Data Mining for Business (BUDT758T)

Project Title: Ad-tracking for conversion success – Fraud Detection

Team Members: Manikanta Koneru, Bharath Kumar Routhu, Meghana Chenreddy,
Jayasree Karthik Nandula, Sai Vaishnav Vinjamuri

ORIGINAL WORK STATEMENT:

We the undersigned certify that the actual composition of this proposal was done by us and is original work.

S.No	***Author Name***	***Author Signature***
1	*Manikanta Koneru*	*Manikanta Koneru*
2	*Bharath Kumar Routhu*	*Bharath Kumar Routhu*
3	*Meghana Chenreddy*	*Meghana Chenreddy*
4	*Jayasree Karthik Nandula*	*Jayasree Karthik Nandula*
5	*Sai Vaishnav Vinjamuri*	*Sai Vaishnav Vinjamuri*

II. Executive Summary

Websites make money when users interact with their advertisements, usually through impressions, interactions, or clicks. For example, an advertiser may pay a publication 20 cents per click. The publisher receives $10 per day or $300 per month if their ad produces 500 clicks each day.

Pay per click, often known as PPC, is the most popular option for many. These websites generate revenue by selling ad space on their platform. PPC sites allow advertisers to access millions of prospective customers while enticing consumers to participate by paying them to see adverts.

An advertising network (such as Google's AdSense or Yahoo! Search Marketing) puts the ad; the publisher publishes the ad; and the advertiser generates the ad and contracts with the advertising network to place it.

Publishers click on adverts posted on their own websites to make income, which results in click fraud under this architecture.

Click fraud is when an individual, computer program or generated script exploits online advertisers by repeatedly clicking on a PPC (pay-per-click) advertisement to generate fraudulent charges. Click fraud drives up advertising costs, lowers conversion rates and skews user data for online businesses.

It is important for a company to understand click fraud and safeguard its advertising budget. This budget helps in the sale of a product. Due to click fraud, the advertising budget gets wasted, not actually reaching potential customers and it can financially impact the company.

Click fraud can be determined through:

1) Continuous clicks from ISP servers that appear to be similar
2) Increase in costs not in line with previous campaigns or
3) Poor conversion rates
4) Other unexplained oddities.

Purpose of the project:

Fraud risk is everywhere, but for companies that advertise online, click fraud can happen at an overwhelming volume, resulting in misleading click data and wasted money. Ad channels can drive up costs by simply clicking on the ad at a large scale.With over 1 billion smart mobile devices in active use every month, companies which invest large shares digital marketing therefore suffers from huge volumes of fraudulent traffic.The objective is to flag IP addresses that click but don’t install the app, through the process of measuring a user’s click journey. The key finding we expect to see is to identify a click that primarily leads to app installation or if it causes dissipation.

Proposal:

Project title: Ad-tracking for conversion success – fraud detection

Dataset: Data of Talking Data firm

Objective: The main objective is to classify a particular transaction as fraudulent or genuine.

Challenges: The main challenge lies in misclassification of a transaction, especially tagging a genuine transaction as fraudulent. Thus, our challenge is to generate a model that provides high accuracy with the least number of genuine as fraudulent misclassifications.

With the given data, we are focusing on clicks that did not lead to app installation.

III. Data Description:

Target variable:

is_attributed - Indicates whether a Click lead to a Download or not(object or categorical -> 0 = Did not download; 1 = Downloaded)

Independent variables or predictors:

IP – IP address of click (object or categorical variable)

App – App id of marketing (object or categorical)

Device – Mobile phone type or device type (object or categorical)

OS – OS version of user mobile phone (object or categorical)

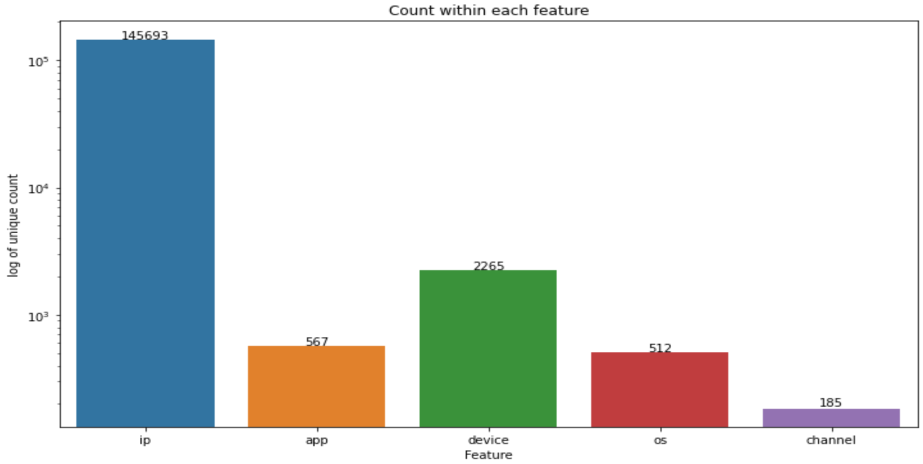
Channel – Channel ID of mobile ad publisher (object or categorical)

Click_time – Timestamp of click (date-time)

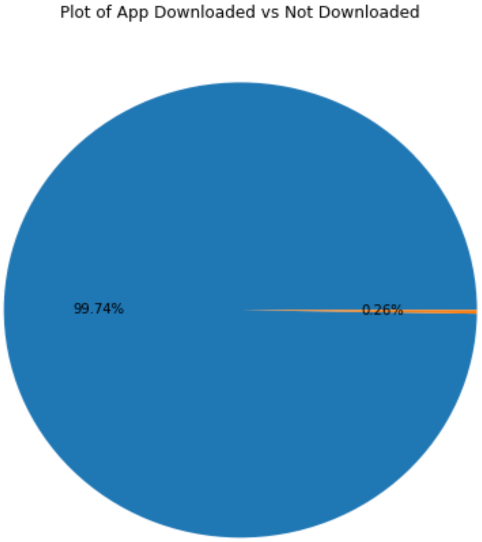
attributed_time – Time of app download (date-time)

EDA (Exploratory Data Analysis) :

Level Count for each Categorical Variables



Plot of App Downloaded vs Not Downloaded for Imbalanced data



Derived variables: A small sample of observations from the data (a few observations to demonstrate the above points).

Day - day on which the click was captured

Hour - captures hour of the day the click was executed

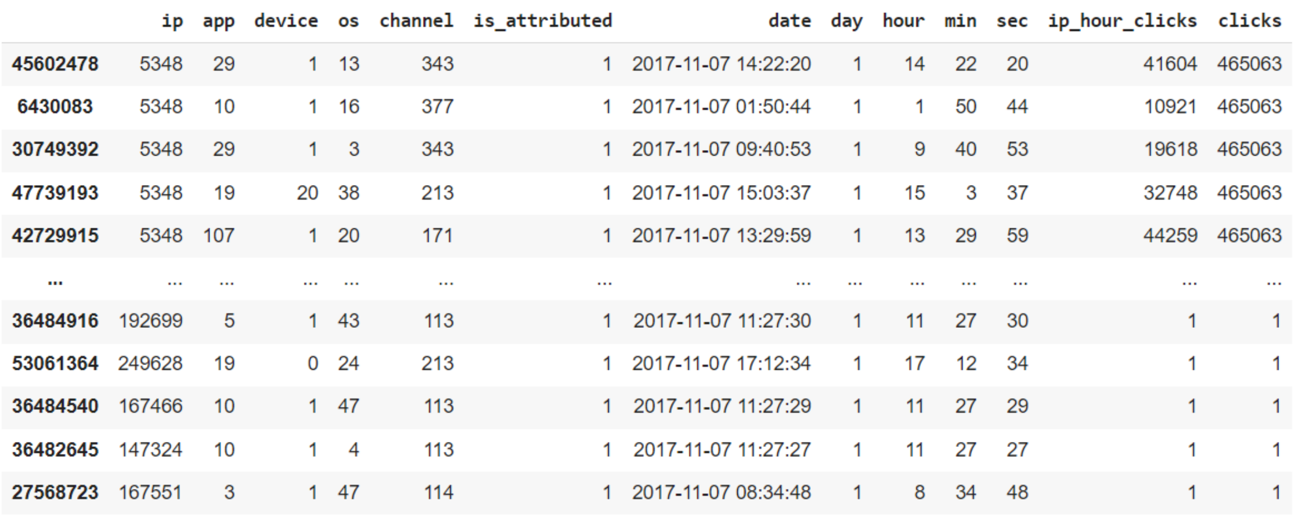
Min - captures minute on which the click was executed

Sec - captures seconds on which the click was executed

ip_hour_clicks - captures clicks generated by an IP in the respective Hour

clicks - Column captures the number of clicks generated by a particular IP in a day

Table below demonstrates the derived variables



	ip	app	device	os	channel	is_attributed	date	day	hour	min	sec	ip_hour_clicks	clicks
45602478	5348	29	1	13	343	1	2017-11-07 14:22:20	1	14	22	20	41604	465063
6430083	5348	10	1	16	377	1	2017-11-07 01:50:44	1	1	50	44	10921	465063
30749392	5348	29	1	3	343	1	2017-11-07 09:40:53	1	9	40	53	19618	465063
47739193	5348	19	20	38	213	1	2017-11-07 15:03:37	1	15	3	37	32748	465063
42729915	5348	107	1	20	171	1	2017-11-07 13:29:59	1	13	29	59	44259	465063
...	...	...	...	...	...	...	...	...	...	...	...	...	...
36484916	192699	5	1	43	113	1	2017-11-07 11:27:30	1	11	27	30	1	1
53061364	249628	19	0	24	213	1	2017-11-07 17:12:34	1	17	12	34	1	1
36484540	167466	10	1	47	113	1	2017-11-07 11:27:29	1	11	27	29	1	1
36482645	147324	10	1	4	113	1	2017-11-07 11:27:27	1	11	27	27	1	1
27568723	167551	3	1	47	114	1	2017-11-07 08:34:48	1	8	34	48	1	1

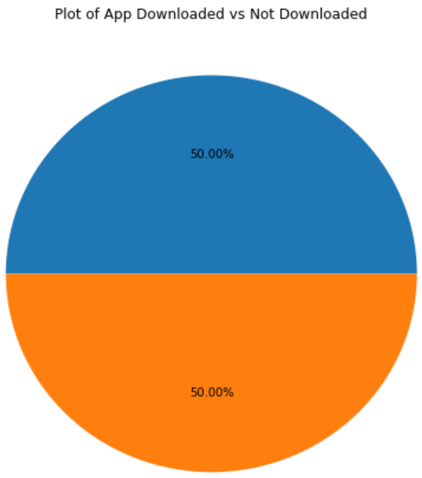
Sample size (*n*) and number of variables (*k*)

Sample size : 180M rows of observations spread across several days of data. Downsized to 2,000,000 rows consisting only of data of Date 7th. (Data downsized due to computational challenges).

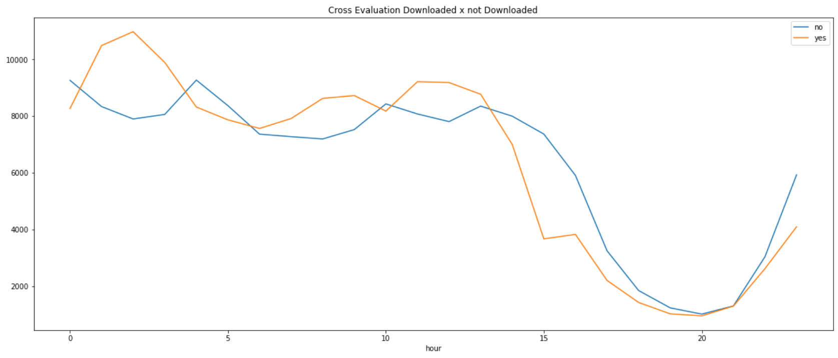
Downsampling:

Number of variables initially present in the downsized dataset are 7 independent variables.One other challenge that hindered the reliability of the model was the highly imbalanced data setIn order to tackle the problem we have downsampled the number of records of the failure scenario to match the number of success scenarios.

Plot of App Downloaded vs Not Downloaded after downsampling



Time Series Analysis for Cross Evaluation Downloaded vs Not Downloaded



Why the data are of interest.

This is an interesting project because this addresses a unique and a major concern in the digital marketing space and scope of improvisation. Also, tagging of fraudulent clicks requires good understanding of dependent variables and deep dive into the digital marketing industry. This is interesting because the data is humongous and real, and gives us an insight into what data and analysis could do to avoid or reduce problems in the world of marketing.

Datasource:

Data is picked from the source - Kaggle. It belongs to the Talking Data firm. The data has been provided here as a part of Kaggle challenge in the past, keeping it open for people to work on providing interesting solutions to prevent ad-click fraud. Choosing the data from this source has helped us explore several different machine learning models and apply them to the problem at hand. The problem and the dataset might have been present for a long time but the approach to solve this problem and focus on achieving higher accuracy & reducing false positives in misclassifications is our unique selling point and could set our solution apart from other solutions offered on Kaggle.

https://www.kaggle.com/competitions/talkingdata-adtracking-fraud-detection/data

III. Research Questions (1 page)

- What does the data indicate?
- Is there a pattern in the data with respect to individual independent variables that can directly influence app installations or conversion success?

  Ex: A certain hour in the day through a certain device and an IP address, maximum clicks have occurred but no app has been downloaded. This is a pattern to be looked for.

- Are there any specific Application IDs of marketing that have a maximum number of downloads or ‘no downloads’ (identified through ‘is_attributed’ variable)?
- Are there any specific Devices that have a pattern of maximum number of downloads or ‘no downloads’ (identified through ‘is_attributed’ variable)?
- Are there any ‘OS’ that have a pattern of maximum number of downloads or ‘no downloads’ (identified through ‘is_attributed’ variable)?
- Are there any ‘Channel IDs of mobile ad publishers’ that have a pattern of maximum number of downloads or ‘no downloads’ (identified through ‘is_attributed’ variable)?
- Is there any specific time in the day when the maximum number of ‘downloads’ or ‘no downloads’ happen? (identified through the ‘is_attributed’ variable)?
- Is there a combination of any of the above independent variables that helps in identifying a possible pattern of maximum downloads or ‘no downloads’ (identified through ‘is_attributed’ variable)?
- The key finding, we expect to see: If we can identify a click that primarily leads to app installation or if it causes dissipation.
- The data set originally had 180M observations, which observations should we downsize so that they form a good representation of the original data?
- The data comprises observations from several dates - which dates are to be considered while building a model?
- Is there any data transformation needed? For example: splitting the date column into date with day of the week, hour of the day etc through the date and time stamp.
- Would grouping the data by a certain value of variable or certain values of variables help us focus on the right part of data for the model, that reduces bias?
- What effect would downsampling the data have on bias?
- The inherent bias in the data would influence model prediction when using models like Naive Bayes. Should we balance the data set?
- What effect does balancing the dataset have on bias - variance?
- There is an additional column created to aggregate the sum of clicks that lead to values given in the ‘is_attributed’ column. Should this be considered before balancing the dataset or after balancing it?
- What prediction accuracy measures are to be used?

- Is there any improvement possible in the obtained accuracy so far?
- Of the data mining techniques used, which one works the best and why?
- With the prediction model that is built alone, is it possible to completely avoid fraud by flagging certain observation types?
- What domain knowledge is necessary in handling the ad-click fraud problem at hand?
- What additional information if present in data would help in much more accurate predictions?

IV. Methodology (1 page)

We have used the below methodology:

Selection of sample: Of the original dataset with 180M observations, the observations of the date 7 have the largest distribution that is close enough to represent the original data. Therefore, observations of Date 7 are selected.

Feature engineering: The variable date in itself is good to provide a good amount of insights if it can be split into day of the week, hour of the day, minute of the hour, etc. Hence, extraction of features from date & time stamps was performed.

Two new columns are created.One column depicts Number of clicks IP made in a day and the other column depicts Number of clicks IP made in an hour.. This is expected to lead to possible pattern discovery of maximum 'no downloads after clicks' occurring from certain IP addresses.

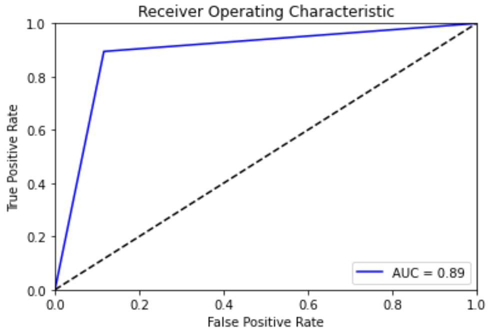
Downsampling: The original data has a larger number of observations with 'is_attributed' variable as 0, indicating that those are the observations for the app not downloaded, when compared to the observations with target variable as 1. Since the sample of interest with Date as 7th, is a good representation of the original, this carries the same bias inherently. To balance the dataset, we have randomly omitted observations that had 0's in the 'is_attributed' column.

Train - test split: Implemented train test split using train_test_split package of sklearn.model_selection

Data mining techniques:

Decision Tree: The basic idea behind any decision tree algorithm is to make the best split possible with the available feature that produces least misclassification errors. Next we make that attribute a decision node and break it into smaller subsets.Finally it starts tree building by repeating this recursive process for each branch until there is no repetition of instances.

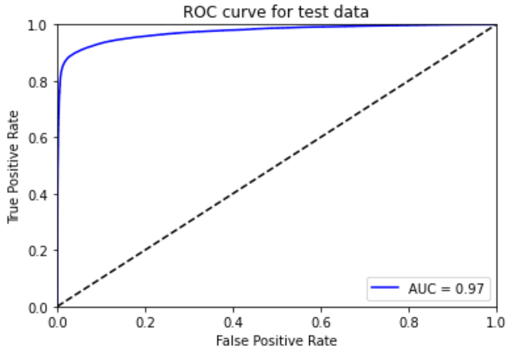
ROC Curve:



Accuracy: *88.94%*

Random Forests: In Random forest algorithm, we select random samples from a given dataset and then construct a decision tree for each sample and get a prediction result from each decision tree and perform a vote for each predicted result.Finally we choose the prediction result that has the more no of votes as the final result.

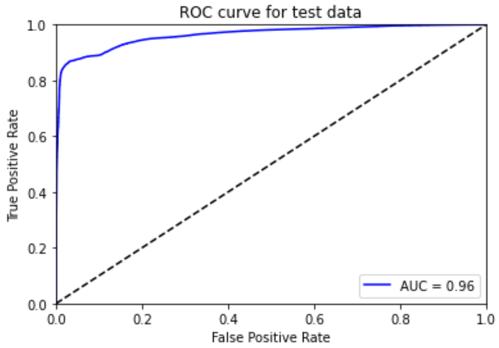
ROC Curve:



Accuracy : *92.92%*

<u>Ada-boost:</u> Adaptive Boosting (Adaboost) is a prominent boosting approach that merges numerous “weak classifiers” into a single “strong classifier”.

ROC Curve*:*



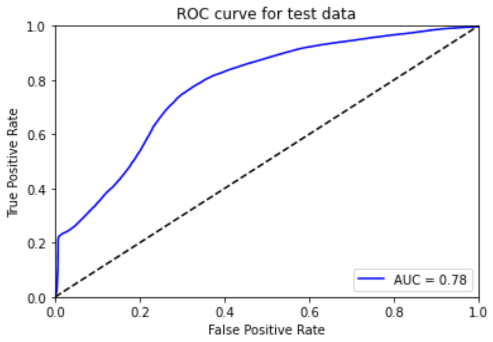
Accuracy *: 91.66%*

<u>KNN Classification:</u> Using this algorithm, a new data point is classified based on similarity in the specific group of neighboring data points. The algorithm calculates the distances between a specific data point in the set and any other K numbers of data points in the dataset that are near to it, then vote for the category with the highest frequency. Typically, Euclidean distance is used to calculate distance. As a result, the final model is just labeled data in a space.

Accuracy *: 87.32%*

<u>Logistic Regression:</u> This is used to predict the probability of a target variable. Helps in predicting the likelihood of an event happening or a choice being made. It has a binary outcome.

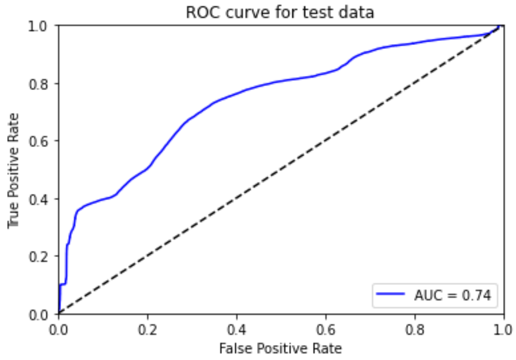
ROC Curve:



Accuracy: *72.31%*

Gaussian Naive Bayes: Gaussian Naive Bayes accepts continuous valued features and models them all as Gaussian (normal) distributions. To build a basic model, assume the data is characterized by a Gaussian distribution with no covariance (independent dimensions) between the parameters.

ROC Curve:



Accuracy: *61.52%*

Neural Network binary Classifier: Neural nets are a means of doing machine learning, in which a computer learns to perform some task by analyzing training examples. The Model is divided into 3 components Input Layer, hidden Layer and Output. These layers have multiple nodes based on the independent and dependent variables and are interconnected with weighted neurons.

Accuracy*: 92.16 %*

V. Results and inferences:

Models built are predictive in nature. Several models were explored, starting with decision tree, logistic regression and Naive Bayes.KNN Observing the prediction accuracies being low, ensemble methods were explored, as they combine weak learners to turn them into a stronger learner. The ensemble methods are Random Forest, Adaptive Boosting(Adaboost).

Based on the given values in the observations, our predictive model goes through iterations of learning to figure out the pattern of when a click leads to app-download (is_attributed = 1) and when a click does not lead to app-download (is_attributed = 0). This helped in accomplishing the goal of our study - for the model to be able to predict the classification of the click (does it lead to a download or not => target variable is_attributed resulting in 1 or 0, respectively) based on the values of independent variables in a given set of observations. Thereby, the model helps differentiate a genuine click from a fraudulent click.

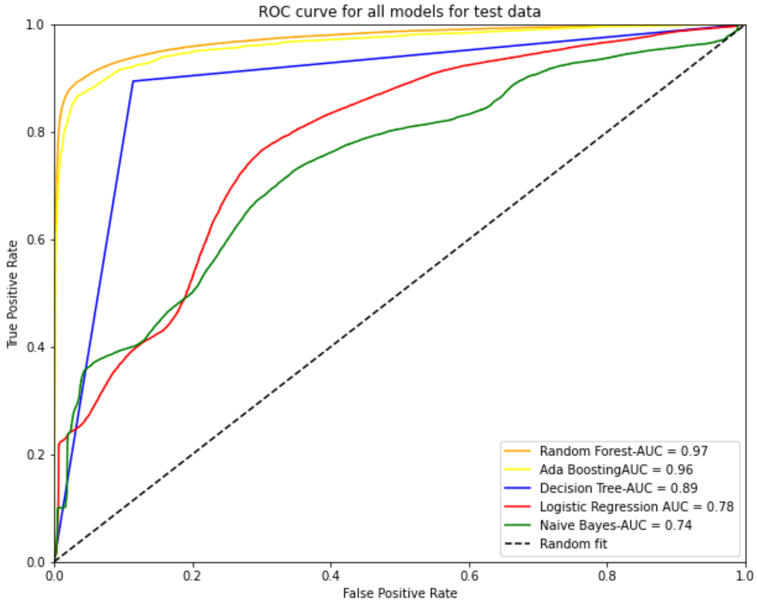
By consolidating the accuracy measures of the tested predictive models, we identified **Random Forest** to be possessing the highest predictive accuracy of all.

Model Metrics:

Model Type	***Accuracy***	***Percentage of False negatives in Misclassification***	***Percentage of False positives in Misclassification***	***Sensitivity***	***Specificity***
Decision Tree	*88.94%*	*47.7%*	*52.3%*	*0.894*	*0.883*
Random Forests	*92.92%*	*75.9%*	*24.1%*	*0.893*	*0.965*
Adaptive Boosting	*91.66%*	*79.1%*	*20.9%*	*0.868*	*0.965*
Logistic Regression	*72.31%*	*47.3%*	*52.7%*	*0.738*	*0.707*
Gaussian Naive Bayes	*61.52%*	*21.3%*	*78.7%*	*0.835*	*0.392*

KNN Classification	*87.32%*	*77.8%*	*22.2%*	*0.798*	*0.942*
Neural network Binary classifier	*92.16%*	*70.3%*	*29.7%*	*0.888*	*0.952*

Plot depicts the Test data ROC curves for all the Models Implemented



VI. Conclusion:

We have implemented a model which can predict the outcome of an ad click based on IP, app, device, OS and channel. In order to equip the model with the capability to detect fraudulent clicks from a certain IP, we have created calculated fields like clicks and ip_hour_clicks.This as a result induces a sense on dependency on the complete data.However, this might not pinpoint a particular IP, the features provide the general sense of flagging such concurrent request as fraudulent.

In addition, the media where the ad campaign is launched can be limited in scope, if some media are not helping in achieving desired conversion rate of click to download, due to more frequent fraudulent clicks in there. The Talking data firm or any firm that engages in ad marketing to target increase in sales, should incorporate the following principles:

1) Keep close tracking of the advertising budget - its flow and returns. Sudden and unexplained surges could be due to click fraud
2) Keep an eye out for poor conversion rates
3) Watch out for costs incurred on a certain campaign which are not in line with previous campaigns.

How to prevent click fraud:

1) Business must advertise on quality
2) Advertise on well-known sites
3) Block certain IP addresses that have such a history
4) Use software that prevents fraud
5) Constantly monitor logs & budgets