Statistical Report on the “Advertising” dataset

from kaggle.com

Alessandro Platania
Gaia Sandra Schillaci

A.Y. 2021-2022

Prof. Salvatore Ingrassia

Index

0. Introduction

The scope of this report is to realize a model that is able to predict, according to some information, whether a consumer will click on an advertisement.

The variable that represents this behaviour is *Clicked.on.Ad*: it can have "0" or "1" as values, meaning that the adv has been clicked or not.

Data have been split in 3 parts, different in terms of number of observations but equal in terms of variables: the training set, that contains 60% of the units of the original dataset (600); the validation set that contains about 20% of the units of the original dataset (200); the test set, that contains about 20% of the units of the original dataset (200). Using the training set and the validation set, we have to find a model which fits best the data in order to predict the response of *Clicked.on.Ad* on the test set.

Data are taken from kaggle.com

Below, the input variables and the response variable are listed in the same order as they are in the dataset.

	Variables name	Type	Meaning
Input Variables	*Daily.Time.Spent.on.Site*	num	Time spent by the consumer on a site in minutes
	Age	int	The consumer's age in years
	Area.Income	num	Average income of geographical area of consumer
	Daily.Internet.Usage	num	Average minutes in a day consumer is on the internet
	Ad.Topic.Line	string	Headline of the advertisement
	City	string	City of the consumer
	Male	binary	Whether or not a consumer was male
	Country	string	Country of the consumer
	Timestamp	POSIXlt	Time at which consumer clicked on an Ad or closed the window
Response Variable	*Clicked.on.Ad*	binary	Whether or not a consumer clicked on an advertisement

Table 0.1 – All the variables explained

As shown, we have 4 numerical/int variables, 3 categorical variables, 2 binary variables (including the response) and the timestamp variable.

There aren't missing values.

In performing the analysis, we want to answer some important questions:

1. *Which model best fit the data? How precise is it to describe our data?*
2. *Which variables contributes to the prediction of the behaviour of the consumer on clicking an ad?*
3. *Is there a variable (or more than one) that is not relevant for our analysis?*
4. *How accurately can we estimate the effect of each variable on Clicked.on.Ad?*
5. *What can we observe analysing the relationships between the target variable and the predictors?*

Then, we start performing the exploratory data analysis to show the behaviour of these variables of our dataset.

1. Exploratory Data Analysis

1.1 Univariate Analysis

Our analysis starts with the training set (also called learning set).

Quantitative variables:

- *Daily.Time.Spent.on.Site*

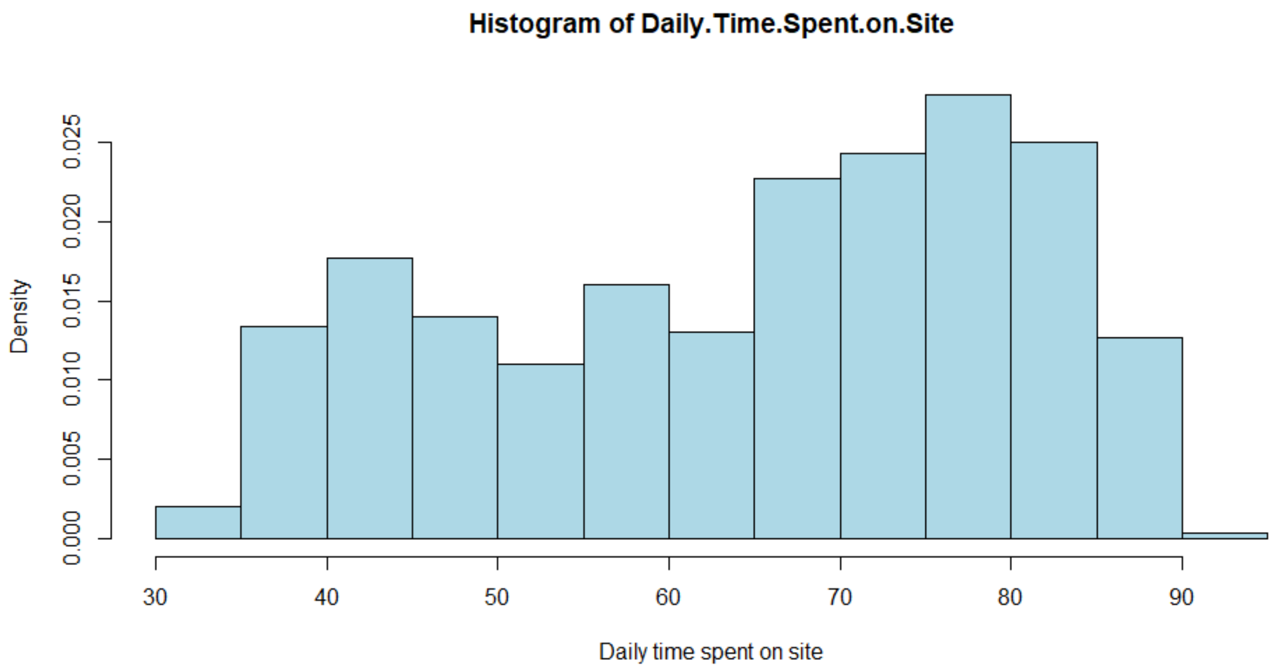


Figure 1.1 - Histogram

Min.	1st Qu.	Median	Mean	3rd Qu.	Max
32.60	50.62	67.88	64.78	78.18	91.43

```
skewness(Daily.Time.Spent.on.Site)

## [1] -0.3539789

kurtosis(Daily.Time.Spent.on.Site)

## [1] 1.893923
```

From the graph we can notice the presence of two peaks, yielding us thinking that there may be two subpopulation that influence the result of whole histogram. According to the values of skewness and kurtosis, we can notice that the distribution is negatively skewed and with a low positive kurtosis. It means that most of the values are plotted on the right side of the graph (higher concentration of large values rather than small), even though the low level of kurtosis highlights the light tail of the distribution (Platykurtic distribution).

- *Age*

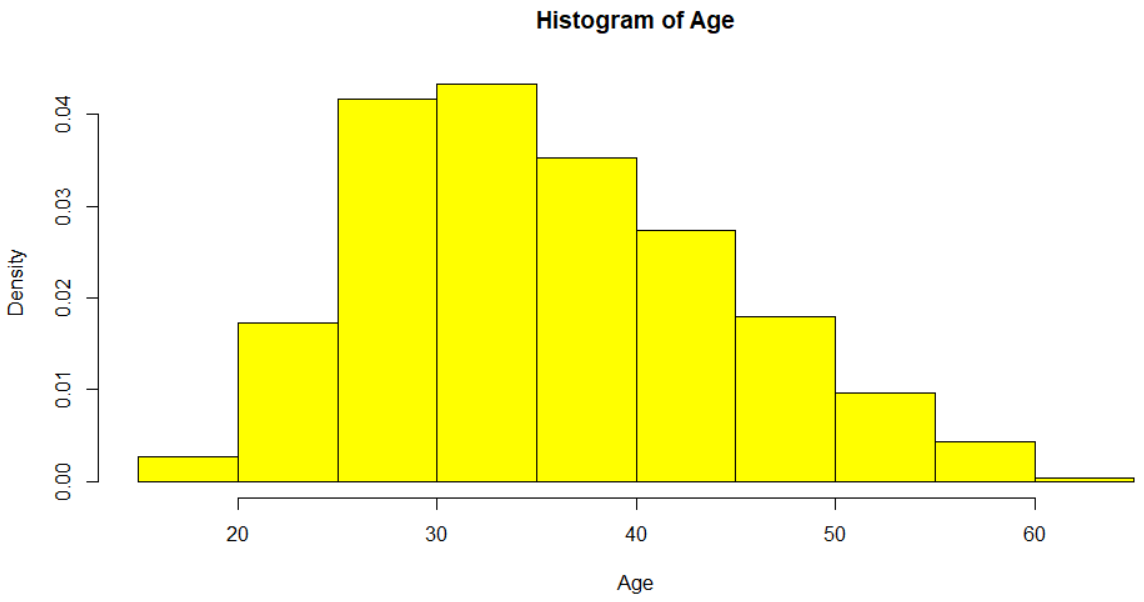


Figure 1.2 – Histogram

Min.	1st Qu.	Median	Mean	3rd Qu.	Max
19.00	29.00	35.00	36.01	42.00	61.00

```
skewness(Age)

## [1] 0.4967485

kurtosis(Age)

## [1] 2.590557
```

From the graph and the values of skewness and kurtosis, we can notice that the distribution is positively skewed and with a low positive kurtosis. It means that most of the values are plots on the left side of the graph (more small values than large), even though the low level of kurtosis highlights the light tail of the distribution (Platykurtic distribution), with a value quite near the one of the normal distribution (kurtosis = 3)

- *Area.Income*

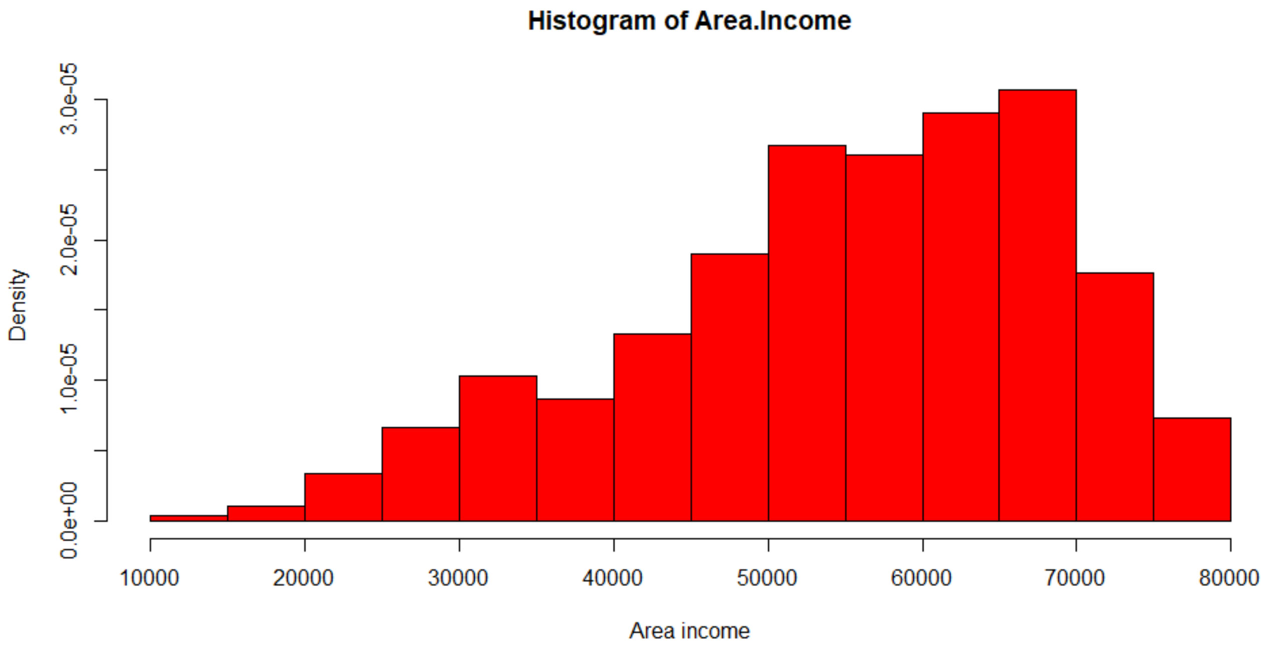


Figure 1.3 – Histogram

Min.	1st Qu.	Median	Mean	3rd Qu.	Max
1458	47117	57024	55154	65883	79485

```
skewness(Area.Income)

## [1] -0.5898708

kurtosis(Area.Income)

## [1] 2.737905
```

From the graph and the values of skewness and kurtosis, we can notice that the distribution is negatively skewed and with a low positive kurtosis. It means that most of the values are plots on the right side of the graph (higher concentration of large values rather than small), even though the low level of kurtosis highlights the light tail of the distribution (Platykurtic distribution), with a value quite near the one of the normal distribution (kurtosis = 3)

- *Daily.Internet.Usage*

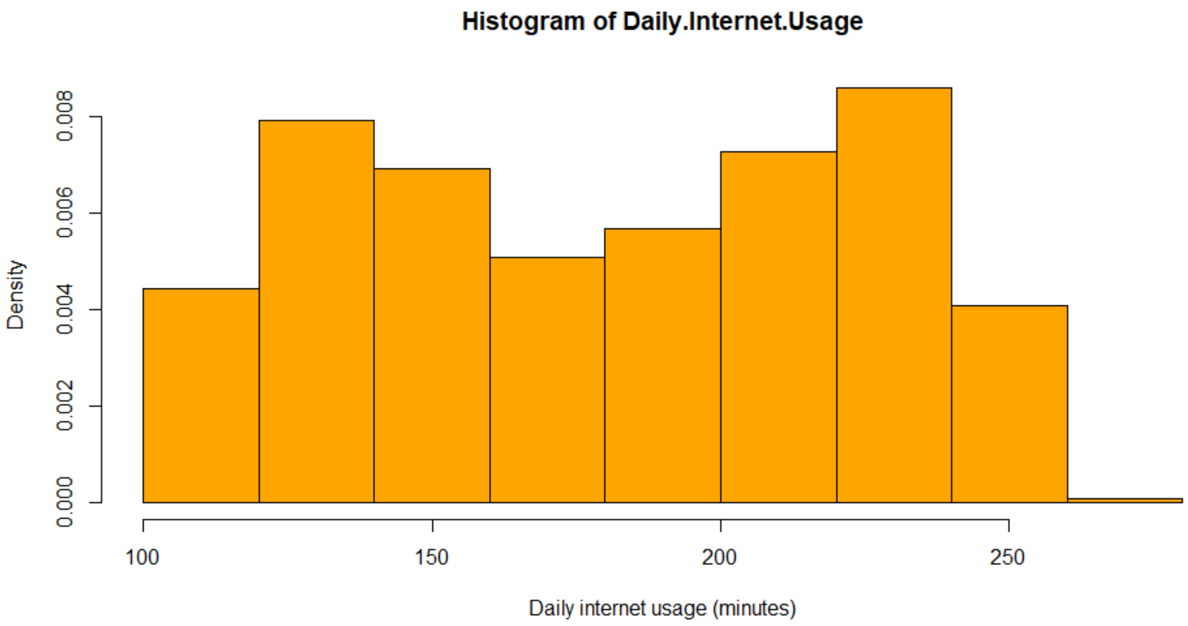


Figure 1.4 - Histrogram

Min.	1st Qu.	Median	Mean	3rd Qu.	Max
104.8	140.7	184.0	180.7	221.5	261.5

```
skewness(Daily.Internet.Usage)

## [1] -0.03172338

kurtosis(Daily.Internet.Usage)

## [1] 1.680776
```

From the graph we can notice the presence of two peaks, yielding us thinking that there may be two subpopulation that influence the result of whole histogram. According to the values of skewness and kurtosis, we can notice that the distribution is negatively skewed and with a low positive kurtosis. It means that most of the values are plots on the right side of the graph, even though the low level of kurtosis highlights the light tail of the distribution (Platykurtic distribution).

- *Male*

0	1
325 *54%*	275 *46%*

0 → people who are NOT male
1 → people who are male

As we can see, the dataset contains more females than males.

- *Clicked.on.Ad*

0	**1**
300	300
50%	*50%*

0 → people who did NOT click on the ad
1 → people who clicked on the ad

Considering this variable, the dataset is perfectly balanced.

Qualitative variables:

The variable *Ad.Topic.Line* has unique values, while *City* and *Countries* presents some values that are shared between the observations, even though the frequencies are so low that are not easily visualizable for all the dataset. There is also another qualitative variable, *Timestamp*, but obviously it has unique values.

1.2 Multivariate Analysis

We computed the correlation matrix.

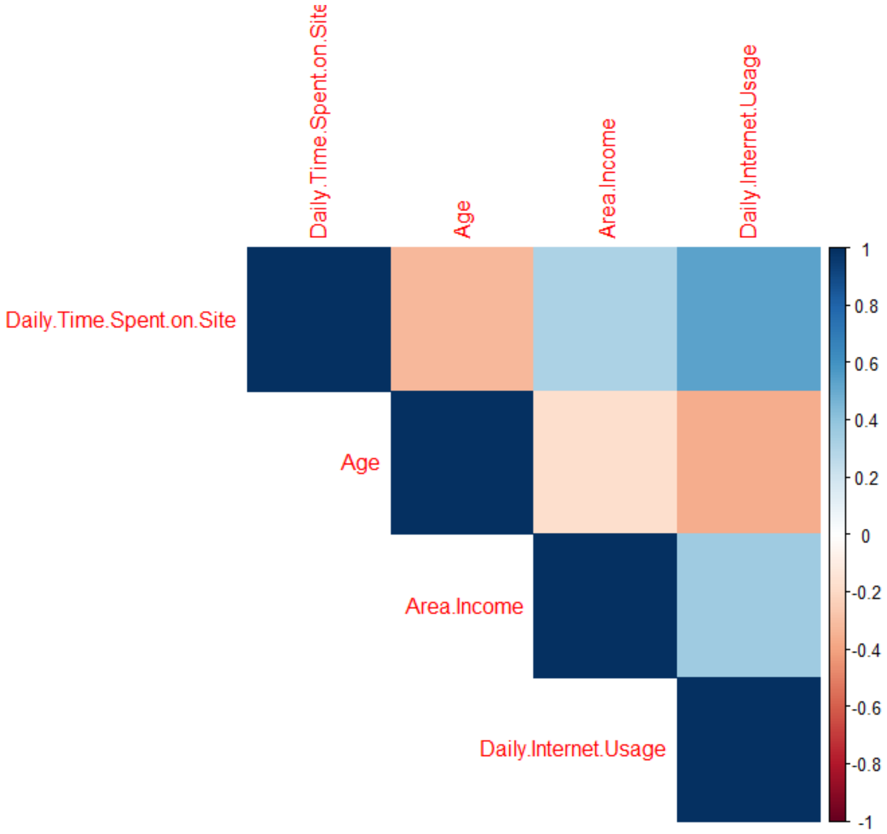


Figure 1.5 – Correlation Matrix based on colors

	Daily.Time.Spent. on.Site	**Age**	**Area.Income**	**Daily.Internet. Usage**
Daily.Time.Spent.on.Site	1.000000000	-0.32220667	0.31248125	0.53964749
Age	-0.322206673	1.000000000	0.31248125	-0.36670982
Area.Income	0.312481254	-0.17432555	1.000000000	0.35634817
Daily.Internet.Usage	0.539647492	-0.36670982	0.35634817	1.00000000

Table 1.1 – Correlation Matrix based on numeric values

From the correlation matrix we can observe an interesting positive correlation between *Daily.Internet.Usage* with *Daily.Time.Spend.on.Site,* while the same *Daily.Internet.Usage* with the other two remaining variables has a low positive correlation (with *Area.Income*) and a low negative correlation (with *Age*).

In the scatterplot matrix below, it is described the conditional distribution between the quantitative variables and the response *Clicked.on.Ad.*

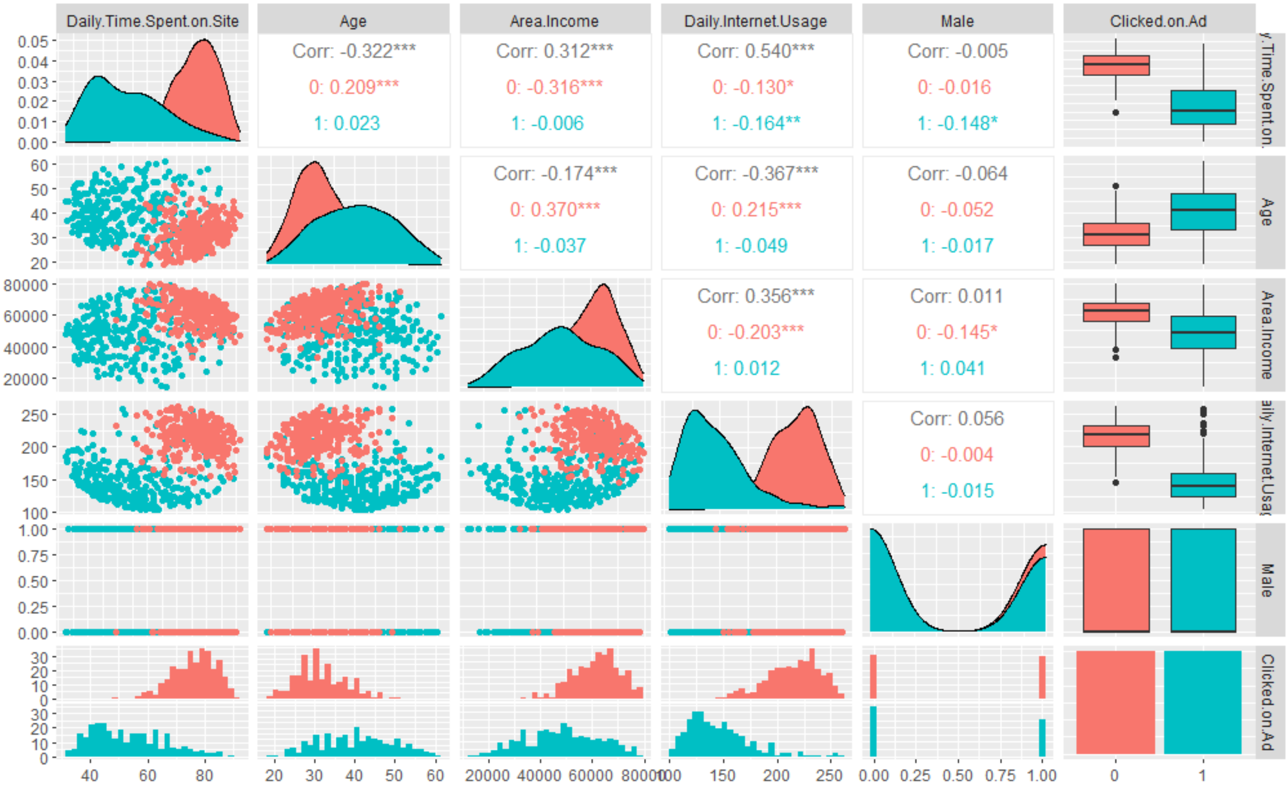


Figure 1.6 – A complete point of view on the relationships between the variables

By looking at the diagonal, we can see the probability density functions of the predictors with respect to our target variable. For instance, we can notice how the more the daily time spent on site increases, the less people click on an ad (similar situation also considering the daily internet usage), while the gender (variable *Male*) seems not to represent an element that gives us relevant information, since the behaviour doesn’t change according to it. Considering the *Age* variable, younger people tend not to click on an ad, as well as the people who have a higher income (according to the *Area.Income* variable in this case).

Looking at the boxplot, we can notice that *Daily.Internet.Usage* presents lots of outliers (especially in the case for which *Clicked.on.Ad* is equal to 1, meaning that even though people who click on ads are those who generally spend less time on internet, there are some exceptions). There are also outliers for *Daily.Time.Spent.on.Site*, *Age* and *Area.Income*, all when *Clicked.on.Ad* is equal to 0.

2. Modelling of the training data

After having performed the exploratory data analysis, the next step is to model the training data in order to find the best fitting. We will proceed according to three different approaches:

- Logistic regression
- Random Forests
- Neural Networks

2.1 Logistic Regression

Logistic regression is a process of modelling the probability of a discrete outcome given input variables.

We are going to perform logistic regression with all the predictors.

	Estimate	Std. Error	Z value	P-value
(Intercept)	3.426e+01	4.896e+00	6.997	2.61e-12
Daily.Time.Spent.on.Site	-2.437e-01	3.786e-02	-6.439	1.21e-10
Age	2.183e-01	4.307e-02	5.069	4.00e-07
Area.Income	-1.699e-04	3.097e-05	-5.486	4.10e-08
Daily.Internet.Usage	-7.788e-02	1.160e-02	-6.716	1.87e-11
Male	-8.146e-01	5.945e-01	-1.370	0.171

AIC: 95.514

Table 2.1 – Logistic regression with all the predictors

The first thing that stands out is that the variable *Male* has a very high p-value (much higher than the suggested reference of 0.05), leading us thinking that the impact of this variable may not be so much relevant with respect to the whole model. Consequently, we performed again the logistic regression, excluding that variable.

	Estimate	Std. Error	Z value	P-value
(Intercept)	3.362e+01	4.843e+00	6.943	3.83e-12
Daily.Time.Spent.on.Site	-2.407e-01	3.743e-02	-6.432	1.26e-10
Age	2.158e-01	4.128e-02	5.228	1.71e-07
Area.Income	-1.663e-04	2.959e-05	-5.619	1.92e-08
Daily.Internet.Usage	-7.858e-02	1.156e-02	-6.800	1.05e-11

AIC: 95.442

Table 2.2 - Logistic regression without the predictor 'Male'

We can see that removing the variable *Male* leads to an even lower AIC.

According to the results obtained, we can draw interesting information. First of all, the big difference between the Null deviance and the Residual deviance suggests that the model with these variables explains a lot better the data with respect to a model with only the intercept (the lower the value of the deviance, the more it states that the model we are using fits data well).

We can highlight that *Daily.Time.Spent.on.Site, Area.Income, Daily.Internet.Usage* have a negative estimate: it means that an increase of these variables is associated with a decrease in the probability of clicking an ad. For instance, keeping all other predictors constant, the odd ratio of *Clicked.on.Ad* for having a one-unit increase of *Daily.Time.Spent.on.Site* is 0.2407 lower (or we can say that the log-odd ratio of *Clicked.on.Ad* for having a one-unit increase of *Daily.Time.Spent.on.Site* decreases by 0.2407 units).

Next, we compute the confusion matrix, which is a contingency table containing the information about actual and predicted classifications:

	Actual classes		
Predicted	0	1	Sum
0	295	10	305
1	5	290	295
Sum	300	300	600

From here, we can calculate the training error, i.e. the misclassification of the observations in the training set.

```
mean(pred.glm2 != Clicked.on.Ad)

## [1] 0.025
```

Our analysis shows that just 2.5% of our observations are not classified correctly. The model seems to work well. We can see also that performing stepAIC (step-by-step iterative construction of a regression model that involves the selection of independent variables to be used in a final model) we are led to an analysis that doesn’t involve the variable *Male*, obtaining the same training error.

Moreover, we performed another analysis with the help of the ROC Curve, that is used to assess the accuracy of a continuous measurement for predicting a binary outcome. The accuracy of a test can be evaluated by considering the true positive rate and the false positive rate.

The overall performance of a classifier, summarized over all possible thresholds, is given by the Area Under the ROC Curve (AUC). An ideal ROC curve will hug the top left corner, so the larger the AUC the better the classifier. In the ideal case, AUC = 1.

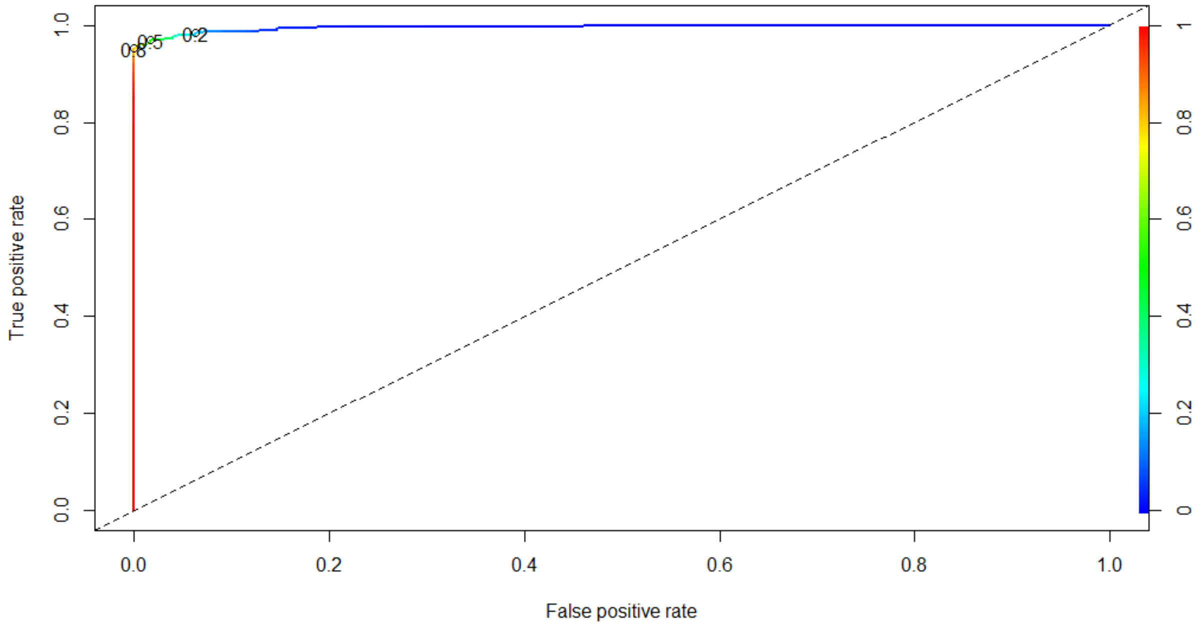


Figure 2.1 – ROC Curve

AUC = 0.9959333

We have obtained a very high value of AUC (almost 1), so we can confirm that the model produces overall accurate results.

2.2 Random Forests

The second method applied to fit our data is the so-called Random forest. It is an ensemble method (a learning techniques that combines several base models in order to produce one optimal predictive model) with the aim of constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees.

By default, Random Forest uses $\sqrt{p}$ variables when building a random forest of classification trees, so in this case 2 variables have been used randomly for each split.

Below, it is shown the importance of the variables according to two indices, the Mean Decrease Accuracy and the Mean Decrease Gini index. The Mean Decrease Accuracy refers to the mean decrease of accuracy predictions on the OOB (out of bag) samples, when a given variable is excluded by the model; the Mean Decrease in Gini coefficient is a measure of how each variable contributes to the homogeneity of the nodes and leaves (*foglie*) in the resulting random forest*.

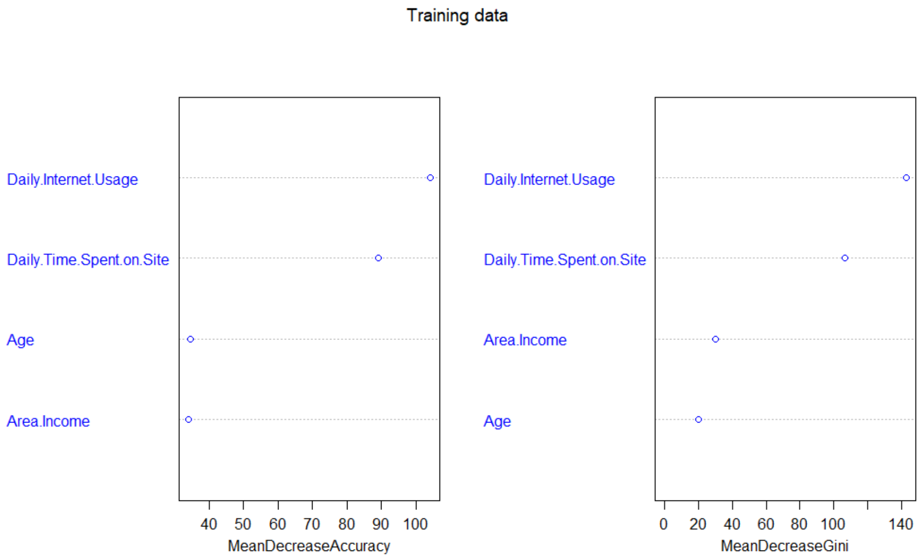


Figure 2.2 - Graphics on Mean Decrease Accuracy and Mean Decrease in Gini coefficient

It's clear that across all of the trees considered in the random forest, *Daily.Internet.Usage* and *Daily.Time.Spent.on.Site* are the most relevant predictors, since their levels of the cited indices are a lot higher than for the other variables, meaning that their absence leads to a relevant decrease in accuracy and that their contribution to the homogeneity of the nodes and leaves (*foglie*) in the random forest is fundamental.

Using Random Forests yields us to the following results:

	Actual classes		
Predicted	0	1	Class error
0	290	10	0.03333333
1	9	291	0.03000000

```
OOB estimate of  error rate: 3.17%
```

As we can see, the general OOB estimate of error rate (3.17%), even though it is a bit higher than the error rate obtained with the logistic regression (2.5%).

** as reference, see Variable importance plot (mean decrease accuracy and mean decrease Gini). (figshare.com)*

2.3 Neural Networks

The third approach to fit our data is through Neural Networks.

With the help of the *keras* package, we realized a model with one hidden layer with 4 neurons and an output with 2 nodes, according to the possible results we can obtain (0 or 1). The *epoch* value (50) counts the number of times an equivalent of the full training set has been processed.

We plot the model in the training set.

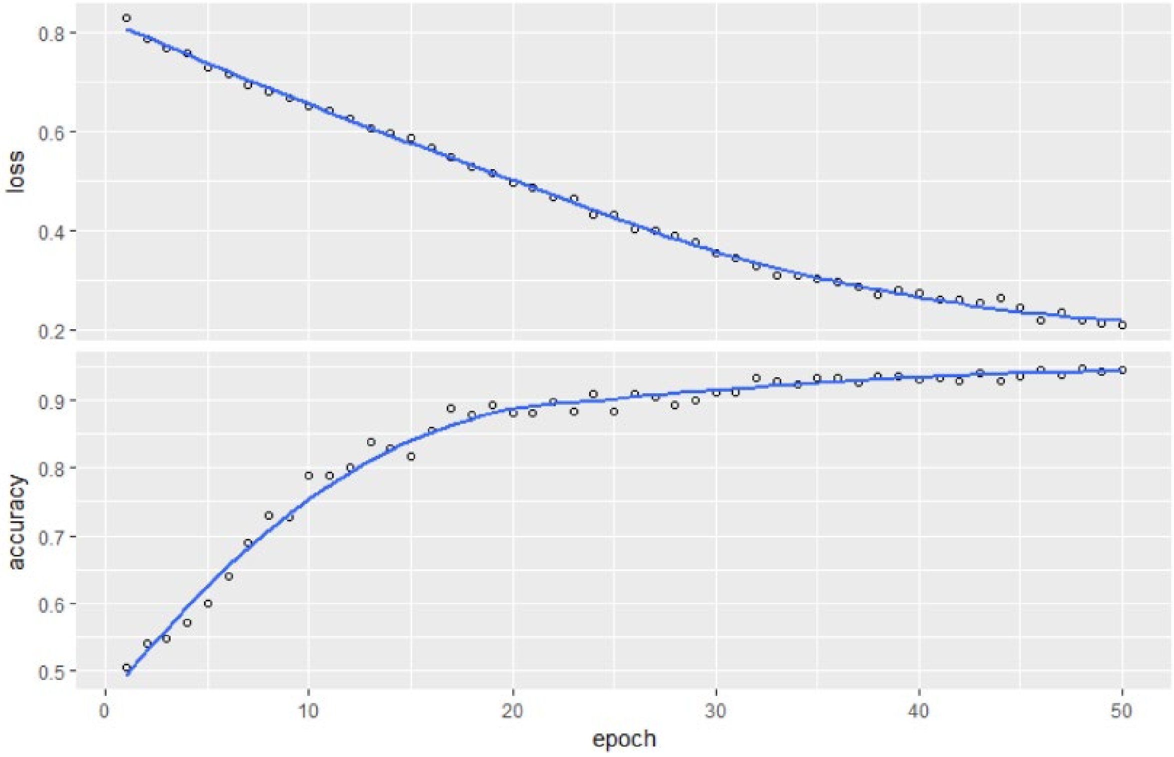
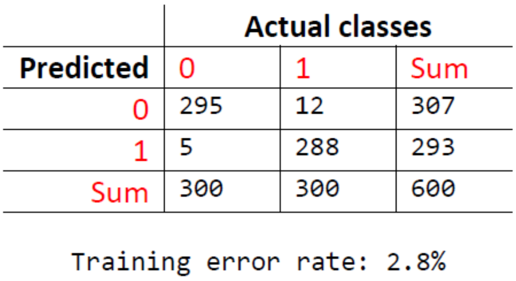


Figure 2.3 – The increment in accuracy according to the increment of the epochs

The accuracy obtained is 97%, a good value but not as good as the accuracy obtained through the logistic regression performed before.

We decided to perform another Neural Network, this time with the help of *neuralnet* package. The model has just one neuron (in one hidden layer). The value of *rep* has been set equal to 3, meaning that the function will create multiple starting weights and fit all of them, then the best one is chosen and realized the confusion matrix below:



	Actual classes		
Predicted	0	1	Sum
0	295	12	307
1	5	288	293
Sum	300	300	600

```
Training error rate: 2.8%
```

According to the results obtained, the error rate is 2.8%, very similar to the one obtained in the previous Neural Network model.

3. Choosing the best model

In order to understand which is the best model to predict the target variable *Clicked.on.Ad*, we have applied the same models used on the training set also into the validation set.

3.1 Logistic Regression

We compute the confusion matrix, to see the misclassifications between the real and the predicted outcome.

	Actual classes		
Predicted	0	1	Sum
0	96	5	101
1	4	95	99
Sum	100	100	200

```
mean(pred.val.glm != validation_data$Clicked.on.Ad)

## [1] 0.045
```

The fitted model yields to a test error rate of 4.5%, a bit higher than the 2.5% of the model with the training data.

Moreover, we can see that there isn't a particular tendency to predict false-positives rather then false-negatives, since the number of errors in both cases is almost the same: in fact, just the 4% of negative observations have been predicted as positives and the 5% of positive obs have been predicted as negatives.

3.2 Random Forests

We perform the same computation consider the model realized with the Random Forests approach.

	Actual classes		
Predicted	0	1	Sum
0	95	6	101
1	5	94	99
Sum	100	100	200

```
mean(rf.test != validation_data$Clicked.on.Ad)

## [1] 0.055
```

The fitted model yields to a test error rate of 5.5%, a bit higher than the 3.17% of the model with the training data and even higher compared to test error rate given by the logistic regression model.

As before, there is consistency in the errors, since there is no big difference between false-positives and false-negatives

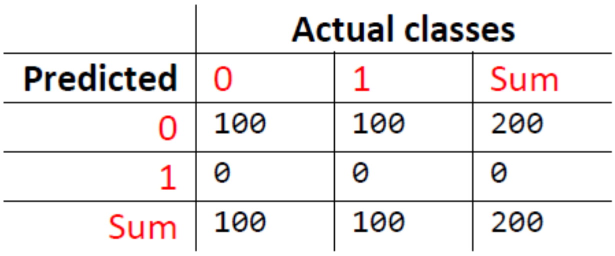
3.3 Neural Networks

To conclude, we perform on the validation data set the Neural Networks model realized with the *keras* and *neuralnet* package.

```
       loss   accuracy

##   0.1675925 0.9600000
```

Model with keras package



	Actual classes		
Predicted	0	1	Sum
0	100	100	200
1	0	0	0
Sum	100	100	200

Model with neuralnet package

```
Error rate: 50%
```

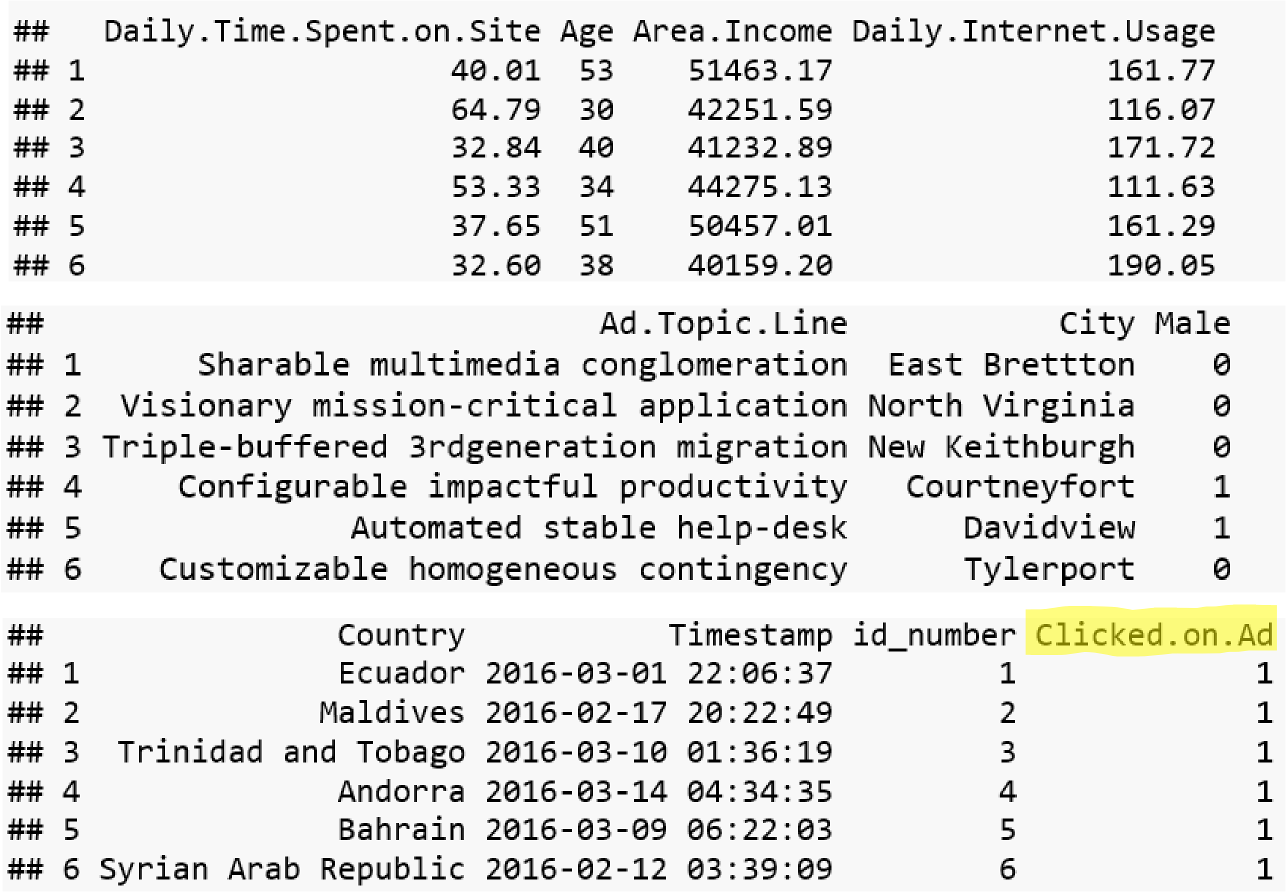
The model realized with *keras* package shows an accuracy rate of 96% while the second with *neuralnet* has an accuracy of 50%. The former model seems to be relevant since it shows similar results to those obtained in the training set, while the latter is not able to make accurate prediction to dataset different from the one in which it was trained.

4. Results

After having trained different models on the training set and tested on the validation set, our decision was to choose logistic regression as the best model to predict the target variable since it seems to be a good trade-off between complexity and overall results obtained.

The next step is to apply the chosen model to the test data. Our expectations are to obtain a test error not far from the 4.5% obtained in the validation set, since the model seems to be robust according to the results of our analysis. Currently, we don’t know which are the actual results.

Here it is shown a preview of the results on the test set.



```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    40.01  53    51463.17               161.77
## 2                    64.79  30    42251.59               116.07
## 3                    32.84  40    41232.89               171.72
## 4                    53.33  34    44275.13               111.63
## 5                    37.65  51    50457.01               161.29
## 6                    32.60  38    40159.20               190.05
```

```
##                             Ad.Topic.Line           City Male
## 1      Sharable multimedia conglomeration  East Brettton    0
## 2  Visionary mission-critical application North Virginia    0
## 3 Triple-buffered 3rdgeneration migration New Keithburgh    0
## 4     Configurable impactful productivity   Courtneyfort    1
## 5               Automated stable help-desk      Davidview    1
## 6    Customizable homogeneous contingency      Tylerport    0
```

```
##               Country           Timestamp id_number Clicked.on.Ad
## 1             Ecuador 2016-03-01 22:06:37         1             1
## 2            Maldives 2016-02-17 20:22:49         2             1
## 3  Trinidad and Tobago 2016-03-10 01:36:19         3             1
## 4             Andorra 2016-03-14 04:34:35         4             1
## 5             Bahrain 2016-03-09 06:22:03         5             1
## 6 Syrian Arab Republic 2016-02-12 03:39:09         6             1
```

Table 4.1 – An overview of what’s inside the final test set

5. Conclusions

To conclude, we highlight the questions proposed at the beginning of the report and answer according to the results obtained.

1. ***Which model best fit the data? How precise is it to describe our data?***
   The best model is generated through the logistic regression and it yields to a training error of the 2.5% in the training set and 4.5% in the validation set. We don't know how accurate it is in test set.

2. ***Is there a variable (or more than one) that is not relevant for our analysis?***
   We have seen that the variable *Male* does not have a relevant impact on our analysis and it has been removed.

3. ***Which variable contributes to the prediction of the behaviour of the consumer on clicking an ad?***
   According to the logistic regression we can affirm that *Daily.Internet.Usage, Age, Area.Income* and *Daily.Time.Spent.on.Site* are all relevant in the contribution of the prediction of the target variable.

4. ***How accurately can we estimate the effect of each variable on Clicked.on.Ad?***
   The effect of each variable can be observed by looking at the value of their estimate, since it refers to the increase in the probability of clicking on an ad. *Daily.Time.Spent.on.Site* and *Age* have the highest estimate, leading us thinking their effect on the target variable is strong. For instance, *Age* has an estimate of `2.183e-01`: it means that a unit increase of *Age* is associated with an increase in the log-odds of *Clicked.on.Ad* by 0.2183 units.

5. ***What can we observe analysing the relationships between the target variable and the predictors?***
   As stated when the multivariate analysis was performed, we can notice how the more the daily time spent on site increases, the less people click on an ad (same for the daily internet usage). Considering the Age variable, younger people tend not to click on an ad, as well as the people who have an higher income (according to the *Area.Income* variable in this case). From the boxplot we can notice that *Daily.Internet.Usage* presents lots of outliers (especially in the case for which *Clicked.on.Ad* is equal to 1, meaning that even though people who click on ads are those who generally spend less time on internet, there are some exceptions). There are also outliers for *Daily.Time.Spent.on.Site*, *Age* and *Area.Income*, all when *Clicked.on.Ad* is equal to 0.

Appendix

```
#detach(training_data)
library(binaryLogic)

## Warning: package 'binaryLogic' was built under R version 4.1.3

library(MASS)
library(corrplot)

## corrplot 0.92 loaded

library(moments)
library(GGally)

## Warning: package 'GGally' was built under R version 4.1.3

## Loading required package: ggplot2

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

library(ROCR)

## Warning: package 'ROCR' was built under R version 4.1.3

library(tree)

## Warning: package 'tree' was built under R version 4.1.3

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.1.3

## randomForest 4.7-1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

library(neuralnet)

## Warning: package 'neuralnet' was built under R version 4.1.3

##
## Attaching package: 'neuralnet'

## The following object is masked from 'package:ROCR':
##
##     prediction
```

```
#install.packages("tensorflow")
library(tensorflow)
library(keras)

## Warning: package 'keras' was built under R version 4.1.3

#install_tensorflow()

training_data <- read.csv("advertising_train.csv", header = TRUE)
training_data$X <- NULL

sum(is.na(training_data))

## 0

attach(training_data)
#as.factor(Male)
#as.factor(Clicked.on.Ad)
#training_data$Male <- as.binary(training_data$Male)
#training_data$Clicked.on.Ad <- as.binary(training_data$Clicked.on.Ad)
training_data$Ad.Topic.Line <- as.factor(training_data$Ad.Topic.Line)
training_data$City <- as.factor(training_data$City)
training_data$Country <- as.factor(training_data$Country)
training_data$Timestamp <- as.factor(training_data$Timestamp)

summary(Daily.Time.Spent.on.Site)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   32.60   50.62   67.88   64.78   78.18   91.43

hist(Daily.Time.Spent.on.Site, freq = FALSE, xlab = "Daily time spent on site",
col = "light blue")
```

```
skewness(Daily.Time.Spent.on.Site)

## [1] -0.3539789

kurtosis(Daily.Time.Spent.on.Site)

## [1] 1.893923

summary(Age)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.00   29.00   35.00   36.01   42.00   61.00

hist(Age, freq = FALSE, xlab = "Age", col = "yellow")
```

```
skewness(Age)

## [1] 0.4967485

kurtosis(Age)

## [1] 2.590557
```

```
summary(Area.Income)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14548   47117   57024   55154   65883   79485

hist(Area.Income, freq = FALSE, xlab = "Area income", col = "red")
```

```
skewness(Area.Income)

## [1] -0.5898708

kurtosis(Area.Income)

## [1] 2.737905

summary(Daily.Internet.Usage)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   104.8   140.7   184.0   180.7   221.5   261.5

hist(Daily.Internet.Usage, freq = FALSE, xlab = "Daily internet usage (minutes)
", col = "orange")
```

```
skewness(Daily.Internet.Usage)

## [1] -0.03172338

kurtosis(Daily.Internet.Usage)

## [1] 1.680776

male_table <- table(training_data$Male)
male_perc <- male_table / length(training_data$Male)
male_perc

##
##         0         1
## 0.5416667 0.4583333

table(Clicked.on.Ad)

## Clicked.on.Ad
##   0   1
## 300 300

summary(Ad.Topic.Line, 5)

##    Length     Class      Mode
##       600 character character

summary(City, 15)

##    Length     Class      Mode
##       600 character character

summary(Country,15)
```

```
##    Length     Class      Mode
##       600 character character

#Multivariate Analysis
training_data_quant_var <- training_data[, c(1,2,3,4)]
cor_quant.var <- cor(training_data_quant_var)
corrplot(cor_quant.var, method="color", type="upper")
```

```
cor_quant.var

##                            Daily.Time.Spent.on.Site        Age Area.Income
## Daily.Time.Spent.on.Site                1.0000000 -0.3222067   0.3124813
## Age                                    -0.3222067  1.0000000  -0.1743255
## Area.Income                             0.3124813 -0.1743255   1.0000000
## Daily.Internet.Usage                    0.5396475 -0.3667098   0.3563482
##                            Daily.Internet.Usage
## Daily.Time.Spent.on.Site            0.5396475
## Age                                -0.3667098
## Area.Income                         0.3563482
## Daily.Internet.Usage                1.0000000

Clicked.on.Ad_factorial <- as.factor(training_data$Clicked.on.Ad)
ggpairs(training_data_quant_var, aes(colour= Clicked.on.Ad_factorial))
```

```
#LOGISTIC REGRESSION
#Logistic regression con tutte le variabili
glm.fit <- glm(Clicked.on.Ad ~ Daily.Time.Spent.on.Site + Age + Area.Income + D
aily.Internet.Usage + Male, family = binomial, data = training_data)
summary(glm.fit)

## 
## Call:
## glm(formula = Clicked.on.Ad ~ Daily.Time.Spent.on.Site + Age + 
##     Area.Income + Daily.Internet.Usage + Male, family = binomial, 
##     data = training_data)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.6047  -0.0845  -0.0148   0.0049   3.5128  
## 
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)    
## (Intercept)               3.426e+01  4.896e+00   6.997 2.61e-12 ***
## Daily.Time.Spent.on.Site -2.437e-01  3.786e-02  -6.439 1.21e-10 ***
## Age                       2.183e-01  4.307e-02   5.069 4.00e-07 ***
## Area.Income              -1.699e-04  3.097e-05  -5.486 4.10e-08 ***
## Daily.Internet.Usage     -7.788e-02  1.160e-02  -6.716 1.87e-11 ***
## Male                     -8.146e-01  5.945e-01  -1.370    0.171    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
```

```
##     Null deviance: 831.777  on 599  degrees of freedom
## Residual deviance:  83.514  on 594  degrees of freedom
## AIC: 95.514
##
## Number of Fisher Scoring iterations: 9
```

```
glm.probs <- predict(glm.fit, type = "response")
pred.glm <- rep(0, length(glm.probs))
pred.glm[glm.probs > 0.5] <- 1
table(pred.glm, Clicked.on.Ad)
```

```
##         Clicked.on.Ad
## pred.glm   0   1
##        0 294   8
##        1   6 292
```

```
conf.matrix <- addmargins(table(pred.glm, Clicked.on.Ad))
conf.matrix
```

```
##         Clicked.on.Ad
## pred.glm   0   1 Sum
##      0   294   8 302
##      1     6 292 298
##      Sum 300 300 600
```

```
mean(pred.glm != Clicked.on.Ad)
```

```
## [1] 0.02333333
```

```
search()
```

```
##  [1] ".GlobalEnv"            "training_data"         "package:keras"
##  [4] "package:tensorflow"    "package:neuralnet"     "package:randomForest"
##  [7] "package:tree"          "package:ROCR"          "package:GGally"
## [10] "package:ggplot2"       "package:moments"       "package:corrplot"
## [13] "package:MASS"          "package:binaryLogic"   "package:stats"
## [16] "package:graphics"      "package:grDevices"     "package:utils"
## [19] "package:datasets"      "package:methods"       "Autoloads"
## [22] "package:base"
```

```
#logistic regression senza la variabile Male
glm2.fit <- glm(Clicked.on.Ad ~ Daily.Time.Spent.on.Site + Age + Area.Income +
Daily.Internet.Usage, family = binomial, data = training_data)
summary(glm2.fit)
```

```
##
## Call:
## glm(formula = Clicked.on.Ad ~ Daily.Time.Spent.on.Site + Age +
##     Area.Income + Daily.Internet.Usage, family = binomial, data = training_d
ata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6306  -0.0757  -0.0176   0.0052   3.3850
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               3.362e+01  4.843e+00   6.943 3.83e-12 ***
## Daily.Time.Spent.on.Site -2.407e-01  3.743e-02  -6.432 1.26e-10 ***
## Age                       2.158e-01  4.128e-02   5.228 1.71e-07 ***
## Area.Income              -1.663e-04  2.959e-05  -5.619 1.92e-08 ***
## Daily.Internet.Usage     -7.858e-02  1.156e-02  -6.800 1.05e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 831.777  on 599  degrees of freedom
## Residual deviance:  85.442  on 595  degrees of freedom
## AIC: 95.442
##
## Number of Fisher Scoring iterations: 9
```

```
glm.probs2 <- predict(glm2.fit, type = "response")
pred.glm2 <- rep(0, length(glm.probs))
pred.glm2[glm.probs2 > 0.5] <- 1
table(pred.glm2, Clicked.on.Ad)
```

```
##          Clicked.on.Ad
## pred.glm2   0   1
##         0 295  10
##         1   5 290
```

```
conf.matrix <- addmargins(table(pred.glm2, Clicked.on.Ad))
conf.matrix
```

```
##          Clicked.on.Ad
## pred.glm2   0   1 Sum
##       0   295  10 305
##       1     5 290 295
##       Sum 300 300 600
```

```
mean(pred.glm2 != Clicked.on.Ad)
```

```
## [1] 0.025
```

```
#logistic regression con stepwise
glm3.fit = glm(Clicked.on.Ad ~ Daily.Time.Spent.on.Site + Age + Area.Income + D
aily.Internet.Usage + Male, family = binomial, data = training_data) %>% stepAI
C(trace = FALSE)
glm.probs3 <- predict(glm3.fit, type = "response")
pred_glm3 <- ifelse(glm.probs3>0.5, 1, 0)
# pred_val.nn <- rep(0, length(p_val))
# pred_val.nn[p_val > 0.5] <- 1
table(pred_glm3, Clicked.on.Ad)
```

```
##          Clicked.on.Ad
## pred_glm3   0   1
##         0 295  10
##         1   5 290
```

```
mean(pred_glm3 != Clicked.on.Ad)
```

```
## [1] 0.025
```

```
#ROC Curve
ROCPred <- ROCR::prediction(glm.probs2, Clicked.on.Ad)
ROCPerf <-performance(ROCPred, "tpr", "fpr")
plot(ROCPerf,colorize=TRUE,lwd=2)
plot(ROCPerf,colorize=TRUE,lwd=2, print.cutoffs.at=c(0.2,0.5,0.8))
abline(a=0,b=1, lty=2)
```

```
ROCauc <-performance(ROCPred, measure ="auc")
ROCauc@y.values[[1]]

## [1] 0.9959333

#RANDOM FORESTS
tree.training <- tree(Clicked.on.Ad_factorial ~ ., data = training_data_quant_v
ar)
tree.training

## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
## 
##  1) root 600 831.800 0 ( 0.50000 0.50000 )  
##    2) Daily.Internet.Usage < 179.31 291 155.900 1 ( 0.07560 0.92440 )  
##      4) Daily.Time.Spent.on.Site < 70.32 246  40.890 1 ( 0.01626 0.98374 )  
##        8) Area.Income < 69037.6 223   0.000 1 ( 0.00000 1.00000 ) *
##        9) Area.Income > 69037.6 23  21.250 1 ( 0.17391 0.82609 ) *
##      5) Daily.Time.Spent.on.Site > 70.32 45  60.570 1 ( 0.40000 0.60000 )  
##       10) Daily.Internet.Usage < 152.975 22   8.136 1 ( 0.04545 0.95455 ) *
##       11) Daily.Internet.Usage > 152.975 23  26.400 0 ( 0.73913 0.26087 ) *
##    3) Daily.Internet.Usage > 179.31 309 201.300 0 ( 0.89968 0.10032 )  
##      6) Daily.Time.Spent.on.Site < 55.995 22   8.136 1 ( 0.04545 0.95455 ) *
##      7) Daily.Time.Spent.on.Site > 55.995 287  86.790 0 ( 0.96516 0.03484 )  
##       14) Age < 49.5 282  58.070 0 ( 0.97872 0.02128 )  
##         28) Area.Income < 38168 5   6.730 1 ( 0.40000 0.60000 ) *
##         29) Area.Income > 38168 277  33.120 0 ( 0.98917 0.01083 ) *
##       15) Age > 49.5 5   5.004 1 ( 0.20000 0.80000 ) *

summary(tree.training)

## 
## Classification tree:
## tree(formula = Clicked.on.Ad_factorial ~ ., data = training_data_quant_var)
## Number of terminal nodes:  8 
## Residual mean deviance:  0.1838 = 108.8 / 592 
## Misclassification error rate: 0.03 = 18 / 600

plot(tree.training, lwd=1,type="uniform")
text(tree.training,cex=0.75,col="blue")
```

```
set.seed(1)
rf.training <- randomForest(Clicked.on.Ad_factorial ~ ., data = training_data_q
uant_var, importance = TRUE)
rf.training
```

```
##
## Call:
##  randomForest(formula = Clicked.on.Ad_factorial ~ ., data = training_data_qu
ant_var,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 3.17%
## Confusion matrix:
##     0   1 class.error
## 0 290  10  0.03333333
## 1   9 291  0.03000000
```

```
importance(rf.training)
```

```
##                                 0        1 MeanDecreaseAccuracy
## Daily.Time.Spent.on.Site 78.29259 53.31715             89.15831
## Age                      35.13174 10.32842             34.74380
## Area.Income              31.93514 17.40877             34.04446
## Daily.Internet.Usage     93.63389 57.40682            104.05590
##                          MeanDecreaseGini
## Daily.Time.Spent.on.Site        106.87157
## Age                              19.63570
## Area.Income                      29.91904
## Daily.Internet.Usage            143.06056
```

```
varImpPlot(rf.training, col="blue", main="Training data")
```

```
#NEURAL NETWORKS
#min-max normalization
training_data_quant_var$Daily.Time.Spent.on.Site <- (training_data_quant_var$Da
ily.Time.Spent.on.Site - min(training_data_quant_var$Daily.Time.Spent.on.Site))
/(max(training_data_quant_var$Daily.Time.Spent.on.Site) - min(training_data_qua
nt_var$Daily.Time.Spent.on.Site))
training_data_quant_var$Age <- (training_data_quant_var$Age - min(training_data
_quant_var$Age))/(max(training_data_quant_var$Age) - min(training_data_quant_va
r$Age))
training_data_quant_var$Area.Income <- (training_data_quant_var$Area.Income - m
in(training_data_quant_var$Area.Income))/(max(training_data_quant_var$Area.Inco
me) - min(training_data_quant_var$Area.Income))
training_data_quant_var$Daily.Internet.Usage <- (training_data_quant_var$Daily.
Internet.Usage - min(training_data_quant_var$Daily.Internet.Usage))/(max(traini
ng_data_quant_var$Daily.Internet.Usage) - min(training_data_quant_var$Daily.Int
ernet.Usage))

#building neural network
set.seed(2)
n <- neuralnet(Clicked.on.Ad ~ Daily.Time.Spent.on.Site + Age + Area.Income + D
aily.Internet.Usage,
                                          data = training_data_quant_var,
                                          hidden = 1,
                                          err.fct = "ce",
                                          linear.output = FALSE,
                                       #  lifesign = "full", #to see every feasible output
                                          rep = 3)
plot(n, rep = 2)
```

```
#prediction
output <- compute(n, training_data_quant_var, rep = 2)
head(output$net.result,11)
```

```
##                 [,1]
##   [1,] 0.999999993
##   [2,] 1.000000000
##   [3,] 1.000000000
##   [4,] 0.999999999
##   [5,] 1.000000000
##   [6,] 1.000000000
##   [7,] 0.999999999
##   [8,] 0.998036042
##   [9,] 1.000000000
##  [10,] 1.000000000
##  [11,] 0.986777442
```

```
head(training_data[1,])
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    34.66  32     48246.6               194.83
##                                       Ad.Topic.Line       City Male  Country
## 1 Customer-focused multi-tasking Internet solution Curtisport    0 Mongolia
##             Timestamp Clicked.on.Ad
## 1 2016-07-14 12:07:10             1
```

```
# output2 <- compute(n2, training_data_quant_var, rep=2)
# head(output2$net.result,30)
# head(training_data[1,])



#confusion matrix
output <- compute(n, training_data_quant_var, rep = 2)
p1 <- output$net.result
pred.nn <- rep(0, length(p1))
pred.nn[p1 > 0.5] <- 1
table(pred.nn, Clicked.on.Ad)
```

```
##        Clicked.on.Ad
## pred.nn   0   1
##       0 295  12
##       1   5 288
```

```
mean(pred.nn != Clicked.on.Ad)
```

```
## [1] 0.02833333
```

```
#provo con KERAS i NN
tensorflow::set_random_seed(42)

## Loaded Tensorflow version 2.9.0

data_scale <- scale(training_data_quant_var)
trainLabels <- to_categorical(training_data$Clicked.on.Ad)


train_NN_model <- keras_model_sequential()
train_NN_model %>% layer_dense(units = 4, activation = 'relu', input_shape = c(
4)) %>%
                  layer_dropout(rate = 0.3) %>%
                  layer_dense(units= 2, activation = "sigmoid") #because we hav
e 2 categories in the response variable
summary(train_NN_model)

## Model: "sequential"
## ___________________________________________________________________________
____
##  Layer (type)                       Output Shape                    Param #
## ===========================================================================
====
##  dense_1 (Dense)                    (None, 4)                       20
##
##  dropout (Dropout)                  (None, 4)                       0
##
##  dense (Dense)                      (None, 2)                       10
##
## ===========================================================================
====
## Total params: 30
## Trainable params: 30
## Non-trainable params: 0
## ___________________________________________________________________________
____

train_NN_model %>% compile(loss = 'binary_crossentropy', optimizer = optimizer_
rmsprop(), metrics ="accuracy")
history <- train_NN_model %>% fit(data_scale, trainLabels, epoch = 50, batch_si
ze = 32)
plot(history)

## `geom_smooth()` using formula 'y ~ x'
```

```
train_NN_model %>% evaluate(data_scale, trainLabels)

##     loss accuracy
## 0.132131 0.970000

#Validation dataset
validation_dataset <- read.csv("advertising_validation.csv", header = TRUE)
validation_dataset$X <- NULL
validation_dataset$Ad.Topic.Line <- as.factor(validation_dataset$Ad.Topic.Line)
validation_dataset$City <- as.factor(validation_dataset$City)
```

```
validation_dataset$Country <- as.factor(validation_dataset$Country)
validation_dataset$Timestamp <- as.factor(validation_dataset$Timestamp)
validation_dataset_quant_var <- validation_dataset[, c(1,2,3,4)]

#prediction with logistic regression in validation set
val.probs = predict(glm2.fit, newdata=validation_dataset, type="response")
pred.val.glm <- ifelse(val.probs > 0.5, 1, 0)
#pred.val.glm <- rep(0, length(val.probs))
#pred.val.glm[val.probs > 0.5] <- 1
table(pred.val.glm, validation_dataset$Clicked.on.Ad)

##
## pred.val.glm  0  1
##            0 96  5
##            1  4 95

mean(pred.val.glm != validation_dataset$Clicked.on.Ad)

## [1] 0.045

#prediction with random forests in validation set
rf.test = predict(rf.training, newdata=validation_dataset, type="response")
table(rf.test, validation_dataset$Clicked.on.Ad)

##
## rf.test  0  1
##       0 95  6
##       1  5 94

mean(rf.test != validation_dataset$Clicked.on.Ad)

## [1] 0.055

#prediction with NN in validation set
output_val <- compute(n, validation_dataset_quant_var, rep=2)
p_val <- output_val$net.result
pred_val.nn <- ifelse(p_val>0.5, 1, 0)
# pred_val.nn <- rep(0, length(p_val))
# pred_val.nn[p_val > 0.5] <- 1
table(pred_val.nn, validation_dataset$Clicked.on.Ad)

##
## pred_val.nn   0   1
##           0 100 100

mean(pred_val.nn != validation_dataset$Clicked.on.Ad)

## [1] 0.5

#prediction with NN CON KERAS in validation set
valLabels <- to_categorical(validation_dataset$Clicked.on.Ad)
data_scale_val <- scale(validation_dataset_quant_var)
train_NN_model %>% evaluate(data_scale_val, valLabels)

##      loss  accuracy
## 0.1675925 0.9600000
```

```
#test_data
test_data <- read.csv("advertising_test.csv", header = TRUE)
test_data$X <- NULL

test.probs = predict(glm2.fit, newdata=test_data, type="response")
pred.test.glm <- ifelse(test.probs > 0.5, 1, 0)

dframe_final <- test_data
dframe_final$Clicked.on.Ad <- pred.test.glm
head(dframe_final)

##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    40.01  53    51463.17               161.77
## 2                    64.79  30    42251.59               116.07
## 3                    32.84  40    41232.89               171.72
## 4                    53.33  34    44275.13               111.63
## 5                    37.65  51    50457.01               161.29
## 6                    32.60  38    40159.20               190.05
##                              Ad.Topic.Line           City Male
## 1       Sharable multimedia conglomeration  East Brettton    0
## 2  Visionary mission-critical application North Virginia    0
## 3 Triple-buffered 3rdgeneration migration New Keithburgh    0
## 4      Configurable impactful productivity   Courtneyfort    1
## 5               Automated stable help-desk      Davidview    1
## 6     Customizable homogeneous contingency      Tylerport    0
##                 Country           Timestamp id_number Clicked.on.Ad
## 1              Ecuador 2016-03-01 22:06:37         1             1
## 2             Maldives 2016-02-17 20:22:49         2             1
## 3  Trinidad and Tobago 2016-03-10 01:36:19         3             1
## 4              Andorra 2016-03-14 04:34:35         4             1
## 5              Bahrain 2016-03-09 06:22:03         5             1
## 6 Syrian Arab Republic 2016-02-12 03:39:09         6             1

write.csv(dframe_final,"advertising_test_results.csv", row.names = FALSE)
```