Independent Study Under Assistant Professor Christoph Riedl

Name : Avirat Gaikwad
Project : Click Through Rate Prediction
Submission Date : 12/11/2017

Programmatic Advertising - CTR Prediction

Introduction:

The Digital Marketing industry has grown exponentially over the years. Earlier, Web Publishers monopolized the online market. The selling of space and ad-banners to the highest bidders led to low selling rate for publishers because of high cost of Digital Advertising. Eventually, Web -publishers decided to take themselves out of the direct sales process. This marked the birth of new market dynamics i.e. Ad Network. The Ad-Network would sell the remaining inventory to the buyers for fraction of the previous cost. Though, the Digital Advertising Industry grew, the click rates continuously dropped down. It became obvious to a lot of companies that the current methodology wasn't the best option for growing sales or attracting new consumers.

Simultaneously, the use of Predictive analytics in varied sectors increased manifold. Advertisers started targeting the people who were most likely to buy the product/service using analytics. From this point forward,[1]Digital Advertising transformed and Programmatic Advertising was born – Connecting the right ads to the right people.

The CRISP-DM Methodology provides a structured approach to planning a data mining project. We will use this methodology in the project:

Stage 1: Determine Business Objectives

The first stage of the CRISP-DM process is to understand what you want to accomplish from a business perspective. The goal of this stage of the process is to uncover important factors that could influence the outcome of the project. Neglecting this step can mean that a great deal of effort is put into producing the right answers to the wrong questions. The main objective of the project is to determine the variables affecting clicks in Online Advertising. This project would answer the following questions:

- Which variables affect clicks in Online advertising?
- Strategies to be developed to increase number of clicks
- Should advertisements be targeted depending on time of the day
- Which different industries can benefit from this project

Business Problem:

Click through rate(CTR) is one of the important metrics to be considered during the Online Marketing Strategy. Click through rate can be calculated as the number of times an Ad has been clicked divided by the total number of impressions. Though CTR does not guarantee a conversion it gives you a base of visitors who will potentially convert. Moreover, [2]CTR

affects Quality Score(QS) which in turn affects the Return on Investment(ROI). Now that we have established why CTR is important, let us focus on the problem.

The Business Problem here is to predict the CTR of the ads. We have a Dataset containing week's worth of data provided by Avazu. Depending on the CTR, we plan to devise Marketing Strategies which can increase the company's revenue through Online Marketing. The main objective of the project is to determine the CTR using Machine Learning techniques.

Stage 2: Data Understanding

-Initial Data Collection Report

Data Source:

Avazu has released its weekly data as a part of Kaggle Research Competition to search for the most accurate Machine Learning(ML) Algorithm for CTR estimation. [11]The link to access the dataset can be found in the references section.

-Dataset Construction:

The dataset includes the following columns which are to be used for analysis:

- id: ad identifier
- click: 0/1 for non-click/click
- hour: format is YYMMDDHH, so 14091123 means 23:00 on Sept. 11, 2014 UTC.
- C1 -- anonymized categorical variable
- banner_pos
- site_id
- site_domain
- site_category
- app_id
- app_domain
- app_category
- device_id
- device_ip
- device_model
- device_type
- device_conn_type
- C14-C21--anonymized

-Data Description Report

Dependent variable:

Click: 0/1 for non-click/click

Independent Variable:

Hour: format is YYMMDDHH, so 14091123 means 23:00 on Sept. 11, 2014 UTC

Banner Position: Banner_pos

Website information (Encrypted): site_id, site_domain, site_category

Application information (Encrypted): app_id, app_domain, app_category
device_id, device_ip, device_model, device_type, device_conn_type

Stage 3: Data Preparation:

Preliminary Work:

The dataset being huge i.e. approximately 1.04GB of Advertising data containing 24 variables and 40428967 rows, I'm still on the first 2 steps in the [3]Cross Industry Process for Data Mining(CRISP DM). Currently, I'm trying to make sense of the data which is step two in the methodology. In order to load the dataset, a connection to the file has to be opened because of the large nature of the dataset.

Then, we have extracted a small subset of the entire dataset in order to fit the models in a smaller time period. Extracted 1,000,000 rows from the actual dataset which will be further split up into Train and Test dataset for analysis.

Planned Analyses:

The nature of the data provided in most datasets is dense. Prior to evaluating the dataset, evaluating which model to be used would be a wrong presumption. A lot of people have been using Logistic Regression for predicting if the Ad would get clicked or not. Also, SVM Model has been developed to classify the clicks. However, SVM Model takes a lot of time to fit over the data provided which is why we will further create a subset of the training data.

This subset will be used to fit the SVM Model. Finally test set will be predicted using the model created on the subset.

Progress Report: Data Cleaning Report

Generated columns:

The Hour variable is in "YYMMDDHH" format which cannot be used for further analysis. Using "strptime" function we change the column to "YYMMDD HHMMSS" format. The next step is to split the columns to get two different columns namely "Date" and "Time" which are generated and added to the dataset. Using weekdays function we convert our date into day of the week for eg: 2014-10-28 is stated as "Tuesday".

In order to interpret the change in variables like banner position, device connection type, device type and C1 as moving down the ranks or reference level, we use relevel function. For eg: we change the reference level of C1 Variable to "1005"

The time has been converted into hourly format using hour function. For simplicity, we also split the entire 24 hours into 4 time-slots i.e. Morning, Afternoon, Evening and Night.

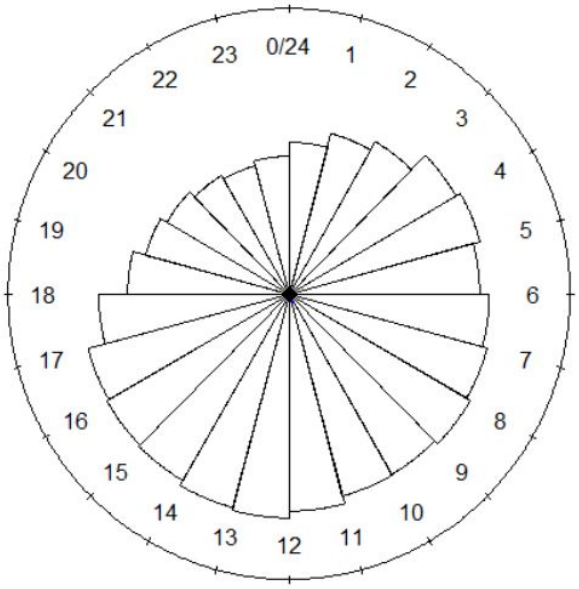
In this case, since a new level has been added to the test set which is not initially present in the train set. On training the model and using predict function, we would eventually get an error. In order to solve this error, we add a row from the test set into training set.

Data Visualization:

Number of clicks/hour

It would be easier to visualize the number of clicks during different times of the day through a circular clock shaped visualization. Here, I have developed a clock structured visualization to depict the number of click = 1 during the entire 24 hour period. We use the package "circular" to develop the visualization i.e. the [4]rose diagram.

The visualization can be seen below:



From the visualization, it can be seen that the maximum number of clicks occur during Morning and Afternoon. This gives, a general idea about the time to focus advertisements to targeted customers to maximize the number of clicks.

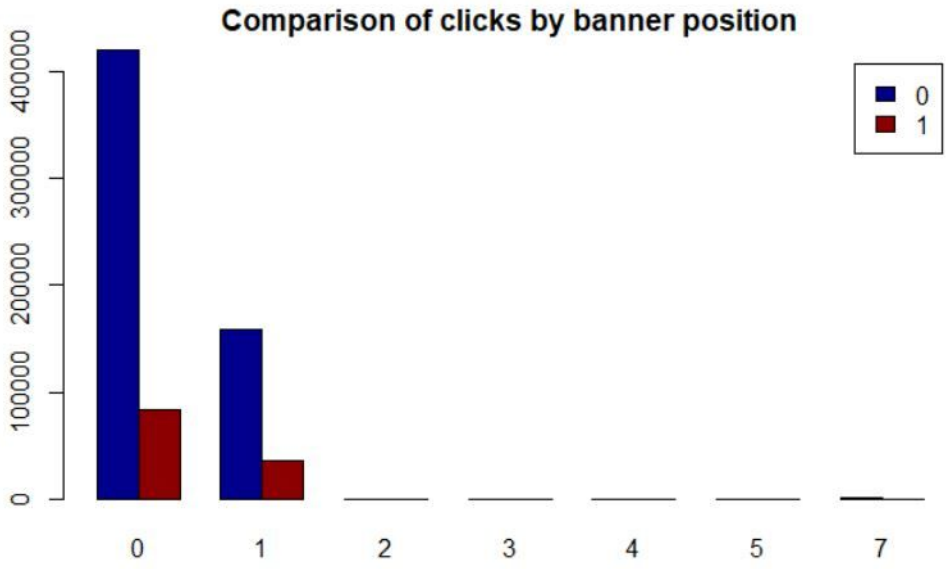
Number of clicks v/s Banner Position

On Comparing, Banner Position with the number of clicks, we can see that 3 banner positions are extremely important. The Banner positions can be seen as below:

1) Banner Position 0
2) Banner Position 1
3) Banner Position 7

The percentage data of clicks is highest in Banner Position 0 in comparison to other banner positions

Percentage for click = 1 for Banner position 0 = 82879/503632 = 0.16456*100 = 16.456%



	0	1	2	3	4	5	7
0	420753	159127	220	25	99	88	521
1	82879	35989	31	7	17	8	237

We can infer from the data tabulated below:

The number of clicks = 0 or click = 1 depending on the banner position on the webpage.

The maximum percentage click = 1 is at Banner position 7

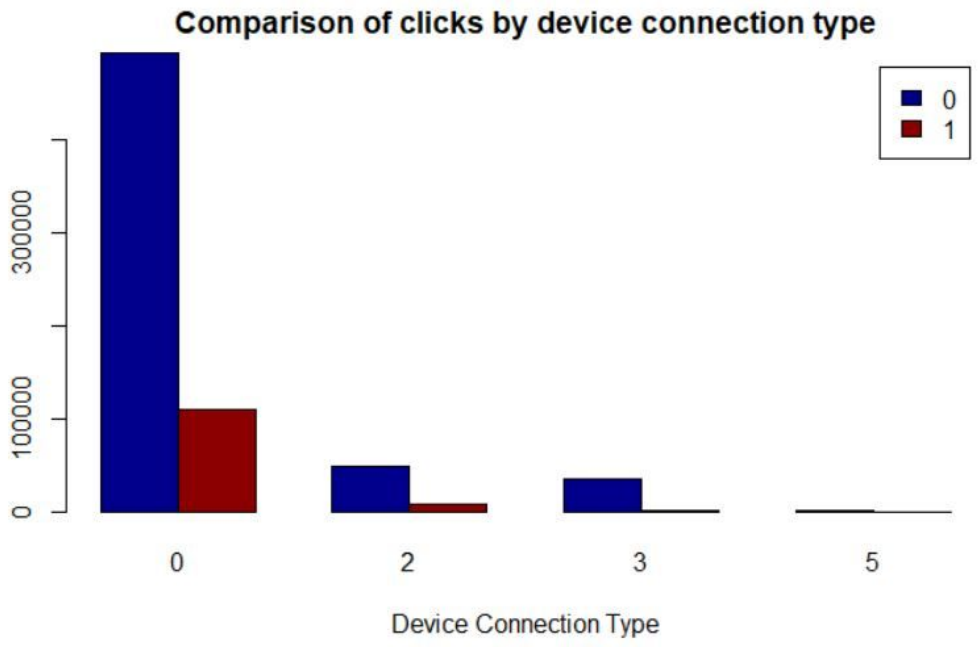
Percentage for click = 1 for Banner position 7 = 237/758 = 0.3122 = 31.22%

Thus, Banner position 7 is most important and should be used in Targeted Marketing.

Number of clicks v/s Device Connection Type

We have maximum data for Device Connection Type 0. Also, from the data it can be inferred that, maximum number of click = 1 are from Device Connection Type 0

Percentage for click = 1 for Device Connection Type 0
=109704/(109704+494588) =0.18154*100 =18.154%



	0	2	3	5
0	494588	49584	35991	670
1	109704	7708	1738	18

Thus, among the people who connect using Device Connection Type 0, 18.154% people tend to click on the advertisement.

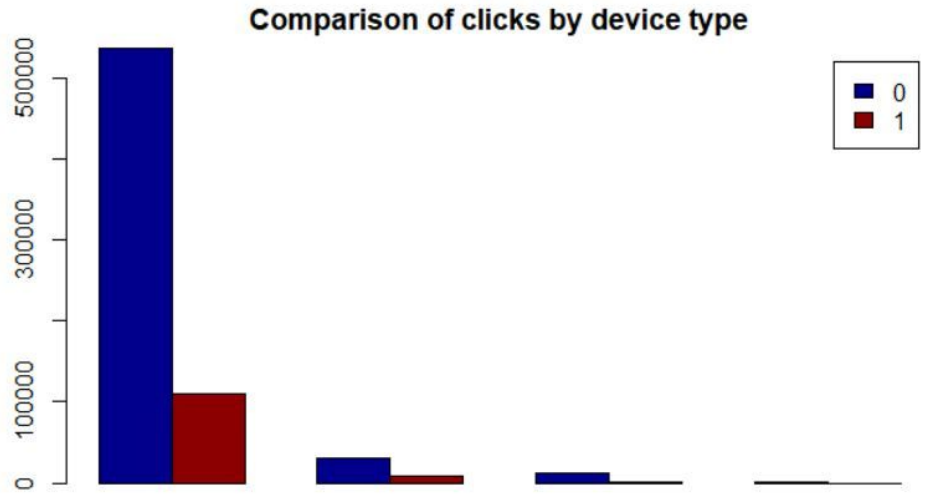
Number of clicks v/s Device Type

The visualization below depicts the variation of the number of clicks against the device type.

Here, from the data table given below we can infer that
The maximum number of click = 1 were obtained from individuals using Device Type 0 while the least were seen from Device Type 4

Percentage for click = 1 for Device Type 0
= 8067/(8067+30345) = 0.21000 = 21.00%

Thus, strategies should be targeted for devices in the following order
1) Device Type 0
2) Device Type 1
3) Device Type 5
4) Device Type 4

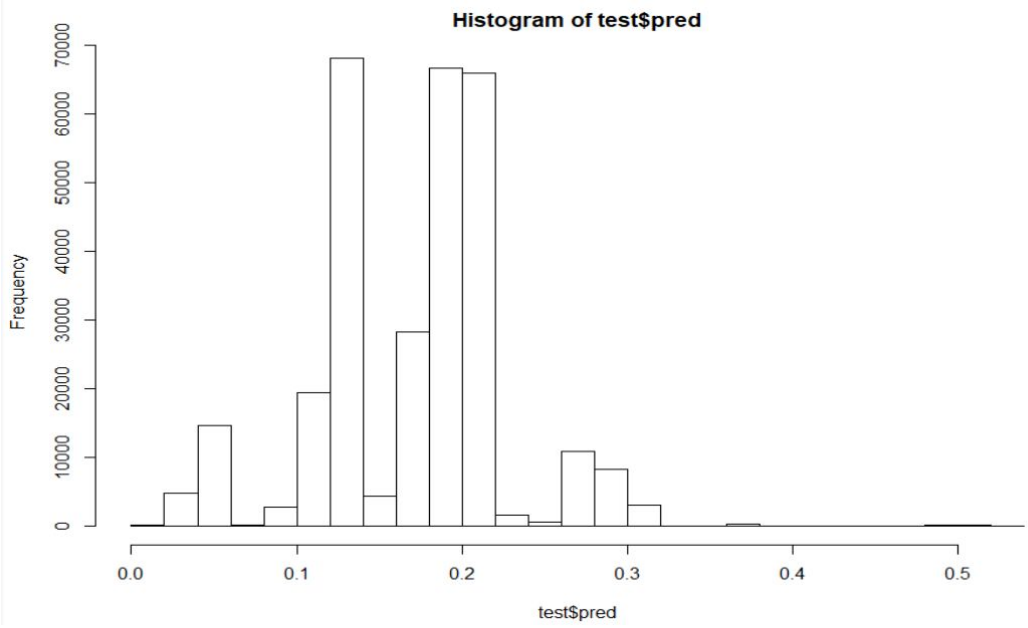


	1	0	4	5
0	536297	30345	12180	2011
1	109579	8067	1329	193

In order to showcase, the number of clicks = 1, the following visualization provides a general idea
The maximum number of clicks were obtained on Tuesday, followed by Wednesday and Thursday.

Probability Distribution

To check the probability distribution of the predictions obtained from the Logistic Regression model, we plot an Histogram. The visualization can be seen below:



It can be interpreted from the Histogram, that majority of probabilities predicted are below 30%. This shows how important it is to change the threshold in the Regression Model from 50% to a lower value for better accuracy.

Stage 4: Data Modelling

Logistic Regression:

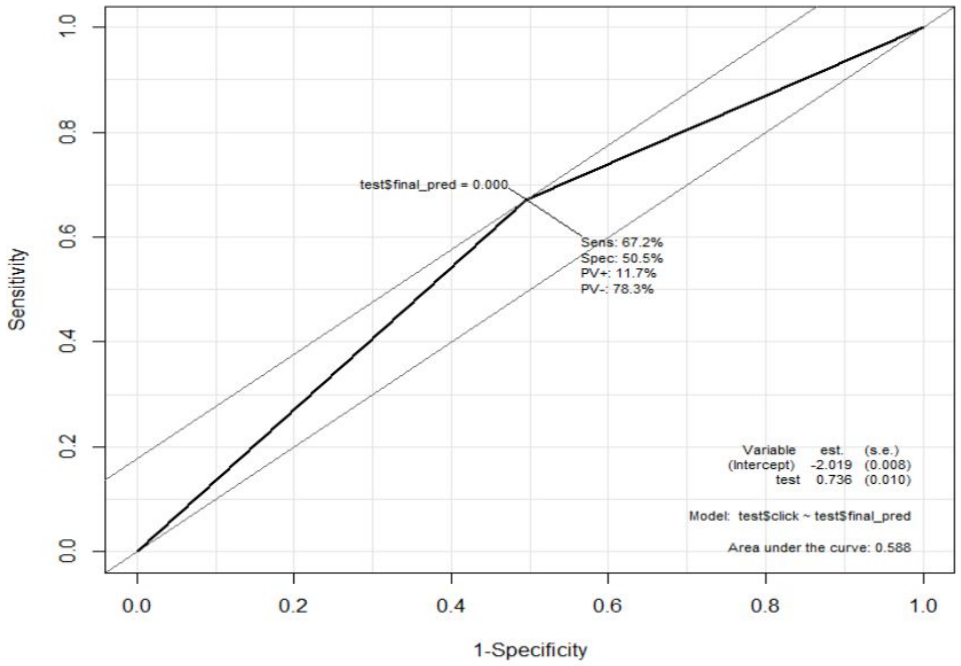
We will be using Logistic Regression to develop Predictive models for predicting the clicks. We know that the original threshold for Logistic Regression is 50% i.e. if the probability is greater than 50% the model will predict click = 1 or else click = 0

However, in this case the probability of clicks is very low.
The number of clicks = 1 in train set are 118858
118858 / (581412 + 118858) = 0.169731 ==> 17% percent of observations are clicks.
This means that we need to change the threshold of the model to get accurate clicks. To determine the optimal threshold we use ROC Curve.

The logistic Regression model detects the important variables which are as follows:

1. Site Category
2. Device Connection Type
3. Time of the day

4. Banner Position
5. C1



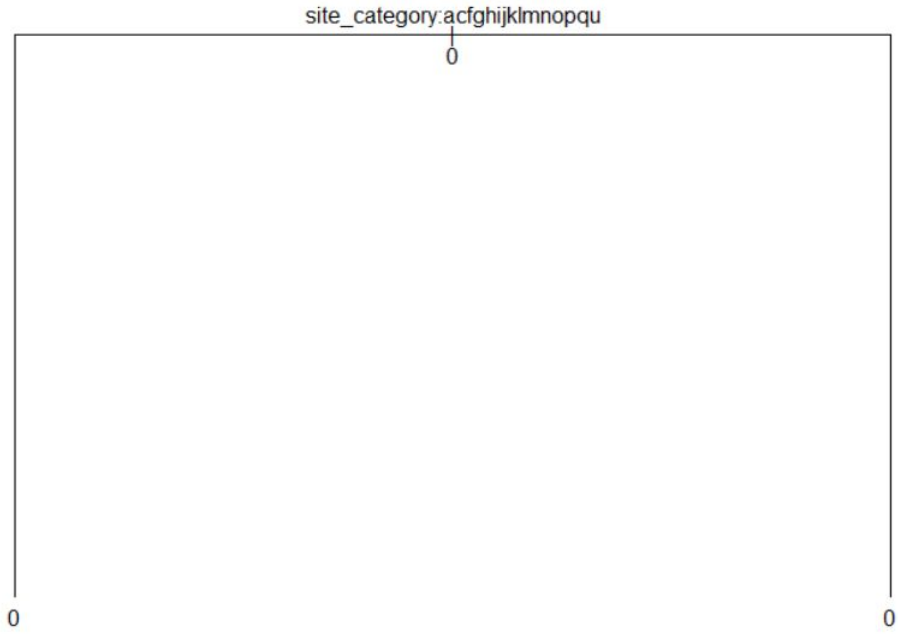
From the [5]ROC Curve, we can see that the maximum value of AUC is 0.588, thus we select the threshold at 0.18. This means that, if the probability of click is greater than 18% then the model will depict click = 1 otherwise click = 0.

Using prop.table, we can check the classification accuracy in the matrix. The prediction accuracy can then be calculated:
Prediction_Accuracy = ((34185 + 125770)/300000)*100 = 53.318%

Thus, we have significant improvement with the base probability of 17%. Though 53% model accuracy is not very useful in comparison to other machine learning models. However, we are predicting human behaviour in this case which is extremely difficult to predict. Thus, any improvement over the base probability can be deemed as useful.

Classification Trees:

The plot obtained from [6]classification tree model shows output as click = 0 as the end node of the tree. Also, it only uses the site_category parameter to classify the clicks ignoring rest of the important parameters from the model. It can be inferred that, because the probability of click = 1 is very low. The classification tree only predicts the majority outcome i.e. click = 0



Therefore, we will not be considering the classification tree model.

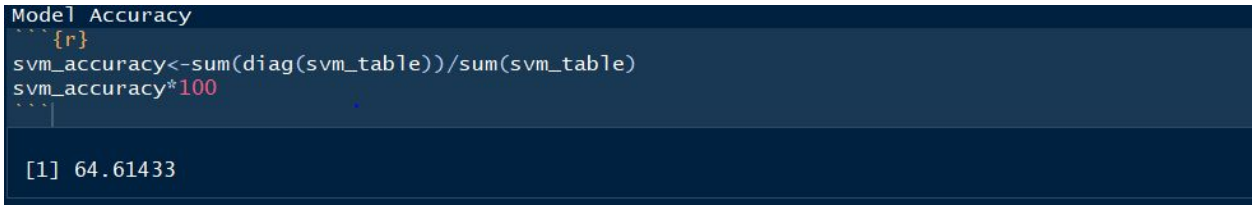
SVM Model:

We create a subset of the main dataset to train the [7]SVM Model over the data. Since, SVM takes a lot of time to train over the model. We extract 10,000 rows from the train dataset and create a subset.

Again, we need to change the threshold to get accurate predictions. In order to get the probabilities from the SVM Model we add probability = TRUE to our training model. The argument probability = TRUE provides us the probabilities of the clicks.

We use, attr(test$pred_svm, "probabilities") to select the probabilities and change the

threshold for the final output i.e. click = 1 or click = 0. The model accuracy is 64.61% which is really high as compared to Logistic Regression Model.



Stage 5: Evaluation

During this evaluation, we assess how useful the model has proved to achieve the business objectives previously set during the first step of the CRISP DM Methodology. The model developed definitely showcases the important parameters for predicting the clicks on an online advertisement. Also, fromt the visualization developed we can see the varying number of clicks during different times of the day. Moreover, we can develop strategies in order to improve our clicks depending on the model. Some of the strategies are discussed below in the Deployment stage.

Assessment of data mining results:

The Logistic Regression model developed during the project improves the accuracy of prediction from 17% which is base probability to 53% which is a significant improvement. Secondly, the SVM Model improves the prediction accuracy to 64.61%. Thus, the project meets its initial business objectives of increasing the clicks predicted over random click prediction(similar to coin-toss)

Approved Model:

The Logistic Regression Model is the approved model for deployment. This is because, it is simpler to train over the dataset as compared to SVM Model which takes longer period of time and adds further complexity to the model.

However, the SVM Model provides better accuracy to the problem. Thus, if time is not a factor and with better computational ability SVM Model is a better choice.

Stage 6: Deployment

In the deployment stage, we will develop strategies depending on the results obtained from the analysis.

Strategy Formulation:

We know that, the number of click = 1 are greatly dependent on the Device used i.e. Device Type.

Thus, we can should target the people using these devices. Though, we have no idea if Device Type 0 is a Laptop or Smartphone, an effective marketing strategy can be developed to reach out to maximum number of people by devising advertisements which suit such platforms.

For eg: If Device Type 0 is an Smartphone then, advertisements should be developed by the company such that the advertisements fit the screen of the Smartphone for easier understanding. Secondly, In order to get more people interested in the advertisement it should be made easily accessible through the device. Some of the advertisements require flash player and cannot be played on the Smartphone which would prevent the consumers who are using the Smartphones from viewing the advertisement. Thus, if the Device Type 0 is a platform wherein the advertisements cannot be successfully displayed, that would lead to loss of revenue.

Secondly, Banner position is one of the most important factors in deciding if the advertisement is going to be clicked. Logically thinking, we tend to observe more of the advertisements which are at the start of the page or at the immediate right hand side of the page. These are the areas which immediately grasp our attention. We can see from Visual analysis that Banner Position 7 provides the maximum number of clicks on the advertisements. Thus, major focus should be placed on acquiring the Banner Position 7(Primary Target) region in the webpage.

Thirdly, Device Connection Type proves to be an important variable in predicting the click. Examples of Device Connection type would be Wifi, 4g Spectrum Data, Google Fibre Connection etc. From our analysis, Device Connection Type 0 seems to be the Connection Type wherein majority of the people who can view the advertisement click on it i.e. click =1. So, before targeting new consumers by purchasing advertising spaces on different webpage, the connection type of the consumer needs to checked to prevent loss in revenue.

According to the Logistic Regression Model, Site Category and C1 are important parameters as well. However, C1 is an anonymized variable and hence strategy formulation using C1 is not possible unless the variable is further explained. Also, Site Categories play an essential

role to check if the advertisement would be clicked or not. Most of the advertisements are clicked from websites which are closely related to the content people are searching for. But, the categories have been anonymized in the data which is why it would be difficult to predict which websites should be targeted. Site categories 28905ebd, 3e814130, 42a36e14, 50e219e0, 70fb0e29, c0dd3be3, dedf689d, f028772b are the ones that are termed extremely relevant by the model. Categories e787de0e, 75fa27f6 and 335d28a8 also are relevant.

Also, Day of the week and the time of the day can help in predicting the clicks on an advertisement. The Regression model shows both of them as an important parameter. For Eg: The model suggests that maximum number of people have clicked on the advertisement on a Wednesday. While, the maximum clicks obtained from the circular chart are in the afternoon and evening time slot. Thus, we can target people depending on the time they log in onto any webpage.

Strategy:

The Strategy that should be finally deployed should include all the points covered above. The following factors are of utmost importance in increasing the clicks on advertisements:

- Device Connection Type
- Device Type
- Banner Position
- Site Category
- Time of the Day
- Day of the week
- C1

Thus, before investing on an advertisement slot on any web-page, the above factors should be properly analysed. These factors increase can help us predict the click rate upto 64.61% accuracy rather than 17% which is the random probability. Using these variables can drastically reduce the revenue spent on advertising as well as selectively target people who will actually click on the advertisement.

Let's understand via an example

[8]Southwest Airlines advertises itself as a low-cost, low-frills carrier with frequent flights to many destinations around the United States. The airline focuses its marketing efforts on middle-class families, small business owners, those traveling short distances, and young adults.

Conversely, United Airlines targets College Graduate and Business People who have a fixed income greater than 50,000$. Because of this targeted marketing, United sells more full-cost fares and increases their overall revenue.

Similarly, we can also devise our Marketing Plan. Deployment of this Marketing plan which is targeted towards consumers rather than a broad spectrum of audience.
An Individual with Device Connection Type 0 using Device Type 0 browsing on a Wednesday during Afternoon or Evening Time Slots on Site Category 3e814130 should be the main target of the company. For such an individual, the company should invest on Banner Position 7 which is deemed important by our model.

Once, the Marketing Campaign has been launched, the company needs to follow the result of the marketing campaign. The company should alter and change the format of the campaign depending on the success of the Marketing campaign. After which, the company needs to focus on expansion and ways to improve future sales.

Risk Assessment:
Due to the large nature of the dataset, the problem would be technological constraints like processing ability of the laptop. RAM Extension would be a solution to speed up the process without causing the laptop to slow down. The dataset contains a number of anonymized categorical variables which are important for analysis. However, no information can be obtained from these variables by us to develop strategies. Since, the Threshold has to be changed and the probability is too low i.e.17% classification trees cannot be used to develop models.

References:

[1]Karen Moked,(2016 December), https://www.digilant.com/digilant_university/players-in-programmatic/
[2]Sydney Hadden, https://www.aabacosmallbusiness.com/advisor/important-click-rate-ctr-really-042603295.html
[3]CRISP DM Methodology, http://www.sv-europe.com/crisp-dm-methodology/
[4]Rose Diagram, https://www.rdocumentation.org/packages/circular/versions/0.4-93/topics/rose.diag
[5]ROC Curves, http://www.dataschool.io/roc-curves-and-auc-explained/
[6]Classification Trees, https://www.r-bloggers.com/classification-trees/
[7]SVM, https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/
[8]Targeted Marketing Guide: http://www.marketing-schools.org/types-of-marketing/targeted-marketing.html
[9] Textbook: Provost, F. & Fawcett, T. “Data Science for Business”, O’Reilly, 2013
[10] Textbook: R for Data Science
Dataset:
[11]https://www.kaggle.com/c/avazu-ctr-prediction/data

Targeted Marketing CTR Prediction

Avirat Gaikwad

October 4, 2017

I have already created a smaller dataset, so we don't need to use the ff function to load the entire file.

```
setwd("C:/Users/Avirat/Downloads")
file<- read.csv("small.csv") #Reading the file
head(file)
```

id	click	hour	C1	banner_pos	site_id	site_domain	site_category	app_id	app_domain	app_category	device_id	device_
3.443805e+18	1	14102618	1005	0	1fbe01fe	f3845767	28905ebd	ecad2386	7801e8d9	07d7df22	a99f214a	7a5f99e
8.774285e+18	0	14102213	1005	0	5bcf81a2	9d54950b	f028772b	ecad2386	7801e8d9	07d7df22	a99f214a	ff7d3a8c
4.674893e+18	0	14102104	1005	1	e151e245	7e091613	f028772b	ecad2386	7801e8d9	07d7df22	a99f214a	df7fc01c
2.205573e+18	0	14102909	1005	0	1fbe01fe	f3845767	28905ebd	ecad2386	7801e8d9	07d7df22	a99f214a	f4e9cdb
7.315236e+18	0	14102110	1005	1	0eb72673	d2f72222	f028772b	ecad2386	7801e8d9	07d7df22	a99f214a	51b5e76
5.523710e+18	1	14102214	1005	1	e151e245	7e091613	f028772b	ecad2386	7801e8d9	07d7df22	a99f214a	72bf3bc

Extracting the information from hour column in the dataset.

```
file$hour <- strptime(file$hour," %y%m%d%H")
class(file$hour)
```

```
## [1] "POSIXlt" "POSIXt"
```

```
file$date <- as.Date(file$hour) #Create a seprate date column
file$date <- weekdays(file$date) # Converting date into day of the week
file$date <- as.factor(file$date)
```

Splitting the file to use the variables we need for analysis

```
file_model<- file[,c(2,4,5,8,14,15,16,25)]
file_model$C1 <- as.factor(file_model$C1)
file_model$click<-as.factor(file_model$click)
file_model$device_type<-as.factor(file_model$device_type)
file_model$device_conn_type<-as.factor(file_model$device_conn_type)
file_model$banner_pos<- as.factor(file_model$banner_pos)

file_model$hr<- hour(file$hour)
file_model$hr <- as.factor(file_model$hr)
str(file_model)
```

```
## 'data.frame':    1000000 obs. of  9 variables:
##  $ click          : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 1 1 1 ...
##  $ C1             : Factor w/ 7 levels "1001","1002",..: 3 3 3 3 3 3 3 3 3 7 ...
##  $ banner_pos     : Factor w/ 7 levels "0","1","2","3",..: 1 1 2 1 2 2 1 1 1 1 ...
##  $ site_category  : Factor w/ 21 levels "0569f928","28905ebd",..: 2 20 20 2 20 20 12 6 6 6 ...
##  $ device_model   : Factor w/ 5153 levels "00097428","0009f4d7",..: 2786 3418 2466 2199 2786 604 4814 2739 3052 241 ...
##  $ device_type    : Factor w/ 4 levels "0","1","4","5": 2 2 2 2 2 2 2 2 2 2 ...
##  $ device_conn_type: Factor w/ 4 levels "0","2","3","5": 1 1 1 1 1 1 1 2 3 2 ...
##  $ date           : Factor w/ 7 levels "Friday","Monday",..: 4 7 6 7 6 7 6 7 6 6 ...
##  $ hr             : Factor w/ 24 levels "0","1","2","3",..: 19 14 5 10 11 15 2 13 20 5 ...
```

Changing the reference level for moving down the ranks output

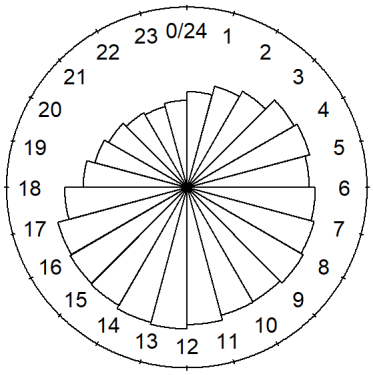
```
file_model$C1 <- relevel(file_model$C1, ref="1005")
file_model$banner_pos<- relevel(file_model$banner_pos, ref="0")
file_model$device_type<- relevel(file_model$device_type, ref="1")
file_model$device_conn_type<- relevel(file_model$device_conn_type, ref="0")

train1 <- file_model
new<-data.frame()
values <- c("Night","Night","Night","Night","Night","Night","Morning","Morning","Morning","Morning","Morning","Morning",
            "AfterNoon","AfterNoon","AfterNoon","AfterNoon","AfterNoon","AfterNoon","Evening","Evening","Evening","Evening",
"Evening","Evening")
a <- values[file_model$hr]

new<-cbind(file_model,a)
head(new)
```

click	C1	banner_pos	site_category	device_model	device_type	device_conn_type	date	hr	a
1	1005	0	28905ebd	8a4875bd	1	0	Sunday	18	Evening
0	1005	0	f028772b	a9fb0439	1	0	Wednesday	13	AfterNoon
0	1005	1	f028772b	7abbbd5c	1	0	Tuesday	4	Night
0	1005	0	28905ebd	6e1e2240	1	0	Wednesday	9	Morning
0	1005	1	f028772b	8a4875bd	1	0	Tuesday	10	Morning
1	1005	1	f028772b	1f0bc64f	1	0	Wednesday	14	AfterNoon
Splittin	g datas	et into test	and train data						

```
set.seed(1)
index<- sample(1:nrow(new), size = 0.3*nrow(new))
test <- new[index,]
train <- new[-index,]
head(train)
```

	click	C1	banner_pos	site_category	device_model	device_type	device_conn_type	date	hr	a
1	1	1005	0	28905ebd	8a4875bd	1	0	Sunday	18	Evening
2	0	1005	0	f028772b	a9fb0439	1	0	Wednesday	13	AfterNoon
5	0	1005	1	f028772b	8a4875bd	1	0	Tuesday	10	Morning
6	1	1005	1	f028772b	1f0bc64f	1	0	Wednesday	14	AfterNoon
7	0	1005	0	76b2941d	ef726eae	1	0	Tuesday	1	Night
10	0	1012	0	50e219e0	0bcabeaf	1	2	Tuesday	4	Night
Visua	lization	to che	ck the times	when the clicks	are maximum i.e	. click = 1				

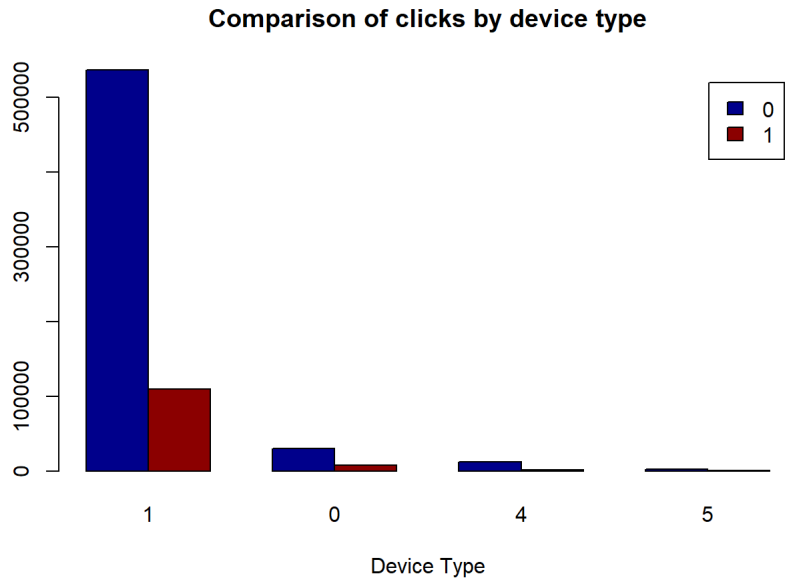
```
clicked <- file[file$click== 1,]
tmp.cir <- circular(hour(clicked$hour)%%24, units = "hours", template = "clock24")
rose.diag(tmp.cir, bins = 24, prop = 3.25)
```




```
options(scipen = 999)

#Grouped Bar Plot for device type
device.type.count <- table(train$click, train$device_type)
device.type.count
```

/	1	0	4	5
0	536296	30345	12180	2011
1	109579	8067	1329	193

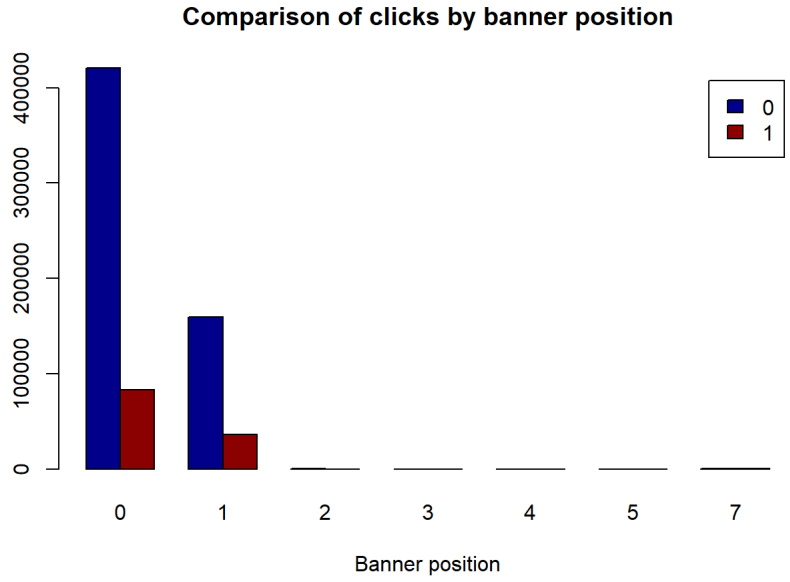
```
barplot(device.type.count, main="Comparison of clicks by device type",
  xlab="Device Type", col=c("darkblue","darkred"),
    legend = rownames(device.type.count), beside=TRUE)
```




```
# Grouped Bar Plot for banner position
banner.pos.count <- table(train$click, train$banner_pos)
banner.pos.count
```

/	0	1	2	3	4	5	7
0	420753	159127	219	25	99	88	521
1	82879	35989	31	7	17	8	237

```
barplot(banner.pos.count, main="Comparison of clicks by banner position",
  xlab="Banner position", col=c("darkblue","darkred"),
    legend = rownames(banner.pos.count), beside=TRUE)
```

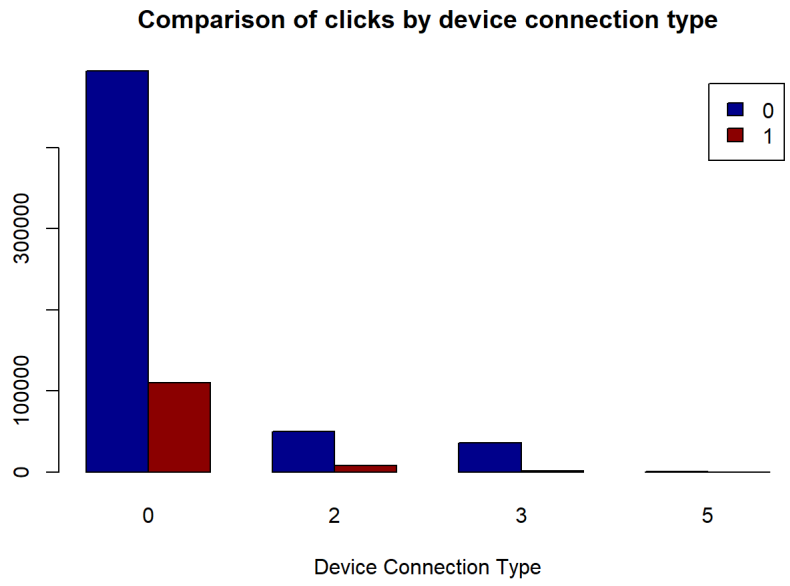



```
# Grouped Bar Plot for device connection type
device.conn.type.count <- table(train$click, train$device_conn_type)
device.conn.type.count
```

/	0	2	3	5

/	0	2	3	5
0	494587	49584	35991	670
1	109704	7708	1738	18

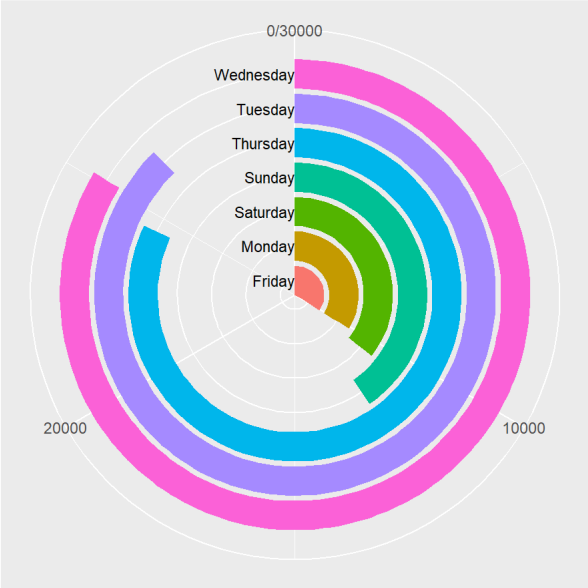
```
barplot(device.conn.type.count, main="Comparison of clicks by device connection type",
  xlab="Device Connection Type", col=c("darkblue","darkred"),
    legend = rownames(device.conn.type.count), beside=TRUE)
```




```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```
# Circular barplot for clicks per day
clicks.only <- train[train$click==1,]
clicks.per.day <- table(clicks.only$date)
clicks.per.day <- as.data.frame(clicks.per.day)
colnames(clicks.per.day) <- c("Day","Clicks")
ggplot(clicks.per.day, aes(x = Day, y = Clicks ,fill = Day)) +
    geom_bar(width = 0.85, stat="identity") +

    # To use a polar plot and not a basic barplot
    coord_polar(theta = "y") +

    #Remove useless labels of axis
    xlab("") + ylab("") +

    #Increase ylim to avoid having a complete circle
    ylim(c(0,30000)) +

    #Add group labels close to the bars :
    geom_text(data = clicks.per.day, hjust = 1, size = 3, aes(x = Day, y = 0, label = Day)) +

    #Remove useless legend, y axis ticks and y axis text
    theme(legend.position = "none" , axis.text.y = element_blank() , axis.ticks = element_blank())
```



The predict function would give us an error since a new level has been added in the test set Adding row with level 2 in banner position column from test to train set would solve this problem

```
level<-test[test$banner_pos =="2",]

train <- rbind(train, level[1,])
tail(train)
```

	click	C1	banner_pos	site_category	device_model	device_type	device_conn_type	date	hr	a
999995	1	1005	0	28905ebd	a0f5f879	1	0	Thursday	22	Evening
999997	0	1005	0	50e219e0	542422a7	1	0	Monday	3	Night
999998	0	1005	0	f028772b	31025cda	1	0	Tuesday	16	AfterNoon
999999	0	1005	0	50e219e0	56f23684	1	2	Thursday	22	Evening
1000000	0	1005	0	3e814130	ef726eae	1	0	Thursday	10	Morning
598482	0	1005	2	28905ebd	36b67a2a	1	0	Saturday	15	AfterNoon

Creating a Logistic regression model:

```
fit<- glm(click~ banner_pos + C1 + site_category + device_type + device_conn_type + a, data = train, family = "binomial")
summary(fit)
```

```
##
## Call:
## glm(formula = click ~ banner_pos + C1 + site_category + device_type +
##     device_conn_type + a, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4335  -0.6513  -0.5654  -0.4867   3.0274
##
## Coefficients: (3 not defined because of singularities)
##                         Estimate Std. Error z value             Pr(>|z|)
## (Intercept)            -3.020223   0.274421 -11.006 < 0.0000000000000002
## banner_pos1             0.110941   0.009849  11.264 < 0.0000000000000002
## banner_pos2             0.950552   0.209044   4.547   0.0000054378225327
## banner_pos3             0.655543   0.442299   1.482             0.138306
## banner_pos4            -0.182553   0.262698  -0.695             0.487109
## banner_pos5            -0.505908   0.369357  -1.370             0.170781
## banner_pos7             1.777159   0.085791  20.715 < 0.0000000000000002
## C11001                 -1.595953   0.507264  -3.146             0.001654
## C11002                  0.558496   0.014375  38.852 < 0.0000000000000002
## C11007                 -1.147125   0.242009  -4.740   0.0000021371508162
## C11008                        NA         NA      NA                   NA
## C11010                 -0.459390   0.078654  -5.841   0.0000000051994640
## C11012                 -0.199524   0.078596  -2.539             0.011130
## site_category28905ebd   1.719214   0.274442   6.264   0.0000000003742985
## site_category335d28a8   0.808217   0.282445   2.861             0.004216
## site_category3e814130   2.100052   0.274456   7.652   0.0000000000000198
## site_category42a36e14   1.652445   0.480270   3.441             0.000580
## site_category50e219e0   1.169378   0.274441   4.261   0.0000203571583582
## site_category5378d028   0.838639   1.089277   0.770             0.441357
## site_category70fb0e29   1.304212   0.310840   4.196   0.0000271956775345
## site_category72722551   0.517231   0.324459   1.594             0.110906
## site_category74073276  -5.750714 119.467656  -0.048             0.961608
## site_category75fa27f6   0.869864   0.281008   3.096             0.001965
## site_category76b2941d  -0.364908   0.307236  -1.188             0.234947
## site_category8fd0aea4   0.235723   0.476137   0.495             0.620548
## site_category9ccfa2ea  -5.838489  48.754788  -0.120             0.904680
## site_categorya818d37a  -7.479288  17.803112  -0.420             0.674404
## site_categorybcf865d9   1.757868   1.175949   1.495             0.134954
## site_categoryc0dd3be3   1.122492   0.297439   3.774             0.000161
## site_categorydedf689d   3.148623   0.296192  10.630 < 0.0000000000000002
## site_categorye787de0e   1.943024   0.713525   2.723             0.006467
## site_categoryf028772b   1.466651   0.274263   5.348   0.0000000891285954
## site_categoryf66779e6  -0.187680   0.285921  -0.656             0.511564
## device_type0                  NA         NA      NA                   NA
## device_type4            0.161703   0.082305   1.965             0.049450
## device_type5                  NA         NA      NA                   NA
## device_conn_type2      -0.142245   0.013307 -10.690 < 0.0000000000000002
## device_conn_type3      -1.143751   0.025661 -44.571 < 0.0000000000000002
## device_conn_type5      -1.517325   0.240927  -6.298   0.0000000003017876
## aEvening               -0.044358   0.009678  -4.583   0.0000045748570766
## aMorning               -0.056863   0.008153  -6.975   0.0000000000030682
## aNight                 -0.034305   0.009033  -3.798             0.000146
##
## (Intercept)           ***
## banner_pos1           ***
## banner_pos2           ***
## banner_pos3
## banner_pos4
## banner_pos5
## banner_pos7           ***
## C11001                **
## C11002                ***
## C11007                ***
## C11008
## C11010                ***
## C11012                *
## site_category28905ebd ***
## site_category335d28a8 **
## site_category3e814130 ***
## site_category42a36e14 ***
## site_category50e219e0 ***
## site_category5378d028
## site_category70fb0e29 ***
## site_category72722551
## site_category74073276
## site_category75fa27f6 **
## site_category76b2941d
## site_category8fd0aea4
## site_category9ccfa2ea
## site_categorya818d37a
```

```
## site_categorybcf865d9
## site_categoryc0dd3be3 ***
## site_categorydedf689d ***
## site_categorye787de0e **
## site_categoryf028772b ***
## site_categoryf66779e6
## device_type0
## device_type4          *
## device_type5
## device_conn_type2     ***
## device_conn_type3     ***
## device_conn_type5     ***
## aEvening              ***
## aMorning              ***
## aNight                ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 638774  on 700000  degrees of freedom
## Residual deviance: 622113  on 699962  degrees of freedom
## AIC: 622191
##
## Number of Fisher Scoring iterations: 9
```

```
test$pred<-predict(fit, test, type="response")
```

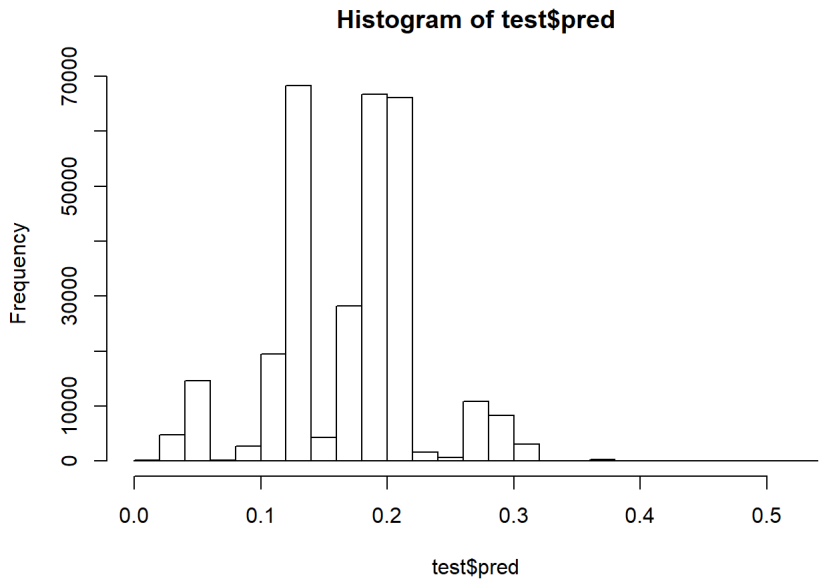
```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
test$final_pred<-ifelse(test$pred> 0.18,1,0)
table(test$click,test$final_pred)
```

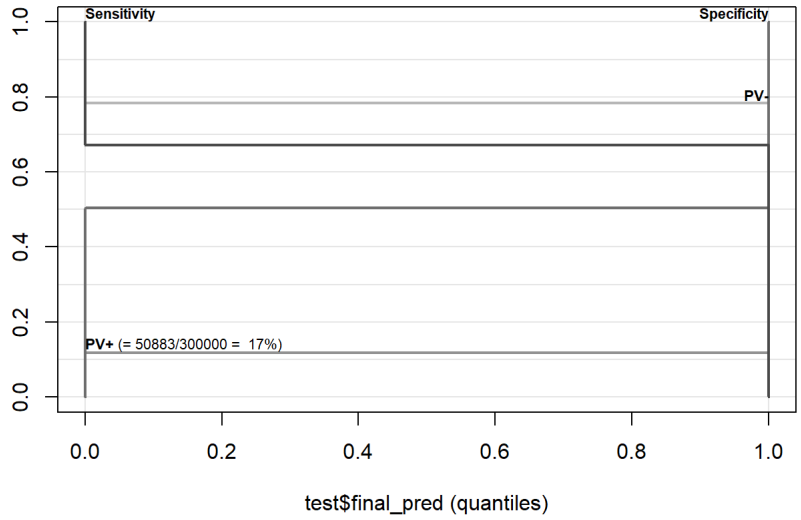
/	0	1
0	125770	123347
1	16698	34185

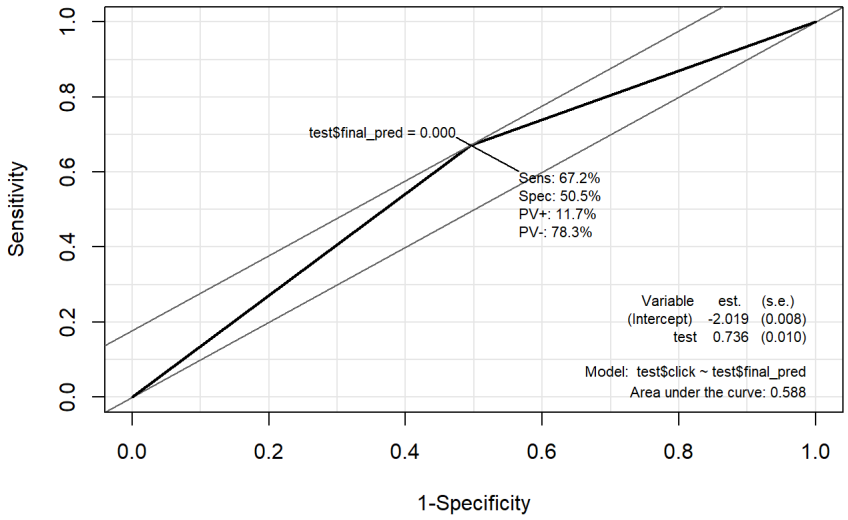
ROC curve is used to find the probability tradeoff. The ROC Curve can be seen below:

```
hist(test$pred)
```




```
ROC(test=test$final_pred,stat = test$click)
```






```
tab<- xtabs(~ click+ final_pred, test)
tab
```

click/final_pred	0	1
0	125770	123347
1	16698	34185

```
prop.table(tab)
```

click/final_pred	0	1
0	0.4192333	0.4111567
1	0.0556600	0.1139500

Here, we can see

```
Prediction_Accuracy = ((34185 + 125770)/300000)*100
Prediction_Accuracy
```

```
## [1] 53.31833
```

Since, training the SVM model takes a lot of time, we will work on a smaller subset of the data to work out a piece of code and then apply it to the entire dataset.

```
subdata<- train[1:10000,]
fit_svm<- svm(click~ banner_pos + C1 + site_category + device_type + device_conn_type + a, data = subdata, probability = TRU
E)
summary(fit_svm)
```

```
##
## Call:
## svm(formula = click ~ banner_pos + C1 + site_category + device_type +
##     device_conn_type + a, data = subdata, probability = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.02380952
##
## Number of Support Vectors:  3361
##
##  ( 1651 1710 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

Predictions using the SVM Model

```
test$pred_svm<- predict(fit_svm, test[1:10], probability = TRUE)
x<-(attr(test$pred_svm, "probabilities"))
head(x)
```

	1	0
265509	0.1650160	0.8349840
372124	0.1650331	0.8349669
572853	0.1650120	0.8349880
908206	0.1650018	0.8349982
201682	0.1650095	0.8349905
898386	0.1650368	0.8349632

```
test$final_predsvm<-ifelse(x[,1]> 0.165,1,0)
svm_table<-table(test$click,test$final_predsvm)
```

Classification Tree model:

```
library(tree)
```

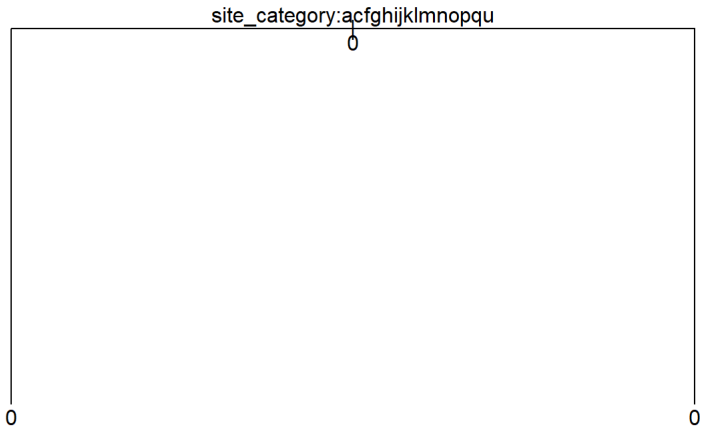
```
## Warning: package 'tree' was built under R version 3.4.2
```

```
fit_tree<- tree(click~ banner_pos + C1 + site_category + device_type + device_conn_type + a, data = train)
summary(fit_tree)
```

```
##
## Classification tree:
## tree(formula = click ~ banner_pos + C1 + site_category + device_type +
##     device_conn_type + a, data = train)
## Variables actually used in tree construction:
## [1] "site_category"
## Number of terminal nodes:  2
## Residual mean deviance:  0.9019 = 631300 / 700000
## Misclassification error rate: 0.1702 = 119168 / 700001
```

We can check the plot obtained from Classification tree

```
plot(fit_tree)
text(fit_tree, all = T)
```



From the plot it can be inferred that,

because the probability of click = 1 is very low. The classification tree only predicts the majority outcome i.e. click = 0 Therefore, we will not be considering the classification tree model.