Online_Cryptography_Ads_using_Machine_Learning Models

Christopher Toromo

2022-05-26

1. Defining the Question

a) Specifying the Data Analytic Question

To create a supervised learning model to help identify which individuals are most likely to click on the ads in the blog.

b) Defining the Metric for Success

The study will be considered successful if we shall be able to get create a model that has an accuracy of 90%.

c) Understanding the context

Advertising is a means of communication with the users of a product or service. Advertisements are messages paid for by those who send them and are intended to inform or influence people who receive them. Advertising is always present, though people may not be aware of it. In today's world, advertising uses every possible media to get its message through. It does this via television, print (newspapers, magazines, journals etc), radio, press, internet, direct selling, hoardings, mailers, contests, sponsorships, posters, clothes, events, colours, sounds, visuals and even people (endorsements). In our model, we shall try to predict the probabilty of a person clicking on an ad.

d). Recording the Experimental Design

The elements of the checklist are

i). Formulate your question

ii). Read in your data

iii). Check the packaging

iv). Run str()

v). Look at the top and the bottom of your data

vi). Check your "n"s

vii). Validate with at least one external data source

viii). Try the easy solution first

ix). Challenge your solution

x). Follow up

e) Data Relevance

The dataset to use for this project can be found by following this link: https://www.bit.ly/IPAdvertisingData

2. Reading the Data

```
advertising <- read.csv("https://www.bit.ly/IPAdvertisingData")
```

3. Checking the Data

a) Checking the top data

```
head(advertising)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    68.95  35    61833.90               256.09
## 2                    80.23  31    68441.85               193.77
## 3                    69.47  26    59785.94               236.50
## 4                    74.15  29    54806.18               245.89
## 5                    68.37  35    73889.99               225.58
## 6                    59.99  23    59761.56               226.74
##                          Ad.Topic.Line           City Male    Country
## 1    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunisia
## 2    Monitored national standardization      West Jodi    1      Nauru
## 3      Organic bottom-line service-desk       Davidton    0 San Marino
## 4 Triple-buffered reciprocal time-frame West Terrifurt    1      Italy
## 5         Robust logistical utilization   South Manuel    0    Iceland
## 6       Sharable client-driven software      Jamieberg    1     Norway
##             Timestamp Clicked.on.Ad
## 1 2016-03-27 00:53:11             0
## 2 2016-04-04 01:39:02             0
## 3 2016-03-13 20:35:42             0
## 4 2016-01-10 02:31:19             0
## 5 2016-06-03 03:36:18             0
## 6 2016-05-19 14:30:17             0
```

b). Checking the bottom data

```
tail(advertising)
```

```
##      Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 995                     43.70  28    63126.96               173.01
## 996                     72.97  30    71384.57               208.58
## 997                     51.30  45    67782.17               134.42
## 998                     51.63  51    42415.72               120.37
```

```
## 999                     55.55  19    41920.79               187.95
## 1000                    45.01  26    29875.80               178.35
##                             Ad.Topic.Line          City Male
## 995         Front-line bifurcated ability  Nicholasland    0
## 996         Fundamental modular algorithm     Duffystad    1
## 997       Grass-roots cohesive monitoring   New Darlene    1
## 998          Expanded intangible solution South Jessica    1
## 999  Proactive bandwidth-monitored policy   West Steven    0
## 1000      Virtual 5thgeneration emulation   Ronniemouth    0
##                       Country           Timestamp Clicked.on.Ad
## 995                   Mayotte 2016-04-04 03:57:48             1
## 996                   Lebanon 2016-02-11 21:49:00             1
## 997  Bosnia and Herzegovina 2016-04-22 02:07:01             1
## 998                 Mongolia 2016-02-01 17:24:57             1
## 999                Guatemala 2016-03-24 02:35:54             0
## 1000                  Brazil 2016-06-03 21:43:21             1
```

c). Checking the Structure of the Dataset

```
str(advertising)
```

```
## 'data.frame':    1000 obs. of  10 variables:
##  $ Daily.Time.Spent.on.Site: num  69 80.2 69.5 74.2 68.4 ...
##  $ Age                     : int  35 31 26 29 35 23 33 48 30 20 ...
##  $ Area.Income             : num  61834 68442 59786 54806 73890 ...
##  $ Daily.Internet.Usage    : num  256 194 236 246 226 ...
##  $ Ad.Topic.Line           : chr  "Cloned 5thgeneration orchestration" "Monitored national standardi
##  $ City                    : chr  "Wrightburgh" "West Jodi" "Davidton" "West Terrifurt" ...
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Country                 : chr  "Tunisia" "Nauru" "San Marino" "Italy" ...
##  $ Timestamp               : chr  "2016-03-27 00:53:11" "2016-04-04 01:39:02" "2016-03-13 20:35:42"
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

d). Checking the shape of our data

```
dim(advertising)
```

```
## [1] 1000   10
```

We have 1000 rows and 10 columns in our dataset

4. Tidying the Dataset

a). Checking the Missing Values

```
colSums(is.na(advertising))
```

```
## Daily.Time.Spent.on.Site                      Age             Area.Income
##                        0                        0                       0
##     Daily.Internet.Usage           Ad.Topic.Line                     City
##                        0                        0                       0
##                     Male                  Country                Timestamp
##                        0                        0                       0
##            Clicked.on.Ad
##                        0
```

From the above, we can see that we do not have Missing Values in the dataset.

b). Checking for Duplicate Values

```
duplicated_rows <- advertising[duplicated(advertising),]

duplicated_rows
```

```
##  [1] Daily.Time.Spent.on.Site Age                      Area.Income
##  [4] Daily.Internet.Usage     Ad.Topic.Line            City
##  [7] Male                     Country                  Timestamp
## [10] Clicked.on.Ad
## <0 rows> (or 0-length row.names)
```

We can also see that we have 0 rows containing duplicates values. This is very import for the consitency of data.

c). Checking for Outliers

We shall use Boxplot to check for outliers in our numeric features

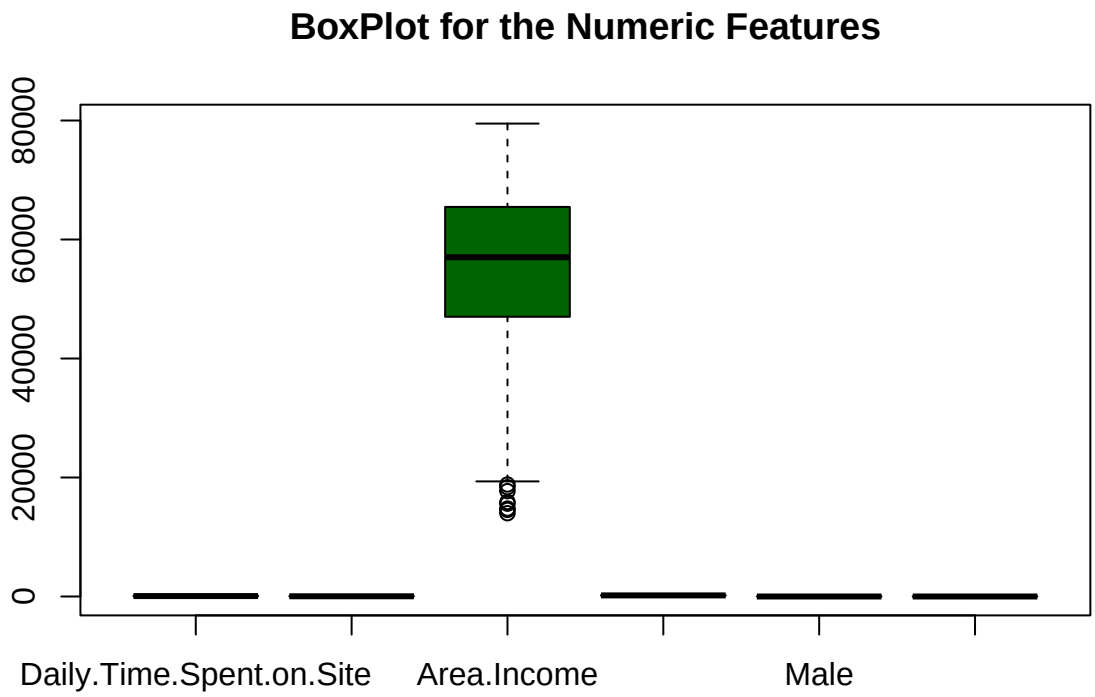
```
# Selecting Numeric columns

library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# getting numeric columns using dplyr() function
numeric_col <- select_if(advertising, is.numeric)

head(numeric_col)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage Male
## 1                    68.95  35    61833.90               256.09    0
## 2                    80.23  31    68441.85               193.77    1
## 3                    69.47  26    59785.94               236.50    0
## 4                    74.15  29    54806.18               245.89    1
## 5                    68.37  35    73889.99               225.58    0
## 6                    59.99  23    59761.56               226.74    1
##   Clicked.on.Ad
## 1             0
## 2             0
## 3             0
## 4             0
## 5             0
## 6             0
```

```r
# Plotting Boxplot for the Numeric columns

boxplot(numeric_col,main="BoxPlot for the Numeric Features", col="darkgreen")
```



We can clearly see we have outliers in our Area.Income feature. We shall be removing the outlier using the Interquartile Range Method

```
outliersValue <- boxplot.stats(advertising$Area.Income)$out

advertising$Area.Income[!advertising$Area.Income %in% outliersValue]
```

```
##   [1] 61833.90 68441.85 59785.94 54806.18 73889.99 59761.56 53852.85 24593.33
##   [9] 68862.00 55642.32 45632.51 62491.01 51636.92 51739.63 30976.00 52182.23
##  [17] 23936.86 71511.08 31087.54 23821.72 64802.33 60015.57 32635.70 61628.72
##  [25] 68962.32 64828.00 38067.08 58295.82 32708.94 46179.97 51473.28 45593.93
##  [33] 25583.29 30227.98 45580.92 61389.50 56770.79 76435.30 57425.87 27508.41
##  [41] 57691.95 59784.18 66572.39 64929.61 57519.64 53575.48 50983.75 67058.72
##  [49] 52723.34 54286.10 61526.25 58526.04 53350.11 62657.53 62722.57 67479.62
##  [57] 75254.88 52336.64 56113.37 24852.90 47708.42 64654.66 71228.44 61601.05
##  [65] 66281.46 73910.90 51317.33 51510.18 61005.87 32536.98 60248.97 74543.81
##  [73] 75509.61 42650.32 58183.04 60465.72 57009.76 54541.56 32689.04 55605.92
##  [81] 63296.87 65653.47 61652.53 30726.26 74535.94 47861.93 73600.28 58543.94
##  [89] 42696.67 37334.78 71392.53 59550.05 64264.25 64147.86 25686.34 52968.22
##  [97] 22473.08 64927.19 51868.85 69456.83 31947.65 51864.77 59593.56 48376.14
## [105] 56884.74 67186.54 46557.92 66541.05 33258.09 72272.90 60333.38 65229.13
## [113] 56067.38 37838.72 72683.35 56729.78 66815.54 60223.52 29727.79 49269.98
## [121] 57669.41 56791.75 63274.88 35466.80 68787.09 61227.59 56366.88 57868.44
## [129] 66618.21 73104.47 21644.91 53817.02 76368.31 67633.44 50335.46 41229.16
## [137] 42581.23 61617.98 70575.60 64122.36 52097.32 65953.76 60192.72 77460.07
## [145] 45716.48 65120.86 49995.63 71718.51 61770.34 69112.84 72524.86 36782.38
## [153] 66699.12 64287.78 56637.59 55787.58 61142.33 61625.87 73234.87 74166.24
## [161] 62669.59 57756.89 58019.64 50960.08 48246.60 28271.84 53767.12 43662.10
## [169] 62238.58 49030.03 76003.47 68094.85 64395.85 70053.27 72423.97 42995.80
## [177] 60309.58 38349.78 63115.34 31343.39 40763.13 36752.24 65044.59 53673.08
## [185] 43444.86 44248.52 62572.88 39840.55 32593.59 41629.86 43313.73 42993.48
## [193] 46004.31 49325.48 51633.34 63363.04 64045.93 73049.30 66624.60 77567.85
## [201] 53431.35 31265.75 74780.74 70410.11 37345.24 66107.84 62336.39 39132.64
## [209] 38745.29 65172.22 68519.96 54774.77 76246.96 65461.92 34127.21 35253.98
## [217] 44893.71 59621.02 20856.54 55353.41 67516.07 68737.75 76893.84 59886.58
## [225] 53441.69 41356.31 49942.66 74430.08 58633.63 72707.87 31092.93 74445.18
## [233] 49309.14 56735.14 40183.75 58348.41 72209.99 62060.11 67113.46 24030.06
## [241] 56180.93 62204.93 60372.64 65280.16 34309.24 59610.81 50278.89 43450.11
## [249] 25408.21 71136.49 63883.81 64902.47 66784.81 62784.85 63727.50 61608.23
## [257] 56782.18 64447.77 42042.95 67669.06 54875.95 73347.67 50199.77 50723.67
## [265] 63450.96 56694.12 70547.16 47391.95 62312.23 63100.13 73687.50 52686.47
## [273] 78119.50 57014.84 27086.40 58337.18 50216.01 53049.44 62927.96 32847.53
## [281] 32006.82 48913.07 69285.69 53700.57 52011.00 46339.25 67938.77 66348.95
## [289] 66873.90 72270.88 61610.05 76560.59 62667.51 75687.46 66744.65 67714.82
## [297] 69710.51 66269.49 60843.32 55041.60 73863.25 62378.05 63336.85 42191.61
## [305] 56194.56 61771.90 61383.79 63924.82 23975.35 70179.11 66524.80 41851.38
## [313] 61275.18 60638.38 47160.53 48537.18 53058.91 68614.98 44174.25 67050.16
## [321] 54520.14 54952.42 69476.42 54989.93 29398.61 42861.42 65883.39 65421.39
## [329] 60953.93 58476.57 66636.84 67430.96 57260.41 66359.32 57587.00 63060.55
## [337] 59998.50 74024.61 60550.66 57983.30 52736.33 46653.75 56986.73 55336.18
## [345] 42162.90 39699.13 56394.82 75044.35 53309.61 58996.12 56605.12 62475.99
## [353] 70492.60 43698.53 57737.51 31281.01 45800.48 42362.49 66691.23 56369.74
## [361] 59397.89 66025.11 68211.35 73608.99 61228.96 72325.91 44559.43 73207.15
## [369] 46722.07 45400.50 41417.27 60845.55 60812.77 64267.88 58151.87 52079.18
## [377] 26023.99 62318.38 56216.57 61806.31 51662.24 67080.94 51975.41 28019.09
## [385] 67744.56 66574.00 30487.48 74903.41 19991.72 66050.63 70449.04 64008.55
```

```
## [393] 70203.74 27262.51 49544.41 28357.27 66929.03 75524.78 66265.34 55993.68
## [401] 56379.30 31215.88 51015.11 46473.14 55479.62 68713.70 34191.23 51067.54
## [409] 46693.76 19345.36 66225.72 38609.20 37713.23 63764.28 41866.55 57846.68
## [417] 69428.73 60283.98 79332.33 53167.68 64564.07 60803.37 28387.42 58849.77
## [425] 65963.37 75180.20 61270.14 56759.48 46160.63 43870.51 50439.49 28028.74
## [433] 64238.71 65816.38 72684.44 38817.40 63976.44 37212.54 52691.79 65499.93
## [441] 63966.72 52400.88 49111.47 41232.89 52140.04 60641.09 74180.05 51869.87
## [449] 48852.58 59144.02 33951.63 58909.36 49850.52 28679.93 69869.66 48347.64
## [457] 45959.86 70005.51 51512.66 25598.75 49282.87 67240.25 42136.33 62589.84
## [465] 67384.31 25603.93 39616.00 28265.81 63879.72 70592.81 76408.19 55015.08
## [473] 51636.12 29359.20 71296.67 46422.76 52802.00 59243.46 35350.55 59677.64
## [481] 70225.60 65791.17 34191.13 51315.38 62790.96 66291.67 68030.18 43974.49
## [489] 49457.48 33987.27 28210.03 75535.14 49158.50 39809.69 65826.53 61172.07
## [497] 42898.21 68333.01 70232.95 63102.19 51847.26 63580.22 47575.44 39031.89
## [505] 70505.06 62161.26 61068.26 49090.51 62330.75 62053.37 61922.06 49525.37
## [513] 53412.32 56681.65 43299.63 47997.75 39131.53 46033.73 65856.74 54787.37
## [521] 69562.46 68447.17 62772.42 78092.95 63649.04 60637.62 27241.11 42760.22
## [529] 59457.52 42907.89 46132.18 46964.11 70377.23 70012.83 56457.01 67279.06
## [537] 54773.99 70783.94 70510.59 64021.55 72042.85 36037.33 67526.92 55121.65
## [545] 63497.62 60879.48 61467.33 70495.64 71222.40 64698.58 32252.38 55316.97
## [553] 47447.89 73474.82 53549.94 58576.12 63373.70 60283.47 37345.34 34886.01
## [561] 67511.86 77988.71 63001.03 61747.98 48467.68 55130.96 79484.80 67307.43
## [569] 27964.60 66431.87 63551.67 40135.06 49101.67 53188.69 49742.83 63394.41
## [577] 64433.99 73884.48 36424.94 28275.48 48098.86 68448.94 66429.84 41768.13
## [585] 57844.96 35684.82 62792.43 51171.23 58847.07 57739.03 64631.22 50337.93
## [593] 67781.31 68863.95 55901.12 64775.10 67686.16 57777.11 46868.53 40926.93
## [601] 22205.74 58920.44 63006.14 24316.61 68348.99 66263.37 63493.60 56984.09
## [609] 51691.55 49911.25 33502.57 65834.97 66176.97 51463.17 41059.64 61428.18
## [617] 51593.46 57518.73 52656.13 52178.98 46239.14 48918.55 65227.79 55002.05
## [625] 52261.73 59448.44 47314.45 55411.06 66504.16 47169.14 70889.68 55358.88
## [633] 56242.70 45522.44 46931.03 55499.69 75805.12 40345.49 33239.20 68033.54
## [641] 38427.66 53185.34 39723.97 43386.07 53922.43 71881.84 47139.21 68877.02
## [649] 65186.58 55424.24 46500.11 58820.16 28495.21 61840.26 37908.29 69805.70
## [657] 60315.19 67323.00 50055.33 43573.66 28186.65 66412.04 63965.16 58342.63
## [665] 33147.19 65899.68 64188.50 58966.22 44078.24 60968.62 65620.25 65496.78
## [673] 52462.04 70582.55 51816.27 23410.75 62729.40 48867.67 50971.73 67990.84
## [681] 43241.19 60082.66 65180.97 67301.39 70701.31 60997.84 60805.93 50711.68
## [689] 41335.84 76480.16 67132.46 52581.16 55195.61 48679.54 63109.74 44490.09
## [697] 57667.99 51824.01 66198.66 73174.19 56593.80 31072.44 66773.83 72553.94
## [705] 43708.88 48453.55 73413.87 58114.30 45465.25 50147.72 61004.51 53898.89
## [713] 59797.64 74623.27 58677.69 62109.80 60583.02 65576.05 73882.91 50468.36
## [721] 51409.45 60514.05 57195.96 52802.58 56570.06 51049.47 66629.61 70185.06
## [729] 43111.41 56435.60 53223.58 57179.91 41521.28 73538.09 63664.32 61757.12
## [737] 71727.51 72203.96 50671.60 47510.42 62466.10 59683.16 41097.17 39799.73
## [745] 76984.21 57877.15 59047.91 72154.68 65704.79 72948.76 73941.91 57887.64
## [753] 62463.70 42838.29 43778.88 71157.05 74159.69 50333.72 33293.78 38641.20
## [761] 49822.78 63891.29 43881.73 48761.14 69758.31 52530.10 58363.12 60575.99
## [769] 48206.04 31523.09 66187.58 69438.04 68016.90 78520.99 31998.72 56909.30
## [777] 61161.29 52340.10 47338.94 50950.24 77143.61 57032.36 48554.45 39552.49
## [785] 36884.23 68783.45 51119.93 44304.13 69718.19 63429.18 65756.36 77871.75
## [793] 47258.59 55984.89 44275.13 25767.16 37605.11 25739.09 60188.38 67682.32
## [801] 44307.18 25371.52 23942.61 50666.50 50356.06 63936.50 69874.18 50038.65
## [809] 67866.95 54645.20 46780.09 67432.49 73392.28 47682.28 56735.83 51013.37
## [817] 69481.85 67033.34 68717.00 59340.99 47968.32 48758.92 61230.03 54755.71
```

```
## [825] 54324.73 52177.40 51163.14 66861.67 63107.88 49206.40 55942.04 33601.84
## [833] 48867.36 56683.32 38260.89 54106.21 71055.22 46403.18 61690.93 26130.93
## [841] 58638.75 47357.39 50086.17 51772.58 47638.30 38987.42 51363.16 35764.49
## [849] 62939.50 58776.67 59106.12 50457.01 54251.78 51920.49 70324.80 52416.18
## [857] 66217.31 60938.73 40243.82 60151.77 45945.88 63430.33 65882.81 64410.80
## [865] 55677.12 75560.65 61067.58 72330.57 32549.95 51257.26 77220.42 52520.75
## [873] 59422.47 22456.04 58443.99 50820.74 67575.12 66522.79 34903.67 43073.78
## [881] 57594.70 66027.31 53012.94 61117.50 52563.22 65773.49 50506.44 66262.59
## [889] 35521.88 62430.55 49597.08 42078.89 46197.59 49957.00 24078.93 53647.81
## [897] 61039.13 46974.15 53042.51 48826.14 58287.86 21773.22 52252.91 27073.27
## [905] 50628.31 36913.51 61009.10 53041.77 40182.84 59419.78 58235.21 68324.48
## [913] 69646.35 54045.39 57806.03 53336.76 50491.45 71455.62 43241.88 58953.01
## [921] 36834.04 66345.10 38645.40 60803.00 33553.90 63071.34 46737.34 55368.67
## [929] 68305.91 39211.49 65956.71 40159.20 40478.83 40468.53 66980.27 34942.26
## [937] 48335.20 42251.59 57330.43 75769.82 51812.71 75265.96 69868.48 72802.42
## [945] 39193.45 56129.89 58996.56 41547.62 59240.24 56725.47 55764.43 64235.51
## [953] 39939.39 63319.99 54725.87 69775.75 57545.56 47051.02 51600.47 68357.96
## [961] 35349.26 69784.85 50760.23 34418.09 20592.99 63528.80 44217.68 47929.83
## [969] 46024.29 51900.03 72188.90 56974.51 25682.65 41884.64 72196.29 54429.17
## [977] 58037.66 64011.26 59967.19 43155.19 51501.38 55187.85 33813.08 36497.22
## [985] 66193.81 66200.96 63126.96 71384.57 67782.17 42415.72 41920.79 29875.80
```

d). Correcting the Data types

We shall be changing the "chr" to factor datatype

```
advertising[sapply(advertising, is.character)] <- lapply(advertising[sapply(advertising, is.character)]
                                       as.factor)
str(advertising)
```

```
## 'data.frame':    1000 obs. of  10 variables:
##  $ Daily.Time.Spent.on.Site: num  69 80.2 69.5 74.2 68.4 ...
##  $ Age                     : int  35 31 26 29 35 23 33 48 30 20 ...
##  $ Area.Income             : num  61834 68442 59786 54806 73890 ...
##  $ Daily.Internet.Usage    : num  256 194 236 246 226 ...
##  $ Ad.Topic.Line           : Factor w/ 1000 levels "Adaptive 24hour Graphic Interface",..: 92 465 567
##  $ City                    : Factor w/ 969 levels "Adamsbury","Adamside",..: 962 904 112 940 806 283
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Country                 : Factor w/ 237 levels "Afghanistan",..: 216 148 185 104 97 159 146 13 83
##  $ Timestamp               : Factor w/ 1000 levels "2016-01-01 02:52:10",..: 440 475 368 57 768 690 1
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

5. Exploratory Data Analysis

a) Univariate Analysis

We shall use the Graphical method to do the Univeriate Analysis

i). Measures of Central Tendency

```
# Mean Age

mean(advertising$Age)
```

```
## [1] 36.009
```

We can see the average age is aroung 36 years

```
# Median Age

median(advertising$Age)
```

```
## [1] 35
```

The median age is 35 years

```
# Average Daily spent on Site

mean(advertising$Daily.Time.Spent.on.Site)
```

```
## [1] 65.0002
```

```
mean(advertising$Daily.Internet.Usage)
```

```
## [1] 180.0001
```

```
# Mode of Area Income

mode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

mode(advertising$Area.Income)
```

```
## [1] 61833.9
```

The modal Area Income is 61833

ii). Measures of Dispersion

```
# Checking the range of the Numeric columns of the Area Income

range(advertising$Area.Income)
```

```
## [1] 13996.5 79484.8
```

```
# Checking the maximum age

max(advertising$Age)
```

```
## [1] 61
```

```
# Checking the minimum age

min(advertising$Age)
```

```
## [1] 19
```

```
# Checking the Variance of the Area Income

var(advertising$Area.Income)
```

```
## [1] 179952406
```

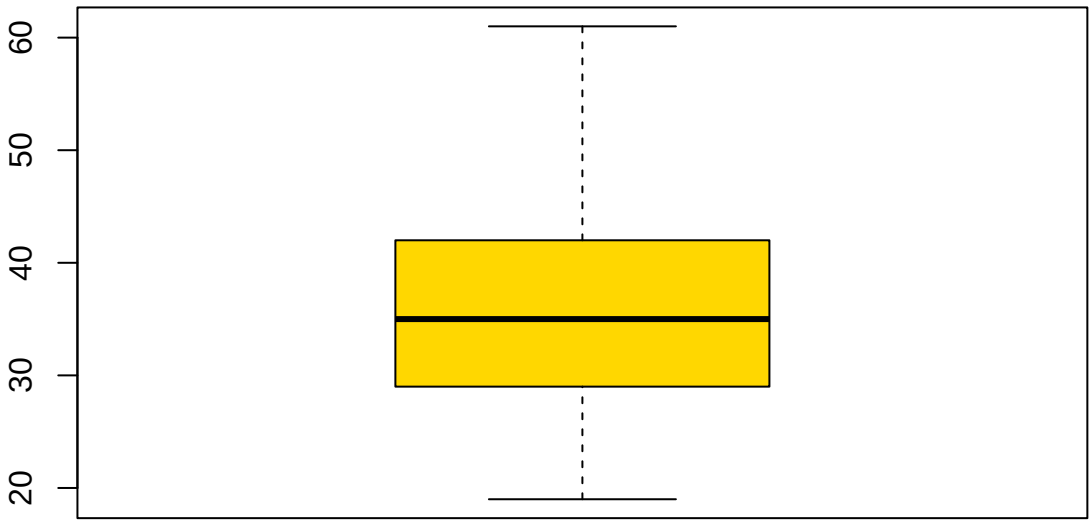
```
# Checking the Standard Deviation of the Area Income

sd(advertising$Area.Income)
```

```
## [1] 13414.63
```

iii) Univariate Graphical

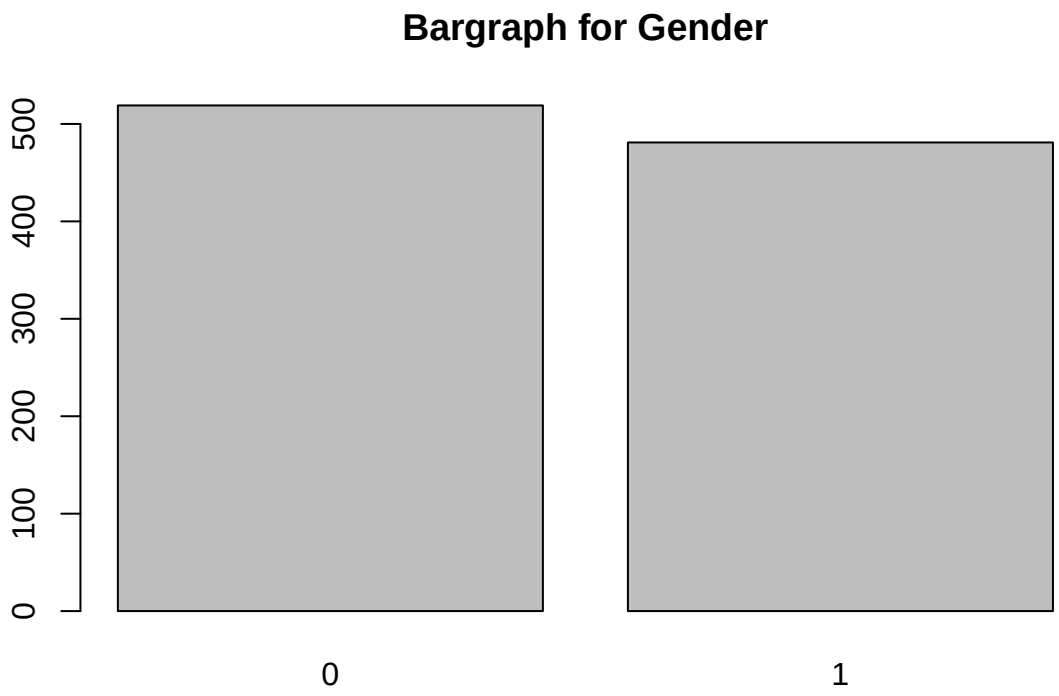
We shall be using the box plot to display our age data

```
# Boxplot for age

boxplot(advertising$Age,col="gold")
```



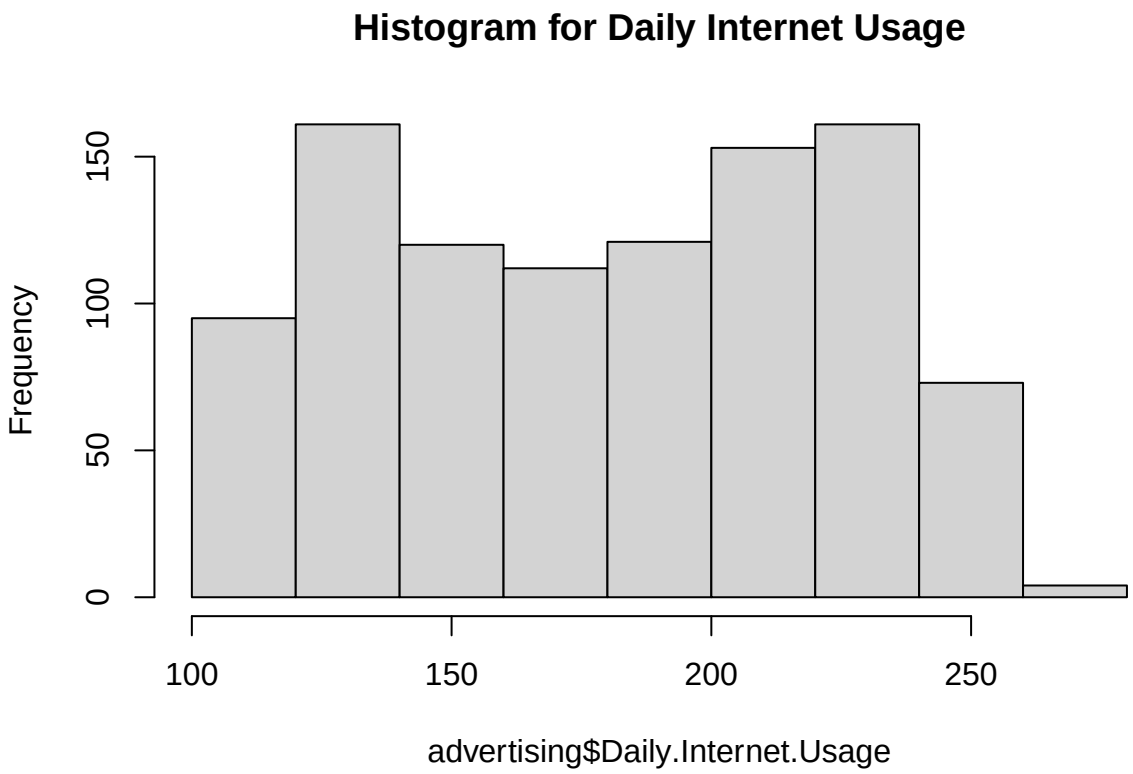
We shall use bar graph to show the gender(Male) feature of our data

```
gender_frequency <- table(advertising$Male)

barplot(gender_frequency, main = "Bargraph for Gender")
```



We have slightly more females than males as clearly shown on the bargraph. The 0 implies Female while 1 implies male

```
# Histogram to show the Daily Internet Usage

hist(advertising$Daily.Internet.Usage, main = "Histogram for Daily Internet Usage")
```



b). Bivariate Analysis

i). Covariance

```
cov(numeric_col)
```

```
##                           Daily.Time.Spent.on.Site           Age   Area.Income
## Daily.Time.Spent.on.Site              251.3370949 -4.617415e+01  6.613081e+04
## Age                                   -46.1741459  7.718611e+01 -2.152093e+04
## Area.Income                         66130.8109082 -2.152093e+04  1.799524e+08
## Daily.Internet.Usage                  360.9918827 -1.416348e+02  1.987625e+05
## Male                                   -0.1501864 -9.242142e-02  8.867509e+00
## Clicked.on.Ad                          -5.9331431  2.164665e+00 -3.195989e+03
##                           Daily.Internet.Usage        Male Clicked.on.Ad
## Daily.Time.Spent.on.Site         3.609919e+02 -0.15018639 -5.933143e+00
## Age                             -1.416348e+02 -0.09242142  2.164665e+00
## Area.Income                      1.987625e+05  8.86750903 -3.195989e+03
## Daily.Internet.Usage             1.927415e+03  0.61476667 -1.727409e+01
## Male                             6.147667e-01  0.24988889 -9.509510e-03
## Clicked.on.Ad                   -1.727409e+01 -0.00950951  2.502503e-01
```

ii). Correlation

```
cor(numeric_col)
```

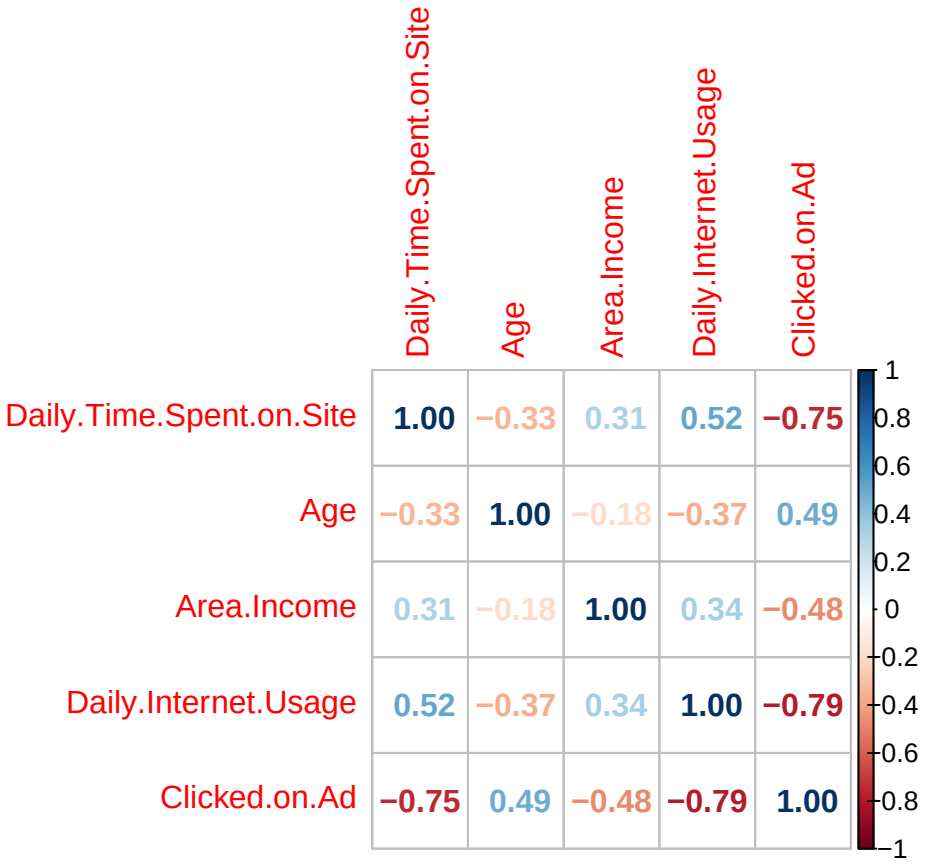
```
##                           Daily.Time.Spent.on.Site         Age  Area.Income
## Daily.Time.Spent.on.Site               1.00000000 -0.33151334  0.310954413
## Age                                   -0.33151334  1.00000000 -0.182604955
## Area.Income                            0.31095441 -0.18260496  1.000000000
## Daily.Internet.Usage                   0.51865848 -0.36720856  0.337495533
## Male                                  -0.01895085 -0.02104406  0.001322359
## Clicked.on.Ad                         -0.74811656  0.49253127 -0.476254628
##                           Daily.Internet.Usage         Male Clicked.on.Ad
## Daily.Time.Spent.on.Site           0.51865848 -0.018950855   -0.74811656
## Age                               -0.36720856 -0.021044064    0.49253127
## Area.Income                        0.33749553  0.001322359   -0.47625463
## Daily.Internet.Usage               1.00000000  0.028012326   -0.78653918
## Male                               0.02801233  1.000000000   -0.03802747
## Clicked.on.Ad                     -0.78653918 -0.038027466    1.00000000
```

We shall now visualize the correlation matrix

```
# Loading the corrplot

library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
# Dropping the Male from numeric column

num_col <- subset(numeric_col, select = -c(Male))


corr_matrix <- cor(num_col)

corrplot(corr_matrix, method='number')
```

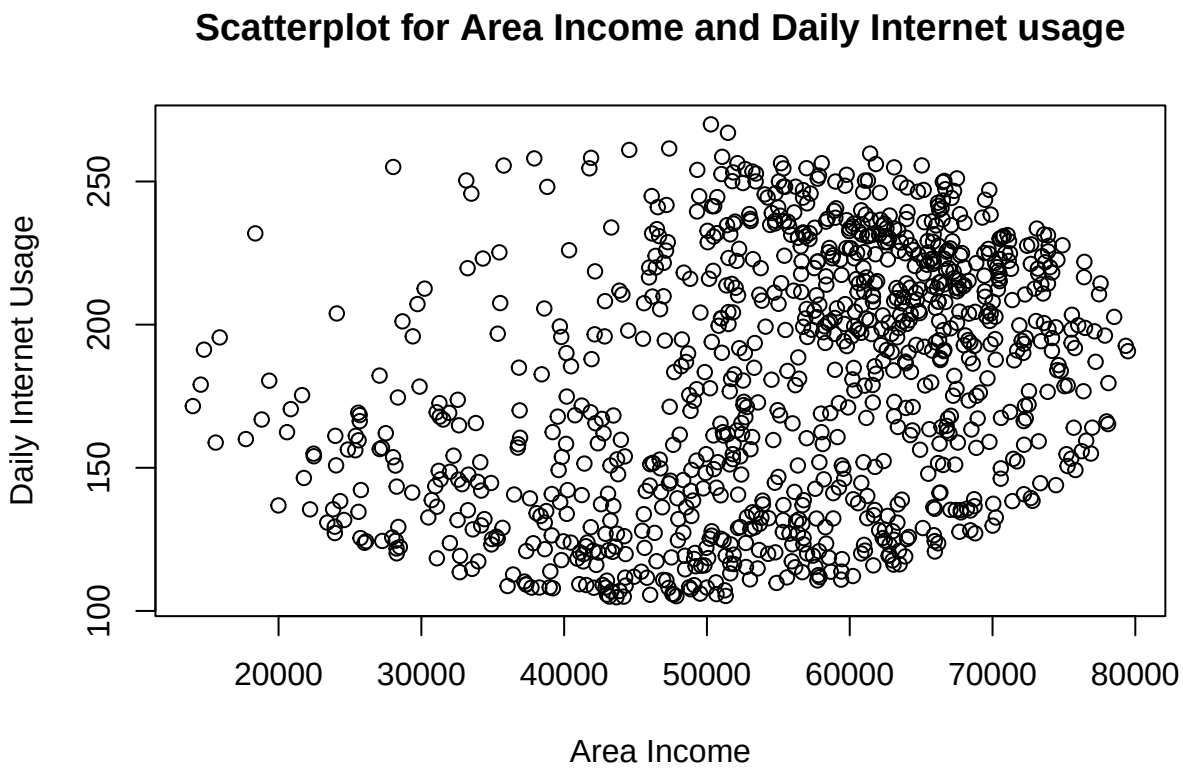


From the above Correlation Matrix, we can see there is strong inverse correction between Daily Internet Usage and Clicked on Ad.

iii). Scatterplot

We shall use the scatterplot to show the relationship between Area Income and Daily Internet Usage

```
area_income <- advertising$Area.Income
daily_usage <- advertising$Daily.Internet.Usage

plot(area_income,daily_usage,xlab="Area Income", ylab="Daily Internet Usage",main="Scatterplot for Area
```



7. Modelling

Since our a research question is a classification problem in nature, we shall be using Logistic model as our baseline model for this study.

i). Logistic Regression Model

This is a statistical model that models the probability of one event (out of two alternatives) taking place by having the log-odds (the logarithm of the odds) for the event be a linear combination of one or more independent variables ("predictors").

```
# Loading required packages
library(e1071)
library(caTools)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```r
# Splitting the data in train and test datasets
# We shall have 80% of the data in our train set and 20% in our test set

# Splitting the data into training and testing sets
set.seed(100)

# Selecting only columns that are relevant to modeling
cols = c('Daily.Time.Spent.on.Site', 'Age', 'Area.Income', 'Daily.Internet.Usage', 'Male', 'Clicked.on.A
advertising = select(advertising, all_of(cols))

train_rows = createDataPartition(advertising$Clicked.on.Ad, p=0.8, list=FALSE)

# Creating the training  dataset
train = advertising[train_rows,]

# Creating the test dataset
test = advertising[-train_rows,]

# Creating the  X and y variables
X = train
y = train$Clicked.on.Ad
```

```r
# Training

mymodel = glm(Clicked.on.Ad ~ ., data = train, family = 'binomial')

summary(mymodel)
```

```
##
## Call:
## glm(formula = Clicked.on.Ad ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.06636  -0.15109  -0.03433   0.01934   3.14389
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              25.8859056  2.9518193   8.769  < 2e-16 ***
## Daily.Time.Spent.on.Site -0.1871771  0.0240198  -7.793 6.56e-15 ***
## Age                       0.1763327  0.0292099   6.037 1.57e-09 ***
## Area.Income              -0.0001252  0.0000205  -6.108 1.01e-09 ***
## Daily.Internet.Usage     -0.0616178  0.0072699  -8.476  < 2e-16 ***
## Male                     -0.4560563  0.4524080  -1.008    0.313
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1109.0  on 799  degrees of freedom
## Residual deviance:  149.9  on 794  degrees of freedom
## AIC: 161.9
##
```

```
## Number of Fisher Scoring iterations: 8
```

```
# Making predictions using the training set
y_pred = predict(mymodel)
```

```
# Running the test through the model

res <- predict(mymodel,test,type = "response" )
head(res)
```

```
##           5           9          11          12          13          16
## 0.020027929 0.004066708 0.999995787 0.003525804 0.999411207 0.917591078
```

```
res <- predict(mymodel,train,type = "response" )
head(res)
```

```
##           1           2           3           4           6           7
## 0.012499167 0.009664894 0.010047351 0.004732640 0.039296829 0.010793026
```

```
# Validating the Model

conf_matrix <- table(Actual_value=train$Clicked.on.Ad, Predicted_value = res > 0.5)
conf_matrix
```

```
##              Predicted_value
## Actual_value FALSE TRUE
##            0   392    8
##            1    16  384
```

```
# Accuracy

(conf_matrix[[1,1]] + conf_matrix[[2,2]]) / sum(conf_matrix)
```

```
## [1] 0.97
```

We can see from the above, the model has an accuracy of 97%, which is not bad but we shall try using another model.

ii). Decision Trees Model

```
set.seed(100)

# Selecting only columns that are relevant to modeling
cols = c('Daily.Time.Spent.on.Site', 'Age', 'Area.Income', 'Daily.Internet.Usage', 'Male', 'Clicked.on.
advertising = select(advertising, all_of(cols))

train_rows = createDataPartition(advertising$Clicked.on.Ad, p=0.8, list=FALSE)

# Creating the training  dataset
```

```
train = advertising[train_rows,]

# Creating the test dataset
test = advertising[-train_rows,]


# Creating the  X and y variables
X = train
y = train$Clicked.on.Ad
```

```
# Training the model

require(tree)
```

```
## Loading required package: tree
```

```
model_d <- tree(Clicked.on.Ad ~ .,
                data = train,
                method = "ranger")
model_d
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 800 200.0000 0.50000
##    2) Daily.Internet.Usage < 177.505 381  25.9400 0.92650
##      4) Daily.Time.Spent.on.Site < 75.675 341   5.8940 0.98240 *
##      5) Daily.Time.Spent.on.Site > 75.675 40   9.9000 0.45000
##       10) Daily.Internet.Usage < 152.975 14   1.7140 0.85710 *
##       11) Daily.Internet.Usage > 152.975 26   4.6150 0.23080 *
##    3) Daily.Internet.Usage > 177.505 419  41.7300 0.11220
##      6) Daily.Time.Spent.on.Site < 54.11 28   0.9643 0.96430 *
##      7) Daily.Time.Spent.on.Site > 54.11 391  18.9800 0.05115
##       14) Area.Income < 33370.9 5   0.8000 0.80000 *
##       15) Area.Income > 33370.9 386  15.3400 0.04145
##         30) Age < 49.5 379  11.6200 0.03166 *
##         31) Age > 49.5 7   1.7140 0.57140 *
```

```
# Predicting the model

y_pred <- predict(model_d)
```

```
# Validating the Model

conf_matrix <- table(Actual_value=train$Clicked.on.Ad, Predicted_value = y_pred>0.5)
conf_matrix
```

```
##              Predicted_value
## Actual_value FALSE TRUE
##            0   387   13
##            1    18  382
```

```r
# Accuracy of the model

(conf_matrix[[1,1]] + conf_matrix[[2,2]]) / sum(conf_matrix)
```

```
## [1] 0.96125
```

The Decision tree has an accuracy of 96%, which is abit lower than the Logistic regression, but still has met our threshold of the 90%.

7. Conclusion

The Logistic regression outperforms the Support Vector Machine in the data analysis and modeling described above. Additional data may be required to improve the model's predicted accuracy/power.

8. Recommedation

The shortest amount of time spent on the site without clicking on an ad was 60 minutes. This is more than the average amount of time spent on the site by those who click on the ad. In order to ensure that the number of clicks on the ad increases, I would propose increasing the number of advertisements throughout the months of February, April, May, June, September, and December. These are the months that have no clicks on the ad.

9. Follow up questions

a) Did we have the right data?

Yes, the dataset available for this analysis was relevant to the research problem.

b) Do we need other data to answer the research question?

No, the dataset provided had relevant information for the research question.