FORECASTING THE FUTURE:MACHINE LEARNING FOR SUCCESSFUL ICO INVESTMENT

Predicting ICO Fundraising Success

Introduction:

Use a variety of machine learning algorithms to predict whether a group or organisation will be able to raise the money it needs from the ICO. The objective of this technical study is to foretell whether or not a certain fundraising team or organisation will be successful in gathering the required funds through an initial coin offering (ICO). Known as "crowdfunding," the activity of utilising the Internet to collect small donations from many individuals is becoming more common. To generate capital, ICOs create and sell digital money, secured by the blockchain, rather than traditional cash. A successful ICO campaign is crucial for a fundraising team or corporation to get the money they need. By creating predictive models, we would be able to evaluate the likelihood of meeting the fundraising goal and provide crucial data for decision making.

The goal is to demonstrate analytical skills by writing a technical report detailing the data understanding, data preparation for modelling and evaluation, modelling techniques applied to the processed data, model evaluation and interpretation, conclusion and report writing.

Data Understanding:

Exploratory data analysis (EDA) will be used to better comprehend the information at hand via the examination of variable distributions, the visualisation of correlations, and the recognition of trends and patterns. These initial exploratory data analysis (EDA) steps provide a good foundation for understanding the data and its characteristics. They allow us to identify any data pre-processing steps that may be required and gain insights into potential predictors for our modelling task.

By performing exploratory data analysis, one reviews the aspects and properties of the data with a "open mind," meaning one does not immediately try to fit the data into a preconceived model. It is used to identify outliers in the data and is typically used upon first seeing the data, prior to the selection of any models for the structural or stochastic components. (Tukey, 1977)

To begin our investigation, we must first familiarise ourselves with the facts at hand.

```
'data.frame':   2767 obs. of  16 variables:
 $ ID                  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ success             : chr  "N" "N" "N" "Y" ...
 $ brandSlogan         : chr  "Is One of Its Kind ERC-20 Decentralized Stable Asset" "The
Ultimate Blockchain Gaming Platform" "Simple Automated Investment App Driven by AI & ML"
"International Real Estate Crowdfunding Platform" ...
 $ hasVideo            : int  1 1 1 1 1 1 1 1 1 1 ...
 $ rating              : num  4 4.3 4.4 4.3 4.3 4.7 4.1 4.5 4.8 4.2 ...
 $ priceUSD            : num  30 0.13 0.01 NA 0.03 0.1 0.02 2.8 50 0.1 ...
 $ countryRegion       : chr  "Singapore" "Malta" "UK" "Netherlands" ...
 $ startDate           : chr  "01/10/2019" "07/09/2018" "01/07/2019" "01/10/2019" ...
 $ endDate             : chr  "01/10/2019" "12/10/2018" "30/06/2020" "15/12/2019" ...
 $ teamSize            : int  31 20 10 27 14 43 20 31 8 29 ...
 $ hasGithub           : int  1 1 1 1 1 1 1 1 1 1 ...
 $ hasReddit           : int  1 1 1 1 1 1 1 1 1 1 ...
 $ platform            : chr  "Ethereum" "XAYA" "Stellar" "Separate blockchain" ...
 $ coinNum             : num  5.10e+05 2.25e+08 5.00e+09 1.25e+08 5.00e+09 ...
 $ minInvestment       : int  0 1 1 1 1 1 1 1 1 1 ...
 $ distributedPercentage: num  0.49 0.41 0.4 0.13 0.5 0.5 0.25 0.1 0.05 0.15 ...
```

Figure 1: Structure of the Data

An overview of the dataset's structure, variable types, and prospective data preparation tasks may be gained from the above comprehension. There are 16 different variables in this dataset, which has 2,767 rows of data.

To indicate whether or not a project received enough money to finish, the "success" variable is presently being kept as a character (chr) data type.

```
    coinNum             priceUSD          teamSize        rating      distributedPercentage
 Min.   :1.200e+01   Min.   :    0.00   Min.   : 1.00   Min.   :1.000   Min.   :  0.000
 1st Qu.:5.000e+07   1st Qu.:    0.04   1st Qu.: 7.00   1st Qu.:2.600   1st Qu.:  0.400
 Median :1.800e+08   Median :    0.12   Median :12.00   Median :3.100   Median :  0.550
 Mean   :8.178e+12   Mean   :   19.01   Mean   :13.11   Mean   :3.121   Mean   :  1.061
 3rd Qu.:6.000e+08   3rd Qu.:    0.50   3rd Qu.:17.00   3rd Qu.:3.700   3rd Qu.:  0.700
 Max.   :2.262e+16   Max.   :39384.00   Max.   :75.00   Max.   :4.800   Max.   :869.750
                     NA's   :180        NA's   :154
```

Figure 2: Summary of the data variables (Data Understanding)

Figure 2 provides information on the range of values for each of the variables, including coinNum, priceUSD, teamSize, rating, and distributedPercentage, as well as their lowest, maximum, mean, median, and quartile values.

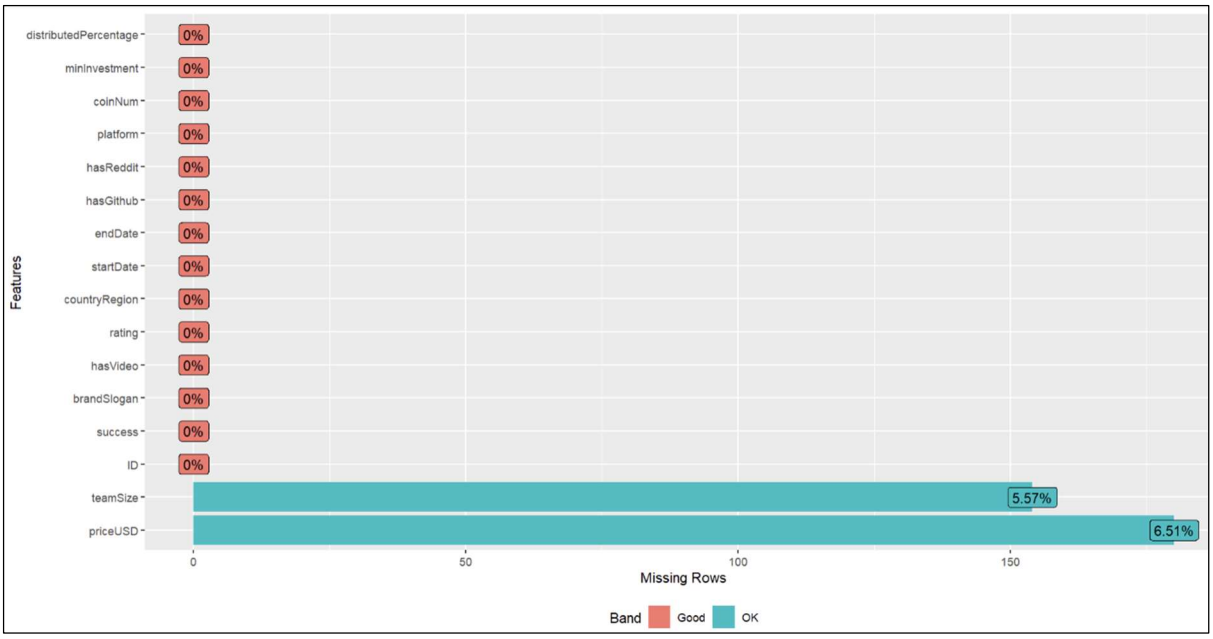
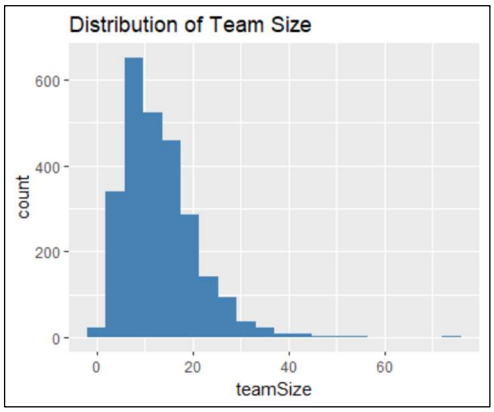
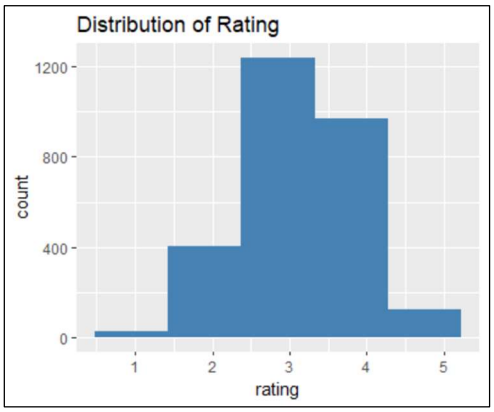


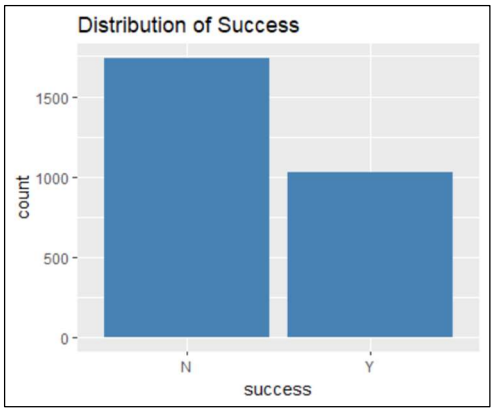
Figure 3: Plot of Missing values in the dataset

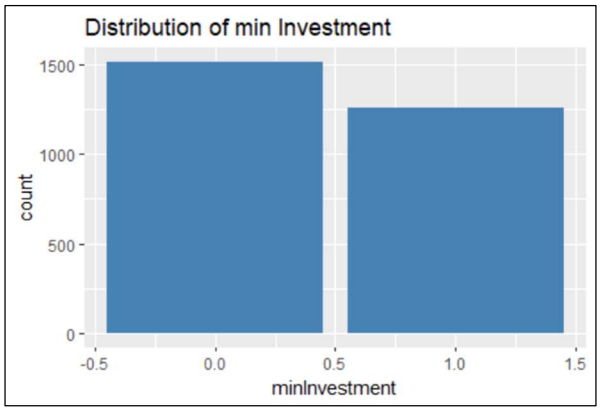
The dataset has some missing data points. As can be seen in Figures 2 and 3, there are 154 and 180 blanks in the teamSize and priceUSD variables, respectively, which accounts for 6.51% and 5.57% of their total missing data.

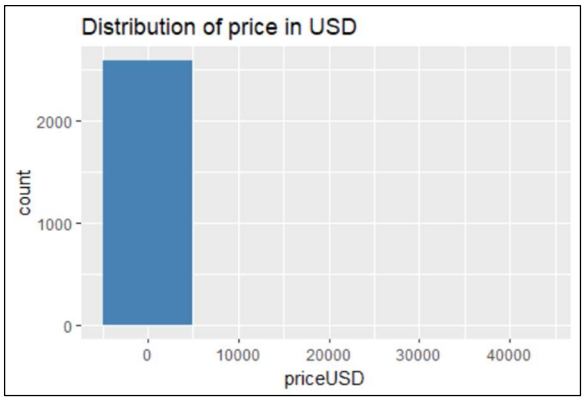
1. <u>Variable Distributions:</u>
   First, we will look at how the various numeric factors, such priceUSD, teamSize, rating, minimumInvestment, and distributedPercentage, are distributed. In order to see how they are distributed and spot any outliers or skewness in the data, we may make histograms or density charts.







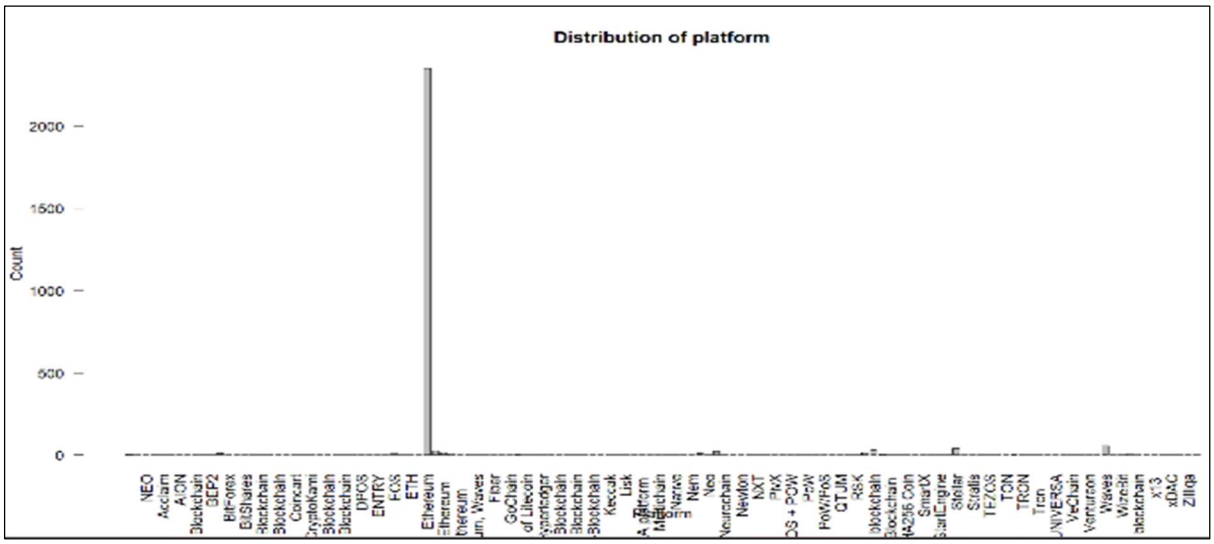




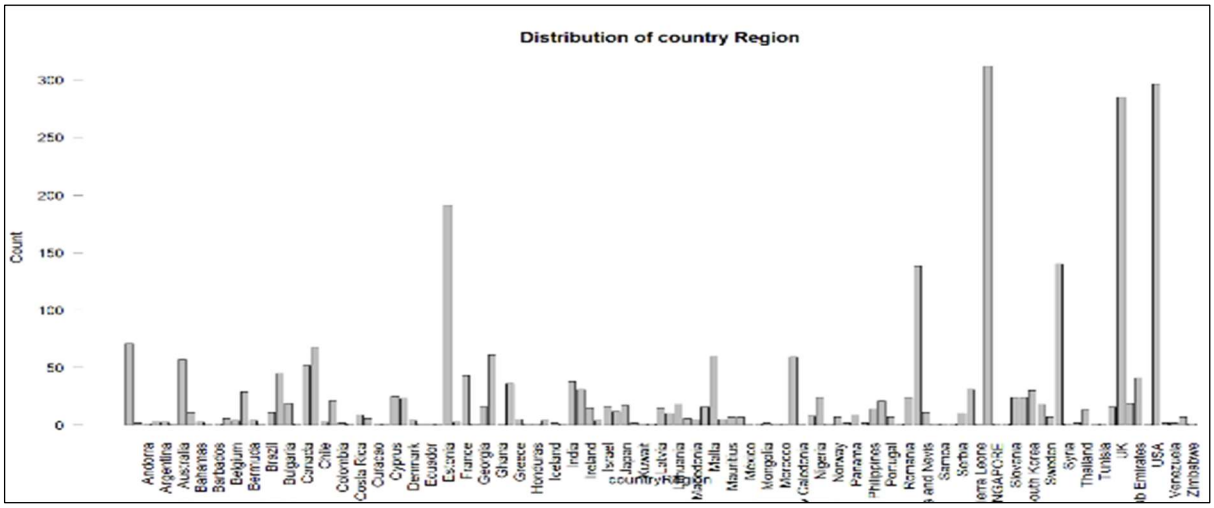
The distributions of the variables have been shown to be uniform, however the distribution of the size of the teams has been found to be positively skewed, also known as a right skew distribution.

2. <u>Categorical Variables:</u>

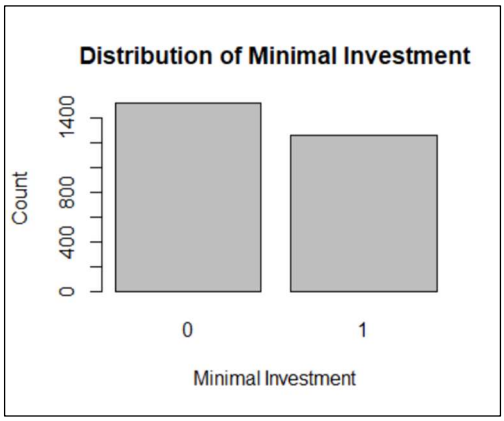
Following that, we will do an analysis of the category factors, such as countryRegion and platform, in addition to the binary indicator variables (minInvestment, hasVideo, hasGithub, and hasReddit). It is possible for us to generate bar plots or pie charts in order to visualise the frequency distribution of each category and identify any categories that are particularly common or uncommon. Because of this study, we will have a better understanding of the distribution of projects across a variety of countries and platforms, as well as the existence of certain characteristics, such as minimum investment, videos, Github, or Reddit sites.

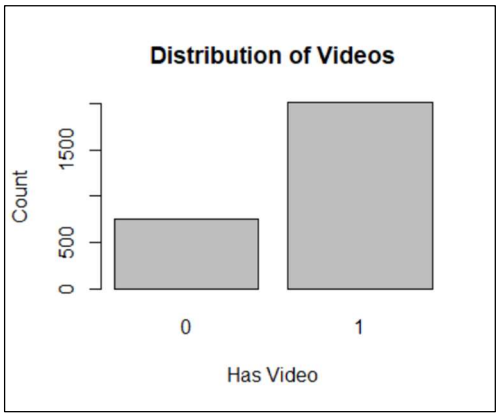


Teams and businesses utilise the platform named "ethereum" at a disproportionately high rate compared to other values.

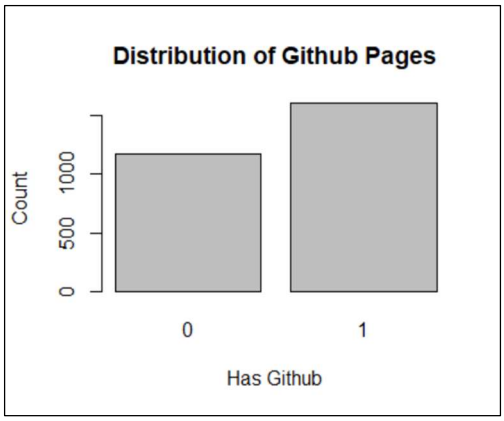


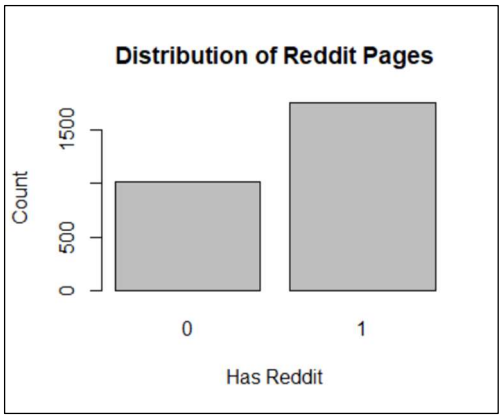
There is a regular distribution seen in the breakdown of countries, regions. The United States and the United Kingdom both have many teams and businesses, but Singapore has the most.





It can be seen from the frequency distribution of the binary indicator of minimal investment that there is a nearly equal distribution. On the other hand, the frequency distribution of binary indicator videos reveals that there are a much greater number of companies or teams that have videos than there are companies or teams that do not have videos.

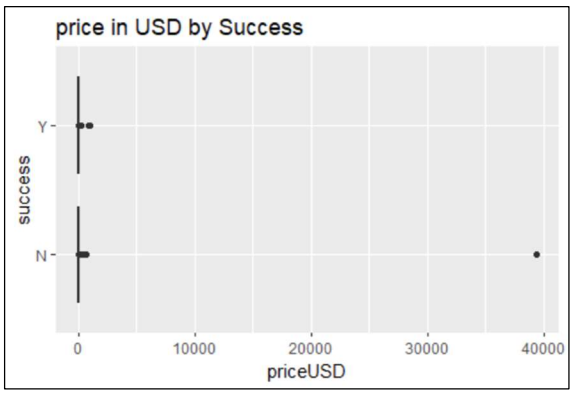


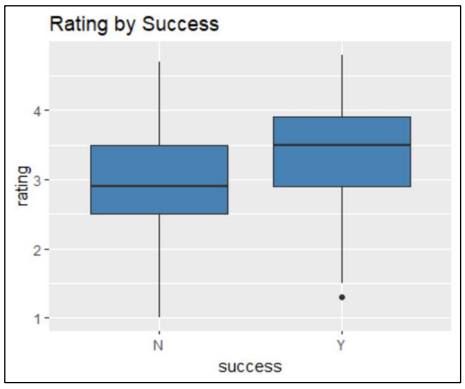


The binary indication Github has a higher frequency of occurrence among teams and businesses that do have Github than among those that do not. In a similar vein, the binomial distribution of Reddit site frequencies reveals that a greater proportion of teams and businesses have a Reddit presence than do not.

3. Relationships and Patterns:

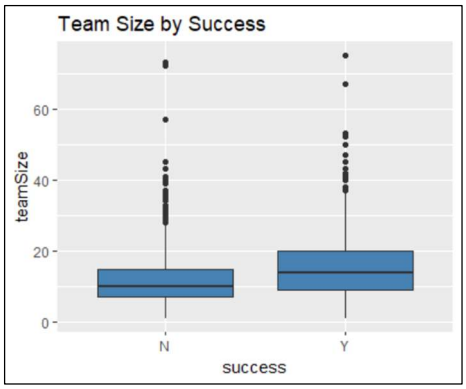
Scatter plots, box plots, and violin plots are just some of the tools we may use to see how different variables are connected. Success (the goal variable) may be correlated with other variables such as priceUSD, teamSize, rating, and distributedPercentage, among others. We can see if there are any trends or patterns in the distributions of these variables that could suggest the effect of these predictors on fundraising success by comparing the distributions or summary statistics of these variables for successful and failed initiatives.

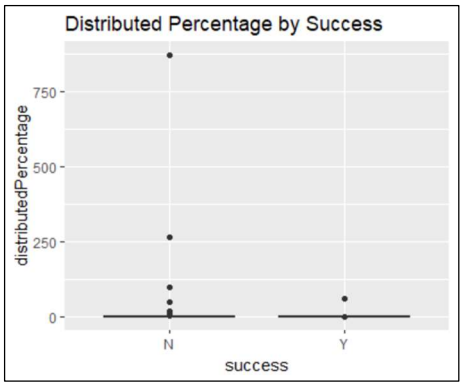




If we look at a scatter plot of price in USD by Success, we don't see a strong relationship between the priceUSD variable and the success attribute. However, the data reveals that the priceUSD variable has a value that is very out of the ordinary when compared to the rest of the values in the dataset.

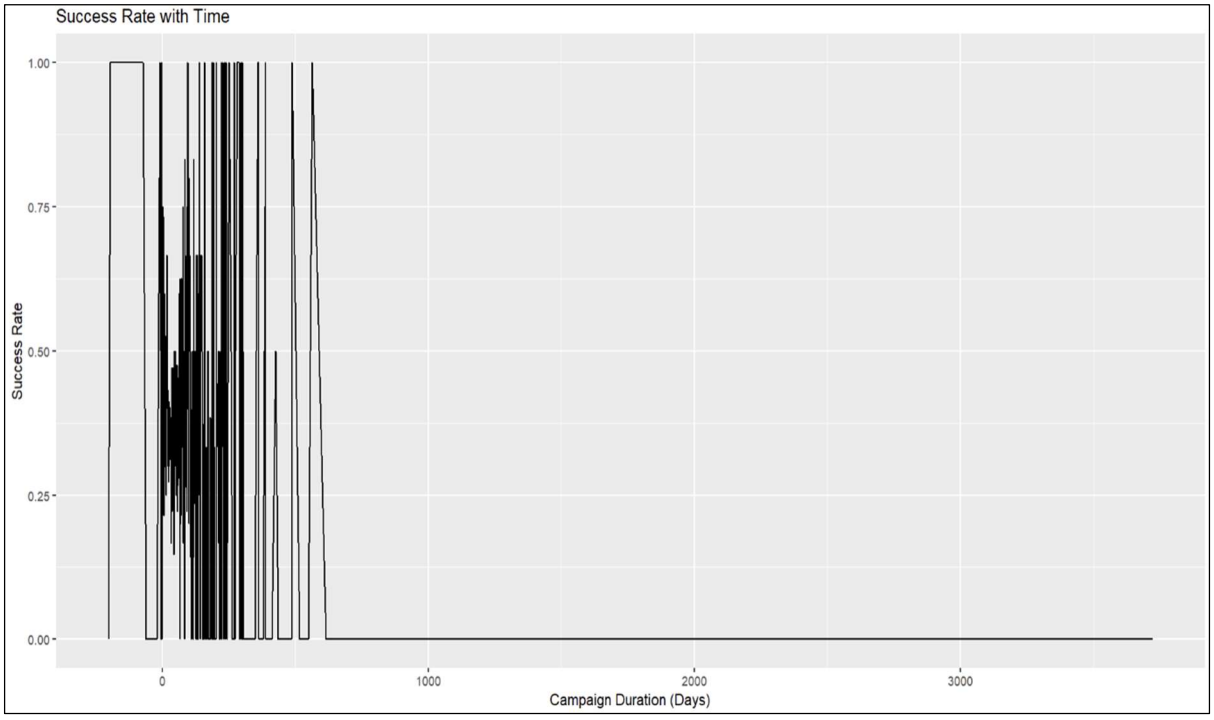
There is some evidence of a relationship between rating and success rate in the box displaying ratings by rate of success.





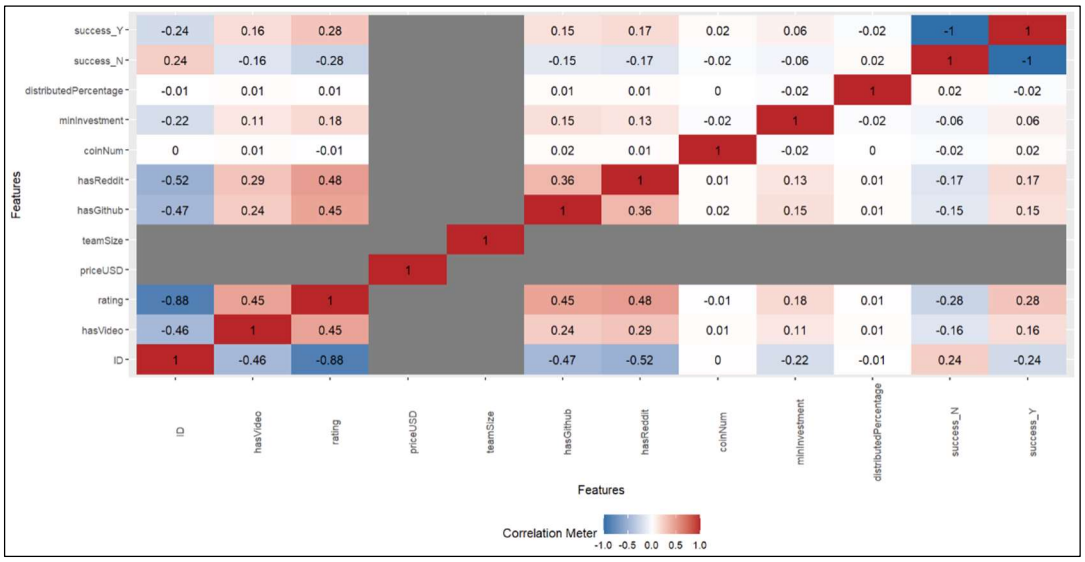
There is a link between the success rate and the size of the team, as well as the distributed percentage variable, which also has an extreme value, making it an outlier.

4. <u>Time-based Analysis:</u>
   We may examine the relationship between campaign length and total funds raised by using the startDate and endDate variables in a time series analysis. Each campaign's lifespan may be estimated, allowing us to see whether there's a connection between effort put in and successful funding. For this investigation, line plots or bar graphs displaying the success rate with time may be very valuable visualisations.



5. Correlation Analysis:

Coefficients of correlation (e.g., Pearson correlation) and a correlation matrix may be computed to help us make sense of the connections between two numerical variables. The presence of significant positive or negative correlations between variables may be used as an indicator of multicollinearity problems or dependencies, and this can be accomplished by using this method.



(Turney, 2022) The most popular tool for quantifying linear relationships is the Pearson correlation coefficient (r). It's a number between -1 and 1 that indicates how strongly and in what direction two variables are related to one another. As a descriptive statistic, the Pearson correlation coefficient is useful for quickly summing up a dataset's key features. The magnitude and direction of a linear connection between two numerical variables are described.

- Positive correlation is shown by a Pearson correlation coefficient (r) value between 0 and 1 - when one variable changes, the other variable also changes in the same direction.
- Value 0 of the Pearson correlation coefficient (r) No link exists between the variables; there is no correlation.
- Between 0 and -1, the Pearson correlation coefficient (r) value demonstrates When one variable changes, the other variable changes in the opposite way, which is known as a negative correlation. (Turney, 2022)

We learned a lot about the dataset, found problems with the data quality, found key predictors, and now will use this knowledge to influence the further data preparation and modelling procedures.

Data Preparation:

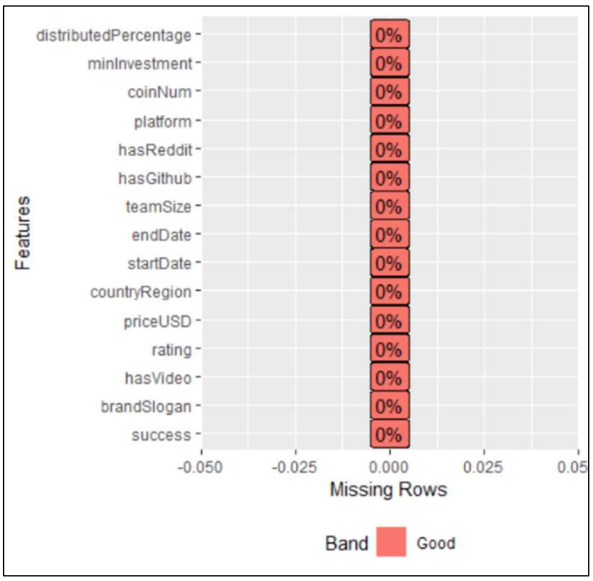
In order to begin the modelling process, the raw data must first be prepared. In order to fix this, the data format must be converted, and missing values, inaccurate values, and outliers must be dealt with. We may develop additional variables that expand upon the existing set in order to enhance our capacity to predict results. By comparing the start and finish dates, we can estimate how long the fundraising campaign was, and by looking at the value of the countryRegion variable, we can find out where the fundraising team is from.

1. Handling Missing Values:

There are missing values in the priceUSD and distributedPercentage variables of the dataset.

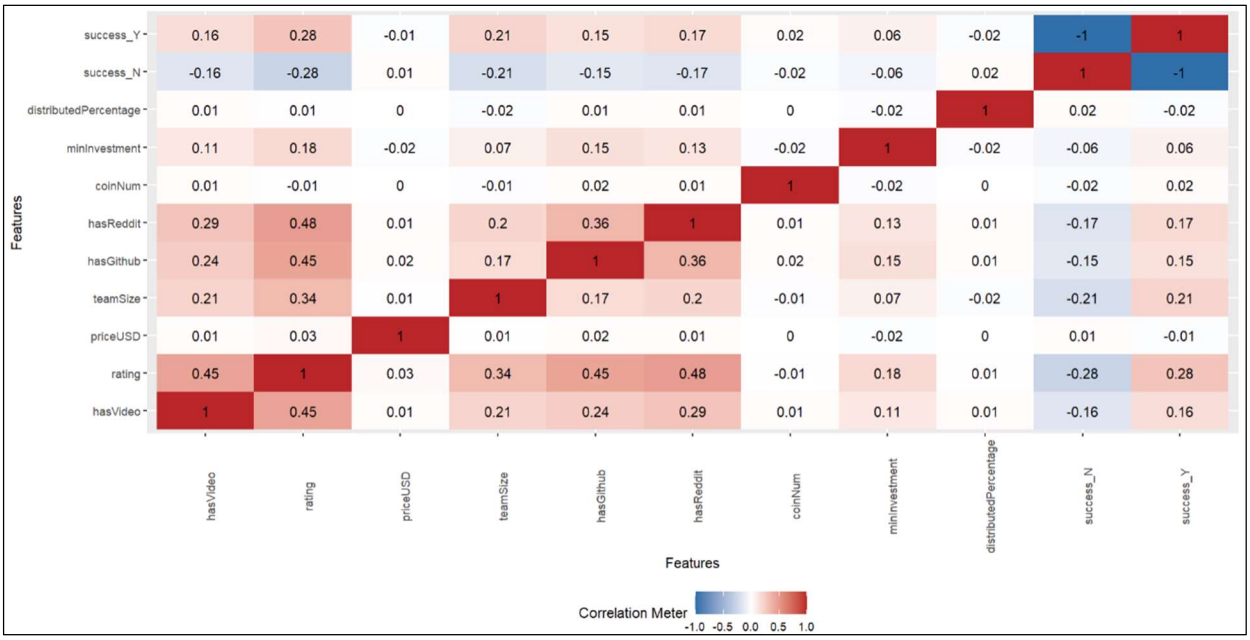
Using Multiple Imputation to handle missing values:

To generate multiple imputations, numerous plausible values for each missing element are generated. This approach works well with complicated datasets because it takes into account the uncertainty of imputed values. Multiple imputation methods may be implemented in R using packages like mice, which use chained equations (MICE).



After doing imputation to address the missing values in the priceUSD and distributedPercentage variables, here is a plot of all the variables that are present in the dataset.

It is clear from looking at the graphic that the dataset does not include any more missing values.



The above figure is a correlation matrix plot which tells that -

hasVideo variable has a moderate and positive correlation with the rating variable, hasVideo also has a weak but positive correlation with hasReddit, hasGithub, teamSize, and success(Yes).

Rating variable has a moderately positive correlation with hasReddit, followed byhasGithub, teamSize, Succes(Yes), and minInvestment.

priceUSD has almost no correlation with any of the other variables.

teamSize variable has a weak but positive correlation with hasGithub, followed by Success(Yes).

hasGithub variable has a moderately strong correlation with hasReddit, and a weak but positive correlation with Success(yes) and minInvestment variables.

hasReddit variable has a weak but positive correlation with success(yes) and minInvestment variables.

coinNum has almost no correlation with any of the other variables.

minInvestment has a weak but positive correlation with success(yes).

distributedPercentage has almost no correlation with any of the other variables.

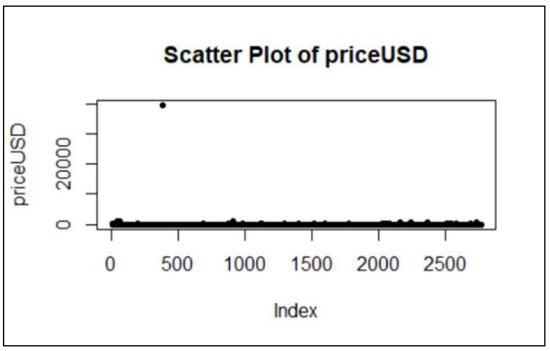
2. Handling Outliers:

(Enderlein, 1987) An outlier is a value or observation that stands in stark contrast to the rest of the data, making it stand out as an anomaly. However, Enderlein (1987) takes it a step further by defining outliers as numbers that depart so much from the mean that one may infer a distinct sampling technique.

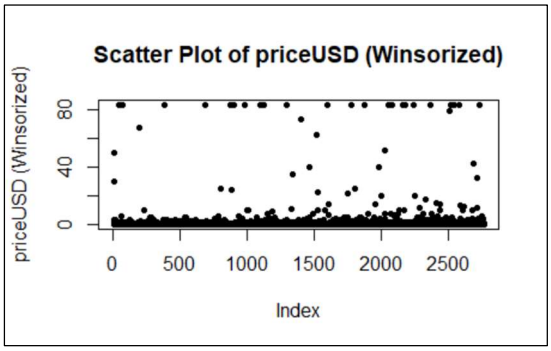
Values that greatly depart from the norm are considered outliers. The effectiveness and efficiency of machine learning models may suffer as a result of them.

Winsorization for handling Outliers:

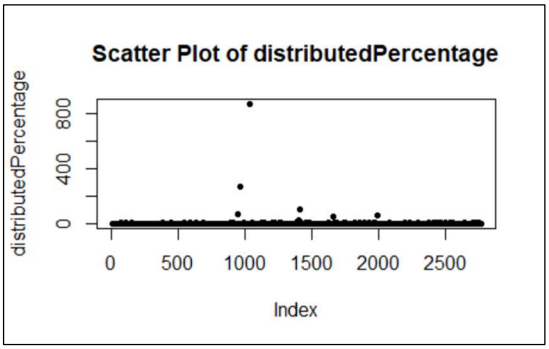
Replace values that are at the extremes of a range with the values that are closest to them within that range (for example, replace values that are above the 95th percentile with the value at the 95th percentile).

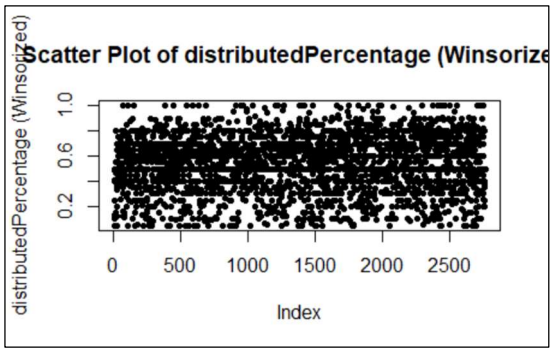


The scatterplot of priceUSD reveals that, with the exception of one value, all of the values for the priceUSD variable are less than 100. There is one priceUSD value that is more than 2,000, making it an extreme or outlier number.



After using winsorization to remove the outlier, the priceUSD data has been shown in this scatterplot. There are no longer any values that are extreme.





The distributedPercentage variable's scatterplots are seen above, both before and after the application of winsorization to deal with outliers.

Modelling:

In this section, we will go through the many methods that have been proposed to classify ICOs so as to predict their success. Models such as support vector machines, decision trees, and k nearest neighbour may all be used. We will explore the merits and shortcomings of each model and their

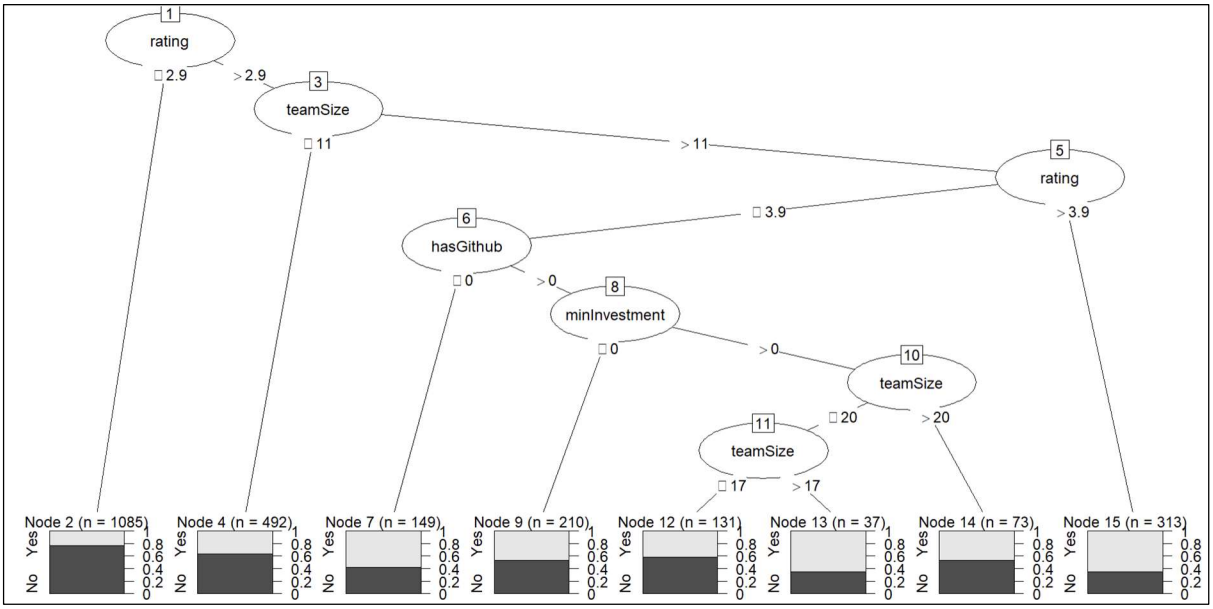
suitability for the task at hand. Models will be chosen based on how well they account for the specifics of the dataset and how easy they are to read or how complicated they are to implement.

1. K-Nearest Neighbours (KNN):

The KNN technique for classifying data is easy to understand and implement. Using the known classes of the K closest neighbours in the feature space, it assigns a label to a new data point. The projection is based on the typical membership of the K nearest neighbours. KNN does not assume anything about the distribution of the data it is using, since it is a non-parametric method.

2. Decision Tree:

A decision tree is a diagram with core nodes that stand in for features, branches that stand in for decision rules, and leaf nodes that stand in for the final conclusion or class label. The decision tree method iteratively divides the data into subgroups with similar levels of the target variable depending on the feature values. Both category and quantitative information may be processed using decision trees.



Classification Tree

Number of samples: 2490

Number of predictors: 9

Tree size: 8

3. Support Vector Machine (SVM):

SVM is an efficient technique for performing classification and regression analyses. In a setting of classification, SVM seeks a hyperplane that most effectively divides data into distinct classes. Depending

on the kernel function used, support vector machines may perform either linear or non-linear class separation.

```
> success_classifier
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 1

Linear (vanilla) kernel function.

Number of Support Vectors : 1592

Objective Function Value : -1585.279
Training error : 0.323091
```

```
> agreement <- success_predictions == SVM_ICO_imputeddata_test$success
> table(agreement)
agreement
FALSE  TRUE
  204   350
> prop.table(table(agreement)) # the prop of 'TRUE' is the recognition accuracy
agreement
   FALSE     TRUE
0.368231 0.631769
```

```
> agreement_rbf <- success_predictions_rbf == SVM_ICO_imputeddata_test$success
> table(agreement_rbf)
agreement_rbf
FALSE  TRUE
  198   356
> prop.table(table(agreement_rbf))
agreement_rbf
    FALSE      TRUE
0.3574007 0.6425993
```

Each of these three models takes a somewhat different tack when it comes to classifying data, and each has its advantages and disadvantages. We can determine which model is best suited to this problem by applying various models and comparing their results.

Model Evaluation:

To evaluate the models, we relied on accurate measurements and established validation methods. In order to train and test our model, we have split the dataset in two. We will use many metrics, such as accuracy, sensitivity, specificity, Error rate, precision, recall, F-measure, and ROC & AUC, to assess the performance of the models. We will compare and contrast the strengths and weaknesses of the different models, as well as the results they create.

Following are the Model Evaluation Measures:

1. Confusion Matrix:
   A matrix that ranks how well predicted labels match the real ones.
   Offers a detailed analysis of the expected and observed class labels, including the proportions of each (true positives, true negatives, false positives, and false negatives).

KNN actual success	KNN predicted success: Yes	No	Row Total
Yes	3	197	200
	0.015	0.985	0.241
	0.300	0.240	
	0.004	0.237	
No	7	623	630
	0.011	0.989	0.759
	0.700	0.760	
	0.008	0.751	
Column Total	10	820	830
	0.012	0.988	

DT actual success	DT predicted success: Yes	No	Row Total
Yes	29	67	96
	0.302	0.698	0.347
	0.617	0.291	
	0.105	0.242	
No	18	163	181
	0.099	0.901	0.653
	0.383	0.709	
	0.065	0.588	
Column Total	47	230	277
	0.170	0.830	

SVM actual success	SVM predicted success: N	Y	Row Total
N	283	53	336
	0.842	0.158	0.606
	0.661	0.421	
	0.511	0.096	
Y	145	73	218
	0.665	0.335	0.394
	0.339	0.579	
	0.262	0.132	
Column Total	428	126	554
	0.773	0.227	

2. Accuracy:

   Accuracy is the ratio of correct predictions with the total number of predictions made. Accuracy determines how accurate the model is in general, measuring how often it makes accurate predictions.

```
> print(paste("KNN Accuracy:", KNN_accuracy))
[1] "KNN Accuracy: 0.754216867469879"
> print(paste("DT Accuracy:", DT_accuracy))
[1] "DT Accuracy: 0.693140794223827"
> print(paste("SVM Accuracy:", SVM_accuracy))
[1] "SVM Accuracy: 0.642599277978339"
```

   The K-Nearest Neighbours (KNN) model is the most accurate, followed by the Decision Tree (DT) model, and then the Support Vector Machine (SVM) model, which is the least accurate of the three.

3. Error Rate:

   Error Rate = 1- Accuracy

```
> print(paste("KNN Error Rate:", KNN_error_rate))
[1] "KNN Error Rate: 0.245783132530121"
> print(paste("DT Error Rate:", DT_error_rate))
[1] "DT Error Rate: 0.306859205776173"
> print(paste("SVM Error Rate:", SVM_error_rate))
[1] "SVM Error Rate: 0.357400722021661"
```

   A KNN model has a low error rate, whereas an SVM model has a high error rate.

4. Sensitivity (True Positive Rate - TPR):

   Evaluation of the model's efficiency in detecting positive cases.
   Sensitivity measures how accurately true positives can be identified.

```
> print(paste("KNN Sensitivity (True Positive Rate):", KNN_sensitivity))
[1] "KNN Sensitivity (True Positive Rate): 0.015"
> print(paste("DT Sensitivity (True Positive Rate):", DT_sensitivity))
[1] "DT Sensitivity (True Positive Rate): 0.302083333333333"
> print(paste("SVM Sensitivity (True Positive Rate):", SVM_sensitivity))
[1] "SVM Sensitivity (True Positive Rate): 0.334862385321101"
```

   In terms of sensitivity, the SVM model is superior to the DT model and superior to the KNN model. In most cases, the SVM model has made accurate predictions.

5. Specificity (True Negative Rate - TNR):

   Measures the model's ability to correctly identify negative instances.
   Specificity measures how many real negatives were correctly classified.

```
> print(paste("KNN Specificity (True Negative Rate):", KNN_specificity))
[1] "KNN Specificity (True Negative Rate): 0.988888888888889"
> print(paste("DT Specificity (True Negative Rate):", DT_specificity))
[1] "DT Specificity (True Negative Rate): 0.900552486187845"
> print(paste("SVM Specificity (True Negative Rate):", SVM_specificity))
[1] "SVM Specificity (True Negative Rate): 0.842261904761905"
```

   The specificity of the KNN model is the greatest, followed by the DT model, and then the SVM model, which has the lowest specificity. The vast majority of false positives may be appropriately classified using the KNN model.

6. Precision:

   The percentage of correct predictions relative to the total number of correct predictions.
   Precision measures how many of positive predictions are truly positive
   Precision is also called as Positive Predictive Value (PPV)

```
> print(paste("KNN Precision:", KNN_precision))
[1] "KNN Precision: 0.3"
> print(paste("DT Precision:", DT_precision))
[1] "DT Precision: 0.617021276595745"
> print(paste("SVM Precision:", SVM_precision))
[1] "SVM Precision: 0.579365079365079"
```

   The precision or PPV of the DT model is greatest, followed by the SVM model, and finally by the KNN model.

7. F-measure:

   Finds a happy medium between precision and recall by averaging the two measurements harmonically.
   The F-measure takes into account both the accuracy and the reliability of a model by combining their respective values.
   The F-score or F-1 is another name for the F-measure.

```
> print(paste("KNN F-measure:", KNN_fmeasure))
[1] "KNN F-measure: 0.0285714285714286"
> print(paste("DT F-measure:", DT_fmeasure))
[1] "DT F-measure: 0.405594405594406"
> print(paste("SVM F-measure:", SVM_fmeasure))
[1] "SVM F-measure: 0.424418604651163"
```

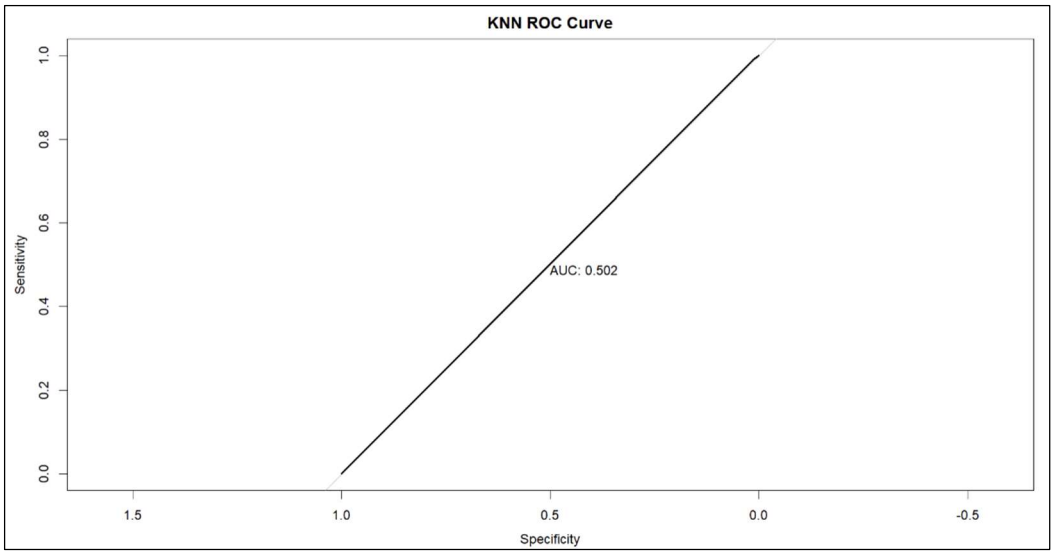
   SVM model has the highest F score, followed by DT model and KNN model has the worst F score.

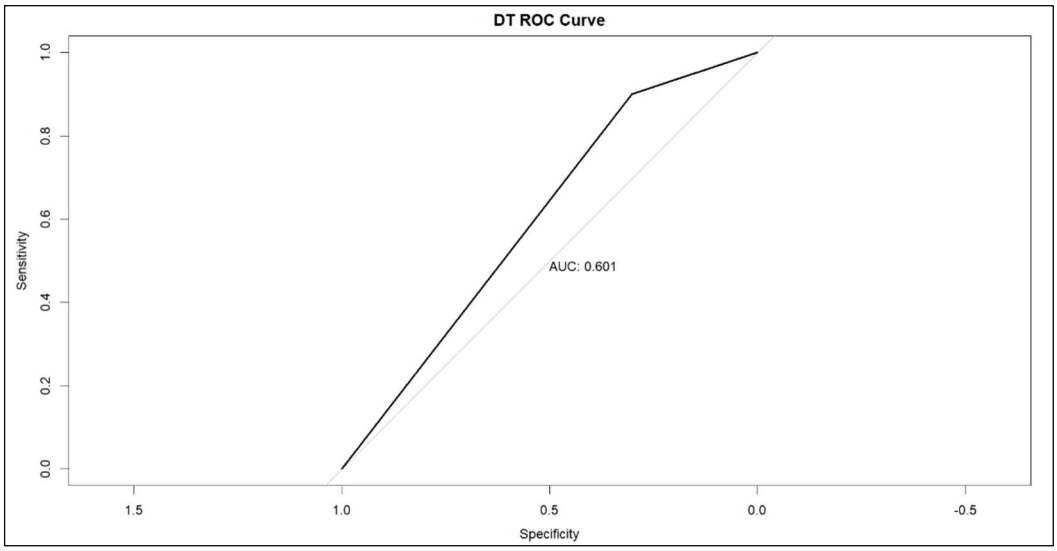
8. ROC Curve (Receiver Operating Characteristic):

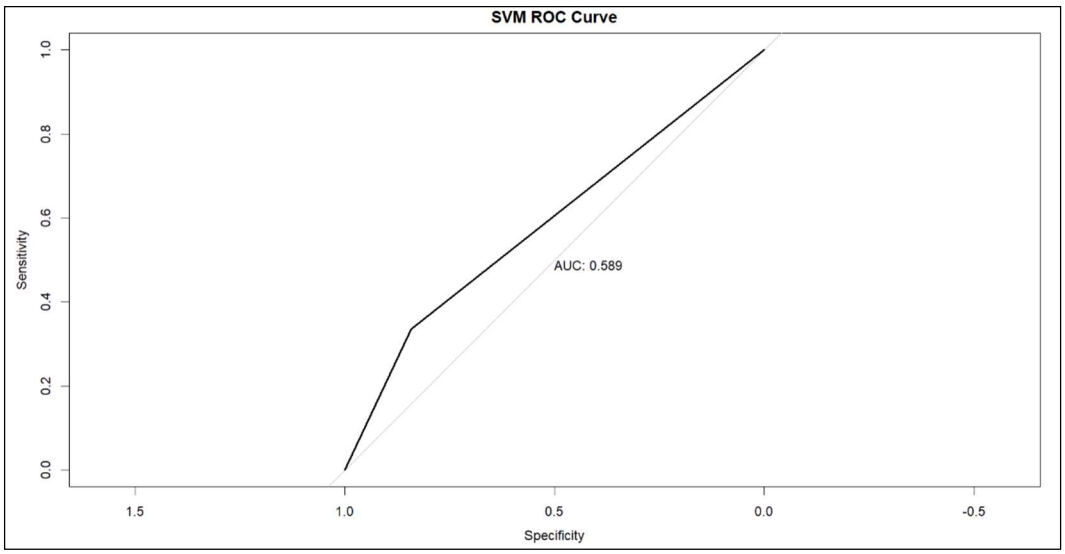
   Displays the model's performance over a range of categorization criteria by plotting the true positive rate versus the false positive rate.

   AUC (Area Under the ROC Curve):

   Area under the receiver operating characteristic (ROC) curve provides a numerical measure of the model's performance. The AUC increases as performance increases.







KNN AUC: 0.502

DT AUC: 0.601

SVM AUC: 0.589

As we know that a higher AUC indicates better performance, DT model has the highest AUC i.e. DT model is the best performer, followed by SVM model and then KNN.

Conclusion (Deployment):

In conclusion, the fundraising teams and companies attempting ICOs can benefit from the insights and predictive models we uncovered in our analysis of the ICO dataset. To begin, we investigated what factors are associated with successful ICO campaigns and why.

Then, the evaluation metrics reveal how well the K-Nearest Neighbours (KNN), Decision Tree (DT), and Support Vector Machine (SVM) models predict ICO campaigns' success.

The **KNN model** has the highest **accuracy**, followed by the DT model and the SVM model, according to the accuracy metric. This shows that the KNN model has the highest proportion of accurate predictions to total predictions.

The error rate, which is the complement of accuracy, shows that the **KNN model** has a **low error rate**, while the SVM model has a higher error rate.

The sensitivity of the models is how well they can identify true positives. The **SVM model** is **superior** to the DT model and the KNN model in terms of sensitivity.

The models' specificity is evaluated by how well they can spot errors. The **KNN model** has the highest **specificity**, followed by the DT model, and then the SVM model.

Precision, also known as Positive Predictive Value (PPV), is a metric used to assess how many positive predictions actually turn out to be correct. The **DT model** is the most **precise**, then the SVM model, and finally the KNN model.

Based on the **F-measure**, which takes into account both precision and recall, the **SVM model** comes out on top, followed by the DT model, and finally the KNN model, which fares the worst.

The models' performance in classifying data according to various criteria can be evaluated using the ROC curve and its corresponding AUC (Area Under the ROC Curve). The **DT model** outperforms the SVM and KNN models and has the highest area under the curve (**AUC**).

In conclusion, the results of the evaluation metrics show that the **Decision Tree model is the most effective** of the three in forecasting the outcome of ICO campaigns, followed by the KNN model. The results should be interpreted with caution, though, given the data's limitations and the larger context. It's possible that the models' performance could benefit from additional research and testing.

References:

TUKEY, J. W. 1977. *Exploratory data analysis*, Reading, MA.

Turney, S. (2022). Pearson Correlation Coefficient (r) | Guide & Examples. *Scribbr*. https://www.scribbr.com/statistics/pearson-correlation-coefficient/

Enderlein, G. 1987. Hawkins, D. M.: Identification of Outliers. Chapman and Hall, London – New York 1980, 188 S., £ 14, 50. *Biometrical Journal*. **29**(2), pp.198–198.

AWAD, M. & KHANNA, R. 2015. *Efficient Learning Machines : Theories, Concepts, and Applications for Engineers and System Designers,* Berkeley, CA, UNITED STATES, Apress L. P.

Rubin, D. B. (1987), Multiple Imputation for Nonresponse in Surveys, New York, John Wiley & Sons.