Will Your ICO Succeed? Predicting ICO Fundraising Success Using Machine Learning Models

1. Introduction

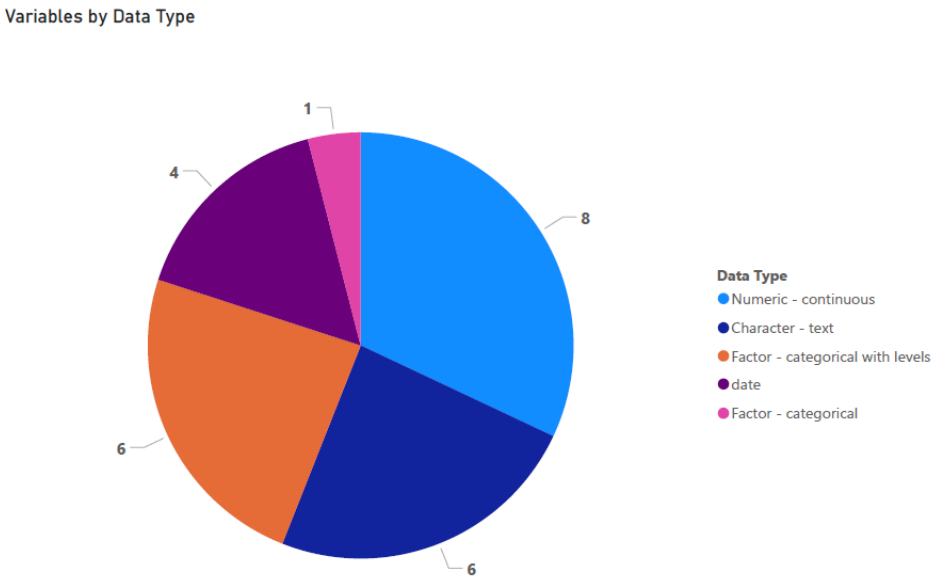
“ICOs are smart contracts based on blockchain technology that are designed for entrepreneurs to raise external finance by issuing tokens without an intermediary.” (Momtaz, 2020). This report focuses on developing different models and testing the models Using various evaluation techniques to choose the best model.

2. Data Understanding and Exploration

The data has 6146 observations an 25 variables.
The success variable shows that there are 2052 successful and 4094 unsuccessful ICOs.
The number tokens sold during the token ranges from as little as 1 to 22.6 quadrillion.
The number of team members associated with the project is between 1 to 66 members.
The dataset contains the following information:

VARIABLE	EXPLANATION	TYPE
country	The country where ICO is launched	Factor
ico_start	The starting date of token offering	date
ico_end	The end date of token offering	date
price_usd	Price of the token during ICO	Numeric
success	Whether the ICO was a success or failure	Factor
distributed_in_ico	Percentage of tokens distributed	Numeric
sold_tokens	Number of tokens sold during the token offering	Numeric
token_for_sale	Number of tokens released during the token offering	Numeric
whitelist	An indicator variable to check if the project used a whitelist process	Factor
kyc	An indicator variable to check if the project used a Know-Your-Customer process	Factor
bonus	An indicator variable to check if the project has information on bonus provided to token buyers	Factor
restricted_areas	Areas where investment in the token offering were restricted	Character
min_investment	Minimum investment required to participate in the ICO	Numeric
mvp	An indicator variable to check if Minimum Viable Product was available at the start of token offering	Factor
pre_ico_start	Start date of the pre-sale round	date
pre_ico_end	End date of the pre-sale round	date
pre_ico_price_usd	Token price at the pre-sale round	Numeric
accepting	Types of currencies accepted during the ICO	Character
link_white_paper	URL to the whitepaper of the project	Character
linkedin_link	URL to the LinkedIn page of the project	Character
github_link	URL to the GitHub link of the project	Character
website	URL to the website of the project	Character
rating	Rating of the ICO	Numeric

teamsize	Size of the team	Numeric
ERC20	Whether the token in ERC20 or not	Factor



3. Data Cleaning and Preparation

3.1 Handling missing values

The missing data values are:

VARIABLE	MISSING VALUES
country	1
ico_start	771
ico_end	907
price_usd	653
success	0
distributed_in_ico	1488
sold_tokens	1855
token_for_sale	1238
whitelist	2413
kyc	21
bonus	21
restricted_areas	3888
min_investment	4132
mvp	4875

pre_ico_start	3513
pre_ico_end	3525
pre_ico_price_usd	4631
accepting	828
link_white_paper	554
linkedin_link	1956
github_link	739
website	739
rating	712
teamsize	1719
ERC20	738

The target variable *success* has no missing value.

The following rows have very few missing data values and since, the rows containing them are deleted from the data:

- *country*
- *price_usd*
- *kyc*
- *bonus*
- ico_duration

Since, the variable ico_duration is created, rows containing missing values of both ico_start and ico_end are deleted.

The following columns are not critical for data analysis and modelling and hence is removed from the data:

- *linkedin_link*
- *github_link*
- *website*
- *link_white_paper*
- *pre_ico_start*
- *pre_ico_end*

The remaining data contain the following percentage of missing values:

VARIABLE	MISSING %
price_usd	4.12
distributed_in_ico	21.50 %

sold_tokens	26.85 %
token_for_sale	16.71 %
whitelist	35.94 %
min_investment	64.52 %
mvp	78.98 %
accepting	10.58 %
rating	10.44 %
teamsize	25.97 %
ERC20	10.74 %
restricted_areas	61.23

The variables *whitelist, min_investment, mvp, restricted_areas,* and *pre_ico_price_usd* have high number of missing values. Due to the high proportion of missing data and the fact that they may vary across projects, imputation is rather risky. Instead, binary indicators were created to retain potentially informative missingness and to know how accessible an ICO is.

For the features *accepting,* mode imputation is done. Since the variable is likely MAR, there is no potential bias in imputation.

Target imputation is done for *sold_tokens*, using *token_for_sale* and *distributed_in_ico*. *sold_tokens* is estimated using the following formula:

sold_tokens = token_for_sale × (distributed_in_ico / 100).

For the following variables with moderate missing values, MICE is used for imputation:

- *price_usd*
- *rating*
- *teamsize*
- *erc20*
- *distributed_in_ico*
- *token_for_sale*

3.2 Handling errors

The variables *'ico_start'* and *"ico_end"* were converted to proper date format.

The following categorical variables were encoded:

- *success*
- *whitelist*
- *kyc*
- *bonus*
- *mvp*

- *accepting*
- *ERC20*

This is to make the variables compatible with the various machine learning algorithms. Most algorithms cannot directly process text or categorical variables and require numerical representation. Binary encoding is used as the variables have only two possible categories and there is no meaningful order. This method is simple and efficient.

Text-heavy fields like *link_white_paper, linkedin_link, github_link,* and *website* were dropped while handling missing values.

For the following features, binary features were created:

- *whitelist*
- *min_investment*
- *mvp*
- *restricted_areas*
- *pre_ico_price_usd*

3.3 Feature Engineering

The length of fundraising period is calculated by creating a feature called 'ico_duration' using ico_start and ico_end. This temporal feature will act as a key indicator.

4. Modelling

The following classification models were applied:

Logistic Regression	This is baseline binary classifier for predicting probability of an event, in this case whether an ICO is successful or not.
Decision Tree	This is a simple, supervised learning model and will be used in splitting data based on features and then can be used for classification.
Random Forest	Another supervised learning model that combines predictions from multiple decision trees and can be used for classification.
Gradient Boosting – XGBoost	This is one of the highest-performing models on tabular data and can predict discrete class labels.

5. Evaluation

With the help of confusion matrix, the following results are achieved:

Model	Accuracy	Sensitivity	Specificity	Precision	F1_Score	AUC
Logistic	0.724	0.6579	0.7584	0.5859	0.6198	0.7750
Decision Tree	0.722	0.7624	0.7118	0.4010	0.5256	0.6924
Random Forest	0.732	0.6657	0.7677	0.6068	0.6349	0.7851
XGBoost	0.714	0.6408	0.7531	0.5807	0.6093	0.7728

Based on the results of the evaluation metrics, Random Forest is the best choice for ICO prediction. Accuracy is the fundamental evaluation metric, and Random Forest has achieved the high score. It provides the most correct predictions. High precision tells that it has highest positive predictions that are actually correct. High F1 score tells that the model is precise and robust.

6. Conclusion

With the data, various models were fitted and evaluated. Random Forest is chosen as the best model for predicting ICO success. It is an effective model for predicting success in classification problems.

References

Lantz, B. 2023. Machine learning with R : learn techniques for building and improving machine learning models, from data preparation to model tuning, evaluation, and working with big data Fourth edition. Birmingham, England: Packt Publishing.

Fisch, C. (2019). Initial coin offerings (ICOs) to finance new ventures. Journal of Business Venturing, 34(1), pp.1–22. https://doi.org/10.1016/j.jbusvent.2018.09.00

Huang, W., Vismara, S. and Wei, X. (2021). Confidence and capital raising. Journal of Corporate Finance, 66, 101900. https://doi.org/10.1016/j.jcorpfin.2021.101900

Momtaz, P.P. (2020). Initial Coin Offerings. PLOS ONE, 15(5), e0233018. https://doi.org/10.1371/journal.pone.0233018

Educative.io (n.d.) 'What is the grepl() function in R?', Educative.io. Available at: https://www.educative.io/answers/what-is-the-grepl-function-in-r (Accessed: 28 March 2025).

R Core Team (n.d.) 'scale: Scaling and Centering of Matrix-like Objects', R Documentation. Available at: https://www.rdocumentation.org/packages/base/versions/3.6.2/topics (Accessed: 28 March 2025).

DataCamp (n.d.) 'Generalized Linear Models Tutorial', DataCamp. Available at: https://www.datacamp.com/tutorial/generalized-linear-models (Accessed: 13 April 2025).

Gorman, B. (2014) 'Decision Trees in R using rpart', GormAnalysis, 24 August. Available at: https://www.gormanalysis.com/blog/decision-trees-in-r-using-rpart/ (Accessed: 13 April 2025).

finnstats (2021) 'Random Forest in R', R-bloggers, 13 April. Available at: https://www.r-bloggers.com/2021/04/random-forest-in-r/ (Accessed: 13 April 2025).

Tatman, R. (n.d.) 'Machine Learning with XGBoost (in R)', Kaggle. Available at: https://www.kaggle.com/code/rtatman/machine-learning-with-xgboost-in-r (Accessed: 13 April 2025).

GeeksforGeeks (2025) 'Evaluation Metrics in Machine Learning', GeeksforGeeks, 5 April. Available at: https://www.geeksforgeeks.org/metrics-for-machine-learning-model/ (Accessed: 14 April 2025).