PROJECT: PREDICTIVE MODELLING TO DETERMINE SUCCESS OF STARTUP IN RAISING FUNDS THROUGH ICO CAMPAIGNS

Programming Language: R

Author: Gaurav Kumar

1. INTRODUCTION:
   1.1. Business Understanding:
      1.1.1. What is an ICO?

Initial Coin Offering (ICO) is like IPO (Initial Public Offering). It is a way for a venture to raise money from public (Fisch, 2019). But there are important differences. In an IPO, an investor gets shares of the company in return whereas in an ICO, an investor gets tokens/coins in return which are based on Distributed Ledger Technology (DLT) (Fisch, 2019). The most popular technology in DLT is blockchain technology. In ICO, the tokens issued have no real world value and they can be used in the ecosystem of ventures' itself which they are building (Fisch, 2019). It depends on venture what utility they want to attach to tokens issued for example if they issue security tokens then it can work like shares issued in an IPO (Fisch, 2019; Momtaz, 2020).

      1.1.2. Why ICOs are popular?

ICOs have become popular among ventures to raise funds because it involves less compliance and no intermediary. All this leads to low cost. Also, ICOs can be traded in secondary markets (Fisch, 2019).

   1.2. Background of task:

The purpose of this report is to build a supervised machine learning model which can predict the success of venture in raising money through ICO. The success is defined as the venture raising the target amount in the set time frame. If they are not able to raise money in the specified time frame, then ICO campaign will be deemed as failure and venture will receive nothing.

The report will deploy 5 ML algorithms viz Decision Tree, Adaboost, Random Forest, Support Vector Machine (SVM) and Artificial Neural Network (ANN) and will attempt to find out which one has the best predictability power.

2. UNDERSTANDING DATA:
   2.1. General Overview:

The dataset has 2767 observations and 16 features. Breakup of features by datatypes is as follows (Appendix -1):

Numeric	: 4 (rating,priceUSD,coinNum,distributedPercentage)
Integer	: 6 (ID, hasVideo,teamSize,hasGithub,hasReddit,minInvestment)
Character	: 6 (success,brandSlogan,countryRegion,startDate,endDate,platform)
Total	: 16

   2.2. Numerical features (Numeric datatype and Integer datatype combined):

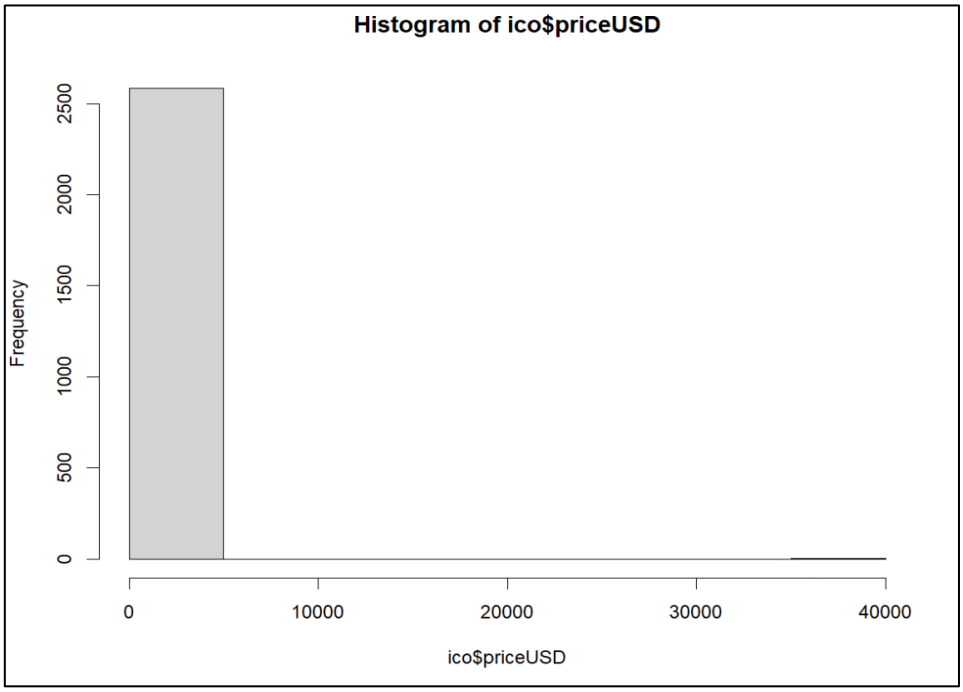
The dataset has 10 numeric features which have been analysed using str() function and summary() function (Appendix-2):

Features which require careful consideration are "priceUSD", "teamSize" and "coinNum".

priceUSD:

The price of each blockchain coin/token issued by venture in US dollars. It has 180 missing values and 152 observations where value is zero.

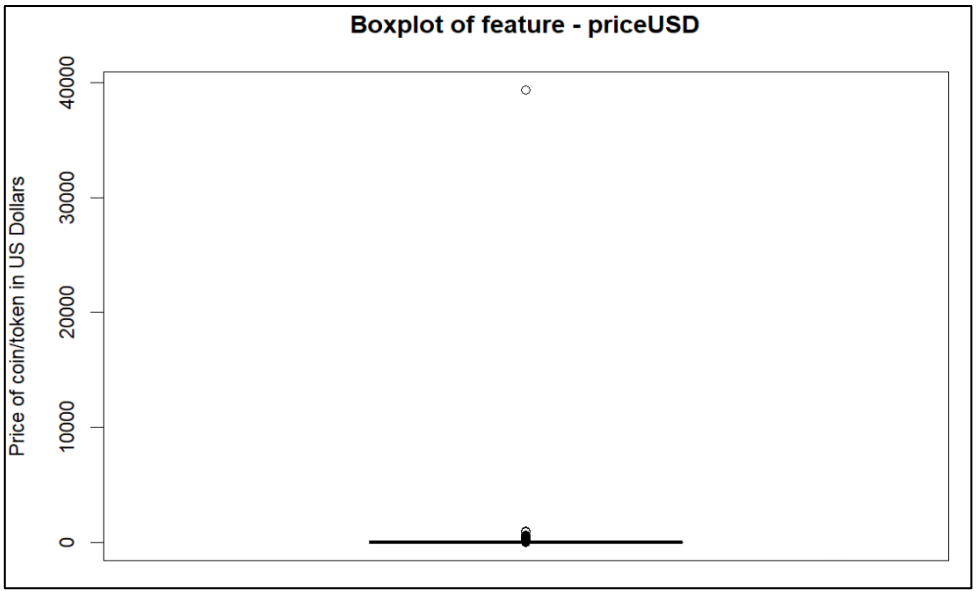
Histogram of priceUSD: (Appendix – 5)



The output shows that there is only one observation which falls in bin 35000 to 40000. Rest all observations lie in bin 0 to 5000. (Appendix - 5).

It is known from summary data that the max value of observation is 39384. It means this is the only observation which is considerably different from rest of the observations.

Further investigation of outlier in priceUSD- Using box and whisker plot:



The extreme of upper whisker is 1.19 as per output of box plot (Appendix – 6). Box plot shows all observations where price of coin/token is more than USD 1.19 as outliers. So, as per boxplot there are 240 observations which are outliers (Appendix – 6, see "$out" of the output).

Further investigation of outlier in priceUSD using descriptive statistics:

Standard Deviation (SD):

- ➔ SD for priceUSD is 775.2871 (Appendix – 7)
- ➔ 3 SD = 3 * 775.2871 = 2325.861

Values which lie more than 3SD are extremely rare (Lantz, 2019, p.57). There is only one observation where priceUSD is more than 3SD which is priceUSD = 39,384.

So, from preceding analysis using histogram, box plot and descriptive statistics, it can be safely assumed that observation where priceUSD = 39384 is extremely rare. Thus, this observation will be removed from dataset otherwise it will distort the model. (Task – 1)

Missing values (Task -2) and zero values (Task -3) will be imputed in data pre-processing section.

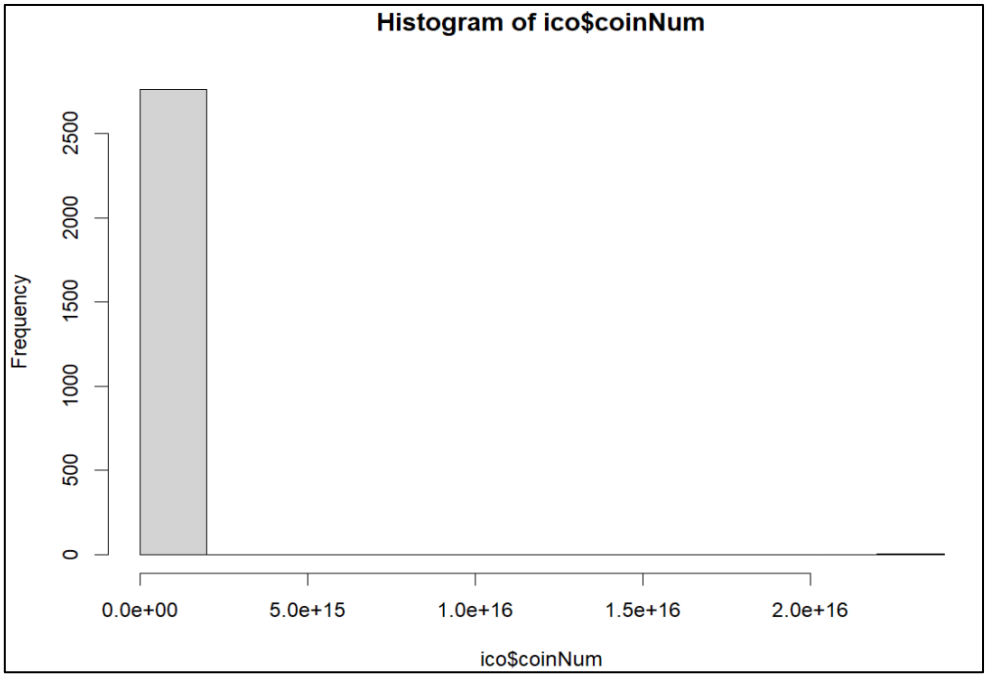
teamSize:

It specifies the number of members in the venture. There are 154 observations where value is missing which shall be imputed in data pre-processing (Task – 4).

coinNum:

This shows the number of coins to be issued which is freely decided by venture.

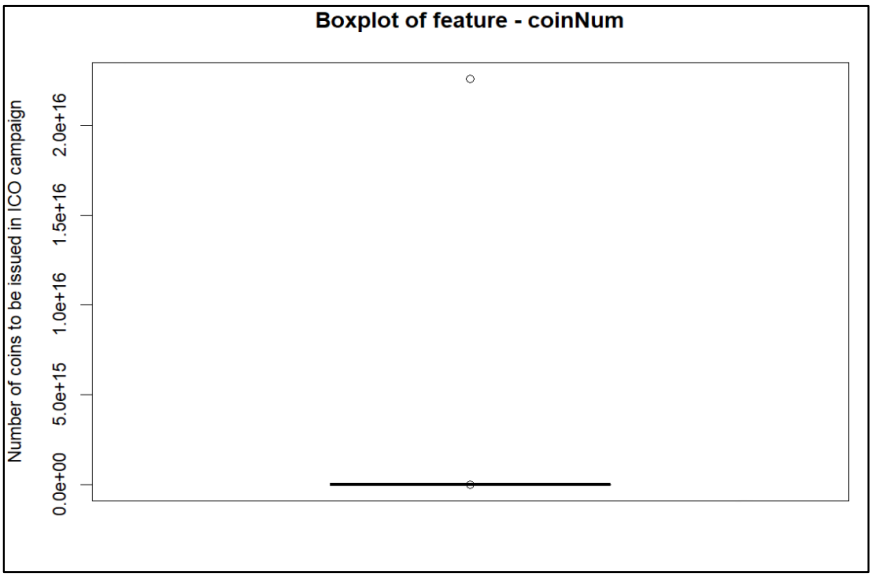
Summary details show a very large maximum value (Appendix – 2, 8) evident in comparison with mean and median. Further investigation using histogram:



The output of hist() function shows there is only one value which falls in the highest bin which is 2.2e+16 to 2.4e+16 (Appendix – 9).

It is known from summary details (Appendix – 2) that the particular observation is the one where coinNum is 2.262e+16 (which is 22619078416760300).

Further investigation of outlier in coinNum using boxplot.



The extreme of upper whisker is 1.42e+09. The analysis of output of boxplot shows that there are 395 observations which fall outside upper extreme of whisker (Appendix – 10, see "$out" of the output, Appendix - 11).

Further investigation of outlier using descriptive statistics:

Standard Deviation(SD):

SD = 4.300018e+14 (Appendix – 12)

3*SD = 1.290005e+15

Again, values more than 3SD are extremely rare (Lantz, 2019, p.57). There is only one observation where value of coinNum is more than 3SD which is coinNum = 2.261908e+16.

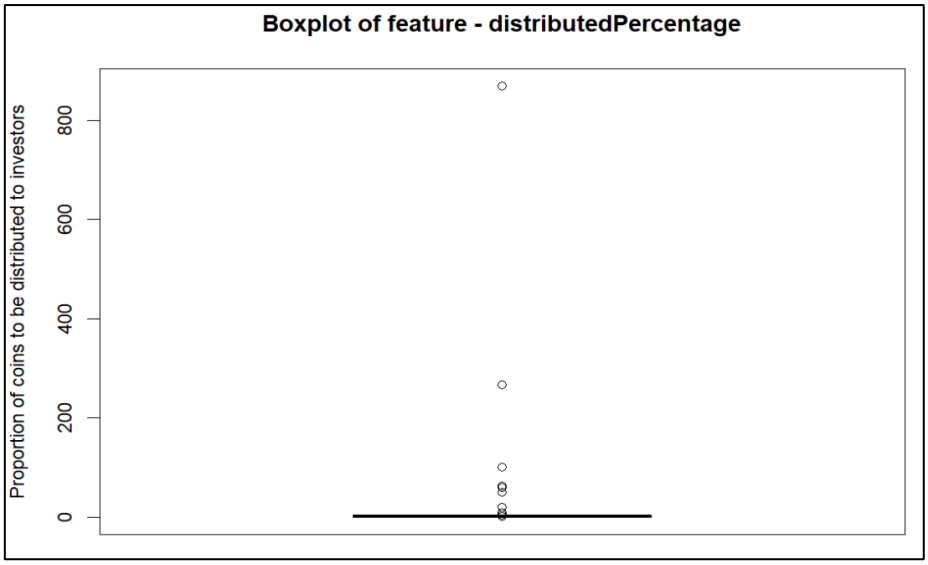
Thus, from preceding analysis using histogram, box plot and descriptive statistics it can be safely assumed that observation with coinNum = 22619078416760300 is an outlier which will be removed to avoid distortion of prediction results (Task – 5).

distributedPercentage:

This figure specifies the percentage of coins distributed to investors out of all coins to be minted.

The summary details show a very large maximum value (Appendix – 2) evident in comparison with mean and median. This requires further investigation.

Further investigation using boxplot: (Appendix – 14)



The output suggests that there are 10 observations which lie outside of extreme of upper whisker which is 1(Appendix – 13). Since, this observation represents percentage, it cannot be greater than 1. So, the 10 observations which are greater than 1 are incorrect values.

Also, the value cannot be zero. This is absurd because no venture would launch ICO campaign offering no coin. There is 1 observation where value is zero. This is again an incorrect value.

So, these 11 values shall be deleted in data pre-processing (Task – 6).

2.3. Categorical features:

2.3.1. Success:

a. This is class label. It tells which ICO campaign has been successful in raising the target amount within campaign period (Appendix – 15):

Success	Number	Percentage
Y	1028	37.15%
N	1739	62.85%
Total	2767	100%

2.3.2. brandSlogan:

a. It carries the text of slogan by venture for ICO campaign.
b. Initially, it was thought that length of slogan could be used as one feature. But studies have shown that length of slogan has no bearing on effectiveness of slogan, so this feature is removed (Kohli, 2013).

2.3.3. countryRegion:

a. It specifies country where venture is located.
b. Analysis of countryRegion shows: (Appendix – 15)
    a. There are 71 observations where value is ‘blank’. Since we do not know the country, these blanks will be replaced by “Unknown” (Task – 7).

b. Visual examination of output of table() function in Appendix – 15 reveals that few countries appear twice due to different case. They are as follows:

Name of country	Count	Name of same country in different case	Count
India	38	india	1
Mexico	7	México	1
Singapore	312	SINGAPORE	1
USA	296	usa	1

These values will be corrected in data pre-processing. (Task – 8)

2.3.4. startDate and endDate:

It shows the date of start and end date of campaign respectively. But as per str() details (Appendix – 1), the values are in character datatype. It needs to be converted to date format.

2.3.5. platform:

It specifies the platform on which the project is based. Output od table() function shows that Ethereium as the most popular platform (Appendix- 16). There are many values which are similar but written differently such as different case, space in suffix, space in prefix which needs to be corrected (Task – 9).

List of data pre-processing tasks from above analysis:

Task 1: Removing the outlier in priceUSD where value = 39,384

Task 2: Imputing missing values in priceUSD

Task 3: Imputing the values where priceUSD is zero

Task 4: Handling missing values in teamSize

Task 5: Removing the outlier in coinNum

Task 6: Handling incorrect values in distributedPercentage

Task 7: Replacing blank value in countryRegion with “unknown”

Task 8: Correcting the names of countries in countryRegion

Task 9: Bringing uniformity in names of platform

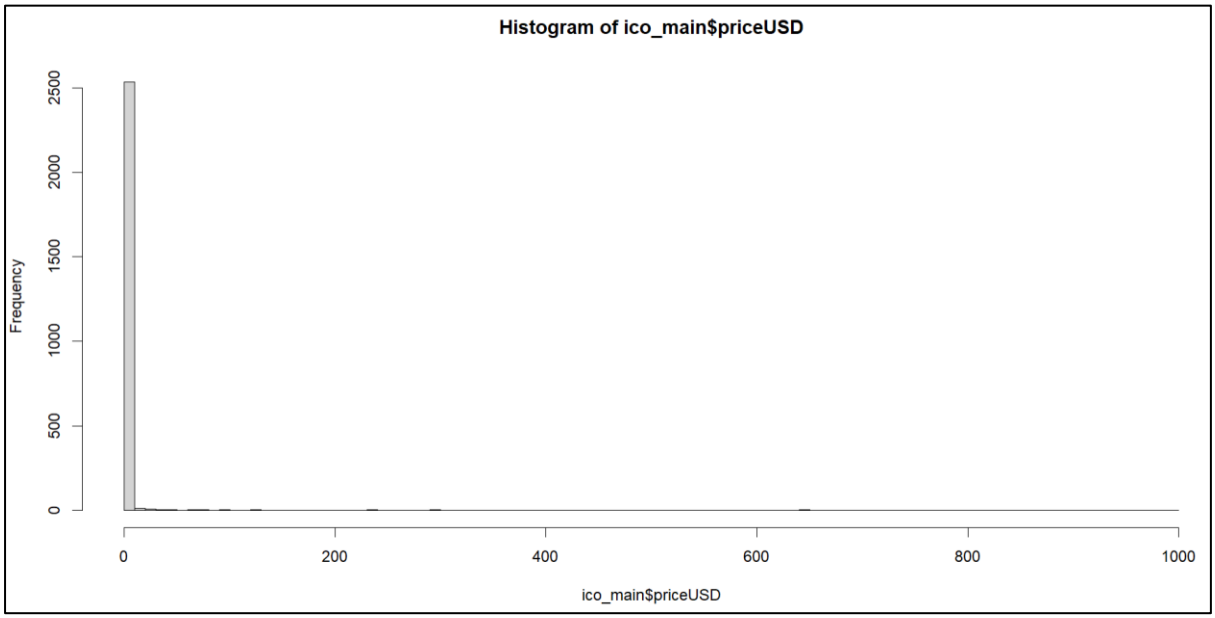
Task 10: Handling the value “Komodo” in feature platform requiring special attention

3. DATA PREPARATION:

In this section, the 10 tasks specified in previous section would be carried out. Before starting, a copy of dataset would be created so that if any manipulation go wrong, original data can be fetched from the copy of dataset.

Task 1: Removing the outlier in priceUSD where value = 39,384 (Appendix – 17).

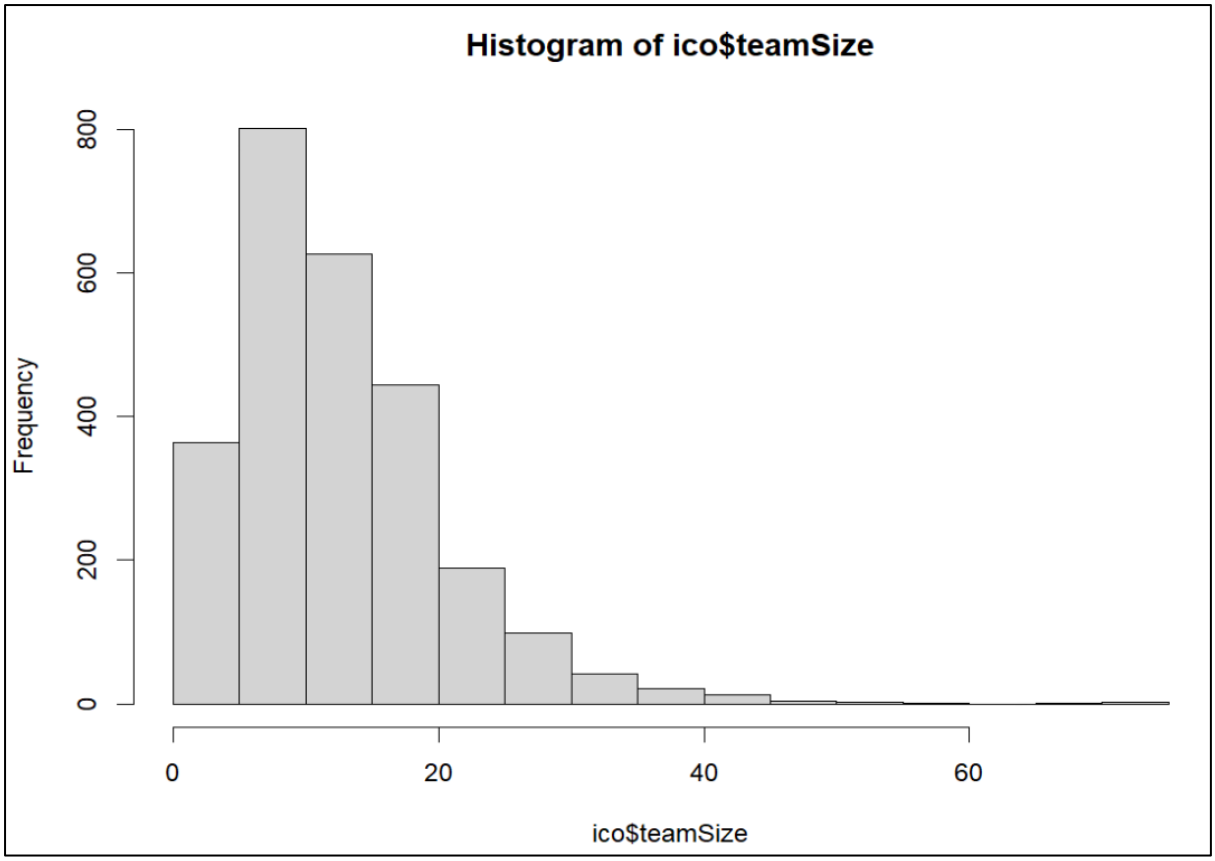
Task 2: Imputing missing values in priceUSD: As discussed previously, there are 180 missing :values. Now, histogram of priceUSD after exclusion of outlier in Task 1:



The output of hist() function suggests that 97.95% (2533 out of 2586) observations (excluding outlier and missing observations that is 2767 – 180 -1 = 2586) lie in the first bin 0 to 10. (Appendix – 18). The distribution is **right-skewed**. Mean is higher than median. Therefore, to maintain the distribution, the 180 missing values will be imputed with median and not mean. Because median is less sensitive to extreme values (Kumar, 2023). (Appendix – 19)

Task 3: Imputing the values where priceUSD is zero: As discussed previously, there are 152 observations where priceUSD is zero. The price of coin/token cannot be zero otherwise how venture will raise money. This indicates incorrect data. Again, based on histogram in preceding paragraph, the zero values will be imputed with median because distribution is right-skewed and median is not sensitive to extreme values. (Appendix – 20)

Task 4: Handling missing values in teamSize: As discussed previously, there are 154 missing values. It is clear from histogram below that the distribution is right-skewed. Thus, missing values will be imputed with median because it is not sensitive to outliers and maintains the distribution. (Appendix – 21)



Task 5: Removing the outlier in coinNum (Appendix – 22)

Task 6: Handling incorrect values in feature "distributedPercentage" (Appendix – 23)

Task 7: Replacing blank value in countryRegion with "unknown": (Appendix – 24)

Task 8: Correcting the names of countries in countryRegion: The values are replaced with name of country having higher count as discussed previously in data understanding section (Appendix – 25)

Task 9: Bringing uniformity in names of platform: There are many discrepancies in feature 'platform'. They mainly pertain to space before and after the name, use of different case and use of alternate names of platform.

Following approach is adopted to bring uniformity in names of platform:

a. Removing spaces in prefix & suffix of name of platform and converting all names to lower case. (Appendix – 27)
b. Changing the names of similar platform to one: (Appendix – 28)
    a. Bitcoin and BTC refer to same platform (bitcoin.org, 2023)
    b. Ethereum and ETH all refer to same platform (ethereum.org, 2023)
    c. Stellar and Stellar Protocol refer to same platform (stellar.org, 2023)
c. Replacing blank value with "unknown". (Appendix – 28)

Task 10: Handling the value “Komodo” in feature platform: This value was encoded in different Unicode which is U+200B which represents zero-width space. Thus, it was not detected as space because it is zero-width. So, this value was detected while running a model and was replaced with “Komodo”. To handle this, “u200b” was replaced with Null value using gsub function which is used for pattern matching and replacement.

4. FEATURE ENGINEERING:

A new feature will be created named “Duration_of_campaign” which will contain the number of days between startDate and endDate. (Appendix – 29)

5. FEATURE SELECTION: (Appendix – 30)

Feature “ID”, “startDate”, “endDate” and “brandSlogan” are deleted. ID has no prediction power as it is just a unique number to identify each ICO campaign (Lantz, 2019, p.78). startDate and endDate are removed as corresponding new feature containing duration has been added. As discussed in section 2.3.2, “brandSlogan” feature is removed.

6. MODELLING:

Five models have been used which are briefly explained below:

Decision Tree (DT):

DT classification algorithm attempts to divide dataset in tree form based on homogeneity. DT will keep on partitioning the dataset until one of the three conditions is met a) all observations in the leaf node are homogenous b) no further feature to split c) a defined condition is met e.g. a limit on number of branches. The biggest advantage of DT is easy interpretability. (Appendix – 31)

As per results of model, the first rule is based on rating <= 3.9 and country = USA. This is evident from data as USA has highest number of ventures bringing ICO campaigns.

Adaboost:

It is a technique to combine various weak learners (base algorithm) to produce highly accurate classification results. It works by calling the base algorithm again and again and in each round, it increases weights of instances which were incorrectly classified. So, in next round base algorithm will be forced to classify instances with increased weights correctly. Final prediction is done by weighted vote of outcomes of each round. (Schapire and Freund, 2012, p.5). (Appendix - 32)

Here, the first split is rating <=3. It is able to split 1227 observations which is 49.5% of total observations.

Random Forest:

It is an ensemble method that combines results of number of small decision trees into a single output. It is based on bagging and random feature selection. The biggest advantage of random forest is that it does not overfit because it is average of output of

multiple trees or based on voting in case of classification (Breiman, 2001; IBMCloudEducation, 2023).

Since random forest cannot process features having more than 53 categories. One hot encoding has been used to create dummy features of character features which are only 2 in our dataframe ie countryRegion and platform (Appendix – 33).

SVM (Support Vector Machines):

SVM works by finding a dividing plane called hyperplane to partition data. The main goal of SVM is to find out support vectors which help in arriving at the hyperplane which is called maximum margin hyperplane (MMH). MMH is a boundary line which divides data. The success of SVM lies in the fact that it can work with data having many features and has ability to find hyperplane which are non-linear. (Appendix – 34).

ANN (Artificial Neural Network):

ANN is inspired from working of a human brain neuron. Multiple inputs are fed into node to give an output. In ANN, weighted inputs are fed to node which deploys function to give output. The main attraction of ANN is that it produces excellent results with data having difficult patterns. (Appendix – 35)

7. EVALUATION:

Following 5 evaluation metrics have been used:

Metrics	Decision Tree	Adaboost	Random Forest	SVM	ANN
Accuracy:	0.6341	0.6268	0.6413	0.6848	**0.7065**
Sensitivity:	0.2574	0.3069	0.3366	0.4158	**0.4851**
Specificity:	**0.8514**	0.8114	0.8171	0.8400	0.8343
Precision:	0.5000	0.4844	0.5152	0.6000	**0.6282**
AUC:	0.5998	0.6194	0.6647	0.7293	**0.7409**

Accuracy:

It gives overall success of model predictions. It determines proportion of all correct predictions. It is useful where aim is to detect accurately both positive class and negative class correctly.

In our case, ANN has highest accuracy of 0.7065. Since, our dataset is imbalanced, with success ICO campaigns forming 37.15% of dataset, we need to complement this metric with another metric. It is because accuracy does not work well with imbalanced dataset. For imbalanced datasets, accuracy can still be high even if a model incorrectly predicts all minority class observations as majority class.

Sensitivity (also known as Recall):

This metric determines how good the model is in predicting positive class correctly out of all positive class in dataset. This metric is useful when aim is not to skip any positive class in prediction though this can lead to increase in false positive.

In our case, ANN has the highest sensitivity 0.4851. It means the best of our model i.e. ANN will be predicting success campaigns as failure 51.49% of the time.

Specificity:

It is similar to sensitivity except that it focuses on negative class. It states the proportion of negative class observations correctly predicted out of all actual negative class observations.

In our case, Decision Tree (DT) has the highest specificity of 0.8514. It means 85.14% of the time model will predict negative class correctly.

Sensitivity and Specificity help in determining whether model is a cautious one or a vigorous one.

Precision:

It tells relevancy of results of model. It tells the proportion of all positive class correctly predicted by a model to all positive class predicted(Lantz, 2019, p.329). It is useful in detecting how much the model is creating/giving false positives while it is identifying true positives.

In our case, ANN has the highest precision of 0.6262. It means of all instances where it has predicted ICO campaign as success, 62.62% were actually successful.

AUC:

This metric is a visualization aid based on sensitivity and specificity. It is an acronym for Area Under ROC (Receiver Operating Characteristics) Curve. It helps in understanding the tradeoffs under different conditions. It suggests for a specific increase in true positive rate (sensitivity) what would be increase in false positive rate (1-specifity). An area of 0.5 means it is just as good as random guess. The model has no prediction value. An area below 0.5 means model is worse than a random guess. A good model is one which has AUC above 0.5.

In our case, ANN has the best AUC score of 0.7409 which means it is has better prediction power than other models.

8. LIMITATIONS OF WORK AND FUTURE WORK:
    8.1. The feature brandSlogan can be used in modelling using text mining or sentiment analysis.
    8.2. Imputing missing values or blank values in priceUSD and teamSize with median can bring bias in results. If possible, attempt can be made to get accurate values from owner of data.
    8.3. Cross fold validation could be used which will give an idea about real world performance of model.

9. CONCLUSION (DEPLOYMENT):

From analysis in preceding sections, it can be noted that Artificial Neural Networks (ANN) has the best prediction performance with Area under ROC Curve (AUC) of 0.7409. Overall, it can be discerned from analysis that none of the model has sensitivity greater than 0.5. It means none of the model can predict success ICO campaign half of the time. On the other hand, all models are doing good in predicting failed ICO campaigns which is evident from values of specificity. This is the reason why accuracy values are in reasonable range. The best model in terms of specificity is Decision Tree (DT).

If explanation of model is the consideration, then Adaboost will be suitable choice as it gives detail about how decision tree has been formed. If computation cost is not an issue, then ANN is the best model.

The report can be useful to startups founders, investors, and regulators. Ventures can use the model to foresee success of their ICO campaign based on features which are used in the models. It can signal them to take corrective action if required in their product strategy. Investors can use the model to get an indication about which ICO campaigns are strong and should be selected for making investment. Regulators can use the model to supplement their understanding of the ICO sector. Using this model, they can predict which ICO campaigns would be successful then they can review this after 6 months or 1 year to see what happened to those ventures and develop an understanding about sector and whether model can be used as early predictor of failure or success.

References:

bitcoin.org. 2023. *Some Bitcoin words you might hear.* [Online]. [Accessed 10-04-2023]. Available from: https://bitcoin.org/en/vocabulary#block

Breiman, L. 2001. Random forests. *Machine learning.* **45**(1), pp.5-32.

ByBitLearn. 2022. *Utility Token.* [Online]. [Accessed 05-04-2023]. Available from: https://learn.bybit.com/glossary/definition-utility-token/

ethereum.org. 2023. *WHAT IS ETHER (ETH)?* [Online]. [Accessed 10-04-2023]. Available from: https://ethereum.org/en/eth/

Fisch, C. 2019. Initial coin offerings (ICOs) to finance new ventures. *Journal of Business Venturing.* **34**(1), pp.1-22.

IBMCloudEducation. 2023. *Random Forest.* [Online]. [Accessed 16-04-2023]. Available from: https://www.ibm.com/in-en/topics/random-forest

Jones, S.S.a.C. 2018. *Buyer Beware: Hundreds of Bitcoin Wannabes Show Hallmarks of Fraud.* [Online]. [Accessed 05-04-2023]. Available from: https://www.wsj.com/articles/buyer-beware-hundreds-of-bitcoin-wannabes-show-hallmarks-of-fraud-1526573115

Kohli, C., Thomas, S. and Suri, R. 2013. Are you in good hands?: slogan recall: what really matters. *Journal of Advertising Research.* **53**(1), pp.31-42.

Kumar, A. 2023. *Python – Replace Missing Values with Mean, Median & Mode.* [Online]. [Accessed 09-04-2023]. Available from: https://vitalflux.com/pandas-impute-missing-values-mean-median-mode/#:~:text=Mean%20imputation%20is%20often%20used,to%20outliers%20than%20the%20mean.

Lantz, B. 2019. *Machine learning with R : expert techniques for predictive modeling.* Third edition. ed. Birmingham: Packt Publishing.

Momtaz, P. 2020. Initial Coin Offerings. *PLoS ONE.* **15**, pp.1-30.

Schapire, R.E. and Freund, Y. 2012. *Boosting foundations and algorithms.* Cambridge, MA: MIT Press.

stellar.org. 2023. *Intro to Stellar.* [Online]. [Accessed 10-04-2023]. Available from: https://stellar.org/learn/intro-to-stellar

APPENDIX

Appendix – 1: Viewing the structure of dataset using str() function:

```
> str(ico)
'data.frame':  2767 obs. of  16 variables:
 $ ID                   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ success              : chr  "N" "N" "N" "Y" ...
 $ brandSlogan          : chr  "Is One of Its Kind ERC-20 Decentralized St
able Asset" "The Ultimate Blockchain Gaming Platform" "Simple Automated In
vestment App Driven by AI & ML" "International Real Estate Crowdfunding Pl
atform" ...
 $ hasVideo             : int  1 1 1 1 1 1 1 1 1 1 ...
 $ rating               : num  4 4.3 4.4 4.3 4.3 4.7 4.1 4.5 4.8 4.2 ...
 $ priceUSD             : num  30 0.13 0.01 NA 0.03 0.1 0.02 2.8 50 0.1 ..
.
 $ countryRegion        : chr  "Singapore" "Malta" "UK" "Netherlands" ...
 $ startDate            : chr  "01/10/2019" "07/09/2018" "01/07/2019" "01/
10/2019" ...
 $ endDate              : chr  "01/10/2019" "12/10/2018" "30/06/2020" "15/
12/2019" ...
 $ teamSize             : int  31 20 10 27 14 43 20 31 8 29 ...
 $ hasGithub            : int  1 1 1 1 1 1 1 1 1 1 ...
 $ hasReddit            : int  1 1 1 1 1 1 1 1 1 1 ...
 $ platform             : chr  "Ethereum" "XAYA" "Stellar" "Separate block
chain" ...
 $ coinNum              : num  5.10e+05 2.25e+08 5.00e+09 1.25e+08 5.00e+0
9 ...
 $ minInvestment        : int  0 1 1 1 1 1 1 1 1 1 ...
 $ distributedPercentage: num  0.49 0.41 0.4 0.13 0.5 0.5 0.25 0.1 0.05 0.
15 ...
```

Appendix – 2: Viewing summary details of features using summary() function:

```
> summary(ico)
       ID           success           brandSlogan          hasVideo
 Min.   :   1.0   Length:2767        Length:2767        Min.   :0.0000
 1st Qu.: 692.5   Class :character   Class :character   1st Qu.:0.0000
 Median :1384.0   Mode  :character   Mode  :character   Median :1.0000
 Mean   :1384.0                                        Mean   :0.7261
 3rd Qu.:2075.5                                        3rd Qu.:1.0000
 Max.   :2767.0                                        Max.   :1.0000

     rating         priceUSD        countryRegion       startDate
 Min.   :1.000   Min.   :    0.00   Length:2767        Length:2767
 1st Qu.:2.600   1st Qu.:    0.04   Class :character   Class :character
 Median :3.100   Median :    0.12   Mode  :character   Mode  :character
 Mean   :3.121   Mean   :   19.01
 3rd Qu.:3.700   3rd Qu.:    0.50
 Max.   :4.800   Max.   :39384.00
                 NA's   :180
   endDate             teamSize       hasGithub        hasReddit
 Length:2767        Min.   : 1.00   Min.   :0.0000   Min.   :0.0000
 Class :character   1st Qu.: 7.00   1st Qu.:0.0000   1st Qu.:0.0000
 Mode  :character   Median :12.00   Median :1.0000   Median :1.0000
                    Mean   :13.11   Mean   :0.5779   Mean   :0.6328
                    3rd Qu.:17.00   3rd Qu.:1.0000   3rd Qu.:1.0000
                    Max.   :75.00   Max.   :1.0000   Max.   :1.0000
                    NA's   :154
   platform            coinNum          minInvestment    distributedPercen
tage
 Length:2767        Min.   :1.200e+01   Min.   :0.0000   Min.   :  0.000
 Class :character   1st Qu.:5.000e+07   1st Qu.:0.0000   1st Qu.:  0.400
 Mode  :character   Median :1.800e+08   Median :0.0000   Median :  0.550
                    Mean   :8.178e+12   Mean   :0.4532   Mean   :  1.061
                    3rd Qu.:6.000e+08   3rd Qu.:1.0000   3rd Qu.:  0.700
                    Max.   :2.262e+16   Max.   :1.0000   Max.   :869.750
```

Appendix – 3: Finding out missing values in feature "priceUSD" which corroborates finding from summary details:

```
> sum(is.na(ico_org$priceUSD))
[1] 180
```

Appendix-4: Finding out number of observations where priceUSD is zero:

```
> count(filter(ico, priceUSD==0))
    n
1 152
```

Appendix – 5: Output of hist() function used for creating histogram for feature "priceUSD":

```
> histd1 <- hist(ico$priceUSD)
> histd1
$breaks
[1]     0  5000 10000 15000 20000 25000 30000 35000 40000

$counts
[1] 2586    0    0    0    0    0    0    1

$density
[1] 1.999227e-04 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000
000e+00
[7] 0.000000e+00 7.730963e-08

$mids
[1]  2500  7500 12500 17500 22500 27500 32500 37500

$xname
[1] "ico$priceUSD"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
```

Appendix – 6: Output of Box and Whisker plot to investigate outlier in feature priceUSD:

```
> box1 <- boxplot(ico$priceUSD, main = "Boxplot of feature - priceUSD",
+                 ylab = "Price of coin/token in US Dollars")
> box1
$stats
     [,1]
[1,] 0.00
[2,] 0.04
[3,] 0.12
[4,] 0.50
[5,] 1.19

$n
[1] 2587

$conf
          [,1]
[1,] 0.1057105
[2,] 0.1342895

$out
  [1]    30.00     2.80    50.00     3.23     2.15  1000.00   888.88     5
.76     1.58
```

```
 [10]     2.77     1.30    67.27     2.00    10.00     2.00     1.46     1
.40     4.50
 [19]     5.00     2.66     3.00     1.30     1.60 39384.00     1.26     2
.47     2.39
 [28]     3.20     2.00     2.50     1.25     5.00     2.00     5.00     2
.00     2.05
 [37]     3.00     1.30   295.34     1.76     1.60     3.66     5.85     2
.80    25.00
 [46]     1.50     1.42     3.30     1.55   124.00    23.91   170.95     1
.47     1.55
 [55]     5.59     3.60     1.66     1.50     2.88    95.71    10.00    10
.00     2.08
 [64]     1.23     2.32     1.63     5.00     2.00     1.65   214.29     1
.21   189.00
 [73]     3.03     1.20     7.00     4.30     8.69     1.52     4.49     2
.00     1.22
 [82]     2.43     2.56   295.34     3.09     1.25    10.73    34.82     1
.50     1.26
 [91]     3.00    73.39     2.58     1.20     2.76     2.04     1.36     4
.00    39.50
[100]     4.65     7.00    62.21     1.44    10.00    22.00     2.00     3
.01    10.00
[109]   230.77    14.00     6.60     1.43     1.64     3.00     3.00     3
.50     1.75
[118]     2.00     2.00     2.75     1.23     4.43     1.37     2.66     4
.76    21.00
[127]   112.00     2.00     1.56    24.64     1.47     2.10     1.25     4
.44     5.00
[136]     3.48     3.17     3.00     7.02     1.40     1.70     4.77     2
.00    14.25
[145]     2.40    39.97     2.75     3.81    20.00     1.33     2.50     1
.25     3.90
[154]    51.65     7.00     1.65     2.30    90.83     6.96   167.77   409
.17     6.69
[163]     1.80     2.70     7.50    10.00     2.12     1.60   647.74     1
.33    10.00
[172]     1.50     1.20     1.55   525.01   642.40     2.13     1.83    20
.00     5.00
[181]     1.64    11.62     4.56     7.23     2.85     2.13    17.34     2
.40     2.92
[190]   609.88     6.88     3.97    14.50     5.29     3.31     3.99     1
.50    10.00
[199]     1.97    14.01     1.58     3.00     4.52    78.80     1.31   129
.74   237.49
[208]     4.50    97.38     1.22     1.25    91.59    13.00     3.51     1
.20    10.00
[217]     1.20     2.19    12.28     3.41     1.50     2.00     2.00     3
.94     9.52
[226]     2.00    42.15     1.20     1.35     1.20    32.00    11.30    11
.72     3.75
[235]   469.86     2.50     3.78     1.50     5.18     3.48

$group
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [41] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [81] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[121] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[161] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[201] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

$names
[1] ""
```

Appendix – 7: Calculating Standard Deviation of priceUSD:

```
> sd(ico$priceUSD, na.rm = TRUE)
[1] 775.2871
```

Appendix – 8: Summary Details of feature "coinNum" expressing number as a whole:

```
> format(summary(ico$coinNum), scientific = FALSE)
                Min.              1st Qu.               Median
"               12" "         50000000" "        180000000" "

                Mean              3rd Qu.                 Max.
"     8177879989176"          600000000" "22619078416760300"
```

Appendix – 9: Output of hist() function used for creating histogram for feature "coinNum":

```
> histd3 <- hist(ico$coinNum)
> histd3
$breaks
 [1] 0.0e+00 2.0e+15 4.0e+15 6.0e+15 8.0e+15 1.0e+16 1.2e+16 1.4e+16 1.6e+
16 1.8e+16
[11] 2.0e+16 2.2e+16 2.4e+16

$counts
 [1] 2766    0    0    0    0    0    0    0    0    0    0    1

$density
 [1] 4.998193e-16 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.00
0000e+00
 [7] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.80
7011e-19

$mids
 [1] 1.0e+15 3.0e+15 5.0e+15 7.0e+15 9.0e+15 1.1e+16 1.3e+16 1.5e+16 1.7e+
16 1.9e+16
[11] 2.1e+16 2.3e+16

$xname
[1] "ico$coinNum"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
```

Appendix – 10: Output of Box and Whisker plot to investigate outlier in feature coinNum:

```
> box2 <- boxplot(ico$coinNum, main = "Boxplot of feature - coinNum",
+                 ylab = "Number of coins to be issued in ICO campaign")
> box2
$stats
         [,1]
[1,] 1.20e+01
[2,] 5.00e+07
[3,] 1.80e+08
[4,] 6.00e+08
[5,] 1.42e+09

$n
[1] 2767
```

```
$conf
          [,1]
[1,] 163479804
[2,] 196520196

$out
  [1] 5.000000e+09 5.000000e+09 2.500000e+09 6.245750e+09 1.000000e+10 2.4
62500e+09
  [7] 2.000000e+09 1.500000e+09 3.250000e+09 4.000000e+10 1.400000e+10 2.9
16000e+09
 [13] 3.250000e+09 2.880000e+09 2.600000e+09 1.388889e+11 6.000000e+09 1.2
60000e+11
 [19] 1.875000e+09 8.500000e+09 1.000000e+10 8.500000e+09 3.500000e+09 5.0
00000e+09
 [25] 8.000000e+09 1.450000e+09 1.431000e+09 6.500000e+10 5.000000e+09 1.5
00000e+09
 [31] 2.300000e+09 3.600000e+09 3.000000e+09 2.750000e+09 1.755000e+10 1.6
40000e+09
 [37] 5.000000e+09 2.500000e+09 2.500000e+11 3.055000e+09 3.500000e+09 4.0
00000e+09
 [43] 1.780000e+09 3.000000e+09 5.250000e+09 2.000000e+09 5.000000e+09 6.0
00000e+09
 [49] 1.800000e+09 7.200000e+09 1.000000e+10 3.000000e+09 2.700000e+10 5.5
00000e+09
 [55] 6.000000e+09 7.200000e+09 3.500000e+09 3.500000e+09 2.100000e+09 4.3
92000e+09
 [61] 5.000000e+09 3.874000e+09 1.000000e+10 2.000000e+09 5.000000e+09 2.0
00000e+09
 [67] 2.000000e+10 1.950000e+09 1.800000e+09 6.660000e+09 1.750000e+09 9.0
00000e+09
 [73] 5.000000e+09 5.000000e+09 3.000000e+09 2.700000e+09 3.000000e+10 2.0
00000e+09
 [79] 2.700000e+09 1.000000e+10 1.025000e+10 4.560000e+09 4.000000e+09 3.0
00000e+09
 [85] 2.400000e+09 4.200000e+09 6.930000e+10 2.165609e+10 3.000000e+09 1.0
00000e+11
 [91] 1.500000e+09 1.470000e+09 5.000000e+09 2.000000e+09 1.750000e+10 2.0
00000e+09
 [97] 7.500000e+10 3.584000e+09 1.500000e+09 2.250000e+10 7.000000e+09 4.2
00000e+10
[103] 4.000000e+09 3.300000e+09 2.475198e+09 2.826250e+09 3.000000e+09 3.0
00000e+09
[109] 2.000000e+09 5.200000e+09 1.800000e+09 3.000000e+09 4.000000e+09 1.5
75000e+09
[115] 4.500000e+09 9.000000e+09 3.000000e+09 1.000000e+11 1.750000e+10 3.6
00000e+09
[121] 3.000000e+09 7.200000e+09 1.800000e+09 1.500000e+09 6.000000e+10 1.5
00000e+09
[127] 2.500000e+09 3.000000e+09 4.000000e+10 1.500000e+09 6.500000e+09 3.7
00000e+09
[133] 3.100000e+09 2.200000e+09 8.140000e+09 5.000000e+09 4.800000e+10 4.5
00000e+09
[139] 8.125000e+09 5.280000e+10 1.500000e+09 1.000000e+10 1.400000e+10 1.0
00000e+10
[145] 5.000000e+09 6.000000e+09 1.209000e+10 1.798654e+10 5.000000e+09 3.0
00000e+09
[151] 2.000000e+09 1.000000e+11 3.750000e+09 1.750000e+10 8.250000e+09 3.5
00000e+09
[157] 8.910000e+10 5.000000e+09 2.200000e+09 8.400000e+09 5.400000e+09 2.0
00000e+09
[163] 2.350000e+09 1.551000e+09 3.000000e+09 3.000000e+10 9.000000e+09 1.2
00000e+12
[169] 7.500000e+09 2.500000e+10 6.200000e+09 4.800000e+09 6.000000e+09 3.0
00000e+09
[175] 1.771429e+09 1.550000e+09 9.600000e+09 6.000000e+09 1.300000e+10 1.5
00000e+09
[181] 2.500000e+09 4.000000e+09 1.500000e+10 5.236000e+09 2.000000e+09 5.5
00000e+11
```

[187] 3.000000e+09 2.000000e+09 4.289400e+09 7.000000e+09 1.692000e+09 3.5
00000e+09
[193] 6.000000e+09 7.000000e+09 1.640000e+09 7.000000e+09 1.000000e+10 1.5
10000e+09
[199] 1.500000e+09 4.000000e+09 2.400000e+10 5.100000e+09 1.400000e+10 4.2
00000e+10
[205] 7.000000e+09 6.500000e+09 1.300000e+10 2.520000e+09 4.000000e+10 5.3
10000e+09
[211] 3.500000e+09 3.000000e+09 2.750000e+09 7.000000e+09 3.500000e+09 1.5
00000e+09
[217] 9.000000e+09 2.750000e+10 3.750000e+09 8.000000e+09 5.000000e+09 5.0
00000e+10
[223] 2.000000e+09 1.000000e+10 5.000000e+09 2.000000e+09 1.704900e+09 3.4
00000e+10
[229] 8.000000e+09 1.500000e+10 1.600000e+09 8.428509e+09 4.000000e+09 2.4
00000e+11
[235] 2.307692e+09 3.552000e+10 1.500000e+09 2.450000e+09 2.980000e+09 2.0
00000e+09
[241] 1.500000e+09 5.800000e+09 7.000000e+09 2.601000e+09 1.600000e+09 7.0
00000e+09
[247] 3.000000e+09 1.500000e+09 4.000000e+10 2.261908e+16 2.700000e+09 2.4
00000e+09
[253] 2.100000e+09 3.000000e+09 6.450000e+09 3.400000e+09 4.800000e+09 3.0
00000e+09
[259] 3.500000e+09 2.000000e+09 4.000000e+09 2.000000e+09 1.500000e+10 7.0
00000e+09
[265] 1.000000e+12 3.000000e+09 4.000000e+10 2.700000e+09 1.600000e+09 3.5
00000e+10
[271] 4.000000e+09 1.200000e+10 3.500000e+09 2.000000e+09 9.000000e+10 6.0
00000e+09
[277] 4.000000e+09 2.200000e+09 6.000000e+09 4.000000e+09 1.500000e+10 4.5
00000e+09
[283] 4.950000e+09 5.500000e+09 3.500000e+09 3.000000e+09 4.000000e+09 9.0
00000e+09
[289] 1.000000e+10 2.550000e+09 8.550000e+09 1.400000e+10 5.000000e+10 4.7
50000e+09
[295] 1.900000e+09 2.000000e+09 2.500000e+11 5.500000e+09 1.000000e+10 2.5
00000e+09
[301] 6.000000e+09 6.500000e+09 1.000000e+10 5.000000e+09 1.500000e+10 4.0
00000e+09
[307] 5.000000e+09 1.260000e+10 2.000000e+10 7.000000e+09 4.200000e+09 3.0
00000e+09
[313] 9.894000e+10 2.450000e+09 1.000000e+10 3.750000e+09 2.800000e+10 2.5
00000e+10
[319] 1.500000e+09 2.000000e+09 4.000000e+09 3.141593e+09 2.100000e+09 2.5
00000e+09
[325] 3.500000e+09 6.000000e+09 8.000000e+09 5.400000e+09 2.000000e+10 5.0
00000e+09
[331] 1.000000e+10 5.000000e+10 1.700000e+09 1.000000e+10 8.000000e+10 4.9
00000e+09
[337] 1.650000e+09 5.040000e+09 1.800000e+09 1.400000e+10 4.620000e+10 5.5
00000e+11
[343] 6.750000e+09 1.000000e+10 7.800000e+09 4.000000e+10 6.000000e+09 3.0
00000e+09
[349] 4.000000e+09 6.000000e+09 1.680000e+10 2.500000e+10 2.400000e+09 1.2
80023e+10
[355] 6.000000e+09 4.000000e+09 7.000000e+09 8.200000e+09 5.000000e+09 3.8
25000e+09
[361] 2.100000e+09 1.575000e+09 3.897741e+09 1.500000e+09 2.250000e+09 5.0
00000e+10
[367] 3.000000e+09 7.500000e+09 2.000000e+09 7.220000e+09 7.000000e+10 1.2
00000e+10
[373] 5.000000e+09 1.000000e+11 3.000000e+09 7.502724e+09 2.000000e+09 9.0
00000e+09
[379] 7.500000e+10 4.000000e+09 3.000000e+10 5.100000e+09 2.220000e+09 4.0
00000e+09
[385] 2.050000e+10 3.200000e+09 2.000000e+09 1.500000e+09 2.500000e+09 6.0
00000e+09
[391] 7.200000e+09 3.000000e+10 7.000000e+09 1.500000e+11 3.800000e+10

```
$group
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [41] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [81] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[121] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[161] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[201] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[241] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[281] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[321] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[361] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1

$names
[1] ""
```

Appendix – 11: Representing figures in $stats from box plot in appendix – 10 in whole numbers:

```
> format(box2$stats, scientific = FALSE)
     [,1]
[1,] "         12"
[2,] "  50000000"
[3,] " 180000000"
[4,] " 600000000"
[5,] "1420000000"
```

Appendix – 12: Calculating Standard Deviation for feature "coinNum":

```
> sd(ico$coinNum)
[1] 4.300018e+14
> 3*sd(ico$coinNum)
[1] 1.290005e+15
```

Appendix – 13: Output of Box and Whisker plot to investigate outlier in feature "distributedPercentage":

```
> > box3 <- boxplot(ico$distributedPercentage, main = "Boxplot of feature
- distributedPercentage",
+                 ylab = "Proportion of coins to be distributed to investo
rs" )

> box3
$stats
     [,1]
[1,] 0.00
[2,] 0.40
[3,] 0.55
[4,] 0.70
[5,] 1.00

$n
[1] 2767
```

```
$conf
          [,1]
[1,] 0.540989
[2,] 0.559011

$out
 [1]   1.66   4.00   9.52  62.50 266.25 869.75  20.00 100.00  50.00  60.00

$group
 [1] 1 1 1 1 1 1 1 1 1 1

$names
[1] "1"
```

Appendix – 14: Using table() function to count categories in feature "success":

```
> table(ico$success)

   N    Y
1739 1028
> prop.table(table(ico$success))*100

       N        Y
62.84785 37.15215
```

Appendix – 15: Using table() function to count categories in feature "success":

```
> rev(sort(table(ico$countryRegion)))

                        Singapore                               USA
                              312                               296
                               UK                           Estonia
                              285                               191
                      Switzerland                            Russia
                              140                               138
                                                     Cayman Islands
                               71                                67
                          Germany                       Netherlands
                               61                                59
                            Malta                         Australia
                               59                                57
                           Canada             British Virgin Islands
                               52                                45
                           France               United Arab Emirates
                               43                                41
                            India                         Gibraltar
                               38                                36
                        Seychelles                         Indonesia
                               31                                31
                      South Korea                             Belize
                               30                                29
                           Cyprus                      South Africa
                               25                                24
                         Slovenia                           Romania
                               24                                24
                          Nigeria                    Czech Republic
                               24                                23
                           Poland                             China
                               21                                21
                          Ukraine                          Bulgaria
                               19                                19
                            Spain                        Lithuania
                               18                                18
                            Japan                           Turkey
                               17                                16
```

Malaysia	Israel
16	16
Georgia	Latvia
16	15
Ireland	Philippines
15	14
Thailand	Italy
13	12
Saint Kitts and Nevis	Brazil
11	11
Austria	Serbia
11	10
Liechtenstein	Panama
10	9
Costa Rica	New Zealand
9	8
Vietnam	Sweden
7	7
Portugal	Norway
7	7
Mexico	Mauritius
7	7
Luxembourg	Croatia
6	6
Belarus	Marshall Islands
6	5
Greece	Macedonia
5	4
Isle of Man	Hungary
4	4
Denmark	Bermuda
4	4
Belgium	Finland
4	3
Chile	Bahamas
3	3
Argentina	Anguilla
3	3
Venezuela	Vanuatu
2	2
Tanzania	Peru
2	2
Pakistan	Mongolia
2	2
Kazakhstan	Iceland
2	2
Colombia	Afghanistan
2	2
Zimbabwe	usa
1	1
Tunisia	Timor-Leste
1	1
Syria	Slovakia
1	1
SINGAPORE	Sierra Leone
1	1
Saudi Arabia	Samoa
1	1
Saint Vincent and the Grenadines	Puerto Rico
1	1
Northern Mariana Islands	New Caledonia
1	1
Morocco	Montenegro
1	1
Monaco	México
1	1
Kyrgyzstan	Kuwait
1	1
india	Honduras

```
                      1                       1
        Guinea-Bissau                   Ghana
                      1                       1
     French Polynesia                   Egypt
                      1                       1
              Ecuador     Dominican Republic
                      1                       1
              Curaçao                 Curacao
                      1                       1
                Congo                Cambodia
                      1                       1
Bosnia and Herzegovina                Barbados
                      1                       1
           Bangladesh                 Armenia
                      1                       1
              Andorra
                      1
```

Appendix – 16: Using table() function to count categories in feature "platform":

```
> table(ico$platform)
```

```
                          Ethereum
                     6                        1
                   NEO                   Komodo
                     1                        1
               Acclaim                   Achain
                     1                        1
                  AION                   Akroma
                     1                        1
      Apollo Blockchain                    Ardor
                     1                        1
                  BEP2                  Bitcoin
                     1                       10
              BitForex                 Bitshares
                     1                        2
             BitShares                    Bitsmo
                     1                        1
            Blockchain                       BTC
                     1                        3
ChainRepublik Blockchain                  Coffe
                     1                        1
              Coincart              Counterparty
                     1                        1
            CryptoKami               CryptoNight
                     1                        1
CryptoNote-based Blockchain                 DAG
                     1                        1
      DECOIN Blockchain                    DPoS
                     1                        3
                  DPOS          ENLTE PLATFORM
                     1                        1
                 ENTRY                     Eos
                     1                        1
                   EOS                   ERC20
                    16                        2
                   ETH                Ethererum
                     1                        2
              Ethereum                 Ethereum
                  2352                       23
            Ethereum                  Ethereum
                    16                        9
         Ethereum                  Ethereum
                     2                        2
       Ethereum, Waves                   Etherum
                     1                        1
```

Fiber	1	Filecoin network	1
GoChain	1	Graphene	4
Hard-Fork of Litecoin	1	Hybrid	1
Hyperledger	2	ICON	3
Infinity Blockchain	1	iOlite Blockchain	1
IOV Blockchain	1	IronGeekChain	1
ISL-Blockchain	1	JPMorganChase	1
Keccak	1	Komodo	3
Lisk	1	Litecoin	1
MAHRA platform	1	Monero	1
Multichain	1	MultiChain	1
Native	2	Neblio	1
Nem	1	NEM	12
Neo	1	NEO	20
Neurochain	1	New Blockchain	1
Newton	1	Nilechain	1
NXT	3	Pivx	1
PivX	1	POS	2
POS + POW	1	POS,POW	1
PoW	2	pow/pos	1
PoW/PoS	1	QRC	1
QTUM	2	Ripemd160	1
RSK	1	Scrypt	14
Separate blockchain	30	Separate Blockchain	6
Separate Blockchain	1	Separate Blockchain	2
SHA256 Coin	1	Slatechain	1
SmartX	1	ST20	1
StartEngine	1	Steem	2
Stellar	41	Stellar Protocol	1
Stratis	2	STRATIS	1
TEZOS	1	Tomochain	2
TON	1	Tron	4
TRON	3	Tron	1
Tron	1	TTchain	1
UNIVERSA		VASYA	

```
                     1                          1
               VeChain          VeChainThor VIP180
                     2                          1
            Ventureon                    Wanchain
                     1                          1
                 Waves                      WAVES
                    56                          1
               WizeBit                        X11
                     1                          8
        X11 blockchain                        X13
                     1                          1
                   x13                        XAYA
                     1                          1
                  xDAC                   YouToken
                     1                          1
               Zilliqa                       Zuum
                     1                          1
```

Appendix – 17: Removing observation where priceUSD = 39384 because it is an outlier

```
> filter(ico_main, priceUSD == 39384)
   ID success                                brandSlogan hasVideo
1 384       N Participate in the Global Revenue of Osmium        1
  rating priceUSD countryRegion  startDate    endDate teamSize
1    4.2    39384        Serbia 09/09/2018 30/07/2019       16
  hasGithub hasReddit platform coinNum minInvestment
1         1         1 Ethereum 2.1e+08             0
  distributedPercentage
1                  0.84
> ico_main <- ico_main[-384,]
> filter(ico_main, priceUSD == 39384)
 [1] ID                    success               brandSlogan
 [4] hasVideo              rating                priceUSD
 [7] countryRegion         startDate             endDate
[10] teamSize              hasGithub             hasReddit
[13] platform              coinNum               minInvestment
[16] distributedPercentage
<0 rows> (or 0-length row.names)
```

Appendix – 18: Visualising the distribution of priceUSD using histogram after deletion of outlier:

```
> hist_priceUSD <- hist(ico_main$priceUSD, breaks = 100)
> hist_priceUSD
$breaks
  [1]    0   10   20   30   40   50   60   70   80   90  100  110  120
 [14]  130  140  150  160  170  180  190  200  210  220  230  240  250
 [27]  260  270  280  290  300  310  320  330  340  350  360  370  380
 [40]  390  400  410  420  430  440  450  460  470  480  490  500  510
 [53]  520  530  540  550  560  570  580  590  600  610  620  630  640
 [66]  650  660  670  680  690  700  710  720  730  740  750  760  770
 [79]  780  790  800  810  820  830  840  850  860  870  880  890  900
 [92]  910  920  930  940  950  960  970  980  990 1000

$counts
  [1] 2533   13    6    4    2    1    2    2    0    4    0    1    2
 [14]    0    0    0    1    1    1    0    0    1    0    2    0    0
 [27]    0    0    0    2    0    0    0    0    0    0    0    0    0
 [40]    0    1    0    0    0    0    0    1    0    0    0    0    0
 [53]    1    0    0    0    0    0    0    0    1    0    0    0    2
 [66]    0    0    0    0    0    0    0    0    0    0    0    0    0
 [79]    0    0    0    0    0    0    0    0    0    0    1    0    0
 [92]    0    0    0    0    0    0    0    0    1

$density
  [1] 9.795050e-02 5.027069e-04 2.320186e-04 1.546790e-04 7.733952e-05
  [6] 3.866976e-05 7.733952e-05 7.733952e-05 0.000000e+00 1.546790e-04
```

```
 [11] 0.000000e+00 3.866976e-05 7.733952e-05 0.000000e+00 0.000000e+00
 [16] 0.000000e+00 3.866976e-05 3.866976e-05 3.866976e-05 0.000000e+00
 [21] 0.000000e+00 3.866976e-05 0.000000e+00 7.733952e-05 0.000000e+00
 [26] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 7.733952e-05
 [31] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [36] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [41] 3.866976e-05 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [46] 0.000000e+00 3.866976e-05 0.000000e+00 0.000000e+00 0.000000e+00
 [51] 0.000000e+00 0.000000e+00 3.866976e-05 0.000000e+00 0.000000e+00
 [56] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [61] 3.866976e-05 0.000000e+00 0.000000e+00 0.000000e+00 7.733952e-05
 [66] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [71] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [76] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [81] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [86] 0.000000e+00 0.000000e+00 0.000000e+00 3.866976e-05 0.000000e+00
 [91] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [96] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 3.866976e-05

$mids
  [1]   5  15  25  35  45  55  65  75  85  95 105 115 125 135 145 155
 [17] 165 175 185 195 205 215 225 235 245 255 265 275 285 295 305 315
 [33] 325 335 345 355 365 375 385 395 405 415 425 435 445 455 465 475
 [49] 485 495 505 515 525 535 545 555 565 575 585 595 605 615 625 635
 [65] 645 655 665 675 685 695 705 715 725 735 745 755 765 775 785 795
 [81] 805 815 825 835 845 855 865 875 885 895 905 915 925 935 945 955
 [97] 965 975 985 995

$xname
[1] "ico_main$priceUSD"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
```

Appendix – 19: Imputing missing values in priceUSD with median:

```
> ico_main[is.na(ico_main$priceUSD),"priceUSD"] <- median(ico_main$priceUS
D, na.rm =TRUE)
> summary(ico_main$priceUSD)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
   0.0000    0.0425    0.1200    3.5528    0.5000 1000.0000
```

Appendix – 20: Imputing zero values in priceUSD with median:

```
> ico_main$priceUSD[ico_main$priceUSD == 0] <- median(ico_main$priceUSD)
> summary(ico_main$priceUSD)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
   0.010    0.060    0.120    3.559    0.500 1000.000
```

Appendix – 21: Imputing missing values in teamSize with median:

```
> ico_main[is.na(ico_main$teamSize), "teamSize"] <- median(ico_main$teamSi
ze, na.rm = TRUE)

> summary(ico_main$teamSize)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    8.00   12.00   13.04   17.00   75.00
```

Appendix – 22: Removing outlier in feature "coinNum":

```
> summary(ico_main$coinNum)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
1.200e+01 5.000e+07 1.800e+08 8.181e+12 6.000e+08 2.262e+16
> which(ico_main$coinNum == 22619078416760300) # to find out the row numbe
r in a dataframe
```

```
[1] 1593
> ico_main <- ico_main[-1593,]
> filter(ico_main, coinNum == 22619078416760300)
 [1] ID                    success               brandSlogan
 [4] hasVideo              rating                priceUSD
 [7] countryRegion         startDate             endDate
[10] teamSize              hasGithub             hasReddit
[13] platform              coinNum               minInvestment
[16] distributedPercentage
<0 rows> (or 0-length row.names)
> summary(ico_main$coinNum)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
1.200e+01 5.000e+07 1.800e+08 3.297e+09 6.000e+08 1.200e+12
```

Appendix – 23: Removing 11 observations where feature 'distributedPercentage' has incor rect values:

```
> filter(ico_main, distributedPercentage == 0)
   ID success                                  brandSlogan hasVideo
1 593       Y Decentralized Continuous Auditing & Reporting        1
  rating priceUSD countryRegion  startDate    endDate teamSize
1    3.7     0.25   Switzerland 09/02/2019 09/08/2019       11
  hasGithub hasReddit platform coinNum minInvestment
1         1         1 Ethereum   2e+06             1
  distributedPercentage
1                     0
> filter(ico_main, distributedPercentage > 1)
    ID success
1   99       Y
2  542       N
3  681       N
4  947       Y
5  965       N
6 1030       N
7 1404       N
8 1408       N
9 1656       N
10 1988      Y
                                                brandSlogan hasVideo
1                iCoin ICO - backed by real diamond mining.        1
2                                             Get Noticed!        1
3                          It's time to create your token        1
4                                  Welcome to Play2Live!        1
5                                The Coin You Can Bank On        1
6  Create Artificial Intelligence And Make Money From Water        1
7                   A new way to contribute to innovation        1
8                                           Sapien Wallet        1
9                               ZooomEx - Your Great Choice        0
10                        First ICO for Adult Entertainment        0
   rating priceUSD        countryRegion  startDate    endDate teamSize
1     4.1     1.00         Sierra Leone 05/08/2019 21/10/2019       13
2     3.8     0.39               Canada 06/04/2019 15/07/2019       10
3     3.7     0.02               France 04/11/2019 30/01/2020        8
4     4.1     0.05                Malta 21/02/2018 14/03/2018       23
5     3.5     2.88             Thailand 01/06/2018 30/06/2018        7
6     3.3     0.50          Switzerland 15/11/2019 15/05/2020        7
7     3.0     0.10 United Arab Emirates 10/07/2017 11/08/2017        9
8     3.0     0.12              Ukraine 27/04/2020 26/07/2020        7
9     4.2     0.03               Russia 19/08/2019 31/08/2019        3
10    2.7     2.75              Estonia 20/07/2017 29/08/2017        9
   hasGithub hasReddit     platform    coinNum minInvestment
1          1         1     Ethereum   10000000             1
2          1         1          NEO    4000000             1
3          0         1        Waves   20000000             1
4          0         0     Ethereum 1308800000             1
5          1         1          X11    6656250             0
6          1         1 Ethereum       17395000             0
```

```
7           1         1     Ethereum    3000000             0
8           0         1     Ethereum    1000000             1
9           0         0     Ethereum   50000000             0
10          0         0     Ethereum    3000000             1
   distributedPercentage
1                   1.66
2                   4.00
3                   9.52
4                  62.50
5                 266.25
6                 869.75
7                  20.00
8                 100.00
9                  50.00
10                 60.00
> ico_main <- ico_main[ico_main$distributedPercentage <= 1,]
> ico_main <- ico_main[ico_main$distributedPercentage != 0,]
> filter(ico_main, distributedPercentage == 0)
 [1] ID                    success               brandSlogan
 [4] hasVideo              rating                priceUSD
 [7] countryRegion         startDate             endDate
[10] teamSize              hasGithub             hasReddit
[13] platform              coinNum               minInvestment
[16] distributedPercentage
<0 rows> (or 0-length row.names)
> filter(ico_main, distributedPercentage > 1)
 [1] ID                    success               brandSlogan
 [4] hasVideo              rating                priceUSD
 [7] countryRegion         startDate             endDate
[10] teamSize              hasGithub             hasReddit
[13] platform              coinNum               minInvestment
[16] distributedPercentage
<0 rows> (or 0-length row.names)
```

Appendix – 24: Replacing blank value in feature 'countryRegion' with "unknown":

```
> table(ico_main$countryRegion)

                     71                  Afghanistan
                      1                            2
                Andorra                     Anguilla
                      1                            3
              Argentina                      Armenia
                      3                            1
              Australia                      Austria
                     57                           11
                Bahamas                   Bangladesh
                      3                            1
               Barbados                      Belarus
                      1                            6
                Belgium                       Belize
                      4                           29
                Bermuda  Bosnia and Herzegovina
                      4                            1
                 Brazil  British Virgin Islands
                     11                           44
               Bulgaria                     Cambodia
                     19                            1
                 Canada               Cayman Islands
                     51                           67
                  Chile                        China
                      3                           21
               Colombia                        Congo
                      2                            1
             Costa Rica                      Croatia
                      9                            6
                Curacao                      Curaçao
                      1                            1
                 Cyprus               Czech Republic
```

25	23
Denmark	Dominican Republic
4	1
Ecuador	Egypt
1	1
Estonia	Finland
190	3
France	French Polynesia
42	1
Georgia	Germany
16	61
Ghana	Gibraltar
1	36
Greece	Guinea-Bissau
5	1
Honduras	Hungary
1	4
Iceland	india
2	1
India	Indonesia
38	31
Ireland	Isle of Man
15	4
Israel	Italy
16	12
Japan	Kazakhstan
17	2
Kuwait	Kyrgyzstan
1	1
Latvia	Liechtenstein
15	10
Lithuania	Luxembourg
18	6
Macedonia	Malaysia
4	16
Malta	Marshall Islands
58	5
Mauritius	Mexico
7	7
México	Monaco
1	1
Mongolia	Montenegro
2	1
Morocco	Netherlands
1	59
New Caledonia	New Zealand
1	8
Nigeria	Northern Mariana Islands
24	1
Norway	Pakistan
7	2
Panama	Peru
9	2
Philippines	Poland
14	21
Portugal	Puerto Rico
7	1
Romania	Russia
24	137
Saint Kitts and Nevis	Saint Vincent and the Grenadines
11	1
Samoa	Saudi Arabia
1	1
Serbia	Seychelles
9	31
Singapore	SINGAPORE
312	1
Slovakia	Slovenia
1	24

```
                South Africa                  South Korea
                          24                           30
                       Spain                       Sweden
                          18                            7
                 Switzerland                        Syria
                         138                            1
                    Tanzania                     Thailand
                           2                           12
                 Timor-Leste                      Tunisia
                           1                            1
                      Turkey                           UK
                          16                          285
                     Ukraine          United Arab Emirates
                          18                           40
                         usa                          USA
                           1                          296
                     Vanuatu                    Venezuela
                           2                            2
                     Vietnam                     Zimbabwe
                           7                            1
> ico_main$countryRegion[nchar(ico_main$countryRegion) == 0] <- "unknown"
> table(ico_main$countryRegion)

                 Afghanistan                      Andorra
                           2                            1
                    Anguilla                    Argentina
                           3                            3
                     Armenia                    Australia
                           1                           57
                     Austria                      Bahamas
                          11                            3
                  Bangladesh                     Barbados
                           1                            1
                     Belarus                      Belgium
                           6                            4
                      Belize                      Bermuda
                          29                            4
      Bosnia and Herzegovina                       Brazil
                           1                           11
      British Virgin Islands                     Bulgaria
                          44                           19
                    Cambodia                      Canada
                           1                           51
             Cayman Islands                       Chile
                          67                            3
                       China                     Colombia
                          21                            2
                       Congo                   Costa Rica
                           1                            9
                     Croatia                     Curacao
                           6                            1
                     Curaçao                      Cyprus
                           1                           25
             Czech Republic                     Denmark
                          23                            4
         Dominican Republic                     Ecuador
                           1                            1
                       Egypt                     Estonia
                           1                          190
                     Finland                      France
                           3                           42
           French Polynesia                     Georgia
                           1                           16
                     Germany                       Ghana
                          61                            1
                   Gibraltar                      Greece
                          36                            5
              Guinea-Bissau                     Honduras
                           1                            1
```

Hungary	Iceland
4	2
india	India
1	38
Indonesia	Ireland
31	15
Isle of Man	Israel
4	16
Italy	Japan
12	17
Kazakhstan	Kuwait
2	1
Kyrgyzstan	Latvia
1	15
Liechtenstein	Lithuania
10	18
Luxembourg	Macedonia
6	4
Malaysia	Malta
16	58
Marshall Islands	Mauritius
5	7
Mexico	México
7	1
Monaco	Mongolia
1	2
Montenegro	Morocco
1	1
Netherlands	New Caledonia
59	1
New Zealand	Nigeria
8	24
Northern Mariana Islands	Norway
1	7
Pakistan	Panama
2	9
Peru	Philippines
2	14
Poland	Portugal
21	7
Puerto Rico	Romania
1	24
Russia	Saint Kitts and Nevis
137	11
Saint Vincent and the Grenadines	Samoa
1	1
Saudi Arabia	Serbia
1	9
Seychelles	Singapore
31	312
SINGAPORE	Slovakia
1	1
Slovenia	South Africa
24	24
South Korea	Spain
30	18
Sweden	Switzerland
7	138
Syria	Tanzania
1	2
Thailand	Timor-Leste
12	1
Tunisia	Turkey
1	16
UK	Ukraine
285	18
United Arab Emirates	unknown
40	71
usa	USA

```
                         1                          296
                  Vanuatu                    Venezuela
                         2                            2
                  Vietnam                     Zimbabwe
                         7                            1
```

Appendix – 25: Replacing names of country where they appear in different case:

```
> ico_main$countryRegion[ico_main$countryRegion == "india"]   <- "India"
> ico_main$countryRegion[ico_main$countryRegion == "México"]  <- "Mexico"
> ico_main$countryRegion[ico_main$countryRegion == "SINGAPORE"]  <- "Singa
pore"
> ico_main$countryRegion[ico_main$countryRegion == "usa"]  <- "USA"
> table(ico_main$countryRegion)

                     Afghanistan                         Andorra
                               2                               1
                        Anguilla                       Argentina
                               3                               3
                         Armenia                       Australia
                               1                              57
                         Austria                         Bahamas
                              11                               3
                      Bangladesh                        Barbados
                               1                               1
                         Belarus                         Belgium
                               6                               4
                          Belize                         Bermuda
                              29                               4
          Bosnia and Herzegovina                          Brazil
                               1                              11
          British Virgin Islands                        Bulgaria
                              44                              19
                        Cambodia                         Canada
                               1                              51
                   Cayman Islands                           Chile
                              67                               3
                           China                        Colombia
                              21                               2
                           Congo                      Costa Rica
                               1                               9
                         Croatia                         Curacao
                               6                               1
                         Curaçao                          Cyprus
                               1                              25
                  Czech Republic                         Denmark
                              23                               4
              Dominican Republic                         Ecuador
                               1                               1
                           Egypt                         Estonia
                               1                             190
                         Finland                          France
                               3                              42
                 French Polynesia                         Georgia
                               1                              16
                         Germany                           Ghana
                              61                               1
                       Gibraltar                          Greece
                              36                               5
                    Guinea-Bissau                        Honduras
                               1                               1
                         Hungary                         Iceland
                               4                               2
                           India                       Indonesia
                              39                              31
                         Ireland                     Isle of Man
                              15                               4
                          Israel                           Italy
                              16                              12
```

Japan	Kazakhstan
17	2
Kuwait	Kyrgyzstan
1	1
Latvia	Liechtenstein
15	10
Lithuania	Luxembourg
18	6
Macedonia	Malaysia
4	16
Malta	Marshall Islands
58	5
Mauritius	Mexico
7	8
Monaco	Mongolia
1	2
Montenegro	Morocco
1	1
Netherlands	New Caledonia
59	1
New Zealand	Nigeria
8	24
Northern Mariana Islands	Norway
1	7
Pakistan	Panama
2	9
Peru	Philippines
2	14
Poland	Portugal
21	7
Puerto Rico	Romania
1	24
Russia	Saint Kitts and Nevis
137	11
Saint Vincent and the Grenadines	Samoa
1	1
Saudi Arabia	Serbia
1	9
Seychelles	Singapore
31	313
Slovakia	Slovenia
1	24
South Africa	South Korea
24	30
Spain	Sweden
18	7
Switzerland	Syria
138	1
Tanzania	Thailand
2	12
Timor-Leste	Tunisia
1	1
Turkey	UK
16	285
Ukraine	United Arab Emirates
18	40
unknown	USA
71	297
Vanuatu	Venezuela
2	2
Vietnam	Zimbabwe
7	1

Appendix -26: Discrepancies in feature "platform":

Discrepancy info	Count		Discrepancy info	Count
Blank	6		_Ethereum	1
			ETH	1
_Komodo	1		Ethererum	2
Komodo	3		Ethereum	2343
			Ethereum_	23
Bitshares	2		Ethereum__	16
BitShares	1		Ethereum___	9
			Ethereum____	1
DPoS	3		Ethereum_____	2
DPOS	1		Etherum	1
Eos	1		Separate blockchain	30
EOS	16		Separate Blockchain	6
			Separate Blockchain_	1
Bitcoin	10		Separate Blockchain__	2
BTC	3			
			Stellar	41
Multichain	1		Stellar Protocol	1
MultiChain	1			
			Stratis	2
Nem	1		STRATIS	1
NEM	12			
			Tron	4
_NEO	1		TRON	3
Neo	1		Tron_	1
NEO	19		Tron____	1
Pivx	1		Waves	55
PivX	1		WAVES	1
X11	7		POS + POW	1
X11 blockchain	1		POS, POW	1
			pow/pos	1
X13	1		PoW/PoS	1
x13_	1			

NOTE: underscore(_) refers to space

Appendix – 27: Removing spaces with nothing in feature "platform":

```
> ico_main$platform <- gsub("\\s+","",ico_main$platform) # To remove space
s in names of platform and replace it with nothing
> ico_main$platform <- tolower(ico_main$platform) # To convert names of pl
atform to lower case
> table(ico_main$platform)
```

	komodo
6	1
acclaim	achain
1	1
aion	akroma
1	1
apolloblockchain	ardor
1	1
bep2	bitcoin
1	10
bitforex	bitshares
1	3
bitsmo	blockchain
1	1
btc	chainrepublikblockchain
3	1
coffe	coincart
1	1
counterparty	cryptokami
1	1
cryptonight	cryptonote-basedblockchain
1	1
dag	decoinblockchain
1	1
dpos	enlteplatform
4	1
entry	eos
1	17
erc20	eth
2	1
ethererum	ethereum
2	2395
ethereum,waves	etherum
1	1
fiber	filecoinnetwork
1	1
gochain	graphene
1	4
hard-forkoflitecoin	hybrid
1	1
hyperledger	icon
2	3
infinityblockchain	iloliteblockchain
1	1
iovblockchain	irongeekchain
1	1
isl-blockchain	jpmorganchase
1	1
keccak	komodo
1	3
lisk	litecoin
1	1
mahraplatform	monero
1	1
multichain	native
2	2
neblio	nem
1	13
neo	neurochain
21	1
newblockchain	newton
1	1
nilechain	nxt
1	3
pivx	pos
2	2
pos,pow	pos+pow
1	1

```
              pow                 pow/pos
                2                       2
              qrc                    qtum
                1                       2
        ripemd160                     rsk
                1                       1
           scrypt         separateblockchain
               14                      39
       sha256coin                slatechain
                1                       1
           smartx                    st20
                1                       1
       startengine                   steem
                1                       2
           stellar           stellarprotocol
               41                       1
          stratis                   tezos
                3                       1
        tomochain                     ton
                2                       1
             tron                  ttchain
                9                       1
         universa                   vasya
                1                       1
          vechain          vechainthorvip180
                2                       1
        ventureon                 wanchain
                1                       1
            waves                  wizebit
               56                       1
              x11             x11blockchain
                7                       1
              x13                     xaya
                2                       1
             xdac                  youtoken
                1                       1
          zilliqa                     zuum
                1                       1
```

Appendix – 28: Bringing uniformity in names of platform in feature "platform":

```
> ico_main$platform[ico_main$platform == "btc"] <- "bitcoin"
> ico_main$platform[ico_main$platform == "stellarprotocol"] <- "stellar"
> ico_main$platform[ico_main$platform == "x11blockchain"] <- "x11"
> ico_main$platform[ico_main$platform == "eth"] <- "ethereum"
> ico_main$platform[ico_main$platform == "ethererum"] <- "ethereum"
> ico_main$platform[ico_main$platform == "etherum"] <- "ethereum"
> ico_main$platform[ico_main$platform == "pos,pow"] <- "pos+pow"
> ico_main$platform[ico_main$platform == "pow/pos"] <- "pos+pow"
> ico_main$platform[nchar(ico_main$platform) == 0] <- "unknown"
> table(ico_main$platform)

              komodo                 acclaim
                   1                       1
              achain                    aion
                   1                       1
              akroma          apolloblockchain
                   1                       1
               ardor                    bep2
                   1                       1
             bitcoin                bitforex
                  13                       1
           bitshares                  bitsmo
                   3                       1
          blockchain chainrepublikblockchain
                   1                       1
               coffe                coincart
                   1                       1
        counterparty              cryptokami
```

1	1
cryptonight	cryptonote-basedblockchain
1	1
dag	decoinblockchain
1	1
dpos	enlteplatform
4	1
entry	eos
1	17
erc20	ethereum
2	2399
ethereum,waves	fiber
1	1
filecoinnetwork	gochain
1	1
graphene	hard-forkoflitecoin
4	1
hybrid	hyperledger
1	2
icon	infinityblockchain
3	1
ioliteblockchain	iovblockchain
1	1
irongeekchain	isl-blockchain
1	1
jpmorganchase	keccak
1	1
komodo	lisk
3	1
litecoin	mahraplatform
1	1
monero	multichain
1	2
native	neblio
2	1
nem	neo
13	21
neurochain	newblockchain
1	1
newton	nilechain
1	1
nxt	pivx
3	2
pos	pos+pow
2	4
pow	qrc
2	1
qtum	ripemd160
2	1
rsk	scrypt
1	14
separateblockchain	sha256coin
39	1
slatechain	smartx
1	1
st20	startengine
1	1
steem	stellar
2	42
stratis	tezos
3	1
tomochain	ton
2	1
tron	ttchain
9	1
universa	unknown
1	6
vasya	vechain
1	2

```
vechainthorvip180                    ventureon
                1                            1
         wanchain                        waves
                1                           56
          wizebit                          x11
                1                            8
              x13                          xaya
                2                            1
             xdac                       youtoken
                1                            1
          zilliqa                         zuum
                1                            1
```

Appendix – 29: Creating new feature "Duration_of_campaign":

```
> ico_main[,"startDate"] <- as.Date(ico_main[,"startDate"],format = "%d/%m
/%Y")
> ico_main[,"endDate"] <- as.Date(ico_main[,"endDate"],format = "%d/%m/%Y"
)
> ico_main$Duration_of_campaign <- difftime(ico_main$endDate,ico_main$star
tDate, units = "days")
> str(ico_main$Duration_of_campaign)
 'difftime' num [1:2754] 0 35 365 75 ...
 - attr(*, "units")= chr "days"
```

Appendix – 30: Feature selection:

```
> ico_main <- ico_main[, - c(1,3,8,9)]
> str(ico_main)
'data.frame':  2754 obs. of  13 variables:
 $ success               : chr  "N" "N" "N" "Y" ...
 $ hasVideo              : int  1 1 1 1 1 1 1 1 1 1 ...
 $ rating                : num  4 4.3 4.4 4.3 4.3 4.7 4.1 4.5 4.8 4.2 ...
 $ priceUSD              : num  30 0.13 0.01 0.12 0.03 0.1 0.02 2.8 50 0.1
...
 $ countryRegion         : chr  "Singapore" "Malta" "UK" "Netherlands" ...
 $ teamSize              : num  31 20 10 27 14 43 20 31 8 29 ...
 $ hasGithub             : int  1 1 1 1 1 1 1 1 1 1 ...
 $ hasReddit             : int  1 1 1 1 1 1 1 1 1 1 ...
 $ platform              : chr  "ethereum" "xaya" "stellar" "separateblockc
hain" ...
 $ coinNum               : num  5.10e+05 2.25e+08 5.00e+09 1.25e+08 5.00e+0
9 ...
 $ minInvestment         : int  0 1 1 1 1 1 1 1 1 1 ...
 $ distributedPercentage: num  0.49 0.41 0.4 0.13 0.5 0.5 0.25 0.1 0.05 0.
15 ...
 $ Duration_of_campaign : 'difftime' num  0 35 365 75 ...
  ..- attr(*, "units")= chr "days"
```

Appendix – 31: Decision Tree model:

Code for Decision Tree Model with AUC and key metrics:

```
ico_dt <- ico_main
#str(ico_main)
#Converting character features to factors
ico_dt[,c("success", "countryRegion", "platform")] <- lapply(ico_dt[,c("su
ccess", "countryRegion", "platform")], factor)
#str(ico_dt)
#Separating Training and Test data
smp_size <- floor(0.9 * nrow(ico_dt))
set.seed(12345)
train_ind <- sample(nrow(ico_dt), smp_size)
ico_dt_train <- ico_dt[train_ind, ]
ico_dt_test <- ico_dt[-train_ind, ]
ico_dt_test_labels <- ico_dt[-train_ind, "success"]
ico_dt_test_labels
```

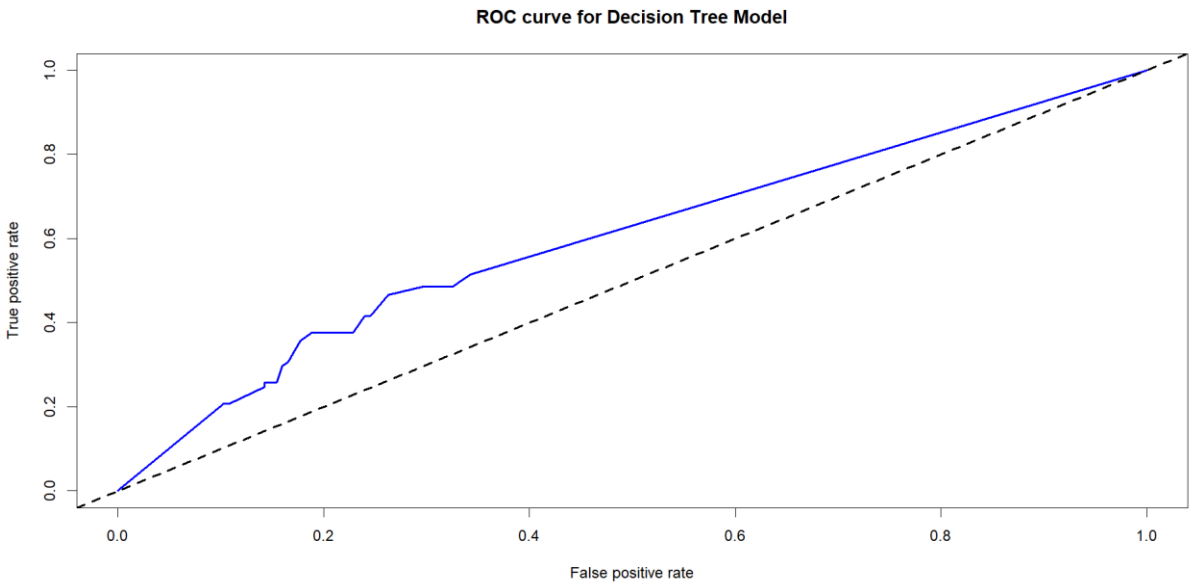
```
# Checking the distribution of classes in training and testing data
prop.table(table(ico_dt_train$success))
prop.table(table(ico_dt_test $success))
#Training the model
library(C50)
library(tidyverse)
dt_model <- C5.0(success ~ ., ico_dt_train, rules = TRUE)
dt_model
summary(dt_model)
#Evaluating the performance of DT model
dt_pred <- predict(dt_model, ico_dt_test, type = "prob" )
dt_pred1 <- predict(dt_model, ico_dt_test)
dt_pred
#Developing ROC Curve
library(ROCR)
predict_object <- prediction(dt_pred[,2],ico_dt_test_labels)
roc_DT <- performance(predict_object, measure = "tpr", x.measure = "fpr")
plot(roc_DT, main = "ROC curve for Decision Tree Model", col = "blue", lwd
= 2)
abline(a = 0, b = 1, lwd = 2, lty = 2)
auc_object_DT <- performance(predict_object, measure = "auc")
auc_object_DT
auc_object_DT@y.values[[1]]
#Getting other metrics Accuracy, Sensitivity, Specificity, Precision, F-me
asure
library(caret)
?confusionMatrix
keymetric_DT <- confusionMatrix(dt_pred1 , ico_dt_test_labels, positive =
"Y", mode = "everything")
keymetric_DT
```

Output:

```
> summary(dt_model)

Call:
C5.0.formula(formula = success ~ ., data = ico_dt_train, rules = TRUE)


C5.0 [Release 2.07 GPL Edition]       Tue Apr 18 18:10:49 2023
-------------------------------

Class specified by attribute `outcome'

Read 2478 cases (13 attributes) from undefined.data

Rules:

Rule 1: (231/53, lift 1.2)
        rating <= 3.9
        countryRegion = USA
        ->  class N  [0.768]

Rule 2: (1227/297, lift 1.2)
        rating <= 3
        ->  class N  [0.758]

Rule 3: (882/219, lift 1.2)
        rating <= 3.9
        hasReddit <= 0
        ->  class N  [0.751]

Rule 4: (1487/403, lift 1.2)
        rating <= 3.3
        ->  class N  [0.729]

Rule 5: (1519/423, lift 1.1)
        teamSize <= 13
        ->  class N  [0.721]
```

```
Rule 6: (1196/372, lift 1.1)
        rating <= 3.9
        Duration_of_campaign > 35
        ->  class N  [0.689]

Rule 7: (7, lift 2.4)
        rating > 3.3
        rating <= 3.9
        countryRegion = UK
        teamSize > 13
        coinNum > 5.25e+08
        ->  class Y  [0.889]

Rule 8: (277/92, lift 1.8)
        rating > 3
        countryRegion in {Austria, Bahamas, Bermuda, Bulgaria, China, Cypru
s,
                          Czech Republic, Dominican Republic, Egypt, Eston
ia,
                          Georgia, Greece, India, Ireland, Japan, Latvia,
                          Liechtenstein, Lithuania, Macedonia, Malta, Moro
cco,
                          Nigeria, Northern Mariana Islands, Norway, Panam
a,
                          Peru, Samoa, Seychelles, Singapore, South Africa
,
                          Spain, Sweden, Thailand, Ukraine, unknown}
        teamSize > 13
        ->  class Y  [0.667]

Rule 9: (53/18, lift 1.8)
        teamSize > 33
        ->  class Y  [0.655]

Rule 10: (959/460, lift 1.4)
        teamSize > 13
        ->  class Y  [0.520]

Default class: N


Evaluation on training data (2478 cases):

            Rules
      ----------------
        No      Errors

        10  756(30.5%)   <<


       (a)   (b)    <-classified as
      ----  ----
      1403   153    (a): class N
       603   319    (b): class Y


      Attribute usage:

      100.00% teamSize
       83.66% rating
       48.26% Duration_of_campaign
       35.59% hasReddit
       20.78% countryRegion
        0.28% coinNum


Time: 0.0 secs
```

```
> auc_object_DT
A performance instance
  'Area under the ROC curve'
> auc_object_DT@y.values[[1]]
[1] 0.5998868

> keymetric_DT
Confusion Matrix and Statistics

          Reference
Prediction   N   Y
         N 149  75
         Y  26  26

               Accuracy : 0.6341
                 95% CI : (0.5742, 0.691)
    No Information Rate : 0.6341
    P-Value [Acc > NIR] : 0.5271

                  Kappa : 0.1213

 Mcnemar's Test P-Value : 1.787e-06

            Sensitivity : 0.2574
            Specificity : 0.8514
         Pos Pred Value : 0.5000
         Neg Pred Value : 0.6652
              Precision : 0.5000
                 Recall : 0.2574
                     F1 : 0.3399
             Prevalence : 0.3659
         Detection Rate : 0.0942
   Detection Prevalence : 0.1884
      Balanced Accuracy : 0.5544

       'Positive' Class : Y
```

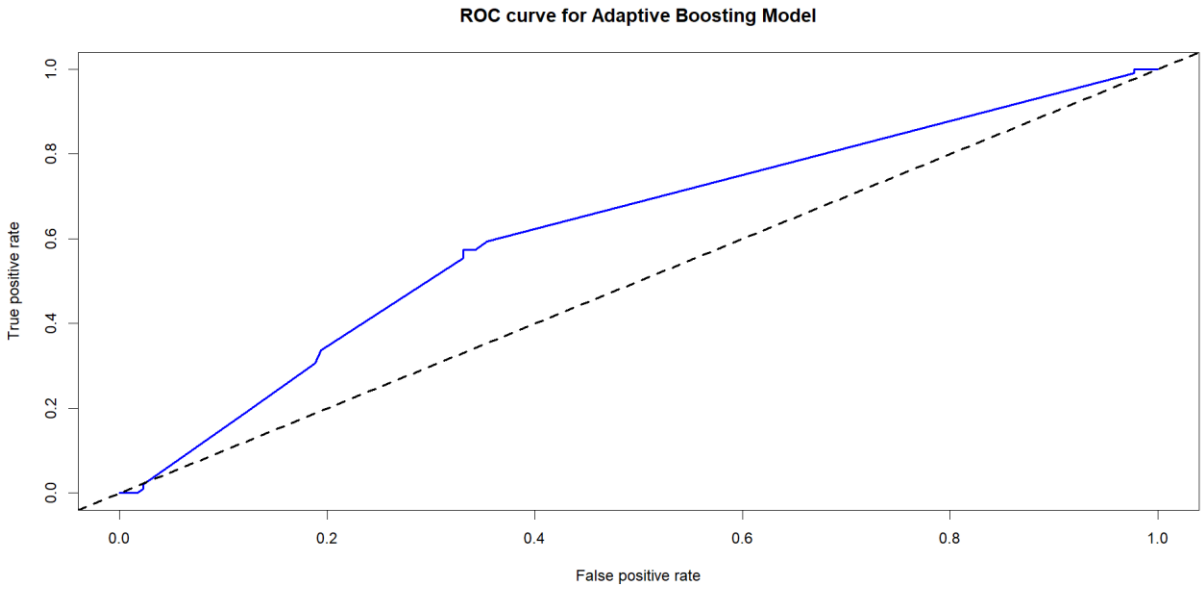


Appendix – 32: AdaBoost Model including key metrics:

```
> #Model based on Adaptive Boosting
> dt_boost <- C5.0(select(ico_dt_train, -success), ico_dt_train$success, t
rails = 10)
> dt_boost

Call:
C5.0.default(x = select(ico_dt_train, -success), y =
 ico_dt_train$success, trails = 10)

Classification Tree
Number of samples: 2478
Number of predictors: 12

Tree size: 37

Non-standard options: attempt to group attributes

> summary(dt_boost)

Call:
C5.0.default(x = select(ico_dt_train, -success), y =
 ico_dt_train$success, trails = 10)


C5.0 [Release 2.07 GPL Edition]       Thu May 18 08:03:06 2023
-------------------------------

Class specified by attribute `outcome'

Read 2478 cases (13 attributes) from undefined.data

Decision tree:

rating <= 3: N (1227/297)
rating > 3:
:...teamSize <= 13: N (544/215)
    teamSize > 13:
    :...rating > 3.9: Y (265/82)
        rating <= 3.9:
        :...countryRegion in {Anguilla,Australia,Belarus,Belgium,Belize,Brazil,
            :                 Costa Rica,Denmark,Germany,Hungary,Indonesia,
            :                 Isle of Man,Israel,Kyrgyzstan,Luxembourg,
            :                 Malaysia,Marshall Islands,Mauritius,Mexico,
            :                 New Zealand,Poland,Portugal,Romania,
            :                 Saint Kitts and Nevis,Serbia,South Korea,
            :                 Turkey}: N (73/19)
            countryRegion in {Afghanistan,Andorra,Argentina,Armenia,Austria,
            :                 Bahamas,Bangladesh,Barbados,Bermuda,
            :                 Bosnia and Herzegovina,Bulgaria,Cambodia,Chile,
            :                 China,Colombia,Congo,Croatia,Curacao,Cyprus,
            :                 Czech Republic,Dominican Republic,Ecuador,Egypt,
            :                 Estonia,Finland,French Polynesia,Georgia,Ghana,
            :                 Greece,Guinea-Bissau,Honduras,Iceland,India,
            :                 Ireland,Italy,Japan,Kazakhstan,Kuwait,Latvia,
            :                 Liechtenstein,Lithuania,Macedonia,Malta,Monaco,
            :                 Mongolia,Montenegro,Morocco,New Caledonia,
            :                 Nigeria,Northern Mariana Islands,Norway,Pakistan,
            :                 Panama,Peru,Philippines,Puerto Rico,
            :                 Saint Vincent and the Grenadines,Samoa,
            :                 Saudi Arabia,Seychelles,Singapore,Slovakia,
            :                 South Africa,Spain,Sweden,Syria,Tanzania,
            :                 Thailand,Timor-Leste,Tunisia,Ukraine,unknown,
            :                 Vanuatu,Venezuela,Vietnam,
            :                 Zimbabwe}: Y (154/51)
            countryRegion = Canada:
            :...Duration_of_campaign <= 30: Y (5/1)
            :   Duration_of_campaign > 30: N (4)
            countryRegion = Cayman Islands:
            :...Duration_of_campaign <= 48: Y (9/2)
            :   Duration_of_campaign > 48: N (5)
            countryRegion = Gibraltar:
```

```
:...coinNum <= 1.04835e+09: N (9/3)
:   coinNum > 1.04835e+09: Y (4)
countryRegion = Netherlands:
:...distributedPercentage <= 0.62: N (4/1)
:   distributedPercentage > 0.62: Y (2)
countryRegion = Russia:
:...rating <= 3.6: N (17/6)
:   rating > 3.6: Y (3)
countryRegion = Slovenia:
:...coinNum <= 1.1538e+08: N (3)
:   coinNum > 1.1538e+08: Y (6)
countryRegion = United Arab Emirates:
:...teamSize <= 33: N (3)
:   teamSize > 33: Y (2)
countryRegion = British Virgin Islands:
:...hasReddit <= 0: Y (2)
:   hasReddit > 0:
:   :...rating <= 3.6: N (5)
:       rating > 3.6: Y (9/3)
countryRegion = France:
:...hasGithub <= 0: N (3)
:   hasGithub > 0:
:   :...distributedPercentage <= 0.75: Y (4)
:       distributedPercentage > 0.75: N (3)
countryRegion = UK:
:...rating <= 3.3: N (6)
:   rating > 3.3:
:   :...coinNum <= 5.25e+08: N (22/9)
:       coinNum > 5.25e+08: Y (7)
countryRegion = USA:
:...hasVideo <= 0: Y (8/2)
:   hasVideo > 0:
:   :...hasReddit > 0: N (25/7)
:       hasReddit <= 0:
:       :...priceUSD <= 0.34: Y (4)
:           priceUSD > 0.34: N (2)
countryRegion = Switzerland:
:...hasReddit <= 0: N (7/1)
    hasReddit > 0:
    :...Duration_of_campaign <= 35: Y (15/2)
        Duration_of_campaign > 35:
        :...hasGithub <= 0: Y (5/1)
            hasGithub > 0:
            :...priceUSD <= 0.02: Y (2)
                priceUSD > 0.02: N (10/2)


Evaluation on training data (2478 cases):

          Decision Tree
        ----------------
        Size      Errors

          37  704(28.4%)   <<


         (a)   (b)    <-classified as
        ----  ----
        1412   144    (a): class N
         560   362    (b): class Y


       Attribute usage:

       100.00% rating
        50.48% teamSize
        17.84% countryRegion
         3.47% hasReddit
         2.22% Duration_of_campaign
         2.06% coinNum
         1.57% hasVideo
         1.09% hasGithub
         0.73% priceUSD
         0.52% distributedPercentage
```

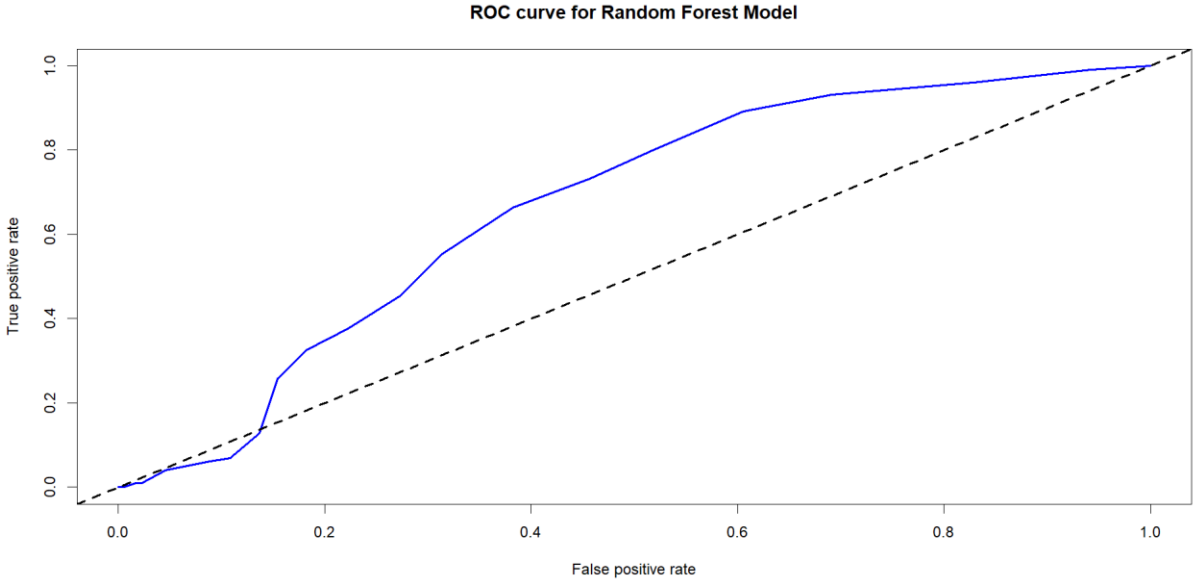
```
Time: 0.0 secs


>
> dt_boost_pred <- predict(dt_boost, ico_dt_test, type = "prob")
> dt_boost_pred1 <- predict(dt_boost, ico_dt_test )
>
> #library(ROCR)
> predict_object_boost<- prediction(dt_boost_pred[,2],ico_dt_test_labels)
> roc_DT_boost <- performance(predict_object_boost, measure = "tpr", x.mea
sure = "fpr")
> plot(roc_DT_boost, main = "ROC curve for Adaptive Boosting Model", col =
"blue", lwd = 2)
> abline(a = 0, b = 1, lwd = 2, lty = 2)
> auc_object_boost <- performance(predict_object_boost, measure = "auc")
> auc_object_boost
A performance instance
  'Area under the ROC curve'
> auc_object_boost@y.values[[1]]
[1] 0.6194908
> #Getting other metrics Accuracy, Sensitivity, Specificity, Precision, F-
measure
> #library(caret)
>
> keymetric_Ada <- confusionMatrix(dt_boost_pred1 , ico_dt_test_labels, po
sitive = "Y", mode = "everything")
> keymetric_Ada
Confusion Matrix and Statistics

          Reference
Prediction   N   Y
         N 142  70
         Y  33  31

               Accuracy : 0.6268
                 95% CI : (0.5668, 0.684)
    No Information Rate : 0.6341
    P-Value [Acc > NIR] : 0.6245144

                  Kappa : 0.1283

 Mcnemar's Test P-Value : 0.0003894

            Sensitivity : 0.3069
            Specificity : 0.8114
         Pos Pred Value : 0.4844
         Neg Pred Value : 0.6698
              Precision : 0.4844
                 Recall : 0.3069
                     F1 : 0.3758
             Prevalence : 0.3659
         Detection Rate : 0.1123
   Detection Prevalence : 0.2319
      Balanced Accuracy : 0.5592

       'Positive' Class : Y
```



Appendix 33: Random Forest with key metric:

```
> ico_RF <- randomForest(success ~ ., data = ico_dt_train, ntree = 20)
Error in randomForest.default(m, y, ...) :
  Can not handle categorical predictors with more than 53 categories.

> ico_RF <- ico_main # Creating a duplicate dataframe
> library(caret)
> ico_RF_dummy <- dummyVars( ~ countryRegion + platform , data = ico_RF) #
Using dummyVars for one hot encoding
> dummy_frame <- data.frame(predict(ico_RF_dummy, ico_RF))# Creating a dat
aframe of dummy features
> library(dplyr)
> ico_RF <- cbind(ico_RF, dummy_frame) #combining dummy dataframe with RF
dataframe
> ico_RF$countryRegion <- NULL # removing country feature from final dataf
rame
> ico_RF$platform <- NULL # removing platform feature from final dataframe
> ico_RF$success <- factor(ico_RF$success) #Converting character features
to factors
> #Separating Training and Test data
> smp_size_RF <- floor(0.9 * nrow(ico_RF))
> set.seed(12345)
> train_ind_RF <- sample(nrow(ico_RF), smp_size)
> ico_dt_train_RF <- ico_RF[train_ind_RF, ]
> ico_dt_test_RF <- ico_RF[-train_ind_RF, ]
> ico_dt_test_labels_RF <- ico_RF[-train_ind_RF, "success"]
> #Training the model
> library(randomForest)
> ico_RF_model <- randomForest(success ~ ., data = ico_dt_train_RF, ntree
= 20)
> #Evaluating the model on test data
> ico_RF_predict <- predict(ico_RF_model, ico_dt_test_RF, type = "prob" )
> ico_RF_predict1 <- predict(ico_RF_model, ico_dt_test_RF)
> #Getting key metrics
> library(ROCR)
> predict_object_RF <- prediction(ico_RF_predict[,2],ico_dt_test_labels_RF
)
> roc_RF <- performance(predict_object_RF, measure = "tpr", x.measure = "f
pr")
> plot(roc_RF, main = "ROC curve for Random Forest Model", col = "blue", l
wd = 2)
> abline(a = 0, b = 1, lwd = 2, lty = 2)
> auc_object_RF <- performance(predict_object_RF, measure = "auc")
> auc_object_RF
```

```
A performance instance
  'Area under the ROC curve'
> auc_object_RF@y.values[[1]]
[1] 0.6647242
> #Getting other metrics Accuracy, Sensitivity, Specificity, Precision, F-measure
> library(caret)
> keymetric_RF <- confusionMatrix(ico_RF_predict1 , ico_dt_test_labels_RF, positive = "Y", mode = "everything")
> keymetric_RF
Confusion Matrix and Statistics

          Reference
Prediction   N   Y
         N 143  67
         Y  32  34

               Accuracy : 0.6413
                 95% CI : (0.5816, 0.6979)
    No Information Rate : 0.6341
    P-Value [Acc > NIR] : 0.4277978

                  Kappa : 0.1659

 Mcnemar's Test P-Value : 0.0006329

            Sensitivity : 0.3366
            Specificity : 0.8171
         Pos Pred Value : 0.5152
         Neg Pred Value : 0.6810
              Precision : 0.5152
                 Recall : 0.3366
                     F1 : 0.4072
             Prevalence : 0.3659
         Detection Rate : 0.1232
   Detection Prevalence : 0.2391
      Balanced Accuracy : 0.5769

       'Positive' Class : Y
```



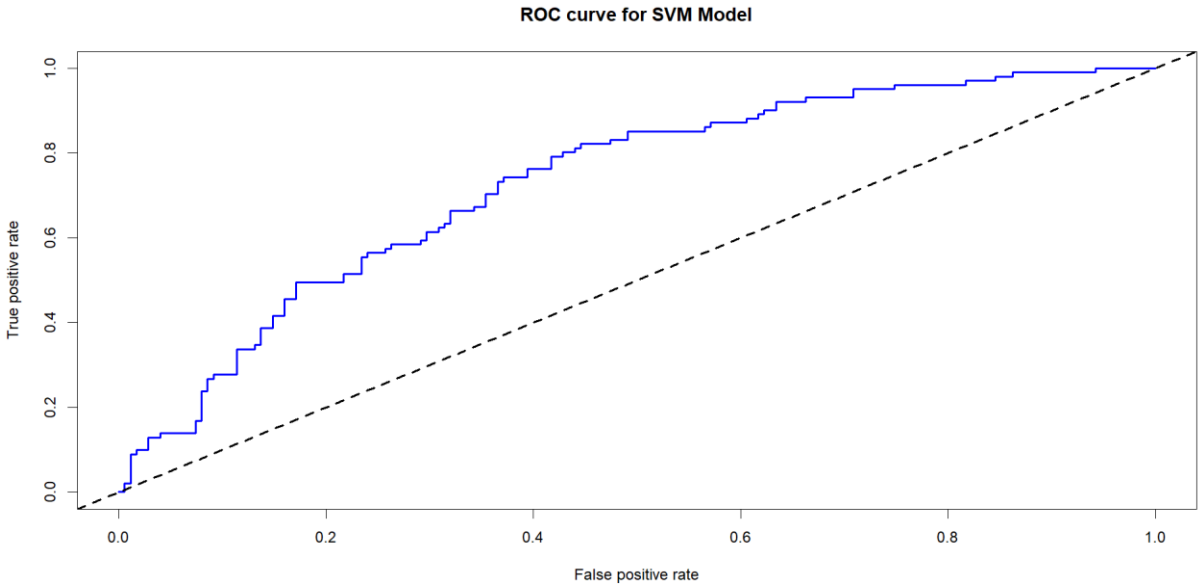

```
> ico_RF_model

Call:
 randomForest(formula = success ~ ., data = ico_dt_train_RF, ntree = 20)
               Type of random forest: classification
                     Number of trees: 20
No. of variables tried at each split: 14
```

```
        OOB estimate of  error rate: 34.79%
Confusion matrix:
     N   Y class.error
N 1277 279   0.1793059
Y  583 339   0.6323210
```

Appendix – 34: R code for SVM:

```
> ico_SVM <- ico_RF
> #Partitioning the data into training and testing data
> smp_size_SVM <- floor(0.9 * nrow(ico_SVM))
> set.seed(12345)
> train_ind_SVM <- sample(nrow(ico_SVM), smp_size_SVM)
> ico_dt_train_SVM <- ico_SVM[train_ind_SVM, ]
> ico_dt_test_SVM <- ico_SVM[-train_ind_SVM, ]
> ico_dt_test_labels_SVM <- ico_SVM[-train_ind_SVM, "success"]
> # Training the model
> library(kernlab)
> SVM_model <- ksvm(success ~ ., data = ico_SVM,
+                           kernel = "vanilladot", prob.model = TRUE)
 Setting default kernel parameters
> SVM_predict <- predict(SVM_model, select(ico_dt_test_SVM, -success), typ
e = "probabilities")
> SVM_predict1 <- predict(SVM_model, select(ico_dt_test_SVM, -success))
> #Getting key metrics
> library(ROCR)
> predict_object_SVM <- prediction(SVM_predict[,2],ico_dt_test_labels_SVM)
> roc_SVM <- performance(predict_object_SVM, measure = "tpr", x.measure =
"fpr")
> plot(roc_SVM, main = "ROC curve for SVM Model", col = "blue", lwd = 2)
> abline(a = 0, b = 1, lwd = 2, lty = 2)
> auc_object_SVM <- performance(predict_object_SVM, measure = "auc")
> auc_object_SVM
A performance instance
  'Area under the ROC curve'
> auc_object_SVM@y.values[[1]]
[1] 0.7293352
> #Getting other metrics Accuracy, Sensitivity, Specificity, Precision, F-
measure
> library(caret)
> keymetric_SVM <- confusionMatrix(SVM_predict1 , ico_dt_test_labels_SVM,
positive = "Y", mode = "everything")
> keymetric_SVM
Confusion Matrix and Statistics

          Reference
Prediction   N   Y
         N 147  59
         Y  28  42

               Accuracy : 0.6848
                 95% CI : (0.6264, 0.7392)
    No Information Rate : 0.6341
    P-Value [Acc > NIR] : 0.044695

                  Kappa : 0.2736

 Mcnemar's Test P-Value : 0.001298

            Sensitivity : 0.4158
            Specificity : 0.8400
         Pos Pred Value : 0.6000
         Neg Pred Value : 0.7136
              Precision : 0.6000
                 Recall : 0.4158
```

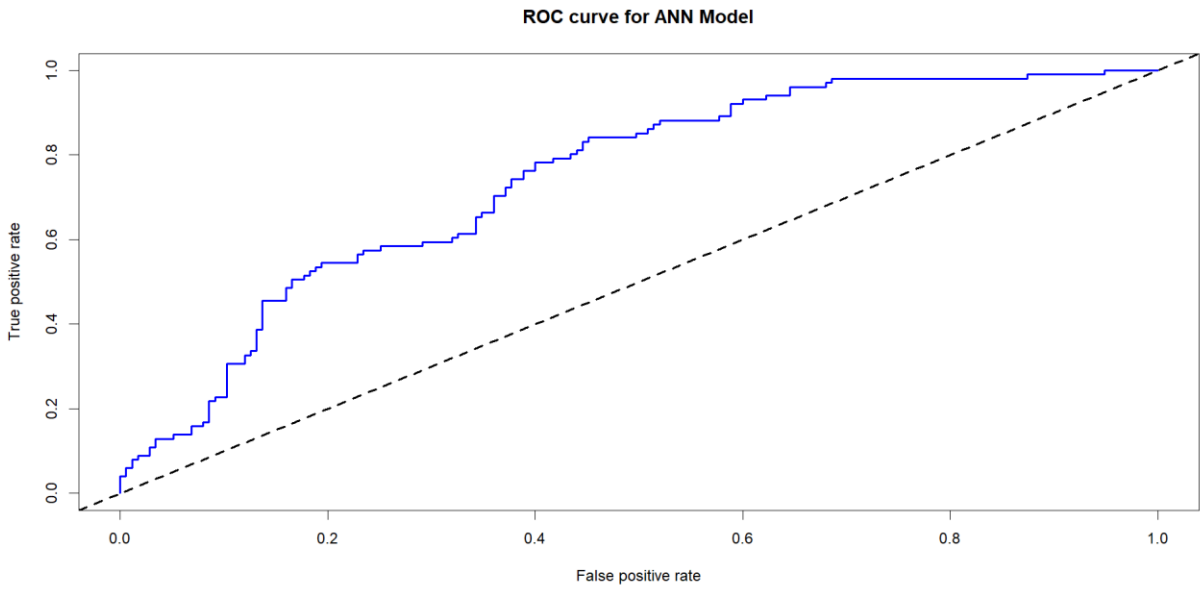
```
                     F1 : 0.4912
            Prevalence : 0.3659
        Detection Rate : 0.1522
  Detection Prevalence : 0.2536
     Balanced Accuracy : 0.6279

      'Positive' Class : Y
```



Appendix – 35: R code for ANN

```
> ico_ANN <- ico_RF
> #normalize complete data frame
> normalize <- function(x) {
+   return((x - min(x)) / (max(x) - min(x)))
+ }
> # apply normalization to entire data frame
> library(dplyr)
> ico_ANN_norm <- as.data.frame(lapply(select(ico_ANN, -success) , normali
ze)) # 'normalize' is the name of function we defined
> ico_ANN_norm <- cbind(ico_ANN_norm, success = ico_ANN$success) #combinin
g "success" feature
> #summary(ico_ANN_norm)
> #Partitioning the data into training and testing data
> smp_size_ANN <- floor(0.9 * nrow(ico_ANN_norm))
> set.seed(12345)
> train_ind_ANN <- sample(nrow(ico_ANN_norm), smp_size_ANN)
> ico_dt_train_ANN <- ico_ANN_norm[train_ind_ANN, ]
> ico_dt_test_ANN <- ico_ANN_norm[-train_ind_ANN, ]
> ico_dt_test_labels_ANN <- ico_ANN_norm[-train_ind_ANN, "success"]
> library(neuralnet)
> set.seed(12345)
> ANN_model <- neuralnet(formula = success ~ .,data = ico_ANN_norm)
> #plot(ANN_model)
> #summary(ANN_model)
> #Evaluating the model on test data
> ANN_predict <- predict(ANN_model, select(ico_dt_test_ANN, -success))
> #ANN gives prediction probabilities; Converting them to class labels
> ANN_predict_df <- data.frame(ANN_predict) #Converting prediciton output
as a dataframe
> class_labels_ANN_test <- ifelse(ANN_predict_df$X2 > 0.5, "Y", "N") #Conv
erting probabilities to class labels
> class_labels_ANN_test <- factor(class_labels_ANN_test)
>
> #Getting key metrics
> install.packages("ROCR")
```

```
Error in install.packages : Updating loaded packages
> library(ROCR)
> predict_object_ANN <- prediction(ANN_predict_df[,2],ico_dt_test_labels_A
NN)
> roc_ANN <- performance(predict_object_ANN, measure = "tpr", x.measure =
"fpr")
> plot(roc_ANN, main = "ROC curve for ANN Model", col = "blue", lwd = 2)
> abline(a = 0, b = 1, lwd = 2, lty = 2)
> auc_object_ANN <- performance(predict_object_ANN, measure = "auc")
> auc_object_ANN
A performance instance
  'Area under the ROC curve'
> auc_object_ANN@y.values[[1]]
[1] 0.7409901
> #Getting other metrics Accuracy, Sensitivity, Specificity, Precision, F-
measure
> library(caret)
Loading required package: ggplot2
Loading required package: lattice
Warning messages:
1: package 'caret' was built under R version 4.2.3
2: package 'ggplot2' was built under R version 4.2.2
> keymetric_ANN <- confusionMatrix(class_labels_ANN_test , ico_dt_test_lab
els_ANN, positive = "Y", mode = "everything")
> keymetric_ANN
Confusion Matrix and Statistics

          Reference
Prediction   N   Y
         N 146  52
         Y  29  49

               Accuracy : 0.7065
                 95% CI : (0.649, 0.7596)
    No Information Rate : 0.6341
    P-Value [Acc > NIR] : 0.006766

                  Kappa : 0.3356

 Mcnemar's Test P-Value : 0.014508

            Sensitivity : 0.4851
            Specificity : 0.8343
         Pos Pred Value : 0.6282
         Neg Pred Value : 0.7374
              Precision : 0.6282
                 Recall : 0.4851
                     F1 : 0.5475
             Prevalence : 0.3659
         Detection Rate : 0.1775
   Detection Prevalence : 0.2826
      Balanced Accuracy : 0.6597

       'Positive' Class : Y
```


ROC curve for ANN Model
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0