Leeds University Business School

UNIVERSITY OF LEEDS

Assessed Coursework Coversheet

For use with *individual* assessed work

Student ID Number:	2	0	1	5	3	8	9	9	6
Module Code:	LUBS5990M								
Module Title:	Machine Learning in Practice								
Module Leader:	Dr Xingjie Wei								
Declared Word Count:	3500								

1.Introduction

In the recent years ICO's (initial coin offerings) have been getting a lot of traction. ICOs are becoming increasingly popular because of their creative fundraising strategy, the possibility for significant returns, innovation-friendly culture, and worldwide reach. Due to these benefits not just niche players but even traditional investors are interested in investing in ICO (Reiff, 2022).

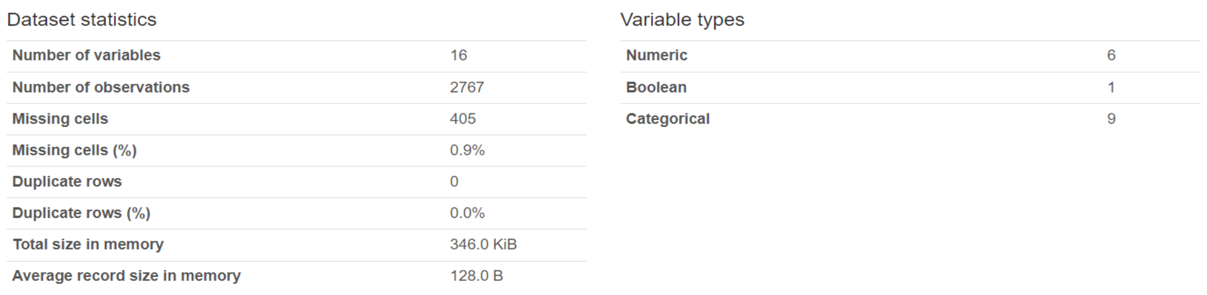
But ICO's have certain potential drawbacks due to which they are classified as high-risk investment vehicles; the absence of a proper legal and regulatory procedures increases the chances of unethical practices and frauds. Since cryptocurrencies compared to other asset classes are highly volatile and ICO's suffers from the same fate. ICO employ the use of the blockchain which makes it an ideal tool for rapid innovation as it eliminated lot of middlemen, which also creates lack of transparency which is another major problem that plagues this industry. This information asymmetry makes it difficult for investors to assess the viability of ICO projects making it challenging to realise the investment potential of the team (Reiff, 2022).

There is dire need for understanding what factors lead to a successful ICO project. The fundraising team sets a target for how much money they want to raise under the "all-or-nothing" framework that is used in a majority of ICO projects. The ICO will be deemed successful if they are able to raise enough money to meet its objective. If not, all their efforts to raise money will be for nothing, and they won't get any money. The objective of this report is to use machine learning to build a classification model on the provided dataset, which considers different attributes of ICO projects to predict if a team will be successful in obtaining funds via an initial coin offering.

2.Data Cleaning and Preparation

For understanding and studying the noise in the dataset a python library ydata_profiling was used to generate a data profiling report for the dataset.

Figure 2.A



Dataset statistics

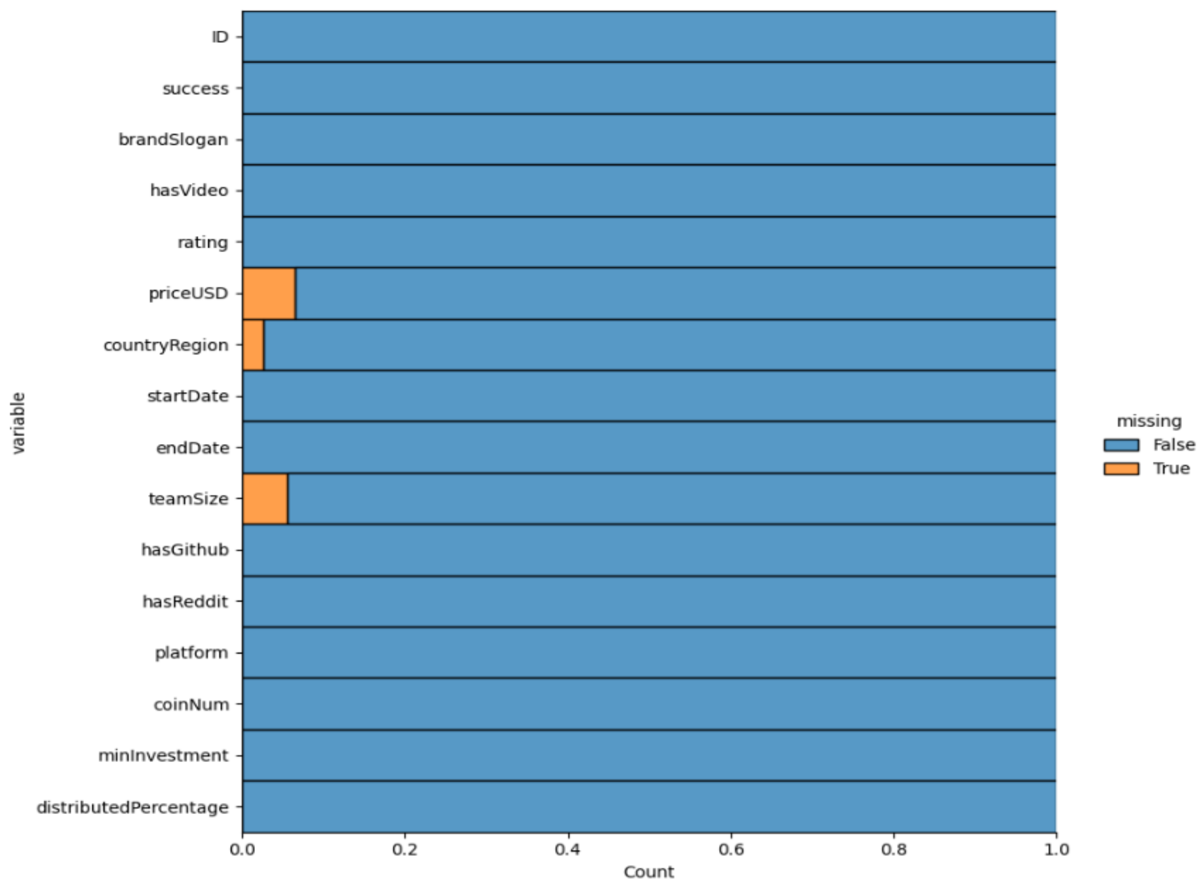
Number of variables	16
Number of observations	2767
Missing cells	405
Missing cells (%)	0.9%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	346.0 KiB
Average record size in memory	128.0 B

Variable types

Numeric	6
Boolean	1
Categorical	9

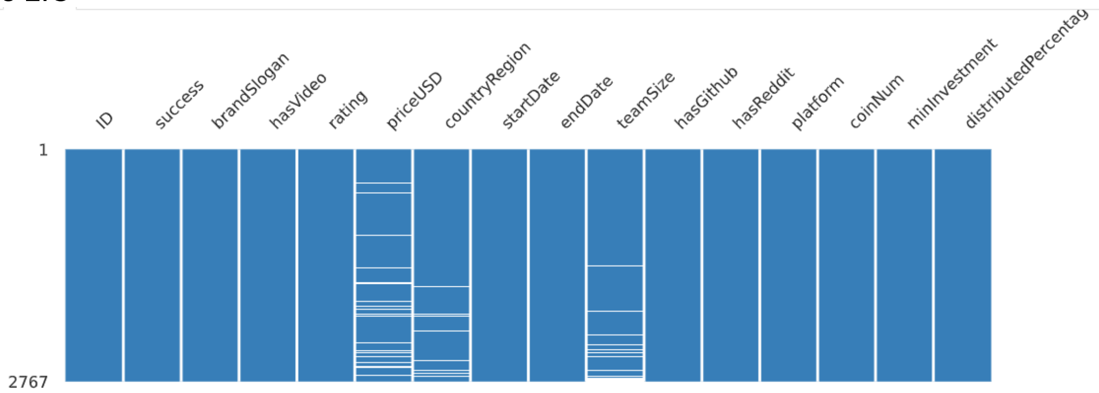
The provided dataset has multiple attributes as can be seen from the above figure there are in total 16 variables and 2767 observations. In total 6 are numeric and 9 are categorical and 1 is Boolean. There appears to be missing values in the datasets. Since the percentage of missing values is only 0.9%, we can safely impute the missing values. The next step is to visualise the missing values.

Figure 2.B



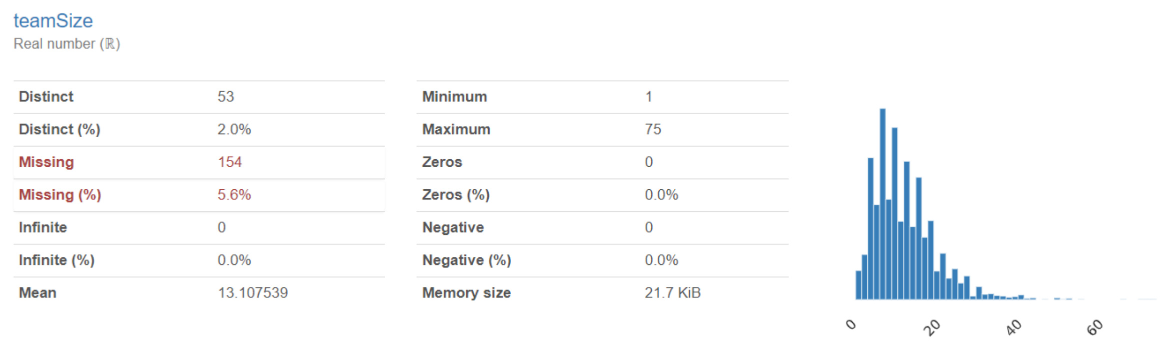
As can be seen the missing values are majorly present in the column countryRegion, priceUSD and teamSize. Only values in priceUSD and teamSize can be imputed.

Figure 2.C



The above figure helps us to further understand the distribution of missing values in the dataset.

Figure 2.D

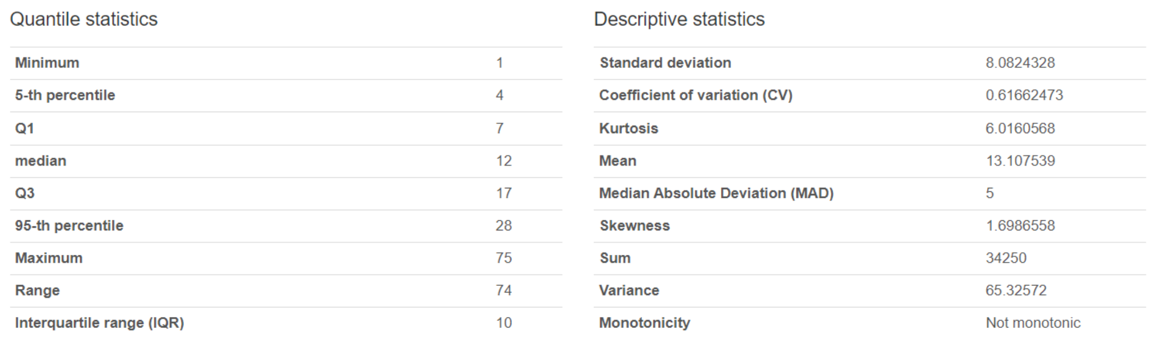


teamSize
Real number (ℝ)

Distinct	53	Minimum	1
Distinct (%)	2.0%	Maximum	75
Missing	154	Zeros	0
Missing (%)	5.6%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	13.107539	Memory size	21.7 KiB

In teamSize column there total 154 missing values and the data is right skewed.

Figure 2.E



Quantile statistics

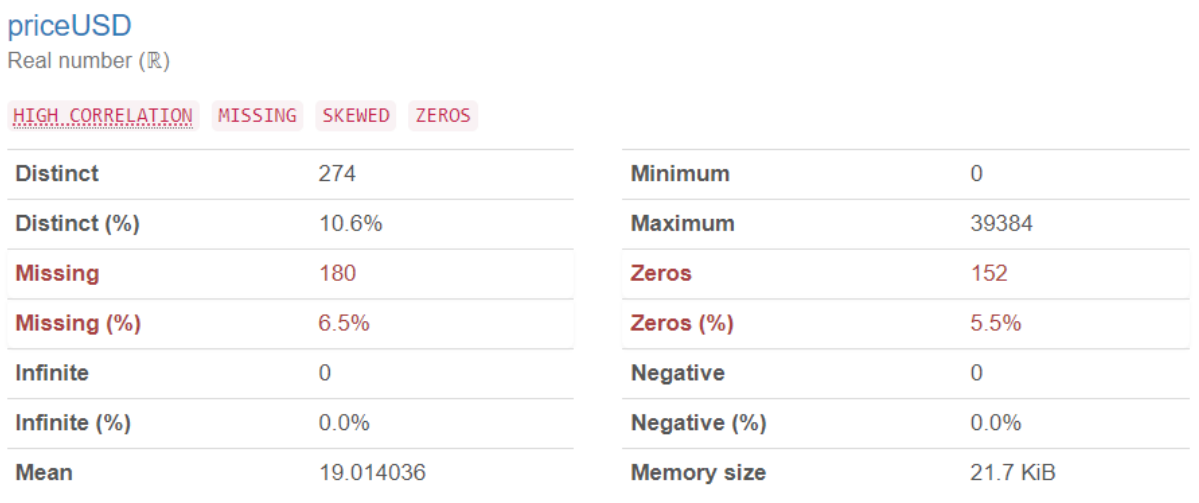
Minimum	1
5-th percentile	4
Q1	7
median	12
Q3	17
95-th percentile	28
Maximum	75
Range	74
Interquartile range (IQR)	10

Descriptive statistics

Standard deviation	8.0824328
Coefficient of variation (CV)	0.61662473
Kurtosis	6.0160568
Mean	13.107539
Median Absolute Deviation (MAD)	5
Skewness	1.6986558
Sum	34250
Variance	65.32572
Monotonicity	Not monotonic

The above figure contains some quantile and descriptive statistics about the teamSize column. The average team size appears to be 13 and the maximum team size is 75 with minimum team size being 1.

Figure 2.F



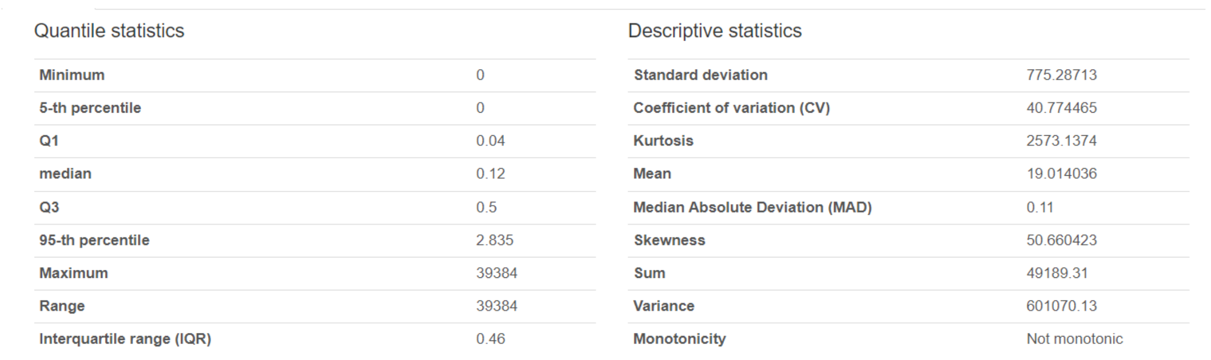
priceUSD
Real number (ℝ)

HIGH CORRELATION MISSING SKEWED ZEROS

Distinct	274	Minimum	0
Distinct (%)	10.6%	Maximum	39384
Missing	180	Zeros	152
Missing (%)	6.5%	Zeros (%)	5.5%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	19.014036	Memory size	21.7 KiB

A total of 180 values were missing in priceUSD column, and it has total 152 rows containing zeros.

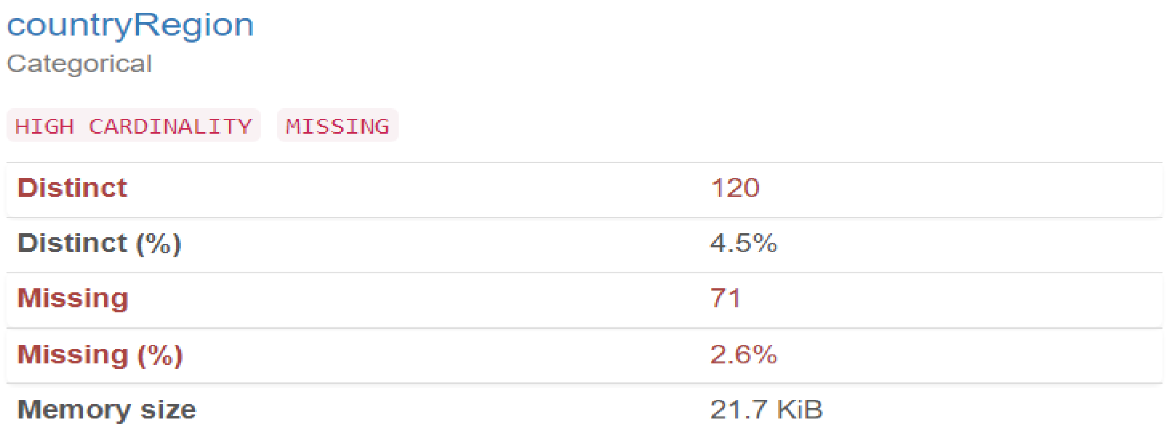
Figure 2.G



Quantile statistics		Descriptive statistics	
Minimum	0	Standard deviation	775.28713
5-th percentile	0	Coefficient of variation (CV)	40.774465
Q1	0.04	Kurtosis	2573.1374
median	0.12	Mean	19.014036
Q3	0.5	Median Absolute Deviation (MAD)	0.11
95-th percentile	2.835	Skewness	50.660423
Maximum	39384	Sum	49189.31
Range	39384	Variance	601070.13
Interquartile range (IQR)	0.46	Monotonicity	Not monotonic

The above figure gives some basic statistics about the priceUSD column, as can be seen the column is highly skewed with skewness being 50.66.

Figure 2.H



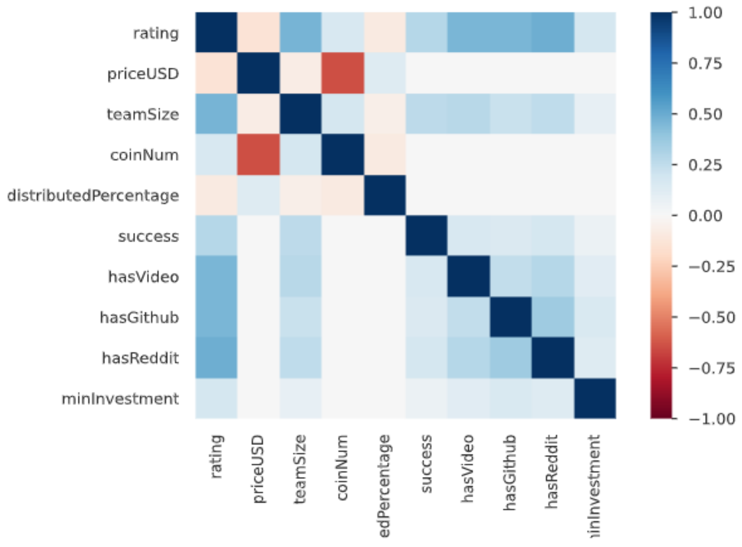
countryRegion
Categorical

HIGH CARDINALITY MISSING

Distinct	120
Distinct (%)	4.5%
Missing	71
Missing (%)	2.6%
Memory size	21.7 KiB

There are 71 missing values in Country region column. Apart from these following issues were found in the dataset.

Figure 2.I



The above heatmap shows the correlation between various columns of the dataset. The columns priceUSD and coinNum show a strong negative correlation. Rating shows a negative correlation with priceUSD but a positive corelation with teamSize and coinNum.

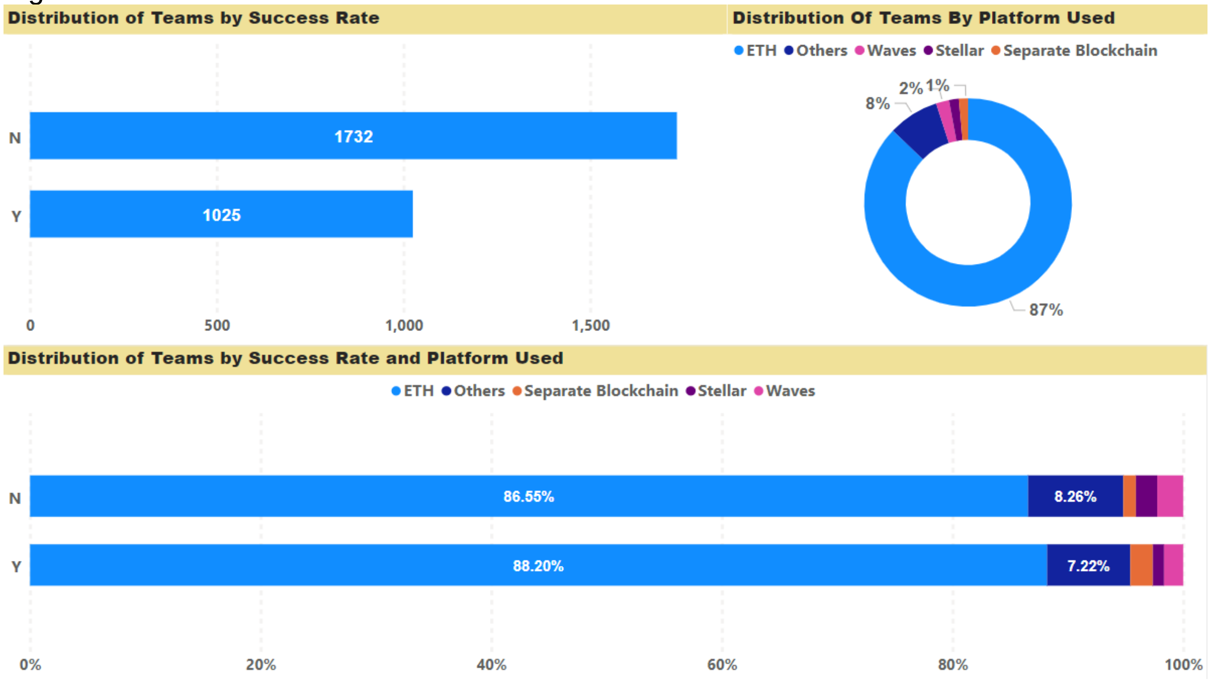
The following transformations were carried out the minimise the noise within the data: -

1) The ID column was dropped from the dataset.
2) The missing values in the numerical columns were imputed using mice technique.
3) To avoid loss of data the missing rows were labelled as the “Location Unknown” in the countryRegion column.
4) There were repetitive values found in columns countryRegion and platform. For example: the spelling of America can be found it two different formats “USA” and “usa” and spelling of Ethereum could be found as “ETH” and “Ethereum”.
5) Such repetitive values were replaced with the correct spellings in the dataset.
6) A new column called durationInWeeks was created using the startDate and endate columns, then the two date columns were dropped from the dataset.
7) Values which were greater than 1 in the distriutedPercentage column were removed.
8) Outliers in the priceUSD and teamSize columns were removed.
9) Values in the numerical columns were normalised using min-max normalisation.
10) Only the top seven countries were kept the rest were labelled as other to reduce the cardinality of countryRegion column.
11) Only the top four platforms were kept to reduce the cardinality of platform column.
12) Further one hot encoding was performed on the countryRegion and platform column since they carry categorical data.
13) Since the priceUSD columns are highly skewed square root and Cube Root Transformations were used to reduce the skewness within the data.

3.Exploratory Analysis

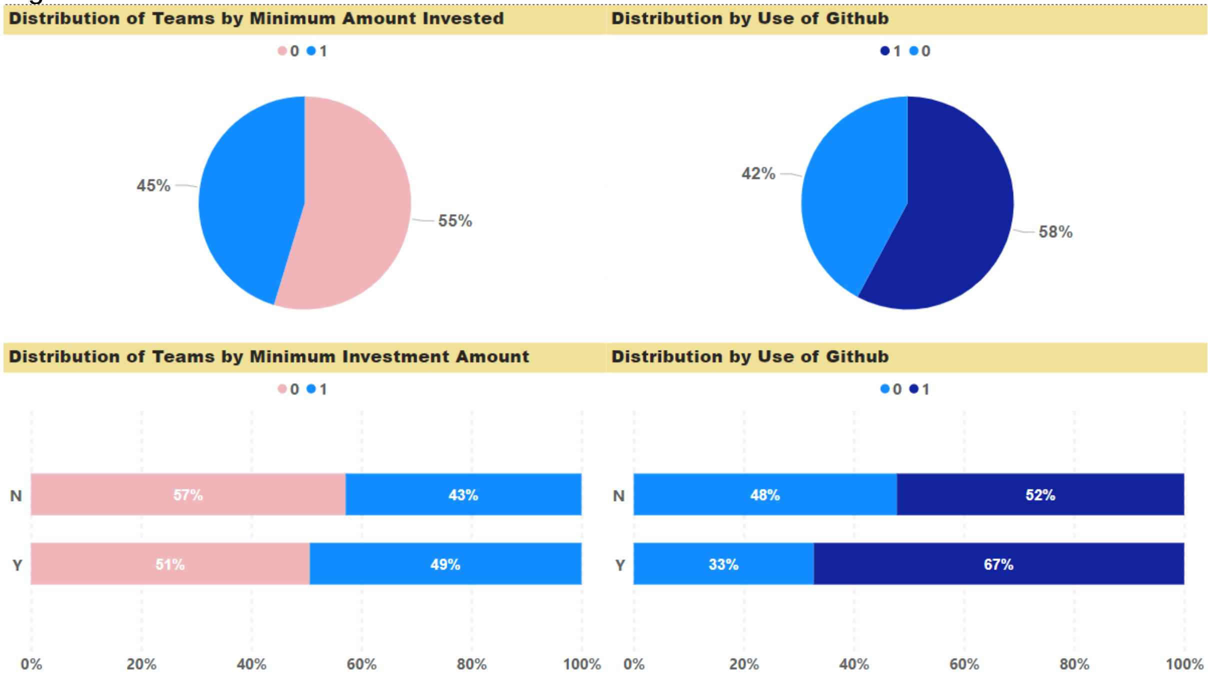
After applying the necessary transformations to the data some exploratory analysis was performed for further understand the nature of the dataset provided.

Figure 3.A



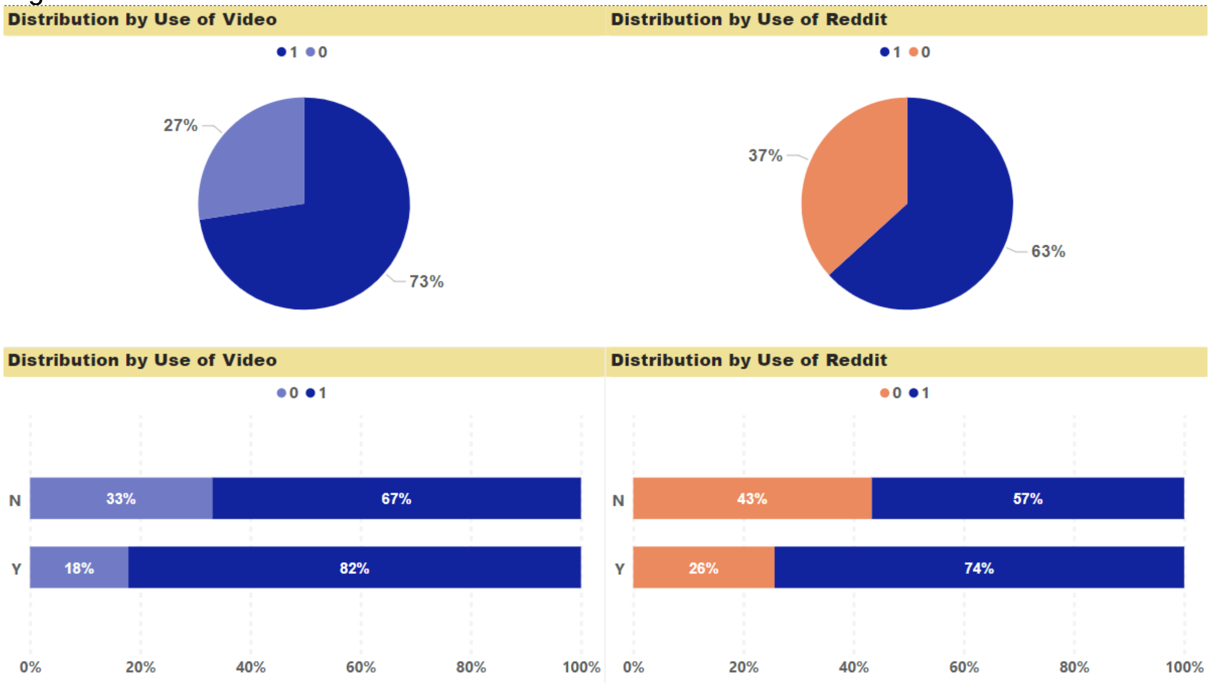
As can be seen from the above figure the data in imbalanced, especially with the platform variable as it shows that 87% of the instances are labelled as Ethereum.

Figure 3.B



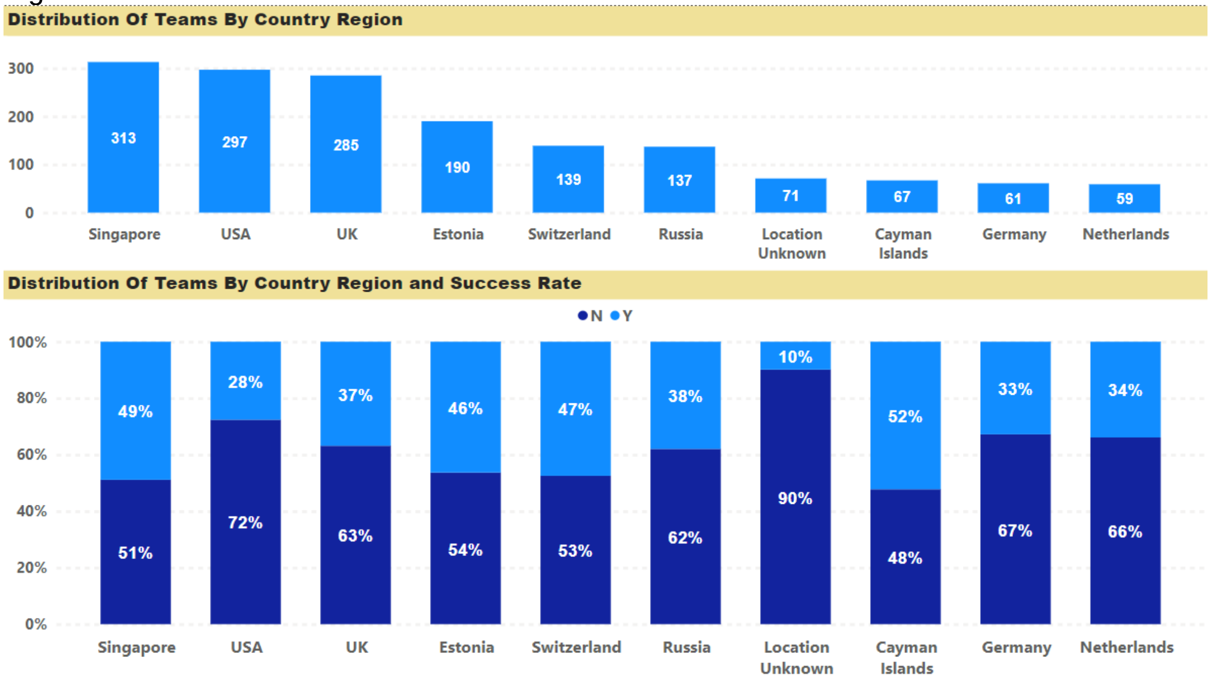
Majority of the teams did not require their investor to invest a minimum amount but on the other hand majority of teams have a github account. Out of all the teams who successfully hit their ICO goals most of them had a github account.

Figure 3.C



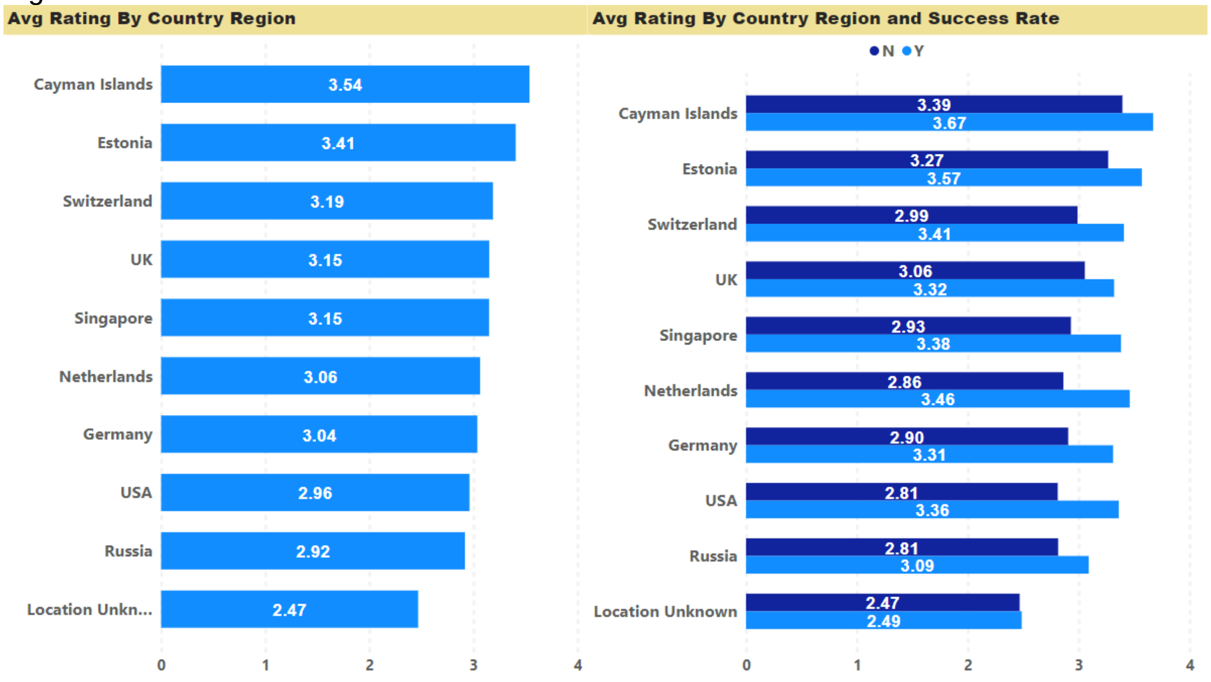
An overwhelming majority of the teams used videos on their landing page to educate the potential investors about their project and majority of them had a reddit presence as well. As can be seen from the previous two figures, having a github account, a video on the landing page and reddit presence does boost a team's chances of winning.

Figure 3.D



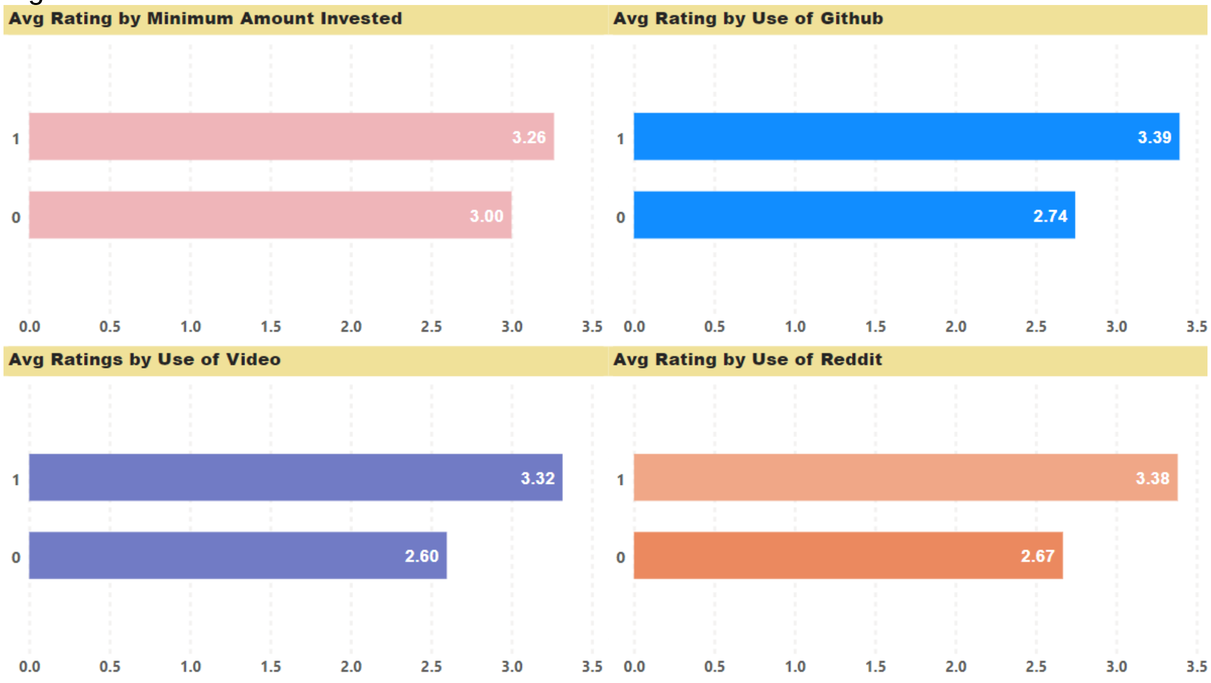
Only the top 10 countries are shown in the above figure. Teams from Cayman Island, Singapore and Switzerland have the highest success rate.

Figure 3.E



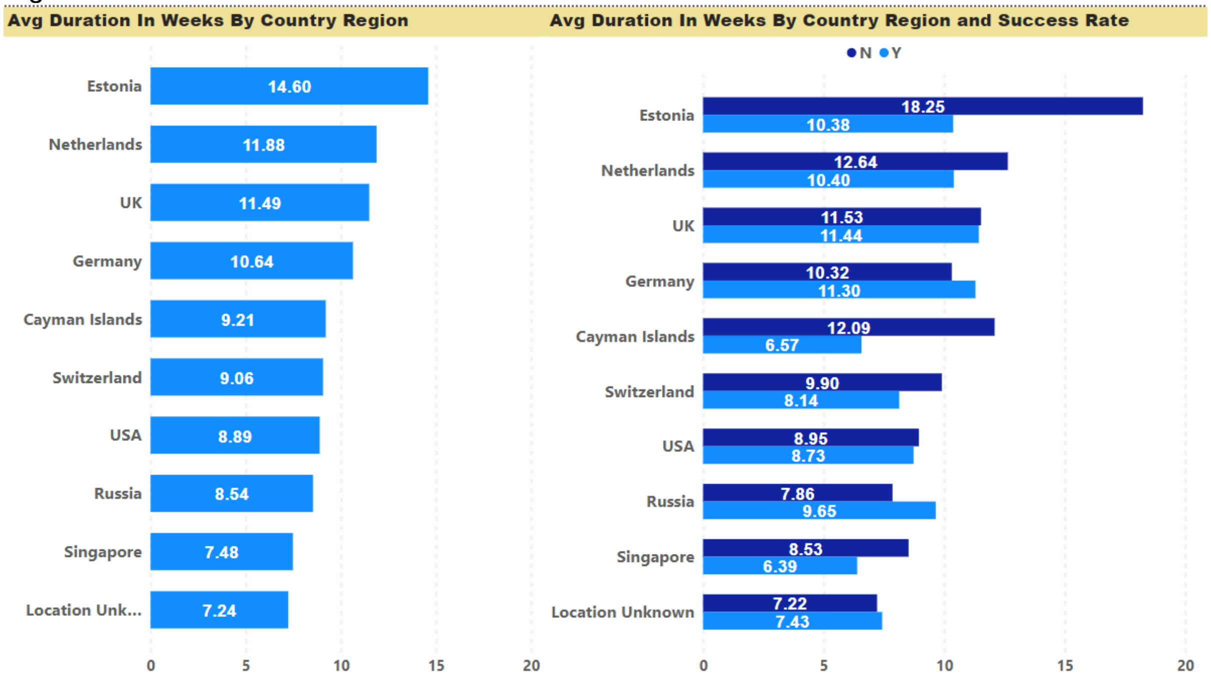
As expected, teams from Cayman Islands, Estonia have a higher average rating compared to rest of the countries. Teams which succeed show a higher average rating compared to teams which failed.

Figure 3.F



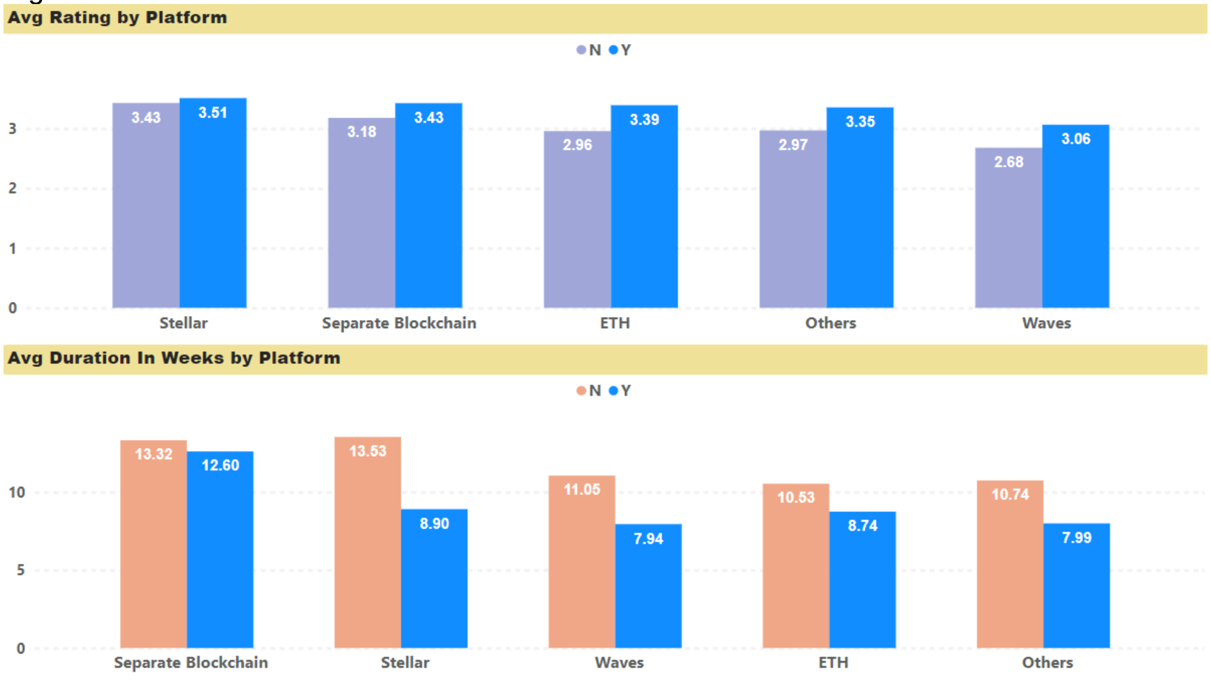
Average rating is higher for teams which have a social media presence and a good marketing.

Figure 3.G



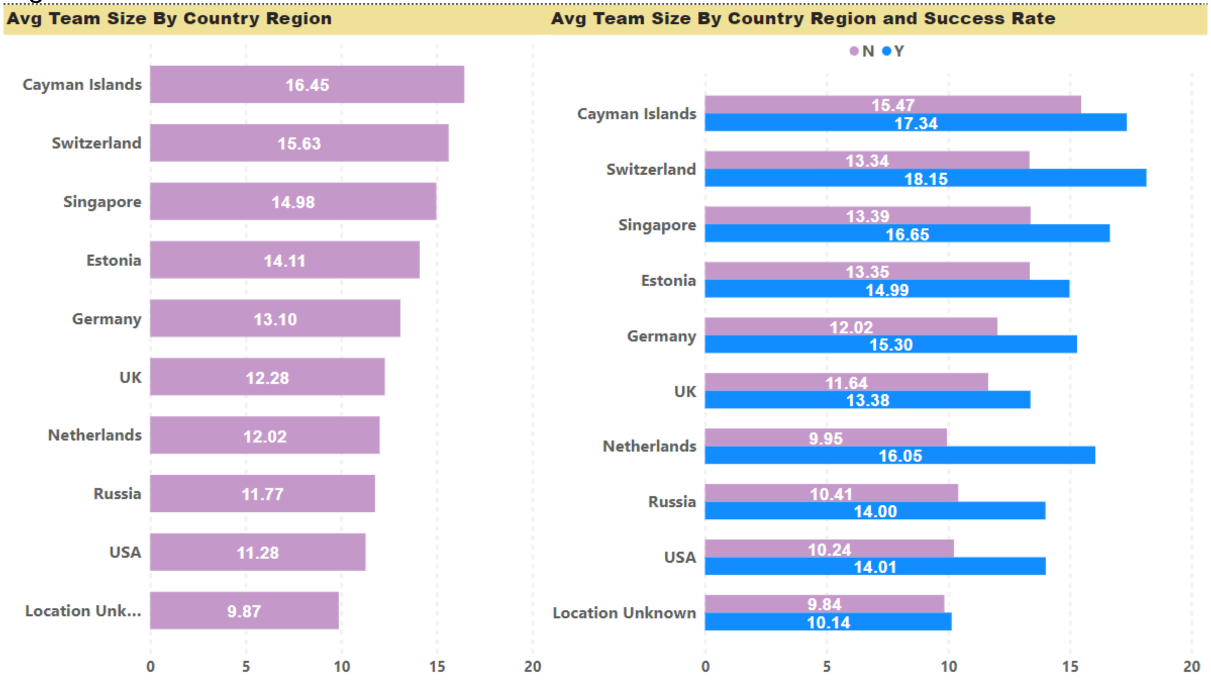
Teams which successfully achieved their fund-raising goal had shorter campaign duration compared to team which failed.

Figure 3.H



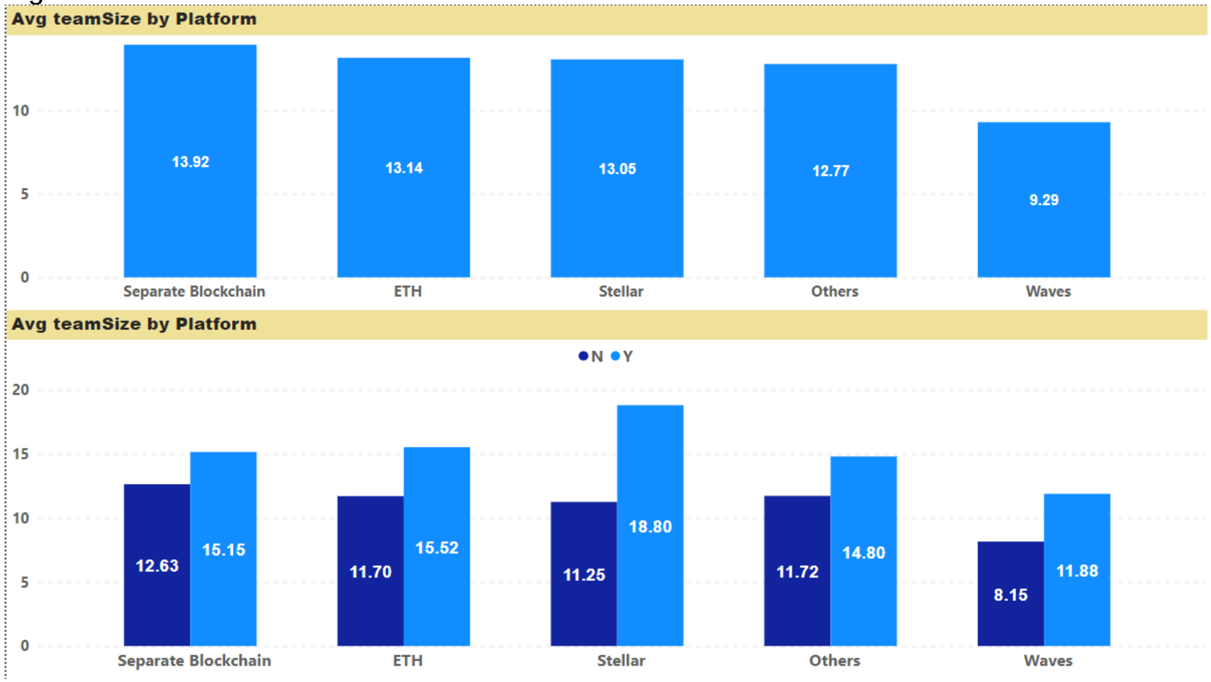
The average campaign duration is higher for teams who decided to use their own blockchain.

Figure 3.I



Average team size is higher in teams which successfully achieved their fundraising goals.

Figure 3.J



The average team size is higher for the teams which decided to build their own blockchain.

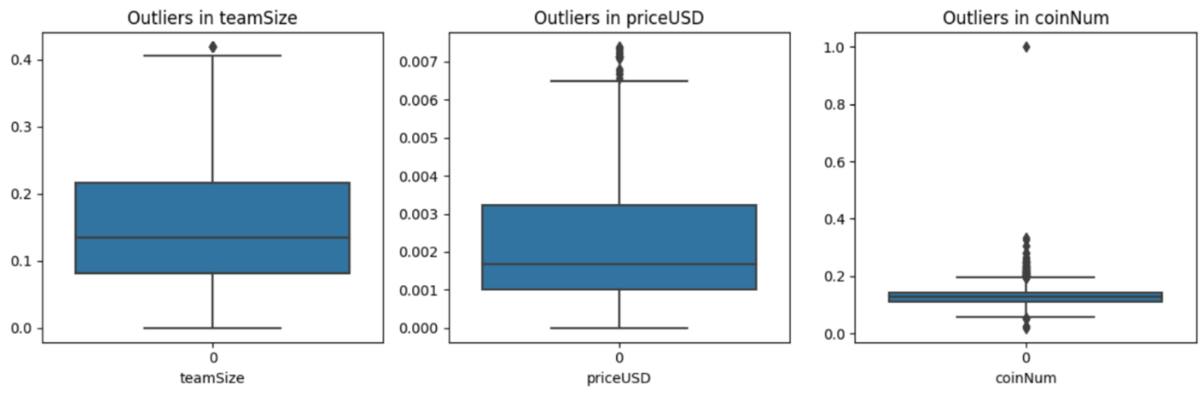
4.Modelling and Evaluation

Choosing the correct classification algorithm is one of the most critical parts of a machine learning pipeline. As per the analysis and data exploration conducted, we understood the following things about the provided data: -

1)The data is imbalanced 67% of the observations are labelled as 'NO'. The imbalance is not just present in target variable, but it is also heavily present in the platform variable of the dataset; over 80% of the records are labelled as Ethereum. Apart from these two columns other categorical variables like hasGithub, hasReddit, hasVideo, minInvestment show a relatively high degree of imbalance.

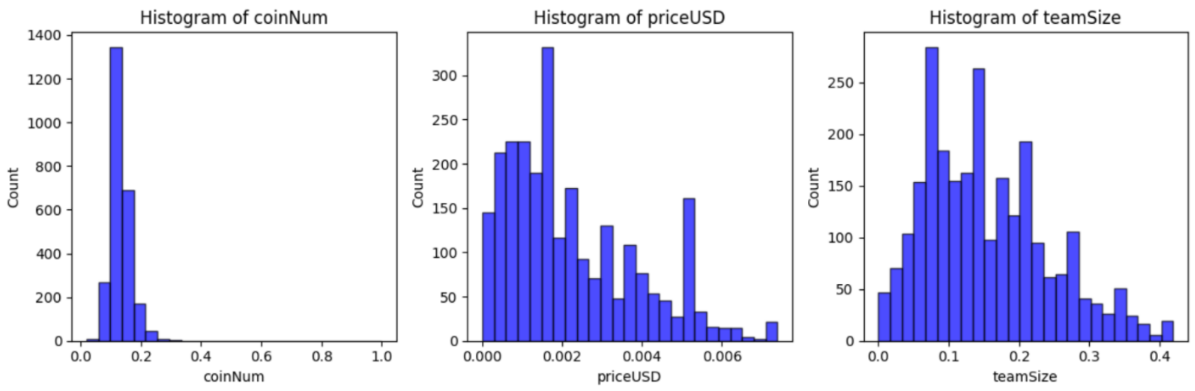
2) The numerical columns namely priceUSD, coinNum, and teamSize exhibit a high degree of skewness. Indicating that these columns contain too many extreme values.

Figure 4.A



Even though after taking necessary step to remove outliers and reduce skewness these extreme values cannot be eliminated completely as seen in the above figure.

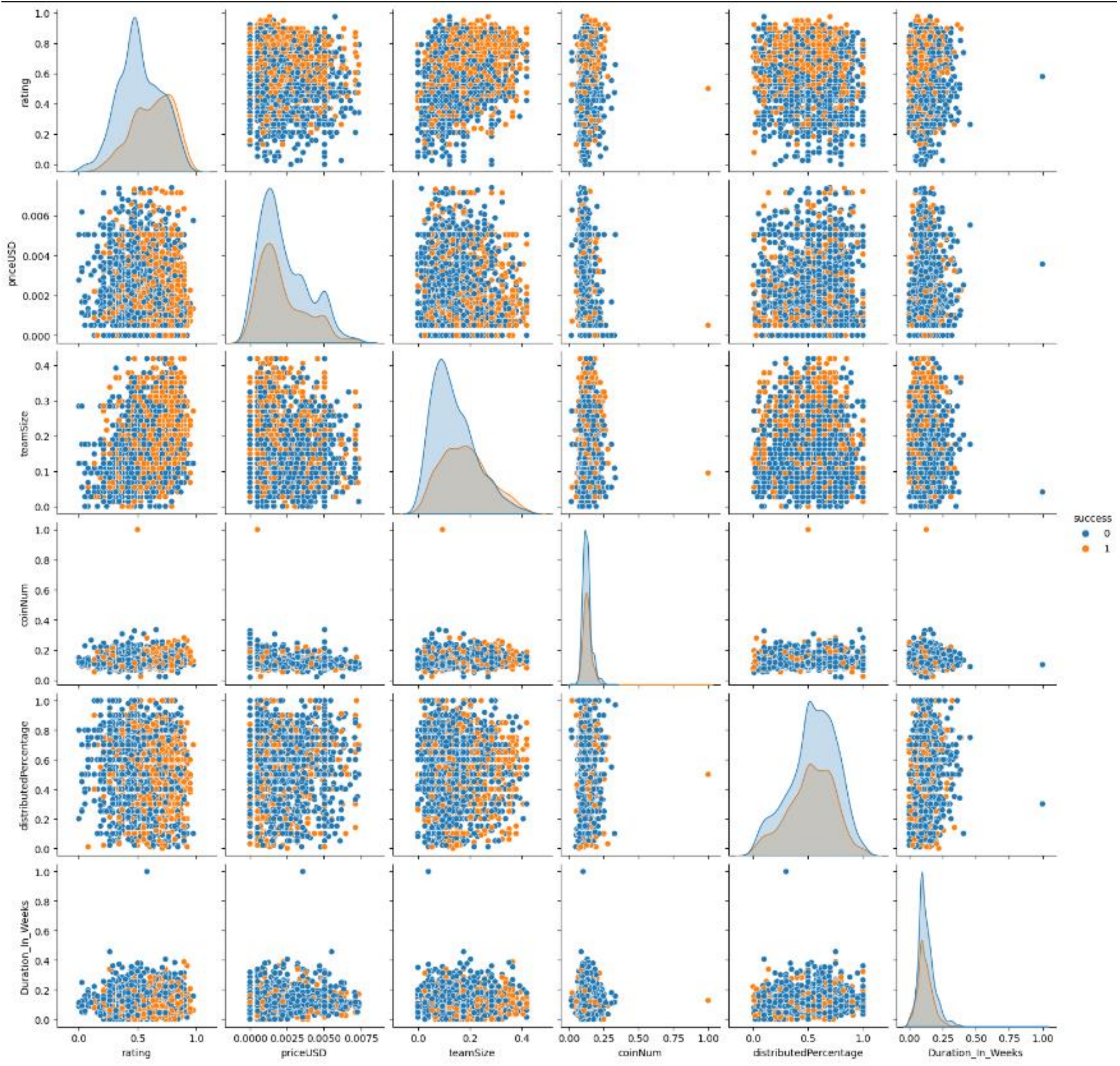
Figure 4.B



The above figure shows the histogram after applying square root and cube root transformations.

6) Some of the variables in the dataset exhibit a high degree of overlapping which increases the complexity of classification.

Figure 4.C



The blue dots indicate the teams which were unsuccessful in achieving their fund-raising goals.

Keeping all the above points in mind we need a model which will not be affected by the imbalanced data and can handle large number of features at the same time it should be less sensitive against extreme values while also be able to deal with overlapping between the datapoints. Hence the following classification algorithms were selected to achieve the objective of predicting the class labels: K-nearest neighbour, Support Vector Machine, Decision Tree Classifier, Random Forest, and Adaptive Boosting.

The following steps were carried out to evaluate the performances of the models. (Brownlee, 2020b)

1) Perform K-fold cross validation.
2) Identify the optimal value of K for k-folding.
3) Identify the best set of values for the hyperparameters of each classifier.
4) Compare the performance of the models on the data using train_test_split method.

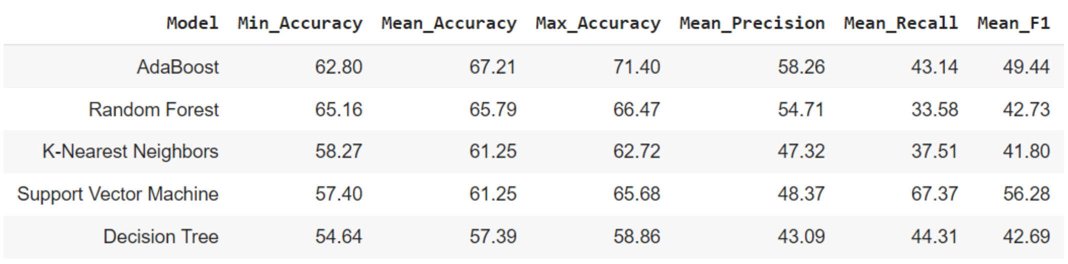
Step-1: Stratified K Fold Cross Validation

Unfortunately, k-fold cross-validation is ineffective for assessing imbalanced datasets. In k-fold cross validation data is divided into k-folds with a uniform probability distribution, but when the distribution is substantially skewed, it is probable that one or more folds may have few or no instances from the minority class, even though this may function perfectly for data with a balanced class distribution. This implies that many or maybe even most model assessments will be inaccurate since all that is required is for the model to properly forecast the majority class. (Brownlee, 2020a)

To overcome this challenge, we use stratified k-folding as it preserves the class imbalance between train and test data for each fold.

The models were deployed with default hyperparameters on the cleaned dataset to check how each of the classification algorithm would perform on the dataset.

Figure 4.D: Performance metrics after 5-fold cross validation



Model	Min_Accuracy	Mean_Accuracy	Max_Accuracy	Mean_Precision	Mean_Recall	Mean_F1
AdaBoost	62.80	67.21	71.40	58.26	43.14	49.44
Random Forest	65.16	65.79	66.47	54.71	33.58	42.73
K-Nearest Neighbors	58.27	61.25	62.72	47.32	37.51	41.80
Support Vector Machine	57.40	61.25	65.68	48.37	67.37	56.28
Decision Tree	54.64	57.39	58.86	43.09	44.31	42.69

The above figure shows mean performance scores for each classifier with 5 fold cross validation. Adaptive boosting gives the highest mean_accuracy of 67.05%, while decision tree algorithm gives the lowest mean_accuracy score of 57.39%.

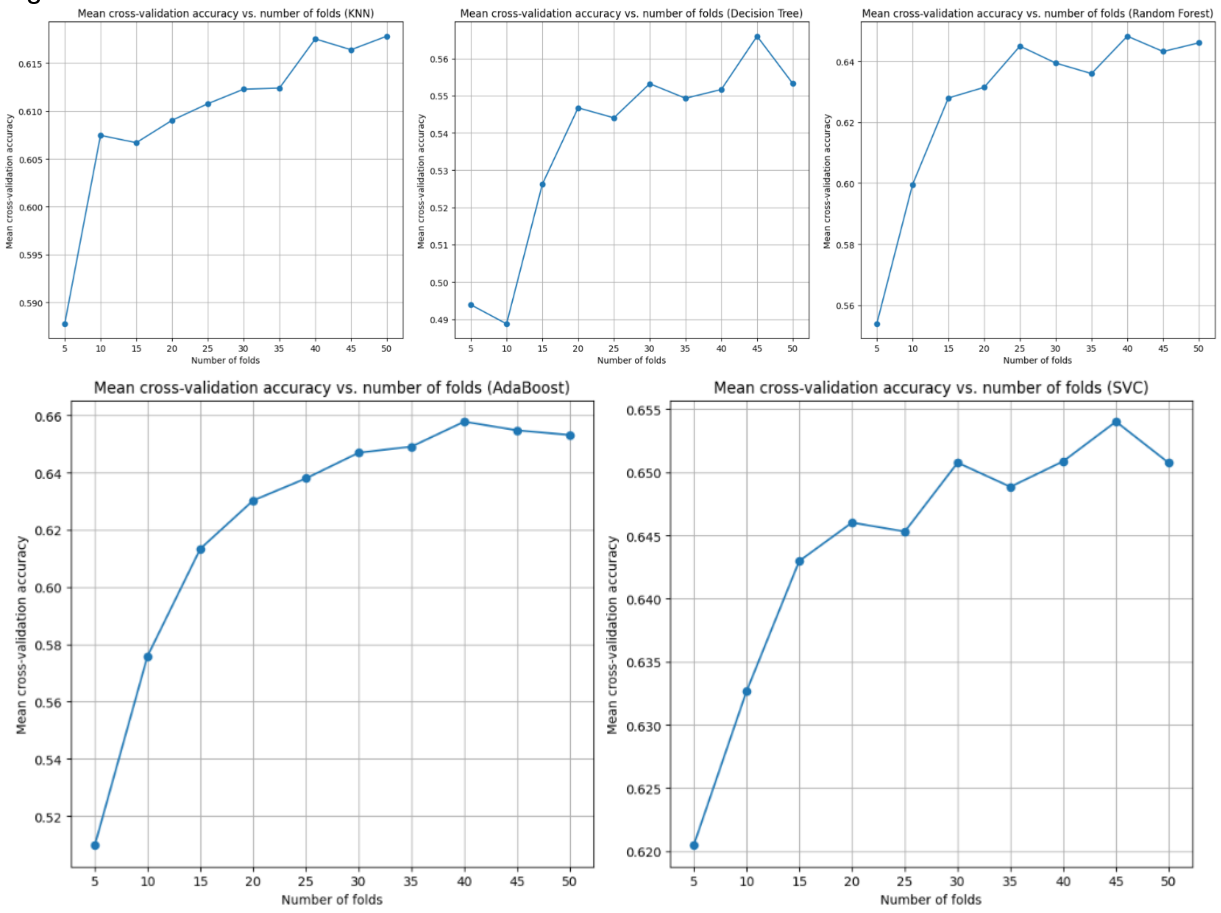
Adaptive boost also has the highest mean_precision score which means it is able to successfully classify majority class which is “NO” with high accuracy.

On the other hand Support vector has the highest mean_recall score but a relatively weak mean_precision score suggesting that it is successfully able to classify the the desired class in the data set with high accuracy.

Step 2: Identify The Optimal Value Of K for k-folding

In the previous step we performed 5-fold cross validation just to get an estimate of the model's performance on the provided dataset. To get an idea about the how the mean_accuracy_score changes as we increase the folds the following graphs were plotted.

Figure 4.E



The largest jump in the mean accuracy for each classifier was seen between 5 and 15 folds. Even though beyond 15 folds the accuracy scores keep increasing the actual accuracy gain keeps diminishing. Increasing the number of folds lead to serious issues like overfitting and unnecessary increase in the computational resources needed to complete the task. As the number of folds increases the models can become sensitive to noise in the data and may struggle to generalise the results on unseen dataset. Hence, for the dataset provided optimal value for the number of folds will be taken as 15.

Figure 4.F: Performance metrics after 15-fold cross validation

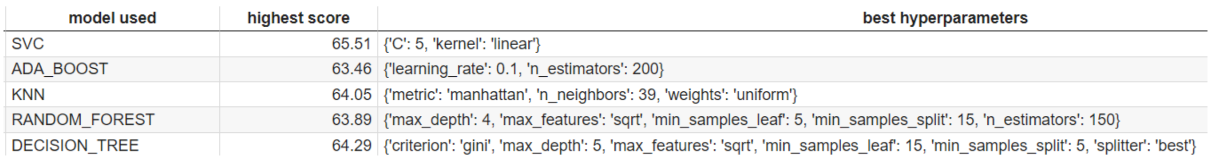
Model	Min_Accuracy	Mean_Accuracy	Max_Accuracy	Mean_Precision	Mean_Recall	Mean_F1
AdaBoost	60.36	66.61	75.74	56.94	42.08	48.21
Random Forest	60.36	64.92	72.19	56.32	34.86	42.73
K-Nearest Neighbors	56.21	61.76	65.68	48.05	37.51	42.01
Support Vector Machine	55.03	61.72	69.23	49.04	69.39	57.39
Decision Tree	50.30	57.27	61.54	42.38	43.78	43.60

Step 3: Identify The Best Set Of Values For The Hyperparameters Of Each Classifier

After getting a sense of how the classifiers were performing on the given data, the next logical step was to indetify the best set of values for hyperparameters of each classifier. Performing hyperparameter tuning has some clear advantages. First, it helps in optimising the performance metrics. Second, it improves the generalisability of the model and makes it better prepared for classifying data which it has not seen. Overall, it improves robustness and stability of the model (Brownlee, 2020b).

To indentify the best set of values for hyperparameters of each classifer a function GridSearchCV was used. K-fold cross-validation is a built-in feature of GridSearchCV while looking for the best hyperparameters. This is so that GridSearchCV can determine which set of hyperparameters works best with unobserved data. Cross-validation simulates this scenario numerous times and averages the outcomes to produce a more reliable indicator of the model's performance on unobserved data.

Figure 4.G: Best set of hyper-parameters



model used	highest score	best hyperparameters
SVC	65.51	{'C': 5, 'kernel': 'linear'}
ADA_BOOST	63.46	{'learning_rate': 0.1, 'n_estimators': 200}
KNN	64.05	{'metric': 'manhattan', 'n_neighbors': 39, 'weights': 'uniform'}
RANDOM_FOREST	63.89	{'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 5, 'min_samples_split': 15, 'n_estimators': 150}
DECISION_TREE	64.29	{'criterion': 'gini', 'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 15, 'min_samples_split': 5, 'splitter': 'best'}

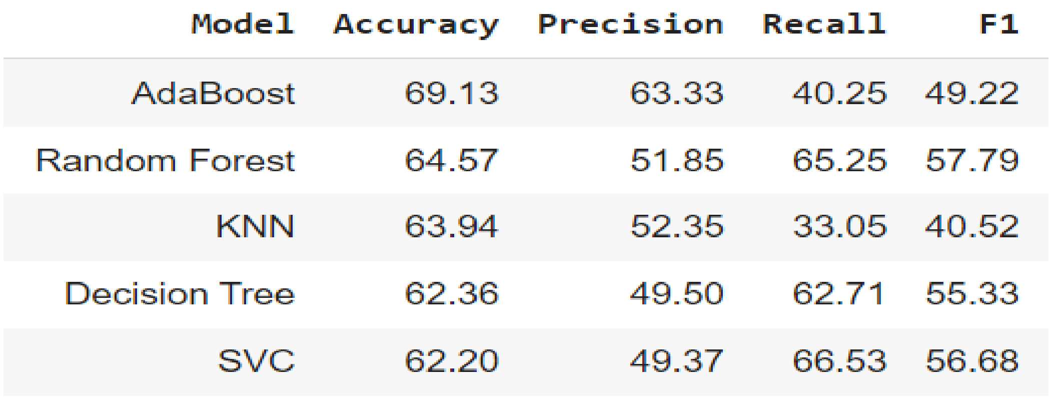
The above data frame shows the best set of values for hyperparameters of each classification model. SVC came on top with the highest score and the kernel which gave the best score with SVC was identified as 'Linear' with a penalty score of '5'.

Step 4: Comparing The Performance Of The Models On The Data Using Train Test Split Method

After identifying the optimum set of values for hyperparameters of each classifier next step was to test classifers on the data using a train_test_split method. Train test split method in python is generally used to get a quick estimate of the performance of the models on unseen data. Unlike k-folding this methods randomly splits the data. This is method is generally used to understand the classification performance on unseen data. Since in our case the we do not have access to new unseen data we will use the original dataset itself. The original dataset will be randomly split and the performances will be compared (Brownlee, 2020).

The figure below shows the scores of the model after using train_test_split.

Figure 4.H

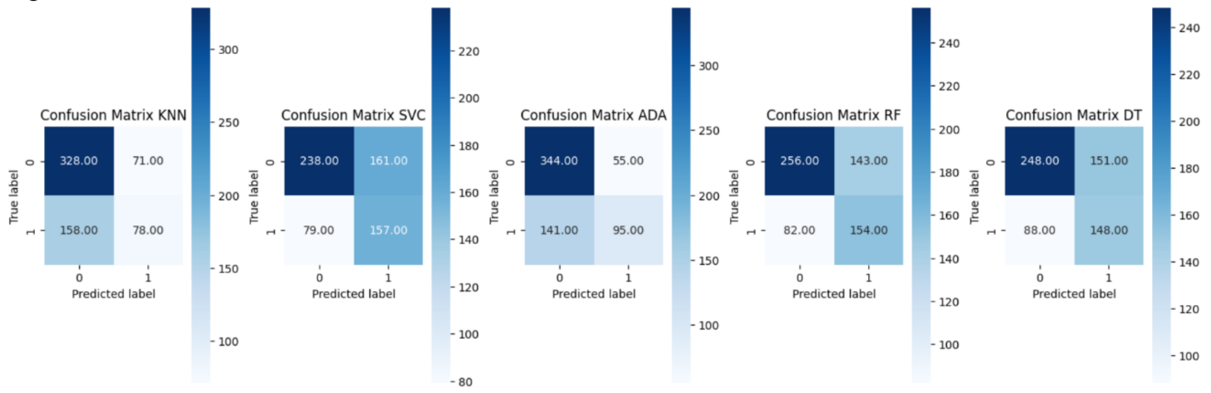


Model	Accuracy	Precision	Recall	F1
AdaBoost	69.13	63.33	40.25	49.22
Random Forest	64.57	51.85	65.25	57.79
KNN	63.94	52.35	33.05	40.52
Decision Tree	62.36	49.50	62.71	55.33
SVC	62.20	49.37	66.53	56.68

As it can be seen the above Adaptive boosting gives the best accuracy score even after hyperparameter tuning, however it has a low recall. On the other hand Random forest has shown considerable improvement in performance. It is the only algorithm which offers the lowest disparity between precision and recall score which is consequesntly reflected in the F1 score as it has the highest F1 score amongst all the classifiers.

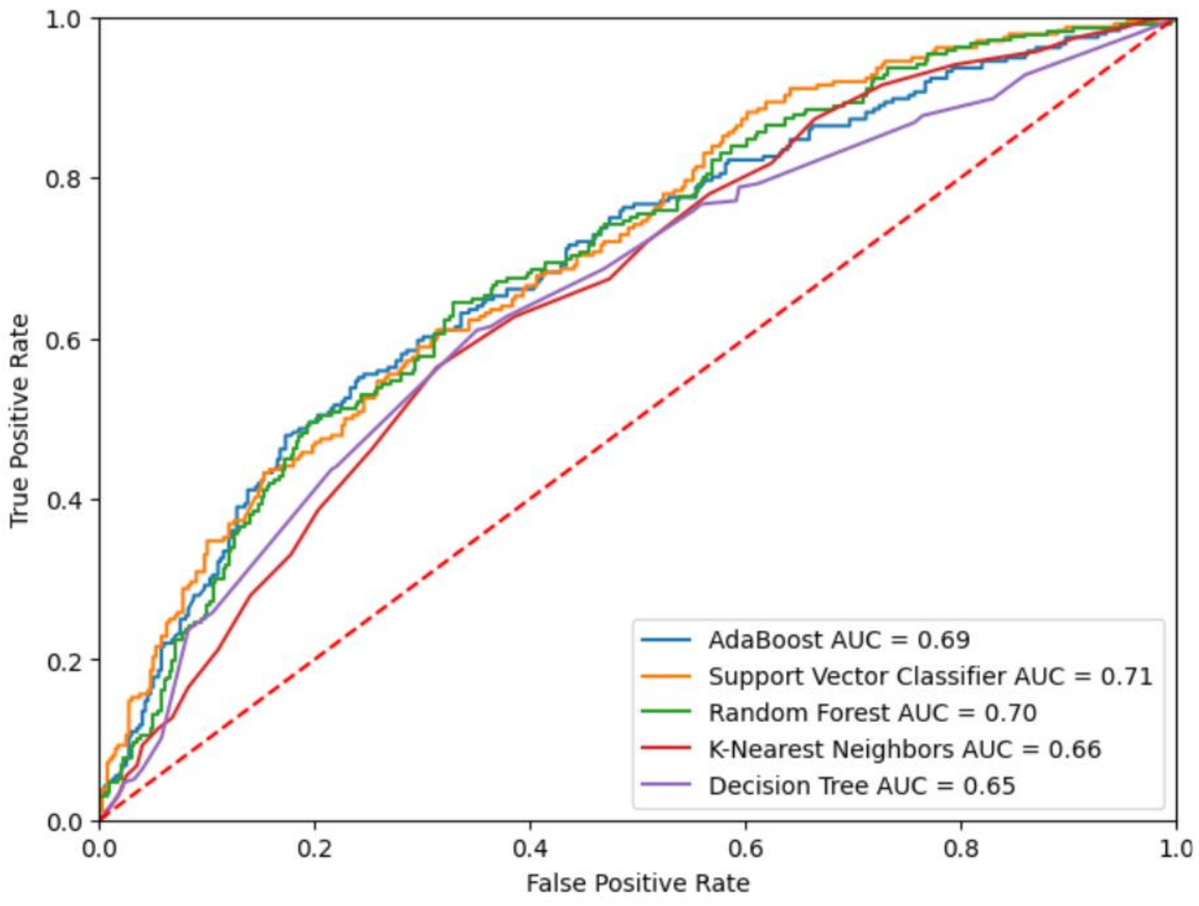
To further understand the differences in the performances of the classifiers let us look at the the confusion following confusion matrices and the roc curves of the classifiers.

Figure 4.I



Support Vector, Decision Tree and Random Forest classifer have the higher number of true positives but they also exhibit a higher number of false positives.

Figure 4.J



5.Conclusion

Before deciding on a single model, it is necessary to understand the implications of committing Type 1 (False Positive) and Type 2 (False Negative) error when it comes to predicting whether a team will be successful in achieving its fund-raising goals via its initial coin offering.

Implications of type 1 error:
If the rate of committing type 1 error is high it means that the precision of the model Is lower, it also implies that the team will fail in its objective of raising money but still it was classified as a positive class, this would lead to loss of investor confidence and erosion of trust as investors will be bleeding money if they decide to trust such model.

Implications of type 2 error:
If the rate of committing type 2 error is high it means that the recall of the model is low, and it implies that the team will succeed but the model classified as failure it failed to achieve its objective. Which results in a missed opportunity for investing.

Ultimately which model to choose is heavily dependent on the type of error we want to reduce. If we want to reduce the false negative, then Support Vector is a preferred model for the use case, out of all the models as it has the highest recall (66.53), auc score (0.71) and the second highest f1 score (56.68).

But, if the preferred choice is to reduce false positives, then the preferred classifier is Ada Boost as it has the highest precision (63.33) but a relatively low recall (40.25), when it comes to auc score (0.69) it is the third best performing model.

The only classifier which ensures both the errors are balanced is Random Forest, as it possesses the highest F1 score (57.79). Even though when it comes to auc score (0.7) it is at the second place, it is a small trade-off to minimise financials losses and missing out on missed investment opportunities.

Even though we have successfully identified which machine learning models are best suited for the task there are a lot of factors that contribute to the success of the ICOs which the dataset fails to capture such as; a well-thought-out and original idea shows that the project has the potential for growth, market acceptance, and long-term success, which makes it appealing to investors, people investing in ICO’s tend to place a great deal of importance on the team, which is leading the ICO, hence a competent and credible team inspires investor confidence, another factor that comes into picture is a well drafted white paper detailing the objectives, technicalities, and execution strategy (Reiff, 2022). To make the models more robust more data is needed.

6.References

1) Brownlee, J. (2020a). *How to Fix k-Fold Cross-Validation for Imbalanced Classification*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/cross-validation-for-imbalanced-classification/.

2) Brownlee, J. (2020b). *Step-By-Step Framework for Imbalanced Classification Projects*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/framework-for-imbalanced-classification-projects/.

3) Brownlee, J. (2020c). *Train-Test Split for Evaluating Machine Learning Algorithms*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/.

4) Reiff, N. (2022). *What Makes for a Successful ICO?* [online] Investopedia. Available at: https://www.investopedia.com/tech/what-makes-successful-ico/.