Leeds University Business School

UNIVERSITY OF LEEDS

Assessed Coursework Coversheet

For use with *individual* assessed work

Student ID Number:	2	0	1	6	2	2	9	8	7
Module Code:	LUBS5990M								
Module Title:	Machine Learning in Practice								
Module Leader:	Xingjie Wei								
Declared Word Count:	3500								

Please Note:

Your declared word count must be accurate, and should not mislead. Making a fraudulent statement concerning the work submitted for assessment could be considered academic malpractice and investigated as such. If the amount of work submitted is higher than that specified by the word limit or that declared on your word count, this may be reflected in the mark awarded and noted through individual feedback given to you.

It is not acceptable to present matters of substance, which should be included in the main body of the text, in the appendices ("appendix abuse"). It is not acceptable to attempt to hide words in graphs and diagrams; only text which is strictly necessary should be included in graphs and diagrams.

By submitting an assignment you confirm you have read and understood the University of Leeds **Declaration of Academic Integrity** (http://www.leeds.ac.uk/secretariat/documents/academic_integrity.pdf).

Predicting success for ICO funding: Application and Comparison of Machine Learning models for predicting outcomes of ICO fundraising campaigns.

1. Introduction

Crowdfunding involves several individuals contributing to raise funds for a certain company or project and is frequently handled online through a listed website or governing body. The company may advertise their initiative through various means to reach their fundraising goal. The Initial Coin Offering, or ICO, is a unique type of online crowdfunding investment (Agrawal, Catalini, & Goldfarb, 2015) in which a company seeking funds offers a new digital coin or token to potential investors in return for Ethereum or Bitcoin, essentially based on distributed ledger technology (Fisch, 2019). These issued coins can subsequently be traded on cryptocurrency exchanges or inside the ecosystem of the proposed enterprise.

This sort of emerging crowdfunding is highly uncontrolled (Cronqvist, Siegel, & Yu, 2015) and draws speculative investors interested in the possible profits from the success of the proposed venture.

ICOs are typically used by early-stage firms or projects that have yet to build a service, platform, or product (Huang, Vismara, & Wei). The goal is to successfully raise funds for the proposed business, service, or platform for which the ICO is being launched.

The primary aim of this study is to apply machine learning models to predict whether a project or company will successfully raise funds using an ICO. To examine the success factor and the elements that contribute to it, over 2000 projects from diverse firms and teams will be employed.

A dataset encompassing different ICO initiatives, and their outcomes was provided to target this investigation. Since the target variable is binary “Yes” or “No” it is understood that this is a classification problem. On this data, four classification models will be trained and tested, and the top performing model on unseen data will be identified.

2. Data Understanding

Dataset: LUBS5990M_courseworkData_202223.csv

The dataset constitutes 16 variables with 2767 observations.

Numerical Variables:

1. ***ID*** – Unique identification number for each fundraising project

2. ***coinNum*** – Number of digital coins to be issued by the project team.

   This column contains extreme datapoints ranging from 12 (minimum value) to 22619078416800000 (maximum value). Hence, the nature of the distribution below.

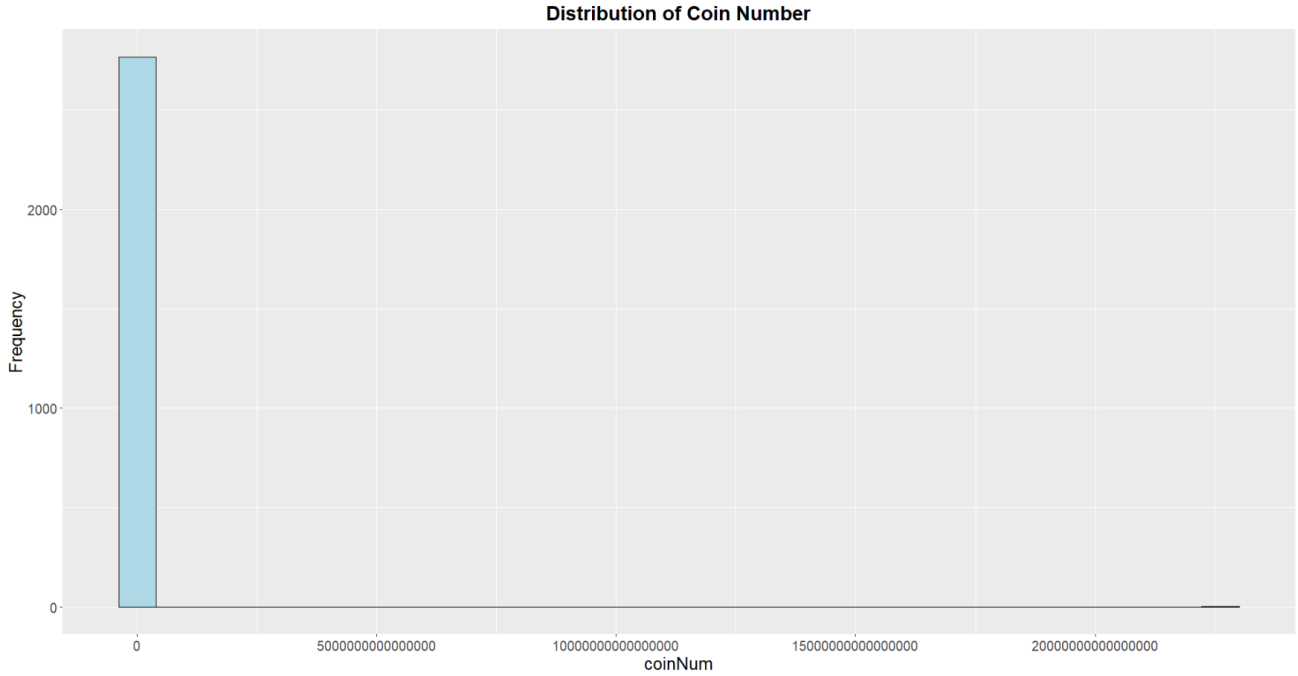

Figure 1: Distribution of Coin Number

3. ***priceUSD*** – Price of each issued coin is US Dollars.

   summary() function of this column reveals that the mean=19.01, min value=0 and max values=39384.0 are significantly apart there is very high variability in this column and indicates presence of a significant number of outliers.

The box plot shows the presence of outliers in this column.

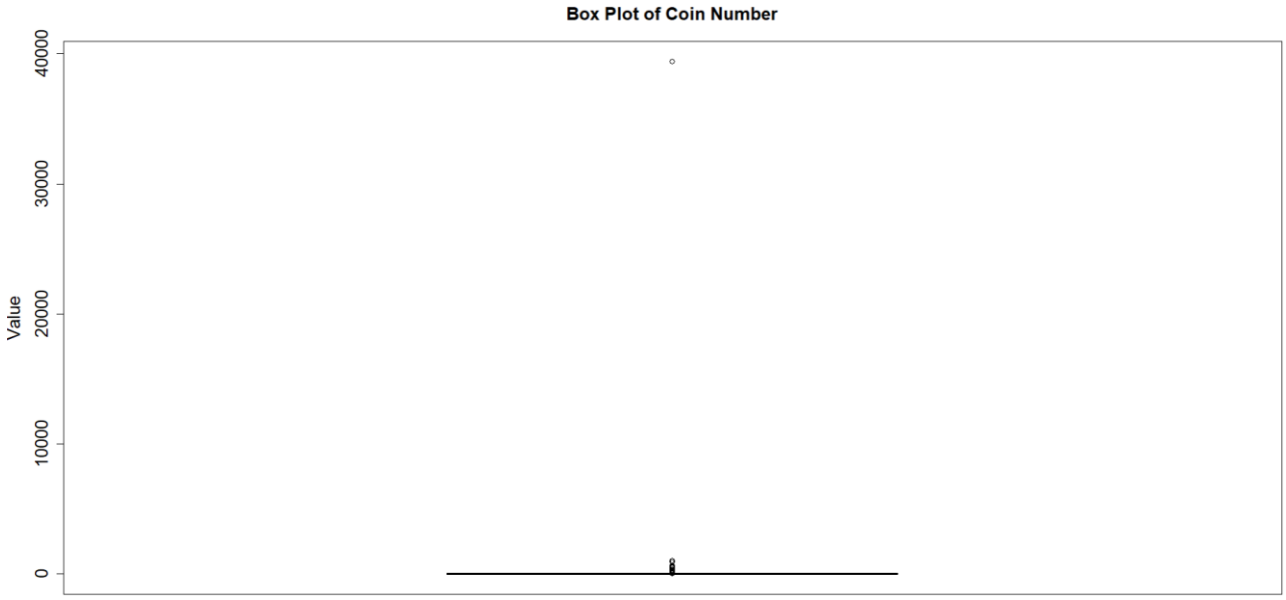

Figure 2: Box Plot of Coin Number

4. ***teamSize*** – Size of the team belonging to a fundraising project. Distribution reveals a certain amount of skewness indicating that majority of the data is centred around the mean however there exist some outliers.

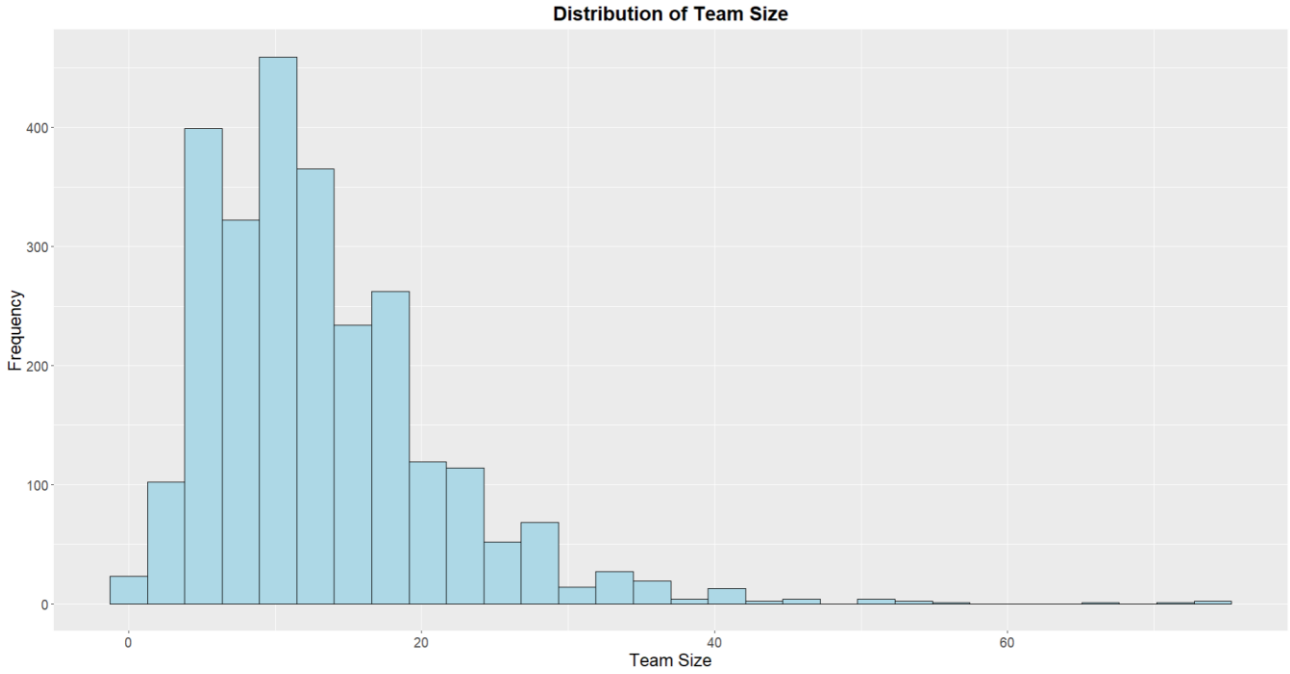

Figure 3: Distribution of Team Size

Boxplot reveals the outlier concentration in this column.

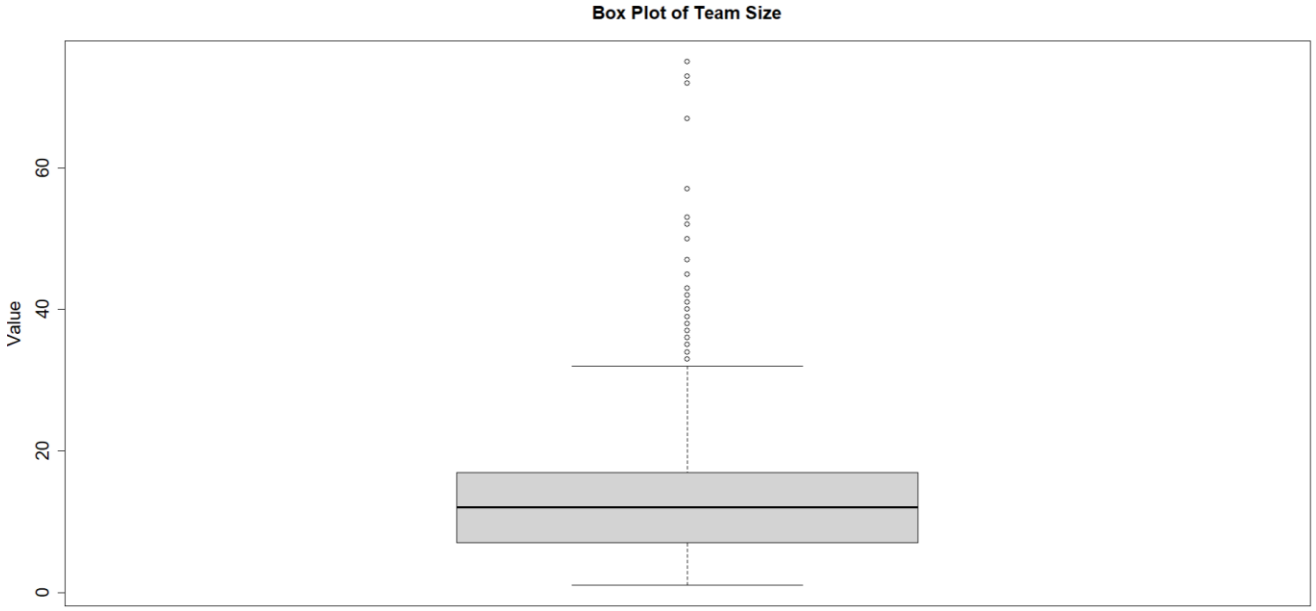

Figure 4: Box Plot of Team Size

5. ***rating*** – Overall score determined by investment experts for a fundraising project. Ranges from 1 (very poor) to 5 (very good).
6. ***minimumInvestment*** – Binary variable having values 1 if there is a minimal investment amount for a project in their campaign page otherwise 0.
7. ***distributedPercentage*** – The percentage of blockchain coin distributed to investors relative to all the coins created.
   Boxplot shows presence of outliers for this column.

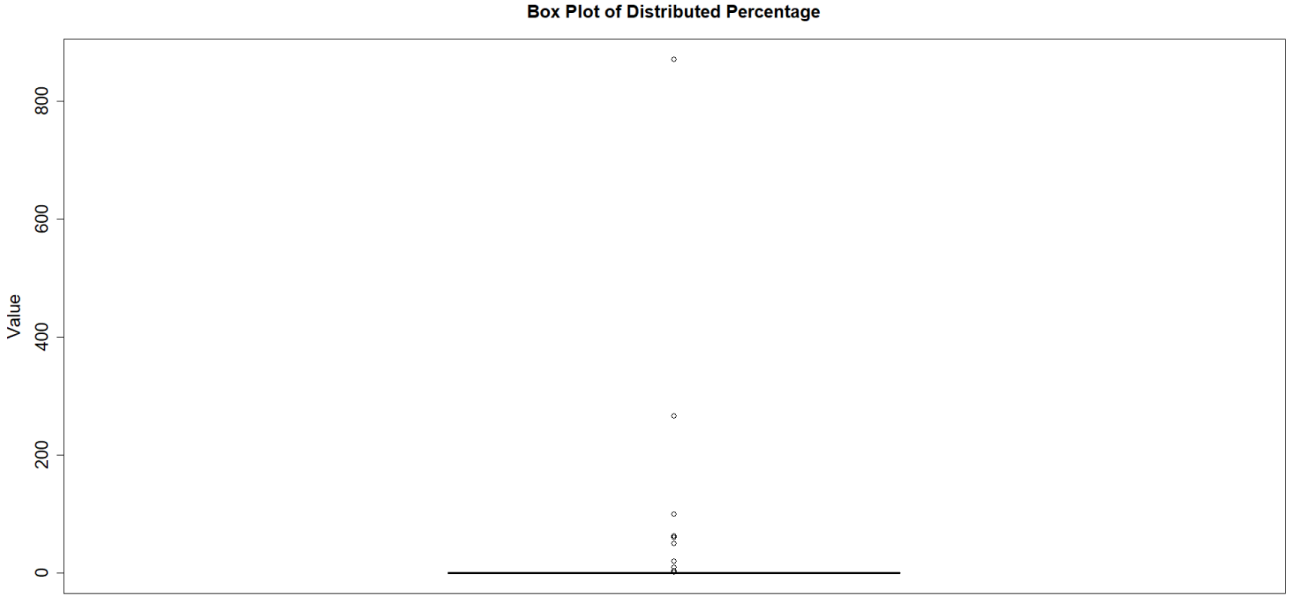

Figure 5: Box Plot of Distributed Percentage

8. ***hasVideo*** – Binary variable having values 1 if there is a video for the fundraising project in their campaign page otherwise 0.

9. ***hasGithub*** – Binary variable having values 1 if there is a Github link provided (open source) for the fundraising project in their campaign page otherwise 0.

10. ***hasReddit*** – Binary variable having values 1 if there is a Reddit link (community discussion) for the fundraising project in their campaign page otherwise 0.

Categorical Variables:

1. ***success*** – The target variable that will be predicted based on the independent variables selected. Essentially an indicator variable with a "Y" value if the project has achieved its fundraising goal otherwise "N" value.

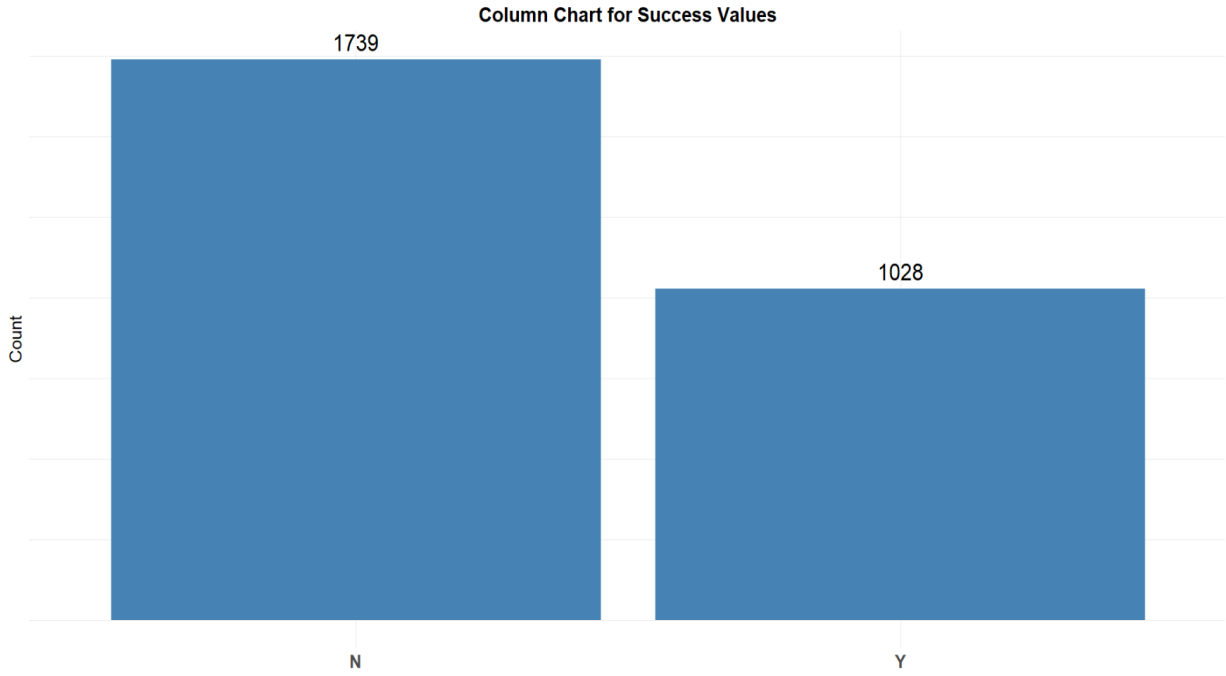

Figure 6: success variable distribution

2. ***startDate*** – Date when the fundraising campaign was started for a project.

3. ***endDate*** – Date when the fundraising campaign ended for a project.

4. ***platform*** – Blockchain platform on which the fundraising project is set up on.

5. ***countryRegion*** – The country to which the team/campaign belongs.

6. ***brandSlogan*** – Text variable indicating the slogan for the fundraising team/project.

3. Data Preparation

3.1. Data Transformation

1. The ***success*** variable was first analysed to understand the type and frequency of values. *Figure 6* above tells us the “Y” values were 1028 number and “N” values were 1739 in number. This was transformed to a factor variable whose values represent 2 levels “Yes” and “No” respectively.

Yes	No
1028	1739

Table 1: success variable distribution

This indicates that the negative outcome of "No" for the success of a fundraising campaign is prevalent in the dataset, accounting for more than 60% of the observations.

2. The ***startDate*** and ***endDate*** have no significance alone and instead used to generate ***daysDuration*** which basically stores the duration in days for the fundraising campaign.

3.2. Outlier Detection

Outlier detection has been done for the columns in above section.

The outliers for all the numeric columns except indicator columns with 0 or 1 value.

rating	*priceUSD*	*teamSize*	*coinNum*	*distributedPercentage*	*daysDuration*
0	240	66	395	10	184

Table 2: Outlier distribution by variable

Reasons for keeping outliers:

- Data Integrity - Preventing loss of information and underfitting as the given dataset has only 2767 observations. Furthermore, the information provided is real world data and we want to be able to accommodate all possibilities occurred so far since ICO is speculative and lack regulation.
- Robustness: The models being used in this analysis are inherently robust to outliers and so removing them will not have much effect on the performance of the final model instead the presence may allow the model to prove versatility.

3.3. Missing data and Imputation

Figure 7 shows the pattern of the missing data and the columns to which they belong. It is observed that ***teamSize*** has a total of 154 missing values, ***priceUSD*** has a total of 180 missing values. There are 9 observations that contain missing data for both the ***teamSize*** and ***priceUSD*** columns whereas there are 145 instances of missing values for ***teamSize*** and 171 instances for ***priceUSD***.

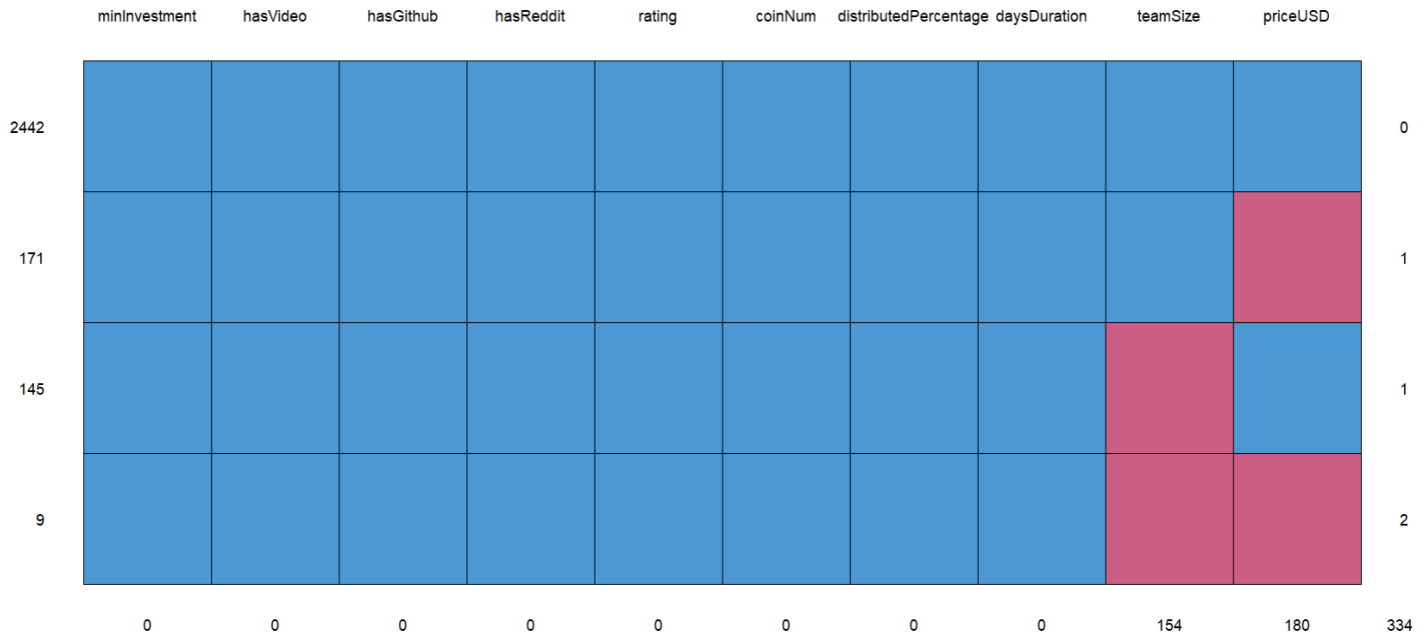

Figure 7: Missing data distribution plot

The *MICE (Multivariate Imputation by Chained Equations)* package in R is used to impute data values. All the numerical columns are extracted and are imputed using 2 methods:

1. **CART**

   CART stands for Classification and Regression Trees and involves imputation using a decision-tree based algorithm and predicts the missing variables based on the other variables in the dataset.

2. **Lasso.norm**

Stands for Lasso with Normalisation. This method combines Lasso regression and normal imputation and is a linear regression technique. Predictor variables are used to predict the missing values. These predictor variables emerge as a result of shrinkage of the coefficients for the less significant variables.

3.4. Results of the imputation:

teamSize has been imputed with both the methods mentioned above and their distributions are compared with the original which contains missing data. The CART imputations are chosen to complete the missing values as the distribution of data in *Figure 8* is closest to the original.

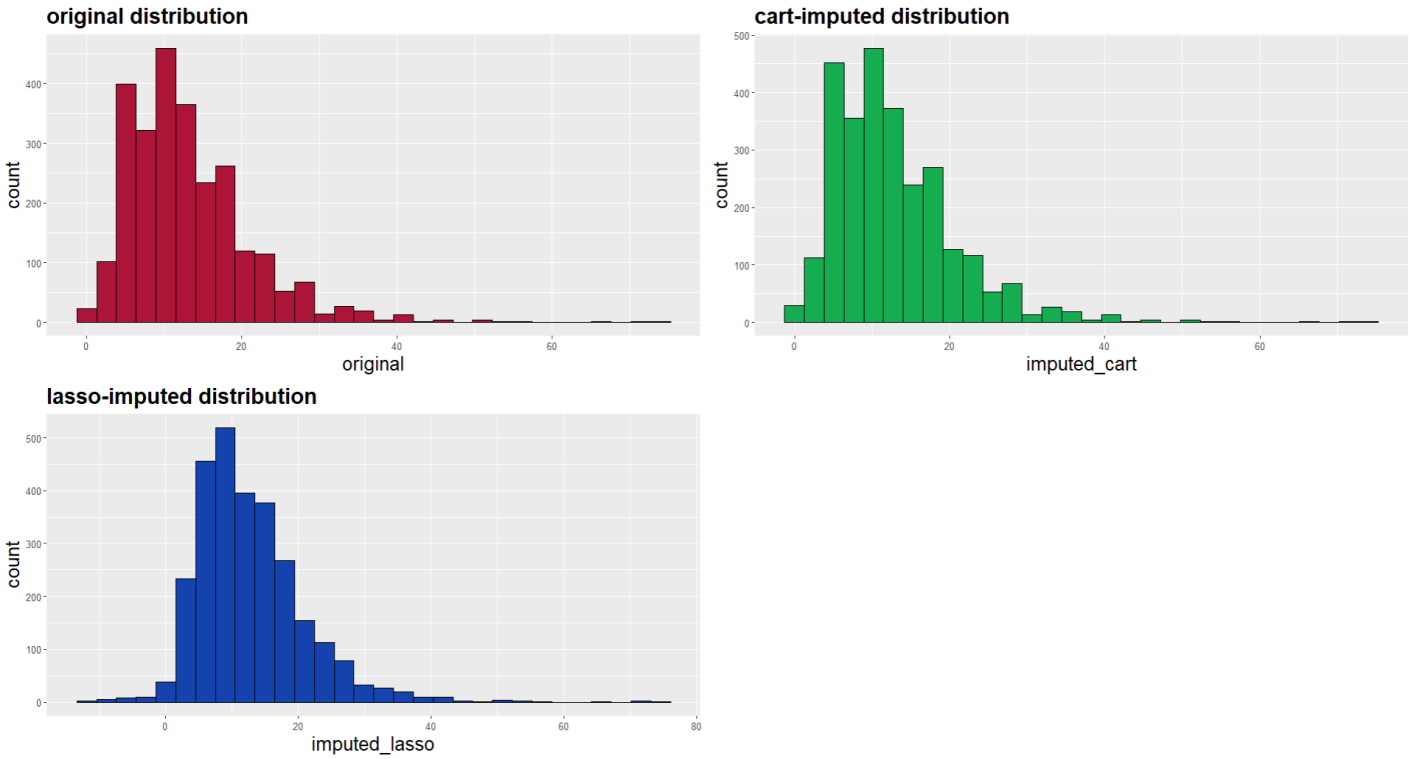

Figure 8: ***teamSize*** *distribution comparison for data before and after imputation*

Likewise, ***priceUSD*** has been imputed with both the methods mentioned above and their distributions are compared with the original. The CART imputations are chosen to complete the missing values as the distribution of data in *Figure 9* shows to be closest to the original containing missing data.

3.5. Feature Selection

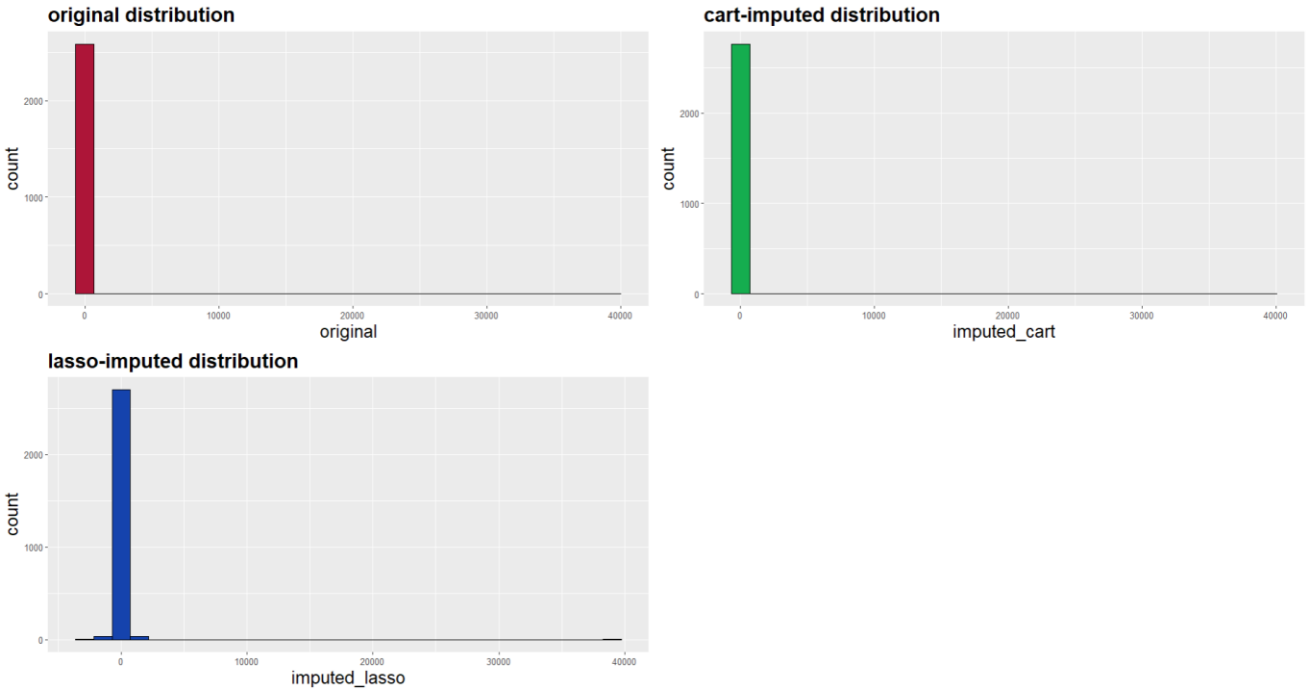

*Figure 9: **priceUSD** distribution comparison for data before and after imputation*

3.5.1 Dropping variables.

ID – It is chosen to drop the ID columns since it is not relevant and is just a column for maintaining index number of observations.

brandSlogan – The Chi-Square test was used to check whether there is a significant correlation of the ***brandSlogan*** with the target variable ***success*** *[Table 3].* It is decided to drop brandSlogan since it has no significant contribution in a 95% confidence interval, therefore we accept the null hypothesis [*Table 3*].

Variable Name	Chi-Square test p-value	Null Hypothesis	Accept/Reject Null Hypothesis
brandSlogan	0.4926	No relation with target variable *success*	Accept Null Hypothesis
countryRegion	***0.00001909***	No relation with target variable *success*	***Reject Null Hypothesis***
platform	0.2505	No relation with target variable *success*	Accept Null Hypothesis

Table 3: Chi-Square test results

***platform* –** The platform variable was coded as well where 1 indicated Ethereum and 0 otherwise.

Table 4 below shows the count of variable for each category after coding the ***platform*** variable as a binary column.

Ethereum	2408
Others	359

Table 4: Ethereum vs other platforms' distribution

It is decided to drop the platform variable as it did not have any significant contribution. The Chi-Squared test was used to confirm variable ***platform*** had no significant correlation with the target variable ***success*** *[Table 3].*

3.5.2. Correlation

Examining the correlation matrix for the remaining columns in *Figure 10* it is observed that most of the variables do not show any significant correlation whether positive (>0.5) or negative (<-0.5).

Hence, it is decided to focus rather on the statistical significance. This helps us to identify variables that have a contribution towards better prediction of the target variable. The correlation matrix in *Figure 10* also confirms that there is no multicollinearity among the variables.

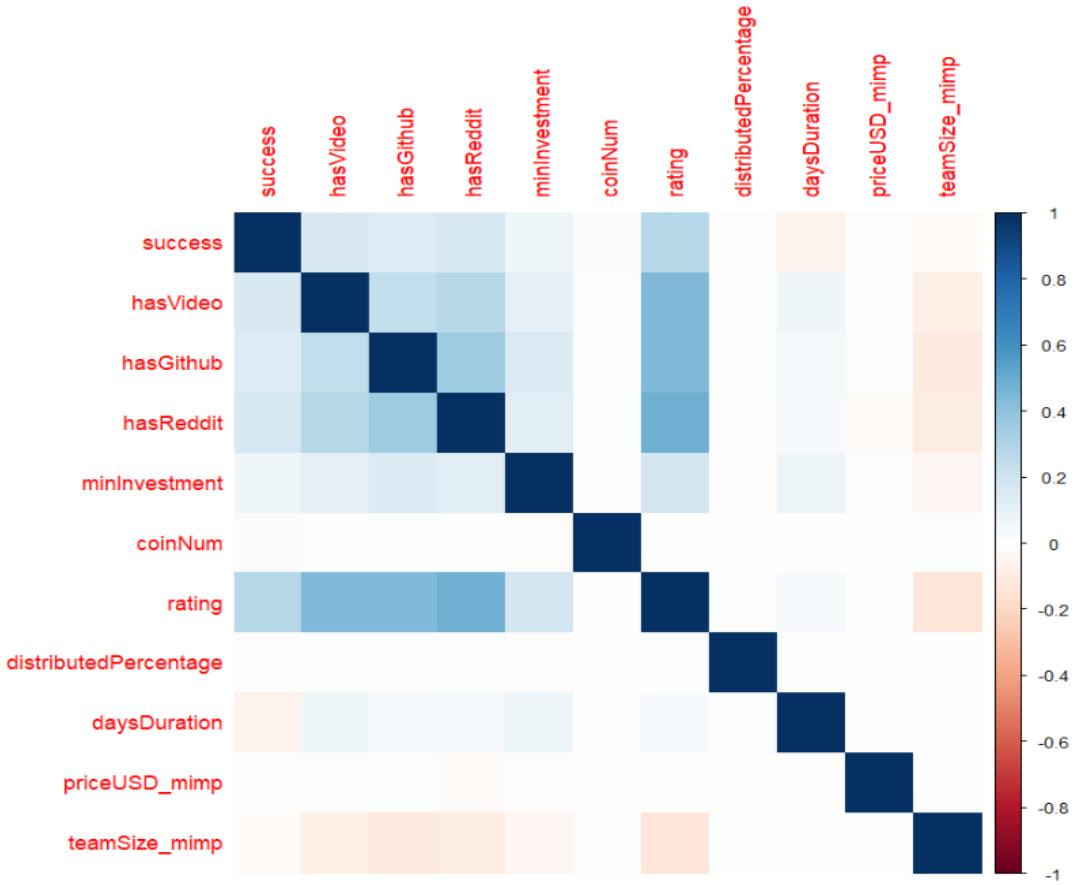

Figure 10: Correlation matrix plot for numerical variables

3.5.3 Features Selected for Modelling

Categorical

countryRegion – Significant correlation was found between this variable with the target variable ***success*** from *Table 3* and according to a study a county's friendliness for ICO's is a good predictor for ***success*** of the project (Ahmad, Kowalewski, & Pisany, 2021). The p-value obtained because of the Chi-Squared test suggests that the null hypothesis can be rejected and that there is a relation between the two variables (p-value<0.05).

Since this variable contains several countries and even some blanks, it was decided to select top 12 ICO-friendly countries and code them as 1 while changing the others to 0, storing them in a numeric binary classification variable ***isICO_Friendly***.

Table 5 below shows the count of ICO campaigns belonging to ICO-Friendly countries.

ICO-Friendly Countries	**1512**
Others	1255

Table 5: ICO Friendly Countries vs others distribution

Numerical

To assess the statistical significance, we will be using the p-value of the coefficients generated by the output of the logistic regression performed for selecting the significant features.

Our null hypothesis states that none of the variables have any correlation with the target ***success*** variable.

3.5.4 Logistic Regression for Feature Selection

Logistic regression helps us find the statistically significant variables that have importance for predicting the target variable. There will be a focus on the p-value of the predictor variables.

Features provided to model:

The variables in *Table 6* are scaled using the *scale()* function in R. It performs standardization of the variables to be centred around their mean i.e., have a mean of zero and scales them by standard deviation i.e., have a standard deviation of one.

Variable Name	Variable Type
success	Factor w/ 2 levels "Yes","No": 2 2 2 1 2 2 1 2 1 1 ...
hasVideo	int 1 1 1 1 1 1 1 1 1 1 ...
hasGithub	int 1 1 1 1 1 1 1 1 1 1 ...
hasReddit	int 1 1 1 1 1 1 1 1 1 1 ...
minInvestment	int 0 1 1 1 1 1 1 1 1 1 ...
coinNum	num -0.019 -0.019 -0.019 -0.019 -0.019 ...
rating	num 1.23 1.65 1.79 1.65 1.65 ...
distributedPercentage	num -0.0327 -0.0373 -0.0378 -0.0533 -0.0321 ...
daysDuration	num -0.6972 -0.3473 2.9523 0.0527 0.5526 ...
priceUSD	num -0.0234 -0.0233 -0.0237 -0.0234 -0.0235 ...
teamSize	num 0.767 -1.1 0.642 0.02 -0.353 ...
isICO_Friendly	num 1 0 1 0 0 1 1 0 0 0 ...

Table 6: Summary of structure for logistic regression feature selection

Result of the model:

Table 7 provides with the p-value for each variable as a result of the logistic regression performed on the dataset. The top five statistically significant variables have been chosen and these features will be provided to all the models and the best performer will be identified.

Variable Name	p-value
hasVideo	**0.026646**
hasGithub	0.304750
hasReddit	**0.154813**
minInvestment	0.314729
coinNum	0.253558
rating	**< 0.0000000000000002**
distributedPercentage	0.379872
daysDuration	**0.000649**
priceUSD	0.670308
teamSize	**0.139209**
isICO_Friendly	**0.003223**

Table 7: p-value for statistical significance analysis of variables

4. Modelling

Model 1: Decision Trees with AdaBoost (DT)

Decision trees algorithm for classification was chosen as it performs effectively for dataset containing non linearities, missing values and variety of variables (Chauhan & Chauhan, 2013).

The classification power of this algorithm is used as it involves splitting the data, feature wise and creating various branches and subbranches. These branches are the decision made by the algorithm at each node and this continues till a stopping point has been reached. Combining the strength with AdaBoost which is an ensemble technique, it can combine weak classifiers to create a strong and accurate one.

Parameters

- *default* parameters
- *trials = 50,* this specifies the number of times the data is split into training and validation sets randomly.

K-Fold Cross Validation method

For our model we first split the data into train and test **80-20 split**. The 80% of the train data will be passed to the k-fold cross validation section. Within that section the data is further split into 'k' number of folds for similar size.

The model gets trained on *'k-1' folds* for the data and is validated on the remaining fold. This takes place in 'k' number of iterations where each fold gets a chance to act as the validation set. This helps the predictive ability of the model, and we calculate the Accuracy and Area Under the Curve at each iteration of the model and in the end consider the mean of both the values to be compared with the performance of the algorithm on the holdout set.

Evaluation Metric	Iteration 1 (1st fold)	Iteration 2 (2nd fold)	Iteration 3 (3rd fold)	Iteration 4 (4th fold)	Iteration 5 (5th fold)	Mean
Accuracy	0.6704289	0.6787330	0.6749436	0.6681716	0.6968326	***0.6778219***
AUC	0.6760248	0.7019180	0.6822086	0.6413273	0.6729073	***0.6748772***

Table 8: 5-Fold Cross-Validation summary for Decision Trees with AdaBoost Classifier

The results obtained above are then compared to the performance of the trained classifier on the 80% split train set. Model is evaluated based on its performance on the 20% split test set. This test set was untouched and did not appear to the classifier during the k-fold cross validation set and is termed as the **holdout set**. Thus, model evaluation will be conducted on unseen data and performance will be evaluated accordingly.

Model 2: Naïve Bayes (NB)

As the name suggests this algorithm is based on Bayes' theorem. It assumes an observation is independent of any class (Rish, 2001), calculates posterior probability of an observation and its feature values, belonging to each class in the data and the class with the highest probability is predicted as the final label for that instance.

Parameters

- *default* parameters

K-Fold Cross Validation method

The method remains the same as specified for the Decision Tree algorithm **above**.

Evaluation Metric	Iteration 1 (1st fold)	Iteration 2 (2nd fold)	Iteration 3 (3rd fold)	Iteration 4 (4th fold)	Iteration 5 (5th fold)	Mean
Accuracy	0.6227390	0.6563307	0.7396907	0.6511628	0.6408269	***0.66215***
AUC	0.6413915	0.6847610	0.7778248	0.6779969	0.6497328	***0.6863414***

Table 9: 5-Fold Cross-Validation summary for Naive Bayes Classifier

Model 3: Support Vector Machine (SVM)

SVM involves finding the hyperplane which separates the data points of different target labels with the largest margin. This hyperplane is called the decision boundary the data points closest to it are called the support vectors. Traditional models minimize minimise empirical training error, but SVM intends to minimise the upper bound generalization error and the goal is to maximise the distance of separation between the hyperplane and the data points (Amari & Wu, 1999).

Parameters

- *default* parameters
- *kernel = rbf,* Radial Basis Function was used for this study as it best captures non-linearities in the dataset and is useful for complex problems.

K-Fold Cross Validation method

The method remains the same as specified for the Decision Tree algorithm **above**.

Evaluation Metric	Iteration 1 (1st fold)	Iteration 2 (2nd fold)	Iteration 3 (3rd fold)	Iteration 4 (4th fold)	Iteration 5 (5th fold)	Mean
Accuracy	0.6749436	0.6266968	0.6493213	0.6817156	0.6681716	***0.6601698***
AUC	0.6721256	0.6058973	0.6372649	0.6582422	0.6469180	***0.6440896***

Table 10: 5-Fold Cross-Validation summary for Support Vector Machine Classifier

Model 4: Random Forest (RF)

This classifier is a tree-based algorithm, and it operates by creating random vector samples obtained independently from the provided input vectors (Pal, 2005). This is another ensemble technique which generates multiple classifiers which are then used to predict the class labels.

Parameters

- *default* parameters
- *mtry = sqrt((ncol(clean_df4)-1)):* The *mtry* parameter will control the number of randomly selected features. Here, it is set to the square root of the total number of columns in the dataset (clean_df4), minus 1 (to exclude the target variable)
- *ntree = 1000*: The *ntree* parameter will specify the number of decision trees to generate in this random forest classifier and it is set to 1000 in this case.
- *replace = TRUE*: This parameter determines that sampling is done with replacement.

K-Fold Cross Validation method

The method remains the same as specified for the Decision Tree algorithm above.

Evaluation Metric	Iteration 1 (1st fold)	Iteration 2 (2nd fold)	Iteration 3 (3rd fold)	Iteration 4 (4th fold)	Iteration 5 (5th fold)	Mean
Accuracy	0.6221719	0.6515837	0.6357466	0.6139955	0.6351351	***0.6317266***
AUC	0.6405317	0.6795427	0.6194727	0.6370177	0.6524660	***0.6458061***

Table 11: 5-Fold Cross-Validation summary for Random Forest Classifier

5. Evaluation

5.1. Data

Models are evaluated using the holdout set (20% test set) which was put aside before the k-fold cross validation. The models are tested on unseen data to gauge their efficiency robustness in a new setting and the best overall performer is identified.

5.2. Method

- ***The ROC curve*** is a visual representing the performance of the model by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold values as seen in *Figure 11*.
- ***AUC*** provides a single summarization of the overall performance of the binary classification of the model. It is the area under the ROC curve and explains how well the model distinguishes between positive and negative classes.

- ***The Confusion Matrix*** is used to obtain a summary of various metrics for the performance of the models as in *Table 12*.

Evaluation Metric	Model 1 (DT)	Model 2 (NB)	Model 3 (SVM)	Model 4 (RF)
AUC	0.6333061	0.6671995	***0.6817712***	0.6428541
Accuracy	0.657	0.6348	***0.6582***	0.639
Precision	0.5767	0.5015	***0.6458***	0.5054
Sensitivity (Recall)	0.4977	0.5724	***0.2853***	0.4653
p-value for 95% CI (Accuracy > No Information Rate)	0.006306	0.48694	***0.001485***	0.4489

Table 12: Performance Evaluation for Classifiers

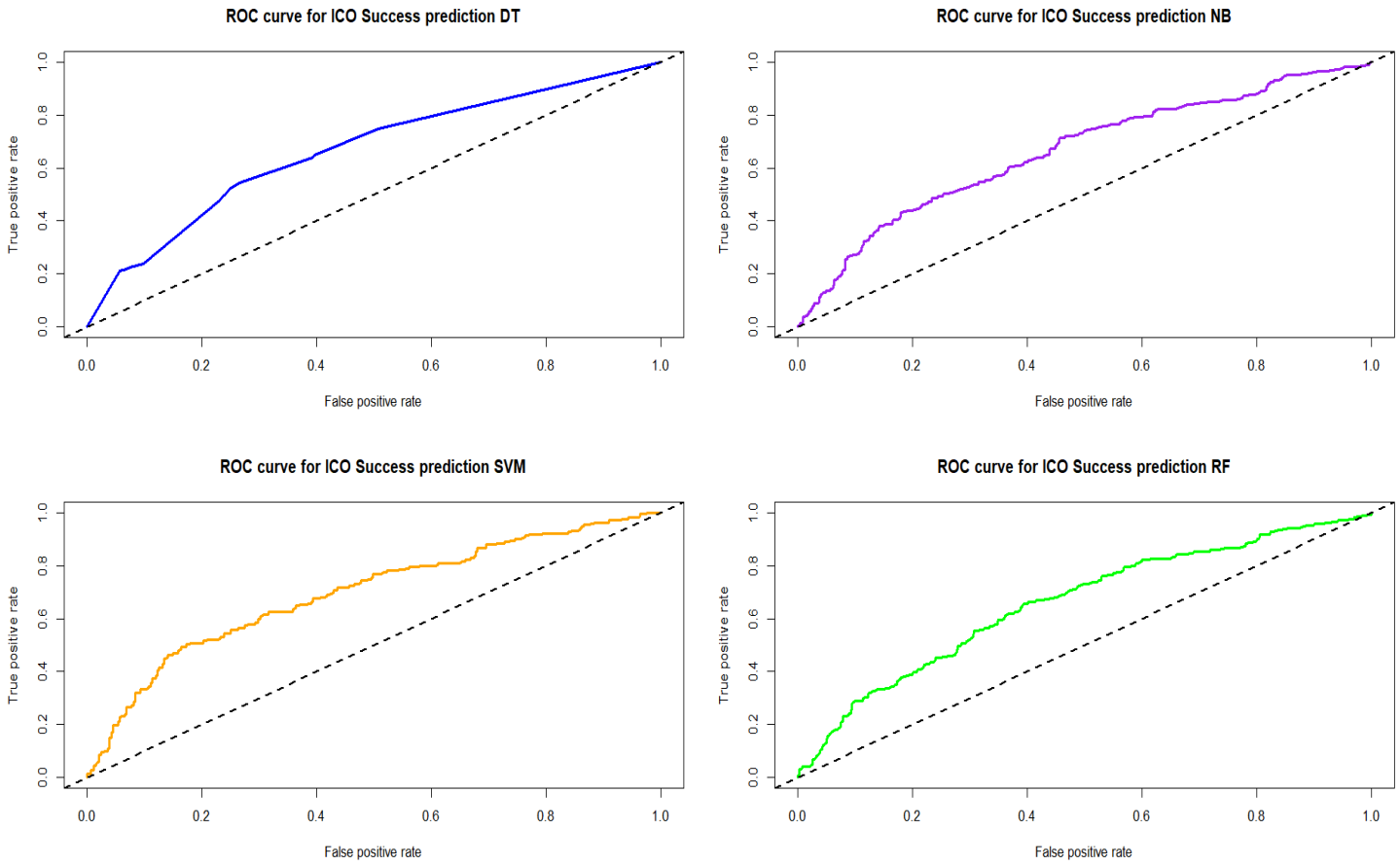

Figure 11: ROC curve for each Classifier

5.3. Conclusion

The Decision Tree Model produces the lowest AUC= but one of the highest Accuracy = 0.657. Precision = 0.5767 which is the ability to rightly predict the positive labels as positive. It also has a decent sensitivity considering the class imbalance where higher number of negative cases (>60% of the total) of the dataset, introduces a bias and affects the model's efficiency to classify the positive cases correctly. Considering all these factors, performance of the decision tree with Ada Boost is statistically significant as proved by the p-value = 0.006306 which states that this performance is not occurring by chance.

The Naïve Bayes model, on the other hand has the lowest Accuracy = 0.6348 but a higher AUC = 0.6671995, however, the p-value = 0. 48694 for the model, does not provide enough statistical evidence for the results obtained not occurring by chance.

The Random Forest classifier performs similar in nature to the Naïve Bayes model and the performance fails to be statistically significant as seen from the p-value = 0.4489 despite obtaining decent values for AUC = 0.6428541, Accuracy = 0.639 and Precision = 0.5054 and Recall = 0.4653.

The Support Vector Machine classifier has the highest accuracy = 0.6582, AUC = 0.6817712, and precision = 0.6458; however, the low sensitivity is due to the label imbalance of the dataset. The p-value = 0.001485 is the lowest and provides the statistical evidence for the performance of this classifier to have not occurred by chance. SVM performs the best for non-linear data and is robust for unseen data as per the analysis conducted.

References

Agrawal, A., Catalini, C., & Goldfarb, A. (2015). Crowdfunding: Geography, Social Networks, and the Timing of Investment Decisions. *Journal of Economics & Management Strategy*.

Ahmad, M. F., Kowalewski, O., & Pisany, P. (2021). What determines initial coin offering success: a cross-countrystudy. *Economics of Iinnovation and New Tecnology*.

Amari, S., & Wu, S. (1999). Improving support vector machine classifiers by modifying kernel. *Neural Networks 12*.

Chauhan, H., & Chauhan, A. (2013). Implementation of decision tree algorithm c4.5. *International Journal of Scientific and Research Publications*.

Cronqvist, H., Siegel, S., & Yu, F. (2015). Value versus growth investing: Why do different investors have different styles? *Journal of Financial Economics*, 333-349.

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

Huang, W., Vismara, S., & Wei, X. (n.d.). Confidence and capital raising. *Journal of Corporate Finance*.

Pal, M. (2005). Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 217-222.

Rish, I. (2001). An empirical study of the naive Bayes classifier. *International Joint Conference on Artificial Intelligence 2001.*