**Ramaiah Institute of Technology**

(An Autonomous Institute, Affiliated to VTU)

MSR Nagar, MSRIT post, Bangalore-54

A Dissertation Report on

**Click-Through Rate Prediction by**

**Naïve Bayers**

Submitted by

**Antariksh Guha 1MS15CS022**

**Ashutosh Kumar 1MS15CS029**

**Dev Sood 1MS15CS039**

**Darshil Vala 1MS15CS038**

**Binit Agarwal 1MS15CS033**

*Bachelor of Engineering in Computer Science & Engineering*

Under the guidance of:  **Sowmya B.J**

**Assistant Professor**

**Computer Science**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**M.S.RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute, Affiliated to VTU)**

**BANGALORE-560054 www.msrit.edu, 2017**

**Introduction:**

**Click-through rate** (**CTR**) is the ratio of users who click on a specific link to the number of total users who view a page, email, or advertisement. It is commonly used to measure the success of an online advertising campaign for a particular website as well as the effectiveness of email campaigns.

Click-through rates for ad campaigns vary tremendously. The very first online display ad shown for AT&T on the website HotWired in 1994, had a 44% click-through rate. Over time the overall rate users click on webpage banner ads has decreased.

## Purpose

The purpose of click-through rates is to measure the ratio of clicks to impressions of an online ad or email marketing campaign. Generally the higher the CTR the more effective the marketing campaign has been at bringing people to a website. Most commercial websites are designed to elicit some sort of action, whether it be to buy a book, read a news article, watch a music video, or search for a flight. People rarely visit websites with the intention of viewing advertisements, in the same way that few people watch television to view the commercials.

While marketers want to know the reaction of the web visitor, with current technology it is nearly impossible to quantify the emotional reaction to the site and the effect of that site on the firm's brand. However, click-through rate is an easy piece of data to acquire. The click-through rate measures the proportion of visitors who initiated an advertisement that redirected them to another page where they might purchase an item or learn more about a product or service. Forms of interaction with advertisements other than clicking is possible, but rare; "click-through rate" is the most commonly used term to describe the efficacy of an advert.

## Construction

The click-through rate is the number of times a click is made on the advertisement divided by the total impressions.

CTR = Number of click-throughs Number of impressions × 100 ( % ) {\displaystyle {\text{CTR}}={{\text{Number of click-throughs}} \over {\text{Number of impressions}}}\times 100(\%)} C:\Users\Anurag\AppData\Local\Microsoft\Windows\INetCache\Content.Word\cap1.png

### Online advertising CTR

The click-through rate of an advertisement is defined as the number of clicks on an ad divided by the number of times the ad is shown (impressions), expressed as a percentage. For example, if a banner ad is delivered 100 times (100 impressions) and receives one click, then the click-through rate for the advertisement would be 1%.

Click-through rates for banner ads have decreased over time. When banner ads first started to appear, it was not uncommon to have rates above five percent. They have fallen since then, currently averaging closer to 0.2 or 0.3 percent. In most cases, a 2% click-through rate would be considered very successful, though the exact number is hotly debated and would vary depending on the situation. The average click-through rate of 3% in the 1990s declined to 2.4%–0.4% by 2002. Since advertisers typically pay more for a high click-through rate, getting many click-throughs with few purchases is undesirable to advertisers. Similarly, by selecting an appropriate advertising site with high affinity (e.g., a movie magazine for a movie advertisement), the same banner can achieve a substantially higher CTR. Though personalized ads, unusual formats, and more obtrusive ads typically result in higher click-through rates than standard banner ads, overly intrusive ads are often avoided by viewers.

Modern online advertising has moved beyond just using banner ads. Popular search engines allow advertisers to display ads in with the search results triggered by a search user. These ads are usually in text format and may include additional links and information like phone numbers, addresses and specific product pages. This additional information moves away from the poor user experience that can be created from intrusive banner ads and provides useful information to the search user, resulting in higher Click-through rates for this format of pay-per-click Advertising. Having high click-through rate isn't the only goal for an online advertiser who will occasionally develop campaigns to raise awareness and sacrifice click-through rate for the overall gain of valuable traffic.

### Estimating the Click-Through Rate for Ads

Search engine advertising has become a significant element of the Web browsing experience. Choosing the right ads for the query and the order in which they are displayed greatly affects the probability that a user will see and click on each ad. This ranking has a strong impact on the revenue the search engine receives from the ads. Further, showing the user an ad that they prefer to click on improves user satisfaction. For these reasons, there is an increasing interest in accurately estimating the click-through rate of ads in a recommender system.

**Implementation**

So, in this project we use Naïve Bayers theorem to predict whether for a given row of attributes of the database the ad on the website is clicked or not. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

**Data Set Description:**

We have taken the click on ads on ads by a company named Avaru for a span of 10 days.

**Source of Dataset:** <https://www.kaggle.com/c/avazu-ctr-prediction/data>

**Attributes Description:**

Id: The ID of the various click that take place on the ads posted by the company. All the rows have different values of ID for the different clicks on the ads. The ID of the clicks have also the values in scientific notation as the transaction IDs are quite long.

Hour: The time of the click on an ad is noted in this attribute. The time for many attributes is the same as a lot of ads are clicked at a given specific time.

C1: The attribute C1 has been dropped in the final calculation of the click of the ads as the value remains as 1005 for many cases and hence it does not have quite a difference in the calculation of probability.

Banner\_pos: The attribute banner\_pos is quite useful to determine the position of the banner in the webpage or the mobile device.

The position has been given as 0 or 1 or 7. 0 can be taken as ad on the top left end and 1 as top left end and 7 as down left or right end of placement of the ad in the screen.

Site\_id: The site id of the various websites where the ads have been posted are done is taken. The site id have been given as random number or alphabets of strings.

Site\_domain: The site domain of the various websites where the ads have been posted are done is taken. The site domain have been given as random number or alphabets of strings.

Site\_category: The site category of the various websites where the ads have been posted are done is taken. The site category have been given as random number or alphabets of strings. The site category can be assumed as anything like shopping or anything of the like.

App\_id: The app id of the various websites where the ads have been posted are done is taken. The app id have been given as random number or alphabets of strings.

App\_domain: The app domain of the various websites where the ads have been posted are done is taken. The app domain have been given as random number or alphabets of strings.

App\_category: The app category of the various websites where the ads have been posted are done is taken. The app id have been given as random number or alphabets of strings. The app category can be assumed as anything like shopping or anything of the like.

Device\_id: The device ID from which the ad has been clicked is taken in this attribute. Different devices have different device id so we get many device ips.

Device\_model: The device model number from which the ad has been clicked is taken in this attribute. Different devices have different device models so we get many device model numbers.

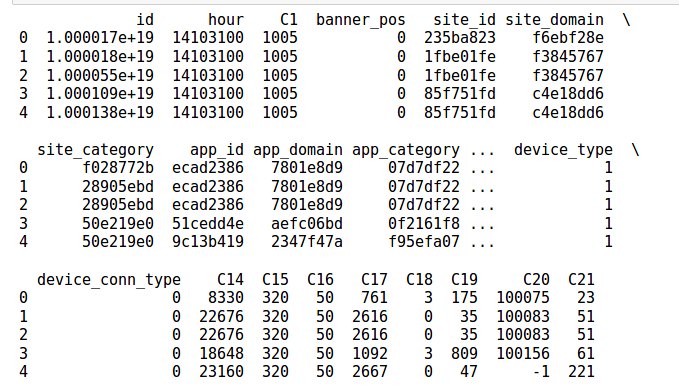
Device\_types: The device type number from which the ad has been clicked is taken in this attribute. Different devices have different device types so we get many device type numbers. The device types can be considered as tablets or PCs or mobile phones.

Device\_conn\_types: The connection types of a device can be of different types like LAN connection, or mobile data or dial up, or broadband. So we can find people clicking on the ads using which kind of connection.

Attributes C14-C21: These different attributes have been kept hidden by the company due to privacy issues. So these can as assumed as regions, countries and the like.

Data set size: 6,311,147,778 bytes (5.88GB)

Number of tuples: 1,37,30,132

****

**Algorithm Description:**

## What is Naive Bayes algorithm?

It is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



Above,

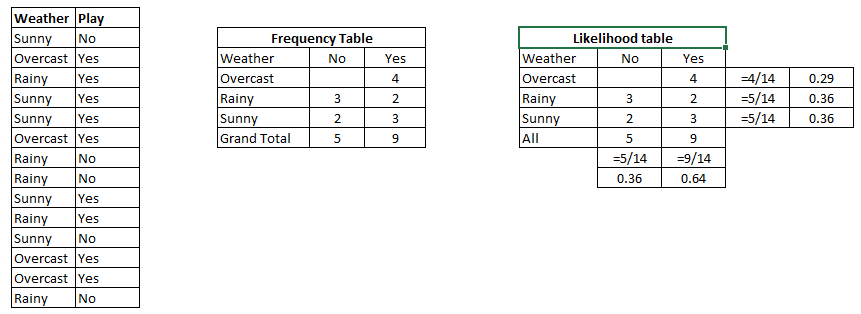
* *P*(*c|x*) is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
* *P*(*c*) is the prior probability of *class*.
* *P*(*x|c*) is the likelihood which is the probability of *predictor* given *class*.
* *P*(*x*) is the prior probability of *predictor*.

## How Naive Bayes algorithm works?

Let’s understand it using an example. Below I have a training data set of website category and corresponding target variable ‘Click’ (suggesting possibilities of clicking on an ad). Now, we need to classify whether users will click or not based on various conditions. Let’s follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like educational probability = 0.29 and probability of clicking is 0.64.



|  |  |
| --- | --- |
| Website category | Click |
| Commercial | No |
| Educational | Yes |
| Banking | Yes |
| Commercial | Yes |
| Commercial | Yes |
| Educational | Yes |
| Banking | No |
| Banking | No |
| Commercial | Yes |
| Banking | Yes |
| Commercial | No |
| Educational | Yes |
| Educational | Yes |
| Banking | No |

|  |  |  |
| --- | --- | --- |
|  | Frequency Table |  |
| Click | No | Yes |
| Educational |  | 4 |
| Banking | 3 | 2 |
| Commercial | 2 | 3 |
| Grand Total | 5 | 9 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Click Likelihood Table |  |  |  |
| Website category | No | Yes |  |  |
| Educational |  | 4 | 4/14 | .29 |
| Banking | 3 | 2 | 5/14 | .29 |
| Commercial | 2 | 3 | 5/14 | .36 |
| All | 5 | 9 |  |  |
|  | 5/14 | 9/14 |  |  |
|  |  |  |  |  |

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

**Problem: User**s will click if category is commercial. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

P(Yes | Commercial) = P( Commercial | Yes) \* P(Yes) / P (Commercial)

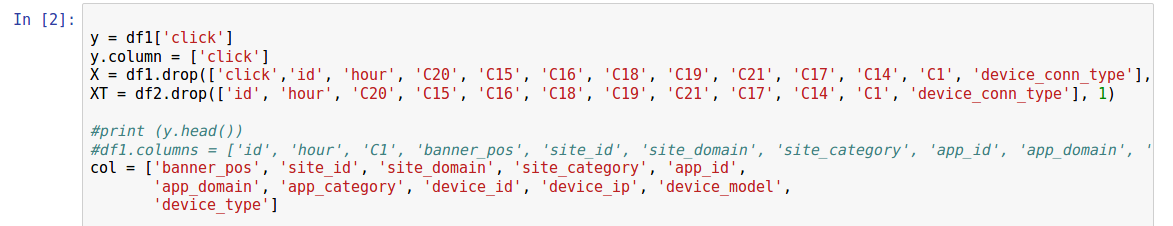
Here we have P (Commercial |Yes) = 3/9 = 0.33, P(Commercial) = 5/14 = 0.36, P( Yes)= 9/14 = 0.64

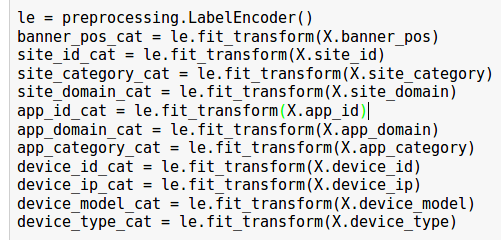
Now, P (Yes | Commercial) = 0.33 \* 0.64 / 0.36 = 0.60, which has higher probability.

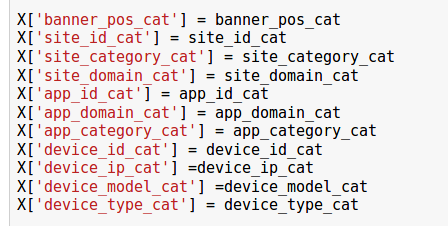
Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

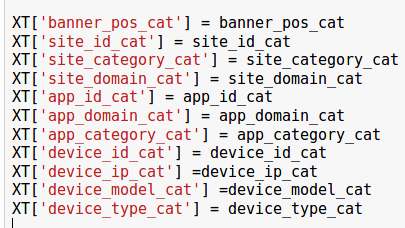
Snapshots:

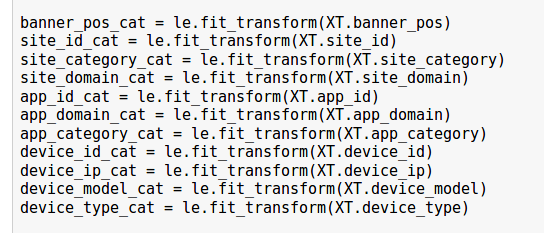
Algorithm code:

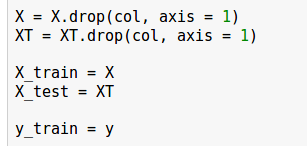


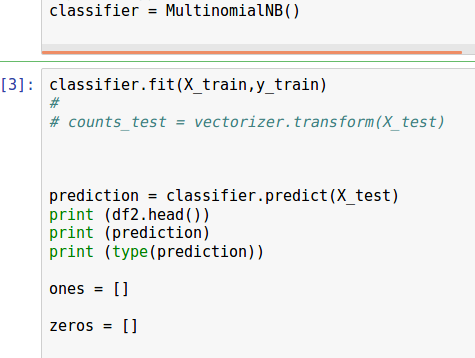


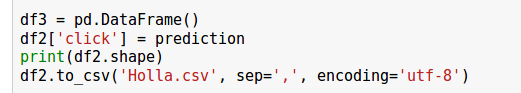










****

**Implementation of Multinomial NB on Click-Through Rate dataset**

### Importing Python Machine Learning Libraries

We need to import pandas, numpy and sklearn libraries.

If you are not setup the python machine learning libraries setup. You can first complete it to run the codes in this articles.

### 

### Data Import

For importing the click data, we are using pandas read\_csv() method. This method is a very simple and fast method for importing data.

The delimiter parameter is for giving the information the delimiter that is separating the data. Here, we are using ‘ \*, \*’ delimiter. This delimiter is to show delete the spaces before and after the data values. This is very helpful when there is inconsistency in spaces used with data values.



### Data preprocessing



Statistics about the dataset after running the above command. You can spend some time here to get in details about each and ever stats provided.

### 

### 

### 

### One-Hot Encoder

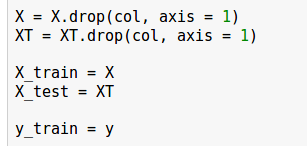
For implementing this, we are going to use LabelEncoder of scikit learn library. For encoding, we can also use the One-Hot encoder. It encodes the data into binary format.

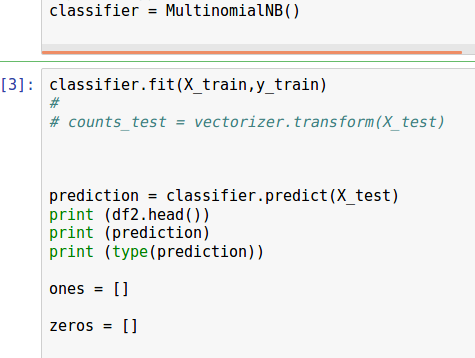
### 

### 

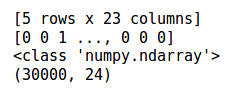
## Data Slicing

Let’s split the data into training and test set. We can easily perform this step using sklearn’s train\_test\_split() method.



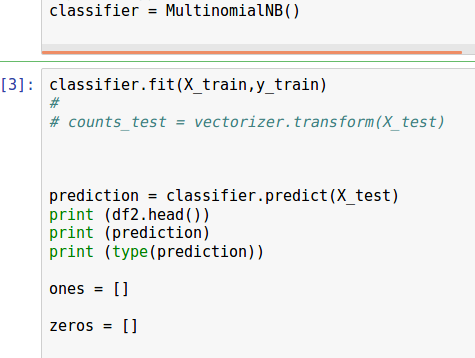


The features\_train & target\_train consists of training data and the features\_test & target\_test consists of testing data.



### Multinomial Naive Bayes Implementation

After completing the data preprocessing. it’s time to implement machine learning algorithm on it. We are going to use sklearn’s MultinomialNB module.



We have built a MultinomialNB classifier. The classifier is trained using training data. We can use fit() method for training it. After building a classifier, our model is ready to make predictions. We can use predict() method with test set features as its parameters.

## What are the Pros and Cons of Naive Bayes?

***Pros:***

* It is easy and fast to predict class of test data set. It also perform well in multi class prediction
* When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
* It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

***Cons:***

* If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
* On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict\_proba are not to be taken too seriously.
* Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

## 4 Applications of Naive Bayes Algorithms

* **Real time Prediction:**Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
* **Multi class Prediction:**This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
* **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
* **Recommendation System:**Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

## How to build a basic model using Naive Bayes in Python?

Again, scikit learn (python library) will help here to build a Naive Bayes model in Python. There are three types of Naive Bayes model under scikit learn library:

* **Gaussian:**It is used in classification and it assumes that features follow a normal distribution.
* **Multinomial:**It is used for discrete counts. For example, let’s say,  we have a text classification problem. Here we can consider bernoulli trials which is one step further and instead of “word occurring in the document”, we have “count how often word occurs in the document”, you can think of it as “number of times outcome number x\_i is observed over the n trials”.
* **Bernoulli:**The binomial model is useful if your feature vectors are binary (i.e. zeros and ones). One application would be text classification with ‘bag of words’ model where the 1s & 0s are “word occurs in the document” and “word does not occur in the document” respectively.

Based on your data set, you can choose any of above discussed model. Below is the example of Gaussian model. In this we have chosen Multinomial as it works better with text and Gaussian works better with numerical values.

Above, we looked at the basic Naive Bayes model, you can improve the power of this basic model by tuning parameters and handle assumption intelligently. Let’s look at the methods to improve the performance of Naive Bayes Model. We recommend you to go through this document for more details on Text classification using Naive Bayes.

## Tips to improve the power of Naive Bayes Model

Here are some tips for improving power of Naive Bayes Model:

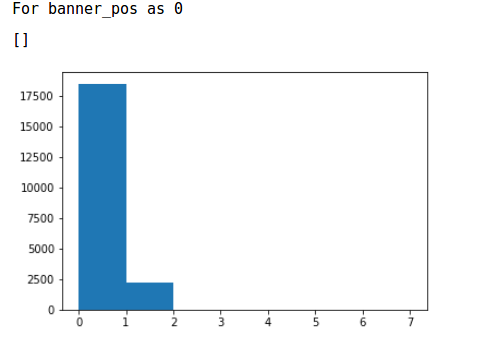
* If continuous features do not have normal distribution, we should use transformation or different methods to convert it in normal distribution.
* If test data set has zero frequency issue, apply smoothing techniques “Laplace Correction” to predict the class of test data set.
* Remove correlated features, as the highly correlated features are voted twice in the model and it can lead to over inflating importance.
* Naive Bayes classifiers has limited options for parameter tuning like alpha=1 for smoothing, fit\_prior=[True|False] to learn class prior probabilities or not and some other options. I would recommend to focus on your  pre-processing of data and the feature selection.
* You might think to apply some classifier combination technique like resembling, bagging and boosting but these methods would not help. Actually, “resembling, boosting, bagging” won’t help since their purpose is to reduce variance. Naive Bayes has no variance to minimize.

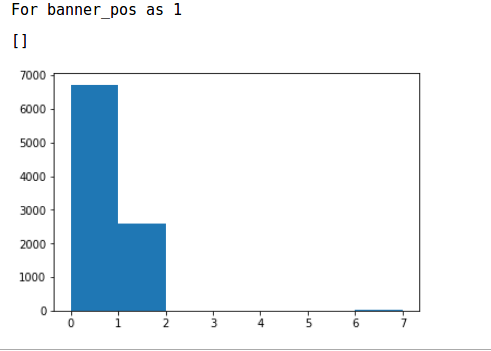
**Social impact:**

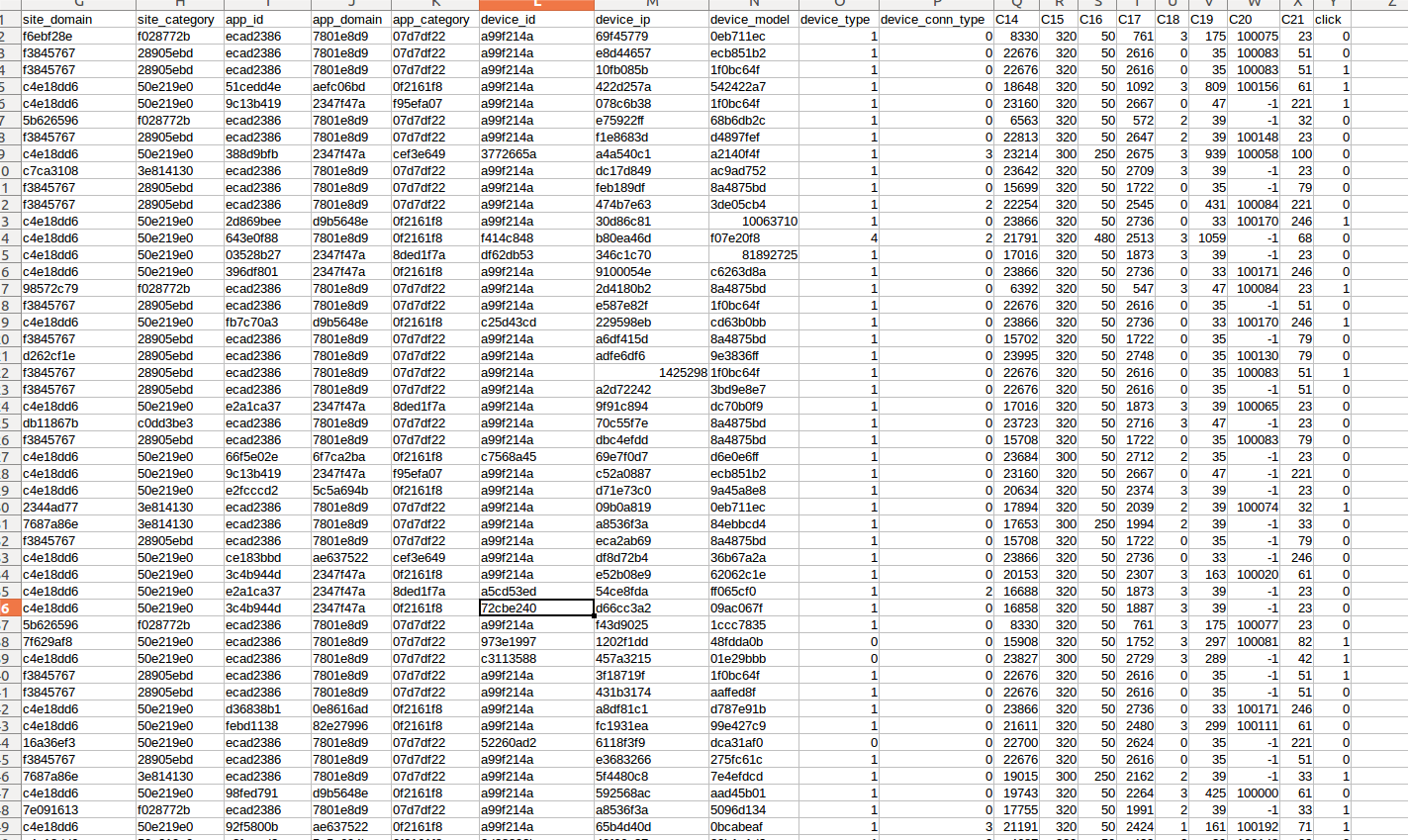
The social impact of click-through rate prediction is great. The economy is vastly dependent on social media and people have taken a liking to it. Imagine going through a day of you life without social media. The whole social media sector has been monetized and people or companies earn a lot through this, thus it is changing ways of life.

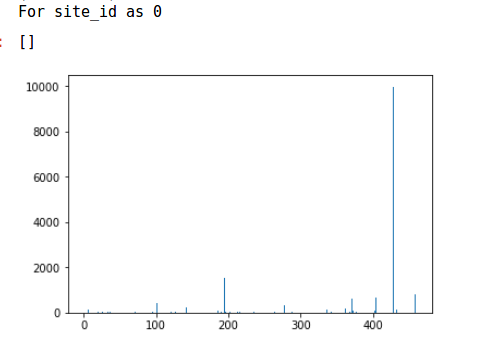
Various social cause organizations like NGOs, orphanages and others have started posting ads for people to donate on online websites. Posting ads costs money and thus it must be spent wisely. They should know when to post a ads, the type of website where it should be posted and the region from where they can get the most donations. This can be used to change lives of the unfortunate many indirectly.

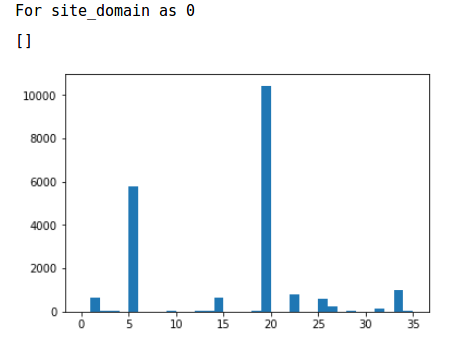
**Graphs and inferences:**

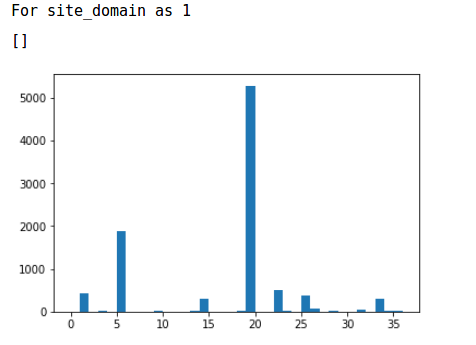


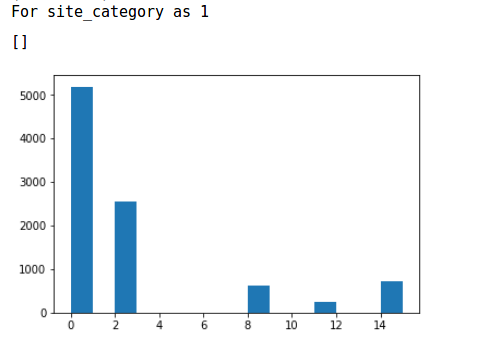


From the above two graphs conclusion can be drawn that the banner\_pos with the banner\_pos value 0 has higher frequency for click 0 hence while prediction the value of the test dataset we can see the prediction for banner\_pos value as 0 is also higher as the data model has been trained with value which have 0 for banner\_pos more, so the banner\_pos with value 0 will lean more towards no click.

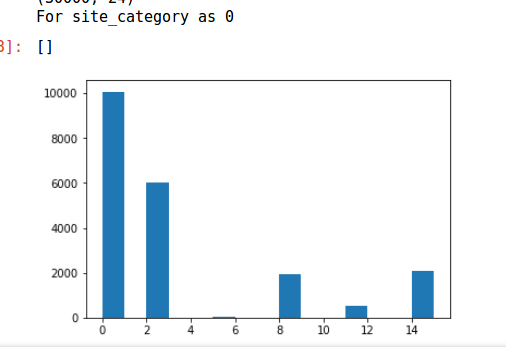
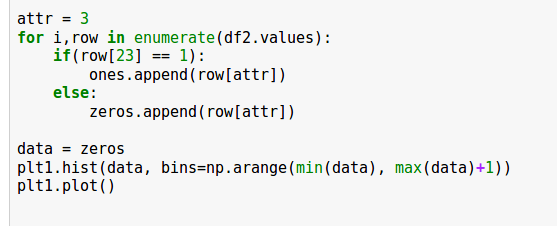






From the above two graphs conclusion can be drawn that the site\_domain with the 19-20 id value has higher frequency for click as 0 hence while prediction the value of the test dataset we can see the prediction for site\_domain value as 19-20 is also higher as the data model has been trained with value which have no click for site\_domain more. Thus site\_domain with 19-20 has mostly no click.

From the above two graphs conclusion can be drawn that the site\_category with the 0-1 id value has higher frequency for click as 0 hence while prediction the value of the test dataset we can see the prediction for site\_category value as 0-1 is also higher as the data model has been trained with value which have no click for site\_category more. Thus site\_category with 0-1 has mostly no click.



The graph plotting code has been given below.(using python library matplotlib).