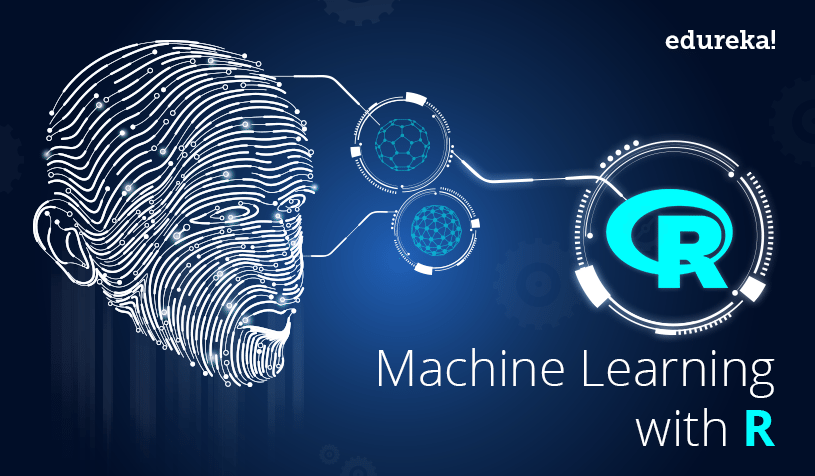
****

**WEBSITE AD CLICK BY USING LOGISTIC REGRESSION**

**Submitted by –**

**Rahul Kumar**

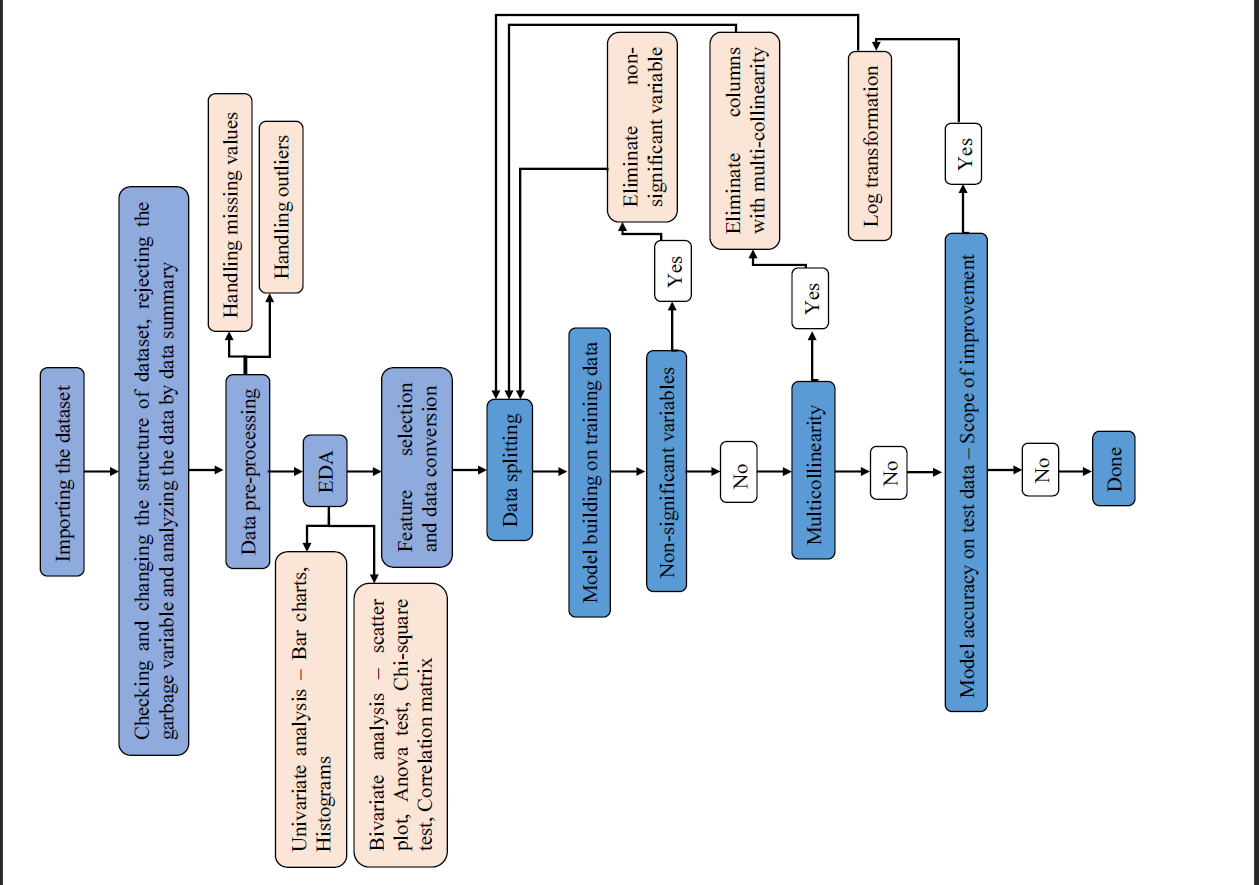
**Ivy Professional School**

**INTRODUCTION**

Marketing a product through internet is now becoming more popular as compared to traditional marketing strategies from recent years and now most of the companies like to advertise their products on websites and social media platforms. However, targeting the right audience is a very big problem with online marketing. Just spending millions of rupees to display the advertisement to the audience who is not likely to buy the product associated with the advertisement can be very costly. So the need of the hour is to derive a way so that relevant advertisements can be shown to relevant audiences. For this it is very important to know whether a particular audience is going to click on the advertisement or not and for that this project focuses on the deployment of logistic regression model to predict website AD-click.

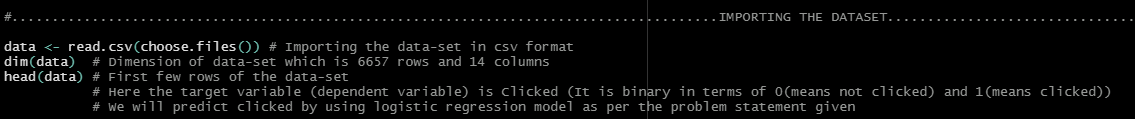
For the sake of model deployment, we will use R studio.

The work flow for the project is –

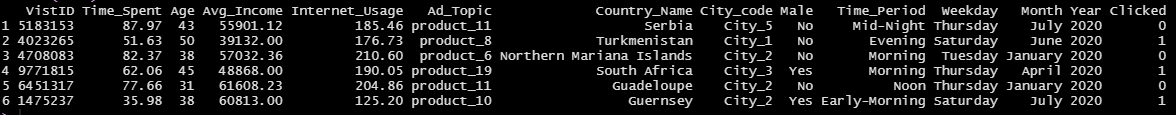


**DATASET AND IMPORTING DATA**

The data set (named as Web\_data) for this project is in csv format and consists of 14 attributes: "Ad\_Topic", "Country\_Name", "City\_code", "Male", "Time\_Period", "Weekday", "Month", "Year", "VistID", "Time\_Spent", "Age", "Avg\_Income", "Internet\_Usage" and “Clicked”. The main attribute of interest is "Clicked" and this is the attribute which we will predict with the help of logistic regression model. This attribute can have two possible outcomes: 0 and 1 where 0 refers to the situation where an audience didn't click on the advertisement, while 1 refers to the case where he clicks on the advertisement. Hence our dependent variable is “Clicked” and all other 13 are independent variable.

Since the dataset is in csv format so it was imported using read.csv as – 

As we can see the dimension of data is (6657,14) which means there are total 6657 data points with 13 independent variable and a dependent variable. The data looks something like below –



**IDENTIFYING AND CHANGING VARIABLE TYPE, ELIMINATION OF GARBAGE VARIABLES AND DATA SUMMARY**

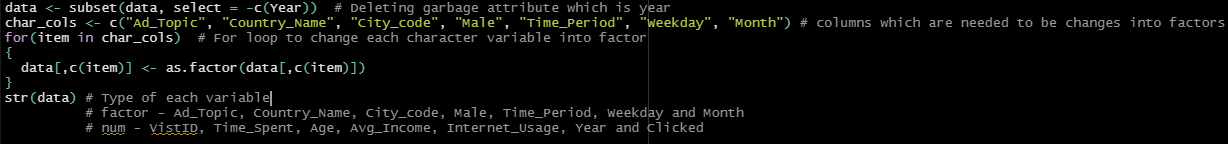
Before starting to deploy any model in machine learning it is very important to understand that what kind of variables we are having, whether it is necessary to convert their type or not and if there is any garbage variable then that needs to be eliminated or not. Garbage variable can be any variable which is not changing with respect to dependent variable or it can be any variable which has no relation with the dependent variable like any ID column or unique column.

Code for the understanding the data type is –



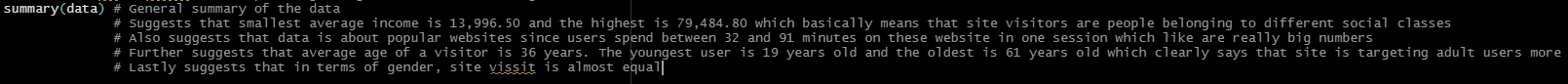
As we can see that we have almost half of them as character variable and half of them as numerical variable. Since R can understand numeric variables and cannot understand the character variables so, we need to change the character variables into factors and later (just before deploying the model) we need to change them into numbers. Also one of the numeric variable “Year” is not changing throughout the whole dataset. Its value is 2020 so it will not contribute anything to the model and hence should be eliminated as garbage variable.

The code for changing character variables into factors (by using for loop) and elimination of garbage variable is –



Now we can clearly see that all the variables are either numeric or factor.

Now to know about the data little bit more, we will use summary function in R which will give us the general statistical parameters for each column. The code for doing it is very simple –



The summary of the data suggests that –

* Smallest average income is 13,996 and the highest is 79,485 which basically means that site visitors are people belonging to different social classes
* Data is about popular websites since users spend between 32 and 91 minutes on these website in one session which like are really big numbers
* Average age of a visitor is 36 years. The youngest user is 19 years old and the oldest is 61 years old which clearly says that site is targeting adult users more
* In terms of gender, site visit is almost equal

**DATA PRE-PROCESSING**

Any data in real world generally contains noises and missing values which makes it unsuitable to be directly used in any machine learning model deployment. In simple words it is not always the case that we will come across the clean and formatted data. Hence data pre-processing is very important tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

In data pre-processing we have generally 2 steps–

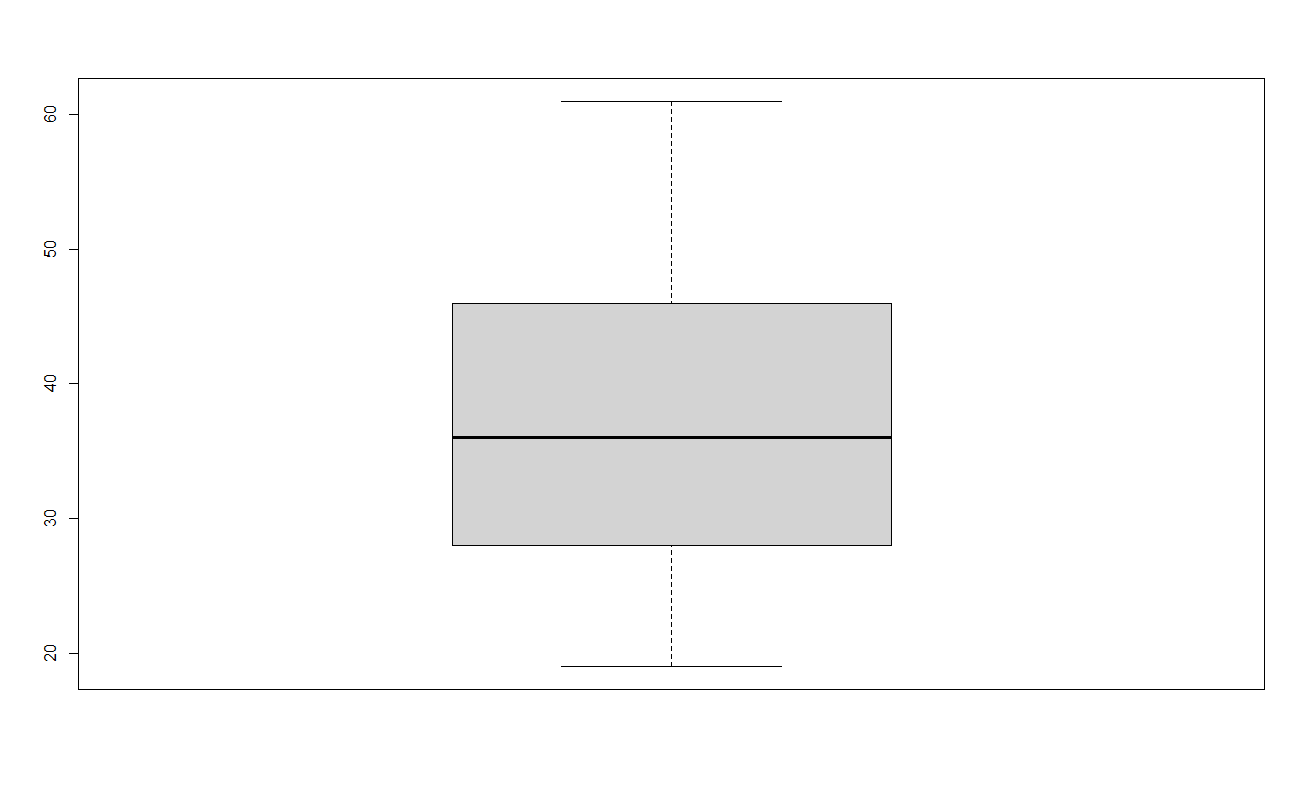
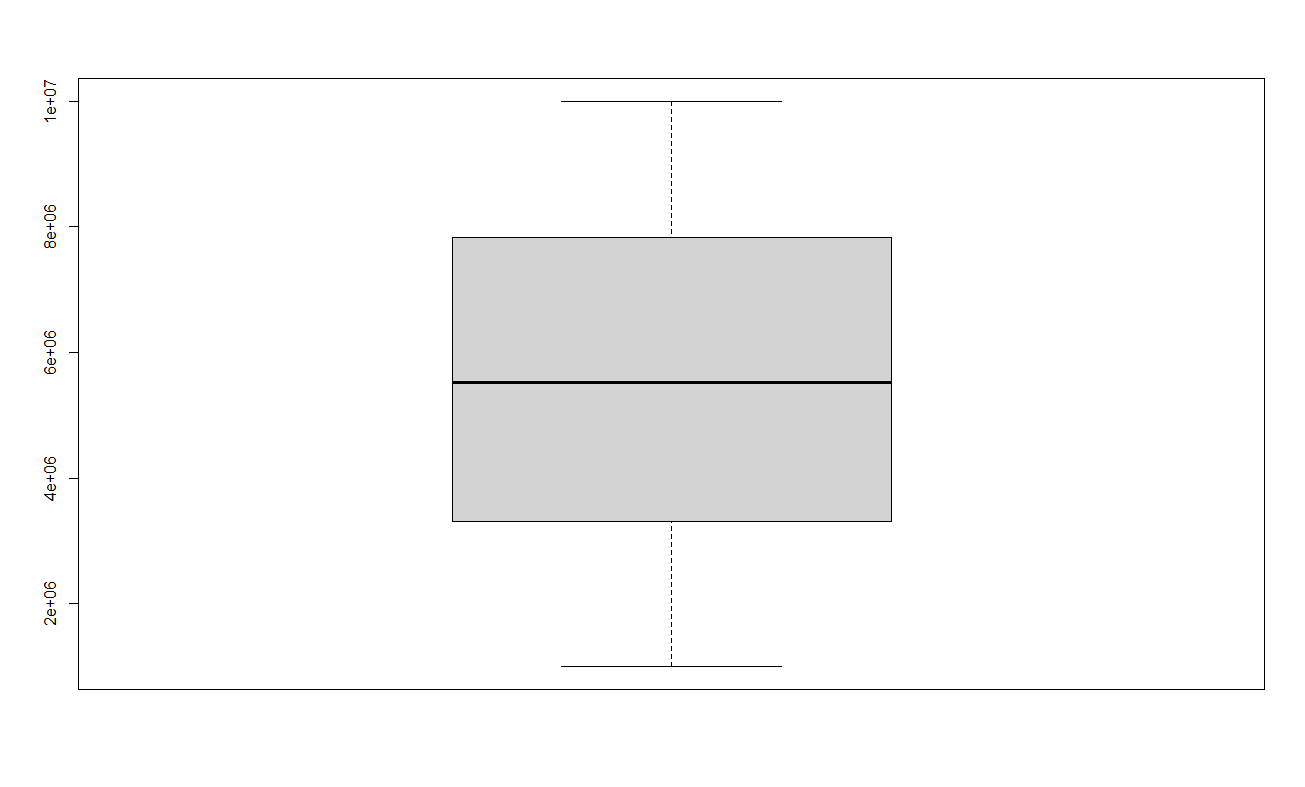
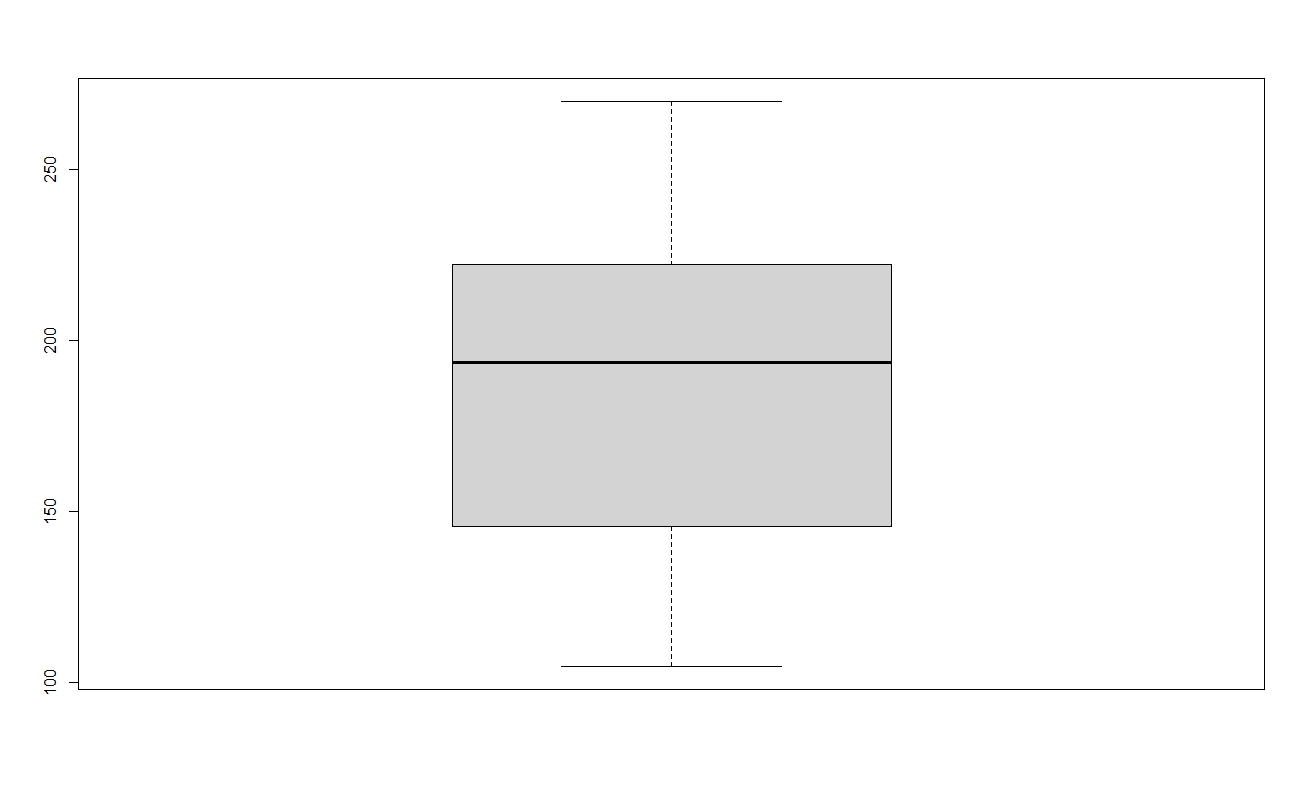
* Handling missing values
* Handling outliers

Any column or attribute having more than 20% missing value is generally eliminated and for the columns having less than 20% missing values – missing values are replaced by either mean (In case data is numerical and there is no outlier), median (In case data is numerical and there is outlier) or mode (In case data is not numerical). The dataset that we are working with has no missing value. The code and result for the same is -



Now coming to outliers, machine learning algorithms like linear regression and logistic regression are very sensitive to the range and distribution of attribute values. Data outliers can spoil and mislead the process of training these models resulting in longer training times, less accurate models and ultimately poorer results. So, it is very important to detect outliers in each continuous attribute and treat those outliers.

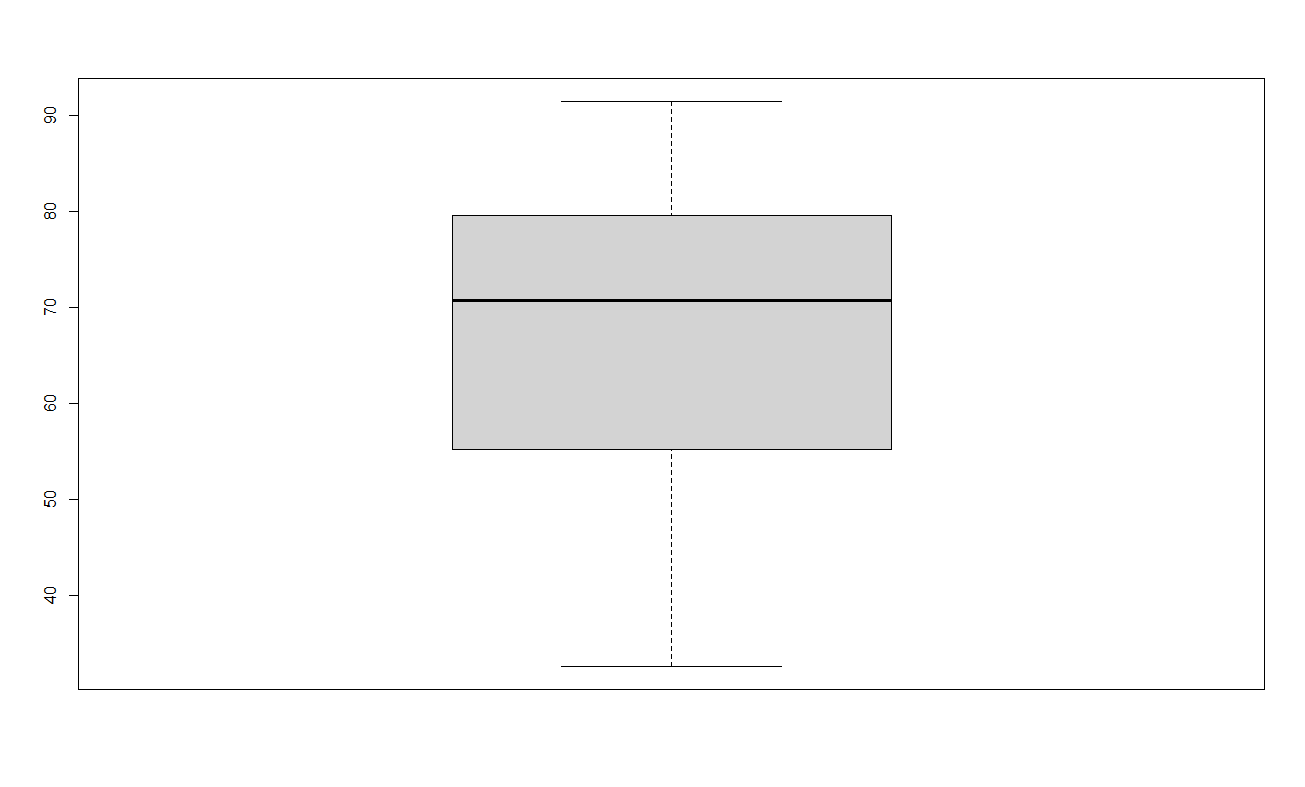
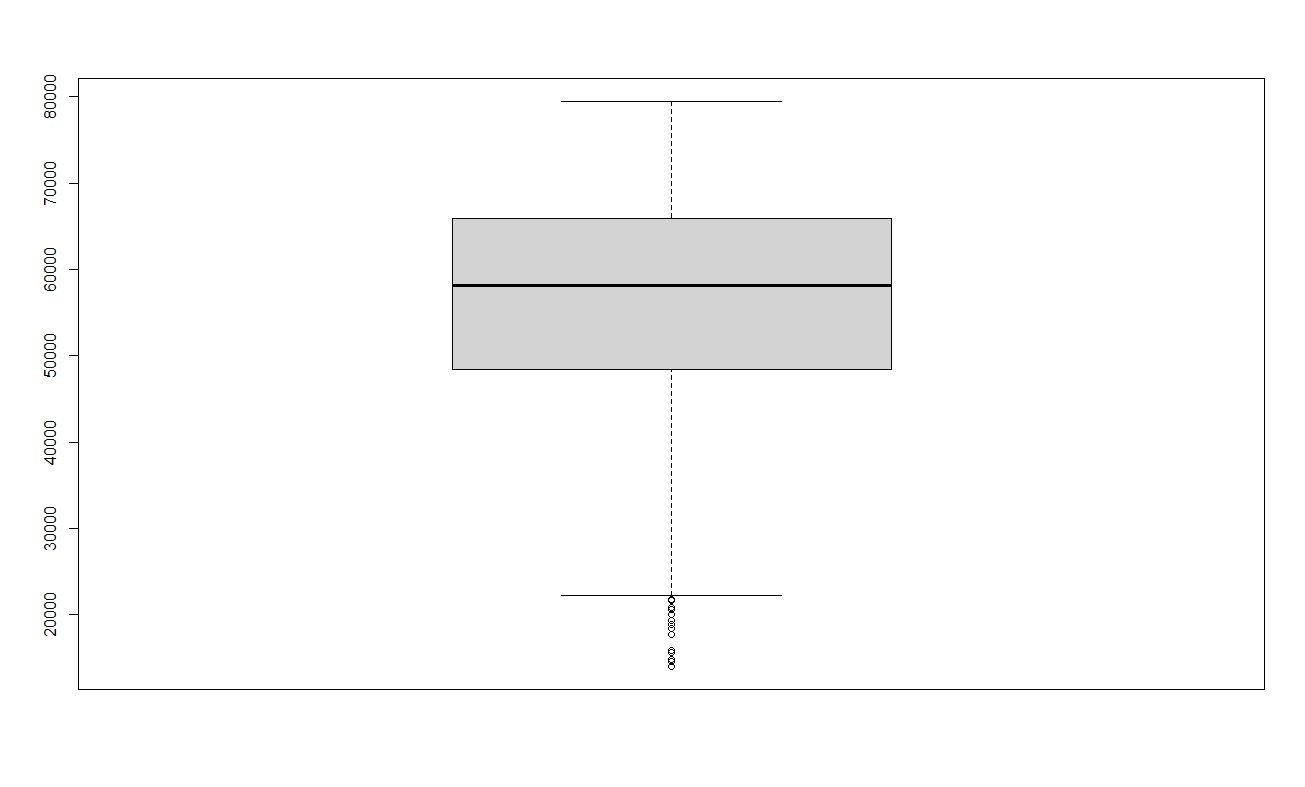
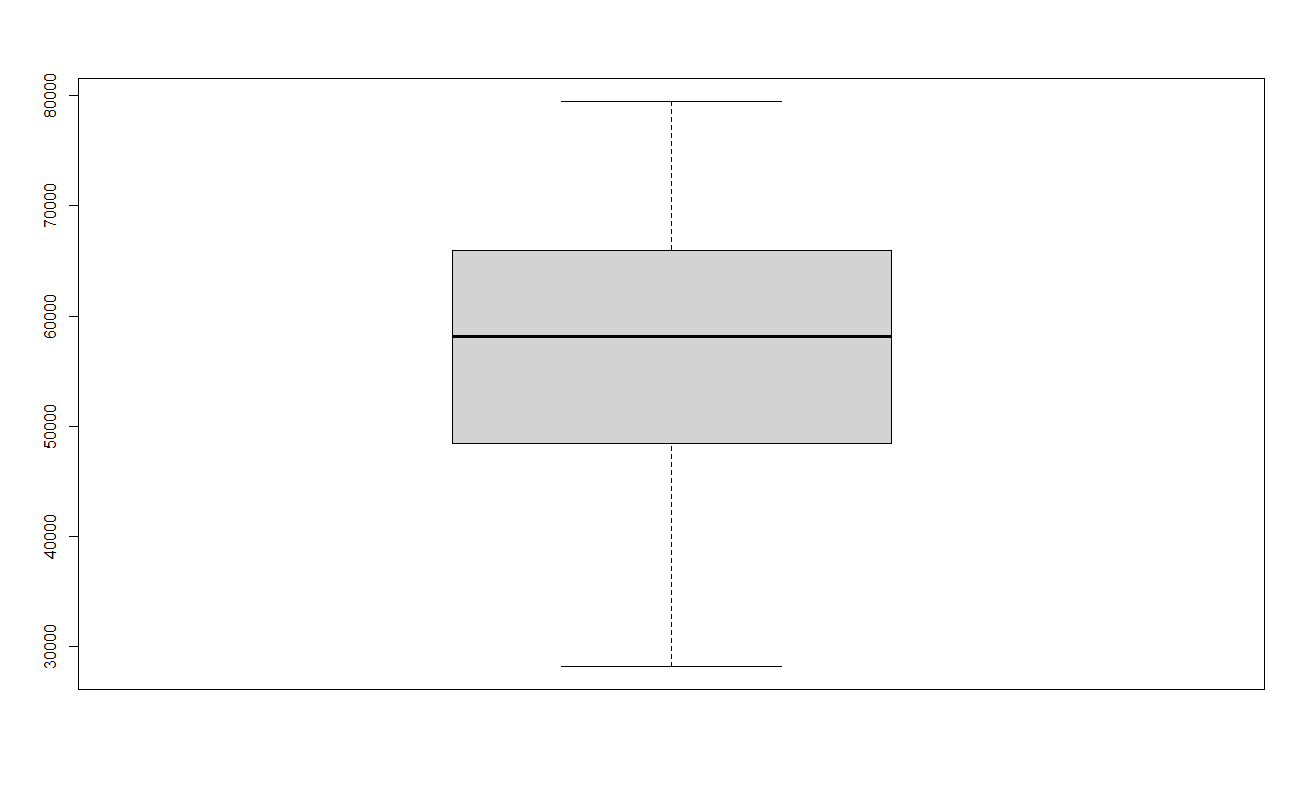
Outliers can be detected by using boxplots (graphically depicting groups of numerical data through their quartiles) and can be treated by changing the outlier points to the maximum point of boxplot (if positive outlier) or to the minimum point of boxplot (if negative outlier).

Boxplot for all the continuous attributes are –

INTERNET\_USAGE

VISIT\_ID

AGE



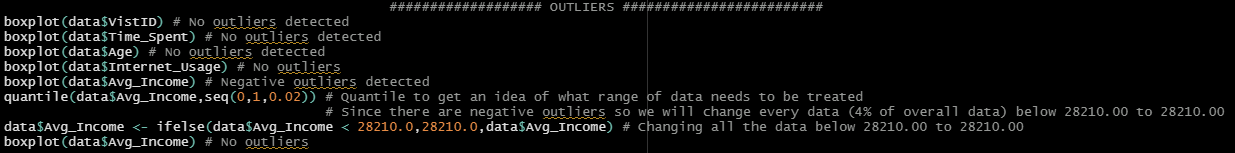
AVG\_INCOMEAFTER TREATMENT

AVG\_INCOME

BEFORE TREATMENT

TIME\_SPENT

We can see here that no attribute has any outlier except “Avg\_income” (negative outliers) and after treatment “Avg\_income” has no outliers also. Code for doing this in R is -



**EDA - UNIVARIATE AND BIVARAITE ANALYSIS**

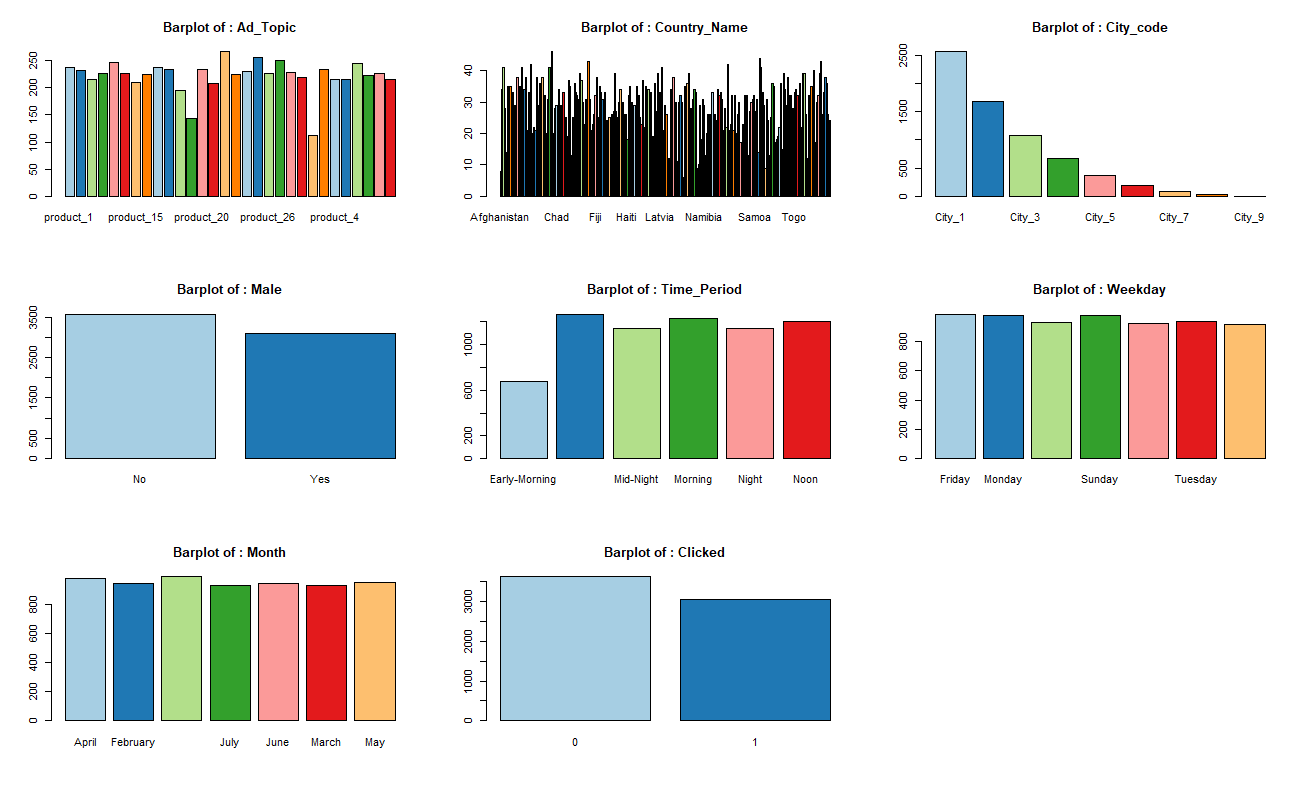
Before training a machine learning model it is also very necessary to do preliminary analysis of data to discover relationships between measures in the data and to gain an insight on the trends, patterns, and relationships among various attributes present in the data. It can be done by using univariate and bivariate analysis.

Both these analysis are the part of exploratory data analysis ( EDA - an approach of analysing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods ) which can be both graphical as well non graphical.

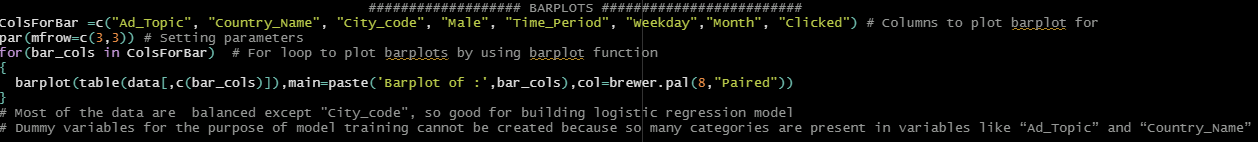
**UNIVARIATE ANALYSIS**

Uni means one and variate means variable, so in univariate analysis, there is only one variable to be separately analysed. The objective of univariate analysis is to derive the data, define and summarize it, and analyse the pattern present in it. In a dataset, it explores each variable separately. It is possible for two kinds of variables- Categorical and Numerical.

For categorical variables, bar plot is there. For the data that we are working with this plot is –

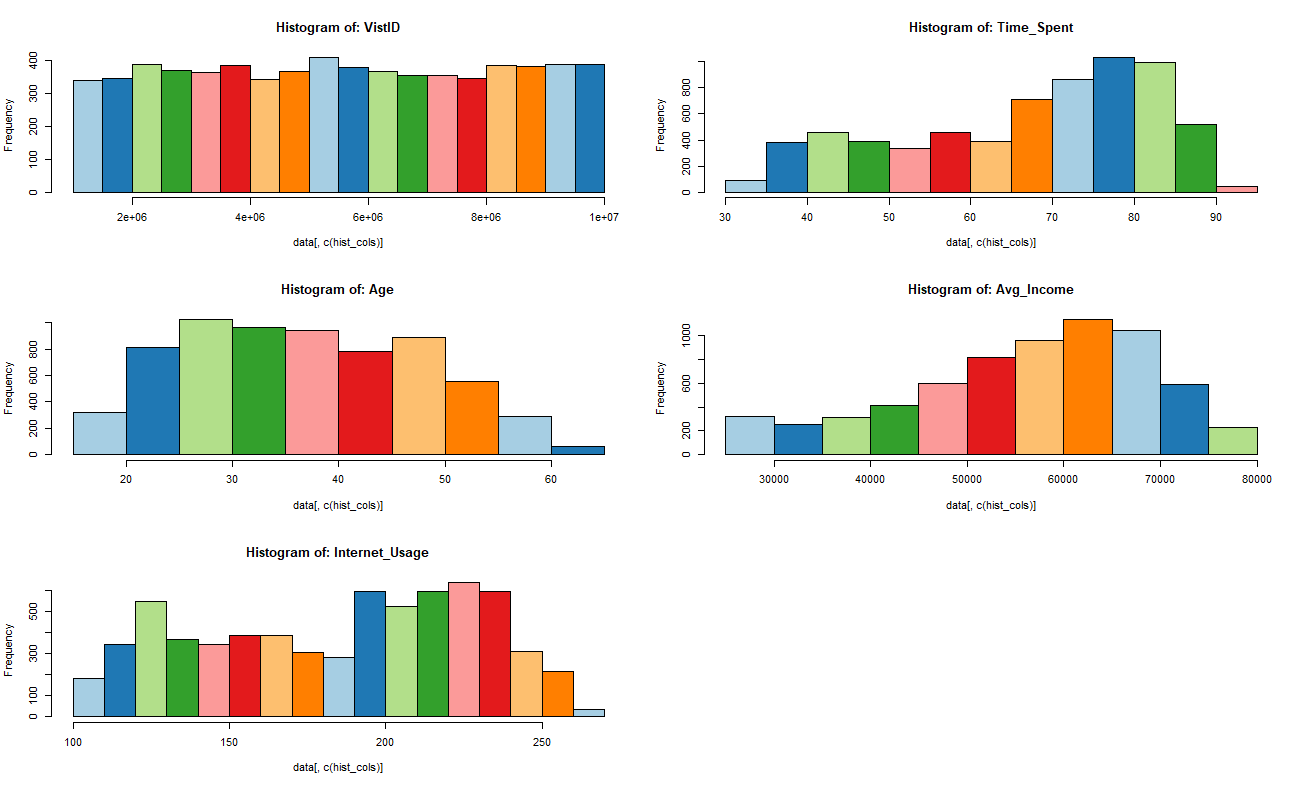


The code for plotting bar plots for the dataset in R is -

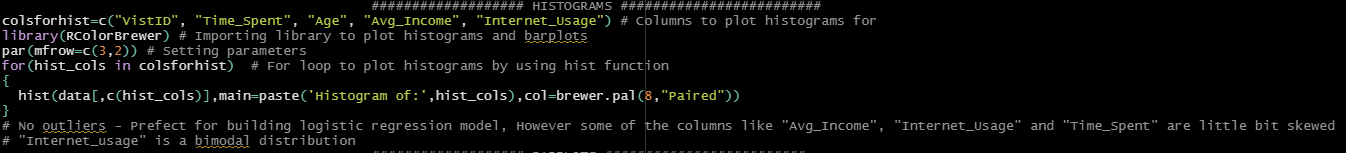


In this code, I have first created a vector containing all categorical variables. Now by using for loop, bar plot for each variable is created using “barplot” function. According to the bar plot, most of the categorical variables are balanced except “City\_code”. In “City\_code” variable, the frequency of data belonging to city\_1 is largest and the frequency of data belonging to city\_9 is smallest. This may affect the model on a small scale because of the biasness of data towards city\_1 but there are chances that it will get rejected during test of significance variable. Also, some variables like “Ad\_Topic” and “Country\_Name” are having so many categories. So, it is not possible to create dummy variables for the purpose of model training.

Similarly, for continuous variable histograms are plotted as –



The code for the same is –



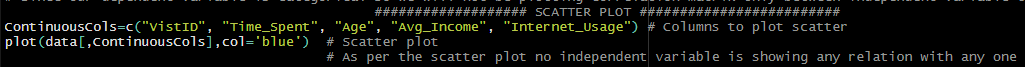
Similar as done in case of bar plots, a vector is created for continuous variable and then by using for loop, histogram for each variable is plotted using “hist” function. According to the histogram plotted, variables like "Avg\_Income", "Internet\_Usage" and "Time\_Spent" are little bit skewed. There are no outliers present but “Internet\_usage” is a bimodal distribution. This can affect the model slightly because normally distributed data are better towards providing better accuracy. This problem can be solved by applying transformation like that of log and then training the model on that transformed data. We will do it after checking the accuracy for model on untransformed data.

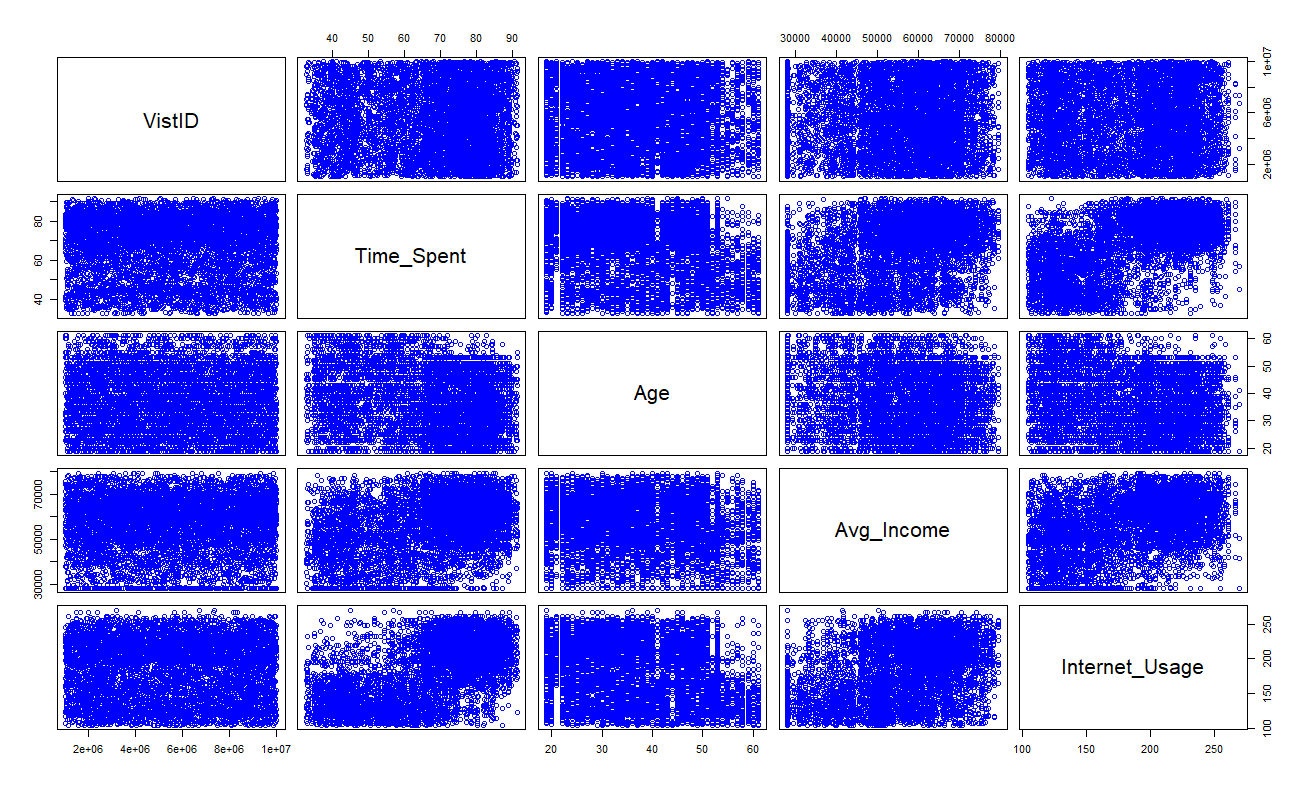
**BIVARIATE ANALYSIS**

Bi means two and variate means variable, so here there are two variables. The analysis is related to cause and the relationship between the two variables. There are many bivariate analysis. These are –

* Scatter plot (Plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data)

The code for creating the scatter plot between variable in the dataset is -



The plot looks something like -

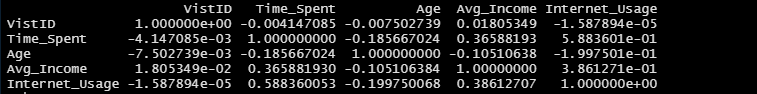
Here we are not taking “Clicked” because it is categorical (with only 0 and 1) which will not make any sense if plotted in scatter plot for finding any relation. According to this plot no two independent variables are correlated closely except “Internet\_usage” and “Time\_Spent” but to asses it more effectively, we will use correlation matrix.

* CORRELATION test (Test to find correlation between variables with value between 0 and 1 for each set of variable as output. It suggests strong correlation between two sets of variable if the value for them is greater than 0.5)

The code to create a correlation matrix is –



The result of this is –

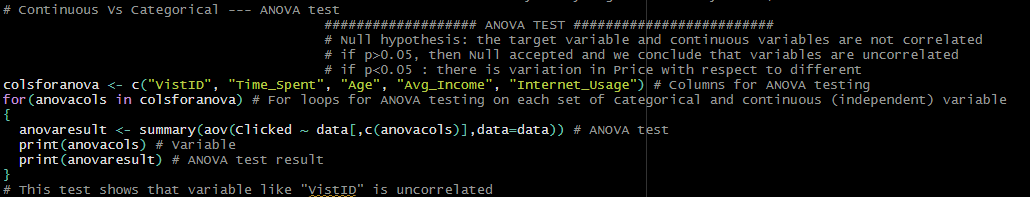


By seeing at the correlation matrix, it can be found that "Time\_Spent" and "Avg\_Income" are correlated (as in case of scatter plot) with each other with a value of ~0.6.It means change in one variable would cause change to another and so the model results can fluctuate significantly. This fluctuation can cause un-stability further causing overfitting. So by analysing the accuracy first without treating it for now, we will again check it by using VIF test (test for multicollinearity) and then will eliminate anyone of them if necessary.

* ANOVA test (Test to find where there are any non-correlated independent variables with dependent variable. It is done between continuous and categorical variable)

Here we basically do hypothesis testing with an assumption that our null hypothesis is that the target variable and continuous variables are not correlated. Now if p>0.05, then we will accept null hypothesis and conclude that variables are uncorrelated. Similarly, if p<0.05, we reject null hypothesis and conclude that variables are correlated

The code for carrying out ANOVA testing is –

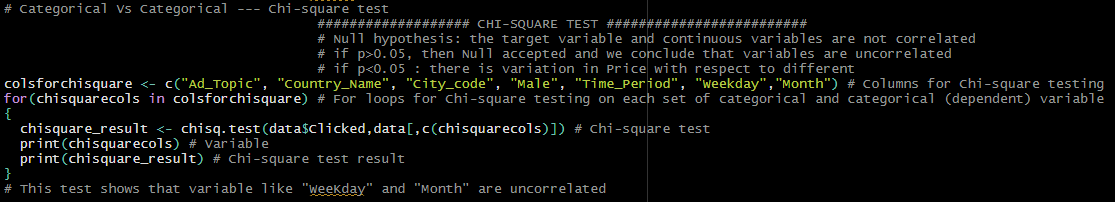


Here a vector of continuous variables is created and then with the help of for loop, P value is calculated by using “aov” function in R. After running the code the test suggested that variable “VistId” is of no use because it is not correlated with the output variable/ dependent variable. So, we will eliminate it during feature selection.

* Chi square test (Test to find where there are any non-correlated independent variables with dependent variable. It is done between two categorical variable)

Here also, we basically do hypothesis testing with an assumption that our null hypothesis is that the target variable and categorical variables are not correlated. Now if p>0.05, then we will accept null hypothesis and conclude that variables are uncorrelated. Similarly, if p<0.05, we reject null hypothesis and conclude that variables are correlated

The code for carrying out chi-square test is –



As done in case of ANOVA testing, a vector of categorical variables is created and then with the help of for loop, P value is calculated by using “chisq.test” function in R. The result from the test suggested that categorical variables like "WeeKday" and "Month" are uncorrelated with the dependent variable and hence we will eliminate these also during feature selection.

**FEATURE SELECTION AND DATA CONVERSION**

Before training a machine learning model, feature selection plays a very major role in improving the model. Some of its benefits include-

* Reduced overfitting problem
* Less redundant data which means less opportunity for the model to make decisions based on noise
* Improved accuracy because of misleading data which further means improved modelling accuracy
* Reduced training time because of less data which basically means that algorithm will train faster

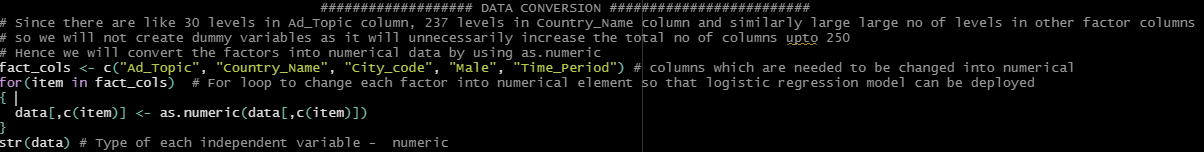
So we will apply feature selection technique and eliminate the unnecessary columns or features found during ANOVA and Chi-square test. The code for the same is –



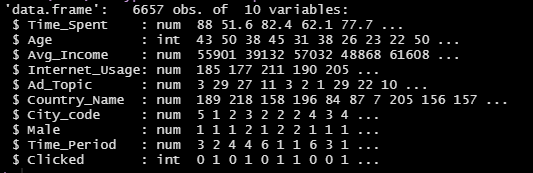
As found in bivariate analysis we have rejected "VistID", "WeeKday" and "Month".

Now after feature selection, feeding the data in proper format into the model is also very important. The performance of a machine learning model not only depends on the model and the feature selected but also on how we process and feed different types of variables to the model. Since most machine learning models only accept numerical variables, pre-processing the categorical variables becomes a necessary step. We cannot just put the character variable or categorical variable directly into the model without converting them into numerical form. So, to do that we will change the factor variable into numerical variable by using “as.numeric” function in R.

The code for conversion is –



Hence the final data set looks like –

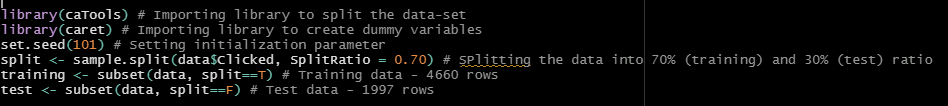


Now we have 9 independent variable with 6657 data points and all of them are in numerical format.

**DATA SPLITTING**

Since training and testing the model on same dataset will not help us to properly check the performance of our model (because model will be biased towards the training data and hence will always give better result on that) so to assess the performance, we will split the dataset into training and test set. It is a fast and easy procedure to perform, the results of which allow us to compare the performance of machine learning algorithms for our predictive modelling problem.

The code to do this is –



After running the above code we get–

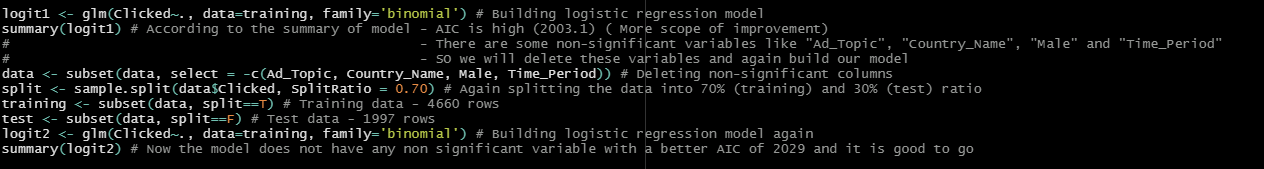
|  |  |  |
| --- | --- | --- |
|  | Training data | Test data |
| Data points | 4660 | 1997 |
| Attributes | 10 | 10 |

**MODEL BUILDING**

As discussed before, the model that we are going to deploy here is logistic regression model. It is a supervised machine learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes (1 and 0).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X.

The code to do this in R with the help of “glm” function is -

****

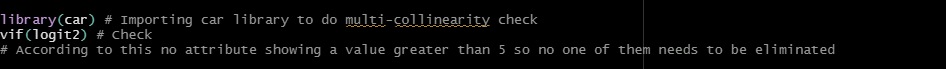
After running the code and training the model, it is giving an AIC score of 2003.1 (The akaike information criterion (AIC) score is an [estimator](https://en.wikipedia.org/wiki/Estimator) of prediction error for a given set of data. It estimates the quality of a model and thus provide us with a model selection criteria. A higher AIC score means less performing model and vice versa). Model is also giving some non-significant (not contributing in the prediction of dependent variable) variables like "Ad\_Topic", "Country\_Name", "Male" and "Time\_Period" which can be eliminated to improve the AIC score further and thus to improve the model ultimately.

Hence the non-significant variables are rejected and after again splitting and training the model, it is giving an AIC score of 2029 which is like similar to before but the model was not having any non-significant variables.

**MULTI-COLLINEARITY CHECK**

Multi-collinearity occurs when two or more independent variables are highly correlated with one another in a regression model. This means that an independent variable can be predicted from another independent variable in a [regression model](https://courses.analyticsvidhya.com/courses/Fundamentals-of-Regression-Analysis?utm_source=blog&utm_medium=what-is-multicollinearity). It can be a serious problem in a regression model because then we would not be able to distinguish between the individual effects of the independent variables on the dependent variable.

It can be detected via various methods out of which one is VIF test. VIF test gives us VIF score of an independent variable which represents how well the variable is explained by other dependent variables. Basically when VIF score is greater than 5 for any independent variable, we reject it as in this case it is showing high correlation with other independent variables. The code to carry out VIF test in R is-



And the result is-

which suggests that no multi-collinearity exists between any independent variable.

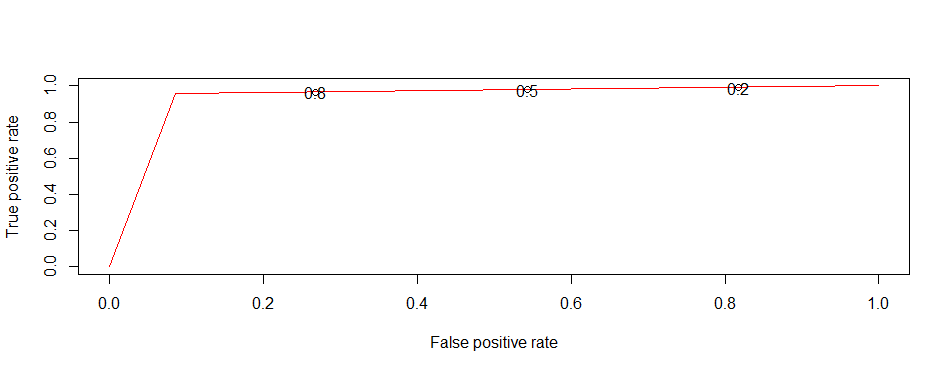
**MODEL ACCURACY**

Checking machine learning model accuracy is the most important step after we have trained a model. It is used to identify whether the model is able to understand the relationships and patterns between variables in a dataset based on the input, or training data. Often, however, techniques of measuring accuracy are used that give grossly misleading results. This can lead to the phenomenon of over-fitting where a model may fit the training data very well, but will do a poor job of predicting results for new data or test data not used in model training. So to avoid this we will use various accuracy measures such as –

* Accuracy - It’s the % ratio of the correctly labelled points to the whole pool of points
* Precision – It’s the ratio of the **correctly**positive labelled points to all positive labelled points
* Recall/Sensitivity - It is the ratio of the correctly positive labelled points to all actually positive points
* F1-score – It considers both precision and recall and **is basically the harmonic mean(average) of the precision and recall**
* Specificity – It’s the **correctly** negative labelled points to all the actual positive points
* Balanced accuracy -  It is the average of the proportion corrects of each class individually

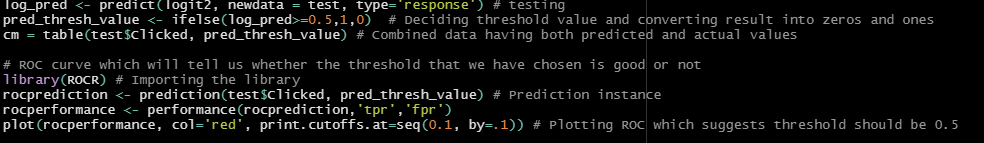
But before that, the model is used to predict on test data and the threshold for prediction is decided (as prediction are in terms of probability values) with the help of ROC curve. A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.  It is basically a probability curve that plots the TPR (True positive rates) against FPR (False positive rates) at various threshold values and essentially help us to choose better threshold value.

The ROC curve obtained is –

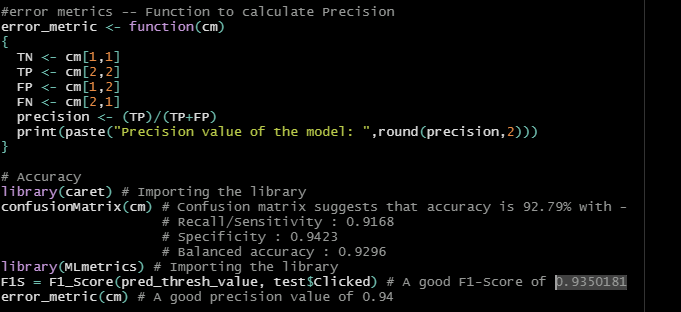


In simple words, we want that threshold which gives higher true positive rates and lower false positive rates. According to ROC the threshold values fulfilling this criteria lies between 0.5 - 0.9. So we will take the threshold of 0.5 and move ahead with that.

The code to plot this ROC curve is -

****

After this various accuracy parameters are calculated as explained before with help of below code –

****

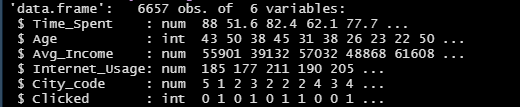
|  |  |
| --- | --- |
| Parameters | Value |
| Accuracy | 92.79% |
| Precision | 0.94 |
| Recall/Sensitivity | 0.9168 |
| F1-score | 0.9350181 |
| Specificity | 0.9423 |
| Balanced accuracy | 0.9296 |

It finally gives -

By seeing at these accuracy scores one can say that model is performing really good, However model can be improved further by applying log transformation to deal with data skewness and in other way to apply feature scaling.

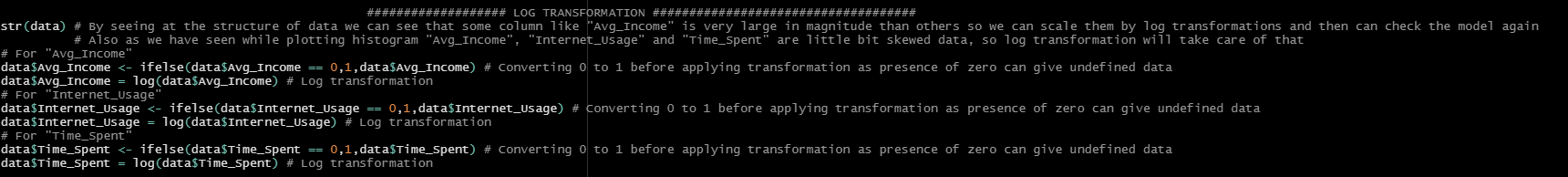
**IMPROVING THE MODEL FURTHER**

After doing so much of elimination and pre-processing our final dataset looks like –

****

By seeing at the structure of data we can see that some column like "Avg\_Income" is very large in magnitude than others so we can scale them by log transformations and then can check the model again. Also as we have seen while plotting histogram "Avg\_Income", "Internet\_Usage" and "Time\_Spent" are little bit skewed data, so log transformation can take care of that.

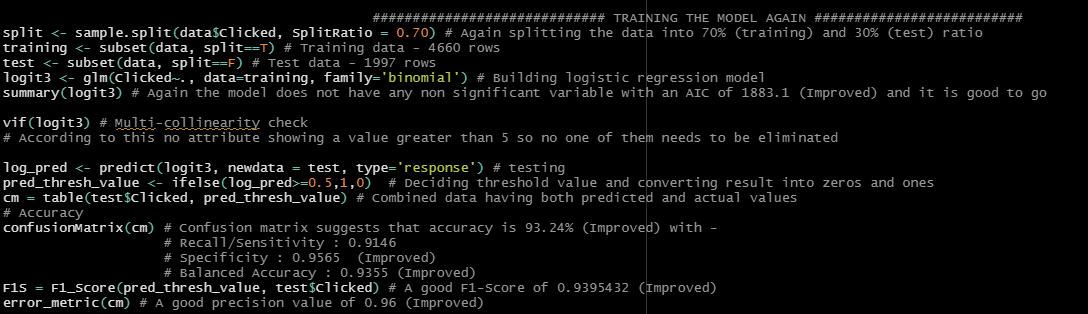
The code to apply transformation is –

****

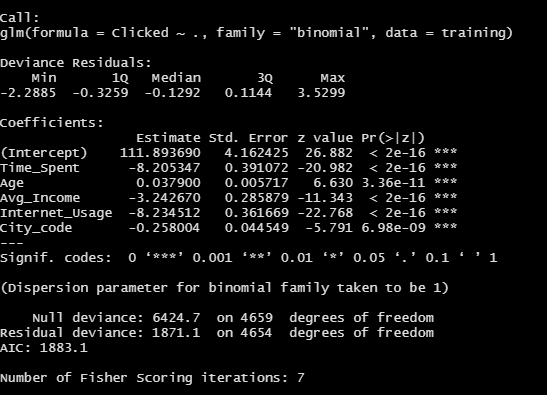
After this the data was again splited and model was again trained. This gave an AIC score of 1883.1 which is like very much improved and accuracy scores are improved also as shown below -

|  |  |
| --- | --- |
| Parameters | Value |
| Accuracy | 93.24% (Improved) |
| Precision | 0.96 (Improved) |
| Recall/Sensitivity | 0.9146 |
| F1-score | 0.9395432 (Improved) |
| Specificity | 0.9565 (Improved) |
| Balanced accuracy | 0.9355 (Improved) |

The Code for the splitting the dataset again and then again training the model is -

****

Now to make proper business decisions, we can use “summary” on our final model to get the value of parameters obtained after training. By applying “summary” on our final model, the value for different parameters obtained are -

****

According to the analysis and model obtained –

* From the data summary and histogram of age, we can conclude that younger users spend more time on the website. This suggests that users of age between 20 to 40 years can be the main target group for the marketing purpose. Product made for middle-aged groups should be marketed here rather than products intended for people over the age of 50 or 60
* It can also be concluded from the summary of data that it is a pretty popular website because users spend between 32 to 91 minutes on it in one session. which simply suggests that company can also go for costly advertisements here
* The summary of the data also suggests that average income of people visiting this website ranges from 13996 to 79485. It means that site visitors are people belonging to different social class and hence the company can advertise both its low cost product as well as high cost product here
* In addition to the third point the average income has negative correlation with the decision that whether a user will click on an AD or not (from parameters value in the final model). It means people having less average income are more likely to click on the ad as compared to people with higher income. So, company should advertise both its high cost and low cost products but the main focus should be on advertising low and average cost products on this website
* Finally it can be concluded that the decision whether a particular user is going to click on an AD or not depends on where he lives, depends on his age, what he earns, how much time he spends on that website and audiences should be targeted on the basis of these variables only rather than targeting on the basis of parameters like country, gender, time period, day, date or year