LOGISTIC REGRESSION(R-Project)

Approach steps and a brief summary about the project

Internet marketing has taken over traditional marketing strategies in the recent past. Companies prefer to advertise their products on websites and social media platforms. However, targeting the right audience is still a challenge in online marketing. Spending millions to display the advertisement to the audience that is not likely to buy your products can be costly.

Data contains website data by users with different countries located in different cities. The final database was quite large, consisting of 10 Columns and 6657 rows.

The main variable we are interested in is 'Clicked on Ad'. This variable can have two possible outcomes: 0 and 1 where 0 refers to the case where a user didn't click the advertisement, while 1 refers to the scenario where a user clicks the advertisement.

The objective was to create a model to predict who will click on the advertisement published on the website.

Following are the steps which were followed to create the model:

We will perform some exploratory data analysis to see how 'Daily Time Spent on Site' in combination with 'Ad Topic Line' affects the user's decision to click on the add.

Step 1.
uploaded the **Raw data**.

Step 2.
Explored the dataset and categories all the variables into three categories. Continuous, Categorical and Qualitative Columns.

summary(InputData)

str(InputData)

To check the dimension of the data set

dim(InputData)

Finding total no. of unique values in each variable at once

lengths(lapply(InputData,unique))

Continuous Cols=c('Time_Spent','Age','Avg_Income','Internet_Usage')

Categorical Cols=c('Male','Time_Period','Weekday','Month','Ad_Topic','Clicked')

Qualitative Cols=c('VisitID','Year','Country_Name','City_code')

We don't consider qualitative data for predictive modelling

It is not a statistically representative form of **data collection. The **qualitative** research process does not provide statistical representation.

Country_Name : Factor w/ 237

**Country Name having factor of 237 levels. We do not consider factor level more than 30 as it will create dummy variable

InputData=InputData[,c(ContinousCols,CategoricalCols)]

Final Data to proceed

head(InputData)

Step 3. Identify the problem.

We will work with the advertising data of a marketing agency to develop a machine learning algorithm that predicts if a particular user will **be clicked** on an advertisement published on website.

Step 4.
The main variable we are interested in is '**Clicked** ' and this is our target variable

Step 5.
The Target variable is a categorical variable hence we will do logistic regression to predict add clicked on website.

Step 6.
Exploratory Data Analysis:

The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

Explored the data set using **Histograms (Continuous Columns) and Bar plot (Categorical Columns).**

Explore each potential predictor for distribution and quality

Library to generate professional colors

```
library(RColorBrewer)
```

Histogram for multiple Column at once

For splitting windows

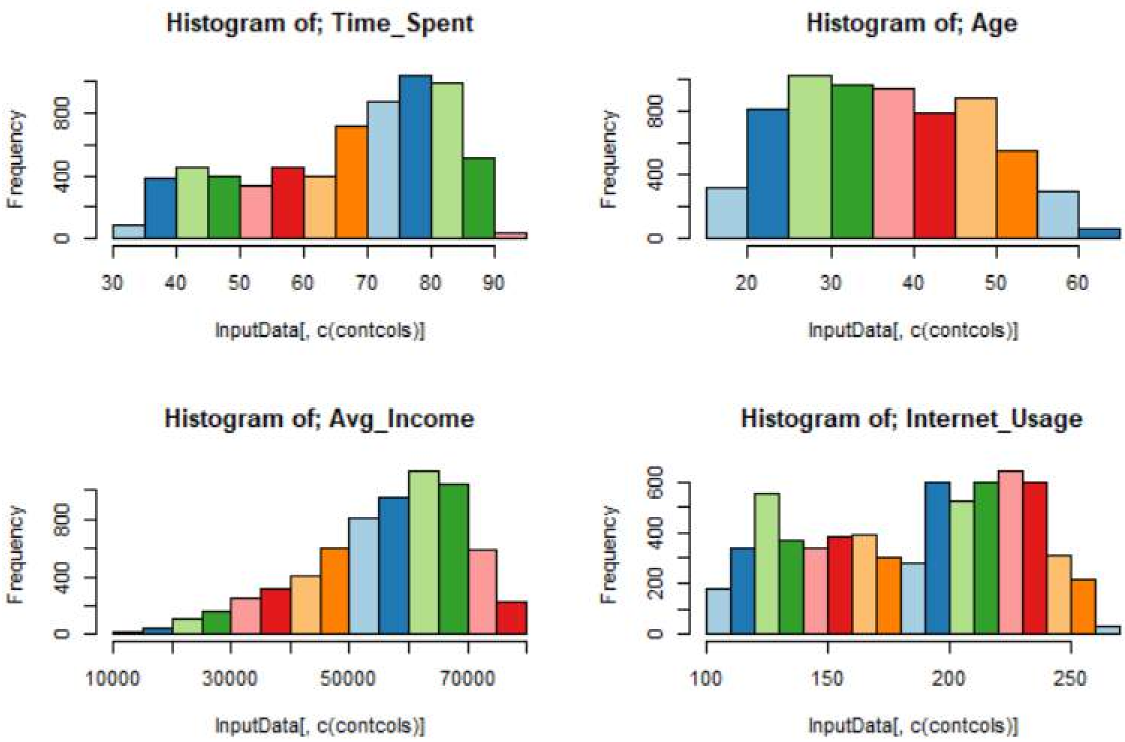
```
par(mfrow=c(3,3))

ColsForHist=c('Time_Spent','Age','Avg_Income','Internet_Usage')
```

Looping to create the histograms for each column

```
for(contcols in ColsForHist) {

hist(InputData[,c(contcols)], main=paste('Histogram of;', contcols),
col=brewer.pal(8,'Paired'))
}
```



Histogram

Changed character to numeric

InputData$Ad_Topic=as.factor(InputData$Ad_Topic)

str(InputData)

Bar plot for multiple categorical variables at once

For splitting windows

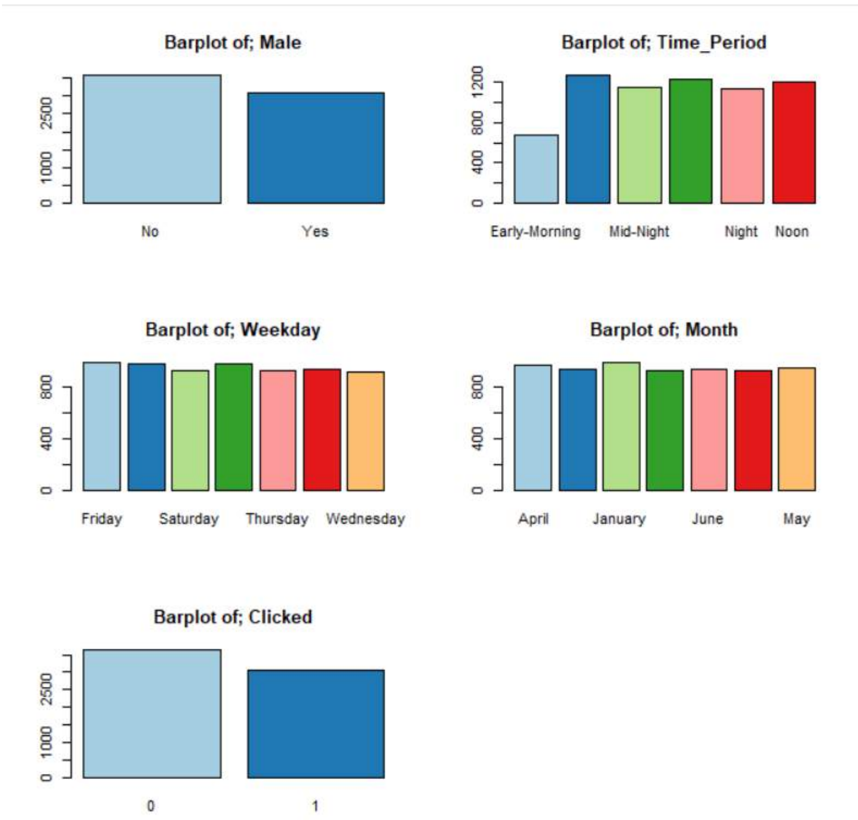
par(mfrow=c(3,3))

ColsForBar=c('Male','Time_Period','Weekday','Month','Clicked','Ad_Topic',)

Looping to create the bar plot for each column

```
for (catcols in ColsForBar) {

barplot(table(InputData[,c(catcols)]), main=paste('Barplot of;',  catcols),
 col=brewer.pal(8,'Paired'))
}
```



Bar Plot

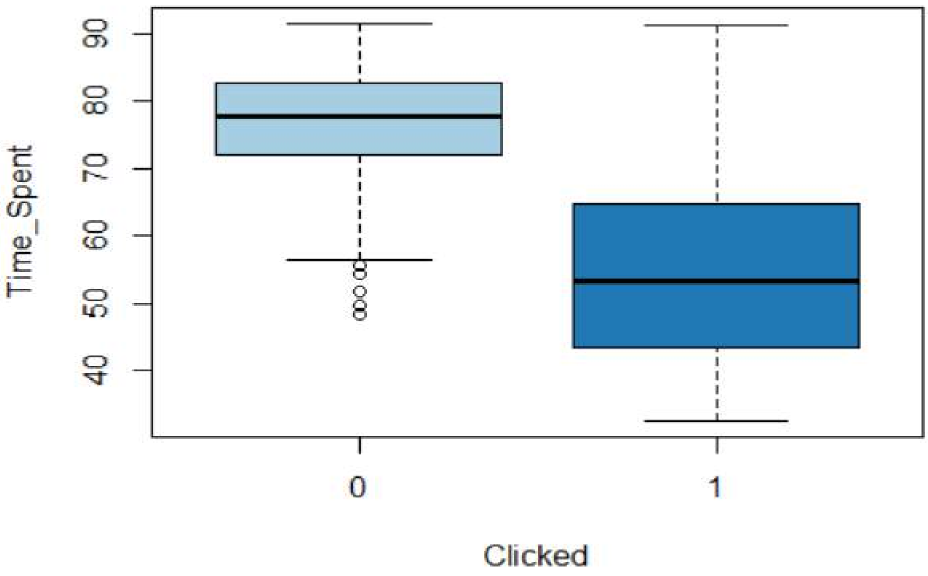
Step 7.

Exploring visual relationship b/w target variable and predictors.

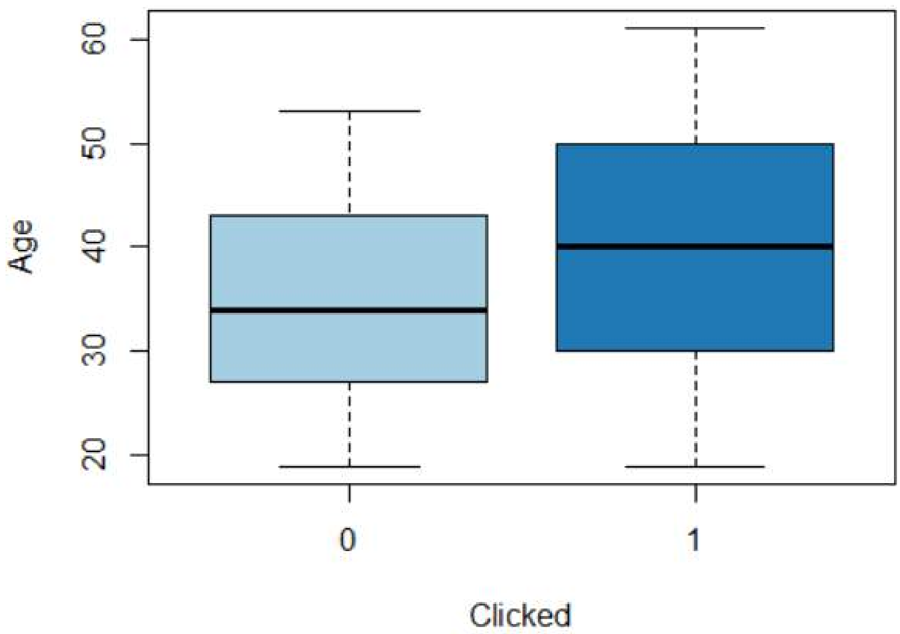
Categorical Vs Continuous --- Box Plot
Categorical Vs Categorical -- Bar chart

Box plot for single continuous variables.

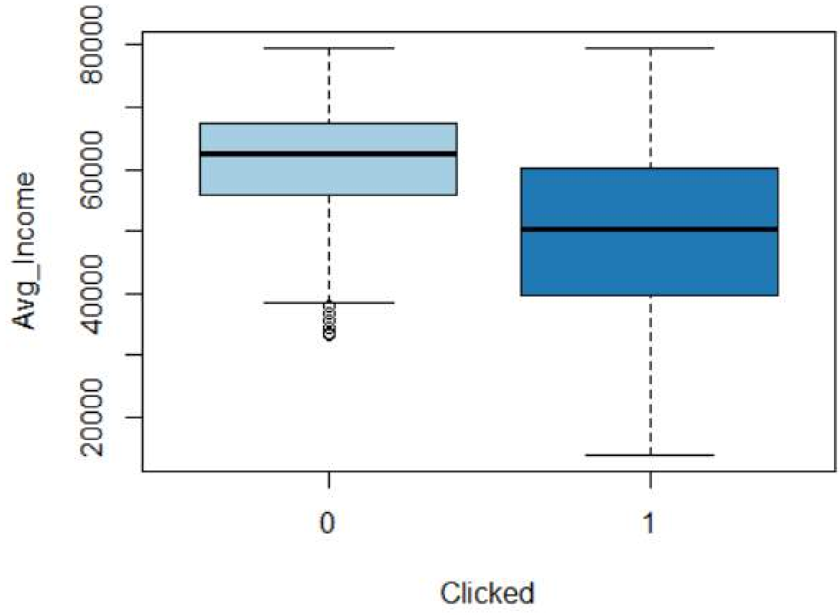
```
boxplot(Time_Spent~Clicked,data=InputData,col=brewer.pal(8,'Paired'))
```



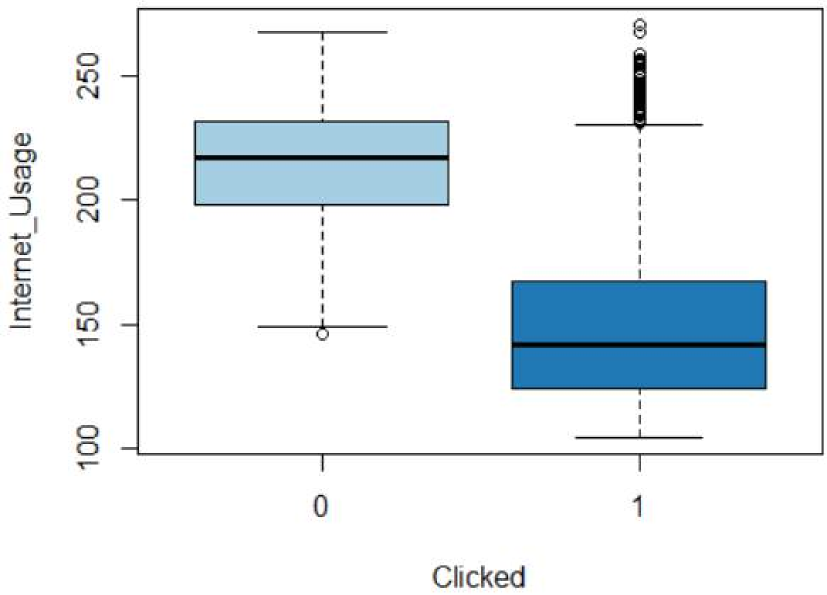
boxplot(Age~Clicked,data=InputData,col=brewer.pal(8,'Paired'))



```
boxplot(Avg_Income~Clicked,data=InputData,col=brewer.pal(8,'Paired'))
```



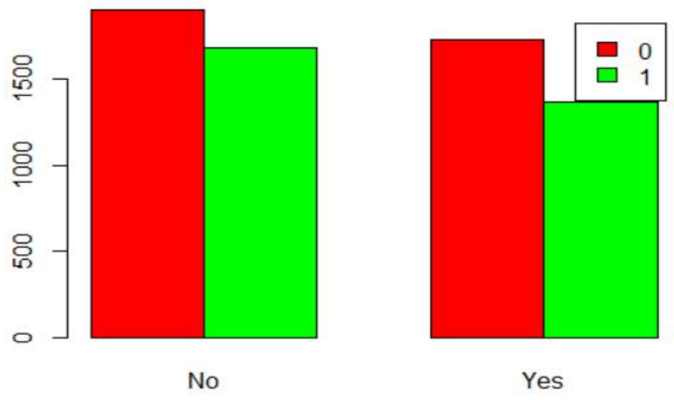

```
boxplot(Internet_Usage~Clicked,data=InputData,col=brewer.pal(8,'Paired'))
```



Categorical Vs Categorical -- Bar chart

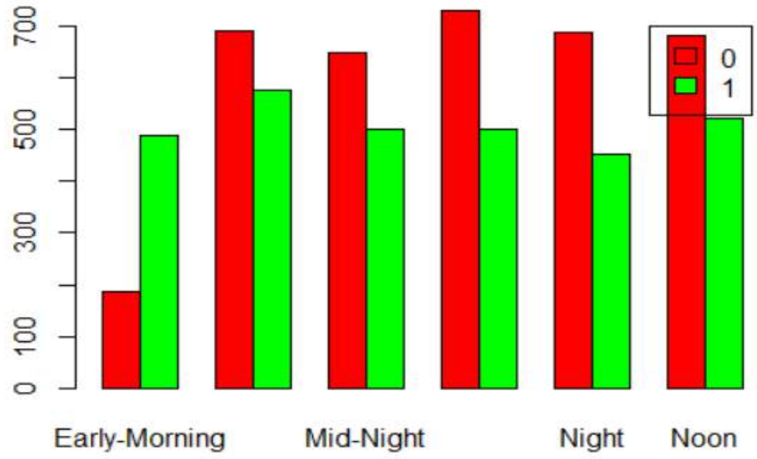
CrossTabResult=table(InputData[,c('Clicked','Male')])

barplot(CrossTabResult, legend=T,beside=T,col=c('Red','Green'))



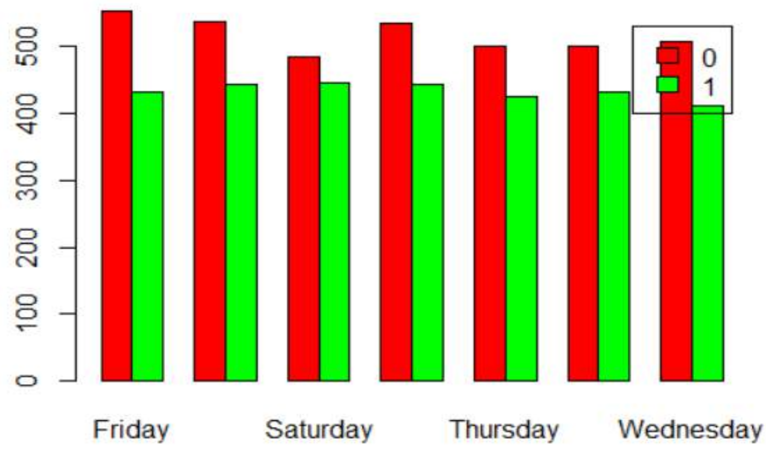
CrossTabResult=table(InputData[,c('Clicked','Time_Period')])

barplot(CrossTabResult, legend=T,beside=T,col=c('Red','Green'))



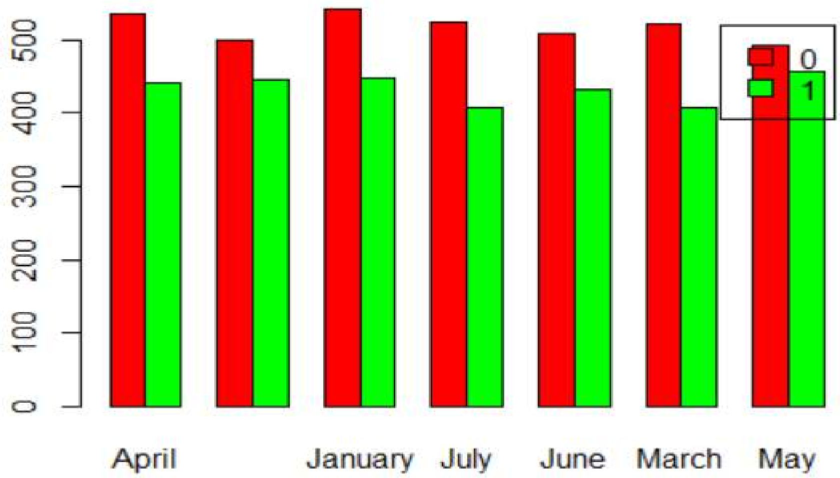
CrossTabResult=table(InputData[,c('Clicked','Weekday')])

barplot(CrossTabResult, legend=T,beside=T,col=c('Red','Green'))



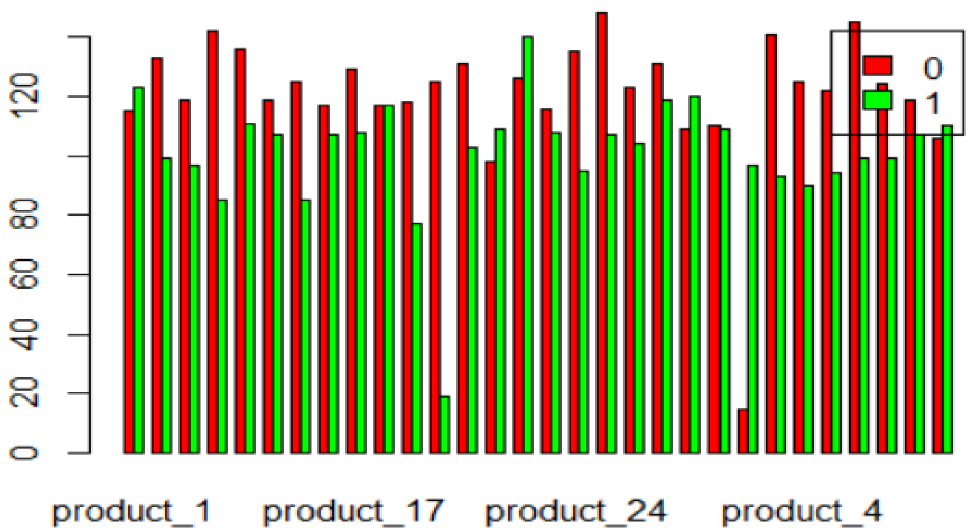
CrossTabResult=table(InputData[,c('Clicked','Month')])

barplot(CrossTabResult, legend=T,beside=T,col=c('Red','Green'))



CrossTabResult=table(InputData[,c('Clicked','Ad_Topic')])

barplot(CrossTabResult, legend=T,beside=T,col=c('Red','Green'))



Step 8.
Statistical Relationship b/w target variable and predictor

Categorical Vs Continuous --- ANOVA
Categorical Vs Categorical -- Chi-square test

Continuous Vs Categorical correlation strength: ANOVA

**Analysis of variance (ANOVA) is a statistical technique that is used to check if the means of two or more groups are significantly different from each other. ANOVA checks the impact of one or more factors by comparing the means of different samples.

F-Statistic is Mean Sq error/residual Mean Square error
H0: Variables are NOT correlated
Small P-Value--> Variables are correlated (H0 is rejected)
Large P-Value--> Variables are NOT correlated (H0 is accepted)

Looping to perform ANOVA test for each column

```
ContinousCols=c('Time_Spent','Age','Avg_Income','Internet_Usage',)

for (contcols in ContinousCols) {
 anovaData=InputData[,c('Clicked',contcols)]
 (print(str(anovaData)))
 print(summary(aov(Clicked~.,data=anovaData)))
}
```

```
'data.frame':   6657 obs. of  2 variables:
 $ Clicked   : int  0 1 0 1 0 1 1 0 0 1 ...
 $ Time_Spent: num  88 51.6 82.4 62.1 77.7 ...
NULL
              Df Sum Sq Mean Sq F value              Pr(>F)
Time_Spent     1  838.3   838.3    6860 <0.0000000000000002 ***
Residuals   6655  813.3     0.1
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
'data.frame':   6657 obs. of  2 variables:
 $ Clicked: int  0 1 0 1 0 1 1 0 0 1 ...
 $ Age    : int  43 50 38 45 31 38 26 23 22 50 ...
NULL
              Df Sum Sq Mean Sq F value              Pr(>F)
Age            1   98.4   98.43   421.7 <0.0000000000000002 ***
Residuals   6655 1553.1    0.23
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
'data.frame':   6657 obs. of  2 variables:
 $ Clicked   : int  0 1 0 1 0 1 1 0 0 1 ...
 $ Avg_Income: num  55901 39132 57032 48868 61608 ...
NULL
              Df Sum Sq Mean Sq F value              Pr(>F)
Avg_Income     1  349.9   349.9    1789 <0.0000000000000002 ***
Residuals   6655 1301.7     0.2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
'data.frame':   6657 obs. of  2 variables:
 $ Clicked       : int  0 1 0 1 0 1 1 0 0 1 ...
 $ Internet_Usage: num  185 177 211 190 205 ...
NULL
                 Df Sum Sq Mean Sq F value              Pr(>F)
Internet_Usage    1  911.2   911.2    8190 <0.0000000000000002 ***
Residuals      6655  740.4     0.1
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Categorical Vs Categorical relationship strength: Chi-Square test

The **Chi-**Square Test** of Independence determines whether there is an association between categorical variables (i.e., whether the variables are independent or related). It is a nonparametric test. This test is also known as: **Chi**-**Square Test** of Association.

H0: Variables are NOT correlated
Small P-Value--> Variables are correlated (H0 is rejected)
Large P-Value--> Variables are NOT correlated (H0 is accepted)

CrossTabResult1=table(InputData[,c('Clicked','Male')])

chisq.test(CrossTabResult1)

Pearson's Chi-squared test with Yates' continuity correction

data: CrossTabResult1
X-squared = 4.8939, df = 1, p-value = 0.02695

```
CrossTabResult2=table(InputData[,c('Clicked','Time_Period')])

chisq.test(CrossTabResult2)
```

Pearson's Chi-squared test

data: CrossTabResult2
X-squared = 227.1, df = 5, p-value < 0.00000000000000022

```
CrossTabResult3=table(InputData[,c('Clicked','Weekday')])

chisq.test(CrossTabResult3)
```

Pearson's Chi-squared test

data: CrossTabResult3
X-squared = 3.6596, df = 6, p-value = 0.7226

Ho is accepted (Weekday)

```
CrossTabResult4=table(InputData[,c('Clicked','Month')])

chisq.test(CrossTabResult4)
```

Pearson's Chi-squared test

data: CrossTabResult3
X-squared = 3.6596, df = 6, p-value = 0.7226

Ho is accepted (Month)

Step 9.
Treating missing values

```
colSums(is.na(InputData))
```

Step 10.

Generating the Data frame for machine learning.

```
TargetVariableName=c('Clicked')
```

Making sure the class of Target variable is FACTOR

```
TargetVariable=as.factor(InputData[, c(TargetVariableName)])
class(TargetVariable)
```

Choosing multiple Predictors which may have relation with Target Variable, based on the exploratory data analysis.

```
BestPredictorVaribles=c('Time_Spent','Age','Avg_Income','Internet_Usage','Male','Time_Period','Ad_Topic')

PredictorVariables=InputData[,c(BestPredictorVaribles)]

DataForML=data.frame(TargetVariable,PredictorVariables)

str(DataForML)
```

Data Splitting:

****Data splitting** is the act of partitioning available **data** into. two portions, usually for cross-validatory purposes. One. portion of the **data** is used to develop a predictive model. and the other to evaluate the model's performance.

We split out data into two portions: For training it's 70% and 30% for testing.

```
TrainingSamplingIndex= sample(1:nrow(DataForML),size=0.7*nrow(DataForML))

DataForMLTrain=DataForML[TrainingSamplingIndex,]

DataForMLTest=DataForML[-TrainingSamplingIndex,]

dim(DataForMLTrain)

dim(DataForMLTest)
```

Logistic Regression:

In statistics, the **logistic regression** is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc.

Creating Predictive models on training data to check the accuracy on test data.

```
startTime=Sys.time()
LR_Model=glm(TargetVariable ~ ., data=DataForMLTrain, family='binomial')
LR_Model
summary(LR_Model)
endTime=Sys.time()
endTime-startTime
```

```
Call:  glm(formula = TargetVariable ~ ., family = "binomial", data = DataForMLTrain)

Coefficients:
        (Intercept)           Time_Spent                  Age           Avg_Income       Internet_Usage
        18.53388895          -0.12299983           0.03806058          -0.00005995          -0.04317160
            MaleYes    Time_PeriodEvening  Time_PeriodMid-Night   Time_PeriodMorning     Time_PeriodNight
        -0.02725206          -0.58644441          -0.85322233          -0.97947417          -0.83009768
    Time_PeriodNoon   Ad_Topicproduct_10   Ad_Topicproduct_11   Ad_Topicproduct_12   Ad_Topicproduct_13
        -0.63749152           0.55925694           0.58689417          -0.22696173           0.65859316
 Ad_Topicproduct_14   Ad_Topicproduct_15   Ad_Topicproduct_16   Ad_Topicproduct_17   Ad_Topicproduct_18
         0.54009826           0.28315596           0.60501211           0.29769899           0.37703540
 Ad_Topicproduct_19    Ad_Topicproduct_2   Ad_Topicproduct_20   Ad_Topicproduct_21   Ad_Topicproduct_22
         0.35666260          -0.87756825           0.00533006           1.05387890           0.84703176
 Ad_Topicproduct_23   Ad_Topicproduct_24   Ad_Topicproduct_25   Ad_Topicproduct_26   Ad_Topicproduct_27
         0.43760810           0.23760643           0.37717857           0.66351699           0.49044193
 Ad_Topicproduct_28   Ad_Topicproduct_29    Ad_Topicproduct_3   Ad_Topicproduct_30    Ad_Topicproduct_4
         0.77099704           0.90683927           2.50694084           0.10813189           0.18752006
  Ad_Topicproduct_5    Ad_Topicproduct_6    Ad_Topicproduct_7    Ad_Topicproduct_8    Ad_Topicproduct_9
         0.20553607           0.54563820           0.39550939           0.74603486           0.67754357

Degrees of Freedom: 4658 Total (i.e. Null);  4619 Residual
Null Deviance:      6428
Residual Deviance: 2010        AIC: 2090
```

Null deviance: 6427.6 on 4658 degrees of freedom
Residual deviance: 2010.4 on 4619 degrees of freedom
AIC: 2090.4

Number of Fisher Scoring iterations: 7

Time difference of 0.1460268 secs

Durbin-Watson Test:

durbinWatsonTest(LR_Model)

lag Autocorrelation D-W Statistic p-value
1 -0.08000955 2.153429 0.05
Alternative hypothesis: rho != 0

Checking Accuracy of model on Testing data

Rejecting the Variables with High probability and accepting the ones which have probability close to zero.

PredictionProb=predict(LR_Model, DataForMLTest, type = "response")
DataForMLTest$Prediction=ifelse(PredictionProb>0.6, 1, 0)
DataForMLTest$Prediction=as.factor(DataForMLTest$Prediction)

head(DataForMLTest)

```
> head(DataForMLTest)
   TargetVariable Time_Spent Age Avg_Income Internet_Usage Male Time_Period   Ad_Topic Prediction
2               1      51.63  50   39132.00         176.73   No     Evening  product_8          1
8               0      82.58  23   61601.05         183.42   No        Noon  product_8          0
12              0      75.71  34   62109.80         246.06   No     Morning product_27          0
14              1      32.60  38   40159.20         190.05   No   Mid-Night product_13          1
15              0      54.35  23   63727.50         211.56   No     Evening  product_5          0
26              0      83.67  44   52140.04         250.35  Yes     Morning product_18          0
> |
```

Creating the Confusion Matrix to calculate overall accuracy, precision and recall on TESTING data.

Confusion Matrix:

Confusion matrices are used to visualize important predictive analytics like recall, specificity, accuracy, and precision. Confusion matrices are useful because they give direct comparisons of values like True Positives, False Positives, True Negatives and False Negatives

Accuracy=True Positive + True Negative /Total Population

library(caret)
AccuracyResults=confusionMatrix(DataForMLTest$Prediction, DataForMLTest$TargetVariable, mode = "prec_recall")

AccuracyResults

```
> AccuracyResults
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1074  106
         1   25  793

               Accuracy : 0.9344
                 95% CI : (0.9227, 0.9449)
    No Information Rate : 0.5501
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.8664

 Mcnemar's Test P-Value : 0.000000000002756

              Precision : 0.9102
                 Recall : 0.9773
                     F1 : 0.9425
             Prevalence : 0.5501
         Detection Rate : 0.5375
   Detection Prevalence : 0.5906
      Balanced Accuracy : 0.9297

       'Positive' Class : 0
```

Since Accuracy Results is a list of multiple items, fetching useful components only.

AccuracyResults[['table']]

AccuracyResults[['byClass']]

```
> AccuracyResults[['table']]
          Reference
Prediction    0    1
         0 1074  106
         1   25  793
> AccuracyResults[['byClass']]
         Sensitivity          Specificity       Pos Pred Value       Neg Pred Value            Precision
           0.9772520            0.8820912            0.9101695            0.9694377            0.9101695
              Recall                   F1           Prevalence       Detection Rate Detection Prevalence
           0.9772520            0.9425186            0.5500501            0.5375375            0.5905906
   Balanced Accuracy
           0.9296716
> |
```

Prediction 0 1

0 139 44

1 12 73

Precision - 0.91

Recall- 0.97

Accuracy (F1) - 94.25 %.

Outcomes:

- People who spent less time on the website are the one who clicked the ad.
- Most of them were male.
- People below the age of 50 years, most likely to click on the ad.
- People with less internet usage are more likely to click on the ad.
- Time frame where people mostly clicked on the ad was either early morning or in the evening and less were on noon.