Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 31, Num. 1 (2024) 50-59

RESEARCH ARTICLE

Predicting Startup Success Using Tree-Based Machine Learning Algorithms

Previsão do sucesso de uma startup usando algoritmos de aprendizado de máquina baseados em árvores

Saifur Rohman Cholil[1,2]*, Rahmat Gernowo[1], Catur Edi Widodo[1], Adi Wibowo[1], Budi Warsito[1], Alauddin Maulana Hirzan[2]

Abstract: Startups are an important element in today's digital economy. Increased interest in startups as a source of innovation and economic growth has prompted many studies to identify factors that can influence startup success. One of the challenges in predicting startup success is the diversity and complexity of the data. This research aims to identify the tree-based algorithm that achieves the highest accuracy in predicting startup success. The study employs tree-based methods using a single estimator (decision tree), ensemble bagging (bagging, random forest, and Extra Trees), and ensemble boosting (AdaBoost, gradient boosting, LGBM, and XGBoost). Model testing is conducted using evaluation matrices such as accuracy, classification model formation and confusion matrix. The results demonstrate that eXtreme Gradient Boosting (XGBoost) is the most effective prediction method for startup success rate when compared to other tree-based algorithms, achieving a high accuracy of 88.1%. The use of tree-based algorithms can provide useful insights for startup entrepreneurs in improving business strategies and decision-making. The key factors that have the most influence on startup success can be identified through the analysis of model test results, which is useful for startup entrepreneurs and investors in improving business performance.
Keywords: Benchmark — Prediction — Startup Tech — Tree-based Algorithms

Resumo: As startups são um elemento importante na economia digital atual. O aumento do interesse nas startups como fonte de inovação e crescimento econômico levou muitos estudos a identificar fatores que podem influenciar o sucesso das startups. Um dos desafios na previsão do sucesso de uma startup é a diversidade e a complexidade dos dados. Esta pesquisa tem como objetivo identificar o algoritmo baseado em árvore que atinge a maior precisão na previsão do sucesso de uma startup. O estudo emprega métodos baseados em árvores usando um único estimador (árvore de decisão), ensemble bagging (bagging, floresta aleatória e Extra Trees) e ensemble boosting (AdaBoost, gradient boosting, LGBM e XGBoost). O teste do modelo é realizado usando matrizes de avaliação, como precisão, formação do modelo de classificação e matriz de confusão. Os resultados demonstram que o eXtreme Gradient Boosting (XGBoost) é o método de previsão mais eficaz para a taxa de sucesso de inicialização em comparação com outros algoritmos baseados em árvore, alcançando uma alta precisão de 88,1%. O uso de algoritmos baseados em árvores pode fornecer percepções úteis para empreendedores de startups no aprimoramento das estratégias de negócios e na tomada de decisões. Os principais fatores que mais influenciam o sucesso de uma startup podem ser identificados por meio da análise dos resultados dos testes de modelos, o que é útil para empreendedores e investidores de startups melhorarem o desempenho dos negócios.
Palavras-Chave: Benchmark — Previsão — Tecnologia de Startups — Tree-based Algorithms

[1] *Information System School of Postgraduate, Universitas Diponegoro, Indonesia*
[2] *Information Technology and Communication, Universitas Semarang, Indonesia*
***Corresponding author**: saifurrohmancholil@students.undip.ac.id; cholil@usm.ac.id
DOI: http://dx.doi.org/10.22456/2175-2745.133375 • **Received:** 21/06/2023 • **Accepted:** 04/11/2023

1. Introduction

Startups are a topic that attracts attention in the business world because they have the potential to generate large profits in

the short and long term[1]. However, startup success is not easy to achieve, because many factors affect its success. According to Aminova and Marchi [2], approximately 90% of startup businesses are failed to run. Hence, many studies have been conducted to develop methods and algorithms to predict startup success including tree-based algorithm[3]. Like boosting algorithms, tree-based algorithms combine weak models to create a powerful and precise predictor. These algorithms make use of ensemble learning to grasp complicated relationships and provide accurate predictions by combining the individual decision tree strengths [4]. The tree-based approach is particularly effective in resolving categorization issues and predicting startup success. Tree-based algorithms have been used in various applications, such as cancer gene classification, breast cancer risk prediction, heart health[5, 6], underground fault classification [7], and predicting of amorphous alloys [8].

Some recent studies have shown that the tree-based algorithm is more effective in predicting startup success compared to other methods. While various approaches have been explored, this particular algorithm has demonstrated promising results in accurately forecasting the potential success of startups [9, 10, 11, 12]. The research shows that the tree-based algorithm is more accurate in breast cancer than other Deep Learning methods. Other research conducted using the tree-based algorithm includes classifying impact damage of fiber-reinforced concrete panels, classifying gene cancer on expression data, and classifying underground fault cables [13]. Tree-based algorithm was also used to detect anomalies on displacement rates and deformation pattern features in Indonesia[14]. The tree-based algorithm can be used to predict factors that affect startup success [15, 16] such as funding especially financial fraud [17], financial distress [18], business strategy, marketing, bankruptcy [19], and product development. Tree-based algorithms possess the ability to learn patterns and trends associated with these factors, allowing for more accurate predictions about startup success. In the article [20], the authors utilized tree-based algorithm to identify factors that influence the success of circular economy strategies in startups. The results indicated that tree-based approach was effective in identifying key factors contributing to the strategy's success, such as capital availability, team skills, and a favorable regulatory environment. Overall, the tree-based algorithm has proven to be effective in various applications for predicting key factors that affect startup success. By using this algorithm, business decision-makers can identify key factors that need to be im- proved to increase startup success. Therefore, further research on the use of the tree-based algorithm in the startup context is expected to provide a clearer and more accurate view of the key factors affecting startup success[21],

The primary objective of this research is to determine the most accurate startup prediction using tree-based classification models capable of assessing a startup's likelihood of success based on pertinent features or factors. Some Tree-based algorithms in machine learning aim to achieve optimal accuracy and are categorized into three main types (e.g. decision tree), ensemble bagging (e.g. bagging, random forest, and Extra Trees), and ensemble boosting (e.g. AdaBoost, gradient boosting, LGBM, and XGBoost). These algorithms play a crucial role in the startup ecosystem. Having an accurate prediction model can help decision-makers evaluate the opportunities and risks of investing in startups.

This includes identifying factors that contribute to startup success, such as funding, failure, founding team experience, business model, or other factors that can be used as features in the prediction model. The ultimate objective is to enhance the understanding of startup success prediction, equipping stakeholders with practical insights to make informed decisions in the realm of startup investment

2. Methods

2.1 Research Method

The research methodology consists of seven stages: data collection, data preprocessing, feature selection, data splitting, model training, model testing, and result analysis. These stages are depicted in Figure 1.

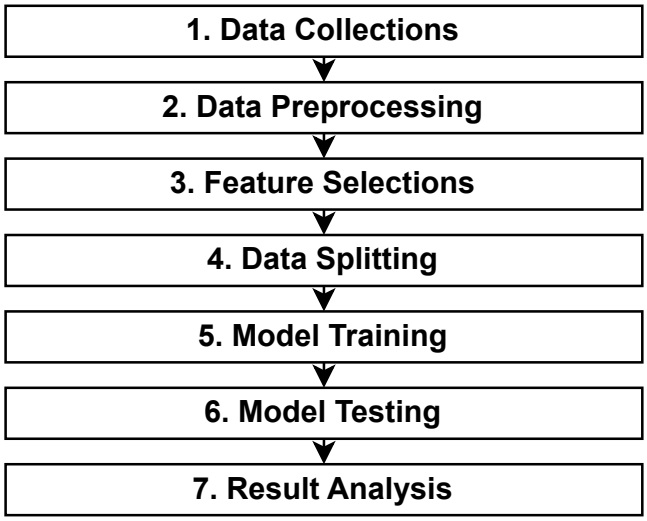

Figure 1. Research Method

2.1.1 Data Collection

First, the data required to test the tree-based algorithm in predicting startup success needs to be collected. The dataset obtained from Kaggle contains information essential for predicting startup success [22] which is startup data in the United States contains 923 rows and 48 features, including quantitative and categorical attributes, such as age at first and last funding year, relationships, funding rounds, total funding (USD), milestones, state, industry type, presence of venture capital (VC), angel investors, and funding rounds (Round A, B, C, D). The target variable "status" categorizes startups as "acquired" or "closed," representing the ultimate measure of startup success or failure, respectively. The dataset's potential to predict startup success allows investors and decision-makers

to gain a competitive advantage by identifying high-growth prospects and fostering a thriving entrepreneurial ecosystem. The dataset was used in data sprint #5 at DPhi, and acknowledgments go to Ramkishan Panthena, a Machine Learning Engineer at GMO, for providing this dataset. The data is processed using machine learning to determine the distribution of successful and unsuccessful (closed) startup data based on the categories presented in Figure 2.

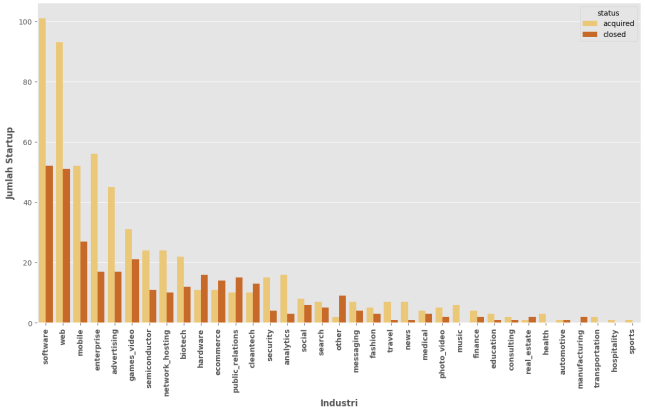

Figure 2. Startup Categories

From Figure 2, it can be seen that the software category has the highest number of successes followed by startups in the web and other categories. The smallest number of successes is the sports, hospitality, and other categories. Next, we will look at the data based on startup funding presented in Figure 3.

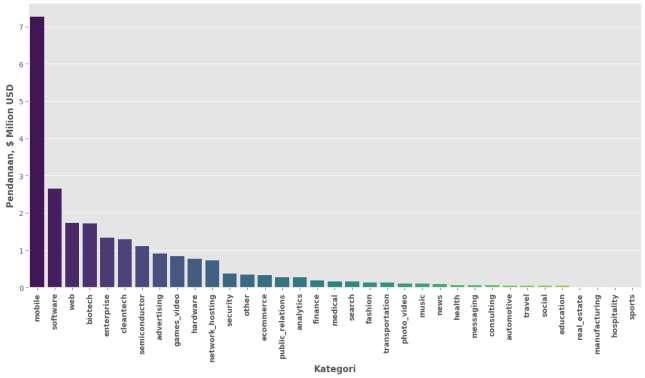

Figure 3. Startup Financing

2.1.2 Data Preprocessing

Once the data has been collected, the next step is to perform data preprocessing. This includes processing missing data, removing irrelevant data, and data normalization. Data preprocessing should also consider the problem of unbalanced data. Furthermore, to aid in data preprocessing, we analyze the correlation matrix of each attribute to identify relationships between variables. The results of this analysis are presented in Figure 4. The attributes with a correlation value close to 1 are selected as training data, resulting in 36 attributes used for the training process.

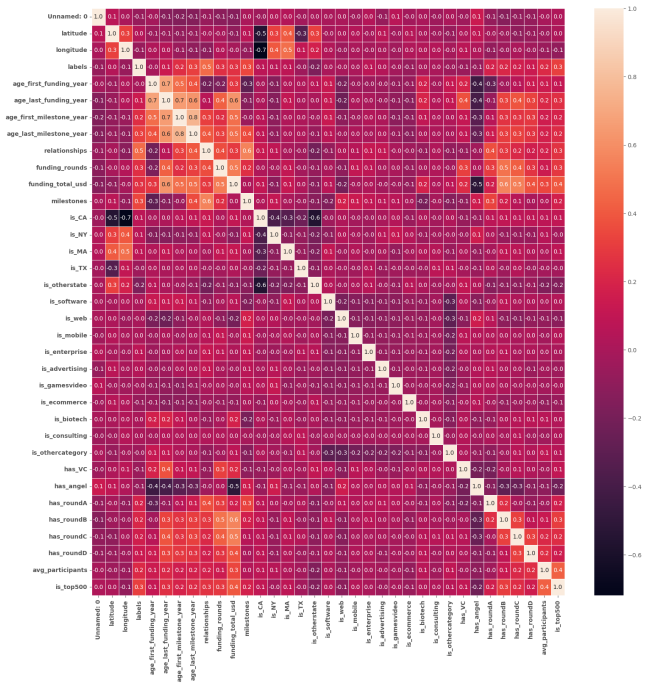

Figure 4. Attributes Correlations

Attribute correlation provides an overview of the relationship between variables and also the contribution of input features to the output. The value of the correlation coefficient ranges from 0 to 1. A value of 0 indicates no correlation, while a value of 1 describes a full correlation. Good selection means that the selected input variables have a small correlation. The features selected in this study are good candidates for investigating ML models. The target distribution used to display the number of successful and unsuccessful startup data is presented in Figure 5.

Figure 5. Target Distributions

Referring to Figure 5, the distribution of targets in successful startups is higher than that of unsuccessful ones. It shows that the success rate is higher than that of unsuccessful startups.

2.1.3 Feature Selection

Feature selection plays a vital role in predicting the success of startups through machine learning. By carefully select-

ing a diverse set of attributes, we aim to pinpoint the most significant factors that influence the outcomes of emerging businesses. This approach not only improves prediction accuracy but also offers valuable insights for entrepreneurs, investors, and policymakers [20]. Our research highlights the importance of feature selection in understanding the complex dynamics of startup success prediction. We selected 26 features for our training. These chosen attributes include Milestones, Category Code, Venture Capital Involvement, Geographical Location, Industry Categorization, and Funding Rounds. Each feature plays a crucial role in predicting startup success, capturing various aspects of a startup's journey. 'Funding_rounds' and 'milestones' signify financial stability and growth progress, while 'category_code' determines market positioning and industry-specific opportunities. Additionally, venture capital involvement, angel investors, and multiple funding rounds reflect the impact of external funding support. Geographical location and industry categorization provide insights into regional advantages and industry-specific growth patterns. The figure in Figure 6 displays the feature importance obtained from our analysis, showcasing the attributes that hold relevance in predicting startup success.

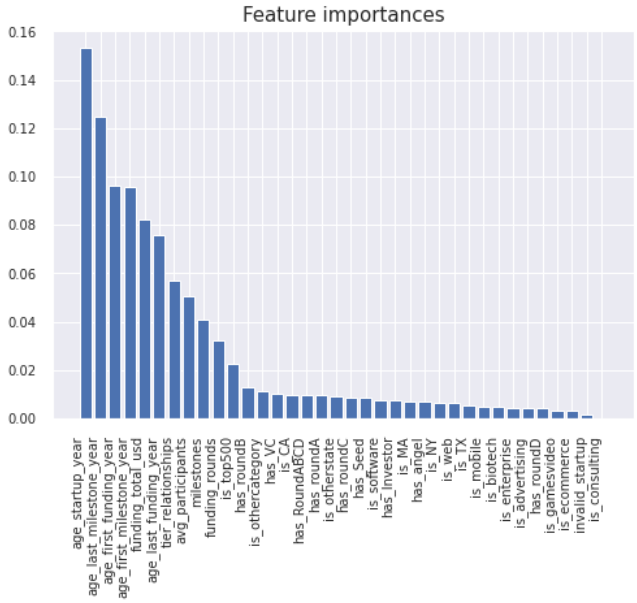

Figure 6. Feature Importance

2.1.4 Data Splitting

After preprocessing the data, the next step is to divide the data into two parts, namely training data and testing data. Training data uses 70% of existing data while testing data uses 30% of existing data that will be used to test the performance of the algorithm. This splitting technique will affect the result of the model[23].

2.1.5 Model Training

The next step is to train the tree-based model on the training data. The model training should consider several parameters, such as the number of decision trees, the maximum depth of the tree, and the learning rate. These parameters can be adjusted to achieve optimal model performance.

2.1.6 Model Testing

Once the model is trained, the next step is to test the performance of the model on the test data. Model performance can be evaluated using evaluation metrics such as accuracy, precision, recall, and F1-score [24, 25]. The phases from Data Collection to this phase is shown in Figure 7.

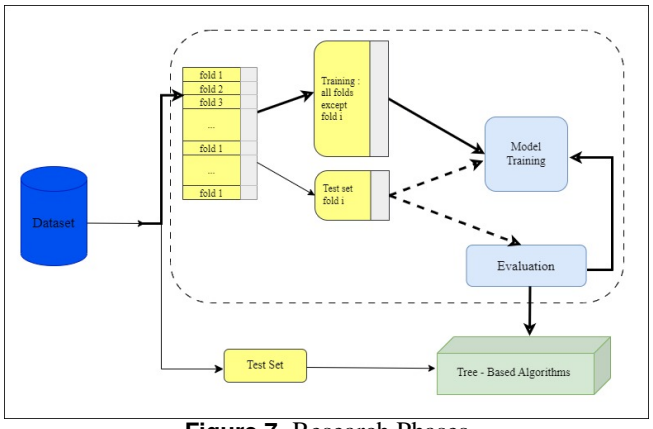

Figure 7. Research Phases

Figure 7 explains the process used to predict startup success. The first step is to create a dataset followed by dividing it into n subsets called folds. Each fold will act as a validation dataset in a particular turn, while the other subsets are used as training datasets. At each iteration, one fold is taken as the validation dataset, while the other folds (the remaining folds) are used as the training dataset. The XGBoost model is trained using a training dataset consisting of all folds except fold i. Fold I, which has been determined as the validation dataset, is used to test the performance of the model after the training process, which aims to help in measuring the performance of the model on data that has never been seen before. The XGBoost model trained using the training dataset creates a small decision tree to make predictions. Model performance is evaluated using the validation dataset with evaluation matrices such as accuracy, precision, recall, or F1-score can be used to measure the quality of model predictions. After all iterations, the XGBoost model is trained using all training subsets except the validation subset at each iteration to create a stronger and more general final model. The use of cross-validation schemes and XGBoost models can obtain more stable model performance estimates and reduce the possibility of overfitting training data.

2.1.7 Result Analysis Method

After carrying out the above steps, the results from model testing need to be analyzed. This includes analyzing the key factors that have the most influence on startup success, as well as analyzing the performance of the XGBoost algorithm in predicting startup success.

2.2 Single Estimator (Decision Tree)

In the context of supervised learning, decision trees have been historically considered as a reliable method for handling categorization challenges. However, it's worth noting that the landscape of machine learning algorithms has evolved, and different approaches have emerged as viable alternatives. For instance, recent studies [21] demonstrate the effectiveness of ensemble methods in addressing categorization difficulties, showcasing the advancements in this domain. This strategy, represented as a model or graph like a tree, involves assessing features at internal nodes. Each branch represents a possible test result, whereas the leaf nodes represent the final categorization result. Decision trees provide an intuitive and interpretable framework for modeling complex relationships within the data [26]. Decision trees are a powerful algorithm for knowledge representation and reasoning under uncertainty. Unlike black-box models such as Nearest Neighbor or Support Vector Machine[23], decision trees offer the advantage of representing the interrelationships among dataset features, providing valuable insights.

The decision tree technique effectively separates the data based on the most discriminative qualities by generating a tree-like structure. Each decision tree internal node represents a feature test, allowing for informative splits that drive the classification process [6]. Predictive variables are employed to partition data into distinct datasets based on the target variable. The decision tree algorithm examines the predictor variable and splits the target variable into two classes. To enhance accuracy, numerous decision trees, such as through bagging, are constructed using bootstrap samples from the training dataset [14].

We will examine x, which is a hyper-rectangle contained in Rd and encompasses target training instances. An a represents the set of attributes $\{a_1, a_2, ..., a_n\}$, while a x represents the set of instances $\{x_1, x_2, ..., x_n\}$. When considering a specific node, denoted as t, A_t denotes the set of eligible attributes for division, where A_t is a subset of A. We use $A_t = \{a'1, a'2, ..., a'lt\}$ to represent the eligible attributes at node t, where lt is the count of eligible attributes at that node, and lt is less than or equal to d. At each node t, the algorithm searches for the attribute $a'j \in$ At that provides the most effective division of the initial sub-space t into one or multiple sub-spaces x_{ti}, satisfying the given conditions in Equation 1.

$$X_{ti} = \left\{ x \in L_{ti} \leq x^{ai} \leq R_{ti} \right\} \tag{1}$$

The value of instance x for attribute $a'j$ is represented by $x'a'j$. L_{ti} and R_{ti} are the left and right bounds of the closed sub-intervals, used to divide the current node t into target nodes ti, according to the attribute $a'j$.

2.3 Ensemble Bagging (Bagging, Random Forest, and ExtraTrees)

The bagging classifier serves as a meta-classifier ensemble technique utilized to enhance the stability and accuracy of tree-based algorithms. By leveraging bootstrap sampling, the bagging classifier combines multiple tree models trained on different data subsets. The resulting ensemble model generates more robust and accurate predictions [27]. This method effectively addresses the challenges of overfitting and variance, leading to improved performance and enhanced predictive capabilities in determining the likelihood of startup success. The Bagging classifier can be utilized to predict the success of startups by employing multi-task learning principles. In this scenario, we have two related learning tasks, namely representation learning and classification, denoted as $T = \{t_1, t_2\}$. The Autoencoder t_1 and the classifier t_2 are jointly trained to enhance the expressiveness of the learned representations and improve the generalization of the startup success predictor. To achieve this, the training process combines supervised and unsupervised loss functions, namely binary cross entropy and mean squared error, respectively. The ultimate goal is to develop a classifier that can accurately predict the success of startups. The final cost function of the supervised autoencoder is defined accordingly to optimize the performance of the Bagging classifier in predicting startup success as shown in Equation 2.

$$\frac{1}{m} \sum_{i=1}^{m} \left[(y) L_r (\hat{x}_i, x_i) + (1-\gamma) L_p (\hat{y}_i, y_i) \right] \tag{2}$$

The network's reconstruction loss (Lr) is of great significance. It is determined by computing the Mean Squared Error loss between the input data (x) and the reconstructed values ($\hat{x}$) in the output layer of the autoencoder, as illustrated in Equation 3.

$$L_r = \frac{1}{m} \sum_{i=1}^{m} (x - \hat{x})^2 \tag{3}$$

The prediction loss, denoted as L_p (as indicated in Equation 4, is used to predict the labels of input data using the learned representations. It is computed as the binary cross-entropy loss between the true labels (yi) of each data sample (xi) and the predicted labels ($\hat{y}i$) based on the learned representations.

$$L_p = \frac{1}{m} \sum_{i=1}^{m} yi \cdot log\left(p\left(\hat{y}i\right)\right) + (1-yi) \cdot log\left(1 - \left(p\left(\hat{y}i\right)\right)\right) \tag{4}$$

To put it differently, following the principles of multi-task learning, the reconstruction loss is integrated into the classification loss of the neural network structure.

The random forest classifier is an ensemble algorithm that builds upon the decision tree method. In this approach, each tree in the random forest is created using a random vector sample, irrespective of the input vector. This intentional randomness brings diversity among the individual trees, effectively reducing the risk of overfitting [28]. By combining

the predictions from multiple randomized trees, the random forest classifier improves accuracy and strengthens the predictive performance for various outcomes. The Extra Trees classifier creates an ensemble of base estimators, just as the random forest classifier. It adds an additional dimension of randomization, though, which broadens the scope of this idea. Each base estimator of the Extra Trees classifier is trained on a random subset of features during the training phase. The diversity among the various trees is increased through this intentional randomization, which also improves generalization and boosts the algorithm's overall predictive power [29].

In the methodology, the approach for achieving accurate predictions using the Random Forest algorithm is described. The process involves several steps.

Step 1: Gathering ntree bootstrap samples from the training set of samples obtained from the decision makers is the first stage. Consider a dataset (C, V) in which the observations are composed of the elements $c_1, c_2, c_3, ..., c_d$ and the replies are composed of $v_1, v_2, v_3, ..., v_d$. A second set of data (C_c, V_c) of size d is produced by randomly picking and replacing data to create bootstrap T datasets of size d.

Step 2: A dataset (C_c, V_c) is utilized to train a decision tree for each $C = 1, 2, 3, ..., c$.

Step 3: To forecast new data, the ntree trees' predictions are integrated. To forecast new observations, each decision tree **Cnew** uses the vote of most of its trees.

Step 4: After the predictions are taken out of the bootstrapped sample, the mean prediction error of each decision tree's prediction is determined. This makes it easier to see an out-of-bag (OOB) mistake. To calculate the error rate, the dataset may also be divided into a train and test set.

2.4 Ensemble Boosting (Gradient Boosting, AdaBoost, LGBM, and XGBoost)

The boosting classifier is introduced as a potent predictive modeling algorithm that combines multiple weak learners to generate sequential strong learners. As an ensemble technique, boosting significantly enhances prediction accuracy by iteratively mitigating bias and transforming it into a more resilient and influential framework [30]. Gradient boosting, AdaBoost, LGBM, and XGBoost are a few examples of boosting techniques.

Friedman popularized the procedure known as gradient boosting, which incorporates statistical modeling concepts from earlier models. Gradient boosting effectively converts weak learning problems into gradient descent problems utilizing a loss function through the iterative addition of trees and parameter modifications. This method enhances predictive capabilities significantly by efficiently leveraging complex patterns within the data[31]. As a result, the model gains a deeper understanding of nuanced linkages, leading to precise and accurate forecasts[32]. XGBoost, on the other hand, is an enhanced version of gradient boosting that incorporates additional feature processing techniques such as parallel processing, tree pruning, and imputation of missing values [22, 24]. Extreme Gradient Boosting (XGBoost) is a decision tree-based machine learning technique that achieves extremely high accuracy and processing efficiency[25]. Gradient-boosting reasoning is used to reduce the chance of overfitting, while sequential training is used to improve predictive models. Cross-validation is used in XGBoost to reduce overfitting and capture complicated non-linear interactions. Multithreading is also used to improve computational accuracy. It is advised for use in the prediction of natural occurrences due to its success in several practical applications[33]. The XGBoost model runs an iterative procedure to optimize the objective function by updating the parameters in one step using the residuals from the previous step [1]. The objective function (J) of the XGBoost model is expressed as $J(\theta)$ in Equation 5.

$$J(\phi) = L(\phi) + \Omega(\phi) \tag{5}$$

where θ represents the learned parameter and $L(\theta)$ is the loss function $\Omega(\theta)$ is the regularization term that governs the model's complexity. The prediction scores from all trees are added together to yield the final score, with the output $(\hat{y}i)$ given in Equation 6.

$$\hat{y}_i = \sum_{k=i}^{k} f_x(x_i), f_k \in F \tag{6}$$

Here, xi signifies the vector of input variables, i denotes the number of inputs, K denotes the number of trees, fk denotes the scoring function of the independent tree structure used to estimate the output in the functional space F, and F denotes the set of all feasible classification and regression trees. Equation expresses the model's objective Equation 7.

$$J(\theta) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{k} \Omega(f_k) \tag{7}$$

where N denotes the number of predictions. Equation expresses the prediction of $\hat{y}I$ at the nth iteration in Equation 8.

$$\hat{y}_i^{(n)} = \hat{y}_i^{(n-1)} + \lambda f_n(x_i) \tag{8}$$

a λ in which the learning rate or step size. As a result, the objective function at the nth round can be written as Equation 9.

$$\begin{aligned} J^n &= \sum_{n=1}^{n} L(y_i, \hat{y}_i^n) + \sum_{n=1}^{n} \Omega(f_k) \\ &= \sum_{n=1}^{n} L(y_i, \hat{y}_i + \lambda f_n(x_i)) + \Omega(f_n) \end{aligned} \tag{9}$$

Equation 10 expresses the regularization term at the nth iteration, and Equation 11 expresses Gain:

$$\Omega(f_n) = \gamma T + \frac{1}{2}\alpha \sum_{i=1}^{T} w_j^2 \tag{10}$$

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\alpha} + \frac{G_R^2}{H_R+\alpha} + \frac{(G_L^2+G_R^2)^2}{H_L+H_R+\alpha}\right] \tag{11}$$

The popular ensemble learning technique known as Adaboost, or Adaptive Boosting, combines several weak learners to produce a powerful classifier. It improves overall performance by repeatedly adjusting the weights of misclassified samples to concentrate on challenging situations [34]. By focusing on the value of difficult cases throughout the model training phase, Adaboost is especially good at solving complicated categorization issues. The gradient boosting architecture known as LightGBM, which stands for Light Gradient Boosting Machine, uses tree-based methods. It is renowned for its great scalability and efficiency, making it appropriate for processing huge amounts of information [35]. To achieve quicker training times and more accuracy, LightGBM uses a unique gradient-based technique that makes use of histogram-based binning and a leaf-wise tree growth strategy. LightGBM is an effective technique for forecasting complicated outcomes in a variety of fields since it can handle high-dimensional and sparse data [36].

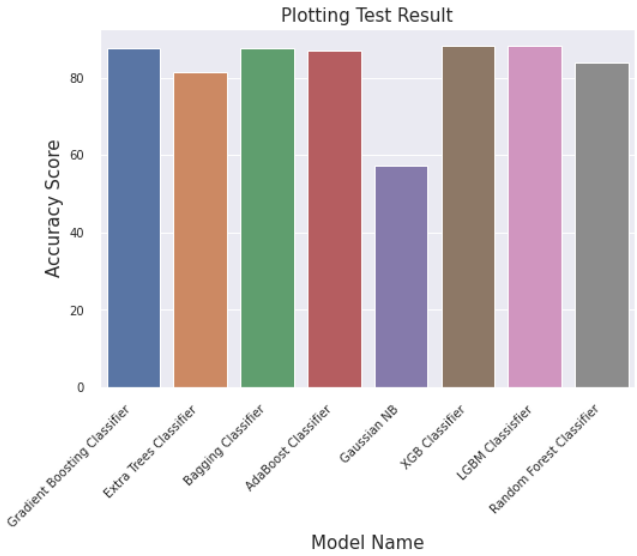

Figure 8. Classification Methods' Accuracy

The next step is to create a classifier model with the extreme gradient boosting method presented in Table 1.

Table 1. Classification Models' Benchmarks

Model	Accuracy	Train & Test Time (second)
XGBoost	0.8810	0.24
LGBM	0.8810	0.10
Gradient Boosting	0.8750	0.27
Bagging	0.8750	0.06s
AdaBoost	0.8690	0.13
Random Forest	0.8393	0.25
Extra Trees	0.8155	0.21
Gaussian NB	0.5714	0.01

3. Results and Discussion

To comprehensively assess the effectiveness of tree-based algorithms in predicting startup success, we aim to compare multiple tree-based algorithms in this study. By considering a range of tree-based approaches, including but not limited to XGBoost, we seek to evaluate their performance and identify the most suitable algorithm for accurate startup success prediction. This comparative analysis will provide valuable insights into the strengths and limitations of different tree-based algorithms, enabling us to make informed recommendations for predicting startup success effectively. The accuracy result shows 98.9% to handle the startup success problem. The accuracy of this method is higher than other methods in the classification model including the K Neighbors method which has an accuracy of 83.1%, the Naïve Bayes method has an accuracy of 93.8%, the SVM method has an accuracy of 96.9% and other methods. The accuracy results of several methods belonging to the classification category are presented in Figure 8.

We compared the performance of several algorithms, including Gradient Boosting Classifier (87.50%), Extra Trees Classifier (81.55%), Bagging Classifier (87.50%), AdaBoost Classifier (86.90%), Gaussian NB (57.14%), XGB Classifier (88.10%), LGBM Classifier (88.10%), and Random Forest Classifier (83.93%). The results demonstrate that the tree-based algorithms, XGB Classifier and LGBM Classifier, achieved the highest accuracy of 88.10%. These models outperformed other algorithms, making them the most suitable for classifying startup success based on the given features. On the other hand, Gaussian NB (Naive Bayes) showed a lower accuracy of 57.14%, indicating that this algorithm may not be the most appropriate choice for this particular task. Furthermore, the computational efficiency of the models varied, with Bagging Classifier demonstrating the shortest train and test time (0.06s), followed closely by Gaussian NB (0.01s), while Gradient Boosting Classifier had the longest train and test time (0.27s). In conclusion, the findings highlight the effectiveness of tree-based algorithms, particularly XGBoost and LGBM, in accurately predicting startup success based

on relevant features. The evaluation of the model used in the startup success classification is presented in Figure 9.

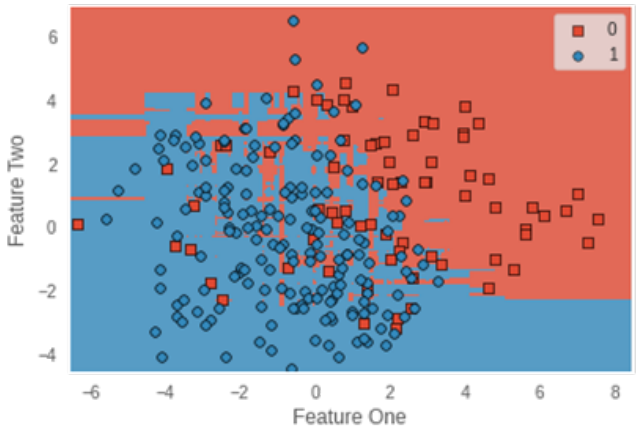

Figure 9. The Model Evaluation

Another evaluation using the confusion matrix to display plots of the accuracy that has been made is presented in Figure 10.

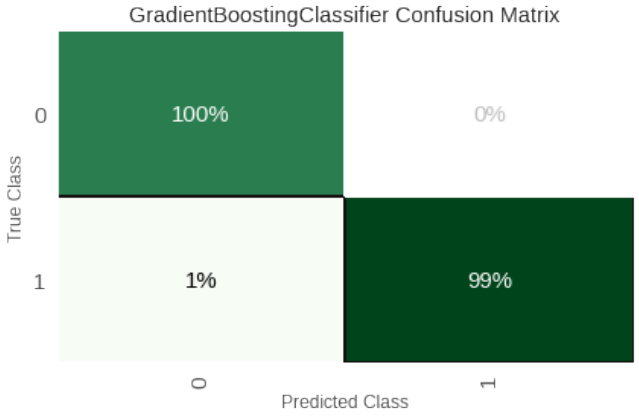

Figure 10. Classification Methods' Accuracy

The model in Figure 9 can predict most of the positive successful startup data (blue dots), but there are some failed startup data that are predicted to be successful (red dots) this is due to the imbalance dataset in the "status" column where the number of successful startups is more than unsuccessful startups.

A confusion matrix is used to evaluate the performance of a classification model which gives an idea of how well the model can classify the data correctly or incorrectly. Confusion matrix consists of four components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Referring to Figure 10, it explains that the True Positive (TP) of 178 is the amount of positive data that is correctly classified by the model. A total of 178 positive data were classified correctly. True Negative (TN) of 73 is the amount of negative data correctly classified by the model. A total of 73 negative data were correctly classified. Thus, the total result is 100%. Meanwhile, False Positive (FP) of 2 is the amount of negative data that is misclassified as positive by the model. A total of 2 negative data were misclassified as positive. The end result for this case is 99%.

4. Conclusion

Tree-based algorithms offer a promising approach for predicting startup success. Extensive studies have consistently demonstrated their superior performance compared to other machine learning methods. Applying tree-based algorithms involves essential steps like data collection, preprocessing, sharing, model training, and testing. Evaluating the model's performance using metrics like f1-score, accuracy, precision, and recall provides valuable insights into the key factors influencing startup success. These insights can guide entrepreneurs in refining business strategies and making informed decisions. Leveraging tree-based algorithms opens new avenues for startup entrepreneurs to gain meaningful insights and enhance their overall business strategies and decision-making processes.

Author contributions

Saifur Rohman Cholil has performed the research and writing article under Rahmat Gernowo and Catur Edi Widodo's supervision. Adi Wibowo and Budi Warsito has performed literature review to support the article and research. Alauddin Maulana Hirzan has done performing document conversion and added relevant citation to the article.

References

[1] DARADKEH, M.; MANSOOR, W. The impact of network orientation and entrepreneurial orientation on startup innovation and performance in emerging economies: The moderating role of strategic flexibility. *Journal of Open Innovation: Technology, Market, and Complexity*, v. 9, n. 1, p. 100004, mar. 2023. ISSN 21998531. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2199853123001063⟩.

[2] AMINOVA, M.; MARCHI, E. The Role of Innovation on Start-Up Failure vs. its Success. *International Journal of Business Ethics and Governance*, p. 41–72, jan. 2021. ISSN 2717-9923. Disponível em: ⟨https://ijbeg.com/index.php/1/article/view/60⟩.

[3] THAI, D.-K. et al. Classification models for impact damage of fiber reinforced concrete panels using Tree-based learning algorithms. *Structures*, v. 53, p. 119–131, jul. 2023. ISSN 23520124. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2352012423005350⟩.

[4] KIRAN, S. et al. A Gradient Boosted Decision Tree with Binary Spotted Hyena Optimizer for cardiovascular disease detection and classification. *Healthcare Analytics*, v. 3, p. 100173, nov. 2023. ISSN 27724425. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2772442523000400⟩.

[5] SNOUSY, M. B. A. et al. Suite of decision tree-based classification algorithms on cancer gene expression data. *Egyptian Informatics Journal*, v. 12, n. 2, p. 73–82, jul. 2011. ISSN 11108665. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S1110866511000223⟩.

[6] GHIASI, M. M.; ZENDEHBOUDI, S. Application of decision tree-based ensemble learning in the classification of breast cancer. *Computers in Biology and Medicine*, v. 128, p. 104089, jan. 2021. ISSN 00104825. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S0010482520304200⟩.

[7] MISHRA, S. et al. Tree Based Fault Classification in Underground Cable. *Procedia Computer Science*, v. 218, p. 524–531, 2023. ISSN 18770509. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S1877050923000340⟩.

[8] BOBBILI, R. Interpretable glass forming ability prediction of amorphous alloys through tree based algorithms. *Materials Letters*, v. 349, p. 134774, out. 2023. ISSN 0167577X. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S0167577X2300959X⟩.

[9] HOU, T. et al. Marine floating raft aquaculture extraction of hyperspectral remote sensing images based decision tree algorithm. *International Journal of Applied Earth Observation and Geoinformation*, v. 111, p. 102846, jul. 2022. ISSN 15698432. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S1569843222000486⟩.

[10] BANSAL, M.; GOYAL, A.; CHOUDHARY, A. A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning. *Decision Analytics Journal*, v. 3, p. 100071, jun. 2022. ISSN 27726622. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2772662222000261⟩.

[11] CINAR, A. C.; KORKMAZ, S.; KIRAN, M. S. A discrete tree-seed algorithm for solving symmetric traveling salesman problem. *Engineering Science and Technology, an International Journal*, v. 23, n. 4, p. 879–890, ago. 2020. ISSN 22150986. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2215098619313527⟩.

[12] AN, Y.; ZHOU, H. Short term effect evaluation model of rural energy construction revitalization based on ID3 decision tree algorithm. *Energy Reports*, v. 8, p. 1004–1012, jul. 2022. ISSN 23524847. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2352484722002402⟩.

[13] RUIZ-VILLAFRANCA, S. et al. A MEC-IIoT intelligent threat detector based on machine learning boosted tree algorithms. *Computer Networks*, p. 109868, jun. 2023. ISSN 13891286. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S1389128623003134⟩.

[14] WIBOWO, A. et al. Anomaly detection on displacement rates and deformation pattern features using tree-based algorithm in Japan and Indonesia. *Geodesy and Geodynamics*, v. 14, n. 2, p. 150–162, mar. 2023. ISSN 1674-9847. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S1674984722000702⟩.

[15] ROCCATELLO, E. et al. Impact of startup and defrosting on the modeling of hybrid systems in building energy simulations. *Journal of Building Engineering*, v. 65, p. 105767, abr. 2023. ISSN 23527102. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2352710222017739⟩.

[16] ANG, Y. Q.; CHIA, A.; SAGHAFIAN, S. Using Machine Learning to Demystify Startups' Funding, Post-Money Valuation, and Success. In: BABICH, V.; BIRGE, J. R.; HILARY, G. (Ed.). *Innovative Technology at the Interface of Finance and Operations: Volume I*. Cham: Springer International Publishing, 2022. p. 271–296. ISBN 978-3-030-75729-8. Disponível em: ⟨https://doi.org/10.1007/978-3-030-75729-8_10⟩.

[17] AFRIYIE, J. K. et al. A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. *Decision Analytics Journal*, v. 6, p. 100163, mar. 2023. ISSN 27726622. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2772662223000036⟩.

[18] AYDIN, N. et al. Prediction of financial distress of companies with artificial neural networks and decision trees models. *Machine Learning with Applications*, v. 10, p. 100432, dez. 2022. ISSN 26668270. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2666827022001074⟩.

[19] SHETTY, S.; MUSA, M.; BRéDART, X. Bankruptcy Prediction Using Machine Learning Techniques. *Journal of Risk and Financial Management*, v. 15, n. 1, p. 35, jan. 2022. ISSN 1911-8074. Disponível em: ⟨https://www.mdpi.com/1911-8074/15/1/35⟩.

[20] OPSTAL, W. V.; BORMS, L. Startups and circular economy strategies: Profile differences, barriers and enablers. *Journal of Cleaner Production*, v. 396, p. 136510, abr. 2023. ISSN 09596526. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S0959652623006686⟩.

[21] LI, T. et al. Economic Granularity Interval in Decision Tree Algorithm Standardization from an Open Innovation Perspective: Towards a Platform for Sustainable Matching. *Journal of Open Innovation: Technology, Market, and Complexity*, v. 6, n. 4, p. 149, dez. 2020. ISSN 21998531. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2199853122011246⟩.

[22] ŻBIKOWSKI, K.; ANTOSIUK, P. A machine learning, bias-free approach for predicting business success using Crunchbase data. *Information Processing & Management*, v. 58, n. 4, p. 102555, jul. 2021. ISSN 03064573. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S0306457321000595⟩.

[23] KIM, S. Y.; UPNEJA, A. Majority voting ensemble with a decision trees for business failure prediction during economic downturns. *Journal of Innovation & Knowledge*, v. 6, n. 2, p. 112–123, abr. 2021. ISSN 2444569X. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2444569X21000081⟩.

[24] BATBOOTI, R. S.; RANSING, R. S. A novel imputation based predictive algorithm for reducing common cause variation from small and mixed datasets with missing values. *Computers & Industrial Engineering*, v. 179, p. 109230, maio 2023. ISSN 03608352. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S0360835223002541⟩.

[25] TAN, B.; GAN, Z.; WU, Y. The measurement and early warning of daily financial stability index based on XGBoost and SHAP: Evidence from China. *Expert Systems with Applications*, v. 227, p. 120375, out. 2023. ISSN 09574174. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S0957417423008771⟩.

[26] JIN, C. et al. Sampling scheme-based classification rule mining method using decision tree in big data environment. *Knowledge-Based Systems*, v. 244, p. 108522, maio 2022. ISSN 09507051. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S0950705122002258⟩.

[27] SUBASI, A.; KADASA, B.; KREMIC, E. Classification of the Cardiotocogram Data for Anticipation of Fetal Risks using Bagging Ensemble Classifier. *Procedia Computer Science*, v. 168, p. 34–39, 2020. ISSN 18770509. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S1877050920303872⟩.

[28] KUMAR, L. S. et al. Random forest tree classification algorithm for predicating loan. *Materials Today: Proceedings*, v. 57, p. 2216–2222, 2022. ISSN 22147853. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2214785321080585⟩.

[29] SHARMA, D.; KUMAR, R.; JAIN, A. Breast cancer prediction based on neural networks and extra tree classifier using feature ensemble learning. *Measurement: Sensors*, v. 24, p. 100560, dez. 2022. ISSN 26659174. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2665917422001945⟩.

[30] NAEM, A. A.; GHALI, N. I.; SALEH, A. A. Antlion optimization and boosting classifier for spam email detection. *Future Computing and Informatics Journal*, v. 3, n. 2, p. 436–442, dez. 2018. ISSN 23147288. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2314728818300746⟩.

[31] KIANGALA, S. K.; WANG, Z. An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment. *Machine Learning with Applications*, v. 4, p. 100024, jun. 2021. ISSN 26668270. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2666827021000050⟩.

[32] Abdullah-All-Tanvir et al. A gradient boosting classifier for purchase intention prediction of online shoppers. *Heliyon*, v. 9, n. 4, p. e15163, abr. 2023. ISSN 24058440. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S2405844023023708⟩.

[33] SHRESTHA, S. M.; SHAKYA, A. A Customer Churn Prediction Model using XGBoost for the Telecommunication Industry in Nepal. *Procedia Computer Science*, v. 215, p. 652–661, 2022. ISSN 18770509. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S187705092202138X⟩.

[34] ZHOU, L. Predicting the Removal of Special Treatment or Delisting Risk Warning for Listed Company in China with Adaboost. *Procedia Computer Science*, v. 17, p. 633–640, 2013. ISSN 18770509. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S1877050913002159⟩.

[35] KILIC, K. et al. Soft ground tunnel lithology classification using clustering-guided light gradient boosting machine. *Journal of Rock Mechanics and Geotechnical Engineering*, p. S1674775523000720, mar. 2023. ISSN 16747755. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S1674775523000720⟩.

[36] WANG, D.-n.; LI, L.; ZHAO, D. Corporate finance risk prediction based on LightGBM. *Information Sciences*, v. 602, p. 259–268, jul. 2022. ISSN 00200255. Disponível em: ⟨https://linkinghub.elsevier.com/retrieve/pii/S002002552200411X⟩.