Leeds University
Business School

UNIVERSITY OF LEEDS

Assessed Coursework Coversheet

For use with *individual* assessed work

Student ID Number:	2	0	1	4	8	4	7	8	1
Module Code:	LUBS5990M								
Module Title:	Machine Learning in Practice								
Module Leader:	Dr. Xingjie Wei								
Declared Word Count:	2997								

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 THE SUCCESS OF INITIAL COIN OFFERING

Using Machine learning algorithms

1 Introduction

A cryptocurrency is a form of digital asset based on a network distributed across many computers. Due to the increasing price of Bitcoin and the advancement of block-chain technology, cryptocurrencies have pulled much attention in recent years. Most cryptocurrencies (for example, Ethereum) undergo a funding stage to raise money before public trading [1]. A new financing method called initial coin offering (ICO) has been widely applied to cryptocurrency projects based on innovative contract technology.

An Initial Coin Offering, generally referred to as an ICO or a token sale, is a fundraising method where tokens in a recently issued cryptocurrency are exchanged to the public for other cryptocurrencies such as Bitcoin or Ethereum [2]. The primary variation of ICOs is that they allow entrepreneurs to raise large sums of money in a short period with little effort while bypassing transaction costs [3]. Most ICO crowdfunding campaigns use the 'All-or-Nothing' model, where the ICO team introduces a fundraising goal. If they were able to raise money to the goal, the ICO is considered a success. Contrarily, if the fundraising is a failure, they hold nothing. Each ICO presents its project and team details on their campaign page.

In this project, we produce models to predict whether an ICO project will raise funding successfully. We investigate various factors affecting ICO success in the past, consequently predicting if a future ICO would succeed or not based on results. We first understand the data provided, then show relationships between variables. Next, we prepare our data by imputing missing values and cleaning data for modelling. We apply various models to our processed data, and finally, based on the performance, we will discuss the results.

2 Data Understanding

The data provided to us, have information for approximately 1600 ICOs. The success and failure of the ICOs have been captured in the data and coded as 1 and 0 respectively. Various other factors which might affect the success rate are also available. In figure 1 we can see distribution for some numeric values. Offered ownership is showed very skewed because of an outlier as 600. Others like social media, have three or four plausible values, which we can see here.

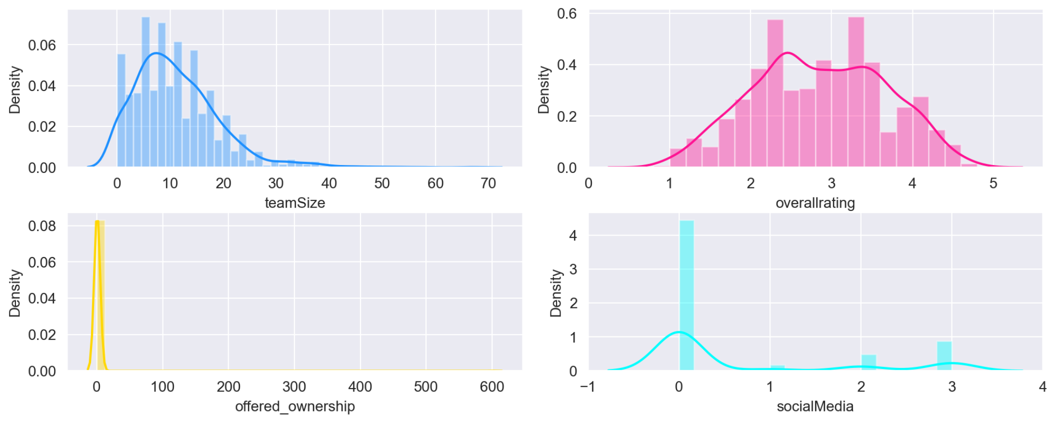


Figure 1: Distribution for some numerical columns

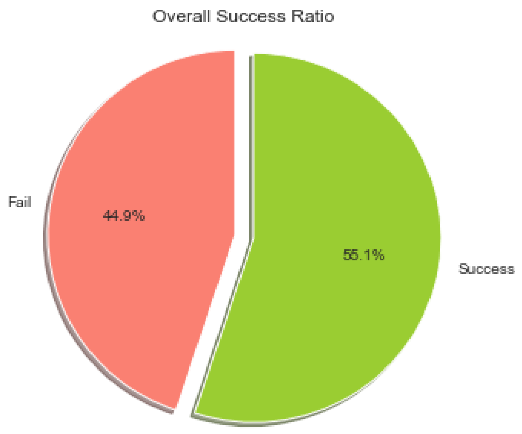


Figure 2: Overall success/failure of ICO ratio

We can observe here that there is not much difference between the overall success and failure for the ICO data from the figure 2.

We will explore some of the factors provided-

- Token Number - The number of tokens to be issued. ICO teams can freely decide the number of tokens to be issued. A higher token number offered usually means a lower price for each token.
- Success - If the ICO project raised funding successfully (achieved their funding goal) then 1, otherwise 0.
- Team Size - The size of team. better team, better quality of ICO product [4].

- Social Media - Popularity on Social Media such as the number of followers on Twitter — more publicity an ICO has, the better chance of it gaining funding [4].

- Overall Rating - Is the quality score provided based on overall performance of the ICO project.

- Offered Ownership - Refers to the percentage of the total tokens that the investors can buy.

- Country - In figure 3 we can see the top 10 country with highest ICO counts. USA has the highest count in number of ICOs followed closely by Russia and the UK. Figure 4, shows us the overall countries ratio with success. It

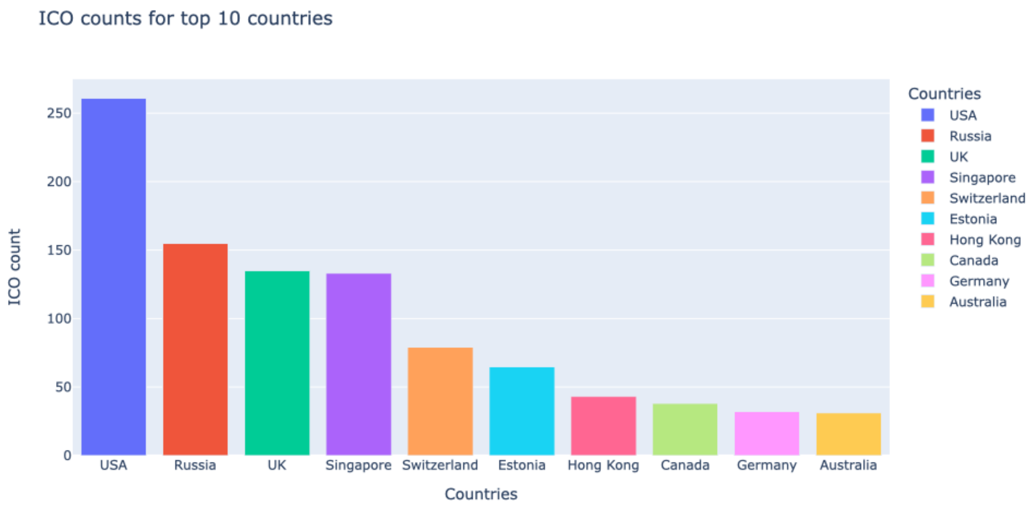


Figure 3: Top 10 countries with highest ICO count

showcases the top 5 countries with the most ICO projects and the success versus failure trend for those countries. Switzerland's success-to-failure ratio is sound, whereas even though there are more ICO projects in the USA and UK, the success-to-failure ratio for each country is very similar.

- Categories - Different areas to which the ICO project belongs has been incorporated under the Categories column.

- Duration - We had the start and end date of each ICO's under the fund raising campaign. We calculated the duration for each. For the rows with missing start date, we will not calculate duration, but let it be imputed.

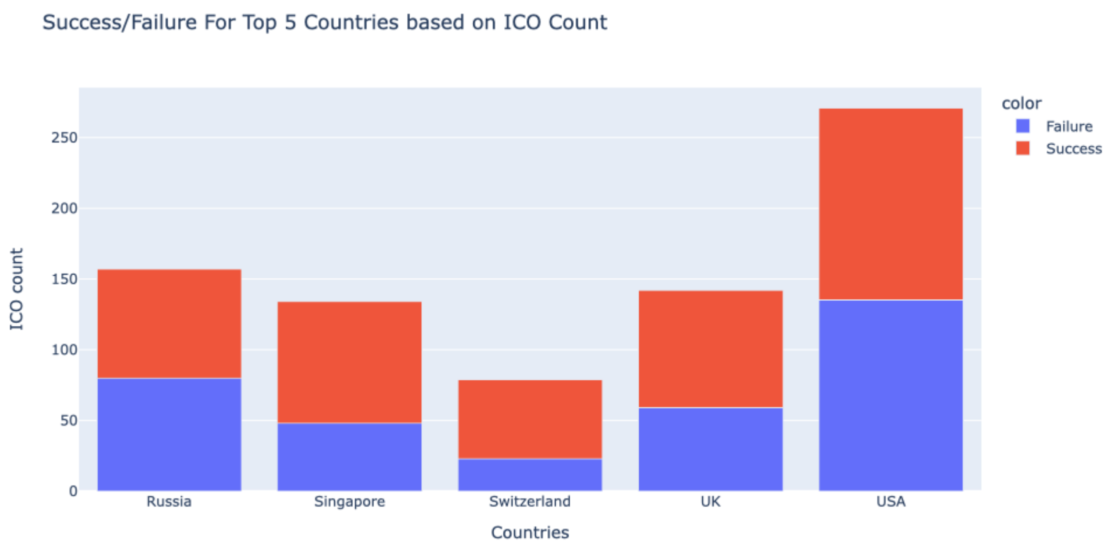


Figure 4: Success vs Failure For Top 5 Countries based on ICO Count

3 Data Preparation

We have data for approximately 1600 ICOs, but this data was inadequate and hence data preparation was required. Figure 5 shows a graphical representation of missing values in the dataset. As we can see, the columns success, teamsize, country, categories, overallrating and enddate have no null values. Before we proceed to impute the other columns, we need to clean our data.

3.1 Data cleaning

Country: The country column has many anomalies where for example USA is mentioned in different formats and so forth. We found the set of unique values and replaced with a single value. We found rows which had more than one country values, we created dummy rows for each country.

Start Date: The column startdate has some missing values, we calculated the average period and imputed the values for those missing 7 values.

Token Price: Before we normalise the price, we had to clean the price values. It is unstructured. For some tokens, the exchange rate is provided in Ethereum, while for some US dollars or other currency is used. So we split our data and transformed it into USD, as the majority of the exchange rated provided are in US dollars. First they are converted into USD-ETH equivalent. For example, $1ETH = 500$ SPIN is converted to $1SPIN = 1/500$ ETH.

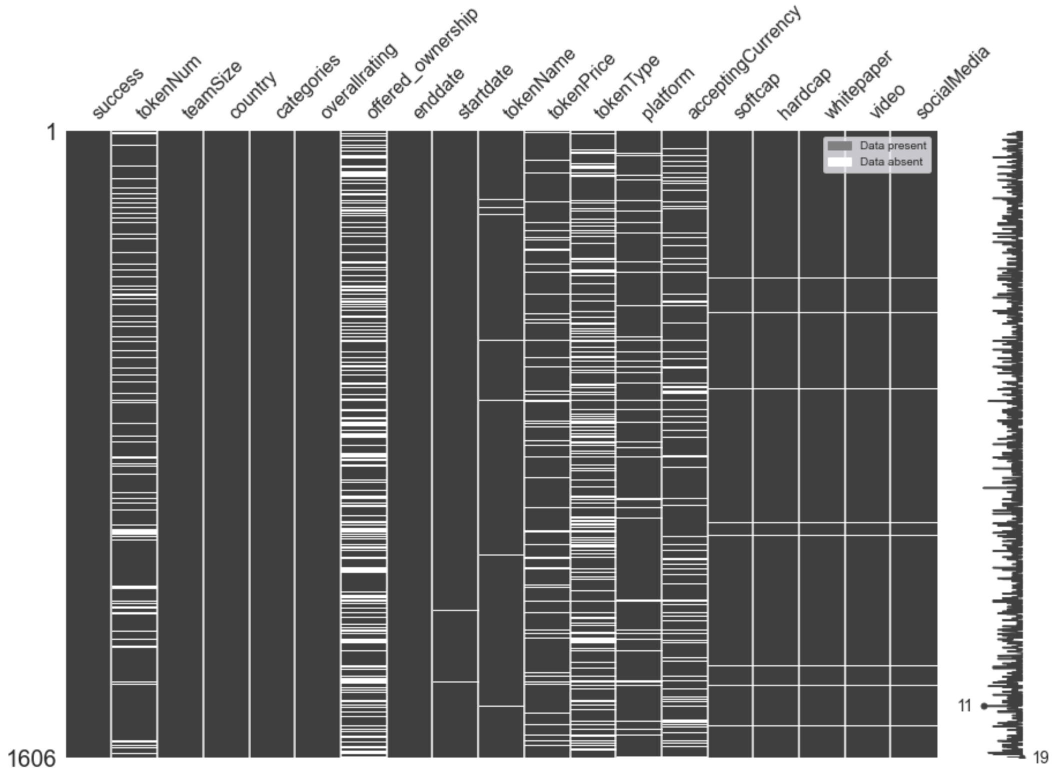


Figure 5: Missing values in the ICO dataset

Offered Ownership: 179 values with 0 tokenNumber but a positive offered-ownership fraction are present. All such 0 tokenNum values are removed which are imputed in the next stage.

3.2 Data Pre-Processing

Country: Once we have our cleaned data, we encoded the country values, since we need numerical data for analysis. We made a new column 'country ID' which had encoded country values.

Period: We have start and end date for each ICO, for which we calculated the absolute difference and assigned the period for each to a new column.

Token Price: Employing Yahoo- Finance historical data, the price value of each token is normalised into USD by converting the exchange value to USD based on the exchange rate at the start date of that particular ICO campaign. This new price is added as a new column, 'tokenPriceinUSD' [5].

Accepting currency, Platform and Categories: These columns have multiple values in certain rows. Before imputation and further analysis, these columns were converted to binary dummy variables. We transform each categorical value into a new categorical column and assign a binary value of 1 or 0 to those columns. These columns have multiple values in each row, so dummy encoding is required to represent various classes at once.

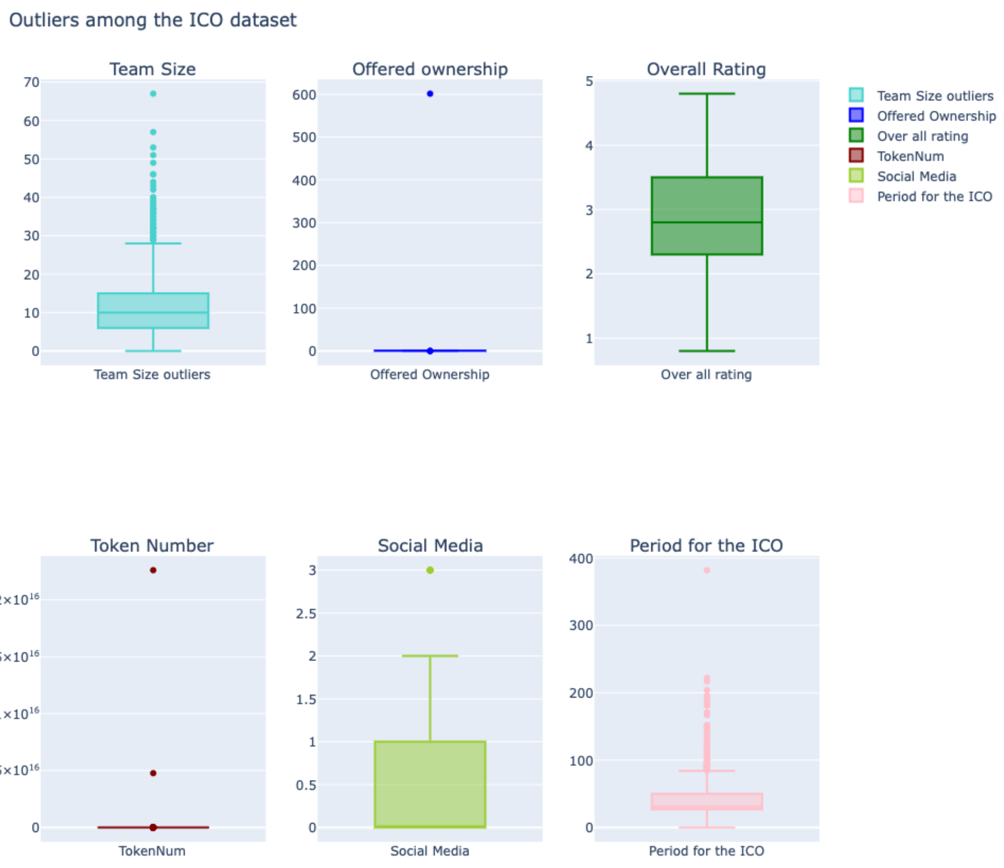


Figure 6: Outliers

Now, once we have cleaned our data, we work on removing the missing values. Figure 6 shows us the outliers in the ICO dataset. As we can see that offered-ownership has one outlier with a value of 600, this was replaced by a null value which was imputed. Similarly, outliers in the variable tokenNum were removed.

Following values were removed as shown in table 1 before moving to the imputation phase. Either these columns have been dummy-encoded to be used for imputation and modelling or they are not relevant in predicting ICO success.

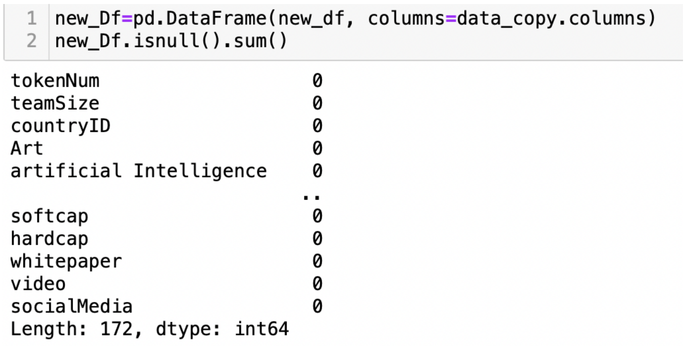
For imputation, **missForest** algorithm is used. It fits a random forest on the observed part and then predicts the missing part. This process of training and

Columns	**Reasoning**
ID	No significance in prediction.
Categories	Column values were dummy-encoded
Country	We have encoded the values
Start and end date	We have derived the period for each ICO project
Accepting currencies	Column values were dummy-encoded
Platform	Column values were dummy-encoded
Token Price	We have added a new normalised column
TokenName	This is a categorical value and doesn't add significance to ICO success prediction
Token Type	No significance for ICO prediction

Table 1: Dropping columns which are not significant for further processing

predicting repeats in an iterative process until a check criterion is met or a maximum number of user-specified iterations is reached [6]. We preferred this over KNN imputation because the results of KNN are also profoundly determined by the value of k, which must be determined on what is essentially a try-it-all approach. On the other hand, Random Forest is non-parametric, so there is no tuning required.[7]

Verification was performed after missforest imputation; no null values were identified. In figure 7 we can see no null values in new cleaned processed dataframe, new_Df.



```
new_Df=pd.DataFrame(new_df, columns=data_copy.columns)
new_Df.isnull().sum()
```

```
tokenNum                   0
teamSize                   0
countryID                  0
Art                        0
artificial Intelligence    0
                          ..
softcap                    0
hardcap                    0
whitepaper                 0
video                      0
socialMedia                0
Length: 172, dtype: int64
```

Figure 7: No null values in proccessed data

4 Modelling

The relationships between various predictors were studied and correlation analysis was performed. It depicts the predictors and output relationships. As shown in the below figure 8, there is no meaningful correlation between any of the numerical predictors. However, since any one of them could be a potential predictor for our model, none of them is dropped from the data.

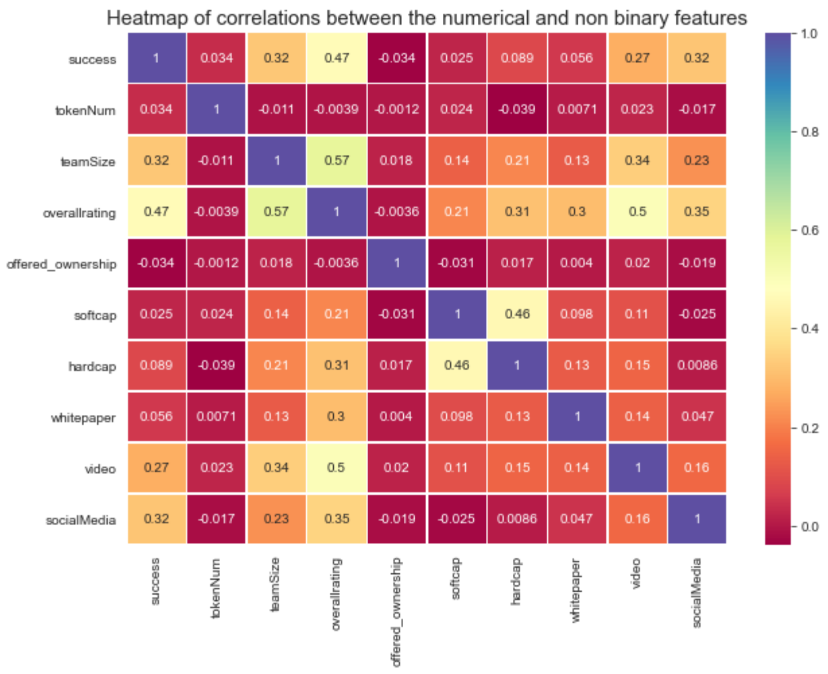


Figure 8: Correlation Heat Map

After cleaning and pre-processing, now we can split our dataset into training and test data sets for our models. We used **Stratified K-fold** method to split our data. This cross-validation object is a modification of K-Fold that returns stratified folds. The folds are made by preserving the percentage of samples for each class [8].

After splitting our data using stratified K-fold we will perform re-sampling on our data using random oversampling.[9] **Random Oversampling** involves selecting random examples from the minority class (in our case, it would be 'failure') with replacement and supplementing the training data with multiple copies of this instance.

Since we have supervised data-set, the data contains the target labels. We will go for supervised learning models. We have applied Random Forest Model (Decision Trees), KNN, Support Vector Classification (SVC), MLP (Neural Networks) , and Logistic regression models.

4.1 Random Forest Classification Model

Random Forest Classification is a supervised learning algorithm that uses ensemble learning method for classification. The ensemble learning method is a technique that links predictions from multiple machine learning algorithms to make a more precise prediction than a single model. Since this is tree-based method, the nodes are not affected by outliers. Thus, we do not scale the data and directly implement the algorithm on the processed clean data.

Random forest will stabilize following some $n - estimators$(because there is no technique to "slow down" the fitting, like boosting).For value as 100, accuracy obtained was 78% , for 200 - 79%. So we took $n - Estimators$ as 300 after vetting out different possibilities of parameter. For our data, we got an accuracy of **80.12%** on the random forest classification model.

Another important benefit of using Random Forest model is Feature importance. When we have a lot of variables in the data and need to reduce them, the relative importance of these variables can be found by using the importance function in the Random forest model.

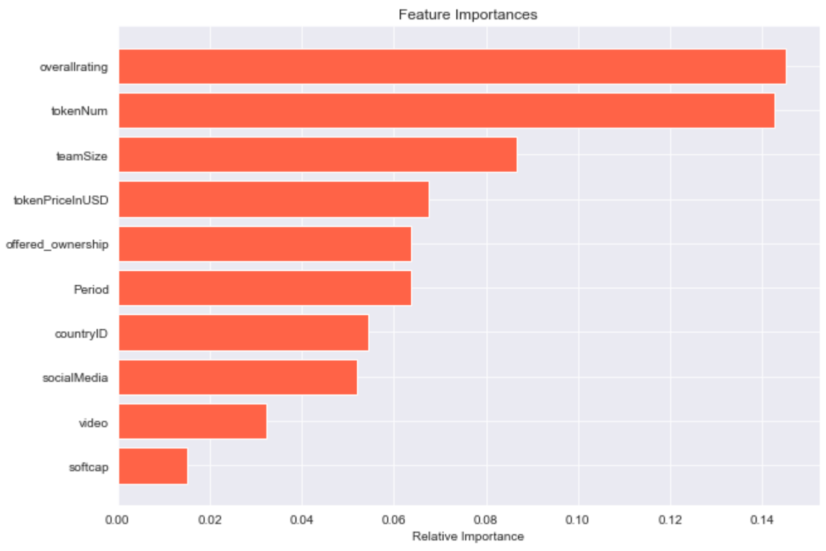


Figure 9: Important features predicted by the Random forest model

The above figure 9 shows the important features of the dataset as predicted by the model. Overallrating and tokenNum are the two most important features, followed by teamsize and tokenprice.

4.2 Support Vector Classification

An SVC (Support Vector Classifier) model's objective is to fit the data we provide, returning a "best fit" hyperplane or kernel (linear, radial or polynomial) that divides or categorises the data. After obtaining the hyperplane, test data is passed to the classifier to see what the "predicted" class is. The train and test data was scaled and factorised before passing it to the model, using the StandardScaler function. The 'RBF', or Radial Basis Function (RBF), kernel was used for better performance of the model. The accuracy achieved after running Linear SVC was **76.08%** . Different hyper-parameters of kernel values were used; for the linear kernel, 72% accuracy was obtained and for polynomial kernel, 69%.

4.3 Multi-layer Perceptron

MLP models are trained on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs. Training includes modifying the parameters, or the weights and biases, of the model in order to reduce error. Then, backpropagation is used to make those weights and bias adjustments comparable to the error. The error itself can be estimated in various ways, including by root mean squared error (RMSE) [10].

There are different hyper-parameters for MLP, including defining the hidden layers. A range of combinations of hidden node layers were evaluated; the optimal model was found to contain four hidden layers with 254, 128, 64, and 32 nodes respectively.An accuracy of **71.11%** was obtained.

4.4 K-Nearest Neighbours

KNN, or K-Nearest Neighbours, is a distance-based classification method. It is a type of lazy learning as it does not strive to construct a prevailing internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the k nearest neighbours of each instance.

To apply the KNN model, we need to find the best value of K, the number of instances that are taken into account for determination of affinity with classes. The value which gives the lowest error rate is the ideal value of K. To find the minimum error rate and the corresponding K value, below in figure 10 we can see the error rate computed against K value. Here the minimum error rate was 0.28 at k=27. So, the optimal K value is 27. An accuracy of **69.25%** was obtained for the KNN model.

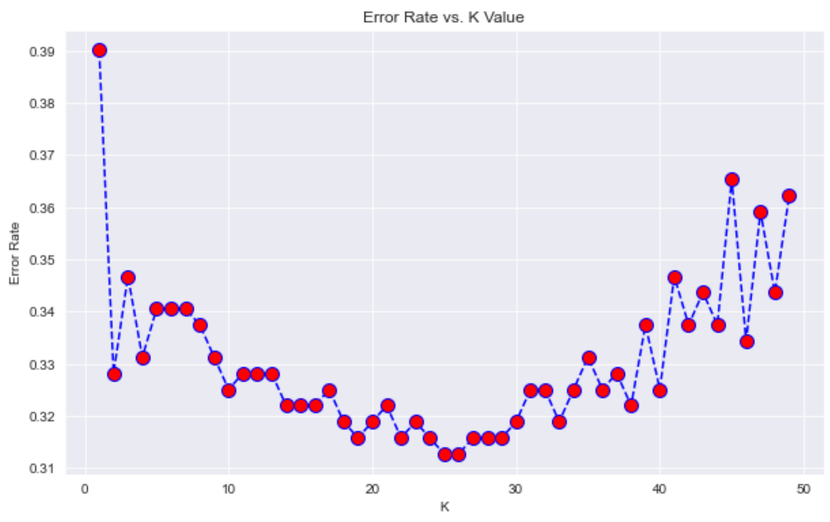


Figure 10: K- value calculation

4.5 Logistic Regression

Logistic regression is a classification algorithm. It is a basic structured classifier. In theory, computation should be faster for this model, compared to other classifiers. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function. The model was trained with processed data. The accuracy for the logistic regression model is **75.77%**. It is observed that the accuracy for the logistic regression model is comparable to that of the highest performing model, the Random forest classifier. This demonstrates our understanding about the computational efficiency of the model.

5 Evaluation

In table 2, we can see the overall accuracy of each model on our processed data. We can see that Random Forest gives the best accuracy, followed by SVC and Logistic Regression. MLP and KNN have less classification accuracy.

Model	**Accuracy of the models**	**F1 score**
Random Forest classification	80.12%	0.81
SVC	76.08%	0.76
Logistic regression	75.77%	0.76
MLP	71.11%	0.75
KNN	69.25%	0.70

Table 2: Accuracy and F1 for the different classified models applied

5.1 F1 score

The F1 Score is given by $2 * ((precision * recall)/(precision + recall))$. It is also called the F Score or the F Measure. Put another way, the F1 score conveys the balance between the precision and the recall [11]. The table 2 shows the F1 score for each model. A good F1 score indicates low false positive and false negative rates, so real errors are correctly identified. An F1 score is of 1 is considered perfect. The Random Forest model has the highest F1 score, followed by SVC and logistic regression.

5.2 ROC and AUC curve

In order to visualise classifier performance, ROC (Receiver operating characteristics) and AUC(Area Under Curve) plots are used. In plain language, the ROC-AUC curve measures the performance at various threshold levels for classification algorithms.

ROC is a probability curve, and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. A higher AUC indicates a higher true negative rate and true positive rate. Sensitivity and Specificity are inversely proportional to each other. So when we increase Sensitivity, Specificity decreases, and vice versa. In the figure 11 we can see the curve for each model. Random Forest has performed better than the other four models, followed closely by SVC and logistic regression.

5.3 Confusion matrix

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and split down by each class. This is the core of the confusion matrix [12]. Figure 12 shows the confusion matrix plotted for each model. In the confusion matrix for Random Forest, there are 114 true positives, which means 114 correctly predicted successes. There are 114 true negatives, or 114 correctly predicted failures. False positives are those classes that are actually failures, but were predicted as successes by the model. There are 31 false positives. Similarly, there are 33 false negatives.

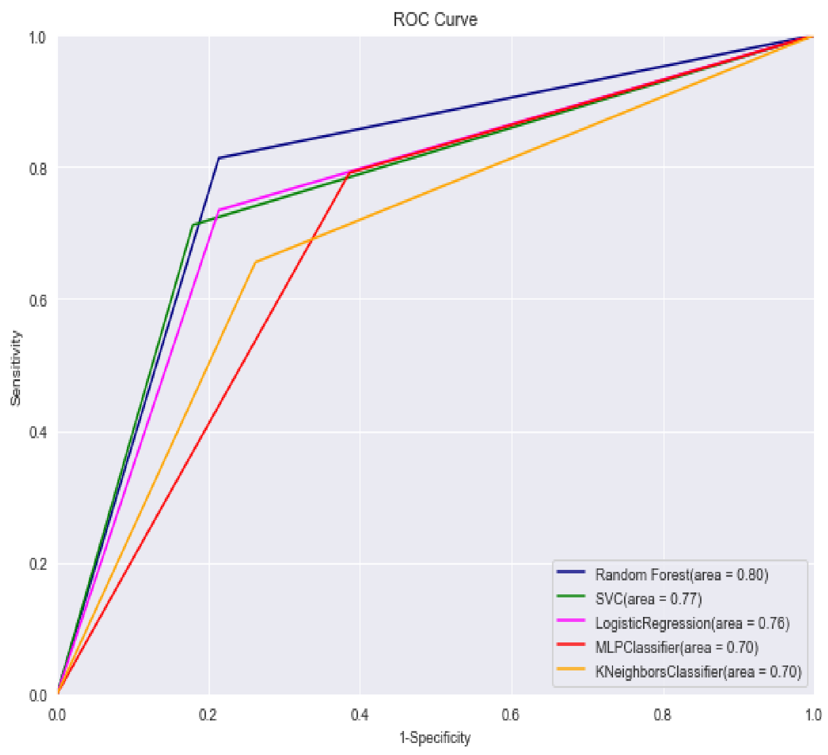


Figure 11: ROC AUC curve

6 Deployment

Results show that of all the classifier models, the random forest classifier has performed the best. This was verified by the different evaluation metrics applied. The random forest model best predicted the true negatives and true positives. The highest false negative rate was obtained with the KNN classifier. The MLP model obtained the highest false positive rate, as seen in the confusion matrix plotted in figure 12.

By ranking features in order of importance as predicted by random forest, in figure 9, we see which variables play an important role in predicting ICO success. Overall rating plays the most important role; it was positively correlated to 'success'. The base country of the project, team size and social media presence also play a vital role. The duration of each ICO campaign is also an important predictor.

6.1 Future work

A model is only as good as the input data. The accuracy and prediction performance of models could be increased by providing more relevant data, which

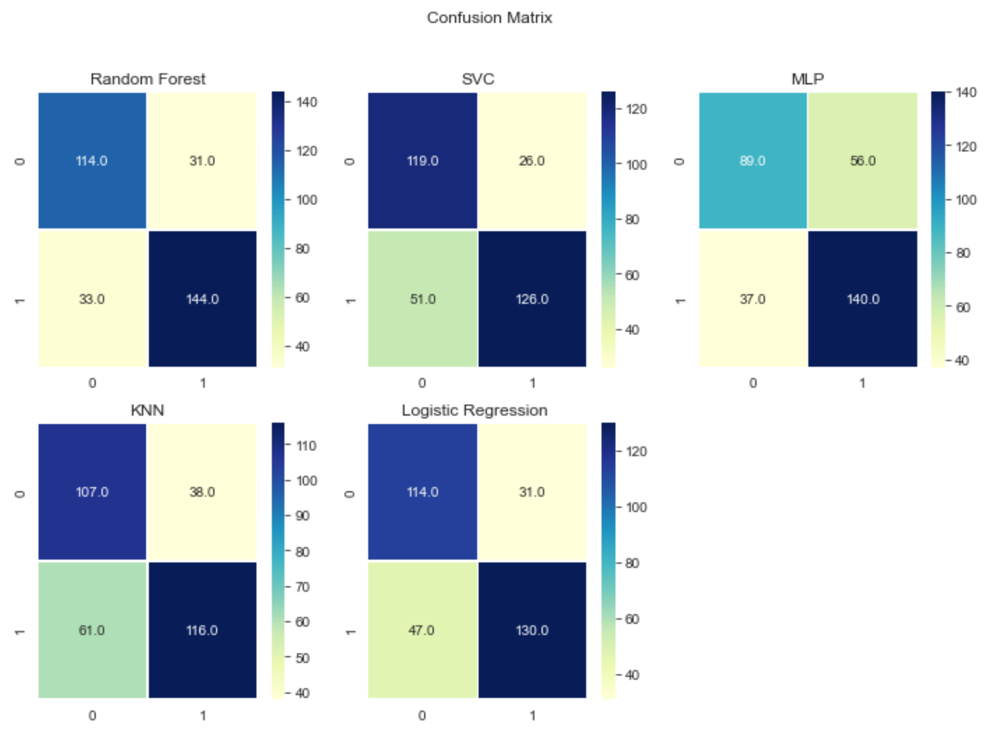


Figure 12: Confusion matrix heat Map

would have high correlation with success or failure of ICO. For example, data on the frequency of transactions per day might be a useful predictor variable. Adding such relevant data would improve classifier performance and enable improved prediction of ICO projects' success.

References

[1] Cryptocurrency Hedge Fund Performance: Risks and Challenges Usman Chohan SSRN Electronic Journal

[2] only one in ten tokens is in use following ICO OECD (2020). $https://www.bloomberg.com/news/articles/2017-10-23/only-one-in-10-tokens-is-in-use-following-initial-coin-offerings$

[3] Wei Xu, Ting Wang, Runyu Chen, J. Leon Zhao, Prediction of initial coin offering success based on team knowledge and expert evaluation, Decision Support Systems, Volume 147, 2021, 113574, ISSN 0167-9236, $https://doi.org/10.1016/j.dss.2021.113574$.

[4] ICO price, $https://towardsdatascience.com/icoomen-using-machine-learning-to-predict-ico-prices-29fa4cec6d86$

[5] Token Price $https://finance.yahoo.com/quote/ETH-USD/history/$

[6] miss forest, $https://rpubs.com/lmorgan95/MissForest$

[7] Miss forest alogrithm, $https://towardsdatascience.com/missforest-the-best-missing-data-imputation-algorithm-4d01182aed3$

[8] Stratified K fold, $https://scikit-learn.org/stable/modules/generated/sklearn.modelselection.Stratif$

[9] Oversampling, $https://beckernick.github.io/oversampling-modeling/$

[10] Multi layer Perceptron, $https://wiki.pathmind.com/multilayer-perceptron$

[11] F1 score, $https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/$

[12] Confusion Matrix, $https://machinelearningmastery.com/confusion-matrix-machine-learning/$