1. INTRODUCTION

New businesses are a crucial factor towards the economic development of a country. In the UK, small and medium enterprises provide three fifth of the employment and their business account for 50% of the turnover of UK companies (Kepka,2020). New businesses need significant investments to build the technology and infrastructure that can support their future goals. Funding can be obtained using personal savings, business loans, crowdfunding, venture capitals, angel investments, etc.

Crowdfunding involves fundraising online where teams present their projects to potential investors in exchange for monetary compensation. Fisch (2019) introduced Initial Coin Offerings (ICO), a cutting-edge alternative to traditional crowdfunding that eliminates the need for a middleman and quickly raise funds online. ICOs use blockchain technology based on distributed ledger technology (DLT) to raise funds by issuing digital coins/tokens known as “cryptocurrencies”. Tokens can be used as assets or utility to purchase products or services and also act as securities within the blockchain ecosystem (Sameeh,2018). Tokens can be traded amongst investors and their transactions are securely stored in a decentralized structure. An ICO’s success is determined by its ability to raise funds through a crowdfunding campaign and the value of the cryptocurrency is influenced by several factors including projects overall rating, token price and coin number.

This report aims to employ various supervised machine learning models to analyse the key factors that determine the success or failure of an ICO campaign and predict the outcome accurately. The dataset is split into training and testing sets. The training set is used to build a model that can identify predictors, while the testing set is used to evaluate the model’s performance.

2. UNDERSTANDING THE RAW DATA

2.1. Data Summary

There are 2767 observations with 16 variables. The data summary obtained using skim() provides information regarding each variable's class type along with distribution, missingness and other important statistics. There are 10 qualitative variables and 6 quantitative variables.

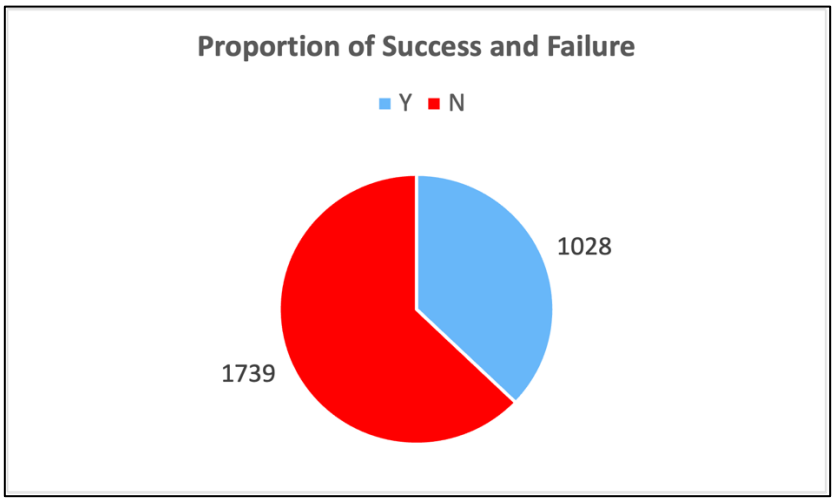
```
── Data Summary ────────────────────────
                           Values
Name                       ico
Number of rows             2767
Number of columns          16
_______________________
Column type frequency:
  character                6
  numeric                  10
_______________________
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
 1 ID                            0         1     1.38e+ 3 7.99e+ 2  1      692.
 2 hasVideo                      0         1     7.26e- 1 4.46e- 1  0        0
 3 rating                        0         1     3.12e+ 0 7.14e- 1  1        2.6
 4 priceUSD                    180         0.935 1.90e+ 1 7.75e+ 2  0        0.04
 5 teamSize                    154         0.944 1.31e+ 1 8.08e+ 0  1        7
 6 hasGithub                     0         1     5.78e- 1 4.94e- 1  0        0
 7 hasReddit                     0         1     6.33e- 1 4.82e- 1  0        0
 8 coinNum                       0         1     8.18e+12 4.30e+14 12 50000000
 9 minInvestment                 0         1     4.53e- 1 4.98e- 1  0        0
10 distributedPercentage         0         1     1.06e+ 0 1.75e+ 1  0        0.4
```

```
           p50         p75    p100 hist
1         1384         2076.  2.77e 3 ▇▁▁▁▁
2            1            1   1   e 0 ▂▁▁▁▇
3          3.1          3.7 4.8 e 0 ▂▃▇▁▂
4         0.12          0.5 3.94e 4 ▇▁▁▁▁
5           12           17   7.5 e 1 ▇▂▁▁▁
6            1            1   1   e 0 ▇▁▁▁▇
7            1            1   1   e 0 ▇▁▁▁▇
8    180000000    600000000   2.26e16 ▇▁▁▁▁
9            0            1   1   e 0 ▇▁▁▁▇
10        0.55         0.7 8.70e 2 ▇
```

2.2. Understanding the variables

2.2.1. Success

The ICO fundraising campaign follow an all or nothing approach. Success is the target variable. If the ICO can raise the required fund within a specific time, they will be deemed as a success or else they will return without anything. Success is represented as 'Y' and failure as 'N'.



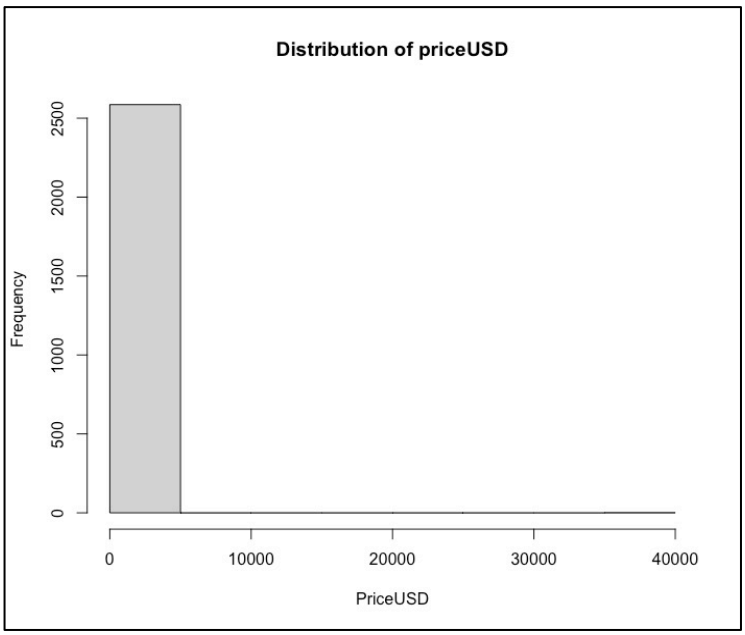
The number of failures is more than that of success in the dataset. This imbalance in data must be dealt accordingly, to ensure that the model is not biased towards the majority class.

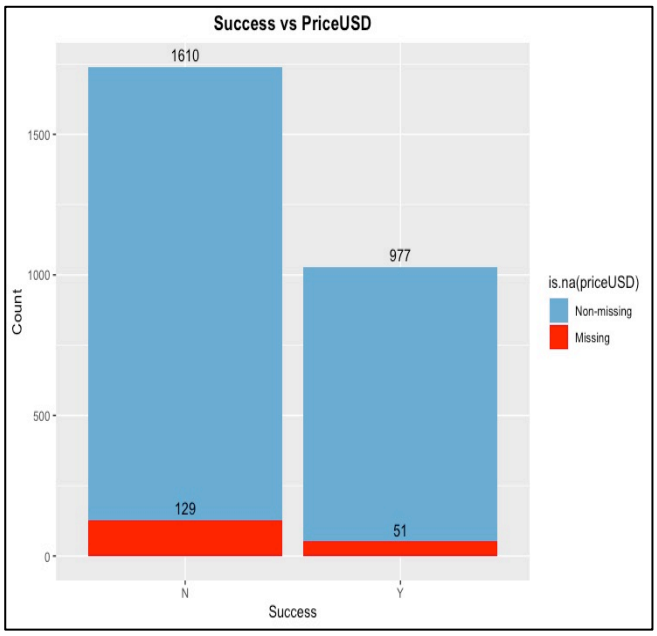
2.2.2. Brand Slogan

This variable represents the slogans used by the fundraising team. The most frequently occurring terms in the slogans have been displayed, with the minimum frequency set to 50.

2.2.3. PriceUSD

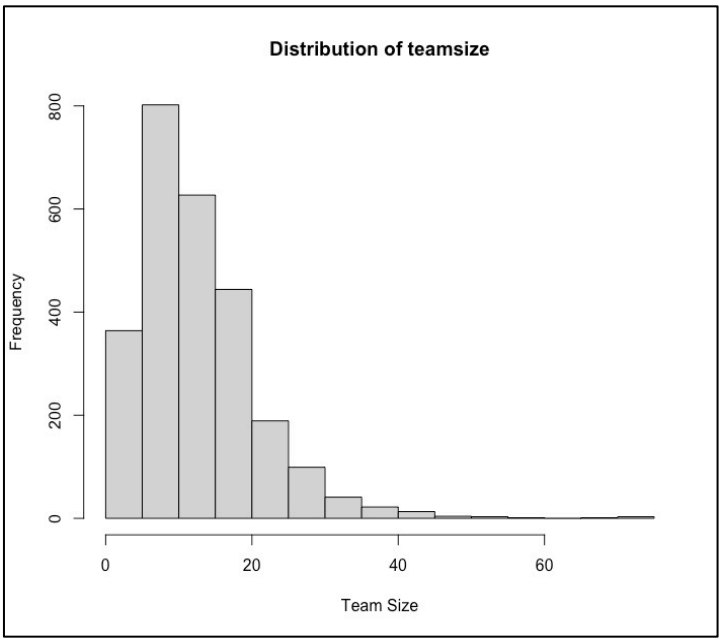
This variable indicates the price of each token issued in US dollars. The data is heavily right skewed with 180 missing values and needs to be treated appropriately. The minimum price is zero. If the price set to zero is leading the project to success, then it is assumed that the team is giving away coins for free as a promotional activity.

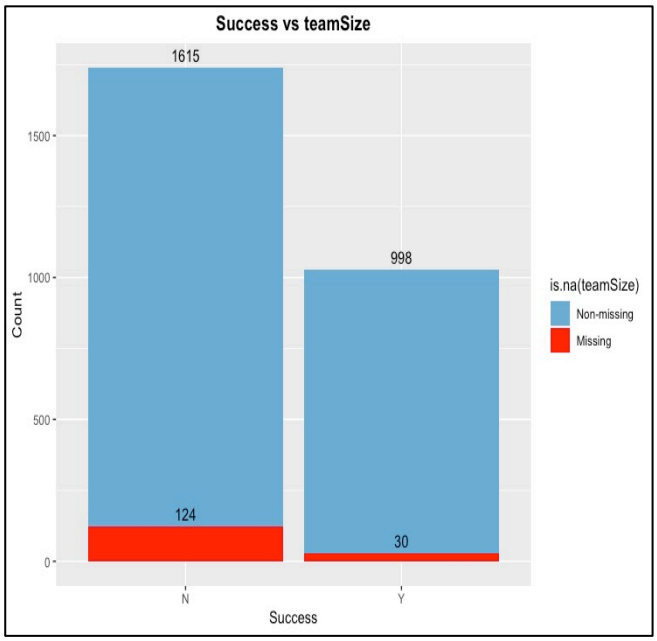




2.2.4. TeamSize

Having a dedicated team is an important indicator for attracting potential investors. This variable contains 154 missing values. The average team size is 13 and the overall size ranges from 1 to 75.

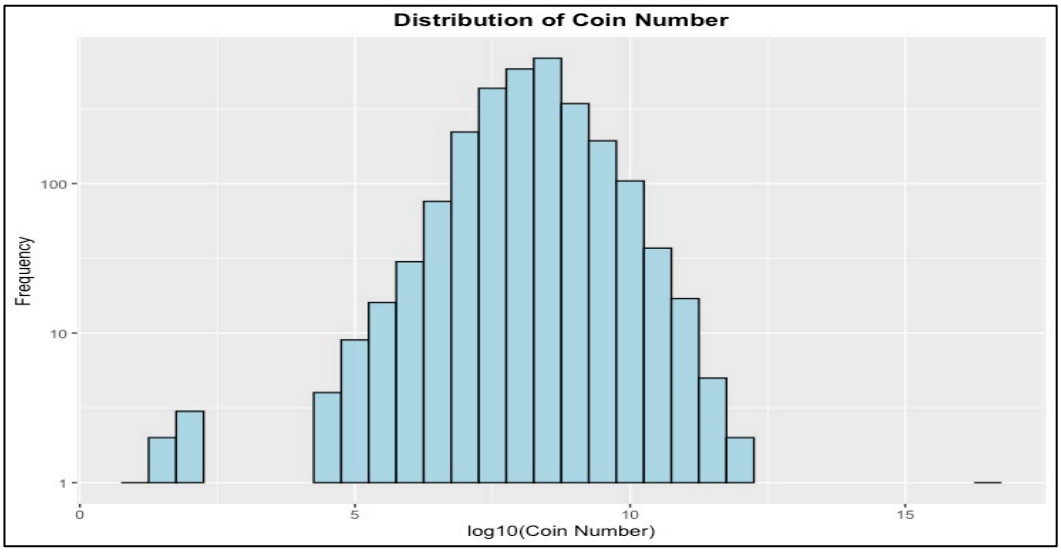




2.2.5. CoinNum

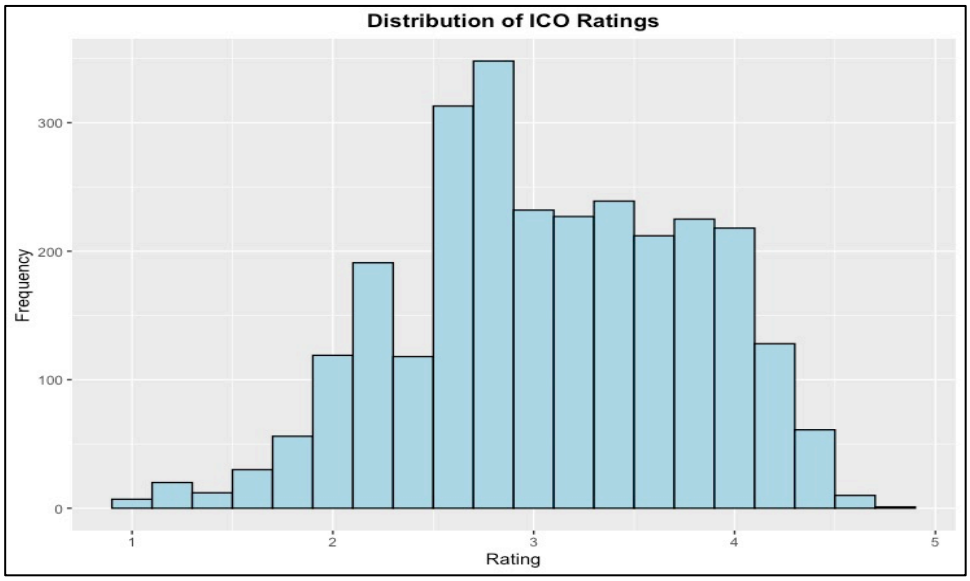
If the number of coins issued by the team is too high, then the value of the coin will decrease which may reduce the incentive of the investor to participate in the campaign. If the number of coins is small, the value of the cryptocurrency will be of high demand.

Since the values are very high, the data has been converted to logarithmic scale for ease of visualisation. There are some outliers in coinNum, yet these are not removed since the team can freely decide on the number of coins.

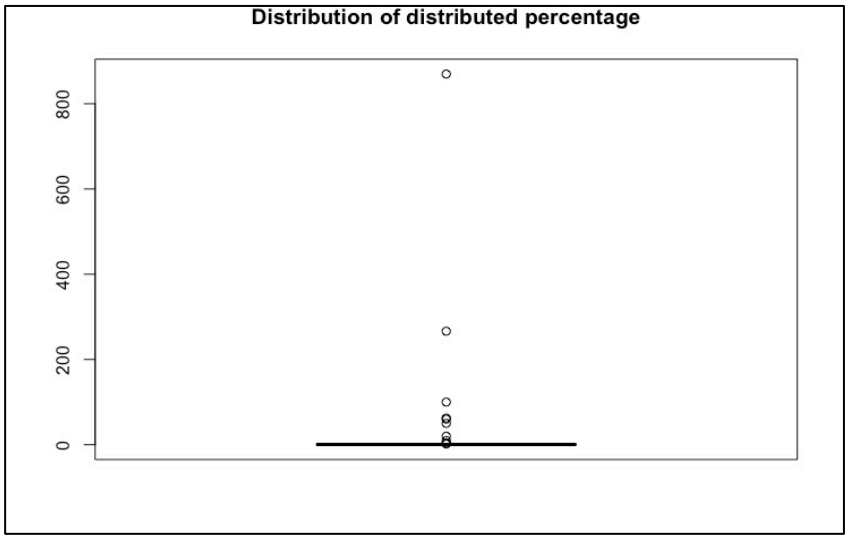


2.2.6. Rating

The overall rating given by investment experts will help investors in evaluating the quality of the ICO project. The rating is between the range 1(poor) and 5(very good). Majority of the rating is between 2 to 4 with the maximum being 4.8. The data is normally distributed.



2.2.7. Distributed Percentage



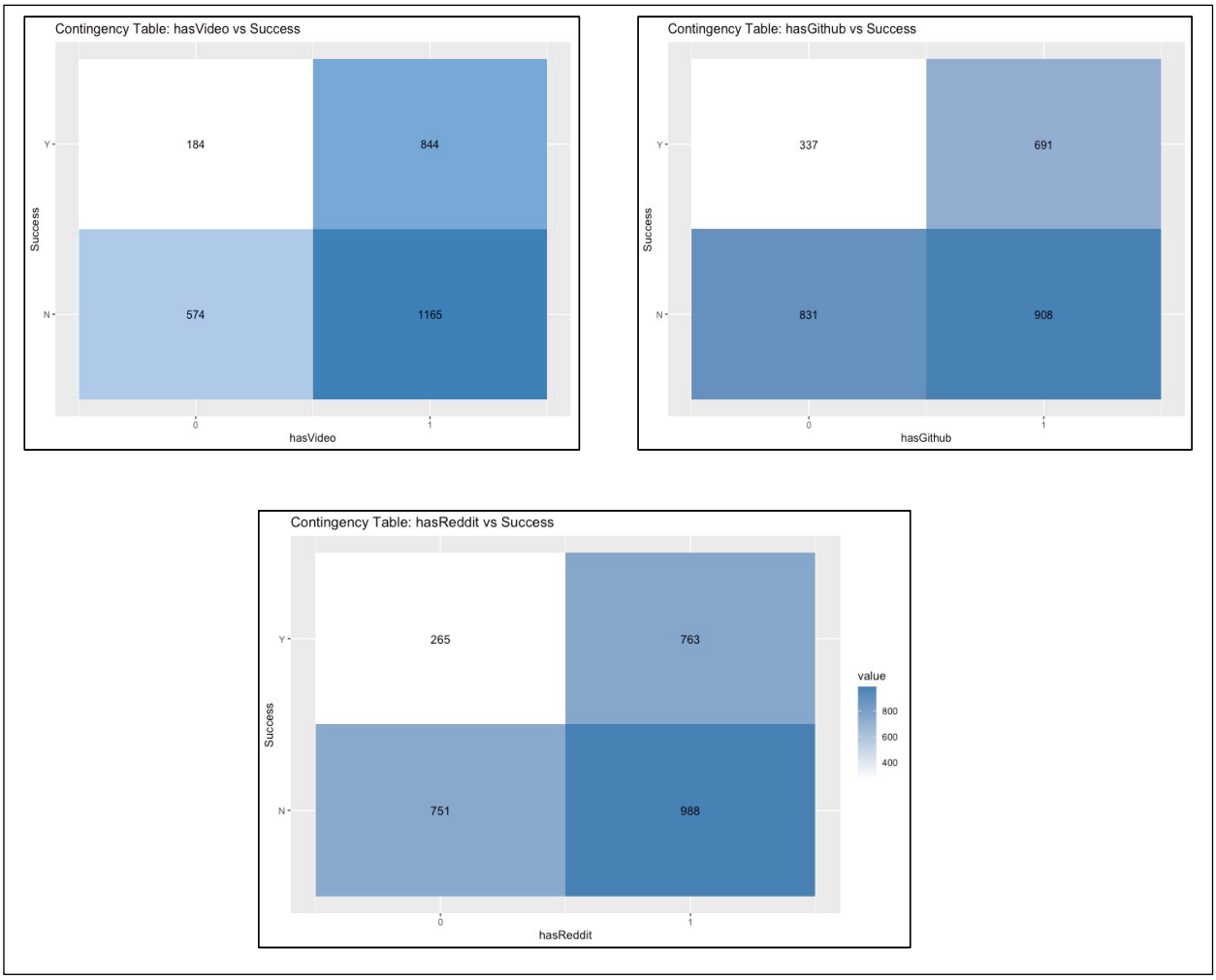
This feature indicates the percentage of coins distributed to the investors, relative to the total number of coins generated. Retaining many coins with the team, indicates the team's commitment towards the success of the project. Certain records of the data are above 100%, which is not appropriate.

Distributed percentage above 1

ID	coinNum	platform	distributedPercentage
99	10000000	Ethereum	1.66
542	4000000	NEO	4.00
681	20000000	Waves	9.52
947	1308800000	Ethereum	62.50
965	6656250	X11	266.25
1030	17395000	Ethereum	869.75
1404	3000000	Ethereum	20.00
1408	1000000	Ethereum	100.00
1656	50000000	Ethereum	50.00
1988	3000000	Ethereum	60.00

2.2.8. Social Media (hasVideo, hasGithub, hasReddit)

These are binary variables representing whether the team is using these social media platforms for promoting their campaign. Contingency table examines the relationship between success and each of the social media platforms.

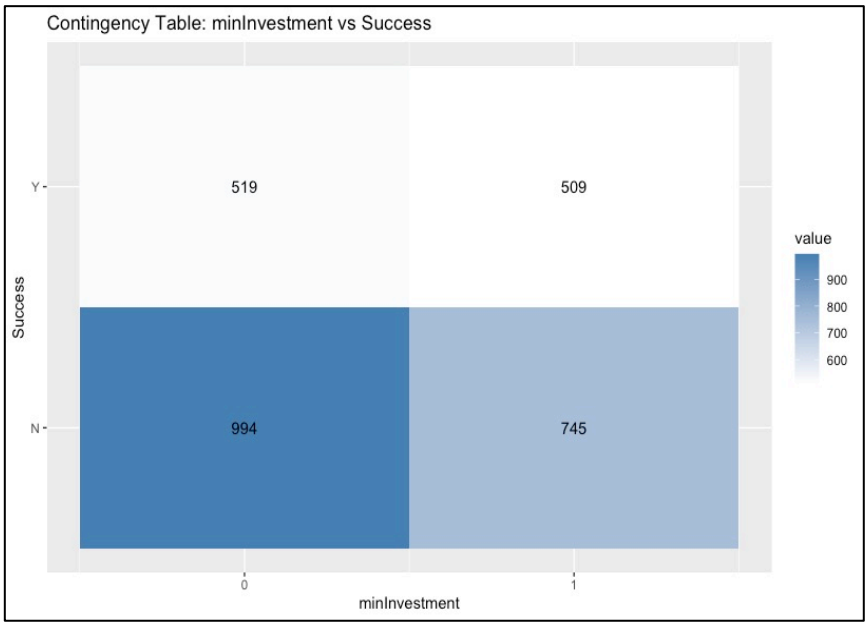


Using the contingency table, the probability of success given these social media platforms is calculated. The table suggests that the impact of social media on the success of the ICO project is 49.3% if all the platforms are used.

Social Media Platform	Probability of success of ICO project
hasVideo = 1	42.01%
hasGithub = 1	43.21%
hasReddit = 1	43.58%
hasVideo=1 & hasGithub=1 & hasReddit = 1	49.3%

2.2.9. minInvestment

This binary variable indicates whether the team has a minimum investment or not. The probability of success of ICO project with minimum investment is 40.59%.



2.2.10. CountryRegion

These are the country of origin of each fundraising team. There are 71 unknown countries. There are anomalies in the data, which needs to be corrected during the data preparation process e.g., usa and USA refer to the same country.

Syria	Tanzania
1	2
Thailand	Timor-Leste
13	1
Tunisia	Turkey
1	16
UK	Ukraine
285	19
United Arab Emirates	usa
41	1
USA	Vanuatu
296	2
Venezuela	Vietnam
2	7
Zimbabwe	
1	

2.2.11. Platform

There are 130 unique blockchain platforms. Observing the top 10 platforms according to the frequency of occurrence, it is evident that some of the platforms are not unique, E.g. Ethereum has been spelled incorrectly.

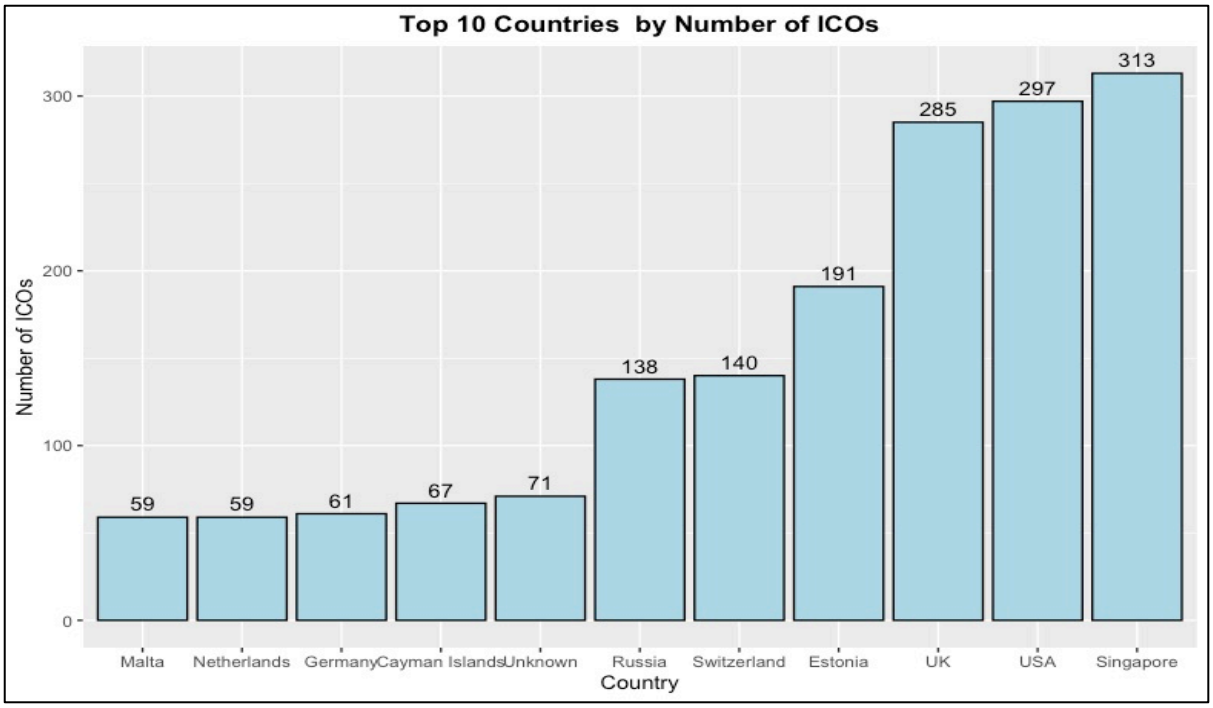
Ethereum	Waves	Stellar
2352	56	41
Separate blockchain	Ethereum	NEO
30	23	20
EOS	Ethereum	Scrypt
16	16	14
NEM	Bitcoin	Ethereum
12	10	9
X11		Separate Blockchain
8	6	6
Graphene	Tron	BTC
4	4	3
DPoS	ICON	Komodo
3	3	3
NXT	TRON	Bitshares
3	3	2
ERC20	Ethererum	Ethereum
2	2	2

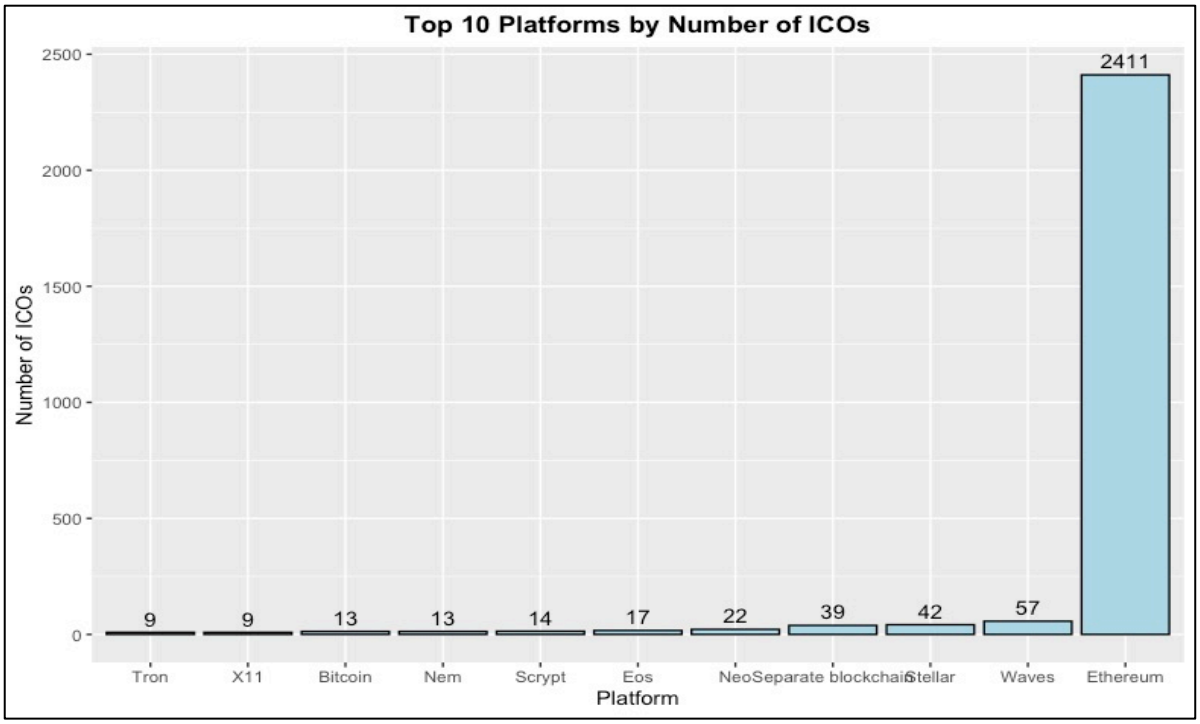
3. DATA PREPARATION

The raw data is prepared by removing the anomalies, outliers and imputing the missing values for providing appropriate data to the model.

3.1. Correcting the spelling mistakes and data types in variables

- Start date and end date columns are not formatted correctly. This is converted to Date format.
- The countryRegion and platform variables are cleaned by removing the spelling mistakes, symbols, punctuations, and unnecessary white spaces. The blank values were replaced with “Unknown”.
- The cleaned data consists of 116 unique countries and 96 unique platforms. It is observed that 87% of the ICO projects use Ethereum as the blockchain platform.

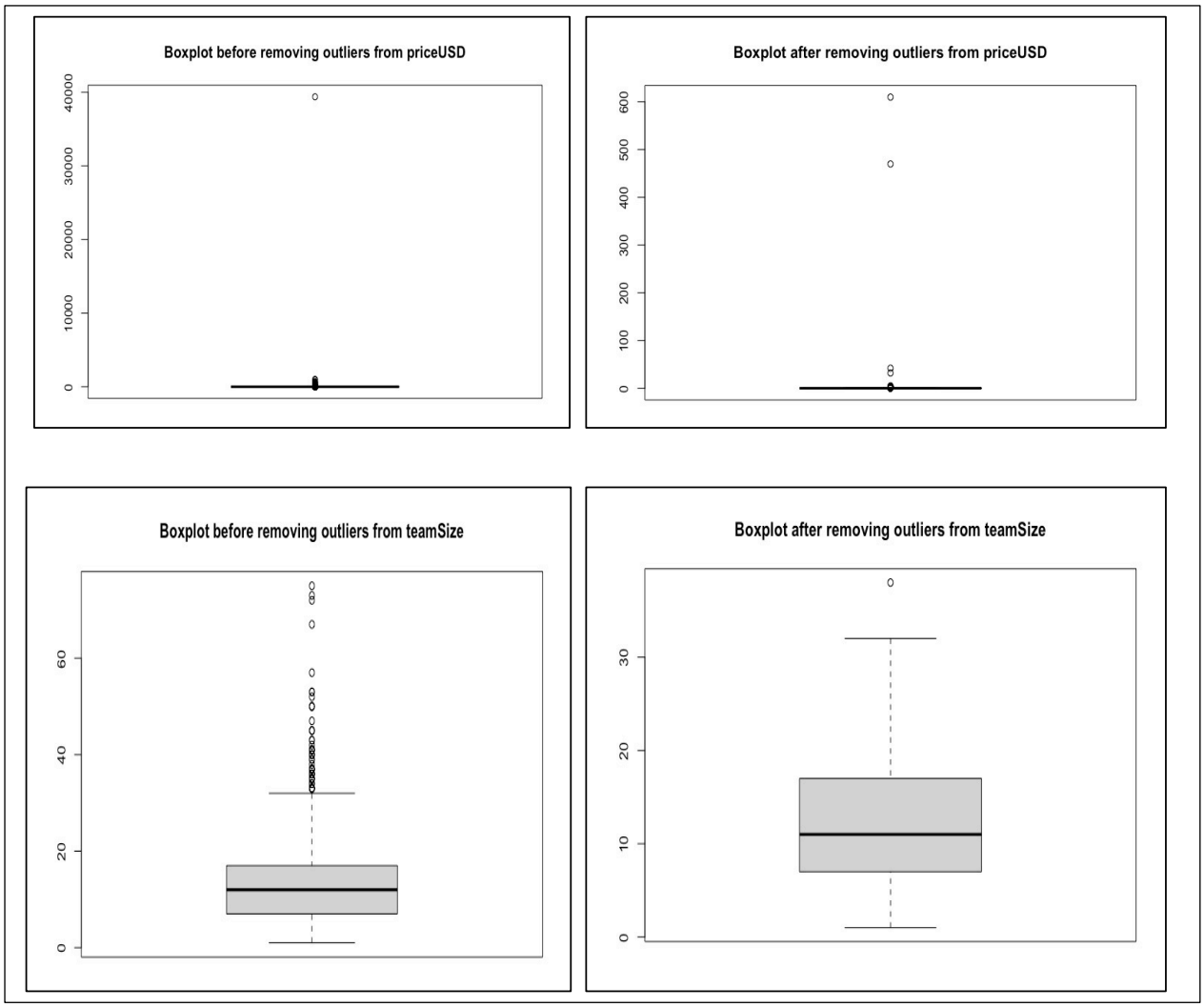




3.2. Outlier Detection

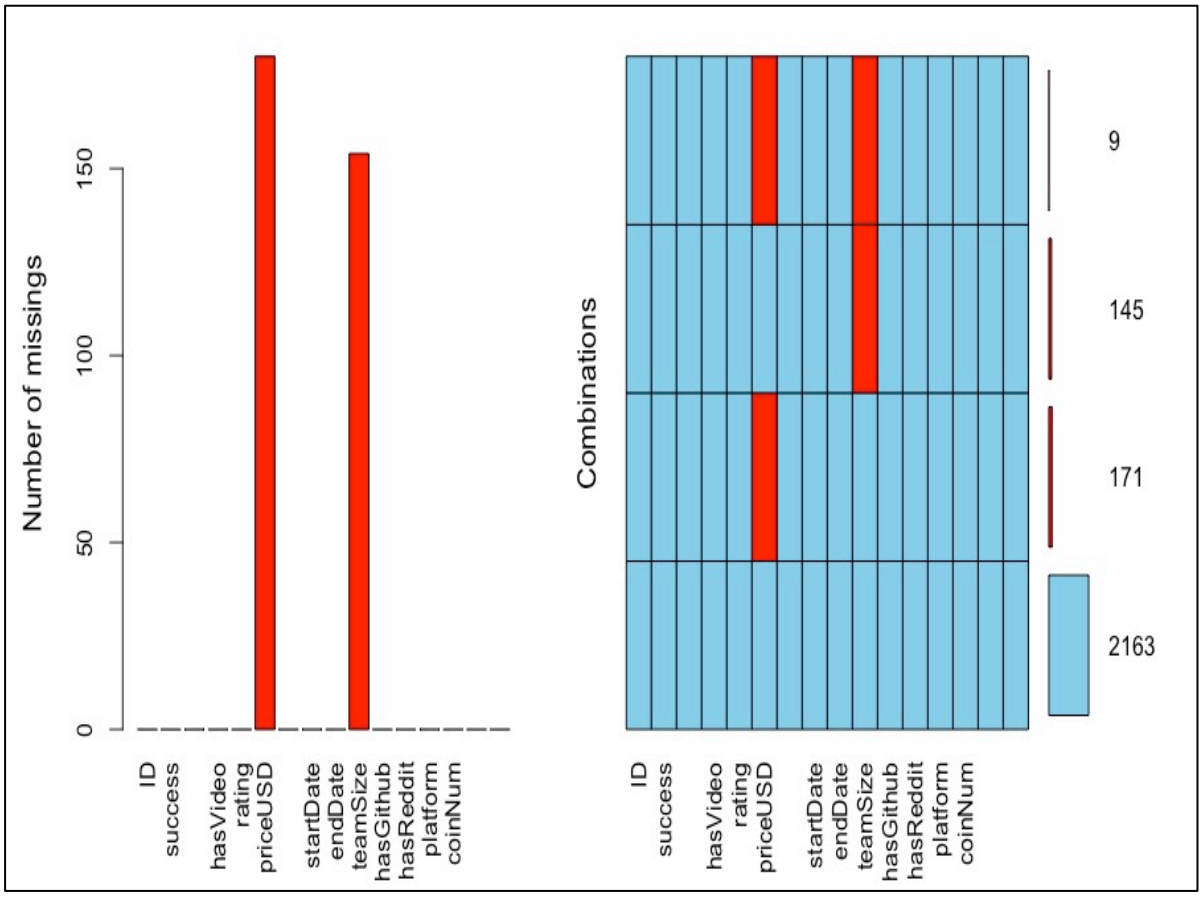
IQR method removes outliers using the interquartile range (IQR). Outliers are values outside the range [Q1 - 1.5 * IQR, Q3 + 1.5 * IQR] where Q1 and Q3 corresponds to the first and third quartile respectively. This range covers 99.3% of the data

(Chaudhary, 2020). IQR is robust with missing values, hence has been used for outlier removal from priceUSD and teamSize. Outlier removal is done by preserving the missing values.

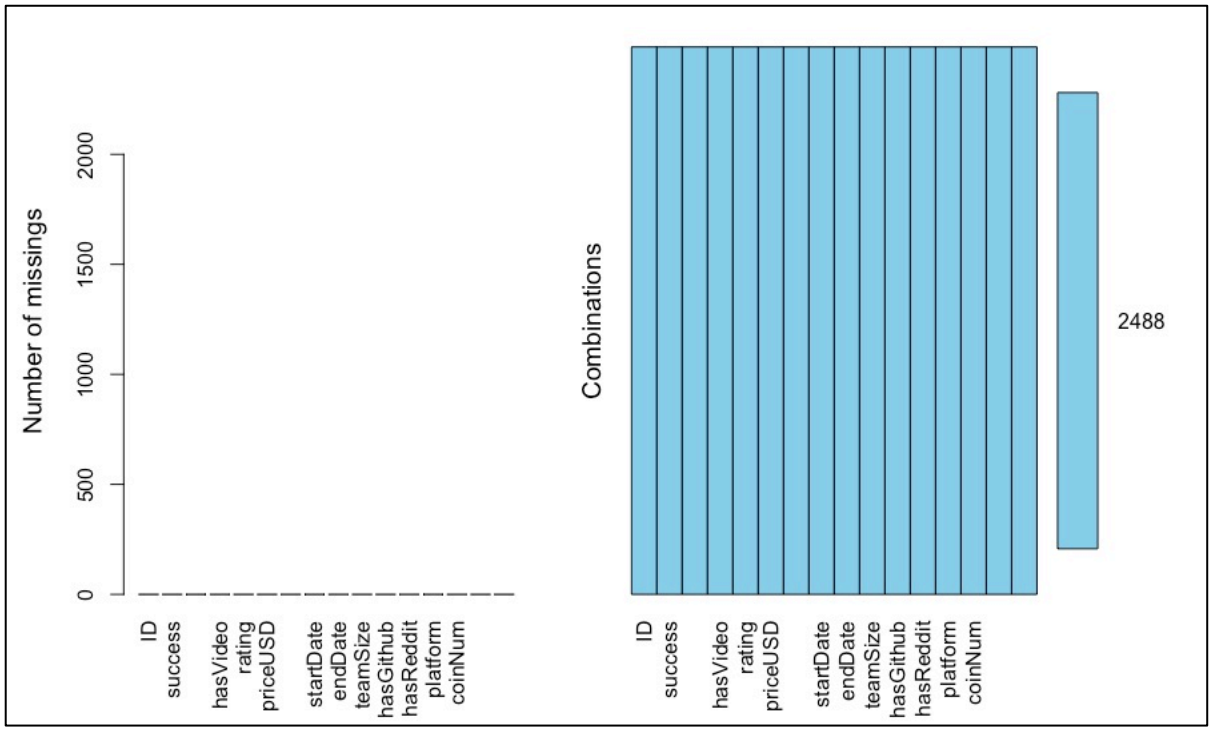


3.3. Imputing Missing Values

Multiple imputation is used to impute the missing values in priceUSD and team size. Multiple imputation involves substituting values using chained equation approach, resulting in complete cases. Multiple imputation is preferred over simple imputation since these variables were missing at random (MAR) (Dziura et al., 2013, p.350). It is ensured that the distribution of data before and after imputation are similar. Figure below, shows the summary of data before and after multiple imputation. The red blocks indicate the missing data and blue indicates the complete data.



Before multiple imputation



After multiple imputation

3.4. Additional Filtration

- The priceUSD column still had extreme outliers. Two values above 400 were removed.
- The minimum value for the price of token was zero. If the price set to zero is not leading the project to success, then these are not significant, hence those records were removed. If the price set to zero led the project to success, then it is assumed to be a promotional activity and these records were retained.
- All observations with distributed percentage above 100% was removed.

3.5. Variable Creation

- Project Duration: Signifies the duration for which the campaign lasted. Some dates were entered incorrectly, causing negative durations which were eliminated.

ID	startDate	endDate	project_duration
42	2018-06-28	2018-06-08	-20
164	2019-08-16	2019-01-25	-203
368	2019-06-17	2019-06-09	-8
1389	2018-11-21	2018-11-20	-1
1753	2018-06-16	2018-04-16	-61
1794	2020-05-21	2020-03-30	-52
1835	2018-10-11	2018-03-29	-196
1923	2019-11-16	2019-10-22	-25

Negative durations

- Goal Amount: Calculated as the product of priceUSD and coinNum, represents the total amount the fund-raising team is aiming to raise through the campaign. This is an important factor that will help investors to assess their potential return on investments (Huang, Vismara and Wei, 2022).
- Investor Coins: Calculated as the product of coinNum and distributedPercentage represents the number of block chain coins that will be distributed to the investors if the goal is met.
- Continent: Represents the continents with respect to the countries of each fundraising team. This variable is created since some countries occur only once, which will bias the model.

```
"Africa"   "Americas" "Asia"     "Europe"   "Oceania"  "Unknown"
```

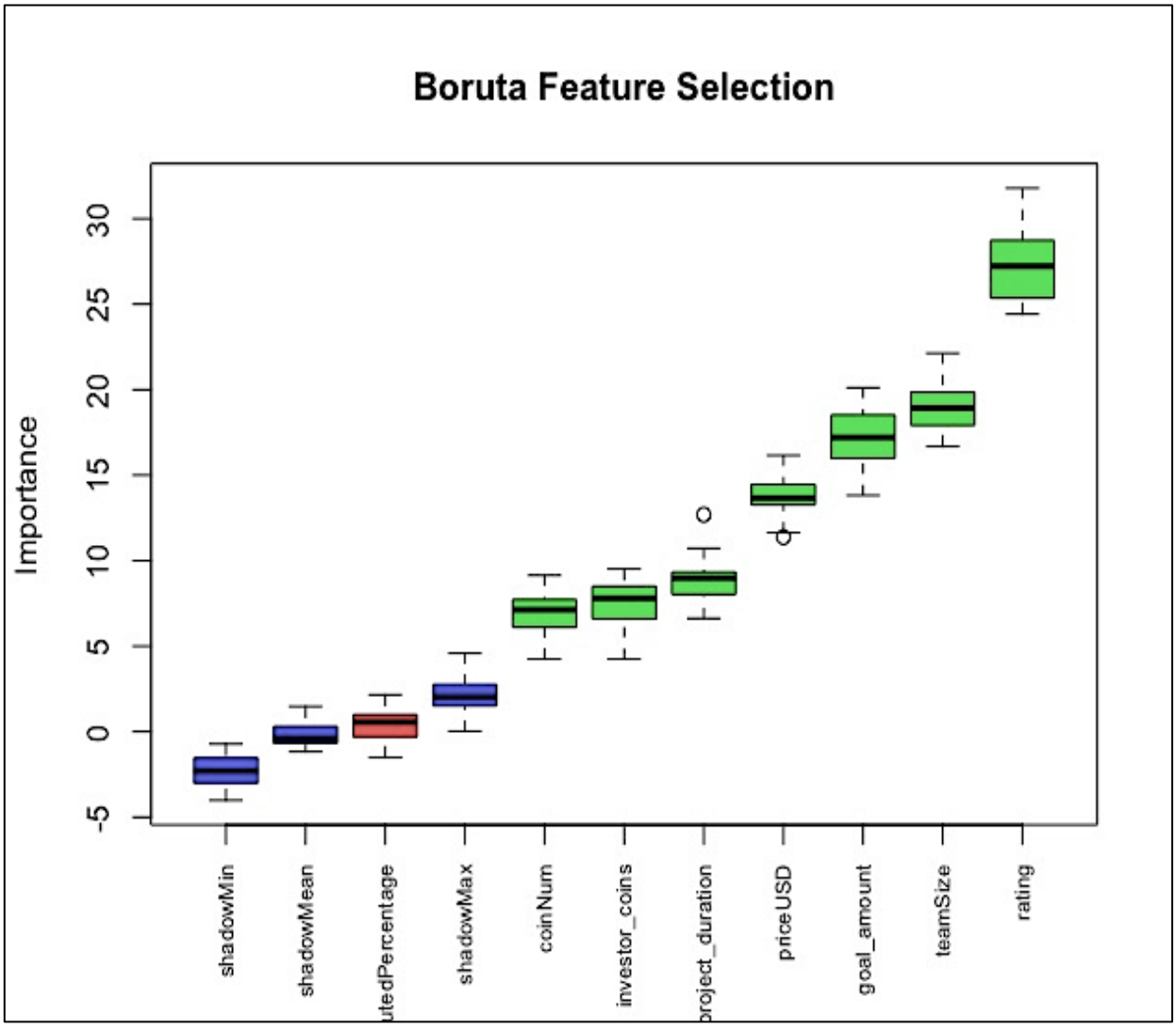
Unique Continents

4. FEATURE SELECTION

Feature selection is a method of reducing the dimensionality of the dataset when developing a classification model for reducing the computational cost of the model as well as for improved performance.

4.1. Feature Selection for Numeric Variables

The Boruta algorithm is a feature selection method based on random forest. It creates shadow features by shuffling the original features and trains a random forest on them. It then compares the importance scores of the original and shadow features and marks the original features that have higher scores as important. This process is repeated for numerous iterations and the significance of the hits is tested using a binomial distribution(Kursa and Rudnicki,2010). Rating has the highest importance and distributed percentage is rejected.



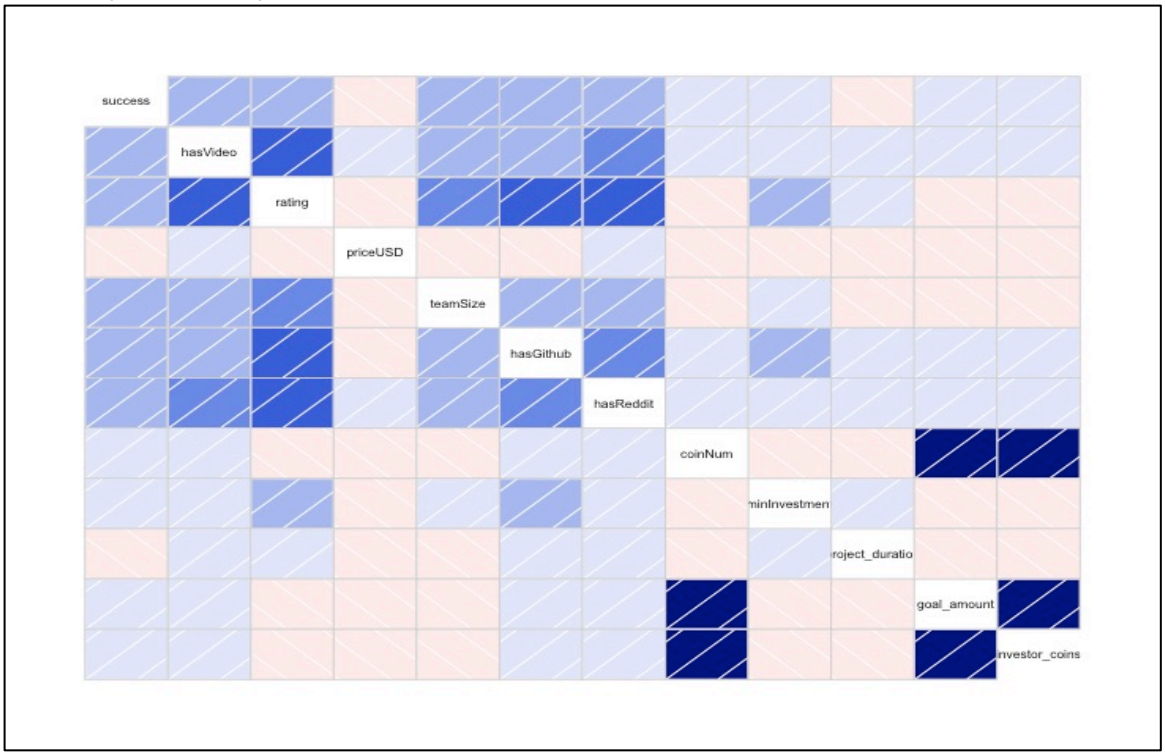
4.2. Feature Selection for categorical variables

Chi-square test is used for analysing the importance of categorical variables with success. The null hypothesis states that the two variables are significantly different. If the p-value is below 0.05 then the null hypothesis is “rejected”, signifying that the variables exhibit some relationship and is considered for further analysis (Lantz,2019).

Variable	Chi-squared	p-value	Null hypothesis
Platform	88.373	0.3509	Accepted
Brand Slogan	2359.8	0.4973	Accepted
Country Region	158.35	0.001155	Rejected
Continent	23.275	0.000299	Rejected

5. ANALYSING CORRELATION BETWEEN VARIABLES IN FINAL DATA

Success is converted to binary variable (0-N, 1-Y) for correlation analysis using corrgram(). All variables except for priceUSD and project duration (light red colour) have positive correlation (blue colour) with success.



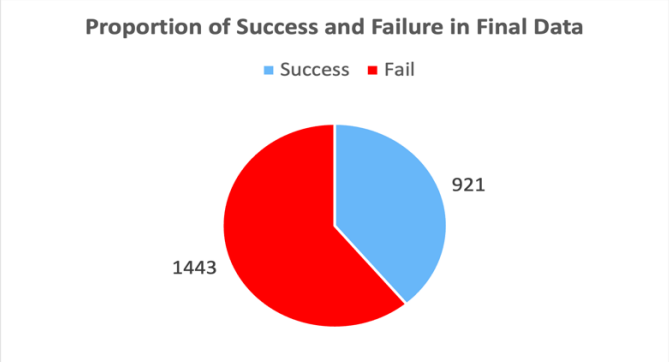
6. FINAL DATA

The final set of variables have been determined based on the results of feature selection. CountryRegion and continent had significant relationship with success, hence only continent is considered. The final dataset consists of 2364 observations and 13 variables.

```
── Data Summary ────────────────────────
                           Values
Name                       final_data
Number of rows             2364
Number of columns          13
_______________________
Column type frequency:
  character                1
  factor                   1
  numeric                  11
________________________
Group variables            None

── Variable type: character ──────────────────────────────────────────────────────
  skim_variable n_missing complete_rate min max empty n_unique whitespace
1 continent             0             1   4   8     0        6          0

── Variable type: factor ─────────────────────────────────────────────────────────
  skim_variable n_missing complete_rate ordered n_unique top_counts
1 success               0             1 FALSE          2 Fai: 1443, Suc: 921

── Variable type: numeric ────────────────────────────────────────────────────────
   skim_variable    n_missing complete_rate     mean       sd p0         p25
 1 hasVideo                 0             1 7.24e- 1 4.47e- 1  0        0
 2 rating                   0             1 3.12e+ 0 7.19e- 1  1        2.6
 3 priceUSD                 0             1 3.03e- 1 1.14e+ 0  0        0.04
 4 teamSize                 0             1 1.27e+ 1 6.58e+ 0  1        8
 5 hasGithub                0             1 5.80e- 1 4.94e- 1  0        0
 6 hasReddit                0             1 6.40e- 1 4.80e- 1  0        0
 7 coinNum                  0             1 9.57e+12 4.65e+14 45 52000000
 8 minInvestment            0             1 4.53e- 1 4.98e- 1  0        0
 9 project_duration         0             1 7.04e+ 1 1.04e+ 2  0       29
10 goal_amount              0             1 9.64e+10 4.65e+12  0  7629280
11 investor_coins           0             1 4.79e+12 2.33e+14  0 25099904.
          p50          p75    p100 hist
 1         1           1    1   e 0 ▂▁▁▇
 2         3.1         3.7  4.7 e 0 ▂▃▇▅▂
 3         0.1         0.39 4.22e 1 ▇▁▁▁▁
 4        12          17    3.8 e 1 ▆▇▅▂▁
 5         1           1    1   e 0 ▇▁▁▁▇
 6         1           1    1   e 0 ▇▁▁▁▇
 7 196627777   550000000    2.26e16 ▇▁▁▁▁
 8         0           1    1   e 0 ▇▁▁▁▇
 9        46          90    3.72e 3 ▇▁▁▁▁
10  21600000    50000000    2.26e14 ▇▁▁▁▁
11  89890000   302500000    1.13e16 ▇▁▁▁▁
```

7. CLASSIFICATION MODELS

Classification is a supervised learning method, where the model tries to predict the correct label of a given input data.



The data is stratified using caret::createDataPartition() to ensure that the distribution of classes in the training and test datasets are similar to that in the original dataset. 80% of the data is given for training and 20% is given for testing the model. For models that work only with numerical variables, the categorical variable continent is one-hot encoded.

7.1. Decision Tree

Decision trees use recursive partitioning to divide data into smaller subsets of homogenous classes. C5.0 algorithm is used to produce the decision tree. The root node is priceUSD and continues to split until the leaf nodes represent homogenous classes.

The best split is determined using information gain, which calculates the entropy before and after the split. Entropy is the randomness within a set of class values. The higher the information gain, the more distinct are the features inside each group compared to others (Lantz,2019).

```
Attribute usage:

100.00% priceUSD
 97.78% rating
 79.23% project_duration
 54.02% teamSize
 38.32% continent
 22.30% hasGithub
 15.86% minInvestment
 14.22% hasReddit
  5.29% hasVideo
  0.85% coinNum
```

The decision tree produced an accuracy of 68.22%. The model begins with the rule that the ICO project is successful, if the priceUSD is less than or equal to 0. This suggests that setting the price of coins to zero, can have a positive relationship to its success. Yet, it is important to note that correlation does not imply causation. This information is based on the pattern in the training data and the success of the ICO project is not solely based on price of the coins. The model considers the rating of the ICO as an important factor (97.78%). ICOs with rating more than 2.9 is likely to succeed. The importance of project duration and team size, indicates that investors may value ICOs with teams having greater expertise and longer preparation periods. Larger teams based in America and Asia have potential success (refer tree structure in appendix).

7.2. Adaptive Boosting of Decision Tree

Adaptive boosting is an ensemble-based method of building many decision trees where the trees vote on the best class for each example. Boosting is done in C5.0 algorithm by adding the parameter called trials which indicates the maximum number of decision trees to be used and helps to prevent overfitting of the model. The algorithm will stop adding trees if the accuracy is not improving by the addition of trials (Lantz,2019). The best parameters was determined by tuning the decision tree model by specifying the grid with model=”tree”, trials=c(5,10,15,20) and winnow=FALSE. The decision tree boosted with 15 trials produced an accuracy of 67.16% with average tree size as 8.5. The model suggests hasVideo, rating, priceUSD and goal amount as the most important factors for the success of an ICO campaign.

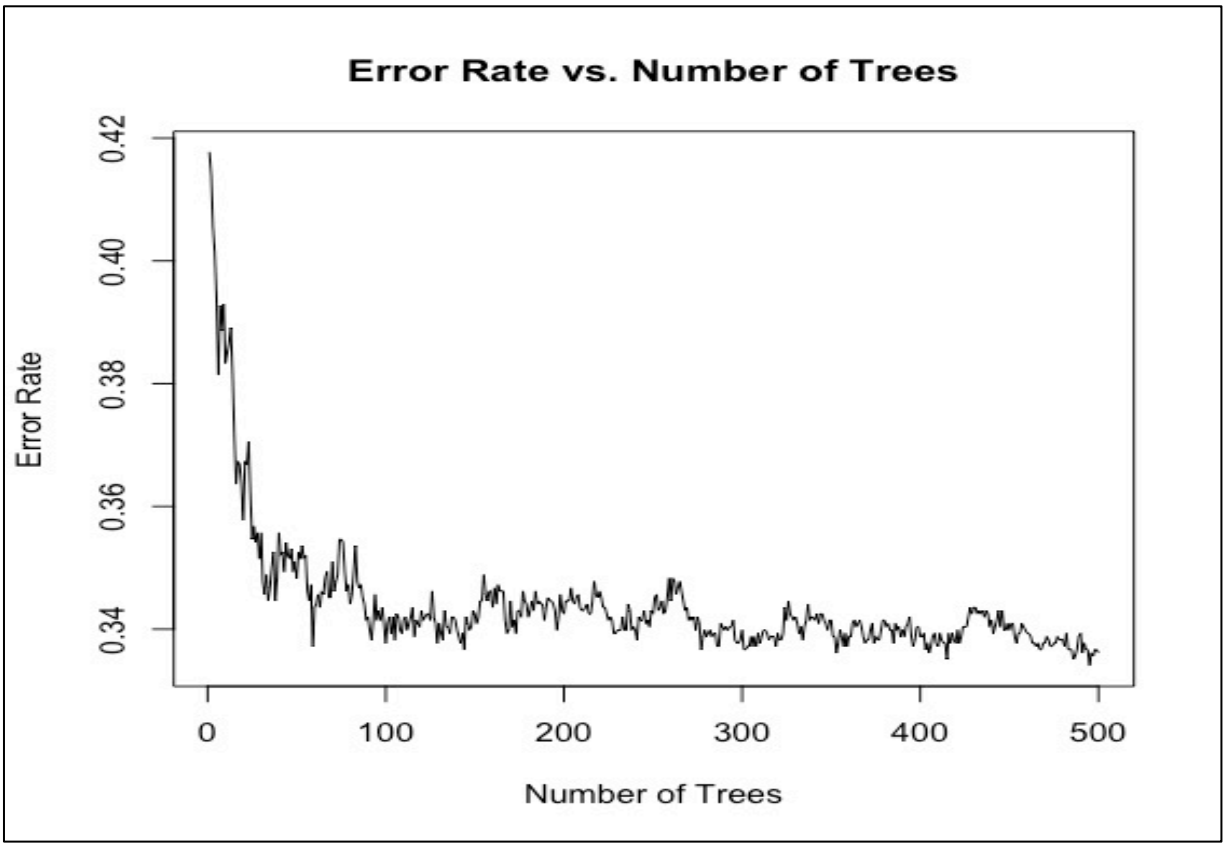
```
Attribute usage:

100.00% hasVideo
100.00% rating
100.00% priceUSD
100.00% goal_amount
 99.74% teamSize
 99.74% project_duration
 98.41% continent
 73.52% hasGithub
 71.46% hasReddit
 43.92% minInvestment
 21.35% coinNum
  7.61% investor_coins
```

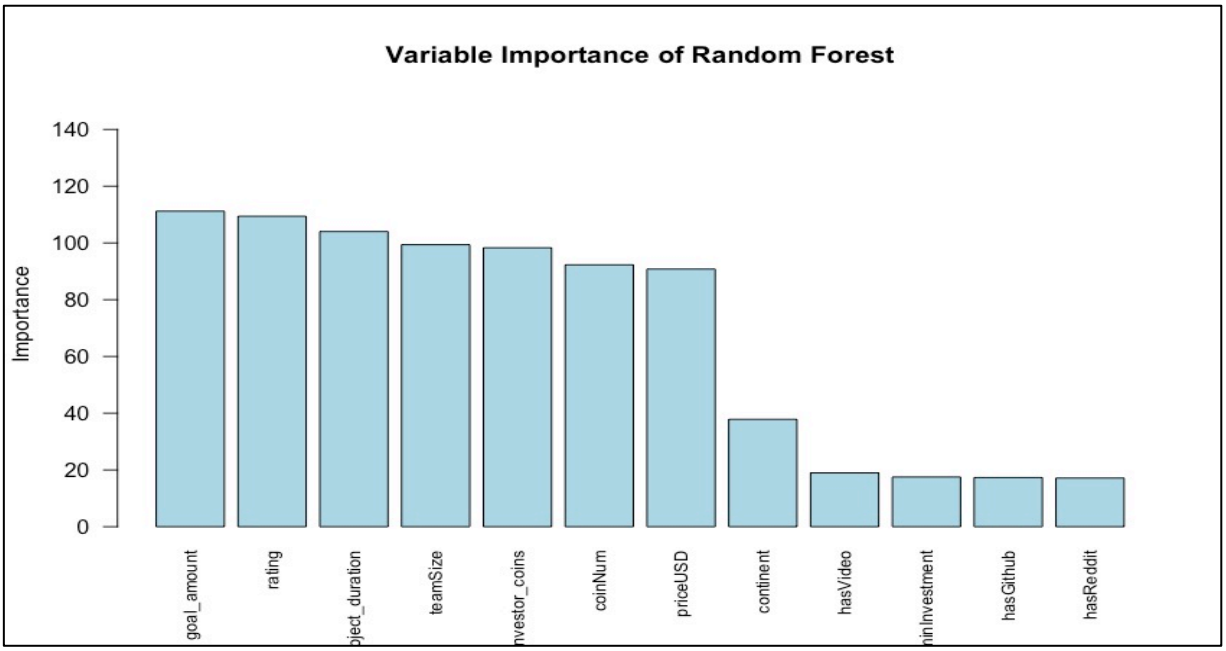
7.3. Random Forest

The ability to handle data with large number of features, noisy data as well as missing data, makes random forest models unique. They are less prone to overfitting since the trees select only the important features for training, which makes them easier to use (Lantz,2019).

The model calculates the average error rate of success and failure with respect to the number of trees. Random forest also gives the variable importance with goal amount having the highest importance followed by rating. PriceUSD had less importance (90%) compared to the other decision tree results.

The best parameters were determined by tuning the model by specifying the grid with ntree = c(100,200,300,400,500) and mtry = c(2,4,6,8), where ntree is the number of decision trees to be included and mtry controls the number of randomly selected features considered for splitting at each node. The random forest model tuned with ntree=500 and mtry=2, produced an accuracy of 68.01%.





7.4. Support Vector Machine (SVM)

Support Vector Machines identify a decision boundary called hyperplane that creates homogeneous partitions of data points. The wider the margin, the lower the generalisation error. Support vectors are the closest points to the maximum margin hyperplane (MMH). Assuming that a non-linear relationship exists between variables, kernel trick is applied (Lantz,2019). SVM with linear kernel produced an accuracy of 69.49% whereas Gaussian radial basis kernel provided the highest accuracy of 69.92%. The agreement variable represents the proportion of cases where the model's predicted type matches the actual type.

```
agreement_vanniladot
    FALSE      TRUE
0.3050847 0.6949153
```

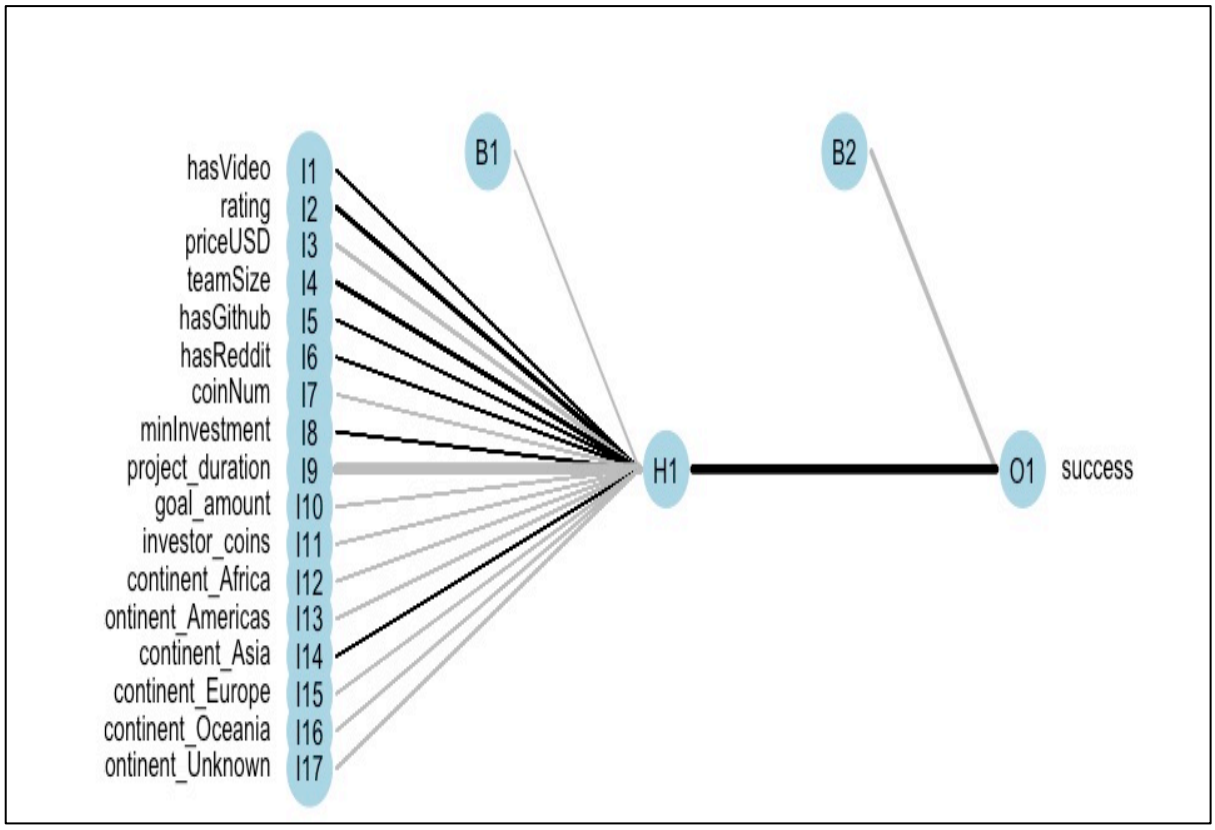
```
agreement_rbfdot
    FALSE      TRUE
0.3008475 0.6991525
```

7.5. K-Nearest Neighbours (k-NN)

The k-NN algorithm assigns a class to unlabelled data by using the majority of the k nearest (similar) examples from the labelled data. Due to this reason, it is referred to as a lazy learner. The data have been rescaled using min-max normalization before feeding it to the model. The similarity between two examples is calculated using Euclidean distance method (Lantz,2019). The value of k=43 represents the square root of the training data which determines how well the model generalises future data. An odd number of k is used to avoid tie votes. The model gave an accuracy of 68.86%.

7.6. Artificial Neural Networks (ANN)

ANNs are versatile learners inspired from human brain activity that can be applied to any machine learning task. The data is rescaled using min-max normalization and is provided to the model for training. The best parameters were determined by tuning the model by specifying the grid with size=c(1,2,3,4,5) and decay=c(0.1,0.01,0.001), were size determines the number of nodes to be used in a hidden layer and weight decay is a regularization technique used to prevent overfitting of the model. The neural network trained with size=1 and decay=0.1 produced an accuracy of 68.86%.



8. PERFORMANCE EVALUATION

Different metrics including accuracy and precision on testing sets with 80-20 split are used for performance evaluation. Accuracy represents the proportion of correctly classified ICO campaigns out of the total predictions, while precision represents the proportion of true positives out of the total positive predictions. Precision helps in examining the reliability of the model in correctly identifying successful projects.

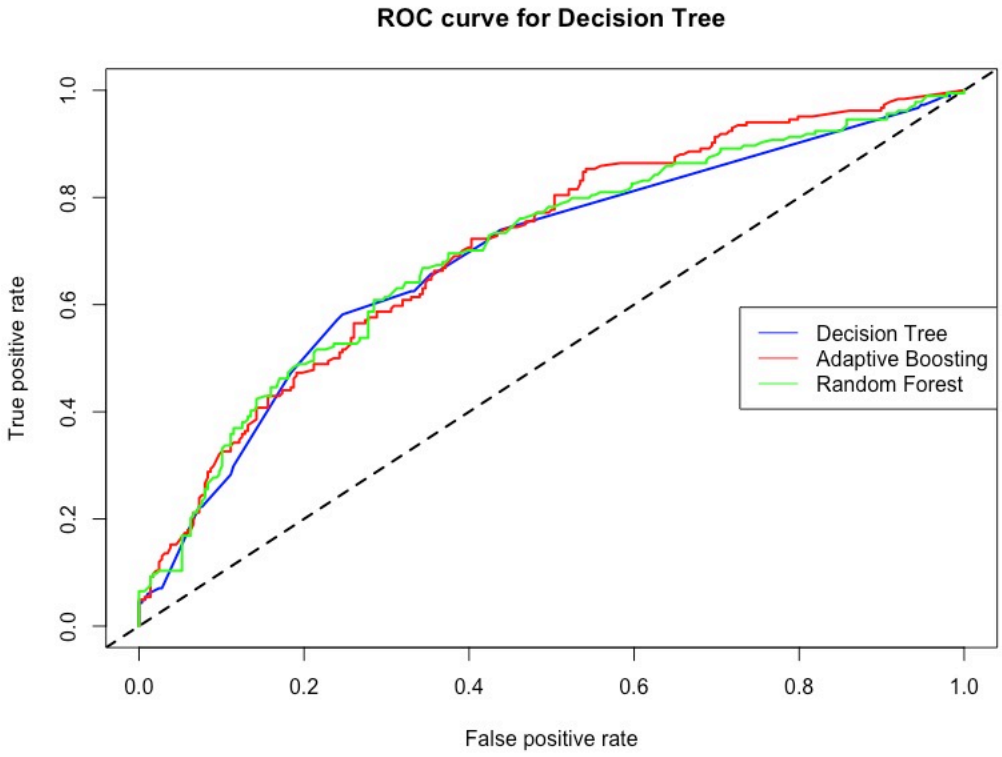
The main concern being prediction of the overall success of the ICO team in reaching its fundraising goal, accuracy is of atmost importance followed by precision which confirms that the model predictions for successful projects are accurate.

Additionally, averaged values of all metrics for 10-fold cross validation have been reported, which creates 10 folds of the entire data split using 90-10 ratio.

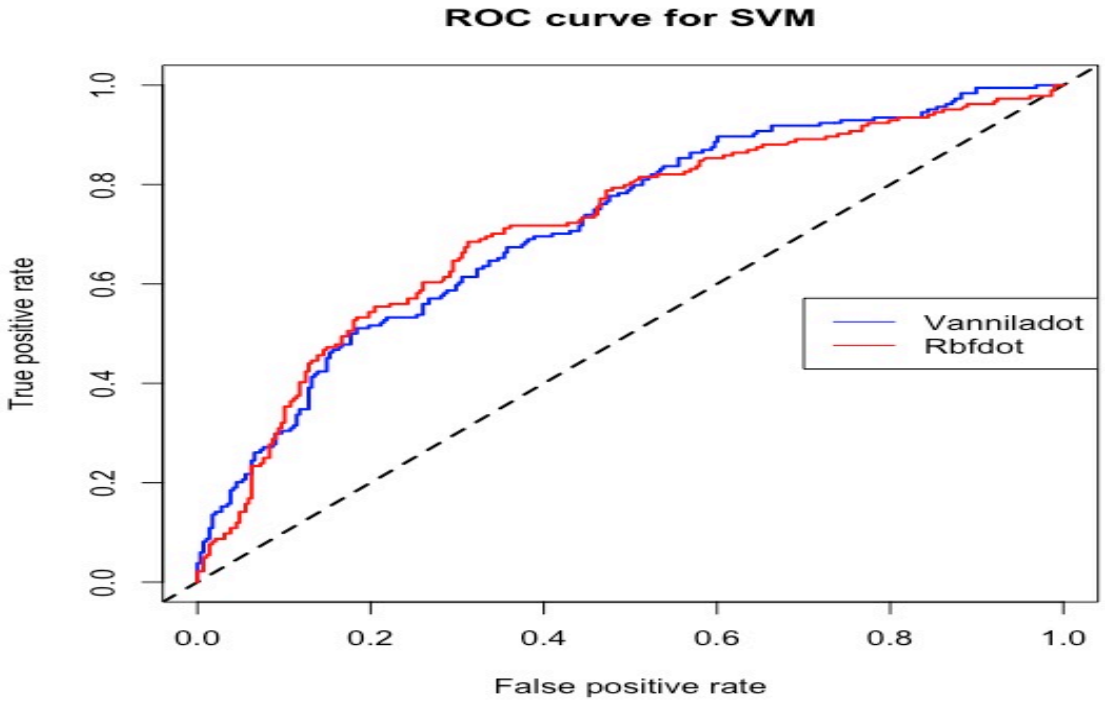
Model	Evaluation	Accuracy	Precision	Specificity	Sensitivity/recall	F1	AUC
Decision Tree	Test Set	**0.6822**	**0.6214**	0.8160	0.4728	0.5370	**0.6896**
	10-fold CV	0.6536	0.5796	0.8039	0.4180	0.4838	
Adaptive Boosting trials=15, model=tree, winnow = FALSE	Test Set	0.6716	0.6465	0.8785	0.3478	0.4523	0.7087
	10-fold CV	0.6717	0.6377	0.8614	0.3747	0.4630	
Random Forest tuned ntree=500, mtry=2	Test Set	**0.6801**	**0.6279**	0.8333	0.4402	0.5176	**0.6998**
	10-fold CV	0.6633	0.6025	0.8274	0.4061	0.4840	
SVM kernel = "vannilladot",	Test Set	0.6949	0.6408	0.8229	0.4946	0.5583	0.7182
	10-fold CV	0.6624	0.5952	0.8088	0.4333	0.4990	
SVM kernel = "rbfdot"	Test Set	**0.6992**	**0.6721**	0.8611	0.4457	0.5359	**0.7170**
	10-fold CV	0.6586	0.6015	0.8385	0.3768	0.4606	
k-NN with k=43	Test Set	**0.6886**	**0.6581**	0.8611	0.4185	0.5116	**0.6981**
	10-fold CV	0.6409	0.5665	0.8282	0.3475	0.4280	
ANN size =1, decay=0.1	Test Set	**0.6886**	**0.6294**	0.8160	0.4891	0.5505	**0.7144**
	10-fold CV	0.6760	0.6181	0.8233	0.4452	0.5156	

All the models generated accuracy in the range of 67-69.9% and precision within the range of 62-67% in the test set. SVM with “rbfdot” generates the highest accuracy of 69.92% with an overall precision of 67.21%. For 10-fold cross validation all models produce accuracy in the range 64-67% and precision of 56-63%.

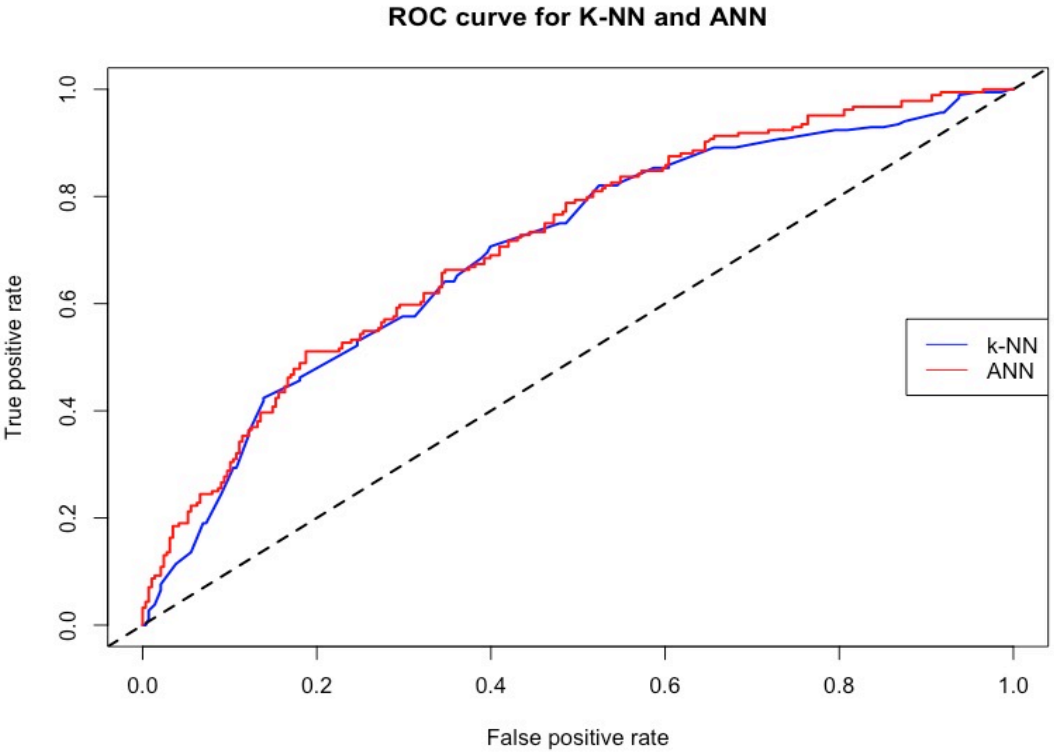
The receiver operating characteristic (ROC) curve examines the trade-off between detecting true positives and avoiding false positives. The perfect classifier has a curve that touches 100% true positive rate. The area under ROC (AUC) measures the closeness of the curve to the perfect classifier. (Lantz,2019).



For all the decision tree models, AUC ranges between 68-70% and comparatively similar accuracies. Adaptive boosting with 67% accuracy has the highest AUC of 70% followed by random forest with 69% AUC.



SVM with “vanniladot” and “rbfdot” kernels proves to be the most efficient in terms of both accuracy as well as AUC. The ROC curve with efficient AUC of 71.82% for “vanniladot” and 71.70% for “rbfdot”, indicates a good balance of TPR: FPR.



Artificial Neural Networks and k-NN produced the same accuracy of 68.86%. ANN has a higher AUC of 71.44% compared to k-NN that has 69.81%.

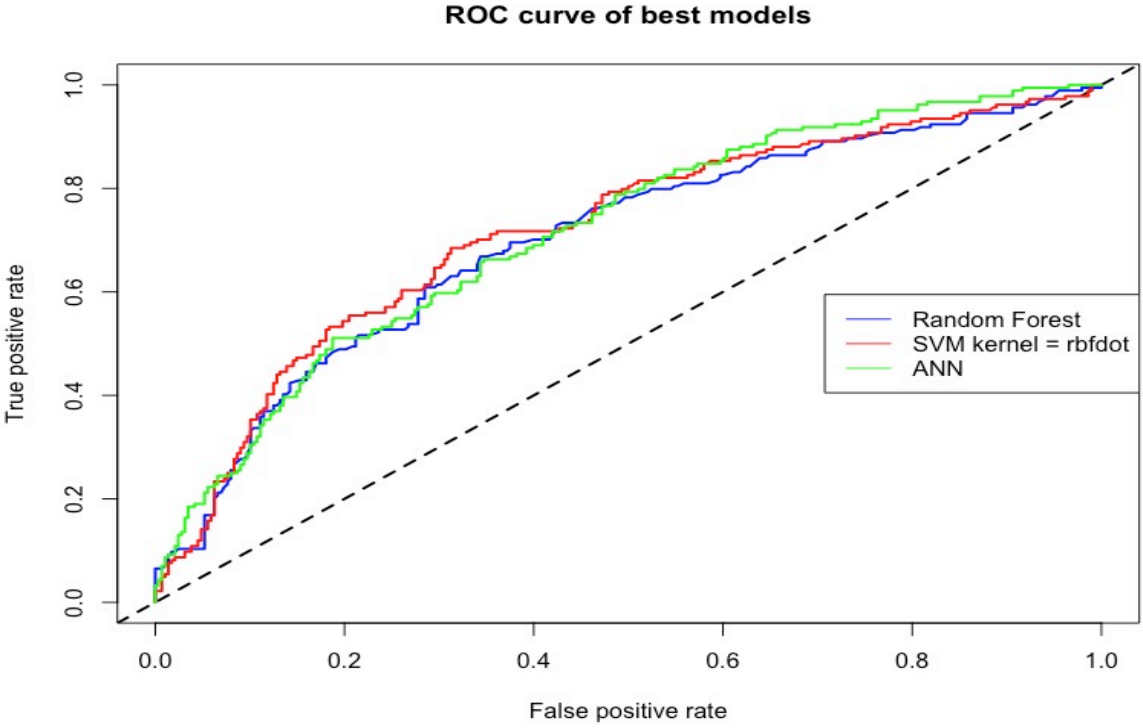
9. DISCUSSION

The best models based on performance evaluation and highest AUC are:

- SVM with Gaussian radial basis kernel (rbfdot)
- Random Forest and
- Artificial neural networks

SVM is the best performer due to its ability to capture complex non-linear relationships. Adaptive boosting had the highest AUC for decision trees but lower accuracy. Random Forest and Decision tree had similar accuracy, hence Random Forest with higher AUC was selected. ANN was chosen over k-NN due to its ability to generalize training data better and make accurate predictions from unseen instances, while k-NN relies heavily on the training instances.

The chosen models are effective in predicting the success of ICO fundraising campaigns because they can handle complex data, identify non-linear relationships, work with high-dimensional data, and recognize intricate patterns. These abilities allow them to extract valuable insights from the data and accurately predict the likelihood of success.



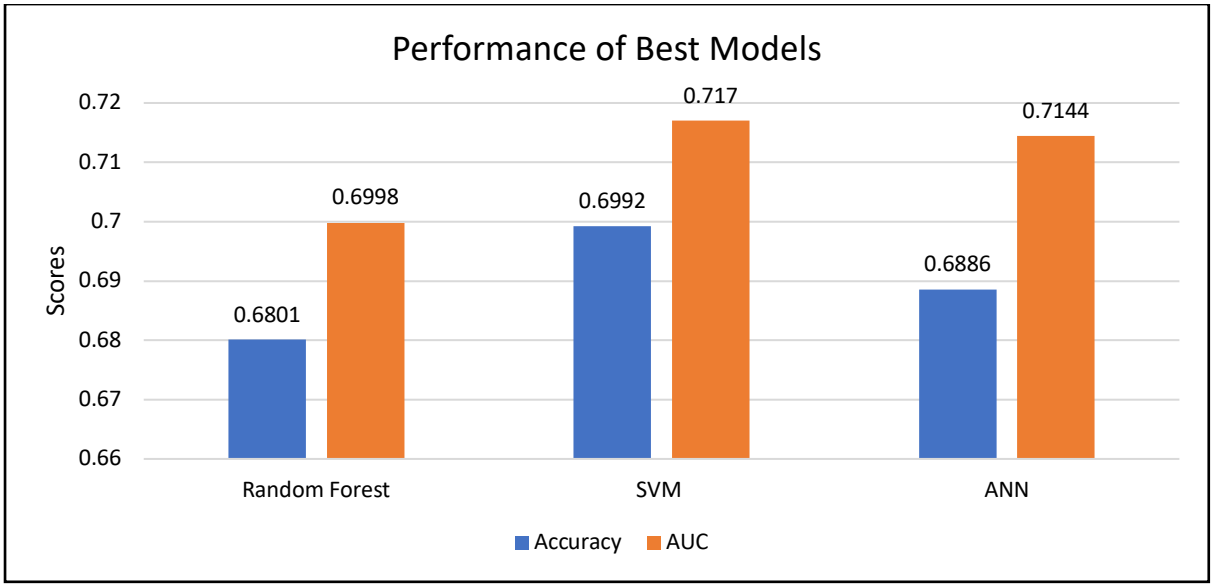
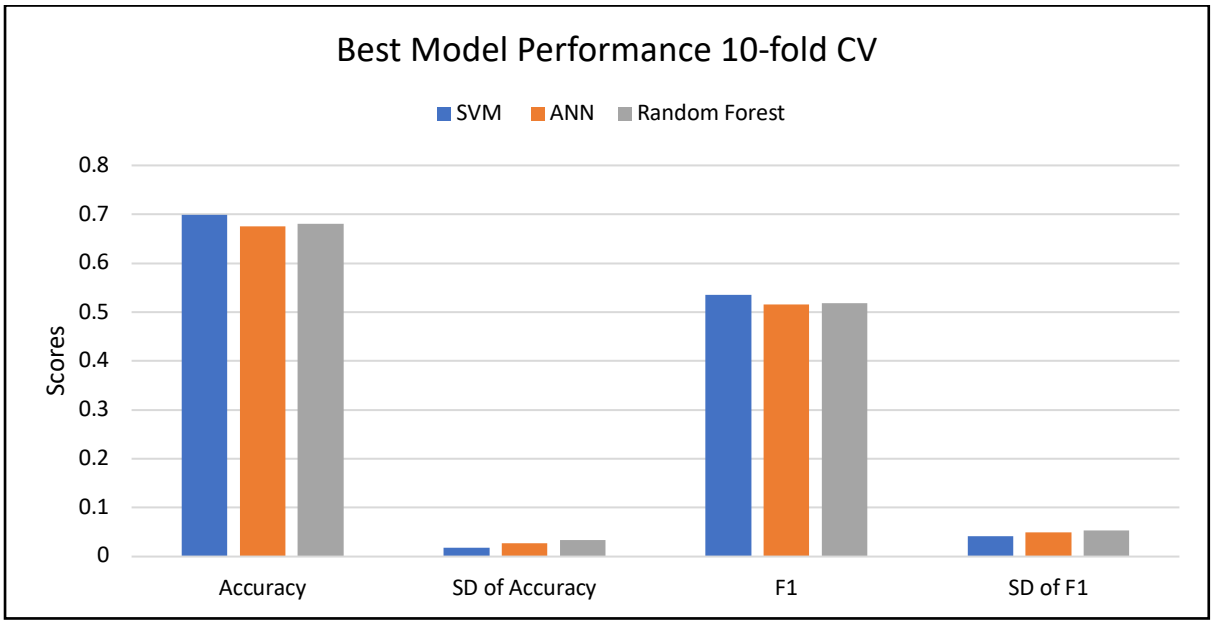


Table shows the performance of the best models for 10-fold cross validation in terms of accuracy and F1 score. F1 score gives the harmonic mean of precision and recall.

Model	Accuracy	SD of Accuracy	F1	SD of F1
SVM	0.6992	0.0177	0.5359	0.0410
ANN	0.6760	0.0268	0.5156	0.0500
Random Forest	0.6801	0.0341	0.5176	0.0532



10. CONCLUSION

Various supervised machine learning models were used to predict whether the fund-raising campaign of companies through ICO will be successful. Goal amount which is derived from price of coins and number of coins issued, is the most important predictor according to Random Forest model which helps the investors to assess their return on investments. The overall rating, project duration and team size are other important predictors. These findings imply that investors seek projects with realistic targets, positive rating, good planning, and competent teams. By finding the optimal decision boundary that maximises the distance between success and failure, the SVM model can accurately classify new ICO projects based on their features and determine the likelihood of their success.

11. LIMITATIONS

The model predictions are based on historical data and may not capture new trends in market conditions. The finding that setting the price of tokens to zero may positively impact the success of the ICO may not be practical in real life. ICOs typically involve some valuation for the tokens issued. The model performance should be evaluated on diverse ICO datasets to ensure generalizability. Additional factors like marketing strategies, token utility and legal compliance could impact the success of ICO which were not included in this analysis.

REFERENCES

1. Kepka, A. (2020). *Business Startup Statistics UK (2020 Update)* | *Fundsquire*. [online]. Accessed: 11-05-2023. Available at: https://fundsquire.co.uk/

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

3. Sameeh, T., 2018. ICO basics–security tokens vs. utility tokens.*Cointelligence. [online] Available at: https://www. cointelligence. com/content/ico-basics-security-tokens-vs-utility-tokens*

4. Dziura JD, Post LA, Zhao Q, Fu Z, Peduzzi P. 2013 Strategies for dealing with missing data in clinical trials: from design to analysis. Yale Journal of Biology and Medicine 86, pp.343-358

5. Huang, W., Vismara, S. and Wei, X., 2022. Confidence and capital raising. *Journal of Corporate Finance*, *77*, p.101900.

6. Kursa, M.B. and Rudnicki, W.R. (2010). Feature Selection with the Boruta Package. *Journal of Statistical Software*, 36(11). doi: https://doi.org/10.18637/jss.v036.i11.

7. Lantz, B., 2019. *Machine learning with R: expert techniques for predictive modeling*. Packt publishing ltd.

APPENDIX

This section contains the tuning results, cross table and confusion matrix of all the machine learning models.

- **Decision tree**

  Cross Table

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


Total Observations in Table:  472


             | actual
   predicted |      Fail |   Success | Row Total |
-------------|-----------|-----------|-----------|
        Fail |       235 |        97 |       332 |
             |     0.498 |     0.206 |           |
-------------|-----------|-----------|-----------|
     Success |        53 |        87 |       140 |
             |     0.112 |     0.184 |           |
-------------|-----------|-----------|-----------|
Column Total |       288 |       184 |       472 |
-------------|-----------|-----------|-----------|
```

  Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction Fail Success
   Fail     235      97
   Success   53      87

               Accuracy : 0.6822
                 95% CI : (0.6381, 0.724)
    No Information Rate : 0.6102
    P-Value [Acc > NIR] : 0.0006891

                  Kappa : 0.3018

 Mcnemar's Test P-Value : 0.0004465

            Sensitivity : 0.4728
            Specificity : 0.8160
         Pos Pred Value : 0.6214
         Neg Pred Value : 0.7078
              Precision : 0.6214
                 Recall : 0.4728
                     F1 : 0.5370
             Prevalence : 0.3898
         Detection Rate : 0.1843
   Detection Prevalence : 0.2966
      Balanced Accuracy : 0.6444

       'Positive' Class : Success
```

Tree structure

```
Call:
C5.0.default(x = select(dt_train, -success), y = dt_train$success)


C5.0 [Release 2.07 GPL Edition]         Tue May 16 02:25:40 2023
-------------------------------

Class specified by attribute `outcome'

Read 1892 cases (13 attributes) from undefined.data

Decision tree:

priceUSD <= 0: Success (42)
priceUSD > 0:
:...rating <= 2.9:
    :...project_duration > 4: Fail (791/180)
    :   project_duration <= 4:
    :   :...hasReddit <= 0: Fail (26/11)
    :       hasReddit > 0: Success (11/2)
    rating > 2.9:
    :...teamSize <= 11:
        :...hasGithub <= 0: Fail (104/29)
        :   hasGithub > 0:
        :   :...continent in {Europe,Unknown}: Fail (136/55)
        :       continent = Oceania:
        :       :...rating <= 3.6: Fail (7/2)
        :       :   rating > 3.6: Success (3)
        :       continent = Asia:
        :       :...minInvestment <= 0: Fail (24/6)
        :       :   minInvestment > 0: Success (26/10)
        :       continent = Africa:
        :       :...coinNum <= 3.874e+09: Fail (14/3)
        :       :   coinNum > 3.874e+09: Success (2)
```

```
        :       continent = Americas:
        :       :...minInvestment <= 0:
        :           :...priceUSD <= 0.05: Fail (9)
        :           :   priceUSD > 0.05: Success (20/5)
        :           minInvestment > 0:
        :           :...priceUSD <= 0.03:
        :               :...rating <= 3.6: Fail (2)
        :               :   rating > 3.6: Success (4)
        :               priceUSD > 0.03:
        :               :...project_duration <= 12: Success (2)
        :                   project_duration > 12: Fail (25)
        teamSize > 11:
        :...project_duration > 159: Fail (65/22)
            project_duration <= 159:
            :...rating > 4: Success (128/37)
                rating <= 4:
                :...continent in {Oceania,Unknown}: Fail (16/6)
                    continent in {Americas,Asia}: Success (193/81)
                    continent = Africa:
                    :...rating <= 3.5: Fail (6/1)
                    :   rating > 3.5: Success (4)
                    continent = Europe:
                    :...hasReddit <= 0:
                        :...hasGithub <= 0:
                        :   :...priceUSD <= 0.16: Success (13/3)
                        :   :   priceUSD > 0.16: Fail (5)
                        :   hasGithub > 0:
                        :   :...project_duration <= 22: Success (5/1)
                        :       project_duration > 22:
                        :       :...priceUSD <= 0.7: Fail (19/2)
                        :           priceUSD > 0.7: Success (2)
                        hasReddit > 0:
                        :...minInvestment <= 0: Fail (88/42)
                            minInvestment > 0:
                            :...hasVideo > 0: Success (92/33)
```

```
                        :...minInvestment <= 0: Fail (88/42)
                            minInvestment > 0:
                            :...hasVideo > 0: Success (92/33)
                                hasVideo <= 0:
                                :...project_duration <= 42: Fail (4)
                                    project_duration > 42: Success (4/1)


Evaluation on training data (1892 cases):

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

            33  532(28.1%)   <<


           (a)   (b)    <-classified as
          ----  ----
           982   173    (a): class Fail
           359   378    (b): class Success


        Attribute usage:

        100.00% priceUSD
         97.78% rating
         79.23% project_duration
         54.02% teamSize
         38.32% continent
         22.30% hasGithub
         15.86% minInvestment
         14.22% hasReddit
          5.29% hasVideo
          0.85% coinNum
```

- **Adaptive boosting of decision tree**

  Tuning Results

```
> best_params
  model trials
3  tree     15
```

  Cross Table

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


Total Observations in Table:  472


             | actual
   predicted |       Fail |    Success | Row Total  |
-------------|------------|------------|------------|
        Fail |        253 |        120 |        373 |
             |      0.536 |      0.254 |            |
-------------|------------|------------|------------|
     Success |         35 |         64 |         99 |
             |      0.074 |      0.136 |            |
-------------|------------|------------|------------|
Column Total |        288 |        184 |        472 |
-------------|------------|------------|------------|
```

Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction Fail Success
   Fail     253     120
   Success   35      64

               Accuracy : 0.6716
                 95% CI : (0.6272, 0.7138)
    No Information Rate : 0.6102
    P-Value [Acc > NIR] : 0.003313

                  Kappa : 0.2469

 Mcnemar's Test P-Value : 0.00000000001509

            Sensitivity : 0.3478
            Specificity : 0.8785
         Pos Pred Value : 0.6465
         Neg Pred Value : 0.6783
              Precision : 0.6465
                 Recall : 0.3478
                     F1 : 0.4523
             Prevalence : 0.3898
         Detection Rate : 0.1356
   Detection Prevalence : 0.2097
      Balanced Accuracy : 0.6131

       'Positive' Class : Success
```

- **Random Forest**

  Tuning Results

```
> best_params
  ntree mtry
5   500    2
```

  Cross Table

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


Total Observations in Table:  472


             | predict
      actual |       Fail |    Success | Row Total |
-------------|------------|------------|-----------|
        Fail |        240 |         48 |       288 |
             |      0.700 |      0.372 |           |
-------------|------------|------------|-----------|
     Success |        103 |         81 |       184 |
             |      0.300 |      0.628 |           |
-------------|------------|------------|-----------|
Column Total |        343 |        129 |       472 |
             |      0.727 |      0.273 |           |
-------------|------------|------------|-----------|
```

Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction Fail Success
   Fail      240      103
   Success    48       81

               Accuracy : 0.6801
                 95% CI : (0.6359, 0.722)
    No Information Rate : 0.6102
    P-Value [Acc > NIR] : 0.0009605

                  Kappa : 0.2892

 Mcnemar's Test P-Value : 0.0000111

             Sensitivity : 0.4402
             Specificity : 0.8333
          Pos Pred Value : 0.6279
          Neg Pred Value : 0.6997
               Precision : 0.6279
                  Recall : 0.4402
                      F1 : 0.5176
              Prevalence : 0.3898
          Detection Rate : 0.1716
   Detection Prevalence : 0.2733
      Balanced Accuracy : 0.6368

       'Positive' Class : Success
```

- **SVM with kernel = “vanniladot”**

Cross Table

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


Total Observations in Table:  472


             | predict
      actual |      Fail |   Success | Row Total |
-------------|-----------|-----------|-----------|
        Fail |       237 |        51 |       288 |
             |     0.718 |     0.359 |           |
-------------|-----------|-----------|-----------|
     Success |        93 |        91 |       184 |
             |     0.282 |     0.641 |           |
-------------|-----------|-----------|-----------|
Column Total |       330 |       142 |       472 |
             |     0.699 |     0.301 |           |
-------------|-----------|-----------|-----------|
```

Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction Fail Success
   Fail     237      93
   Success   51      91

               Accuracy : 0.6949
                 95% CI : (0.6512, 0.7362)
    No Information Rate : 0.6102
    P-Value [Acc > NIR] : 0.00007731

                  Kappa : 0.3311

 Mcnemar's Test P-Value : 0.0006339

            Sensitivity : 0.4946
            Specificity : 0.8229
         Pos Pred Value : 0.6408
         Neg Pred Value : 0.7182
              Precision : 0.6408
                 Recall : 0.4946
                     F1 : 0.5583
             Prevalence : 0.3898
         Detection Rate : 0.1928
   Detection Prevalence : 0.3008
      Balanced Accuracy : 0.6587

       'Positive' Class : Success
```

- **SVM with kernel = “rbfdot”**

Cross Table

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


Total Observations in Table:  472


             | predict
      actual |      Fail |   Success | Row Total |
-------------|-----------|-----------|-----------|
        Fail |       248 |        40 |       288 |
             |     0.709 |     0.328 |           |
-------------|-----------|-----------|-----------|
     Success |       102 |        82 |       184 |
             |     0.291 |     0.672 |           |
-------------|-----------|-----------|-----------|
Column Total |       350 |       122 |       472 |
             |     0.742 |     0.258 |           |
-------------|-----------|-----------|-----------|
```

Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction Fail Success
   Fail     248      102
   Success   40       82

               Accuracy : 0.6992
                 95% CI : (0.6556, 0.7402)
    No Information Rate : 0.6102
    P-Value [Acc > NIR] : 0.0000345782

                  Kappa : 0.3266

 Mcnemar's Test P-Value : 0.0000003071

            Sensitivity : 0.4457
            Specificity : 0.8611
         Pos Pred Value : 0.6721
         Neg Pred Value : 0.7086
              Precision : 0.6721
                 Recall : 0.4457
                     F1 : 0.5359
             Prevalence : 0.3898
         Detection Rate : 0.1737
   Detection Prevalence : 0.2585
      Balanced Accuracy : 0.6534

       'Positive' Class : Success
```

- **k-NN**

Cross Table

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


Total Observations in Table:  472


             | predict
      actual |       Fail |    Success | Row Total |
-------------|------------|------------|-----------|
        Fail |        248 |         40 |       288 |
             |      0.699 |      0.342 |           |
-------------|------------|------------|-----------|
     Success |        107 |         77 |       184 |
             |      0.301 |      0.658 |           |
-------------|------------|------------|-----------|
Column Total |        355 |        117 |       472 |
             |      0.752 |      0.248 |           |
-------------|------------|------------|-----------|
```

Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction Fail Success
   Fail     248     107
   Success   40      77

               Accuracy : 0.6886
                 95% CI : (0.6446, 0.7301)
    No Information Rate : 0.6102
    P-Value [Acc > NIR] : 0.0002407

                  Kappa : 0.2993

 Mcnemar's Test P-Value : 0.00000005222

            Sensitivity : 0.4185
            Specificity : 0.8611
         Pos Pred Value : 0.6581
         Neg Pred Value : 0.6986
              Precision : 0.6581
                 Recall : 0.4185
                     F1 : 0.5116
             Prevalence : 0.3898
         Detection Rate : 0.1631
   Detection Prevalence : 0.2479
      Balanced Accuracy : 0.6398

       'Positive' Class : Success
```

- **ANN**

Tuning Results

```
> best_params
  size decay
3    1   0.1
```

Cross Table

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


Total Observations in Table:  472


             | actual
   predicted |       Fail |    Success | Row Total |
-------------|------------|------------|-----------|
        Fail |        235 |         94 |       329 |
             |      0.498 |      0.199 |           |
-------------|------------|------------|-----------|
     Success |         53 |         90 |       143 |
             |      0.112 |      0.191 |           |
-------------|------------|------------|-----------|
Column Total |        288 |        184 |       472 |
-------------|------------|------------|-----------|
```

Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction Fail Success
   Fail      235      94
   Success    53      90

               Accuracy : 0.6886
                 95% CI : (0.6446, 0.7301)
    No Information Rate : 0.6102
    P-Value [Acc > NIR] : 0.0002407

                  Kappa : 0.3179

 Mcnemar's Test P-Value : 0.0009698

            Sensitivity : 0.4891
            Specificity : 0.8160
         Pos Pred Value : 0.6294
         Neg Pred Value : 0.7143
              Precision : 0.6294
                 Recall : 0.4891
                     F1 : 0.5505
             Prevalence : 0.3898
         Detection Rate : 0.1907
   Detection Prevalence : 0.3030
      Balanced Accuracy : 0.6526

       'Positive' Class : Success
```