Ahmednagar Jilha Maratha Vidya Prasarak Samaj`s

NEW ARTS, COMMERCE AND SCIENCE COLLEGE, AHMEDNAGAR

DEPARTMENT OF STATISTICS

2019-2020

A PROJECT REPORT ON

Prediction of Revenue from Advertisements by using Machine Learning Techniques

Submitted By

1) Kandekar Sandip Ramdas

2) Kardile Rutuja Dipak

3) Sonawane Harshada Suresh

Under The Guidance Of

Prof. B.K.Thorve Sir

New Arts Commerce And Science College,

Ahmednagar

DEPARTMENT OF STATISTICS

CERTIFICATE

This is to certify that the _____of class MSc.-II has completed all assigned project of the *" Prediction of Revenue from Advertisemments by using Machine Learning Techniques''*

As laid down by the "Savitribai Phule Pune University" for the acadmic year 2019-2020

Project Guide **Head Of Department**

External Examiner

ACKNOWLEDGEMENT

An accomplishment and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Mr. M.S.Kasture Sir, Head of the department of Statistics and also an experienced project guide, for providing me an opportunity to do the project work and giving us all support and guidance which made me complete the project duly. I am extremely thankful to him for providing such a nice support and guidance, although he had busy schedule managing the regular lectures as well as important meetings.

I owe my deep gratitude to our project guide Mr. B.K.Thorve sir, Professor at department of Statistics, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all the Teaching Staff of department of Statistics which helped us in successfully completing our project work. I would like to thank our senior Pandit Kalhapure for his guidance and help. Also, I would like to extend our sincere esteems to all the young people who gave their valuable time as well as their responses which was indeed essential for our project analysis.

INDEX

ABSTRACT

Advertising is the integral part of our daily life. It is a pervasive method of marketing in society which encourages people to purchases goods and services. Advertising contributes to bring about all round development of the economy by increasing demand and by encouraging economic activities which in turn improves the income.

A huge advertisement datasets are available on internet and many sites which are used in real world applications. Currently advertisement is one of the leading income source in society .There are various data mining technique to predict the revenue of advertisement. A total of 15438 observations are in prebid datasets. We used to 4 techniques which are random forest, decision tree, multiple regression and support vector regression to compare the accuracy of prediction of revenue.

Key Words: Advertisement, Regression, Machine Learning, Data Mining, Support vector regression, Radom forest, Decision Tree, Accuracy.

INTRODUCTION

In today's world all of us are under the influence of "Advertisement" Right from buying groceries to children study material, finding a holiday spot to watching movie , selecting restaurant for dinner to booking a banquet hall for special event and searching educational institutional to hunting for a company to find jobs, almost act is guided and decided by advertisements.

What is an advertisement?

Advertisement is an efficient and effective technique to promote goods, services and ideas. It is a paid from of non-personal communication where in business information is made available for potential customers.

Behind every advertisement, our main aim is to earn more and more revenue. Advertisement revenue is the monetary income that individuals and business earn from displaying paid advertisement on their websites, social media, channels, or other platforms. There are various types of advertisement such as Online advertisement, Social media marketing, Banner advertisement Radio advertisement, Television, Newspaper. As well as which type of advertisement will earn more revenue such as banner advertisement, online advertisement etc. Banner advertising refers to the use of rectangular graphic display that stretches across the top, bottom or sides of a websites or online media property. In banner advertisement rendered size is most important factor that means which dimension of banner looks of banner looks good for advertisement to gain more profit.

The purpose of this project is to introduced the different factors behind advertisement in these which is effective. For analysis we used secondary data in that there are total 13 (factors) or variables such as

Currency: - USD currency was used in this data. i.e. United States Dollar

CPM: - CPM is Cost Per Mile and CPM is calculated by formula, (revenue/impression) *1000

Bidder code: -The unique code Prebid is used by ad servers line items to identify the bidder.

Media type: - A media type is a two-point identifier for file formats and format contents transmitted on the internet. In this data banner is taken as media type.

Device type: - A device type is a group of devices. It always you to gather device and define common means to process the data they transmit i.e.. mobile, desktop. Which is are in our data.

Host: - A host in publishing advertisement is an agency or board who are providing all funds during it like TOI (Times of India).

Rendered size: -Rendered size is the dimension of the banner.

Request: -An ad request is counted whenever your site request ads to be displayed. It is the number of ad units that requested for ads.

Impression: - It is an ad view. It is a point at which an ad viewed once by a visitor or display once on web.

Bids-count: - Bid is the maximum amount of money an advertiser is willing to pay for each click on an advertisement.

Publisher: - These advertisements were published by "Times Of India" an Indian publisher.

Revenue: -Revenue is the income in which earn from the displaying paid advertisement on social media, channel or websites.

Many data mining techniques have been used over time by researcher to predict which factor gives more revenue. We have also proposed an effective data mining technique to predict the important or effective factor. Then we compare different techniques of regression (response variable is continuous) and chi-square test to check the correlation them.

We have studied the above aspect and carried out our project by keeping in mind the steps involved in the definition of the statistics i.e. "collection of the data, presentation of the data, analysis of the data and drawing relevant conclusion from it".

DATA DESCRIPTION

We have secondary data which we were taken from Kaggle. In our data total number of observations are 15438 with 13 variables in which 8 are categorical and 5 are numeric. This data describes terminology used for advertisements and whether or not at which level these terminologies will have affected on revenue of advertisement. Given below is the description and coding of the variables.

Description:

Categorical Variable:

1) **Time Keys**: It represents date, on which date in the data has been recorded, here the data of 9 days i.e 9 dates are given in dataset.

Time key is coded as,

16/09/2019 - 1

17/09/2019 - 2

18/09/2019 - 3

19/09/2019 - 4

20/09/2019 - 5

21/09/2019 - 6

22/09/2019 - 7

23/09/2019 – 8

24/09/2019 – 9

2) **Bidder code**: The unique code prebid. The header bididng code in the page header executes and calls all demand partners like , ix, openx, rubicon, sonobi simultaneously to bid on this impression.
   i) Ix - This module connects publisher to Index. Exchanges (ix) network of demand sources through prebid. It is compatible with both older ad unit formats
   Where the sizes & media type properties are placed of the top level of the ad unit.
   ii) Openx - openx delivery domain a platform id provided by our openx representative. Customers float minimum price in use.
   iii) Rubicon – Array of page specific keywords may be referred in rubion project reports. Rubion project does not make concurrent banner and video request.

iv) Sonobi - The sonobi bidder adaptor requires setup and approval from sonobi account manager. Coma separated list of keywords about the site.

Bidder code is coded as,
Ix – 1
Openx - 2
Rubion – 3
Sonobi – 4

3) **Currency** : USD (United States Dollar) is currency in our data. The USD is the most traded currency in the forex market and can be paired with all other major currencies currency is coded as,
USD - 1
4) **Publisher Name** : TOI (Times of India)
Times of India is an Indian English language daily newspaper owned by The times group. It is owned and publish by Bennett, coleman & co.Ltd .
publisher name is coded as,
TOI – 1.
5) **Media Type**: One of the most important feature of style sheet is that they specify how document is to be presented on different media on the screen, on paper etc. In our data media type is banner and
banner is coded as,
Banner – 1
6) **Device type** - Device type is a group of devices. It allows us to gather device and define common means to process the data they transmit ,
device type is coded as ,
Mobile-1, Desktop – 2
7) **Host** - Advertising host is a company providing best and affortable website designing, website development and other things so that we gain profit from those advertisements. Times of India is the host of our dataset &
host is coded as,
Times of India - 1
8) **Rendered Size** : Dimension of Image
Basically, when we are rendering an image we have 3 variables to consider.
   i) DPI/PPI (Dots per inch/pixels per inch)
   ii) Pixel size
   iii) Print size

   Rendered size is coded as,

   300*250 – 1
   300*600 – 2
   728*90 – 3
   320*250 – 4
   320*50 – 5
   970*30 – 6
   320*50 – 7

Numerical Variable:

9) **Request:** An request is counted whenever our site request ads to displayed. It is the number of ad unites that request ads or search quires. If we are a social media content creator, this social media post request from will highly ease your business and will help you to keep your request in order. If we are social media content creator, this social media post request form will highly ease your business and will help you to keep your request in order.
10) **Impression** – Impression is reffered as an ad view means is a metric used to quantity the display of an advertisement on web page. In some cases online advertisement, which often pays on per impression basis.
11) **Bids-count** : Bids is a maximum amount of money an advertiser is willing to pay for each click on an advertisement
12) **CPM** – cost per thousand also cost per mile is a marketing term used to denote the price of 1000 advertisement impression on one web page
    CPM = (cost/ impression)*1000
13) **Revenue** : Revenue is the financial income when earn from the displaying paid advertisement on social media , website.
    Revenue is the response variable in the data.

OBJECTIVE

- **To study the relation between response and others variables.**
- **Which factors are mostly responsible to increase revenue.**
- **To study, to fight competition in the market and to increase the sales (by observing revenue).**
- **To study different machine learning techniques such as Random Forest, Decision Tree, Multiple Linear Regression, support vector regression.**
- **Choose best models from these various machine learning techniques.**

METHODOLOGY

Regression Versus Classification Problems

Variables can be characterized as either quantitative or qualitative (also known as categorical). Quantitative variables take on numerical values. Examples include age, height and temperature. On the other hand, qualitative variables take on values in one of K different classes, or categories. Examples of qualitative variables include gender (male or female), the brand of product purchased (brand A, B, or C) and a patient is suffering from a liver disease (Yes or No). We tend to refer to problems with a quantitative response as regression problems, while those involving a qualitative response are often referred to as classification problems.

So, in this project I'm focusing on to regression methods. Here are some regression techniques as follows.

Multiple regression

In statistical modelling, regression analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome variable') and one or more independent variables (often called 'predictors', 'covariates', or 'features'). The most common form of regression analysis is linear regression, in which a researcher finds the line (or a more complex linear combination) that most closely fits the data according to a specific mathematical criterion. For example, the method of ordinary least squares computes the unique line (or hyperplane) that minimizes the sum of squared distances between the true data and that line (or hyperplane).

Regression analysis is primarily used for two conceptually distinct purposes. First, regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Second, in some situations regression analysis can be used to infer causal relationships between the independent and dependent variables. Importantly, regressions by themselves only reveal relationships between a dependent variable and a collection of independent variables in a fixed dataset.

Support Vector Regression (SVR)

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already

requested from the problem. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration. However, the main idea is always the same to minimize error.

Decision Tree

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

A decision tree consists of three types of nodes:

1. Decision nodes – typically represented by squares
2. Chance nodes – typically represented by circles
3. End nodes – typically represented by triangles

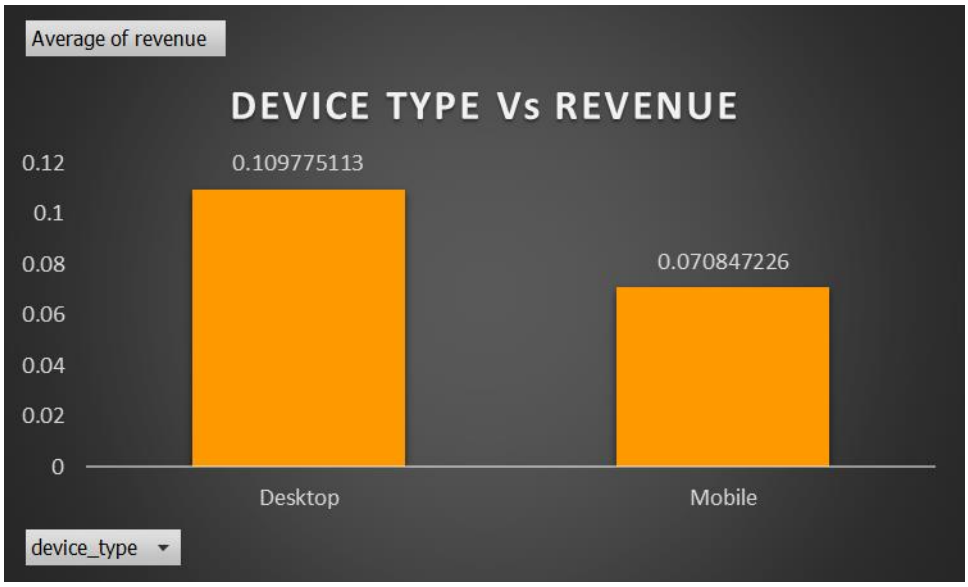
Advantages

1. Are simple to understand and interpret. People are able to understand decision tree models after a brief explanation.
2. Have value even with little hard data. Important insights can be generated based on experts describing a situation (its alternatives, probabilities, and costs) and their preferences for outcomes.
3. Help determine worst, best and expected values for different scenarios.

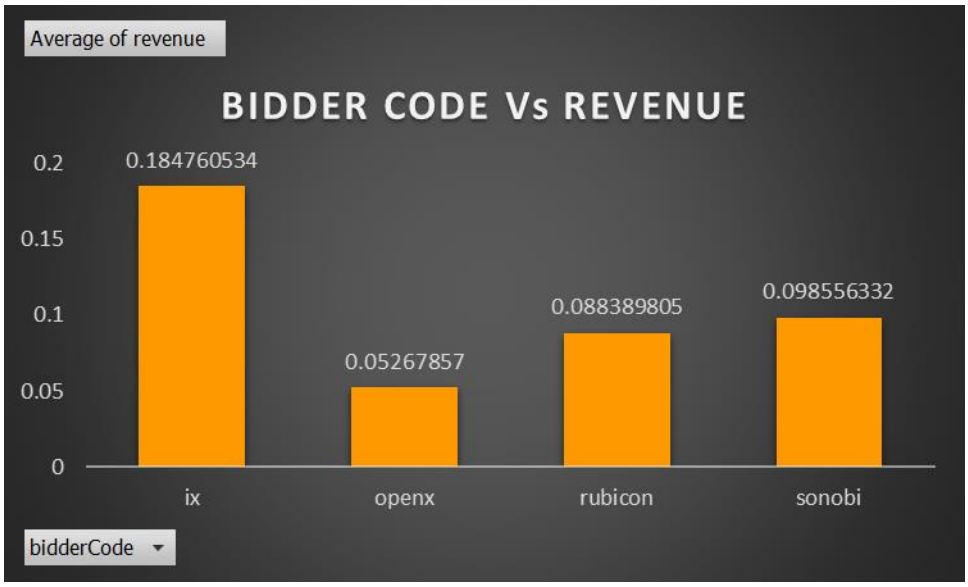
Random Forest Classifier

Random forest is a supervised learning algorithm. Random forest classifier creates a set of decision trees from randomly selected subset of training set. It aggregates the votes from different decision trees to decide the final class of the test object. The basic parameters of Random Forest Classifier are total number of trees to be generated and decision tree related parameters like minimum split, split criteria. It can be used for both classification and regression problem. In our data we trained our model by “bagging” method. By bagging method, we can increase accuracy by combination of learning models. Random Forest give more randomness to the model, while growing the trees. Another use of Random Forest Classifier is that it gives the relative important variables to prediction. We can also observe which of the variables less contribute to the model or does not contribute to the overall prediction and we can drop that variable also.

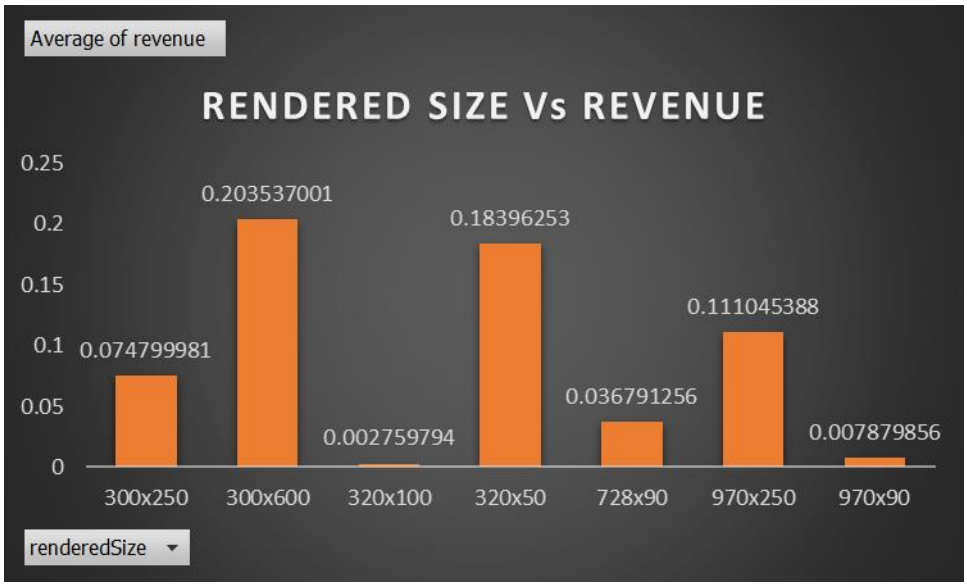
DATA VISUALIZATION



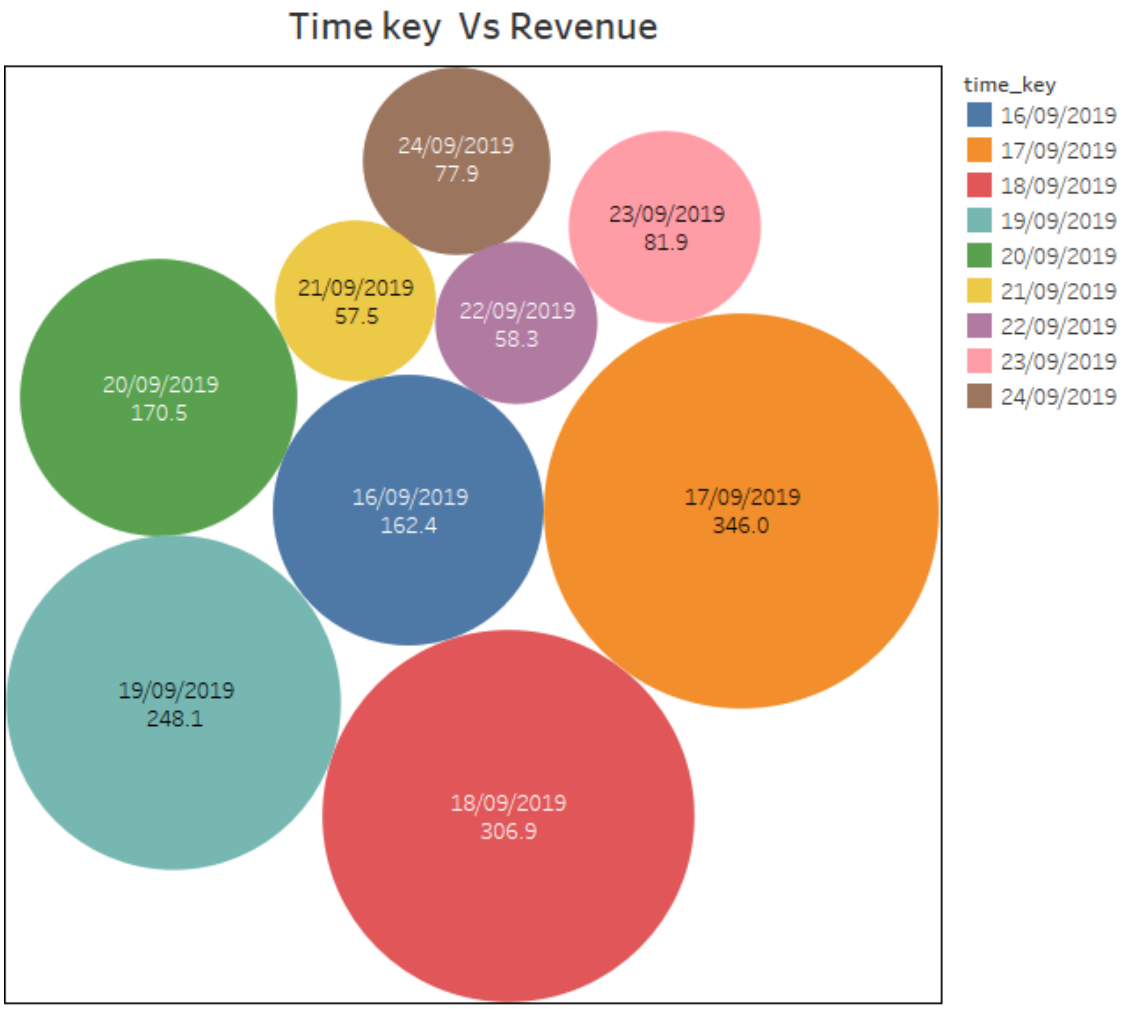
Interpretation: From above graph we can say that contribution of desktop device type is more than mobile device type, hence we gain more revenue through desktop device type.



Interpretation: In above graph, ix bidder code gives more contribution, whereas rubion & sonobi gives approximately same contribution and openx gives lowest contribution, Hence, we conclude that, by using ix bidder code we can gain more revenue.



Interpretation: In above graph, from all 7 rendered size 300×600 & 320×50 are contributed approximately same as compared to others. Hence, we conclude that 300×600 & 320×50, these sizes of banner gives more revenue.



Time_key and sum of revenue. Color shows details about time_key. Size shows sum of revenue. The marks are labeled by time_key and sum of revenue.

Interpretation: From above graph we can see that revenue is high on 17/09/2019 (Tuesday) which is 346.0. Where on date 21/09/2019 (Saturday) revenue is less which is 57.5.

CHI-SQUARE TEST

#chisq test

- **Biddercode & device type:**

H0: The two variables biddercode and device type are independent.

Vs

H1: The two variables biddercode and device type are dependent.

```
> chisq.test(Data$bidderCode,Data$device_type,correct = FALSE)

        Pearson's Chi-squared test

data:  Data$bidderCode and Data$device_type
X-squared = 282.44, df = 3, p-value =2.2e-16
```

Decision Rule: Here p value = 2.26e-16 is less than l.o.s =0.05 then We reject H0 at 5% l.o.s.

Conclusion: we reject H0 since given evidence is enough to conclude that bidder code and Device type are dependent.

- **Bidder code and rendered size:**

H0: The two variables biddercode and rendered size are independent.

Vs

H1: The two variables biddercode and rendered size are dependent.

```
> chisq.test(Data$bidderCode,Data$renderedSize,correct = FALSE)

        Pearson's Chi-squared test

data:  Data$bidderCode and Data$renderedSize
X-squared = 1646.5, df = 18, p-value = 2.2e-16
```

Decision Rule: Here p value = 2.26e-16 is less than l.o.s =0.05 then

We reject H0 at 5% l.o.s.

Conclusion: we reject H0 since given evidence is enough to conclude that bidder code and Rendered size are dependent.

- **Device type and rendered size :**

H0: The two variables device_type and rendered size are independent.

Vs

H1: The two variables device_type and rendered size are dependent.

```
> chisq.test(Data$device_type,Data$renderedSize,correct = FALSE)

        Pearson's Chi-squared test

data:  Data$device_type and Data$renderedSize
X-squared = 3127.1, df = 6, p-value =2.2e-16
```

Decision Rule: Here p value = 2.26e-16 is less than l.o.s =0.05 then We reject H0 at 5% l.o.s.

Conclusion: we reject H0 since given evidence is enough to conclude that device type and Rendered size are dependent.

Correlation between Continuous Variable:

	Request	**Impression**	**Bids count**	**cpm**	**revenue**
Request	1.0000	0.704978458	0.93591396	0.010587433	0.66594747
Impression	0.70497846	1.0000000	0.78533406	0.004143869	0.92718888
Bids count	0.93591396	0.785334061	1.0000	0.074744273	0.80280169
Cpm	0.01058743	-0.00414863	0.07474427	1.0000000	0.09426785
revenue	0.66594747	0.927188880	0.828169	0.09426852	1.0000000

```
>
```

MULTIPLE LINEAR REGRESSION

```
> Data <- read.csv("C:\\Users\\kardi\\OneDrive\\Desktop\\final project\\FINAL DATA FOR PROJECT.csv")
> str(Data)
'data.frame':          15436 obs. of  13 variables:
 $ time_key      : Factor w/ 9 levels "16/09/2019","17/09/2019",..: 1 1 1 1 1 1 1 1 1 ...
 $ bidderCode    : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ currency      : Factor w/ 1 level "USD": 1 1 1 1 1 1 1 1 1 ...
 $ publisher_name: Factor w/ 1 level "TOI": 1 1 1 1 1 1 1 1 1 ...
 $ mediaType     : Factor w/ 1 level "banner": 1 1 1 1 1 1 1 1 1 ...
 $ device_type   : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 ...
 $ host          : Factor w/ 1 level "timesofindia.com": 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize  : Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
 $ Request       : int  18 42 42 6 12 18 30 18 0 78 ...
 $ Impression    : int  2 5 4 1 2 3 5 2 1 1 ...
 $ bids_count    : int  3 29 15 1 3 7 11 10 0 48 ...
 $ cpm           : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue       : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
> #Some variables are categorical therefore we convert them to factor
> data_factor <- as.data.frame(lapply(Data[,c(2:8)],factor))
> NewData <- data.frame(Data[,-c(2:8)],data_factor)
> str(NewData)
'data.frame':          15436 obs. of  13 variables:
 $ time_key      : Factor w/ 9 levels "16/09/2019","17/09/2019",..: 1 1 1 1 1 1 1 1 1 ...
 $ Request       : int  18 42 42 6 12 18 30 18 0 78 ...
 $ Impression    : int  2 5 4 1 2 3 5 2 1 1 ...
 $ bids_count    : int  3 29 15 1 3 7 11 10 0 48 ...
 $ cpm           : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue       : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
 $ bidderCode    : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ currency      : Factor w/ 1 level "USD": 1 1 1 1 1 1 1 1 1 ...
 $ publisher_name: Factor w/ 1 level "TOI": 1 1 1 1 1 1 1 1 1 ...
 $ mediaType     : Factor w/ 1 level "banner": 1 1 1 1 1 1 1 1 1 ...
 $ device_type   : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 ...
 $ host          : Factor w/ 1 level "timesofindia.com": 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize  : Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
> #We have to remove variables which have only one level
> NewData <- NewData[,-c(1,8:10,12)]
> str(NewData)
'data.frame':          15436 obs. of  8 variables:
 $ Request     : int  18 42 42 6 12 18 30 18 0 78 ...
 $ Impression  : int  2 5 4 1 2 3 5 2 1 1 ...
 $ bids_count  : int  3 29 15 1 3 7 11 10 0 48 ...
 $ cpm         : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue     : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
 $ bidderCode  : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ device_type : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize: Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
```

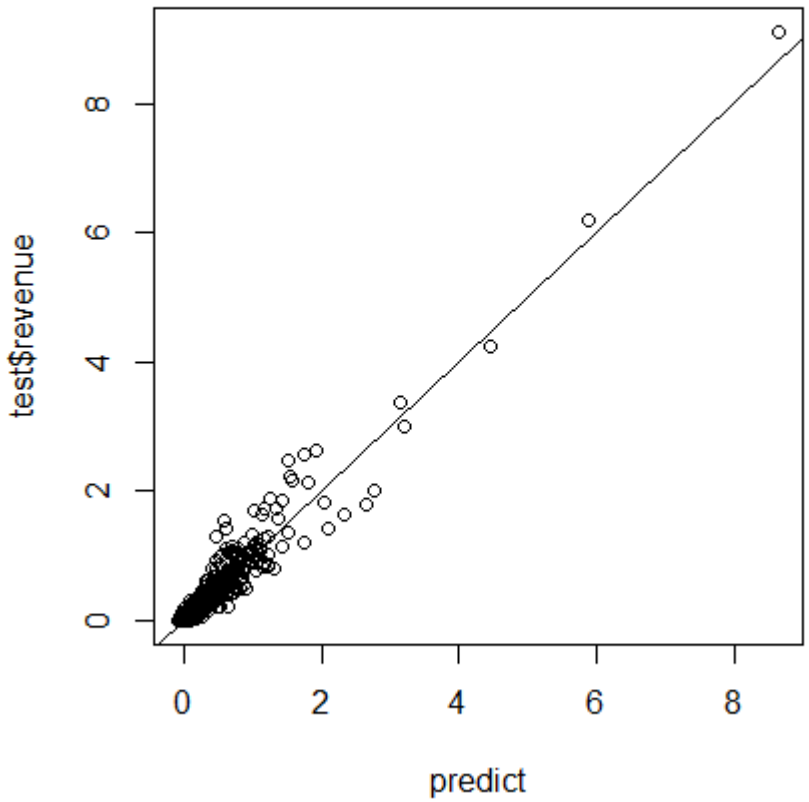
```
> summary(NewData)
   Request         Impression       bids_count          cpm              revenue          bidderCode
 Min.   :   0.0   Min.   :  1.00   Min.   :  0.0   Min.   : 0.100   Min.   :0.000100   ix     :2249
 1st Qu.:  46.0   1st Qu.:  2.00   1st Qu.:  13.0   1st Qu.: 1.333   1st Qu.:0.002968   openx  :36
45
 Median :  176.0   Median :  5.00   Median :  54.0   Median : 2.167   Median :0.012286   rubi
con:3782
 Mean   :  634.5   Mean   :  34.86   Mean   :  190.9   Mean   : 2.840   Mean   :0.097798   sonobi
:5760
 3rd Qu.:  588.0   3rd Qu.:  23.00   3rd Qu.:  198.0   3rd Qu.: 3.542   3rd Qu.:0.065918
 Max.   :27809.0   Max.   :3521.00   Max.   :9427.0   Max.   :69.080   Max.   :9.106830

  device_type     renderedSize
 Desktop:10684   300x250:8728
 Mobile : 4752   300x600:1826
                 320x100:  97
                 320x50 :1453
                 728x90 :1550
                 970x250:1419
                 970x90 : 363
> NewData <- na.omit(NewData)
> sum(is.na(NewData))
[1] 0

> #splitting of the observation to 80% train and 20% for test
> set.seed(22)   #To get unique random samples
> final_data <- NewData[sample(nrow(NewData)),]
> train <- final_data[1:as.integer(0.8*nrow(final_data)),]
> test <- final_data[-c(1:as.integer(0.8*nrow(final_data))),]
> sum(is.na(NewData))    #There is no missing value in our data
[1] 0

> summary(Data$revenue)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.000100 0.002968 0.012286 0.097798 0.065918 9.106830

> ##Building multiple regression model

> model_lm <- lm(revenue~.,data=train)
> summary(model_lm)

Call:
lm(formula = revenue ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-1.27716 -0.01849 -0.00101  0.01258  1.70054

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)        -1.006e-02  2.700e-03  -3.724 0.000197 ***
Request            -9.961e-05  1.706e-06 -58.395  < 2e-16 ***
Impression          2.058e-03  1.306e-05 157.595  < 2e-16 ***
bids_count          4.883e-04  7.004e-06  69.707  < 2e-16 ***
cpm                5.826e-03  3.141e-04  18.550  < 2e-16 ***
bidderCodeopenx    -1.580e-02  2.779e-03  -5.686 1.33e-08 ***
bidderCoderubicon  -6.776e-03  2.826e-03  -2.398 0.016518 *
bidderCodesonobi    4.833e-03  2.616e-03   1.847 0.064700 .
device_typeMobile   6.644e-04  1.994e-03   0.333 0.739025
renderedSize300x600  1.294e-02  2.713e-03   4.768 1.88e-06 ***
renderedSize320x100 -7.306e-03  1.063e-02  -0.687 0.491923
renderedSize320x50  -6.551e-02  3.119e-03 -21.007  < 2e-16 ***
renderedSize728x90  -1.286e-02  2.832e-03  -4.540 5.68e-06 ***
renderedSize970x250 -1.149e-02  2.941e-03  -3.907 9.40e-05 ***
renderedSize970x90  -1.256e-02  5.433e-03  -2.311 0.020825 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.08939 on 12333 degrees of freedom
Multiple R-squared:  0.9067,            Adjusted R-squared:  0.9066
F-statistic:  8562 on 14 and 12333 DF,  p-value: < 2.2e-16
```

```
> vif(model_lm)

                GVIF Df GVIF^(1/(2*Df))
Request       9.028383  1       3.004727
Impression    2.948996  1       1.717264
bids_count   12.208453  1       3.494060
cpm           1.113727  1       1.055333
bidderCode    1.230728  3       1.035207
device_type   1.308470  1       1.143884
renderedSize  1.654700  6       1.042861
```

```
> predict <- predict(model_lm,newdata=test[,-5])      #5th variable is response(revenue)

> mean((predict-test$revenue)^2)            #MSE

[1] 0.006570432
```

```
> plot(predict,test$revenue)
> abline(0,1)
```




```
sqrt(mse)

[1] 0.0810582
```

Interpretation: Here variation inflation factor of bidscount & Request is larger than 5, since multicolinearity is present in the data. Hence we reduce multicolinearity and again fit the model.

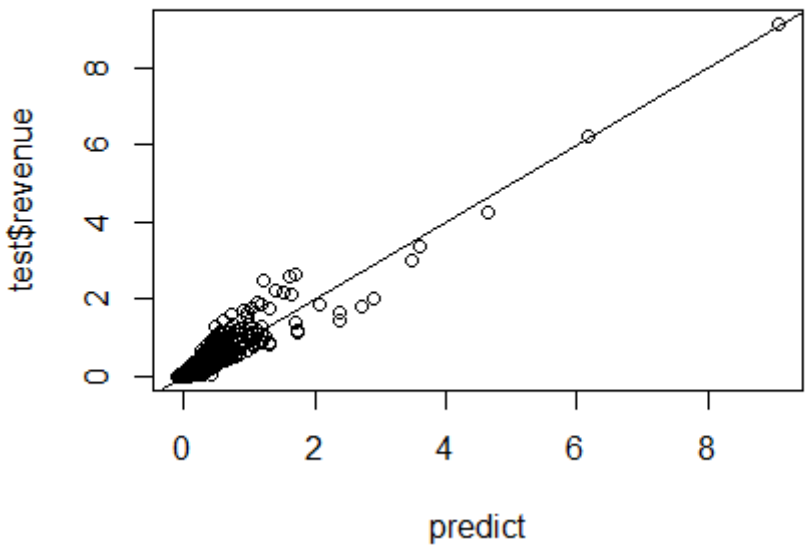
- For reducing multicolinearity we takecombination of two variables so we take percentage of bidscount & Request that means we take percentage of how many people are requested and from that how many are click on that advertisement.
- For reducing multicolinearity we again fit the model. We take one variable percentage means it is percentage = (bidscount/request)*100.

```
> Data <- read.csv("C:\\Users\\kardi\\OneDrive\\Desktop\\final project\\New data nnnn.csv")
> str(Data)
'data.frame':          15436 obs. of  12 variables:
 $ time_key      : Factor w/ 9 levels "16/09/2019","17/09/2019",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
 $ bidderCode    : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ currency      : Factor w/ 1 level "USD": 1 1 1 1 1 1 1 1 1 1 ...
 $ publisher_name: Factor w/ 1 level "TOI": 1 1 1 1 1 1 1 1 1 1 ...
 $ mediaType     : Factor w/ 1 level "banner": 1 1 1 1 1 1 1 1 1 1 ...
 $ device_type   : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 1 ...
 $ host          : Factor w/ 1 level "timesofindia.com": 1 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize  : Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
 $ Impression    : int  2 5 4 1 2 3 5 2 1 1 ...
 $ cpm           : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue       : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
 $ percentage    : num  21.1 69.8 37.2 28.6 30.8 ...
> #Some variables are categorical therefore we convert them to factor
> data_factor <- as.data.frame(lapply(Data[,c(2:8)],factor))
> NewData <- data.frame(Data[,-c(2:8)],data_factor)
> str(NewData)
'data.frame':           15436 obs. of  12 variables:
 $ time_key      : Factor w/ 9 levels "16/09/2019","17/09/2019",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Impression    : int  2 5 4 1 2 3 5 2 1 1 ...
 $ cpm           : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue       : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
 $ percentage    : num  21.1 69.8 37.2 28.6 30.8 ...
 $ bidderCode    : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ currency      : Factor w/ 1 level "USD": 1 1 1 1 1 1 1 1 1 1 ...
 $ publisher_name: Factor w/ 1 level "TOI": 1 1 1 1 1 1 1 1 1 1 ...
 $ mediaType     : Factor w/ 1 level "banner": 1 1 1 1 1 1 1 1 1 1 ...
 $ device_type   : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 1 ...
 $ host          : Factor w/ 1 level "timesofindia.com": 1 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize  : Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
> #We have to remove variables which have only one level
> NewData <- NewData[,-c(1,7:9,11)]
> str(NewData)
'data.frame':           15436 obs. of  7 variables:
 $ Impression  : int  2 5 4 1 2 3 5 2 1 1 ...
 $ cpm         : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue     : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
 $ percentage  : num  21.1 69.8 37.2 28.6 30.8 ...
 $ bidderCode  : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ device_type : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize: Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
> summary(NewData)
   Impression          cpm              revenue            percentage       bidderCode     device_type
 Min.   :   1.00   Min.   : 0.100   Min.   :0.000100   Min.   :  3.03   ix     :2249   Desktop:10684
 1st Qu.:   2.00   1st Qu.: 1.333   1st Qu.:0.002968   1st Qu.: 22.14   openx  :3645   Mobile : 4752
 Median :   5.00   Median : 2.167   Median :0.012286   Median : 31.91   rubicon:3782
 Mean   :  34.86   Mean   : 2.840   Mean   :0.097798   Mean   : 35.47   sonobi :5760
 3rd Qu.:  23.00   3rd Qu.: 3.542   3rd Qu.:0.065918   3rd Qu.: 43.16
 Max.   :3521.00   Max.   :69.080   Max.   :9.106830   Max.   :100.00

  renderedSize
 300x250:8728
```

```
 300x600:1826
 320x100:  97
 320x50 :1453
 728x90 :1550
 970x250:1419
 970x90 : 363
> NewData <- na.omit(NewData)
> sum(is.na(NewData))
[1] 0
> #splitting of the observation to 80% train and 20% for test
> set.seed(22)   #To get unique random samples
> final_data <- NewData[sample(nrow(NewData)),]
> train <- final_data[1:as.integer(0.8*nrow(final_data)),]
> str(train)
'data.frame':          12348 obs. of  7 variables:
 $ Impression  : int  1 251 34 9 7 1 3 45 6 1 ...
 $ cpm         : num  5.45 1.62 1.19 3.06 6.83 ...
 $ revenue     : num  0.00545 0.40542 0.04057 0.02758 0.0478 ...
 $ percentage  : num  34.62 7.15 26.2 42.97 42.97 ...
 $ bidderCode  : Factor w/ 4 levels "ix","openx","rubicon",..: 3 3 4 4 2 1 3 4 4 3 ...
 $ device_type : Factor w/ 2 levels "Desktop","Mobile": 2 2 1 1 1 1 1 1 2 1 ...
 $ renderedSize: Factor w/ 7 levels "300x250","300x600",..: 1 1 1 1 5 1 5 2 1 1 ...
> test <- final_data[-c(1:as.integer(0.8*nrow(final_data))),]
> str(test)
'data.frame':          3088 obs. of  7 variables:
 $ Impression  : int  1 7 1 8 49 344 7 1 8 2 ...
 $ cpm         : num  3.191 0.944 4.16 0.807 1.876 ...
 $ revenue     : num  0.00319 0.00661 0.00416 0.00646 0.09191 ...
 $ percentage  : num  39.1 37.2 45 23.7 33.4 ...
 $ bidderCode  : Factor w/ 4 levels "ix","openx","rubicon",..: 3 4 2 4 4 1 4 4 4 4 ...
 $ device_type : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 2 1 1 2 1 2 ...
 $ renderedSize: Factor w/ 7 levels "300x250","300x600",..: 1 1 1 1 4 1 1 4 1 1 ...
> sum(is.na(NewData))    #There is no missing value in our data
[1] 0
> ##Building multiple regression model
> model_lm <- lm(revenue~.,data=train)
> summary(model_lm)

Call:
lm(formula = revenue ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-1.14560 -0.02637 -0.00431  0.01657  2.13778

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    9.424e-03  3.595e-03  -2.622  0.00876 **
Impression      2.573e-03  9.320e-06 276.032  < 2e-16 ***
cpm            8.555e-03  3.799e-04  22.522  < 2e-16 ***
```

```
percentage          4.190e-04  5.151e-05   8.134 4.56e-16 ***
bidderCodeopenx     -2.668e-02  3.271e-03  -8.155 3.81e-16 ***
bidderCoderubicon   -3.351e-02  3.293e-03 -10.177  < 2e-16 ***
bidderCodesonobi    -5.305e-04  3.085e-03  -0.172  0.86346
device_typeMobile   -1.792e-02  2.347e-03  -7.635 2.42e-14 ***
renderedSize300x600  2.445e-02  3.171e-03   7.712 1.33e-14 ***
renderedSize320x100 -5.766e-03  1.255e-02  -0.459  0.64593
renderedSize320x50  -5.991e-02  3.687e-03 -16.249  < 2e-16 ***
renderedSize728x90  -6.107e-03  3.342e-03  -1.827  0.06772 .
renderedSize970x250  2.581e-02  3.408e-03   7.573 3.90e-14 ***
renderedSize970x90  -3.289e-02  6.385e-03  -5.151 2.63e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1053 on 12334 degrees of freedom
Multiple R-squared:  0.8704,             Adjusted R-squared:  0.8703
F-statistic:  6374 on 13 and 12334 DF,  p-value: < 2.2e-16


> predict <- predict(model_lm,newdata=test[,-3])      #3th veriable is response(revenue)
> predict <- as.numeric(predict)
> mse=mean((predict-test$revenue)^2)            #MSE
> mse
[1] 0.008914026
> sqrt(mse)
[1] 0.09441412
> plot(predict,test$revenue)
> abline(0,1)
```



>

Interpretation: From above predict Vs actual graph many points are along the line since model is good fit for the data.

```
> vif(model_lm)
                GVIF Df GVIF^(1/(2*Df))
Impression   1.081160  1        1.039789
cpm          1.173216  1        1.083151
percentage   1.189026  1        1.090425
bidderCode   1.196925  3        1.030413
device_type  1.305362  1        1.142524
renderedSize 1.587875  6        1.039285
```

Interpretation: All variation inflation factors are less than 2 since **multicolinearity is reduced.**

The regression equation is,

Revenue= 9.424e-03 + 2.573e-03 * Impression+ 8.555e-03 cpm + 4.190e-04 *percentage+
(-2.668e-02) bidderCodeopenx+
(-3.351e-02) bidderCoderubicon + (-5.305e-04) bidderCodesonobi +
(-1.792e-02) device_typeMobile+
(2.445e-02) renderedSize300x600 + (-5.766e-03) renderedSize320x100+
(-5.991e-02) renderedSize320x50 +
(-6.107e-03) renderedSize728x90+ (2.581e-02) renderedSize970x250 +
(-3.289e-02) renderedSize970x90.

Here, adjusted R sq is 0.8704 means 87.04% variation in revenue is explained by variation in Percentage ((bidscount/request)*100) , biddercode ,rendered size and device type.

#ANOVA
Here regression df is 13 and residual df is 12334.

H0: Regression model is insignificant.
Vs
H1: Regression model is significant.

Here pvalue = 2.2e-16 <0.05
Hence, we reject H0 at 5% l.o.s

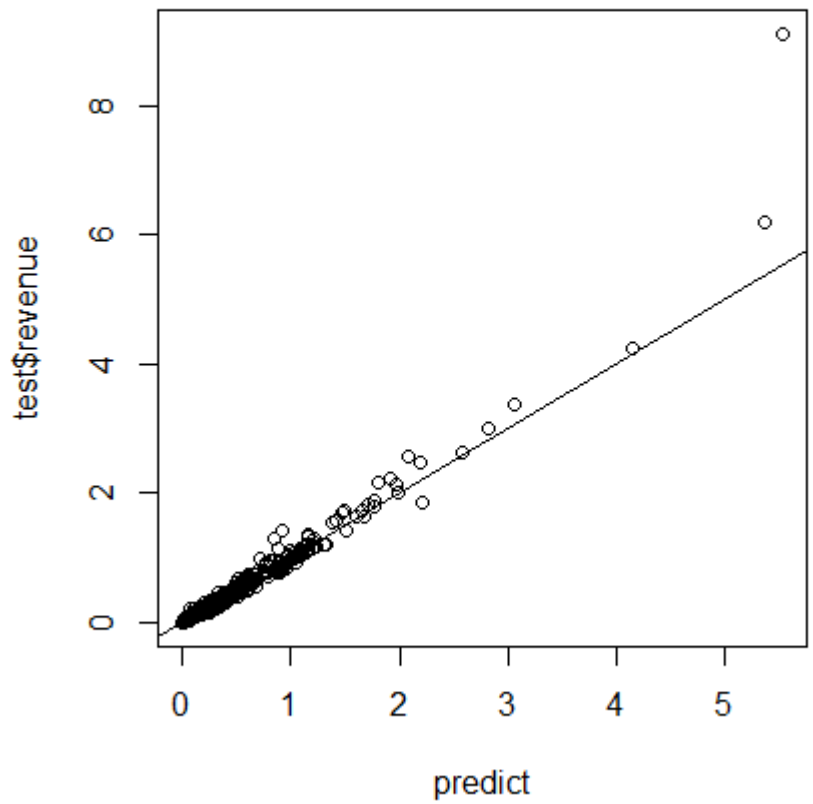
Conclusion: Here we reject H0, that regression model is significant.
We can say that regression **model is good fit for data.**

#MSE =0.008914026 & RMSE = 0.09441412
Here MSE of this model is 0.008914026 which is less since our model is good fit for data.

RANDOM FOREST

```
Data <- read.csv("C:\\Users\\kardi\\OneDrive\\Desktop\\final project\\FINAL DATA FOR P
ROJECT.csv")
> str(Data)
'data.frame':          15436 obs. of  13 variables:
 $ time_key      : Factor w/ 9 levels "16/09/2019","17/09/2019",..: 1 1 1 1 1 1 1 1 1 ...
 $ bidderCode    : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ currency      : Factor w/ 1 level "USD": 1 1 1 1 1 1 1 1 1 ...
 $ publisher_name: Factor w/ 1 level "TOI": 1 1 1 1 1 1 1 1 1 ...
 $ mediaType     : Factor w/ 1 level "banner": 1 1 1 1 1 1 1 1 1 ...
 $ device_type   : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 ...
 $ host          : Factor w/ 1 level "timesofindia.com": 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize  : Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
 $ Request       : int  18 42 42 6 12 18 30 18 0 78 ...
 $ Impression    : int  2 5 4 1 2 3 5 2 1 1 ...
 $ bids_count    : int  3 29 15 1 3 7 11 10 0 48 ...
 $ cpm           : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue       : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
> #Some variables are categorical therefore we convert them to factor
> data_factor <- as.data.frame(lapply(Data[,c(2:8)],factor))
> NewData <- data.frame(Data[,-c(2:8)],data_factor)
> str(NewData)
'data.frame':          15436 obs. of  13 variables:
 $ time_key      : Factor w/ 9 levels "16/09/2019","17/09/2019",..: 1 1 1 1 1 1 1 1 1 ...
 $ Request       : int  18 42 42 6 12 18 30 18 0 78 ...
 $ Impression    : int  2 5 4 1 2 3 5 2 1 1 ...
 $ bids_count    : int  3 29 15 1 3 7 11 10 0 48 ...
 $ cpm           : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue       : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
 $ bidderCode    : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ currency      : Factor w/ 1 level "USD": 1 1 1 1 1 1 1 1 1 ...
 $ publisher_name: Factor w/ 1 level "TOI": 1 1 1 1 1 1 1 1 1 ...
 $ mediaType     : Factor w/ 1 level "banner": 1 1 1 1 1 1 1 1 1 ...
 $ device_type   : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 ...
 $ host          : Factor w/ 1 level "timesofindia.com": 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize  : Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
> #We have to remove variables which have only one level
> NewData <- NewData[,-c(1,8:10,12)]
> str(NewData)
'data.frame':          15436 obs. of  8 variables:
 $ Request     : int  18 42 42 6 12 18 30 18 0 78 ...
 $ Impression  : int  2 5 4 1 2 3 5 2 1 1 ...
 $ bids_count  : int  3 29 15 1 3 7 11 10 0 48 ...
 $ cpm         : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue     : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
```

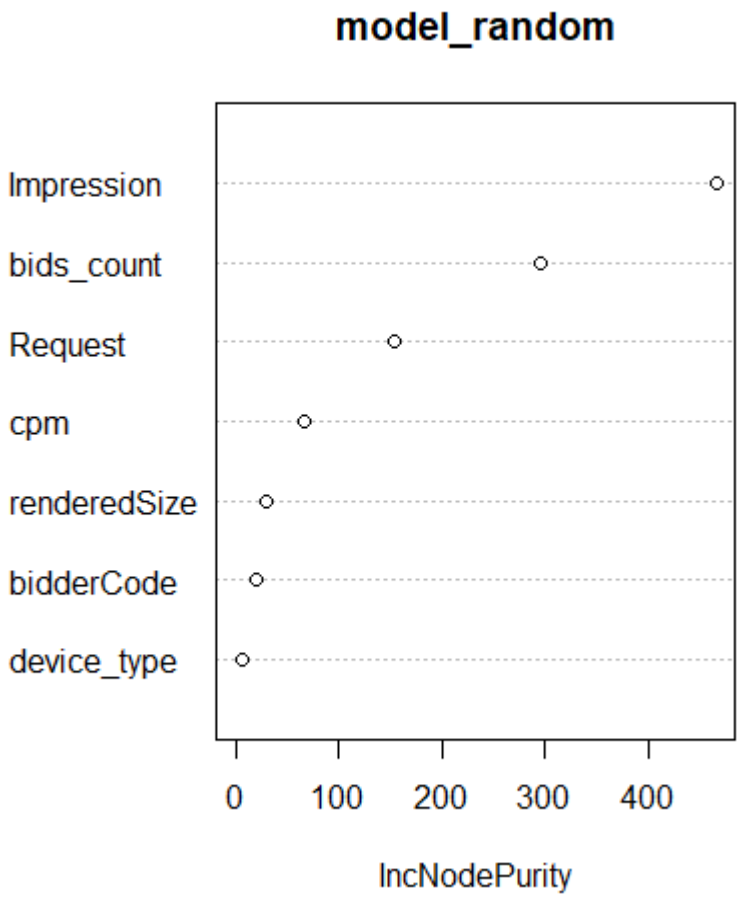
```
 $ bidderCode  : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ device_type : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize: Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
> summary(NewData)
   Request         Impression       bids_count          cpm              revenue          bidderCode
 Min.   :   0.0   Min.   :  1.00   Min.   :   0.0   Min.   : 0.100   Min.   :0.000100   ix    :2249
 1st Qu.:  46.0   1st Qu.:  2.00   1st Qu.:  13.0   1st Qu.: 1.333   1st Qu.:0.002968   openx :3
645
 Median :  176.0   Median :  5.00   Median :  54.0   Median : 2.167   Median :0.012286   rub
icon:3782
 Mean   :  634.5   Mean   :  34.86   Mean   :  190.9   Mean   : 2.840   Mean   :0.097798   sonob
i :5760
 3rd Qu.:  588.0   3rd Qu.:  23.00   3rd Qu.:  198.0   3rd Qu.: 3.542   3rd Qu.:0.065918
 Max.   :27809.0   Max.   :3521.00   Max.   :9427.0   Max.   :69.080   Max.   :9.106830

  device_type     renderedSize
 Desktop:10684   300x250:8728
 Mobile : 4752   300x600:1826
                 320x100:  97
                 320x50 :1453
                 728x90 :1550
                 970x250:1419
                 970x90 : 363
> NewData <- na.omit(NewData)
> sum(is.na(NewData))
[1] 0
> #splitting of the observation to 80% train and 20% for test
> set.seed(22)   #To get unique random samples
> final_data <- NewData[sample(nrow(NewData)),]
> train <- final_data[1:as.integer(0.8*nrow(final_data)),]
> test <- final_data[-c(1:as.integer(0.8*nrow(final_data))),]
> sum(is.na(NewData))    #There is no missing value in our data
[1] 0
> summary(Data$revenue)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.000100 0.002968 0.012286 0.097798 0.065918 9.106830
> #Building Random Forest Model
> library(randomForest)
> model_random <- randomForest(revenue~.,data=train)
> predict <- predict(model_random,newdata = test[,-5])
> mean((predict-test$revenue)^2)         #MSE
[1] 0.005223723

> plot(predict,test$revenue)
> abline(0,1)
```



Interpretation: From above plot we can see that many points are along the line since our random forest model is good fit for the data.

```
>
 importance(model_random)          #Variable importance
              IncNodePurity
 Request         152.875407
 Impression       465.572074
 bids_count       296.026736
 cpm             66.832335
 bidderCode        19.941530
 device_type        6.410859
 renderedSize      29.357386
```

```
> varImpPlot(model_random)
```



Interpretation: In the random forest model we see that Impression is the most important factor that affect on the Revenue where bids_count is the second most important factor that affect on the revenue. Device type is less important variable that affect to increase the revenue.

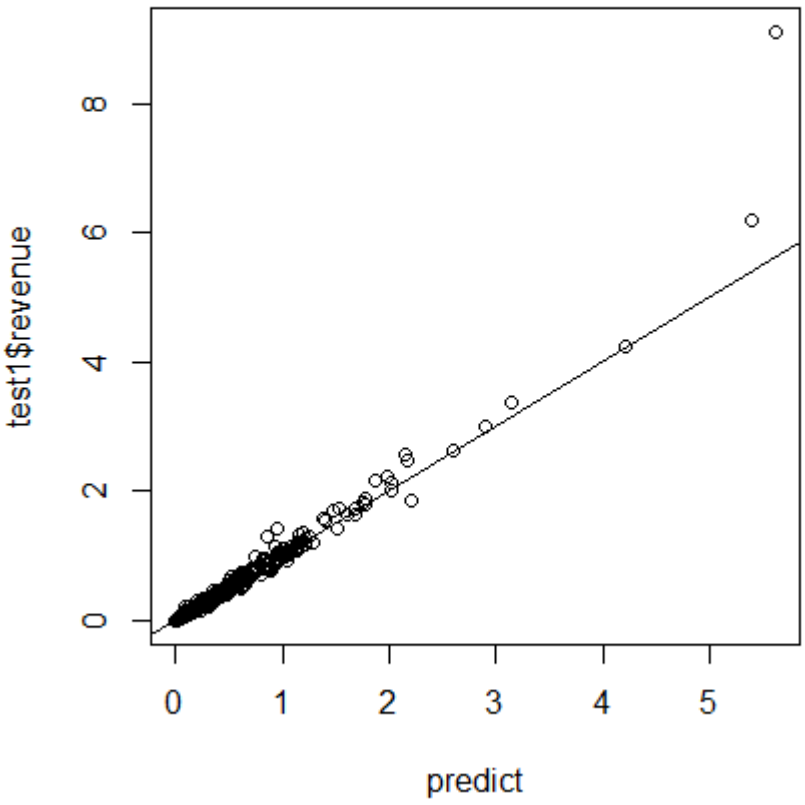
Since delete less important variable and again fit random forest model

```
#Deleting less important variables from the data
> train1 <- train[,-c(7)]
> test1 <- test[,-c(7)]
> model_random <- randomForest(revenue~.,data=train1)
> predict <- predict(model_random,newdata = test1[,-5])
> mean((predict-test1$revenue)^2) #MSE
```

[1] 0.004806483
sqrt(mse)
[1] 0.0693288

> plot(predict,test1$revenue)
> abline(0,1)

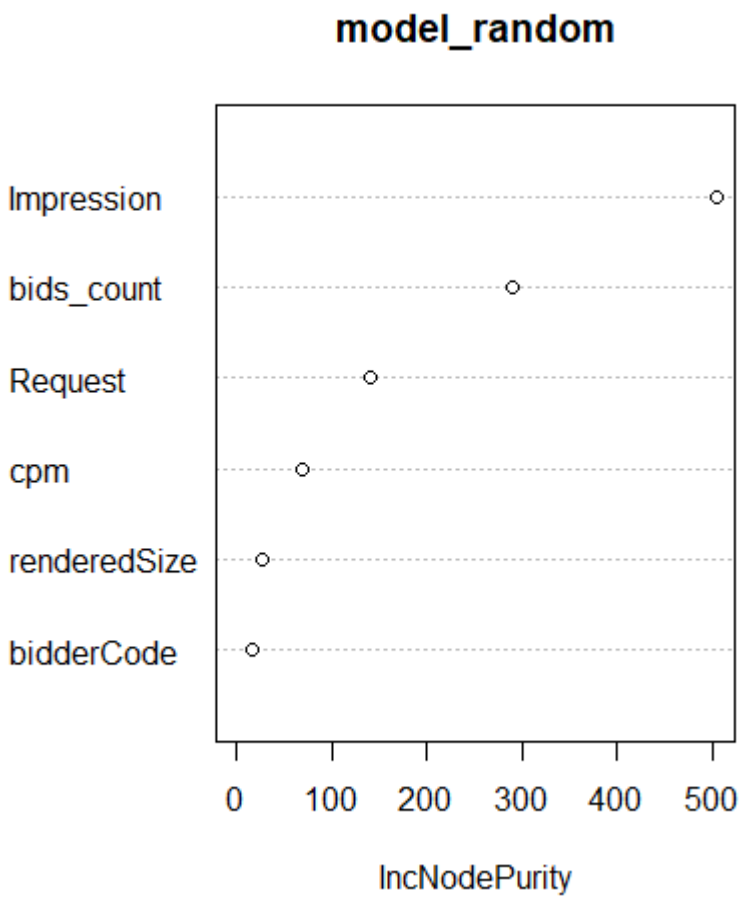
>



Interpretation: From above plot we can see that many points are along the line since our random forest model is good fit for the data

varImpPlot(model_random)

>



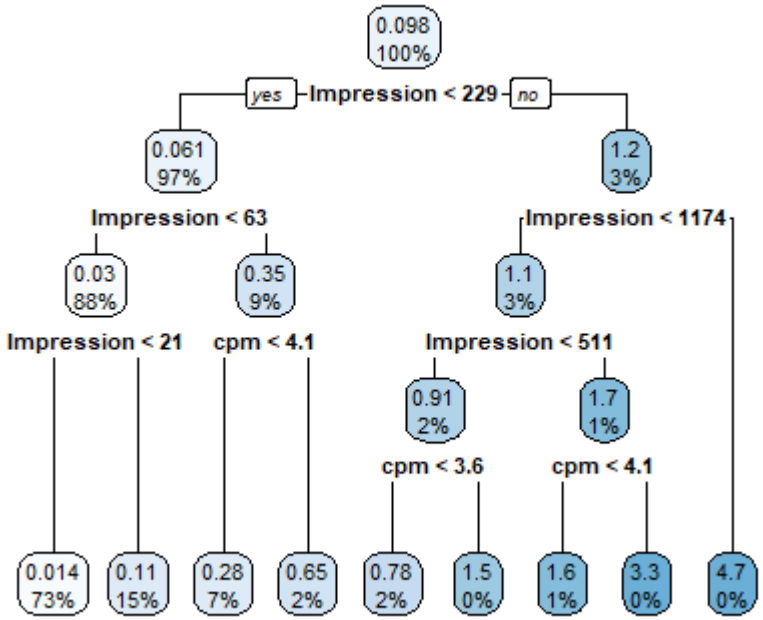
Interpretation: After deleting less important variable we get less MSE i.e 0.004806 according to the previous MSE i.e 0.005223.

Here MSE is less since our model is good fit for the data.

DECISION TREE

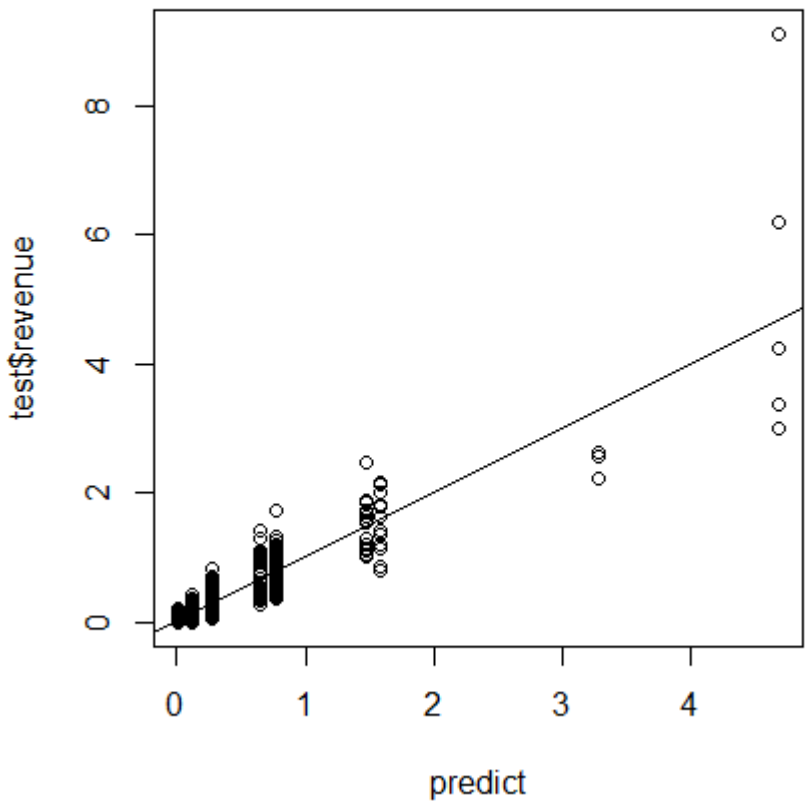
```
Data <- read.csv("C:\\Users\\kardi\\OneDrive\\Desktop\\FINAL DATA FOR PROJECT.csv"
)
> View(Data)
> str(Data)
'data.frame':          15436 obs. of  13 variables:
 $ time_key      : Factor w/ 9 levels "16/09/2019","17/09/2019",..: 1 1 1 1 1 1 1 1 1 ...
 $ bidderCode    : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ currency      : Factor w/ 1 level "USD": 1 1 1 1 1 1 1 1 1 ...
 $ publisher_name: Factor w/ 1 level "TOI": 1 1 1 1 1 1 1 1 1 ...
 $ mediaType     : Factor w/ 1 level "banner": 1 1 1 1 1 1 1 1 1 ...
 $ device_type   : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 ...
 $ host          : Factor w/ 1 level "timesofindia.com": 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize  : Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
 $ Request       : int  18 42 42 6 12 18 30 18 0 78 ...
 $ Impression    : int  2 5 4 1 2 3 5 2 1 1 ...
 $ bids_count    : int  3 29 15 1 3 7 11 10 0 48 ...
 $ cpm           : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue       : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
> #Some variables are categorical therefore we conevert them to factor
> data_factor <- as.data.frame(lapply(Data[,c(2:8)],factor))
> NewData <- data.frame(Data[,-c(2:8)],data_factor)
> str(NewData)
'data.frame':          15436 obs. of  13 variables:
 $ time_key      : Factor w/ 9 levels "16/09/2019","17/09/2019",..: 1 1 1 1 1 1 1 1 1 ...
 $ Request       : int  18 42 42 6 12 18 30 18 0 78 ...
 $ Impression    : int  2 5 4 1 2 3 5 2 1 1 ...
 $ bids_count    : int  3 29 15 1 3 7 11 10 0 48 ...
 $ cpm           : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue       : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
 $ bidderCode    : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ currency      : Factor w/ 1 level "USD": 1 1 1 1 1 1 1 1 1 ...
 $ publisher_name: Factor w/ 1 level "TOI": 1 1 1 1 1 1 1 1 1 ...
 $ mediaType     : Factor w/ 1 level "banner": 1 1 1 1 1 1 1 1 1 ...
 $ device_type   : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 ...
 $ host          : Factor w/ 1 level "timesofindia.com": 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize  : Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
> #We have to remove veriables which have only one level
> NewData <- NewData[,-c(1,8:10,12)]
> str(NewData)
'data.frame':          15436 obs. of  8 variables:
 $ Request     : int  18 42 42 6 12 18 30 18 0 78 ...
 $ Impression  : int  2 5 4 1 2 3 5 2 1 1 ...
 $ bids_count  : int  3 29 15 1 3 7 11 10 0 48 ...
 $ cpm         : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue     : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
 $ bidderCode  : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
```

```
 $ device_type : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize: Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
> summary(NewData)
   Request         Impression       bids_count          cpm              revenue          bidderCode
 Min.   :    0.0   Min.   :   1.00   Min.   :    0.0   Min.   :  0.100   Min.   :0.000100   ix     :2249
 1st Qu.:   46.0   1st Qu.:   2.00   1st Qu.:   13.0   1st Qu.:  1.333   1st Qu.:0.002968   openx  :3645
 Median :  176.0   Median :   5.00   Median :   54.0   Median :  2.167   Median :0.012286   rubicon:3782
 Mean   :  634.5   Mean   :  34.86   Mean   :  190.9   Mean   :  2.840   Mean   :0.097798   sonobi :5760
 3rd Qu.:  588.0   3rd Qu.:  23.00   3rd Qu.:  198.0   3rd Qu.:  3.542   3rd Qu.:0.065918
 Max.   :27809.0   Max.   :3521.00   Max.   :9427.0    Max.   :  69.080   Max.   :9.106830

  device_type     renderedSize
 Desktop:10684   300x250:8728
 Mobile : 4752   300x600:1826
                 320x100:  97
                 320x50 :1453
                 728x90 :1550
                 970x250:1419
                 970x90 : 363
> NewData <- na.omit(NewData)
> sum(is.na(NewData))
[1] 0
> #splitting of the observation to 80% train and 20% for test
> set.seed(22)   #To get unique random samples
> final_data <- NewData[sample(nrow(NewData)),]
> train <- final_data[1:as.integer(0.8*nrow(final_data)),]
> test <- final_data[-c(1:as.integer(0.8*nrow(final_data))),]
> sum(is.na(NewData))    #There is no missing value in our data
[1] 0


fit=rpart(revenue~.,data=train,method='anova')

rpart.plot(fit)
```



Interpretation: From above tree we can see that variable **impression** is at top means at root node so it gain more information about data and **cpm** is another variable that is at leaf node means is also important variable that gain more information about data.

```
rpart.plot(fit)
> str(train)
'data.frame':          12348 obs. of  8 variables:
 $ Request     : int  25 9663 2106 390 262 89 209 318 138 0 ...
 $ Impression  : int  1 251 34 9 7 1 3 45 6 1 ...
 $ bids_count  : int  8 690 551 167 112 34 10 246 23 0 ...
 $ cpm         : num  5.45 1.62 1.19 3.06 6.83 ...
 $ revenue     : num  0.00545 0.40542 0.04057 0.02758 0.0478 ...
 $ bidderCode  : Factor w/ 4 levels "ix","openx","rubicon",..: 3 3 4 4 2 1 3 4 4 3 ...
 $ device_type : Factor w/ 2 levels "Desktop","Mobile": 2 2 1 1 1 1 1 1 2 1 ...
 $ renderedSize: Factor w/ 7 levels "300x250","300x600",..: 1 1 1 1 5 1 5 2 1 1 ...
> predict=predict(fit,newdata=test[,-5],type='matrix')
> mean((predict-test$revenue)^2)
[1] 0.01563719

sqrt(mse)
[1] 0.1250488

> plot(predict,test$revenue)#MSE
> abline(0,1)



>
```



Interpretation: From above plot we can say that many points are of some distance from Straight line since this model is not much good fit for the data**.**

And here MSE =0.01563719 & RMSE= 0.1250488 is somewhat larger than other model Since this model is not much good .

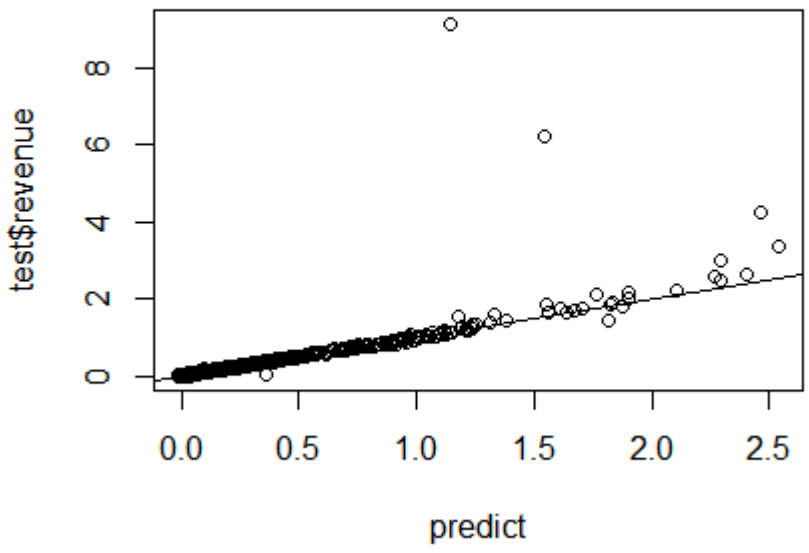
SUPPORT VECTOR REGRESSION

```
> Data <- read.csv("C:\\Users\\kardi\\OneDrive\\Desktop\\FINAL DATA FOR PROJECT.csv",header=TRUE)
> str(Data)
'data.frame':          15436 obs. of  13 variables:
 $ time_key      : Factor w/ 9 levels "16/09/2019","17/09/2019",..: 1 1 1 1 1 1 1 1 1 ...
 $ bidderCode    : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ currency      : Factor w/ 1 level "USD": 1 1 1 1 1 1 1 1 1 ...
 $ publisher_name: Factor w/ 1 level "TOI": 1 1 1 1 1 1 1 1 1 ...
 $ mediaType     : Factor w/ 1 level "banner": 1 1 1 1 1 1 1 1 1 ...
 $ device_type   : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 ...
 $ host          : Factor w/ 1 level "timesofindia.com": 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize  : Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
 $ Request       : int  18 42 42 6 12 18 30 18 0 78 ...
 $ Impression    : int  2 5 4 1 2 3 5 2 1 1 ...
 $ bids_count    : int  3 29 15 1 3 7 11 10 0 48 ...
 $ cpm           : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue       : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
> #Some variables are categorical therefore we conevert them to factor
> data_factor <- as.data.frame(lapply(Data[,c(2:8)],factor))
> NewData <- data.frame(Data[,-c(2:8)],data_factor)
> str(NewData)
'data.frame':          15436 obs. of  13 variables:
 $ time_key      : Factor w/ 9 levels "16/09/2019","17/09/2019",..: 1 1 1 1 1 1 1 1 1 ...
 $ Request       : int  18 42 42 6 12 18 30 18 0 78 ...
 $ Impression    : int  2 5 4 1 2 3 5 2 1 1 ...
 $ bids_count    : int  3 29 15 1 3 7 11 10 0 48 ...
 $ cpm           : num  1.77 2.78 3.05 1.2 3.16 ...
 $ revenue       : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
 $ bidderCode    : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ currency      : Factor w/ 1 level "USD": 1 1 1 1 1 1 1 1 1 ...
 $ publisher_name: Factor w/ 1 level "TOI": 1 1 1 1 1 1 1 1 1 ...
 $ mediaType     : Factor w/ 1 level "banner": 1 1 1 1 1 1 1 1 1 ...
 $ device_type   : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 ...
 $ host          : Factor w/ 1 level "timesofindia.com": 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize  : Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
> #We have to remove veriables which have only one level
> NewData <- NewData[,-c(1,8:10,12)]
> str(NewData)
'data.frame':          15436 obs. of  8 variables:
 $ Request     : int  18 42 42 6 12 18 30 18 0 78 ...
 $ Impression  : int  2 5 4 1 2 3 5 2 1 1 ...
 $ bids_count  : int  3 29 15 1 3 7 11 10 0 48 ...
 $ cpm         : num  1.77 2.78 3.05 1.2 3.16 ...
```

```
 $ revenue     : num  0.00355 0.0139 0.01218 0.0012 0.00632 ...
 $ bidderCode  : Factor w/ 4 levels "ix","openx","rubicon",..: 1 1 2 2 3 3 4 4 4 2 ...
 $ device_type : Factor w/ 2 levels "Desktop","Mobile": 1 1 1 1 1 1 1 1 1 1 ...
 $ renderedSize: Factor w/ 7 levels "300x250","300x600",..: 1 2 1 2 1 2 1 2 1 1 ...
> summary(NewData)
    Request         Impression       bids_count          cpm              revenue          bidderCode
 Min.   :    0.0   Min.   :   1.00   Min.   :    0.0   Min.   :  0.100   Min.   :0.000100   ix    :2249
 1st Qu.:   46.0   1st Qu.:   2.00   1st Qu.:   13.0   1st Qu.:  1.333   1st Qu.:0.002968   openx :3
645
 Median :  176.0   Median :   5.00   Median :   54.0   Median :  2.167   Median :0.012286   rub
icon:3782
 Mean   :  634.5   Mean   :  34.86   Mean   :  190.9   Mean   :  2.840   Mean   :0.097798   sonob
i :5760
 3rd Qu.:  588.0   3rd Qu.:  23.00   3rd Qu.:  198.0   3rd Qu.:  3.542   3rd Qu.:0.065918
 Max.   :27809.0   Max.   :3521.00   Max.   :9427.0    Max.   :  69.080   Max.   :9.106830

  device_type     renderedSize
 Desktop:10684   300x250:8728
 Mobile : 4752   300x600:1826
                 320x100:  97
                 320x50 :1453
                 728x90 :1550
                 970x250:1419
                 970x90 : 363
> NewData <- na.omit(NewData)
> sum(is.na(NewData))
[1] 0
> #splitting of the observation to 80% train and 20% for test
> set.seed(22)   #To get unique random samples
> final_data <- NewData[sample(nrow(NewData)),]
> train <- final_data[1:as.integer(0.8*nrow(final_data)),]
> test <- final_data[-c(1:as.integer(0.8*nrow(final_data))),]
> sum(is.na(NewData))    #There is no missing value in our data
[1] 0
> # support vector regression
 #library(e1017)

> model_svm=svm(revenue~.,data=train,type="eps-regression",kernel="radial")
> summary(model_svm)

Call:
svm(formula = revenue ~ ., data = train, type = "eps-regression", kernel = "radial")


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.06666667
    epsilon:  0.1
```

Number of Support Vectors: 447

```
> predict=predict(model_svm,newdata=test[,-5])

> length(predict)
[1] 3088

> length(test$revenue)
[1] 3088

> mean((predict-test$revenue)^2)
[1] 0.02950947

sqrt(mse)
[1] 0.1717832


> plot(predict,test$revenue)#MSE
> abline(0,1)


>
```



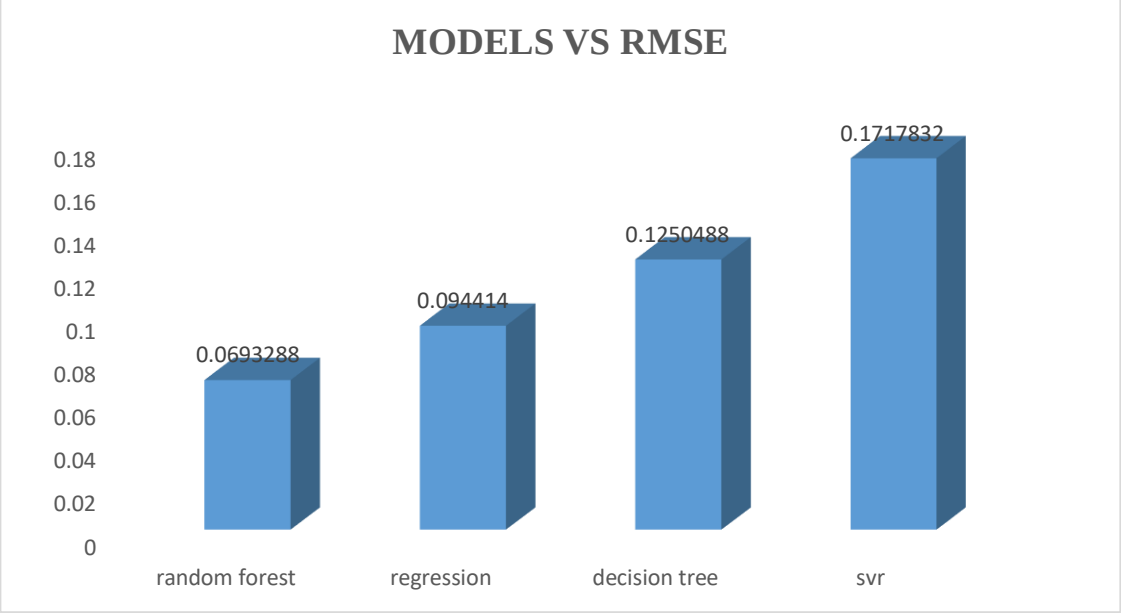
Interpretation:

Here MSE of given model is 0.0295, which is somewhat larger than other model hence our model is not much good fit for data.

Many points are along the straight line since support vector is good model.

COMPARISON WITH EXISTING MODEL

Techniques	RMSE
Random forest	0.0693288
Regression	0.094414
Decision Tree	0.1250488
SVR	0.1717832



Conclusion: Here RMSE of SVR model is high i.e 0.1717832 & RMSE of Random forest model is comparatively less i.e 0.0693288 since our random forest model is more accurate than the other models Decision tree, SVR and Multiple Regression.

Random forest model gives most accurate result than other models.

OVERALL CONCLUSION

From Data visualization:

- From barplot of Device type Vs Revenue, we conclude that from Desktop advertisement profit is more than advertisement publish on Mobiles.

- From barplot of Bidder code Vs Revenue, ix bidder code contribute more to gain more profit. Using ix code more advertisements were open so it contribute more.

- From all 7 rendered size, 300×600 size of advertisement people should be more preferable since this rendered size contribute more to gain profit.

- From graph of time key Vs Revenue, on day 17/09/2019 revenue is high i.e. 346.0. People have seen more advertisement on this day company gain more profit on this day. On 21/09/2019 people have seen less advertisement so company gain less revenue (profit).

From Model Fitting:

- By comparing all existing model and from that, we say that random forest model give more accurate result.

- From random forest model, we can say that impression is most important variable from all variables i.e. people have seen advertisement is give more profit than other variables, since impression is most effectible factor to increase revenue (profit).

LIMITATIONS

1. The study should be only for sample size 15437.

2. Times of India is only publisher for the advertisement.

3. Banner is only media type for publishing advertisement.

4. Our study is limited for 9 days only.

FUTURE SCOPE

In our analysis, we came to decision that impression and bids count are two main important factor. So to refer this result many companies and agencies will be trying to increase profit by focusing on those 2 factor i.e. impression and bids count. So there main aim is how they will work hard for to increase impression and bids count.

REFERENCES

- Introducing a linear regression analysis.
- **“**Data mining concept and techniques **”** second edition, Jiawei Han and Micheline kamber.
- Statistical computing R software.

SOURCES

- Google ,Wikipedia.
- Youtube.
- Analytics vidya