INDRODUCTION

In the digital age, crowd funding emerging as a cornerstone of this paradigm shift. By leveraging digital platforms, new entrepreneurs and innovators can seek expand their domain of crowd funding initiatives. A company needs capital, capital can be obtained by different ways loans, venture capitals, crowd funding are some of the methods any investment can be raised. Crowd funding has emerged as a popular method for fundraising, allowing individuals and companies to obtain funds for their projects or ventures from a large number of contributors, typically over the Internet. While traditional crowd funding involves raising funds in exchange for monetary compensation, initial coin offerings (ICOs) represent a novel approach in which companies raise funding by issuing and selling digital coins based on block chain technology.

The success of an ICO campaign is crucial for fundraising teams or companies, as it determines whether they can achieve their funding goals. Understanding the factors that influence the success of ICO campaigns is essential for both investors and fundraising teams to make informed decisions.

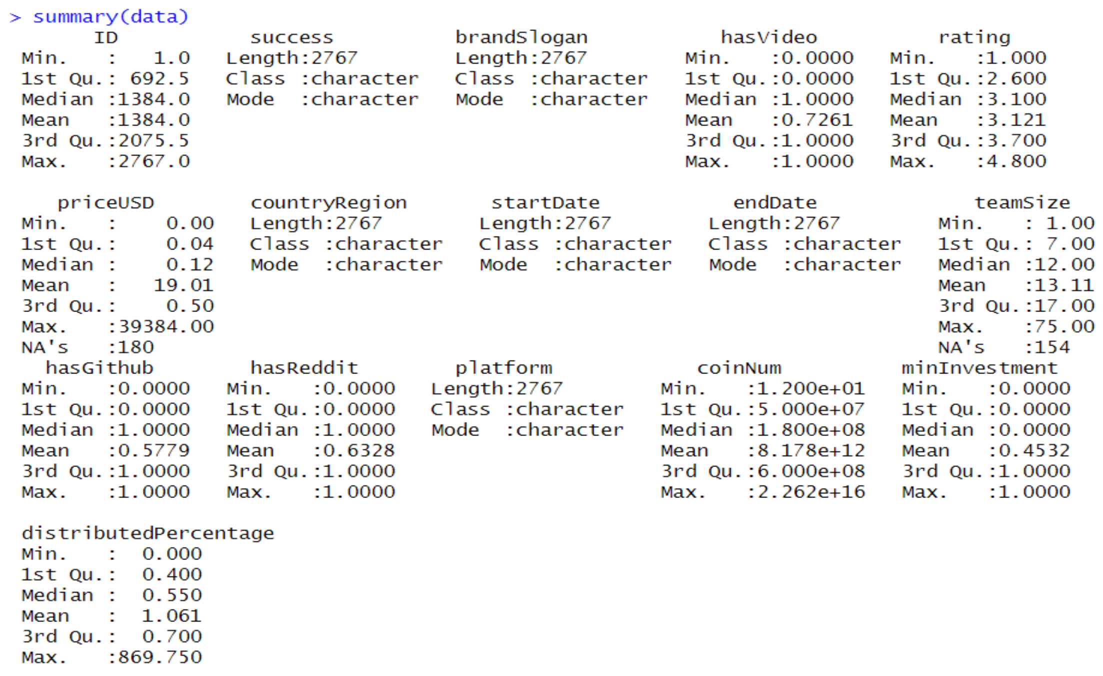
In this technical report, we aim to predict the success of ICO campaigns using various machine learning models. We have been provided with a dataset containing attributes of ICO projects from different fundraising teams or companies. These attributes include variables such as the number of blockchain coins to be issued, the price of each coin, the duration of the fundraising campaign, the team size, and various indicators related to the campaign's online presence and promotional activities.

Our task is to analyze the dataset, preprocess the data, and apply different classification techniques to predict whether a fundraising project will reach its funding goal successfully. We will evaluate the performance of the models using robust measures and provide insights into the key factors influencing the success of ICO campaigns.

1. DATA UNDERSTANDING:

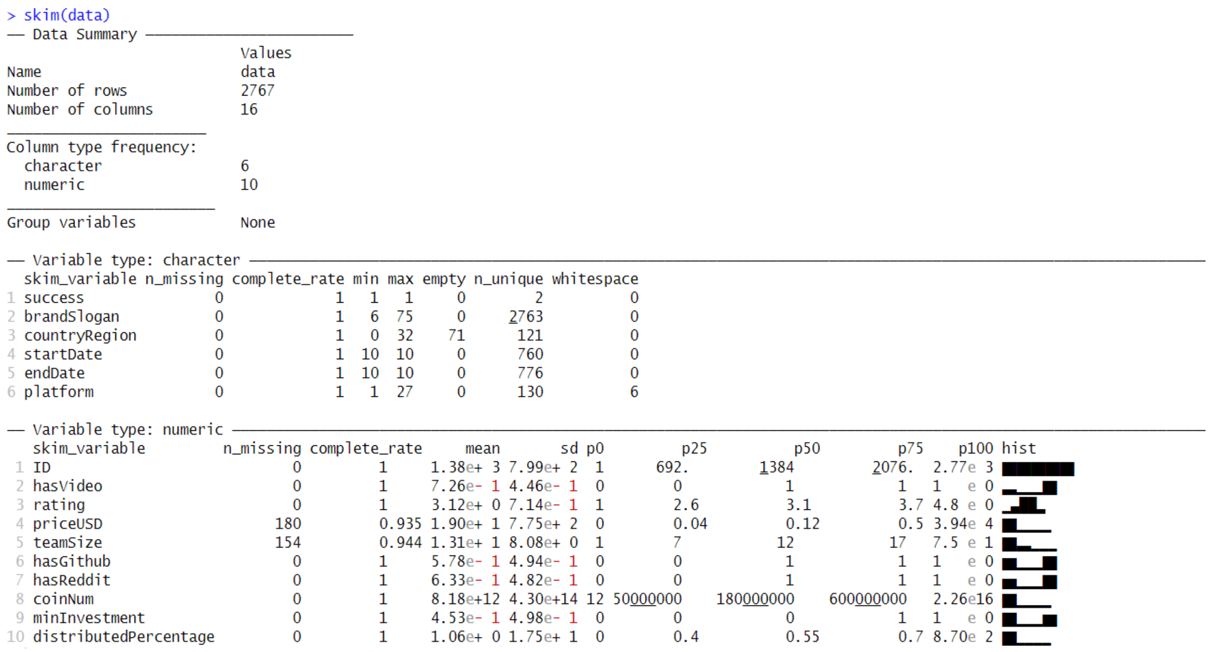
Overview of the Dataset

In this data, there are 2767 observations and 16 variables or features. These variables provide information about different aspects or attributes of the data points. Here the function summary() is used to get an overview of the data.



```
> summary(data)
       ID            success            brandSlogan           hasVideo          rating     
 Min.   :   1.0   Length:2767        Length:2767        Min.   :0.0000   Min.   :1.000  
 1st Qu.: 692.5   Class :character   Class :character   1st Qu.:0.0000   1st Qu.:2.600  
 Median :1384.0   Mode  :character   Mode  :character   Median :1.0000   Median :3.100  
 Mean   :1384.0                                         Mean   :0.7261   Mean   :3.121  
 3rd Qu.:2075.5                                         3rd Qu.:1.0000   3rd Qu.:3.700  
 Max.   :2767.0                                         Max.   :1.0000   Max.   :4.800  

    priceUSD         countryRegion        startDate            endDate             teamSize    
 Min.   :    0.00   Length:2767        Length:2767        Length:2767        Min.   : 1.00  
 1st Qu.:    0.04   Class :character   Class :character   Class :character   1st Qu.: 7.00  
 Median :    0.12   Mode  :character   Mode  :character   Mode  :character   Median :12.00  
 Mean   :   19.01                                                            Mean   :13.11  
 3rd Qu.:    0.50                                                            3rd Qu.:17.00  
 Max.   :39384.00                                                            Max.   :75.00  
 NA's   :180                                                                 NA's   :154    
   hasGithub        hasReddit        platform            coinNum           minInvestment   
 Min.   :0.0000   Min.   :0.0000   Length:2767        Min.   :1.200e+01   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:0.0000   Class :character   1st Qu.:5.000e+07   1st Qu.:0.0000  
 Median :1.0000   Median :1.0000   Mode  :character   Median :1.800e+08   Median :0.0000  
 Mean   :0.5779   Mean   :0.6328                      Mean   :8.178e+12   Mean   :0.4532  
 3rd Qu.:1.0000   3rd Qu.:1.0000                      3rd Qu.:6.000e+08   3rd Qu.:1.0000  
 Max.   :1.0000   Max.   :1.0000                      Max.   :2.262e+16   Max.   :1.0000  

 distributedPercentage
 Min.   :  0.000      
 1st Qu.:  0.400      
 Median :  0.550      
 Mean   :  1.061      
 3rd Qu.:  0.700      
 Max.   :869.750      
```

For looking deeper into the data the skim() function from the skimr. Here we could identify that there are a few missing features from PriceUSD, teamSize, countryRegion and some whitespaces in platform.



```
> skim(data)
── Data Summary ────────────────────────
                           Values
Name                       data  
Number of rows             2767  
Number of columns          16    
_______________________          
Column type frequency:           
  character                6     
  numeric                  10    
________________________         
Group variables            None  

── Variable type: character ────────────────────────────────────────────────────────────────────────
  skim_variable n_missing complete_rate min max empty n_unique whitespace
1 success               0             1   1   1     0        2          0
2 brandSlogan           0             1   6  75     0     2763          0
3 countryRegion         0             1   0  32    71      121          0
4 startDate             0             1  10  10     0      760          0
5 endDate               0             1  10  10     0      776          0
6 platform              0             1   1  27     0      130          6

── Variable type: numeric ──────────────────────────────────────────────────────────────────────────
   skim_variable         n_missing complete_rate     mean       sd p0        p25         p50          p75    p100 hist 
 1 ID                            0         1     1.38e+ 3 7.99e+ 2  1       692.        1384         2076.  2.77e 3 ▇▇▇▇▇
 2 hasVideo                      0         1     7.26e- 1 4.46e- 1  0         0            1            1   1   e 0 ▃▁▁▁▇
 3 rating                        0         1     3.12e+ 0 7.14e- 1  1         2.6          3.1          3.7 4.8 e 0 ▁▃▇▅▁
 4 priceUSD                    180         0.935 1.90e+ 1 7.75e+ 2  0         0.04         0.12         0.5 3.94e 4 ▇▁▁▁▁
 5 teamSize                    154         0.944 1.31e+ 1 8.08e+ 0  1         7           12           17   7.5 e 1 ▇▂▁▁▁
 6 hasGithub                     0         1     5.78e- 1 4.94e- 1  0         0            1            1   1   e 0 ▆▁▁▁▇
 7 hasReddit                     0         1     6.33e- 1 4.82e- 1  0         0            1            1   1   e 0 ▅▁▁▁▇
 8 coinNum                       0         1     8.18e+12 4.30e+14 12 50000000   180000000    600000000   2.26e16 ▇▁▁▁▁
 9 minInvestment                 0         1     4.53e- 1 4.98e- 1  0         0            0            1   1   e 0 ▇▁▁▁▆
10 distributedPercentage         0         1     1.06e+ 0 1.75e+ 1  0         0.4          0.55         0.7 8.70e 2 ▇▁▁▁▁
```

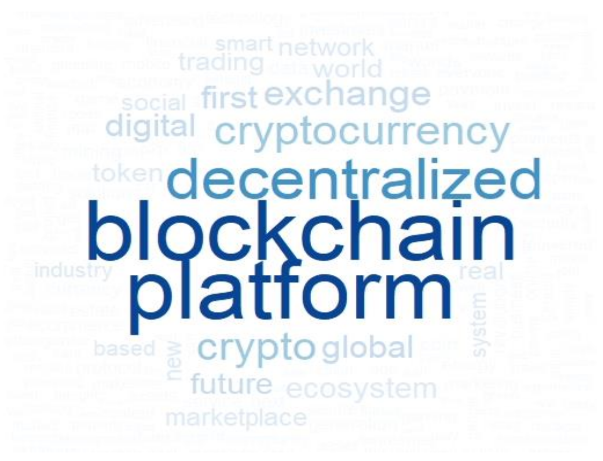
There are a total of 16 features, the breakdown of features by data types is as follows:

Characters	success, brandSlogan, countryRegion, startDate, endDate, platform
Integer	ID, hasVideo, teamSize, hasGithub, hasReddit, minInvestment
Numeric	rating, priceUSD, coinNum, distributedPercentage

2.1 Understanding the variables:

Brand Slogan

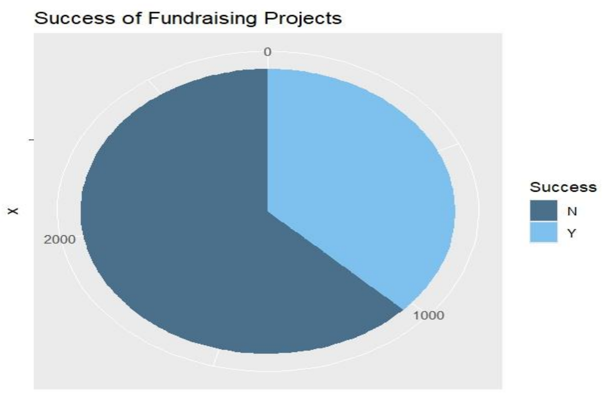
The 'brandSlogan' variable represents the slogans used by fundraising teams. A word cloud visualization technique has been employed to identify the most frequently used words in this feature. Words that appear prominently in the word cloud are those that are utilized most frequently in slogans.



Success

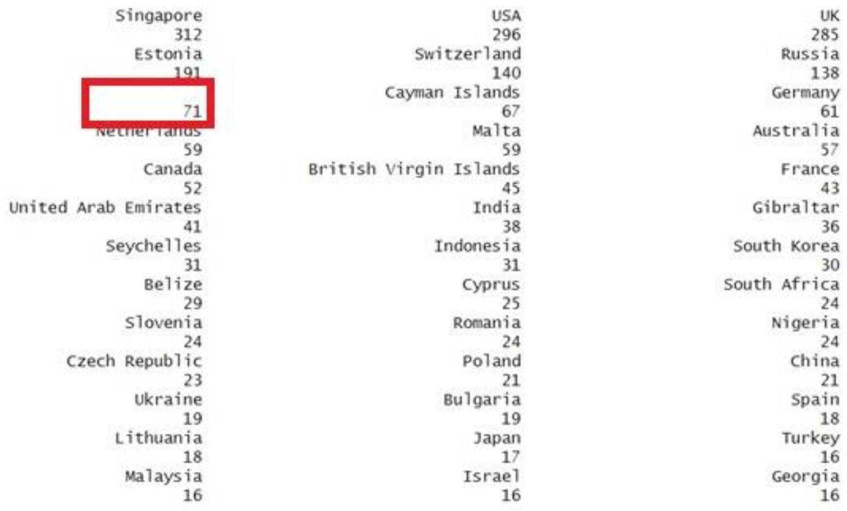
The 'success' variable denotes the success rate of campaigns or coins based on specified features. Analysis reveals a higher frequency of failures compared to successful projects. If the ICO could rise the required amount within the specific period of time then only they are considered to be a success denoted by the 'Y'

The disparity between successful and unsuccessful outcomes underscores the challenges inherent in achieving fundraising goals within the designated time constraints. Understanding the factors contributing to project success or failure is crucial for optimizing fundraising strategies and improving overall campaign effectiveness.



countryRegion

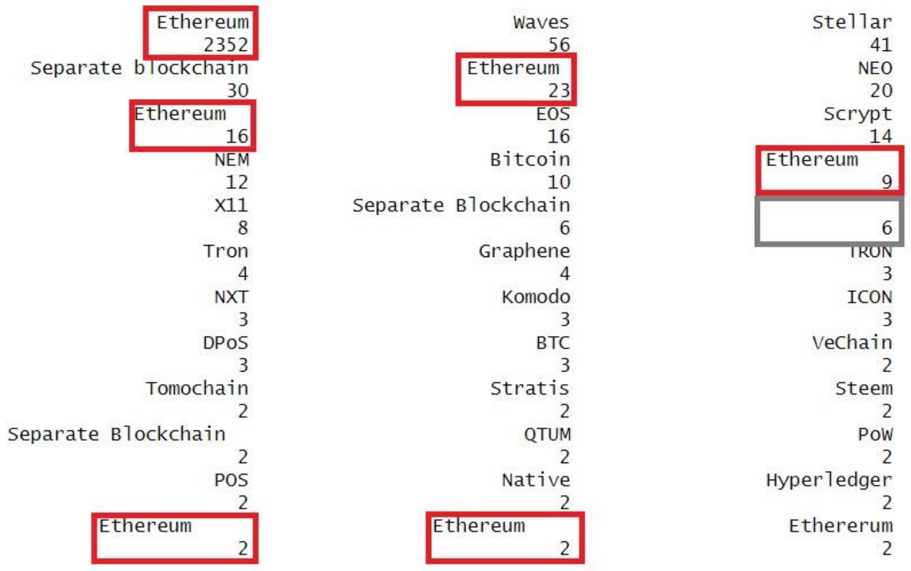
The 'countryRegion' variable represents the geographical locations of fundraising teams. It is notable that 71 observations have missing country information. Additionally, there are instances of duplicated country names, such as 'usa' and 'USA', which may introduce inconsistency in the dataset.



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

Platform

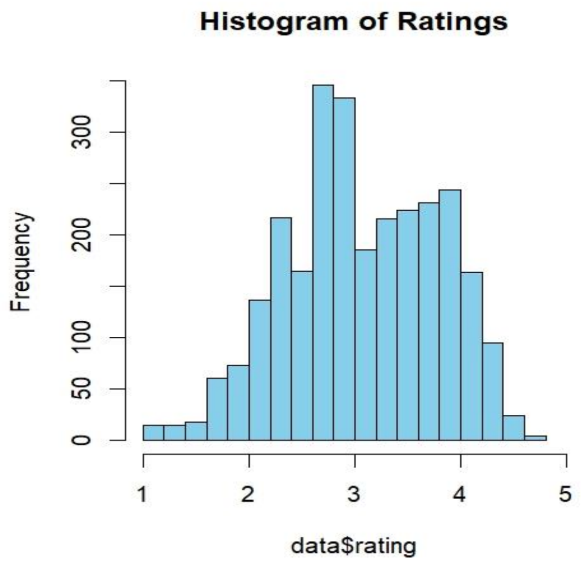
The 'platform' variable indicates the block chain platforms utilized in the fundraising efforts. It is evident that Ethereum is the most frequently occurring platform in the dataset, with numerous instances of misspellings and some missing values. The inconsistencies in spelling and the presence of missing values may introduce challenges in data analysis and interpretation.



```
            Ethereum                 Waves       Stellar
                2352                    56            41
  Separate blockchain             Ethereum           NEO
                  30                    23            20
           Ethereum                    EOS        Scrypt
                  16                    16            14
                 NEM               Bitcoin     Ethereum
                  12                    10             9
                 X11   Separate Blockchain
                   8                     6             6
                Tron              Graphene          TRON
                   4                     4             3
                 NXT               Komodo          ICON
                   3                     3             3
                DPoS                   BTC       VeChain
                   3                     3             2
           Tomochain               Stratis         Steem
                   2                     2             2
Separate Blockchain                   QTUM           PoW
                   2                     2             2
                 POS                Native   Hyperledger
                   2                     2             2
      Ethereum                Ethereum          Ethererum
                   2                     2             2
```

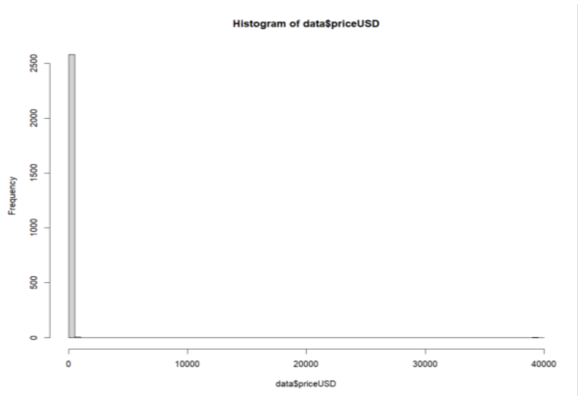
Ratings

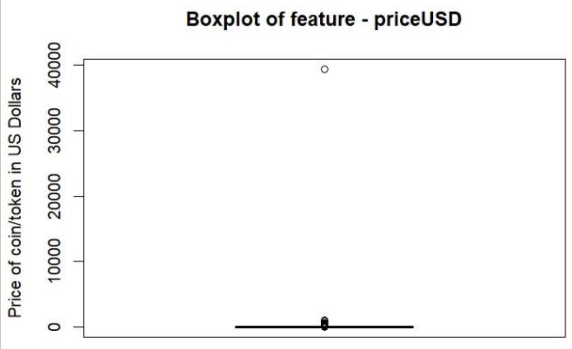
The assessment provided by investment professionals serves as a crucial measure for investors evaluating the quality of ICO (Initial Coin Offering) projects. These ratings commonly span from 1 (low) to 5 (excellent), with most falling between 2 and 4. It's worth noting that the dataset includes a maximum rating of 4.8, signifying a notably high level of project excellence.

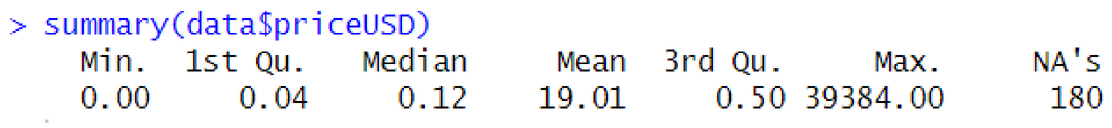


PriceUSD

The 'priceUSD' variable denotes the price of each token issued in US dollars. From the analysis we can





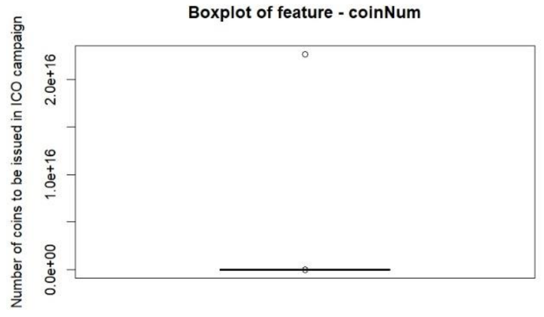


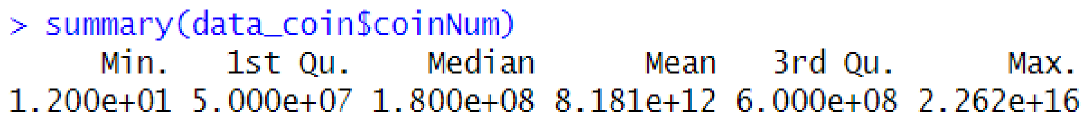
```
> summary(data$priceUSD)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's 
    0.00     0.04     0.12    19.01     0.50 39384.00      180 
```

In this analysis, it is evident that there exists an outlier within the dataset, considering that the mean priceUSD stands at merely 19.01, with a median of 0.12. The value of 39384 notably deviates from this trend, indicating a substantial anomaly.

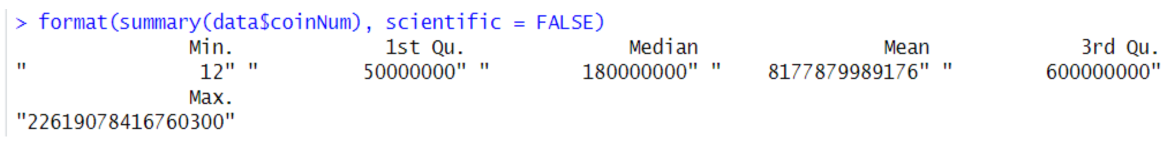
CoinNUM

The coinNUM has no missing values but has outlier, here we can see that that the max value is a really high number.





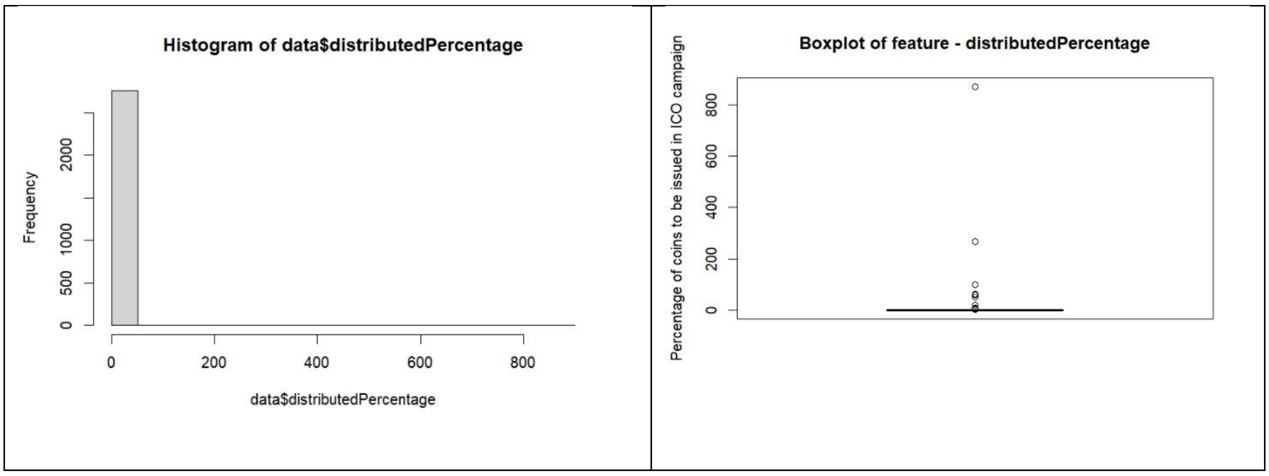
```
> summary(data_coin$coinNum)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
1.200e+01 5.000e+07 1.800e+08 8.181e+12 6.000e+08 2.262e+16 
```



```
> format(summary(data$coinNum), scientific = FALSE)
                 Min.                 1st Qu.                  Median                    Mean                 3rd Qu. 
"                 12" "          50000000" "         180000000" "      8177879989176" "         600000000" 
                 Max. 
"22619078416760300" 
```

distributedPercentage

distributedPercentage represents the percentage of blockchain coins distributed to investors relative to all the coins created for an ICO project. This metric reflects the level of coin distribution to external investors compared to the total supply of coins.



DATA PRE-PROCESSING

Preparation

The elimination of noise from datasets is a crucial objective in data cleaning, as noise can impede various forms of data analysis. While many current data cleaning techniques target the removal of noise caused by low-level data errors stemming from imperfect data collection processes, it's important to note that irrelevant or marginally relevant data objects can also pose significant obstacles to effective data analysis. (Gudivada et al., 2017)

The data cleaning is categorised into 3 methods,

1. Missing/repeated/Incorrect data handling
2. Imputing the data

3. Handling outliers

countryRegion

Here variable in the dataset revealed the presence of missing values and inconsistencies in naming conventions. To address these issues and ensure data integrity, the following steps were undertaken:

Handling Missing Values: 'countryRegion' variable identified 71 missing values.They were replaced with the label 'unknown', signifying instances where the country region information was not provided.

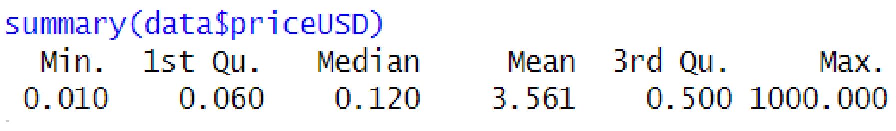
Standardization of Country Names: To enhance data consistency and clarity, a uniform naming convention was applied to the 'countryRegion' variable. Specific country names were standardized as follows:

- 'india' → 'India'
- 'México' → 'Mexico'
- 'SINGAPORE' → 'Singapore'
- 'usa' → 'USA'
- 'Curaçao' → 'Curacao'
- Other entries remained unchanged.

PriceUSD

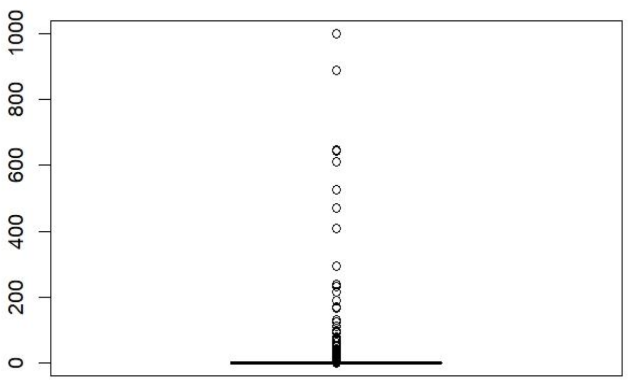
During the initial analysis of the dataset, an outlier was identified in the 'priceUSD' variable, with a value of 39384. This outlier may significantly skew the distribution and distort the results of subsequent analyses.

This outlier deviates significantly from the typical range of prices observed in the dataset.



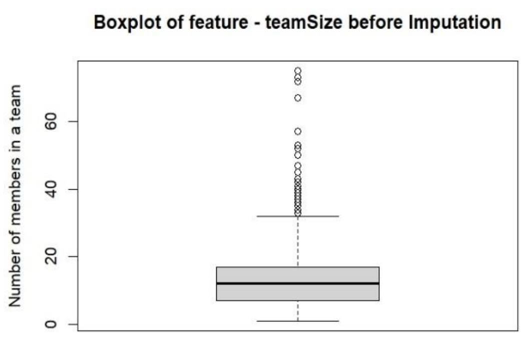
```
summary(data$priceUSD)
   Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
  0.010    0.060    0.120    3.561    0.500 1000.000
```

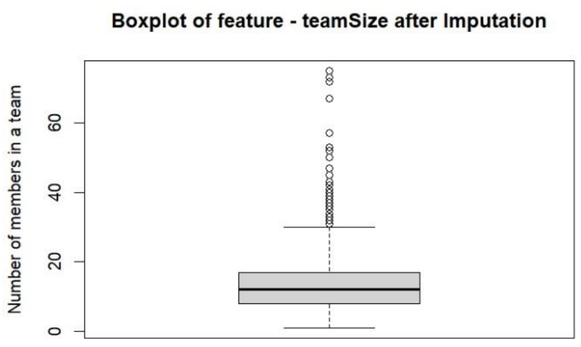
There are some missing values as well, we will be imputing the data with the mean of the PriceUSD variable. Here's after cleaning and imputing the data here, this is the histogram.



TeamSize

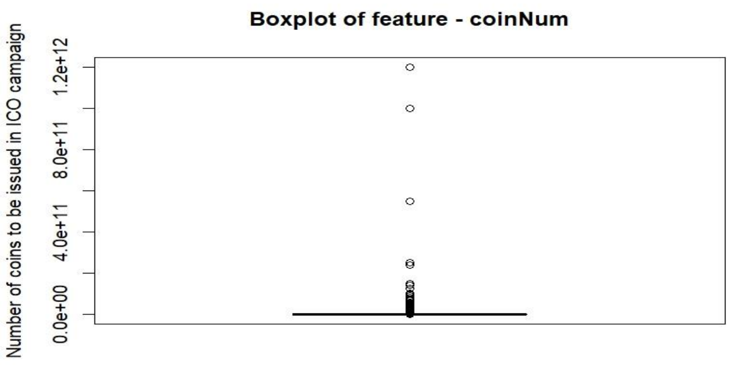
There were a total of 154 missing values they replaced with the median team size, calculated as 12 members, while omitting any missing values during the calculation.





CoinNum

The observation containing the outlier value was removed from the dataset. Since there are no missing values the outlier is removed and the dataset. An outlier was identified at observation 1593, where the number of coins issued was 22,619,078,416,760,300. The maximum number of coins issued decreased to 1.200e+12.



DistributedPercentage

It was found that there was only one observation with a value of 0%, indicating no distribution of coins to investors. However, this project was marked as successful, signifying an anomaly in the dataset. Therefore, this observation was retained for further analysis.

Another aspect of the analysis involved identifying observations where the 'distributedPercentage' exceeded 100%, which is not feasible as it implies a distribution of more coins than were created. These observations were deemed as outliers and were subsequently removed from the dataset to ensure data integrity. There were a total of 10 outliers, These outliers were removed from the dataset.

Platform

Leading and trailing whitespace in the platform names were removed, and all platform names were converted to lowercase for uniformity.

Several platform names were identified for standardization to improve clarity and consistency in the dataset. Notable changes include:

- 'btc' was standardized to 'bitcoin'.
- 'stellarprotocol' was standardized to 'stellar'.
- 'x11blockchain' was standardized to 'x11'.
- Variants of 'ethereum' (such as 'eth', 'ethererum', 'etherum') were standardized to 'ethereum'.
- Platforms combining Proof of Stake (POS) and Proof of Work (POW) mechanisms were categorized under 'pos+pow'.
- Other minor adjustments were made to streamline platform names.

Any zero-width space characters present in the platform names were removed to ensure consistency and avoid encoding issues.

Adding a Feature Date

Date

The transformation of start and end dates into the new feature 'Duration_of_campaign' provides valuable insights into the temporal aspect of each campaign. However, during the creation of this new feature, several inconsistencies were identified:

- Negative Duration Values: Approximately 12 entries exhibited negative values for the campaign duration, which is logically implausible given the chronological nature of campaign periods. Consequently, these entries were deemed erroneous and removed from the dataset.
- Zero Duration Values: While most negative duration entries were addressed, a few records with zero duration values remained in the dataset. It's noteworthy that some successful campaigns lacked precise start or end dates, leading to the presence of such anomalies. These instances were retained in the dataset for further analysis, as they may represent either data inaccuracies or genuinely successful campaigns without specified dates.

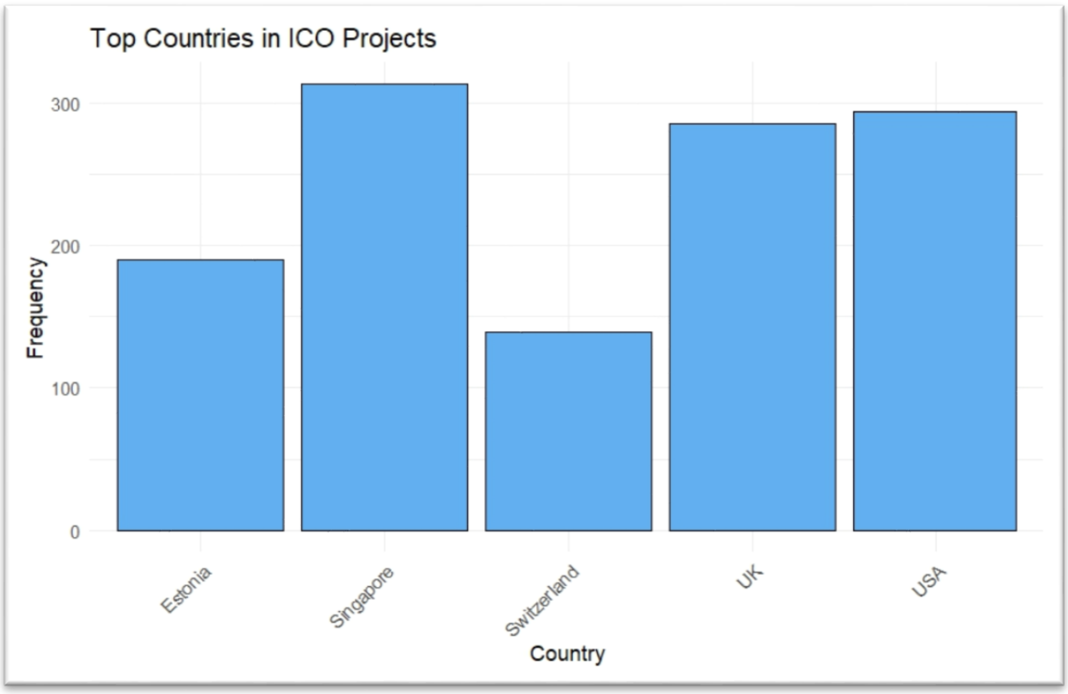
Country

- Order of Countries

The dataset revealed insights into the distribution of fundraising teams across different regions. Among these, the top three countries with the highest frequency of fundraising teams are as follows:

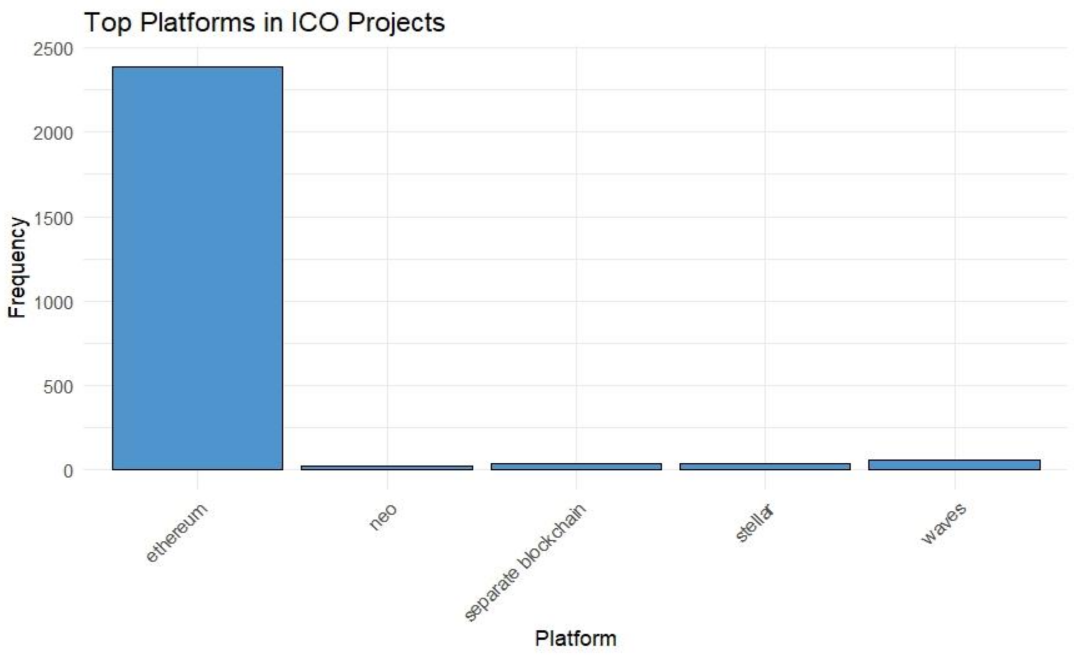
1. Singapore
2. United States (USA)
3. United Kingdom (UK)

A binary feature was introduced to signify whether an observation corresponds to one of the top three countries. This newly created attribute, named 'top_countries_indicator', serves to differentiate observation. The new observations associated with Singapore, the USA, or the UK are assigned a value of 1, while all other observations receive a value of 0.



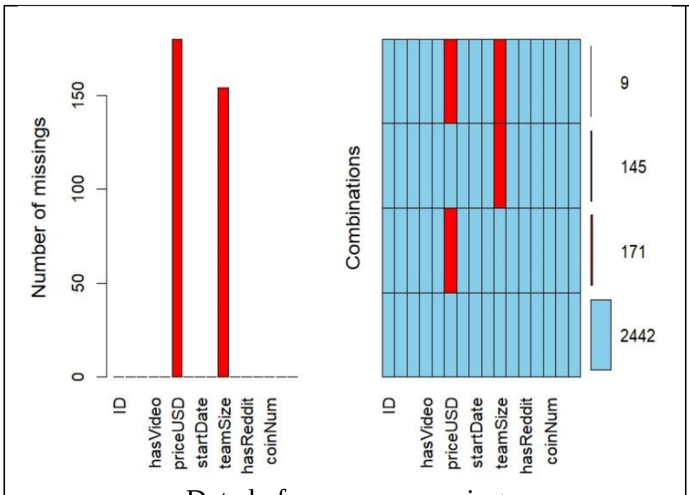
Platform

New feature called "top_platform_indicator." This feature categorizes the platform variable, signifying whether a project is built on Ethereum or not. We've assigned a value of 1 to projects built on Ethereum, considering its prominence in the blockchain landscape, and 0 to all other platforms.

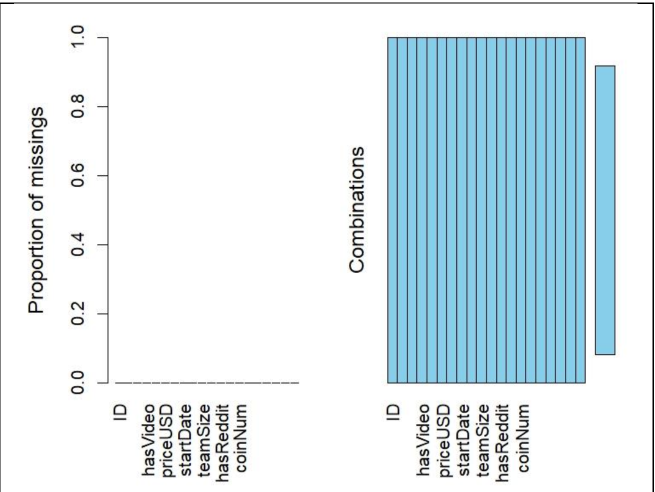


Final data

Dealt with missing values



Data before pre-processing



Data after pre-processing

MODELLING

Data Splitting

To ensure robust model evaluation, we partitioned our dataset into training and testing sets using an 80-20 split. This division allows us to train the models on a majority of the data while preserving a separate portion for unbiased performance evaluation.

Decision Tree

Decision trees stand as versatile machine learning tools adept at tackling classification and regression challenges with ease. Their intuitive nature facilitates the creation of decision rules based on input features, enabling straightforward interpretation of complex data relationships. Decision trees are intuitive models that recursively partition the data into subsets based on feature values, ultimately leading to a decision or prediction. They are well-suited for capturing complex relationships within the data and offer interpretability through easily understandable decision rules.

The decision tree begins with the root node, which evaluates the feature ‘rating’. If the rating is less than or equal to 3.3, the decision tree branches to further nodes. The decision tree branches into multiple nodes based on different feature conditions, such as Duration_of_campaign, hasReddit, and teamSize. Each node represents a decision point where the algorithm evaluates a specific feature to make predictions.

- For ICO projects with a rating less than or equal to 3.3 and a Duration_of_campaign greater than 7 days, the model predicts failure.
- If the rating is greater than 3.3 and the teamSize is less than or equal to 14, the model predicts failure.
- Projects with a rating greater than 3.3 and a teamSize greater than 14 are predicted to succeed.
- The presence of hasReddit also influences predictions, with projects having Reddit presence more likely to succeed.

Adaptive Boosting of Decision Trees (ADABoost)

AdaBoost, an ensemble learning algorithm, uses the pooled knowledge of numerous weak learners to create a strong classifier. AdaBoost refines its predicting accuracy by repeatedly modifying the weights of misclassified examples, increasing robustness to noisy input and boosting overall performance.

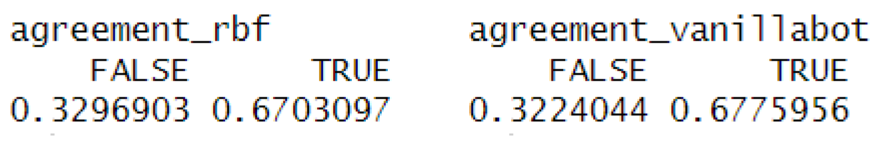
Random Forest

Random Forest is a powerful ensemble approach that uses the variety of several decision trees to provide robust and reliable forecasts. Random Forest, using bootstrap aggregation and feature randomization, reduces the danger of overfitting while retaining high levels of prediction accuracy, making it a reliable ally in our modelling armoury.

Support Vector Machine (SVM)

Support Vector Machine (SVM) stands out as a robust and flexible algorithm highly regarded for its effectiveness in both classification and regression endeavors. Leveraging the identification of the optimal hyperplane to delineate class boundaries within the feature space, SVM exhibits remarkable proficiency in managing intricate data distributions and excels in capturing nonlinear associations.

Two SVM models were constructed using different kernels: vanilladot and rbfdot. The goal is to assess the effectiveness of these models in predicting campaign success based on various features. Two SVM models with different kernels: one with a linear kernel (vanilladot) and another with a radial basis function kernel (rbfdot).



```
agreement_rbf              agreement_vanillabot
    FALSE      TRUE              FALSE      TRUE
0.3296903 0.6703097        0.3224044 0.6775956
```

K-Nearest Neighbors (k-NN)

K-Nearest Neighbors (k-NN) epitomizes simplicity in machine learning models. By assigning labels or predicting values based on the consensus of neighbouring data points, k-NN offers a straight forward approach to classification and regression tasks, demonstrating resilience to noise and outliers.

- The chosen value of K was initially set to the square root of the training size, resulting in K = 46.

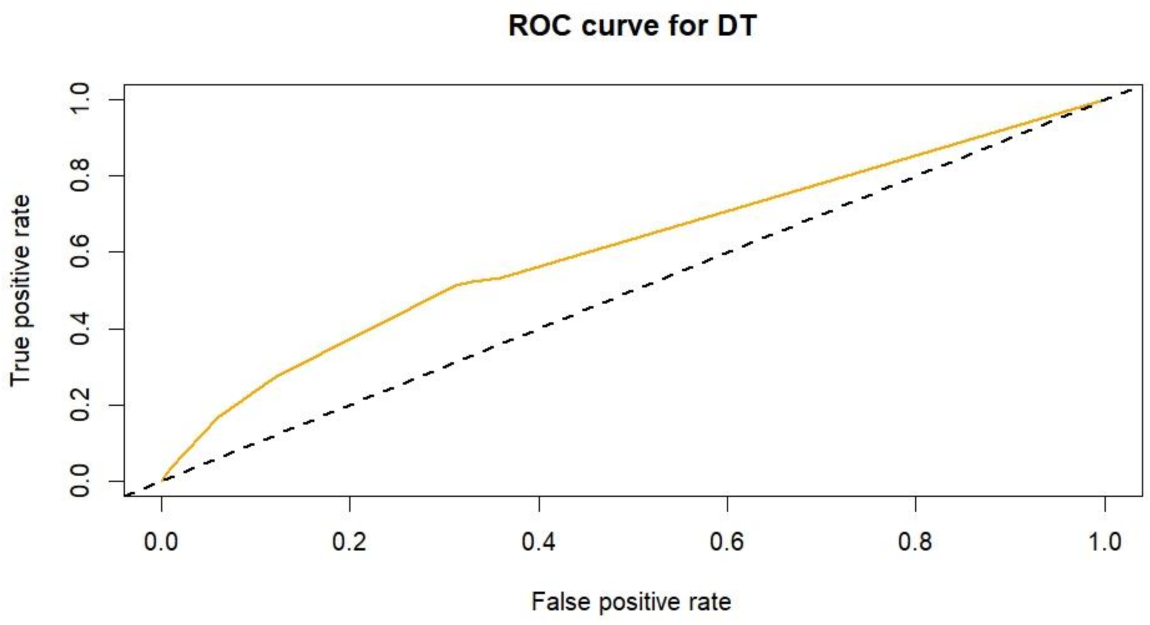
EVALUATION

Decision Tree

- Accuracy: The model achieved an accuracy of 66.67%, indicating its overall predictive performance.
- Sensitivity: The model's sensitivity (recall) was 27.46%, suggesting its ability to correctly identify successful projects.
- Specificity: The specificity of the model was 87.92%, indicating its capability to accurately identify unsuccessful projects.
- Precision: The precision of the model was 55.21%, reflecting its ability to avoid false positives.
- F1 Score: The F1 score, a balance between precision and recall, was 36.68%.

Actual / Predicted	**Negative (N)**	**Positive (Y)**
Negative (N)	313	140
Positive (Y)	43	53

- AUC: The area under the ROC curve was 61.01%, indicating the model's ability to discriminate between success and failure.



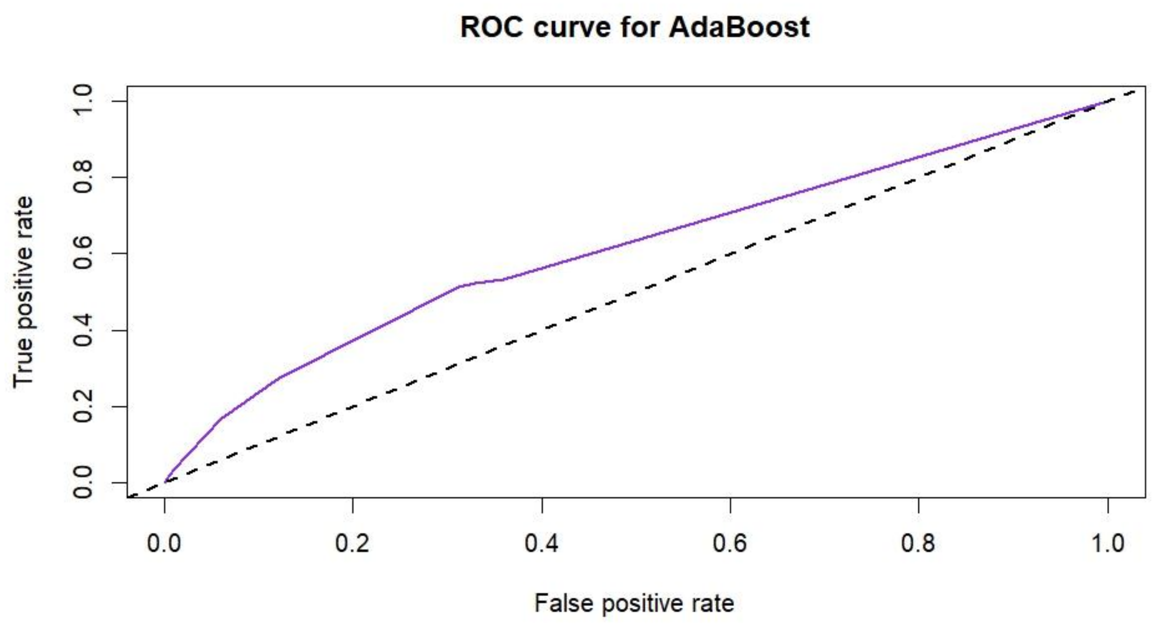
Adaptive Boosting of Decision Tree

- Accuracy: The model achieved an accuracy of 66.67%, indicating its overall predictive performance.
- Sensitivity: The model's sensitivity (recall) was 27.46%, suggesting its ability to correctly identify successful projects.
- Specificity: The specificity of the model was 87.92%, indicating its capability to accurately identify unsuccessful projects.
- Precision: The precision of the model was 55.21%, reflecting its ability to avoid false positives.
- F1 Score: The F1 score, a balance between precision and recall, was 36.68%.

Confusion Matrix

Actual / Predicted	**Negative (N)**	**Positive (Y)**
Negative (N)	313	140
Positive (Y)	43	53

AUC: The area under the ROC curve was 61.01%, indicating the model's ability to discriminate between success and failure.



Random Forest

We evaluated the performance of the random forest model using the following metrics:

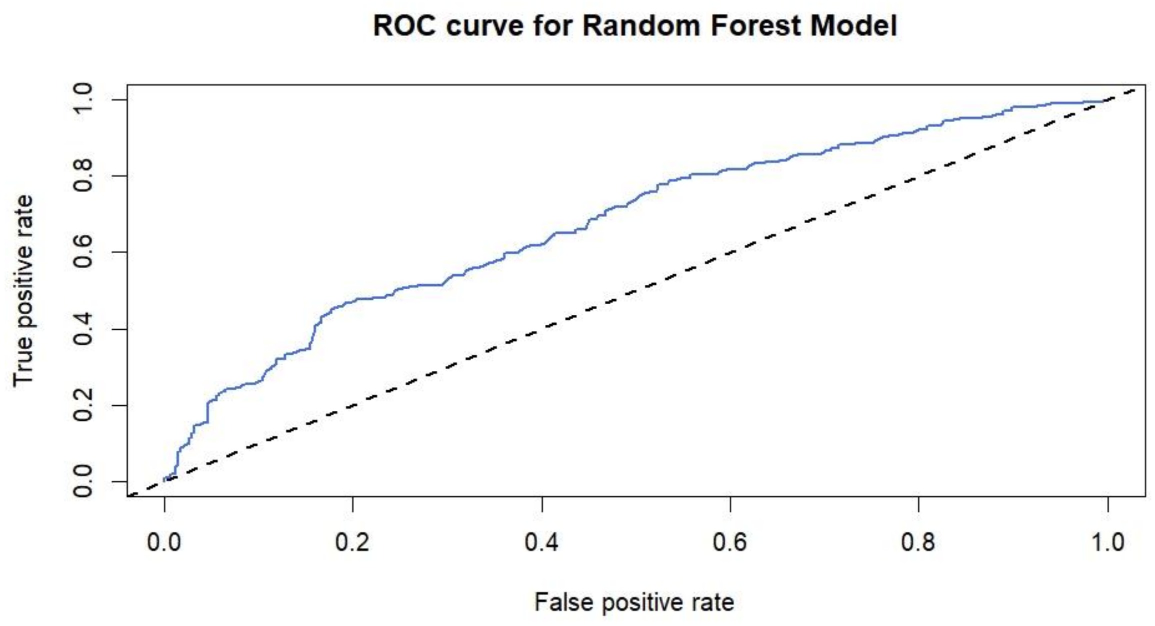
- Accuracy: 68.25%
- Sensitivity (Recall): 44.12%Specificity: 82.56%
- Precision (Positive Predictive Value): 60%
- F1 Score: 50.85%

Confusion Matrix

The confusion matrix provides further insight into the model's performance:

Actual / Predicted	**Negative (N)**	**Positive (Y)**
Negative (N)	284	114
Positive (Y)	60	90

The ROC curve illustrates the trade-off between sensitivity and specificity. The Area under the Curve (AUC) value for the ROC curve is 0.6766, indicating the model's ability to distinguish between success and failure.



Analysis of feature importance reveals the following key factors influencing the success of crowd funding campaigns:

- Rating
- Team size
- Duration of the campaign
- Distributed percentage
- Presence of video
- Minimum investment required
- Presence on GitHub and Reddit
- Indicators for top countries and platforms

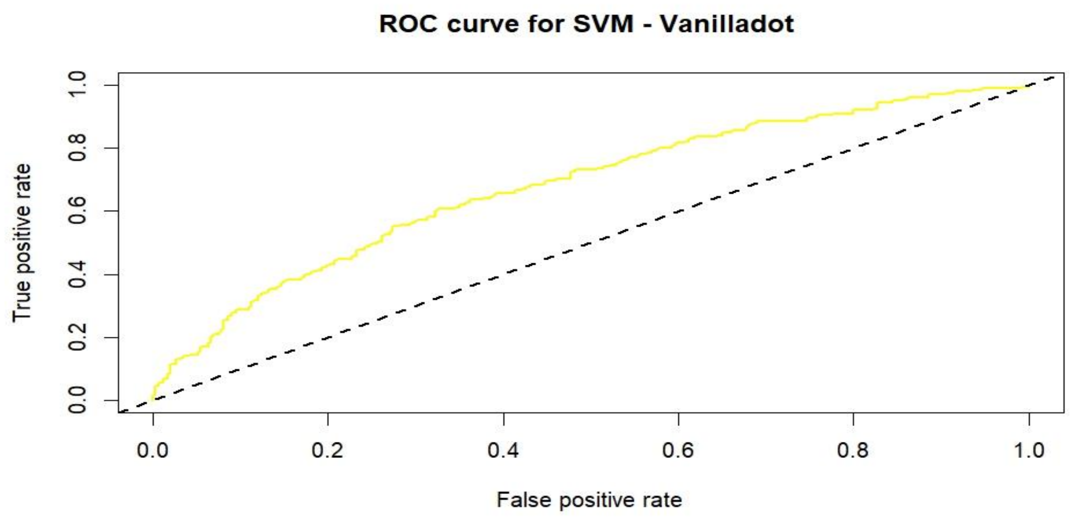
Support Vector Machine (SVM)

Support Vector Machine (SVM) with Vanilla Dot kernel

- The model achieves an accuracy of 67.76%, indicating its overall effectiveness in correctly classifying crowdfunding campaigns as successful or unsuccessful.
- Sensitivity (True Positive Rate): 59.43%
- Specificity (True Negative Rate): 68.85%
- Positive Predictive Value (Precision): 31.34%
- Negative Predictive Value: 87.64%
- F1 Score: 41.04%

Confusion Matrix:

Actual / Predicted	**Negative (N)**	**Positive (Y)**
Negative (N)	302	46
Positive (Y)	131	70



The AUC value of 67.83% indicates that the model performs better than random guessing in distinguishing between successful and unsuccessful campaigns.

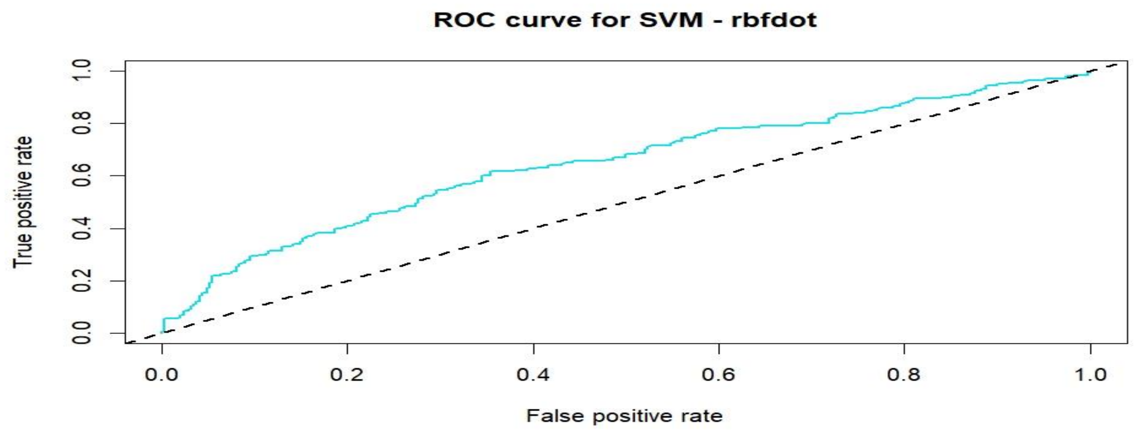
Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel

- Accuracy: The model achieves an accuracy of 67.03%, indicating its overall effectiveness in correctly classifying crowdfunding campaigns as successful or unsuccessful.
- Kappa Score: 21.09%
- Sensitivity (True Positive Rate): 59.43%
- Specificity (True Negative Rate): 68.85%
- Positive Predictive Value (Precision): 31.34%
- Negative Predictive Value: 87.64%
- F1 Score: 41.04%

Confusion Matrix:

Actual / Predicted	Negative (N)	Positive (Y)
Negative (N)	305	43
Positive (Y)	138	63

The AUC value of 64.61% indicates that the model performs better than random guessing in distinguishing between successful and unsuccessful campaigns.



K-Nearest Neighbors (k-NN)

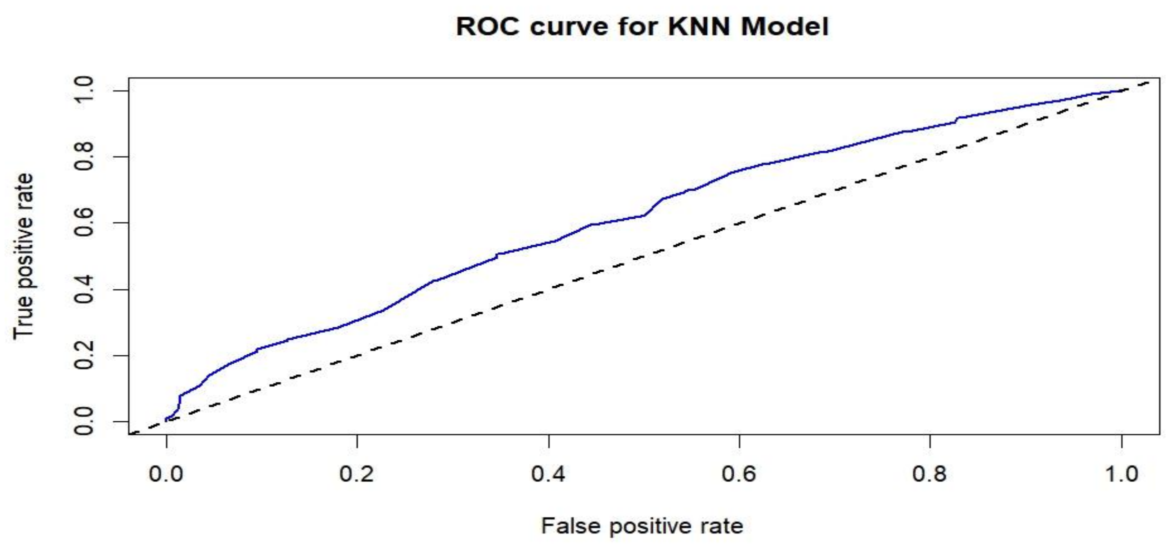
- Accuracy: The model achieved an accuracy of 64.3%, indicating its overall predictive performance.
- Sensitivity: The model's sensitivity (recall) was 26.94%, suggesting its ability to correctly identify successful projects.
- Specificity: The specificity of the model was 84.55%, indicating its capability to accurately identify unsuccessful projects.
- Precision: The precision of the model was 48.6%, reflecting its ability to avoid false positives.
- F1 Score: The F1 score, a balance between precision and recall, was 34.67%.

Confusion Matrix

The confusion matrix provides further insight into the model's performance:

Actual / Predicted	**Negative (N)**	**Positive (Y)**
Negative (N)	301	141
Positive (Y)	55	52

AUC: The area under the ROC curve was 61.25%, indicating the model's ability to discriminate between success and failure.



CONCLUSION

For Random Forest, based on the analysis of the random forest model, we can conclude that factors such as rating, team size, duration of the campaign, and engagement on platforms like GitHub and Reddit play significant roles in predicting the success of crowd funding campaigns. Campaign organizers can leverage these insights to optimize their strategies and improve the likelihood of success.

Decision tree model provides valuable insights by analysing key features such as rating, Duration_of_campaign, and teamSize, the model offers predictions on the likelihood of project success.

Ad Adaptive Boosting of Decision Tree Based on the evaluation metrics, the AdaBoost ensemble of Decision Trees shows moderate performance in predicting project success, AdaBoost ensemble model provides valuable insights into the success of blockchain projects.

The SVM model with a Vanilla Dot kernel & SVM model with an RBF kernel demonstrates moderate predictive performance in determining the success of crowd funding campaigns. Key features such as project rating, campaign duration, team size, and online presence play crucial roles in predicting campaign outcomes

Based on these metrics, the Random Forest model emerges as the top performer, boasting the highest accuracy (68.25%), sensitivity (44.12%), and precision (60.00%). It also achieves a relatively high AUC score of 67.66%. Following closely behind are the SVM models with both Vanilla Dot and RBF kernels, demonstrating comparable accuracy and AUC values.

In summary, the models are ranked as follows from best to worst:

- Random Forest
- SVM with Vanilla Dot Kernel
- SVM with RBF Kernel
- Decision Tree and Adaptive Boosting of Decision Tree
- K-Nearest Neighbors

Limitations

Incomplete Consideration of Factors: The analysis overlooked certain factors like marketing strategies, token utility, and legal compliance, which could significantly impact the success of ICO campaigns. These factors are vital for attracting investors and ensuring project sustainability.

Handling Missing Values: The approach to handling missing values, such as imputing them with the median, might introduce bias. Obtaining accurate data from the source could provide more reliable insights.

Limited Evaluation Techniques: The analysis mainly relied on standard evaluation methods, neglecting advanced techniques like k-fold cross-validation. Implementing such techniques could enhance model accuracy and reliability.

Data Imbalance: The dataset exhibited class imbalance, potentially affecting model performance metrics like AUC. Addressing this imbalance using techniques like oversampling or undersampling may be necessary.

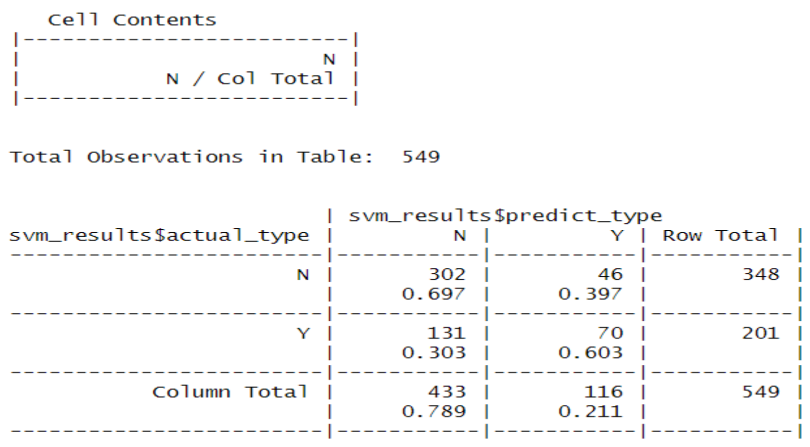
Model Complexity: The simplicity of the decision tree and AdaBoost models might have limited their ability to capture complex data patterns effectively. Utilizing more sophisticated models or ensemble methods could yield better results.

Reference

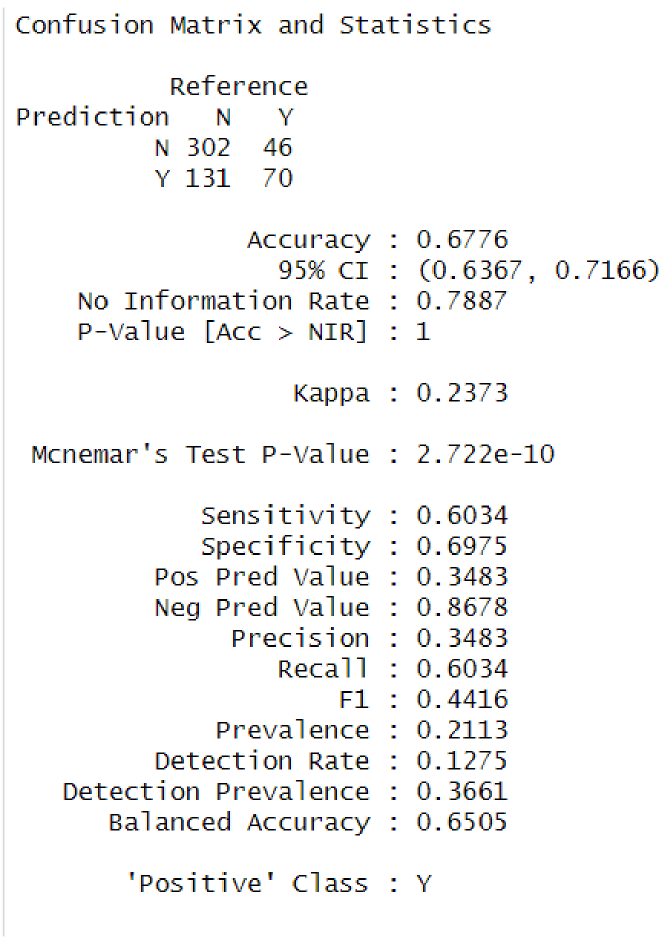
Gudivada, V., Apon, A. and Ding, J., 2017. Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *International Journal on Advances in Software*, *10*(1), pp.1-20.

APPENDIX

SVM - Vanillabot



```
   Cell Contents
|-------------------------|
|                       N |
|           N / Col Total |
|-------------------------|

Total Observations in Table:  549

                         | svm_results$predict_type
svm_results$actual_type |         N |         Y | Row Total |
------------------------|-----------|-----------|-----------|
                      N |       302 |        46 |       348 |
                        |     0.697 |     0.397 |           |
------------------------|-----------|-----------|-----------|
                      Y |       131 |        70 |       201 |
                        |     0.303 |     0.603 |           |
------------------------|-----------|-----------|-----------|
           Column Total |       433 |       116 |       549 |
                        |     0.789 |     0.211 |           |
------------------------|-----------|-----------|-----------|
```



```
Confusion Matrix and Statistics

          Reference
Prediction   N   Y
         N 302  46
         Y 131  70

               Accuracy : 0.6776
                 95% CI : (0.6367, 0.7166)
    No Information Rate : 0.7887
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2373

 Mcnemar's Test P-Value : 2.722e-10

            Sensitivity : 0.6034
            Specificity : 0.6975
         Pos Pred Value : 0.3483
         Neg Pred Value : 0.8678
              Precision : 0.3483
                 Recall : 0.6034
                     F1 : 0.4416
             Prevalence : 0.2113
         Detection Rate : 0.1275
   Detection Prevalence : 0.3661
      Balanced Accuracy : 0.6505

       'Positive' Class : Y
```

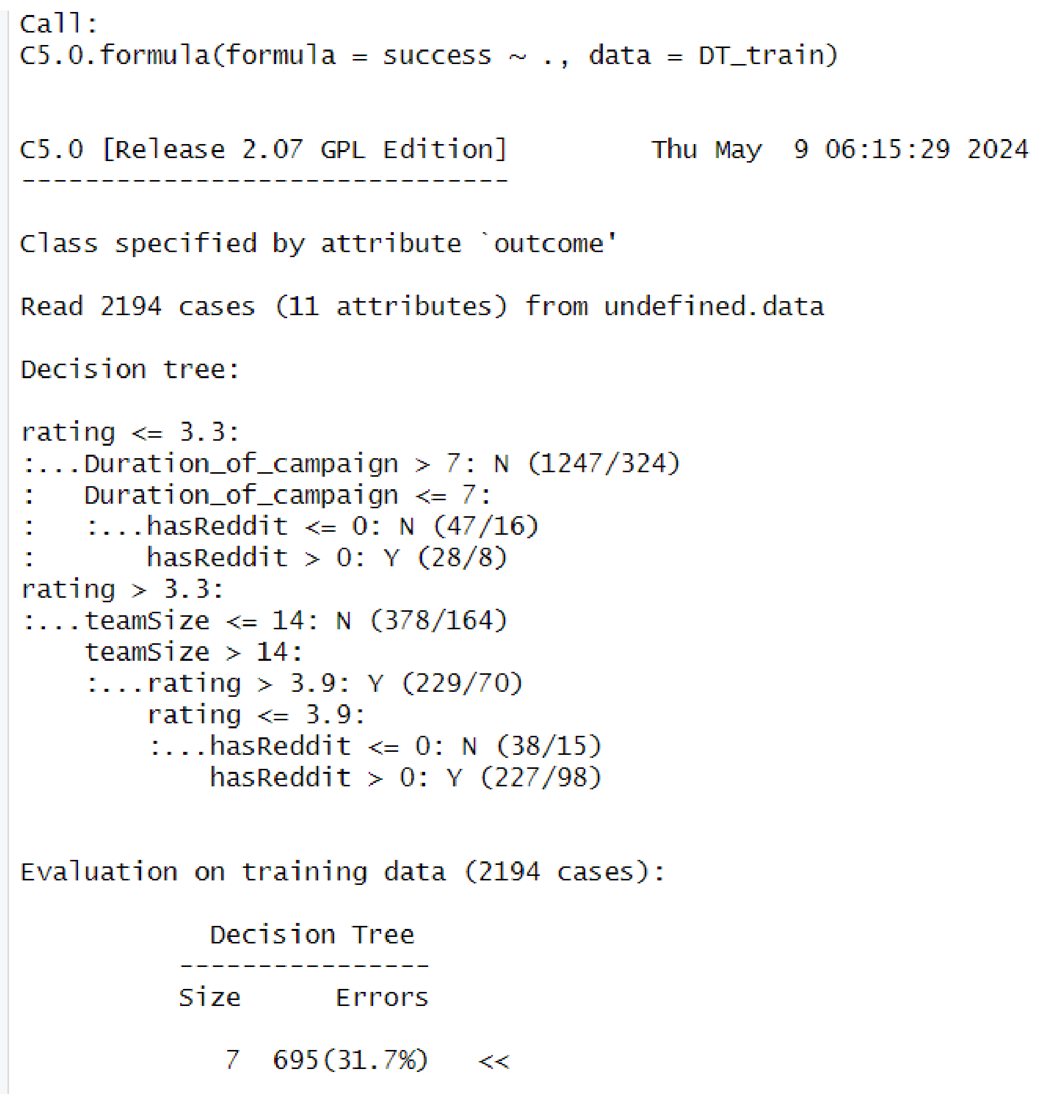
SVM – rbfbot

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Col Total |
|-------------------------|


Total Observations in Table:  549


             | N 
           Y |         N |         Y | Row Total | 
-------------|-----------|-----------|-----------|
           N |       305 |        43 |       348 | 
             |     0.688 |     0.406 |           | 
-------------|-----------|-----------|-----------|
           Y |       138 |        63 |       201 | 
             |     0.312 |     0.594 |           | 
-------------|-----------|-----------|-----------|
Column Total |       443 |       106 |       549 | 
             |     0.807 |     0.193 |           | 
-------------|-----------|-----------|-----------|
```

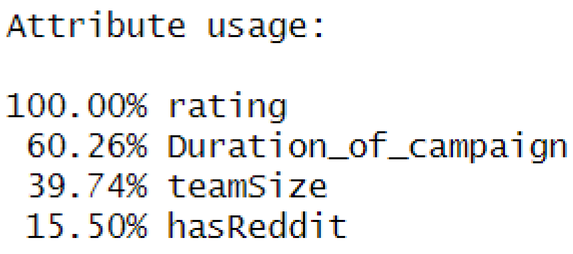
```
Confusion Matrix and Statistics

          Reference
Prediction   N   Y
         N 305  43
         Y 138  63
                                          
               Accuracy : 0.6703          
                 95% CI : (0.6292, 0.7095)
    No Information Rate : 0.8069          
    P-Value [Acc > NIR] : 1               
                                          
                  Kappa : 0.2109          
                                          
 Mcnemar's Test P-Value : 2.809e-12       
                                          
            Sensitivity : 0.5943          
            Specificity : 0.6885          
         Pos Pred Value : 0.3134          
         Neg Pred Value : 0.8764          
              Precision : 0.3134          
                 Recall : 0.5943          
                     F1 : 0.4104          
             Prevalence : 0.1931          
         Detection Rate : 0.1148          
   Detection Prevalence : 0.3661          
      Balanced Accuracy : 0.6414          
                                          
       'Positive' Class : Y               
```

Decision Tree

```
Call:
C5.0.formula(formula = success ~ ., data = DT_train)


C5.0 [Release 2.07 GPL Edition]         Thu May  9 06:15:29 2024
-------------------------------

Class specified by attribute `outcome'

Read 2194 cases (11 attributes) from undefined.data

Decision tree:

rating <= 3.3:
:...Duration_of_campaign > 7: N (1247/324)
:   Duration_of_campaign <= 7:
:   :...hasReddit <= 0: N (47/16)
:       hasReddit > 0: Y (28/8)
rating > 3.3:
:...teamSize <= 14: N (378/164)
    teamSize > 14:
    :...rating > 3.9: Y (229/70)
        rating <= 3.9:
        :...hasReddit <= 0: N (38/15)
            hasReddit > 0: Y (227/98)


Evaluation on training data (2194 cases):

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

             7  695(31.7%)   <<


           (a)   (b)    <-classified as
          ----  ----
          1191   176    (a): class N
           519   308    (b): class Y


        Attribute usage:

        100.00% rating
         60.26% Duration_of_campaign
         39.74% teamSize
         15.50% hasReddit
```



```
Call:
C5.0.formula(formula = success ~ ., data = DT_train)


C5.0 [Release 2.07 GPL Edition]         Thu May  9 06:15:29 2024
-------------------------------

Class specified by attribute `outcome'

Read 2194 cases (11 attributes) from undefined.data

Decision tree:

rating <= 3.3:
:...Duration_of_campaign > 7: N (1247/324)
:   Duration_of_campaign <= 7:
:   :...hasReddit <= 0: N (47/16)
:       hasReddit > 0: Y (28/8)
rating > 3.3:
:...teamSize <= 14: N (378/164)
    teamSize > 14:
    :...rating > 3.9: Y (229/70)
        rating <= 3.9:
        :...hasReddit <= 0: N (38/15)
            hasReddit > 0: Y (227/98)


Evaluation on training data (2194 cases):

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

             7  695(31.7%)   <<
```



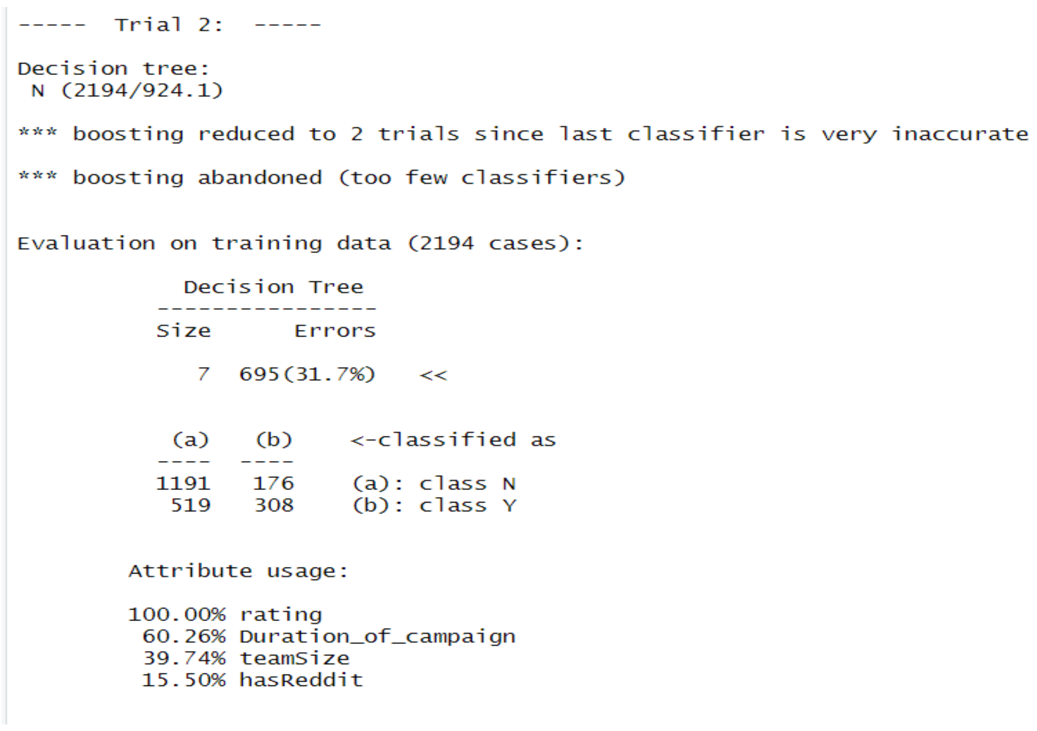
```
Attribute usage:

100.00% rating
 60.26% Duration_of_campaign
 39.74% teamSize
 15.50% hasReddit
```

```
   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  549


             | DT_test$success
     DT_pred |         N |         Y | Row Total |
-------------|-----------|-----------|-----------|
           N |       313 |       140 |       453 |
             |     0.570 |     0.255 |           |
-------------|-----------|-----------|-----------|
           Y |        43 |        53 |        96 |
             |     0.078 |     0.097 |           |
-------------|-----------|-----------|-----------|
Column Total |       356 |       193 |       549 |
-------------|-----------|-----------|-----------|
```

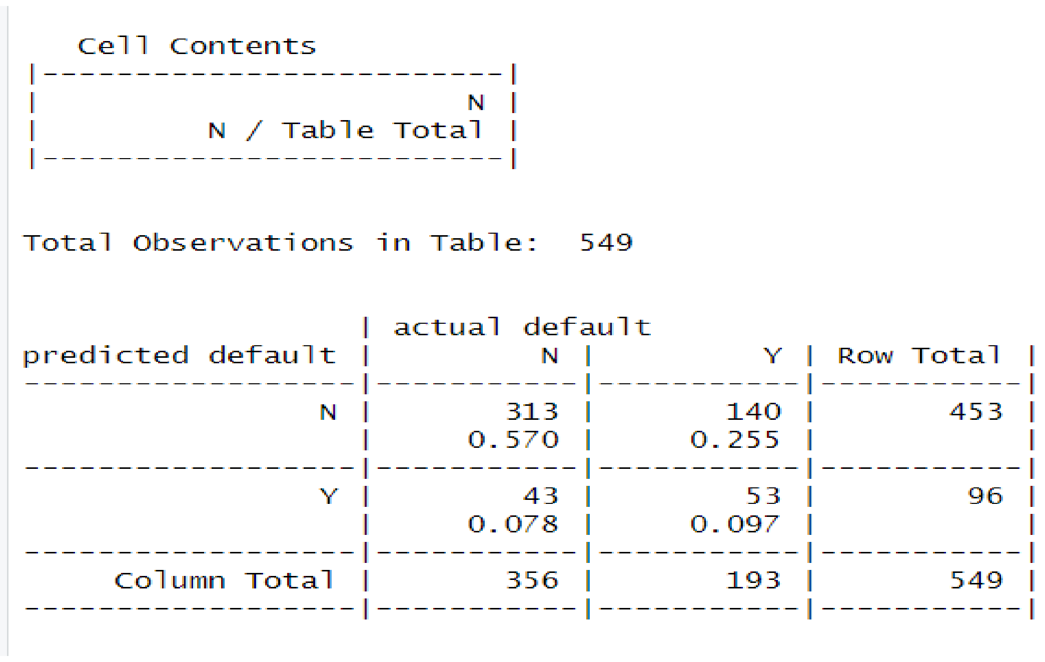
```
Confusion Matrix and Statistics

          Reference
Prediction   N   Y
         N 313 140
         Y  43  53

               Accuracy : 0.6667
                 95% CI : (0.6255, 0.706)
    No Information Rate : 0.6485
    P-Value [Acc > NIR] : 0.1982

                  Kappa : 0.1738

 Mcnemar's Test P-Value : 1.279e-12

            Sensitivity : 0.27461
            Specificity : 0.87921
         Pos Pred Value : 0.55208
         Neg Pred Value : 0.69095
              Precision : 0.55208
                 Recall : 0.27461
                     F1 : 0.36678
             Prevalence : 0.35155
         Detection Rate : 0.09654
   Detection Prevalence : 0.17486
      Balanced Accuracy : 0.57691

       'Positive' Class : Y
```

Adaptive Boosting

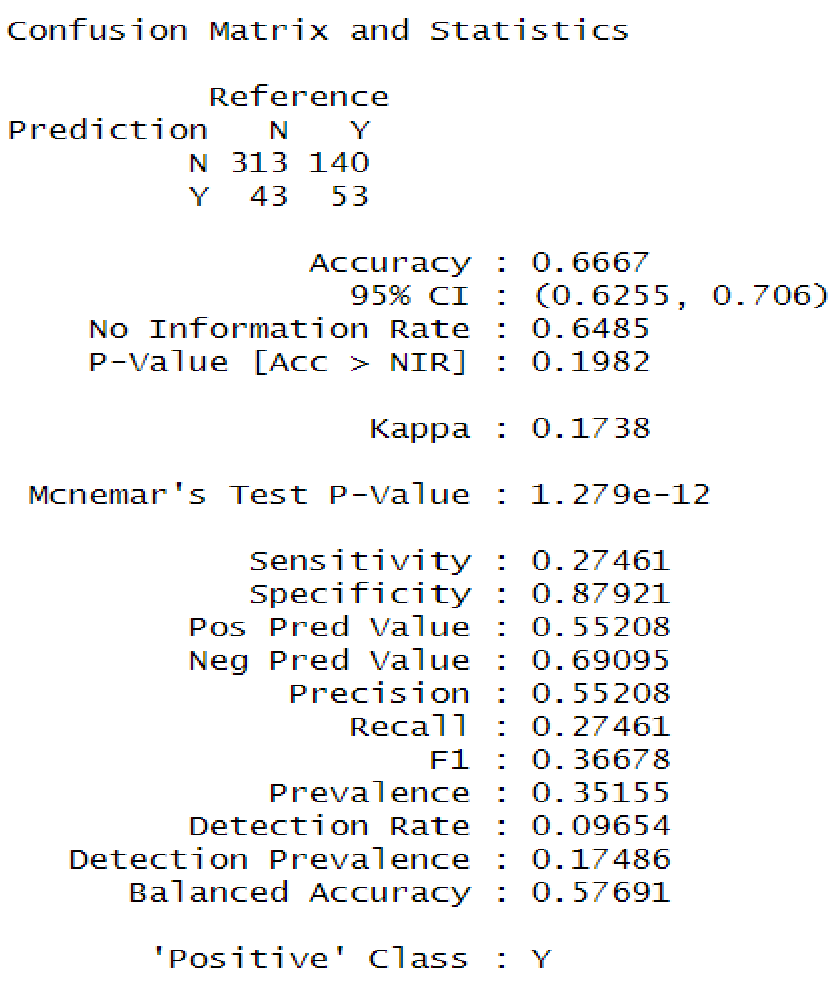
```
Class specified by attribute `outcome'

Read 2194 cases (11 attributes) from undefined.data

-----  Trial 0:  -----

Decision tree:

rating <= 3.3:
:...Duration_of_campaign > 7: N (1247/324)
:   Duration_of_campaign <= 7:
:   :...hasReddit <= 0: N (47/16)
:       hasReddit > 0: Y (28/8)
rating > 3.3:
:...teamSize <= 14: N (378/164)
    teamSize > 14:
    :...rating > 3.9: Y (229/70)
        rating <= 3.9:
        :...hasReddit <= 0: N (38/15)
            hasReddit > 0: Y (227/98)

-----  Trial 1:  -----

Decision tree:

rating <= 2.9: N (920.5/284.1)
rating > 2.9:
:...Duration_of_campaign > 303: N (25.1/3.4)
    Duration_of_campaign <= 303:
    :...teamSize <= 7: N (145.5/53.7)
        teamSize > 7: Y (1102.9/508.4)
```



```
-----  Trial 2:  -----

Decision tree:
 N (2194/924.1)

*** boosting reduced to 2 trials since last classifier is very inaccurate

*** boosting abandoned (too few classifiers)


Evaluation on training data (2194 cases):

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

             7   695(31.7%)   <<


           (a)   (b)    <-classified as
          ----  ----
          1191   176    (a): class N
           519   308    (b): class Y


        Attribute usage:

        100.00% rating
         60.26% Duration_of_campaign
         39.74% teamSize
         15.50% hasReddit
```



```
   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  549


                  | actual default
predicted default |         N |         Y | Row Total |
------------------|-----------|-----------|-----------|
                N |       313 |       140 |       453 |
                  |     0.570 |     0.255 |           |
------------------|-----------|-----------|-----------|
                Y |        43 |        53 |        96 |
                  |     0.078 |     0.097 |           |
------------------|-----------|-----------|-----------|
     Column Total |       356 |       193 |       549 |
------------------|-----------|-----------|-----------|
```



```
Confusion Matrix and Statistics

          Reference
Prediction   N   Y
         N 313 140
         Y  43  53

               Accuracy : 0.6667
                 95% CI : (0.6255, 0.706)
    No Information Rate : 0.6485
    P-Value [Acc > NIR] : 0.1982

                  Kappa : 0.1738

 Mcnemar's Test P-Value : 1.279e-12

            Sensitivity : 0.27461
            Specificity : 0.87921
         Pos Pred Value : 0.55208
         Neg Pred Value : 0.69095
              Precision : 0.55208
                 Recall : 0.27461
                     F1 : 0.36678
             Prevalence : 0.35155
         Detection Rate : 0.09654
   Detection Prevalence : 0.17486
      Balanced Accuracy : 0.57691

       'Positive' Class : Y
```

KNN

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  549


                  | wbcd_test_pred
wbcd_test_labels  |         N |         Y | Row Total |
------------------|-----------|-----------|-----------|
                N |       302 |        54 |       356 |
                  |     0.848 |     0.152 |     0.648 |
                  |     0.680 |     0.514 |           |
                  |     0.550 |     0.098 |           |
------------------|-----------|-----------|-----------|
                Y |       142 |        51 |       193 |
                  |     0.736 |     0.264 |     0.352 |
                  |     0.320 |     0.486 |           |
                  |     0.259 |     0.093 |           |
------------------|-----------|-----------|-----------|
     Column Total |       444 |       105 |       549 |
                  |     0.809 |     0.191 |           |
------------------|-----------|-----------|-----------|
Confusion Matrix and Statistics

          Reference
Prediction   N   Y
         N 301 141
         Y  55  52

               Accuracy : 0.643
                 95% CI : (0.6013, 0.6831)
    No Information Rate : 0.6485
    P-Value [Acc > NIR] : 0.6243

                  Kappa : 0.128

 Mcnemar's Test P-Value : 1.268e-09

            Sensitivity : 0.26943
            Specificity : 0.84551
         Pos Pred Value : 0.48598
         Neg Pred Value : 0.68100
              Precision : 0.48598
                 Recall : 0.26943
                     F1 : 0.34667
             Prevalence : 0.35155
         Detection Rate : 0.09472
   Detection Prevalence : 0.19490
      Balanced Accuracy : 0.55747

       'Positive' Class : Y
```