Big Data Analytics

Project 2 Report - Team 4, Spring 2018

TalkingData - Fraud Detection

Submitted By:-

Anuj Vyas

Harsh Pathak

Nguyen Vo

Rushikesh Naidu

Guided By:-

Prof. Yanhua Li

Abstract:

Fraud risk has incremented to huge numbers since the inception of click ads in mobile devices and digital devices. For companies that advertise online, click fraud can happen in huge volumes and this leads to misleading click data and wasted money. This means that companies pay up for these ads and the costs go up with every click and every click does not result in the application advertised but fraudulent links. This causes major losses and waste of money. TalkingData is China's largest independent big data service platform which covers 70% of active mobile devices. However, 90% of these clicks are potentially fraudulent. Currently TalkingData collects all the information about fraud clicks to blacklist IPs and devices. The project aims at developing an algorithm for TalkingData that dynamically predicts whether the clicks will eventually result in a downloaded application.

Introduction:

The digitalized promoting system comprises of 1) distributors that create content to draw customers to their locales and 2) sponsors that compensation to put advertisements – show, video, seek – on those distributors' destinations. In years past, enormous publicists worked straightforwardly with huge distributors and set their advertisements specifically ("co-ordinate purchases"). However, with the multiplication of long-tail, specialty sites, a framework was expected to put promotions crosswise over huge quantities of minor destinations. Automatic advertisement tech was made to do this, and promotion trades were conceived. In any case, in their push to develop as fast as could reasonably be expected, advertisement trades let any sites include themselves into the trade and convey promotions to profit; this escape clause at last enabled a huge number of phone sites to enter the computerized promotion biological system. Some of these locales were out and out false — they didn't considerably try to make substance to pull in genuine gatherings of people.

The locales were made exclusively to convey advertisements; and their "movement" was totally phony as well — bots modified to over and again stack pages to make the promotions stack. The phony movement, counterfeit impressions, and phony snaps made by these advertisement misrepresentation bots botch up investigation. This terrible information implies that investigations, advancements, and prescient models are likewise totally fouled up. As a partner put it "without clean information, you have nothing." I may contend, that it's far and away more terrible than nothing; since you may wind up unintentionally sending more cash to the awful folks. While prescient investigation look forward into the future and web examination take a gander at memorable information, the ongoing theme that is valid for both is "waste in, rubbish out." But on account of promotion misrepresentation, it's extremely difficult to tell what's trash, on the grounds that the movement of bots may look better than average in the information.

The precision of the information is of most extreme significance to doing great advertising. By understanding what bots can do, you won't just have the capacity to recognize their action in your own investigation, yet in addition find a way to limit its negative effect on the legitimacy of your models and examinations. In this article, the methods talked about won't require propelled math or measurements. So business experts and examination professionals alike ought to have the capacity to incorporate them immediately. Battling extortion ought not be one individual's employment nor should it remove time and assets from just examination experts. By seeing how advertisement misrepresentation functions in basic terms, it ought to wind up some portion of the every day routine of all promoting, business, specialized, and investigation staff to perceive the indications of extortion; and find a way to limit it and enhance business results.

The extensive increase of fraud clicks and rise of fake IPs connection have caused a lot of wastage of money and companies are currently looking for ways to minimize this problem. In order to do this several companies use several different processes. TalkingData is the largest independent big data platform for mobile devices. Due to this increase in fraud clicks and IPs, the companies paying for ads are severely facing loss of money. To prevent this, the current technique employed by TalkingData extensively revolves around tracking the clicks by users which do not result in a downloaded application and eventually blacklisting these channel, IPs and devices. In this project, we propose an algorithm that dynamically predict whether a click would result into a downloaded application or would it redirect to a fraudulent site.

The project has employed two techniques to understand the data and predict the fake clicks. The first method is the scalable deep learning method where the dataset is divided into partitions in order to remove the class imbalance in the principal dataset and with these datasets we apply deep learning algorithm on all the datasets. The dataset is divided in a way that the majority class is divided into partitions of the size of minority class. Applying deep learning feed forward network (with binary cross entropy as loss function) on each of these result in definitions for each point. We then apply ensemble technique to define the class which the point in consideration belongs to. The second method is using several machine learning models and constantly optimizing the algorithms to obtain the best possible accuracy. The final goal of the project is to merge these techniques to find a compromise between accuracy and computation efficiency in terms of time and memory.

Motivation

Business Impact:

In spite of the broad selection of web analytics devices and expanded information availability, information around a user's realness isn't portion of the choice making handle for a few of the biggest sponsors, organizations and systems. So, the address for anybody buying activity online remains – is this client indeed genuine?

The business will impact if the companies implement Ad-tracking fraud detection in their organization.

1. Expand income potential

By presenting real-time Genius about the quality of the source, fraud detection approves marketers to make skilled media buying decisions before conversion metrics are available. This capacity to shortly filter out fraudulent sources opens the doorways to new channels, which previously concerned a high degree of hazard and provided a decrease ROI to the advertiser.

2. Recapture lost probability costs

At the equal time, organizations can reallocate advertising dollars from fraud to higher excellent media buys. This is feasible thru a obvious media shopping for process, which sheds greater light on the fine of character sources. By transferring resources to greater profitable sources of traffic, marketers can further increase revenue.

3. Minimise chargebacks with real-time knowledge

Certain performance metrics can take days or even months to process, but fraud detection is accessible in real-time, frequently earlier than the writer has been compensated for the site visitors or the user has even made a purchase. This allow companies to pinpoint and cast off fraud earlier than paying the source, or worse, dropping treasured commercial enterprise companions and incurring chargebacks.

4. Emphasis on quality

Businesses make smarter media buying selections and appeal to new manufacturer names to make bigger the breadth of business, all while regaining the transparency and manage over the media sources. Providing higher first-rate facts is a competitive gain that attracts new enterprise and permits businesses to open up a conversation about increasing the relationship with current clients.

5. Centralised fraud intelligence

Fraud detection services advantage from the intelligence gathered from thousands of millions of customers and information acquired from years of ride in the area. This center of attention affords the most complete solution that would be very hard to strengthen in-house. Since online fraud is constantly evolving to sidestep detection methods, solely a centralised fraud brain provider can stay on top of the modern day tactics.

Method

Data Sets:

Each row of the training data contains a click record, with the following features.

•IP: IP address of click.

•App: app id for marketing.

•Device: device type id of user mobile phone

•OS: OS version id of user mobile phone/device

•Channel: channel id of mobile ad publisher

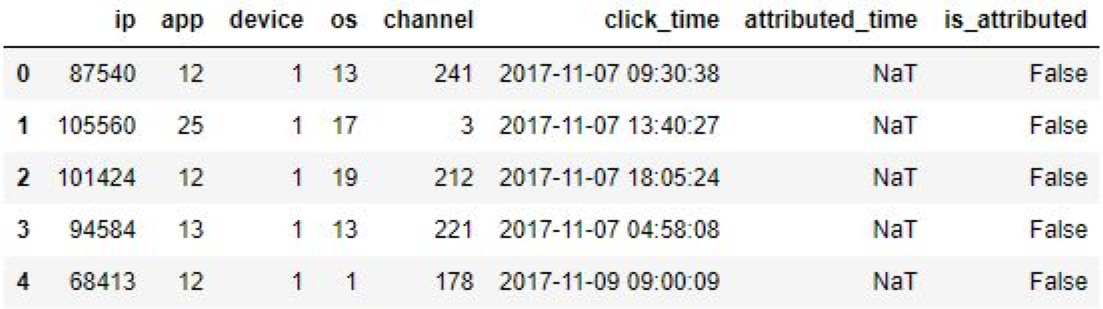
•Click_time: timestamp of click (UTC)

•Attributed_time: Time of the app download after clicking the advertisement

•Is_attributed: To be predicted.

Exploratory Data Analysis:

The dataset consists of 8 attributes containing 200 million observations in terms of click information for the past 4 days. This information is segregated in terms of application, IP, device, OS, Channel and Click_time. The dataset however upon being downloaded requires a lot of cleaning with several missing values and undefined observation. Thus, we start the exploratory analysis by initially analyzing the dataset.



	ip	app	device	os	channel	click_time	attributed_time	is_attributed
0	87540	12	1	13	241	2017-11-07 09:30:38	NaT	False
1	105560	25	1	17	3	2017-11-07 13:40:27	NaT	False
2	101424	12	1	19	212	2017-11-07 18:05:24	NaT	False
3	94584	13	1	13	221	2017-11-07 04:58:08	NaT	False
4	68413	12	1	1	178	2017-11-09 09:00:09	NaT	False

Table 1: Dataset

The figure shows the top 5 rows of the dataset. The attributes IP, app, device, os and channel have been converted into integers from strings in order to perform exploratory analysis. The clicktime is converted into date and the attributed time and is_attributed are boolean values. Ip in the dataset is the IP to which the advertisement redirects the user. The application is the application number given to the application whose advertisement is being telecasted. The device represents the type of device which is used by the user. The channel is the channel of connection from the click to the application downloading network. The attributed time is the timestamp of the time at which the application was downloaded. It is not available for clicks which did not result in application downloaded and just for the ones which resulted in application downloaded. The is_attributed is true for values where there is a time stamp on the attibuted_time because it means that the advertisement was not fraudulent.

With all the attributes defined, the project started exploring the dataset and understanding trends in the data and how to prepare it to perform predictive algorithms to achieve the outcome.

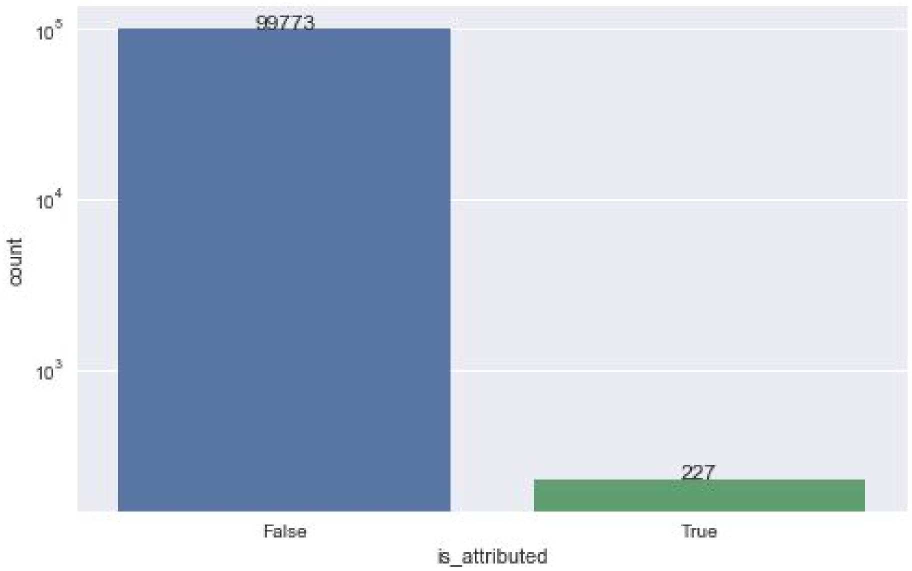


Figure 1: Class Imbalance in outcome

The outcome variable which defines a non-fraud advertisement is given by is_attributed by the dataset. The above graph shows two classes of the attributed variable. It shows the extreme class imbalance in the attribute. There are 99773 observations which were fraudulent corresponding to 227 observations which are non-fraud. This significant difference needs to considered into consideration to avoid any bias in the applied mathematical model.

The IP, application, device and os are all independent values and in order to understand the number of independent attributes running on the network it is important to understand the total number of unique values. This could help in understanding if particular attributes are fraudulent to block them for the future.

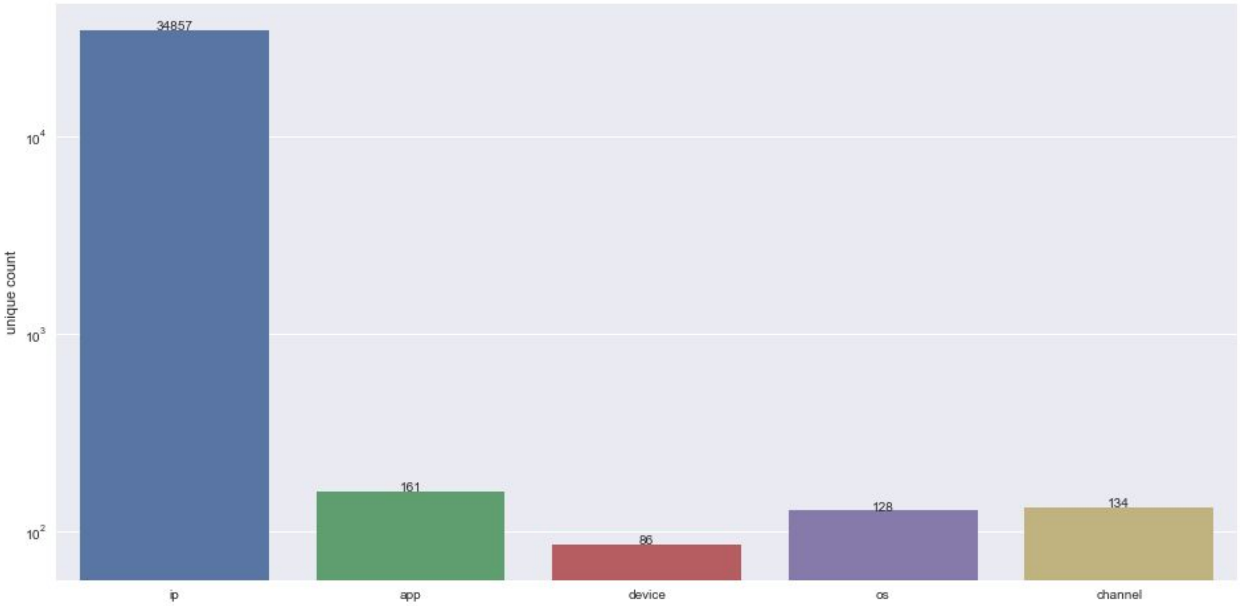


Figure 2: Unique values in all the factors

The above plot shows that IP have a significant number of unique values which explains the fact that there are a number of IPs which are redirecting users to different places from clicks. These may be real or fraudulent. The other attributes have understandably few unique values in terms of applications, devices, operating systems and channel.

Now, understanding the unique values of the attributes would be incomplete without knowing the how each of these unique values differ in terms of number of clicks. We maintain a hypothesis here that there would be more number of clicks through some particular unique factors and relatively fewer clicks with other ones.

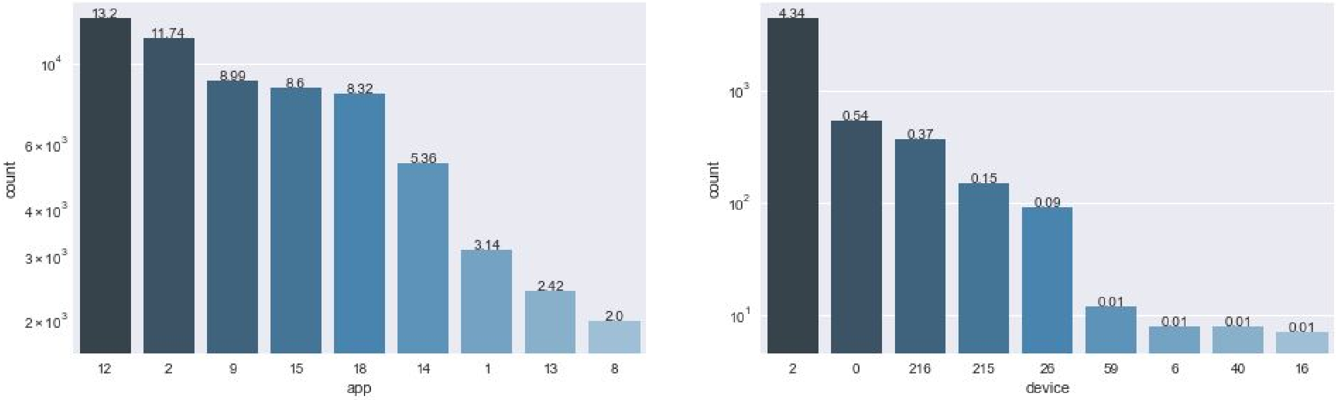


Figure 3: Division of clicks across all unique values in App and Device

The figure shows that the more than 60% of the clicks come from the top ten applications. Also there are some very significant devices which are used to click these advertisement and they correspond to Iphones and Android phones.

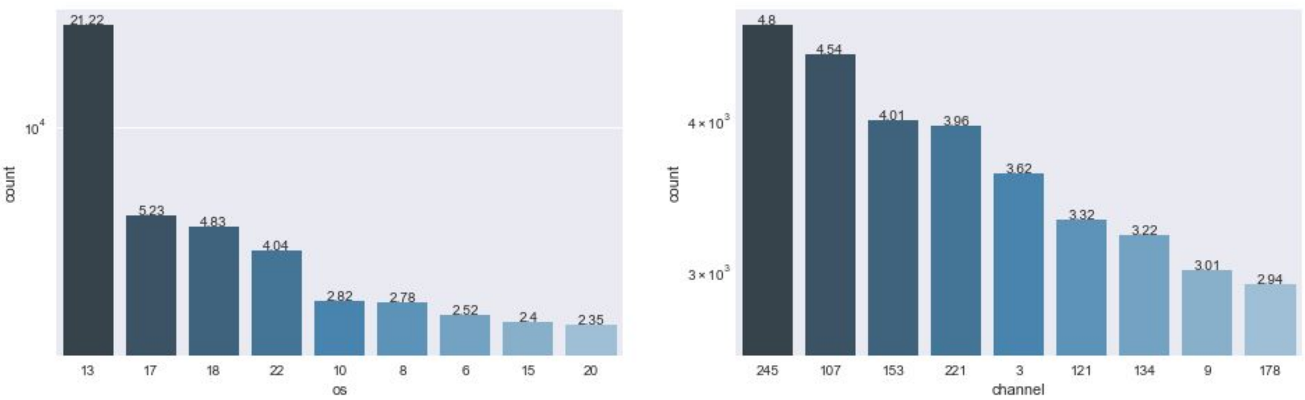


Figure 4: Division of clicks across all unique OS and channel

We can view from the plot that operating system 13 has the most number of click and these clicks are significantly higher than the other operating systems. This shows the importance and the popularity of this operating system. On the right hand plot we can view the division of clicks over the different channel. It shows that over 70% of the clicks are concentrated on the top 10 channels displayed.

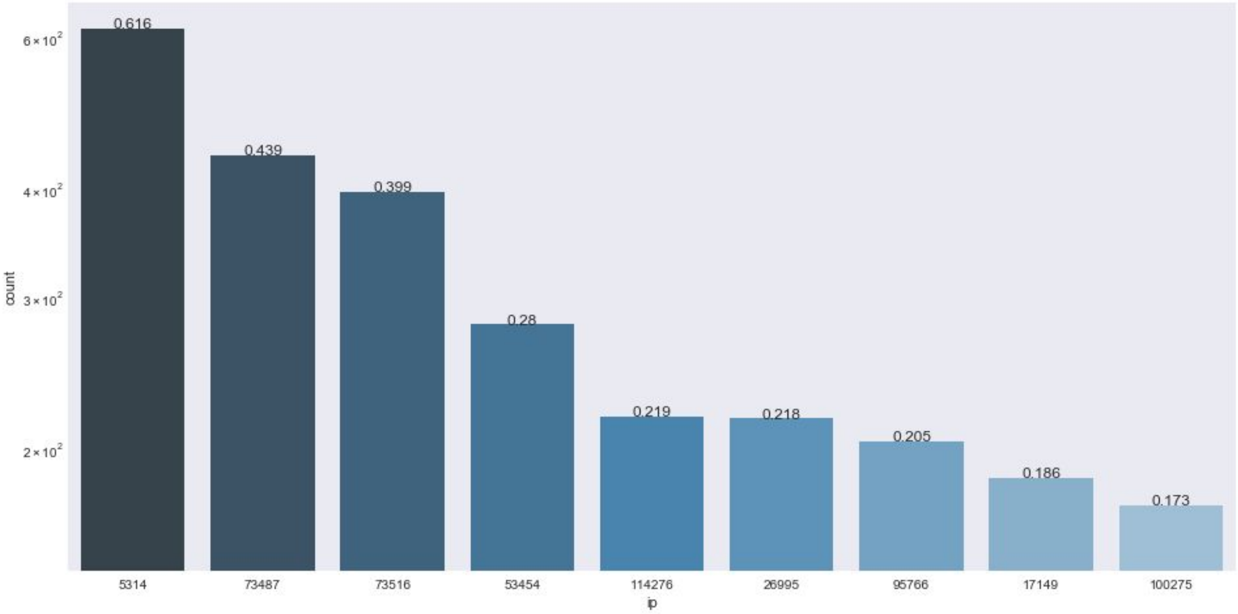
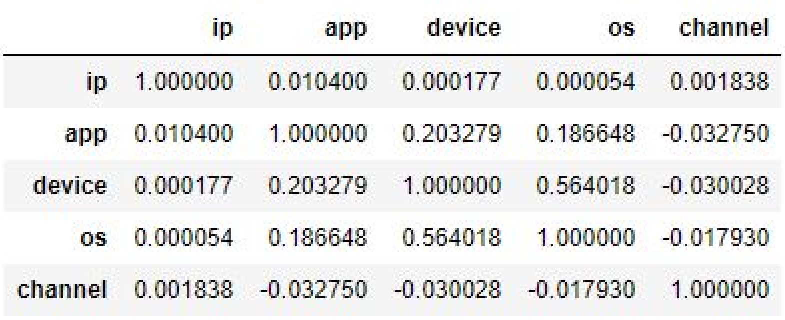


Figure 5: Division of clicks across all unique IPs

The plot above shows the division of clicks in terms of IPs. There are a number of IPs in the dataset however the top three IPs are responsible for more than 50% of the clicks for the applications.

Now that we have understood how every attribute differs in terms of uniqueness and number of clicks, it is important to understand the correlation between these attributes. The correlation matrix provides a dynamic view of how the all the different factors involved in the mobile phones and advertisement links affect each other. We understand that their would be a correlation between a device and os but it would be interesting to understand other hidden correlations.



	ip	app	device	os	channel
ip	1.000000	0.010400	0.000177	0.000054	0.001838
app	0.010400	1.000000	0.203279	0.186648	-0.032750
device	0.000177	0.203279	1.000000	0.564018	-0.030028
os	0.000054	0.186648	0.564018	1.000000	-0.017930
channel	0.001838	-0.032750	-0.030028	-0.017930	1.000000

Table 2: Correlation between factors

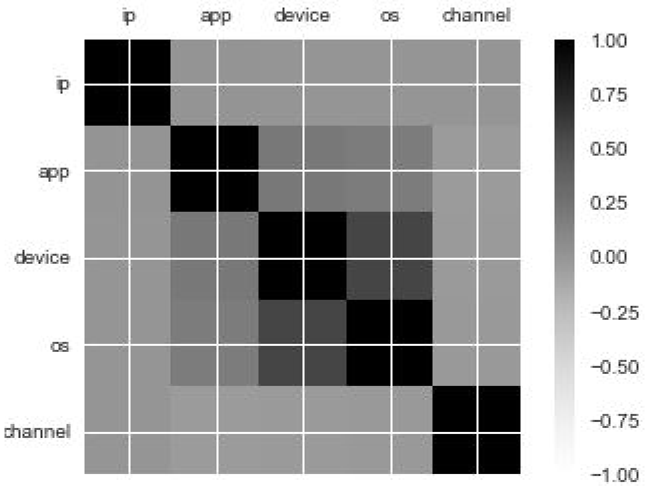


Figure 6: Correlation Matrix

The plot computed above is called as the correlation matrix. A darker colored square indicates a direct relationship between the two components. If the square is extremely darker it means that as one factor increases the other also increases. On the other hand, a white square indicates a direct negative correlation. If one factor increases, the other decreases.

We can see that the only strong correlation we can observe from the dataset is between device and os. There is also a weak correlation between device and app and app and os. The specific degree of correlation is also displayed in the table above.

The dataset also provides the time at which the click was encountered by the user. It is important to understand specific timeframe which are popular for users in order to explain what time periods have the most clicks and if they corresponding to any trends with respect to frauds. Thus, we divide the timestamp into the hour of the day and the day of the month and based on this we understand the trends of downloaded applications and basic click times over the day and the month.

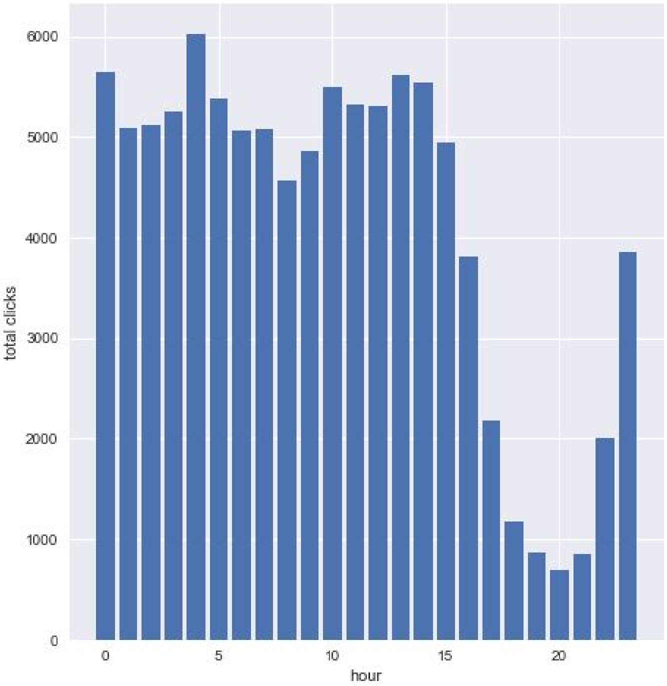


Figure 7: Clicks with respect to hours

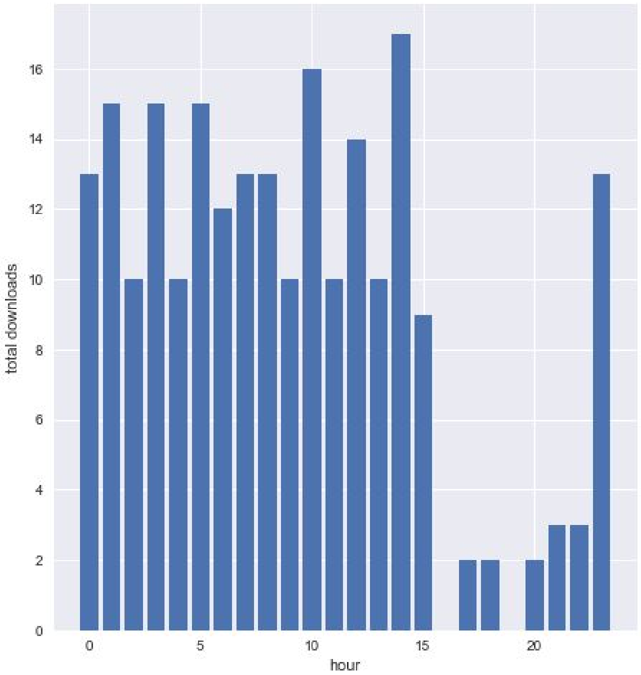


Figure 8: Downloads with respect to hours

The plots displayed above show the fluctuations of clicks for different time of the day. From the graph we can see that there is no particular trends in the clicks and the corresponding downloaded application expect the fact that most clicks are during the day and the traffic goes down as the end of the day.

Scalable Deep Learning:

In order to predict the click time we downloaded the dataset from the kaggle. As the dataset was to large we decided to divide the dataset into smaller fragments. And perform experiments on those fragments to get the better results. The Dataset contains 7.1 GB of raw data. Out of which there is approximately 7 GB of Negatives i.e. app is not download after the advertisement is clicked. And 27 MB of Positives i.e. each time the advertisement is clicked the app is downloaded.

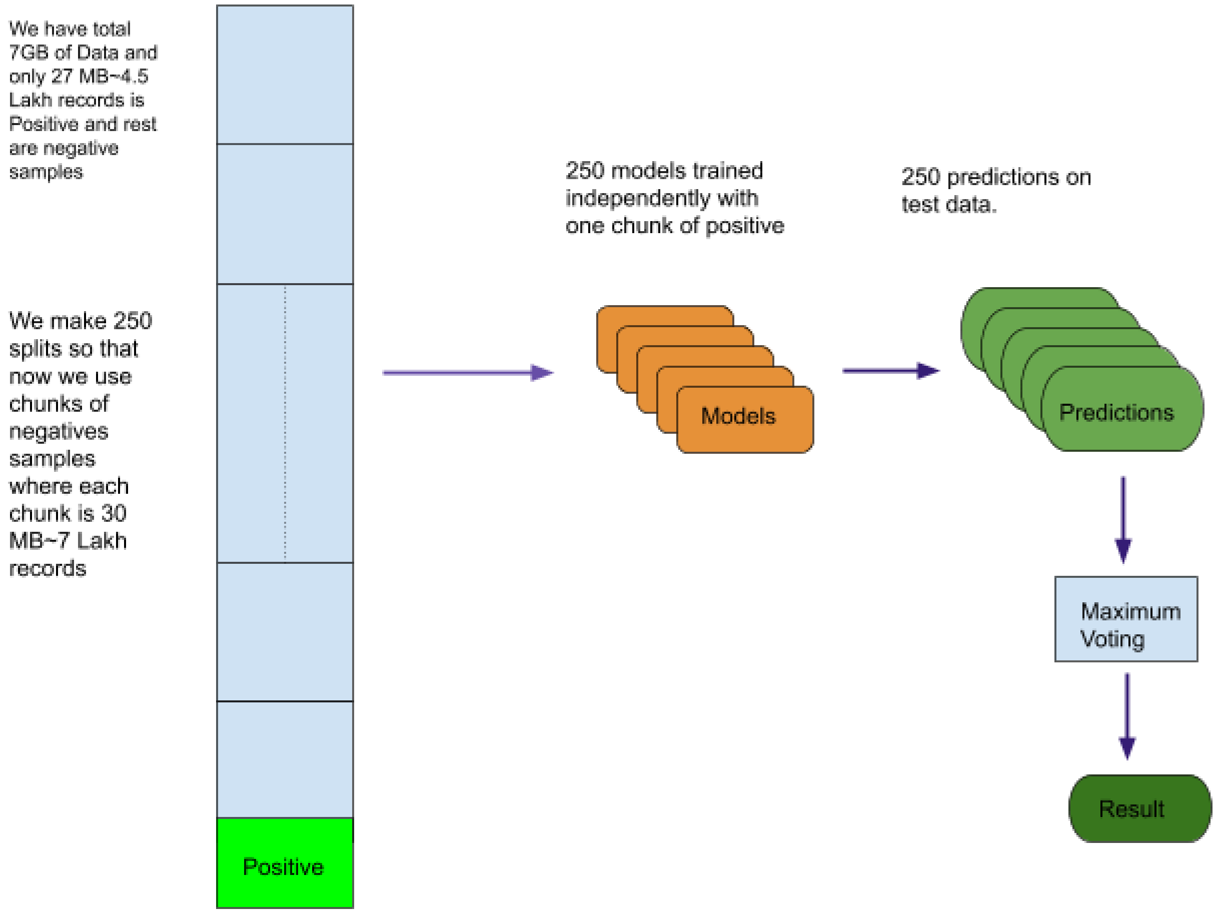

We have total 7GB of Data and only 27 MB~4.5 Lakh records is Positive and rest are negative samples
We make 250 splits so that now we use chunks of negatives samples where each chunk is 30 MB~7 Lakh records
Positive
250 models trained independently with one chunk of positive
Models
250 predictions on test data.
Predictions
Maximum Voting
Result

Figure 9: Scalable Deep Learning Models

Then after segmenting the data we developed models for classifying the data. We developed 250 models consists of 30 MB of negatives and 27 MB of positives. Now each of the models developed we calculate the parameters which is then tested with kaggle test dataset.

Deep Learning Architecture:

Here, we are using simple feed forward neural network with all the inputs from the data file. Following is a 4 fully connected hidden layer and finally we calculate binary cross entropy as our loss. Activation functions that we use is ReLU for hidden units and Sigmoid for last layer.
All layers are added with 20% probability of dropout so that there is an effect of regularization in the model.

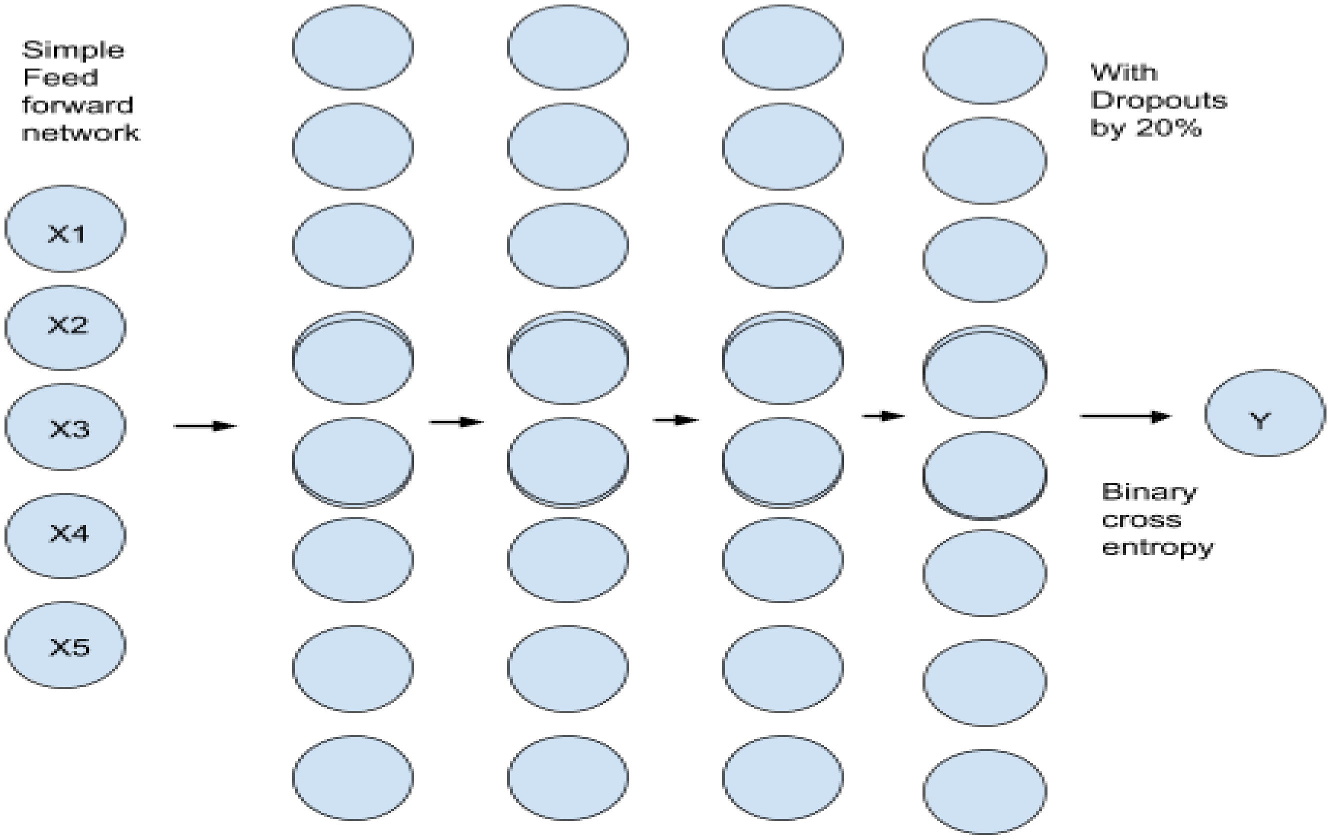


Figure 10: Deep Learning Architecture

More Feature Engineering:

In feature engineering we have basically added new features by grouping various columns and then plot the importance of each added combination of feature is evaluated using XGboost algorithm. For example, IP address is grouped with time and OS to contribute to the new feature in our Light GBM model. Below is the plot that explains the importance of every feature we used in our model.

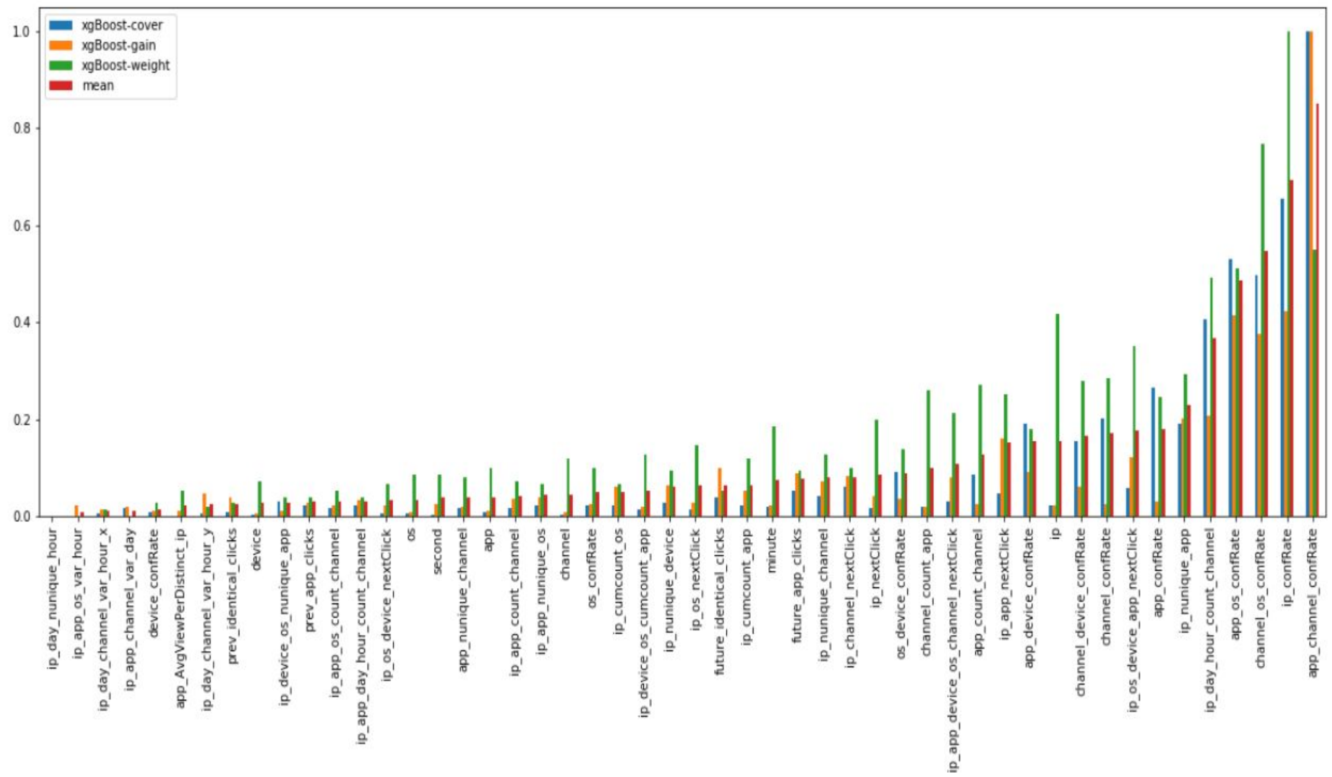


Figure 11: Feature Engineering

Results

We applied two techniques one was inspired by most of the Kaggle Kernels was feature Engineering followed by XG Boost or Light GBM boost method which solves the imbalance issue and give high accuracy as a result. Architecture 2: Feature Engineering and **Light GBM** boost: **96.3%** was achieved. Secondly, we used Simple 4-Layer Feed Forward Accuracy from Kaggle Test: **58.4%** with an added intuition of scaling the the algorithm so that we can iterate over whole dataset. Although we did not get a high accuracy in the second method but the approach is more robust and can be scaled easily.

Discussion

In this project, we deal with a very large scale dataset which is over 7.1Gb of dataset. For traditional machine learning algorithms such as SVM, Gaussian Mixture Models or other gradient-based algorithms, it is almost impossible to fit these models on this large dataset. The reason is that these algorithms relied on full update of the whole batch of data which require all data instances loaded on memory. Furthermore, computing gradient of the loss function with respect to parameters and update parameters based on large amount of dataset is a challenging task for traditional machine learning algorithms. Some tree-based algorithms such as Random Forest and XGBoost may overcome the issue by concurrently fitting multiple trees on multiple machines. However, they still suffer a serious issue in which fitting each tree requires all dataset instances loaded on memory. This is insufficient for big data. Therefore, a solution for these issues is to update model with mini-batch. Deep Learning models are perfect for long-term training with mini-batch. It has been shown that [4] training Deep Learning models can be trained effectively with mini-batch and it is scalable algorithms. Since the main mission of the class is to learn how to analyze the big dataset and fitting algorithms on large-scale datasets, using Deep Learning models for this project is a good decision. Furthermore, even when traditional machine learning models work for large dataset, their capability to capture aspects of dataset is limited. For example, in SVM, we only need to learn only a vector w standing for coefficients of fitted hyperplane or in Random Forest, each tree only capture small aspect of the dataset. Deep Learning, on the other hand, can model very complicated non linear relationship between parameters and capture complex underlying characteristics of the dataset. After careful consideration of multiple algorithms, we decided to apply Deep Learning model on this large dataset to build scalable and effective algorithms.

Conclusion

In conclusion, in this project we deal with a very practical problem in which we aim to detect fraudulent activities of mobile phone users. With the explosion of mobile devices, there are many it is very convenient for people to access the Internet. Unfortunately, many people take advantage of this openness to do fraudulent activities. As we showed in this project, many mobile phone users click on ads running on some websites without actually installing the application, leading to wasted money. With huge dataset collected by TalkingData, we built machine learning models to detect fraudulent activities. In the first step, we conduct data analysis to understand how fraudulent activities are different from legitimate downloads. We found that more than 60% of the clicks come from the top ten applications. Also there are some very significant devices which are used to click these ads from iOS and Android devices. Furthermore, more than 50% of clicks come from 3 devices. This is a clear signal of bot activities. Based on insights derived from the data analysis, we build machine learning including traditional machine learning algorithms and scalable Deep Learning algorithm. Our main target in this project is not only deal with a practical problem but also learn how Deep Learning model can be applied in this domain. For future work, we will further exploit other Deep Learning architecture such as AutoEncoder [1], Convolutional-Deconvolutional [2] or recently emerged model Matrix Capsule [3]. We have tried two methods here one pure Machine Learning with Light GBM boost which seems to do a pretty good job with the unbalanced data but there are issues with its scalability. Secondly, Deep Learning showing how powerful this architecture can become being scalable even if we have Terabytes of data. In future, we will combine both these methods and submit the higher accuracy in Kaggle community also addressing the issue of scalability, following Big Data Analytics norms and solve this problem.

References

[1] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 2010.

[2] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1417{1426. IEEE, 2017.

[3] Sara Sabour, Nicholas Frosst, and Georey E Hinton. Dynamic routing between capsules. In Advances in Neural Information Processing Systems, pages 3859{3869, 2017

[4] Priya Goyal, Piotr Dollar, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: training imagenet in 1 hour. arXiv preprint arXiv:1706.02677, 2017

[5] https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection/data

[6]https://www.kaggle.com/asraful70/talkingdata-added-new-features-in-lightgbm?scriptVersionId=3331854/code