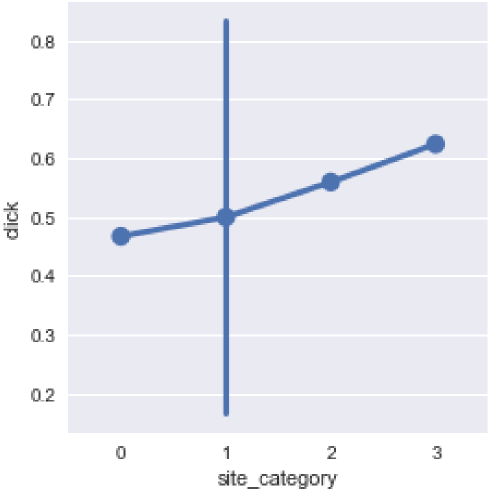
This type of feature engineering was done for all categorical variables in order to provide a clear view of the data as well as reducing the amount of categories present in each feature.

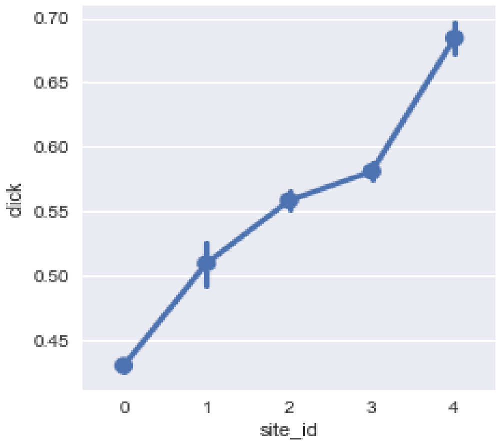
- banner_pos:



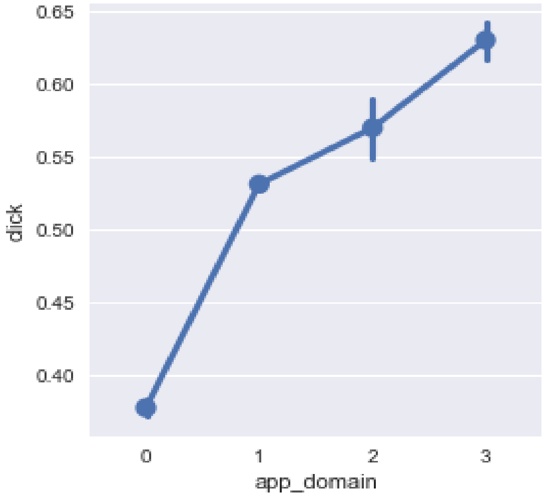
- site_id, site_domain, site category:

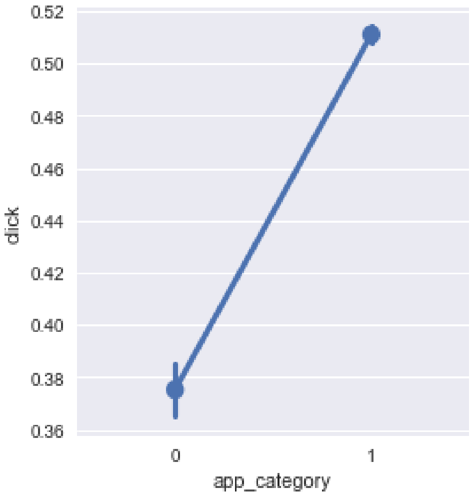




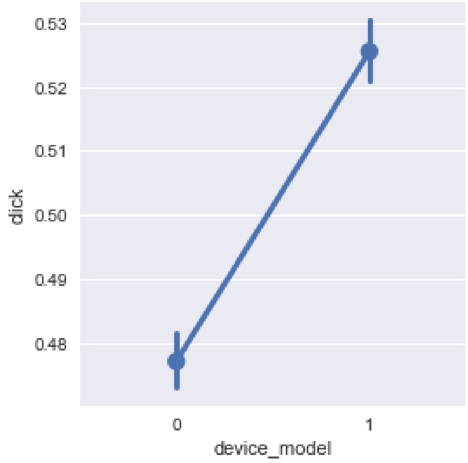


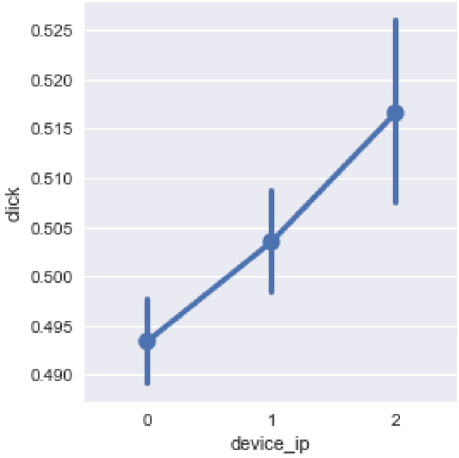
- app_domain, app_category:

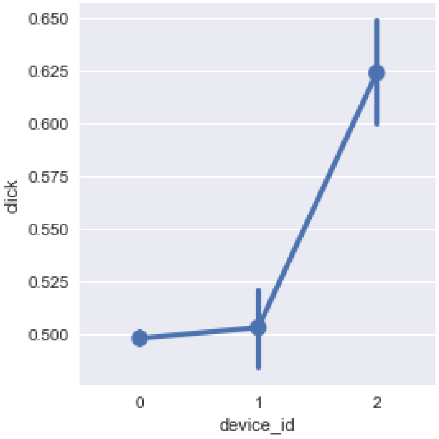


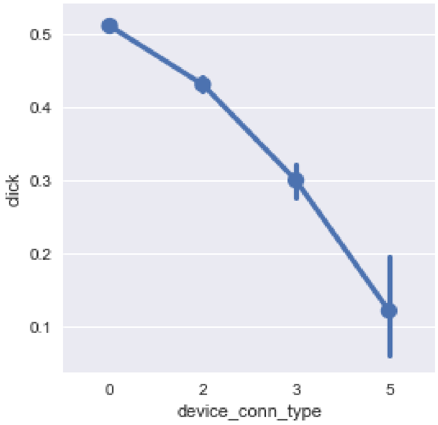


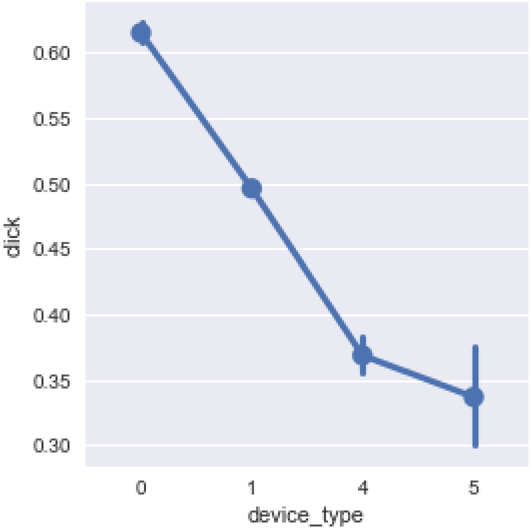
- device_id, device_ip, device_model, device_type, device_conn_type:



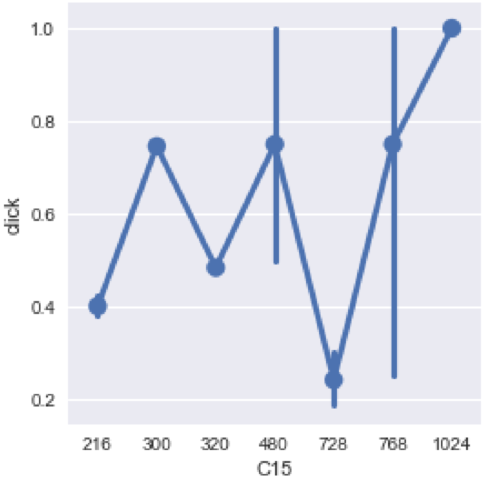


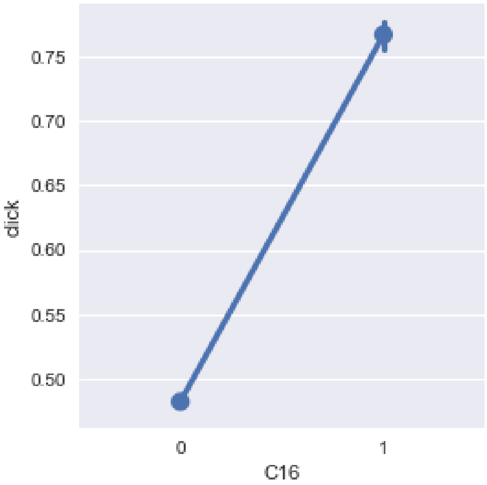

0.650
0.625
0.600
0.575
0.550
0.525
0.500
dick
0
1
2
device_id

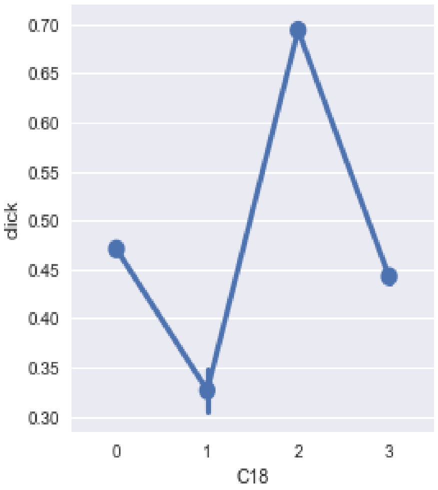

0.5
0.4
0.3
0.2
0.1
dick
0
2
3
5
device_conn_type


0.60
0.55
0.50
0.45
0.40
0.35
0.30
dick
0
1
4
5
device_type

- C15, C16, C18:

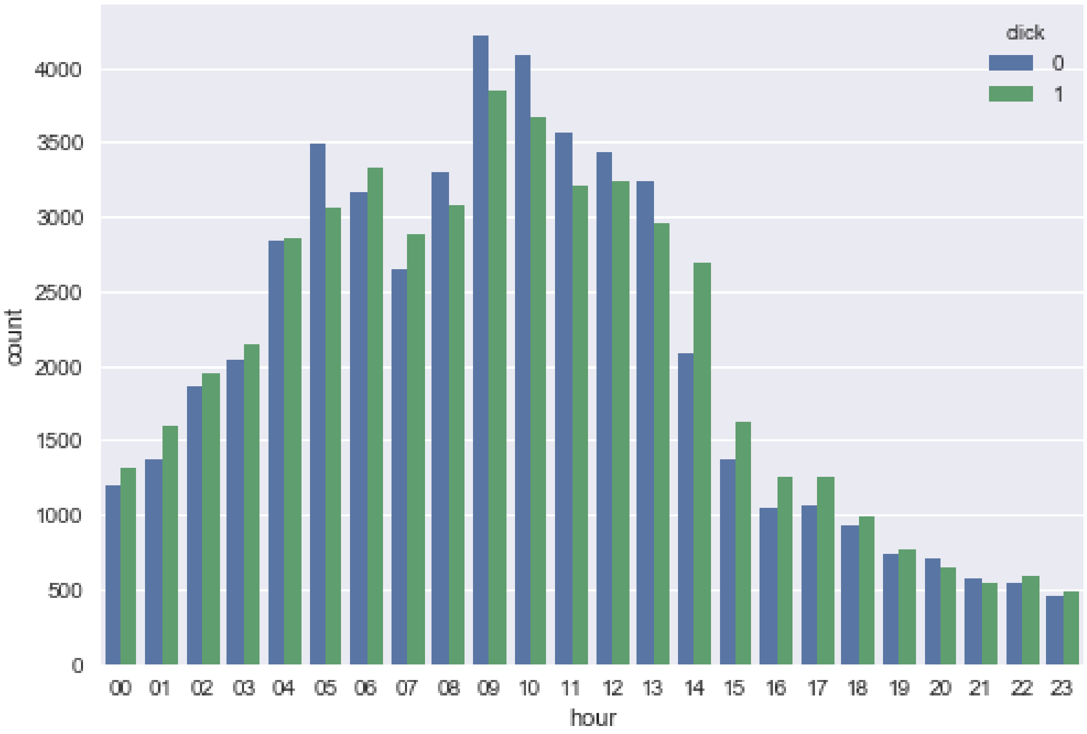

1.0
0.8
0.6
0.4
0.2
dick
216
300
320
480
728
768
1024
C15


0.75
0.70
0.65
0.60
0.55
0.50
dick
0
1
C16



- Hour:

Initially the data for this feature was in the format MMDDYYHH. From this HH i.e. the exact hour of the day was extracted for comparing the number of clicks throughout the day. We find that, the number of clicks increase during the day time and is maximum at 9 am and is comparatively high after 2PM.



From this we infer that it is wise to place our ads on the website during the latter part of the day.

ATTRIBUTE SELECTION:

The original data had 22 attributes. It might not seem to be appropriate to use all the features as it might result in a complex model. Hence Recursive Feature Elimination using Cross Validation or RFECV was used to select the optimal number of features. Random forest classifier was used to fit the data and select the optimal features based on their rank. The optimal number of feature selection given by the algorithm was 16.

FEATURES	RANK
C14	1
C20	2
Hour	3
C21	4
C17	5
device_ip	6
site_domain	7
site_id	8
device_model	9
app_domain	10
C19	11
device_conn_type	12
app_id	13
device_id	14
banner_pos	15
site_category	16
C18	17
device_type	18
C16	19
app_category	20
C15	21
C1	22

MODEL SELECTION AND VALIDATION:

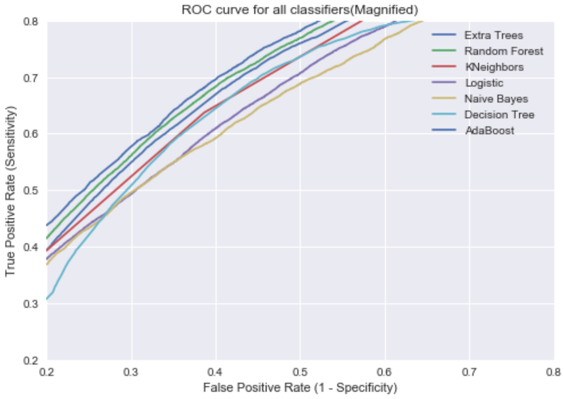
The following classification algorithms were used for our analysis:

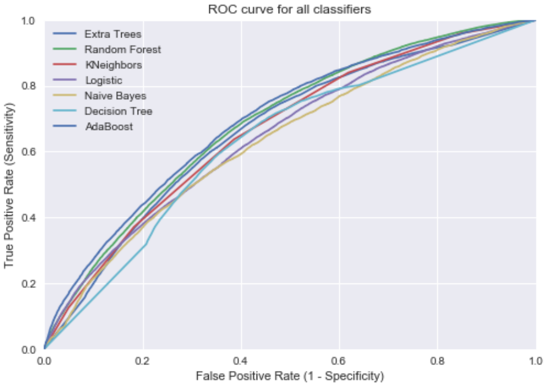
- Extra Trees
- Random Forest
- KNN
- Logistic
- Naive Bayes
- Decision Tree
- AdaBoost

Model was trained with 240K rows of data and validated with 60K records. Below are the results of the algorithms with the important metrics.

Algorithms	Accuracy	F1 Score	ROC AUC Score
AdaBoost	0.647166667	0.656977121	0.696067633
Random Forest	0.640283333	0.643262095	0.688126876
Extra Trees	0.630966667	0.623550614	0.665670167
KNeighbors	0.625216667	0.627248164	0.665366359
Logistic	0.6048	0.606008241	0.6487891
Naive Bayes	0.597466667	0.598977186	0.634369743
Decision Tree	0.620066667	0.610843661	0.63316261

We find that the comparative accuracy scores seem to be maximum for random forest and AdaBoost. However, considering the F1 score and AUC score, AdaBoost gives better results. Therefore, for our data AdaBoost is the best algorithm.





In evidence to the inference with respect to the model selection, we find that Adaboost is the best model for our data based on ROC Curve. It has the highest value of 0.696067633.

INSIGHTS:

At an abstract level, without considering the models, we can find that if the ads are showcased after 2pm and on banner position 7, the probability of clicks is high and so revenue generation will be more.

If ads are displayed on device type 0, the probability of clicks in high compared to other device types.

Site IDs that fall under category 4 and device ID under category 2 have maximum clicks. These can be considered for revenue increase.

CONCLUSION:

Through these analysis and model validation, the primary objective of this project to predict whether a mobile ad will be clicked or not has been achieved with a good accuracy considering the size of the sample dataset.

A major challenge of this project was the number of observations to load and analyze. But conversion of CSV to SQLiteDB data made the loading and analysis much easier.

Several algorithms like Tree based, Ensemble, Linear were used for this project. But based on the AUC score and F1 score, AdaBoost turned out to be the best classification algorithm to be used in our model.

In future, we must come up with a way to deal with the entire data so that we have more accurate results.

REFERENCES:

https://www.kaggle.com/c/avazu-ctr-prediction

http://machinelearningmastery.com/compare-machine-learning-algorithms-python-scikit-learn/