Journal of Administrative and Economic Sciences • Vol 16, Issue 2 (2023) • p 99-115

Predictive Analytics for Startups Success: Acquisition Prediction based on Machine Learning Techniques

Alaeddine Mihoub

Department of Management Information Systems and Production Management

College of Business and Economics, Qassim University

P.O. Box: 6640, Buraidah: 51452 • Kingdom of Saudi Arabia

a.mihoub@qu.edu.sa

Abstract:

With the rise of the internet, startups number has substantially increased in the last two decades. Although many of them have succeeded to revolutionize their sectors, many of them have shut down a few months or even a few years after the foundation. In this paper, we propose a machine learning approach for predicting startups success based on historical data. Almost 840 startup data have been finely preprocessed to extract 35 features that characterize each a particular aspect of the studied startups. Afterward, computational models based on machine learning techniques were developed and tested using a cross-validation approach. The main objective is to predict the success of the startup, especially in terms of mergers and acquisitions. In particular, several models have been applied namely Artificial Neural Networks (a.k.a. ANNs), Support Vector Machines (a.k.a. SVMs), Random Forests, Bagging, Stacking, and Gradient Boosting. Overall results were very promising since the best model succeeded in the prediction with an accuracy rate of 85%. Furthermore, a feature importance study was also conducted to analyze the best predictors of a startup acquisition.

Keywords: Predictive Analytics, Startup Success, Machine Learning, Acquisition Prediction.

JEL Classification codes: C52; C53; C63; G34; M13.

1. Introduction

In recent years, startups have gained a lot of attention throughout the world, and their number is steadily increasing [1]. Startups are now being acknowledged as important drivers of economic development and job creation. Through innovative and scalable technologies, they may offer significant solutions and hence act as catalysts for socioeconomic growth and change [2].

However, not every startup is successful. Numerous studies indicate that nine out of ten companies fail [3]. There are several causes for such failures. Demand from the market is one of the factors. Numerous ventures center on a product for which demand has diminished over time or never existed. Other causes include a lack of financial resources over time, a lack of experience, bad timing, poor management, legal issues, and an ineffective team, among others [4].

Therefore, forecasting startup success is critical for both new businesses and venture capital (VC) organizations [5]. It represents a hot research topic that has attracted a lot of attention in recent years [6]. Indeed, for young businesses, anticipating their own and their rivals' future growth may be of great assistance in adjusting their development plans and successfully seizing chances. In addition, predicting the future performance of new businesses helps venture capital firms strike a better balance between their profits and their risks.

For these reasons, modern entrepreneurs must evaluate if their firm is on the correct path. Although there is no precise formula for startup success, there are two main types of companies that tend to do well. Having the business listed on a public stock market and giving shareholders the possibility of selling their shares to the broader public is one way to success. This is known as an "Initial Public Offering" (IPO). Another option is an acquisition by a larger company (a "merger or acquisition," or "M&A"), in which the founders and investors get immediate cash in return for their ownership stake. The term "exit strategy" is often used to describe these success types [7].

To assess the future of a startup, a large part of investors relies on their personal experience [8]. Other classic approaches rely mainly on a series of indicators or rule-based models [9]. However, in the past few years, machine learning has advanced tremendously and achieved considerable success in a variety of fields. [10]–[13]. Thus, in many recent studies [5], machine learning algorithms have been adopted as base models for predicting startups' success or failure.

In this paper, we develop several fine-tuned machine learning models for predicting the outcomes of a startup. In our work, a startup is considered successful if it concludes an M&A deal. To this end, we have used historical data of nearly 840 startups to extract relevant features for startup outcomes prediction. In addition, a feature importance study is carried out to evaluate the contribution of each feature to the company success.

The rest of this article is organized as follows: section 2 reviews the literature of prediction models for startup success. Section 3 introduces the theoretical background behind machine learning models. Our methodology is presented in section 4. Experiments are detailed in section 5. All results are discussed in section 6. Section 7 concludes the paper and proposes some perspectives for future work.

2. Related Work

The literature on startup success prediction can be reviewed from two different angles.

Firstly, related research can be analyzed from the type of model point of view. Initially, researchers were mainly relying on expert systems [14] as well as rule-based approaches [9]. But with the rise of Artificial Intelligence, more research was put into exploiting the advancement of machine learning methods.

For instance, authors in [3] presented a methodology for forecasting the success or failure of a startup based on a wide range of variables, including seed investment, series A funding, fundraising time, etc. Data was meticulously collected from different sources like Tech Crunch and Crunchbase. Different machine learning models were developed above this data such as Bayesian Networks and Random Forests. Overall precision rates were quite interesting ranging - depending on the model - from 73% to 96%.

Based also on machine learning models, the work presented in [15], developed an XGBoost model to predict IPO and M&A exits. The model's performance was very good when evaluated using a leave-one-out cross-validation approach. The recorded evaluation metrics were respectively 84% for the accuracy and 0.91 for the Area Under Curve metric (a.k.a. AUC).

Furthermore, hybrid models were also explored such as the approach presented in [16]. In the latter work, researchers have used a hybrid intelligence approach, which integrates the strengths of both humans and computer analysis, to foresee the long-term success of new businesses and the results were very promising. Indeed, the proposed Hybrid Intelligence approach has demonstrated great prediction performance, especially in the face of high levels of uncertainty.

Secondly, related work can be studied from the perspective of the predicted target variable. While some studies try to predict only the survival of the startup, other approaches have tried to predict the potential of an M&A deal. Furthermore, some studies have gone beyond M&A prediction and attempted to forecast the destiny of a firm by considering four potential issues: IPO, M&A, Remaining Private, and Failure.

For instance, for the survival prediction, authors in [17] analyzed more than 180,000 tweets from the Twitter accounts of 253 new businesses using context-related machine learning approaches. The results show that the created models were successful in discriminating between failing and successful enterprises in up to 76% of cases.

Merger and Acquisition prediction represent also a challenging topic in literature [18]. For example, authors in [1] proposed several machine learning classifiers such as Logistic Regression, Decision Trees, Gradient Boost, Random Forest, and Neural networks for M&A forecasting. The training data for these models comprise crucial characteristics like valuations, fundraising rounds, and investments, among others. As a result, the models were able to achieve an accuracy of about 92%.

Furthermore, with a more accurate target variable, authors in [5] have used a large data set of Crunchbase startup enterprises and developed a machine learning-based model referred to as "CapitalVX" to predict startup outcomes, such as an IPO, M&A, failure,

or staying private. Using a wide feature set, the accuracy of the model was estimated to be between 80 and 89%. According to this study, these computational approaches may be very beneficial for the company's stakeholders as well as prospective investors.

Seeing previous work in literature, we propose in this paper a carefully designed approach that tries to investigate the best machine learning models for startup success prediction. In addition, a preprocessing step was deeply undertaken to smartly extract the best features that explain the company's success and optimize models' outputs. Moreover, several evaluation metrics were finely chosen to assess the proposed models from different perspectives. In the next section, we review some related theoretical principles to our prediction problem.

3. Theoretical Background

In this section, we will go over some scientific concepts that are related to our challenge especially prediction techniques as well as classification and regression problems.

3.1 Prediction techniques

Given a set of available variables (also known as explanatory variables, input variables, independent variables, etc.), the objective of prediction is to estimate a variable of interest (a.k.a. explained variable, output variable, dependent variable, target variable, etc.). This predicted variable can relate to a future event such as the success/failure of a company or also a real number such as the probability of an IPO or an M&A deal.

Practically, recent prediction methods are increasingly related to machine learning techniques thanks to the huge developments experienced in this field. Machine learning was previously defined by Arthur Samuel (an AI pioneer in the 1950s) as "the field of study that gives computers the ability to learn without explicitly being programmed". In case we dispose of both input and target variables, we are typically in the special context of supervised learning [19] (cf. Figure 1).

Indeed, in a supervised machine learning context, we dispose of two steps as shown in Figure 1: the training step and the testing step. In the training step, the data is collected and processed to serve as training data (i.e. input data) to a machine learning model. The more data a program has, the better it is. Once the model is trained and optimal parameters were found, we can proceed to the testing step where the model is tested on unseen data. Afterward, results must be compared to the ground-truth data to evaluate the performance of the model. After testing and validation, the model can be deployed for real applications. Next, we will discuss the two main types of prediction models in the context of supervised learning, namely classifiers and regressors.

3.2 Classification and Regression

In a classifier [19], the predicted variable can be either a binary variable (0 or 1) or a multi-class variable (when there are more than two classes). An example of a classifier whose target variable is binary ("Two-class classification") consists in estimating whether a company will succeed or fail. An example of a classifier with several

categories ("Multi-class classification") consists in predicting whether a company will end up with an IPO or M&A or Private or Failure. In a regression, the target variable is a continuous variable (cf. Figure 1) and the prediction model is called a Regressor.

An example of regression consists in estimating the time that a company should survive under certain conditions or also the probability of failure within a certain duration. The most classic and popular model is linear regression [20].

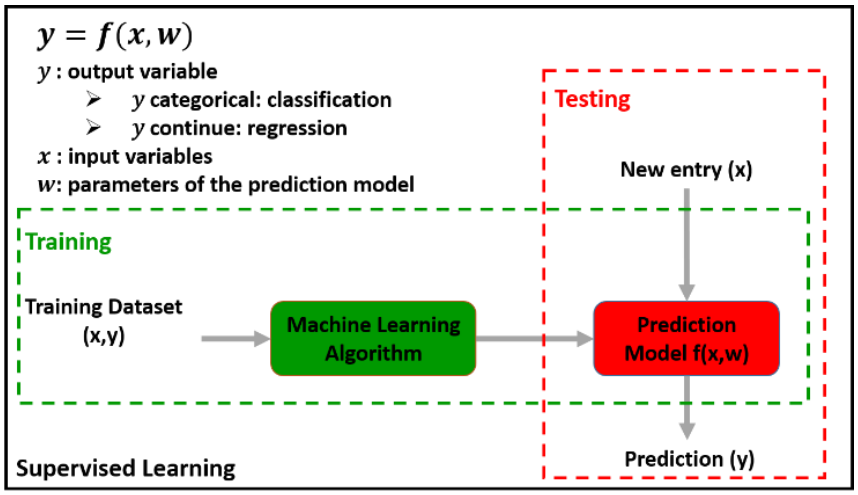

Figure 1: Overview of the Supervised Learning.

4. Approach

In this section, we start with an overview of our approach followed by a brief description of machine learning models as well as their assessment metrics.

4.1 Overview

The challenge of predicting the future of a startup can be seen as a classification problem. In our particular case, it can be considered as a binary classification since we have two possible output classes which are success or failure. The problem can be formulated as follows: each startup i (where $i = 1..n$, and n is the total number of startups) dispose of a set of features $(x_{ij})_{j=1..m}$ (where $x_i \in R^m$, and m is the total number of features) from which we can predict y_i that represents the output class of that startup. The prediction y_i for each startup i can be further detailed as follows:

$$y_i = \begin{cases} 1: suceess \\ 0: failure \end{cases}$$

This way, the whole dataset can be seen as a set of $(x_i, y_i)_{i=1..n}$ where n is the total number of startups. The role of the machine learning model - as described in Figure 2 - is to map the set of features $(x_{ij})_{j=1..m}$ to the right output y_i. In the next paragraph we briefly review the main machine learning models that were carefully selected to respond to our classification challenge.

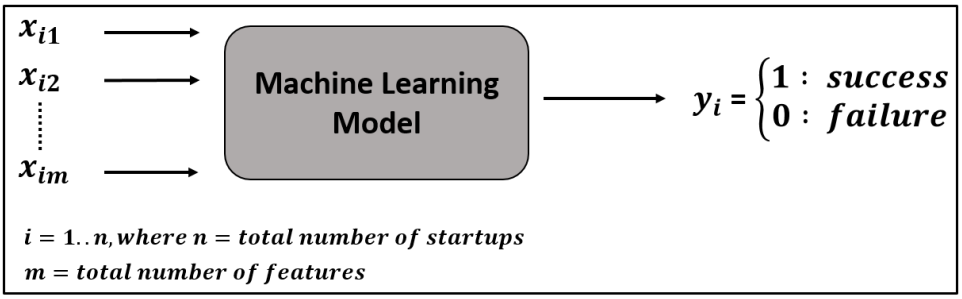

Figure 2: Proposed approach based on machine learning models.

4.2 Machine Learning Models

In our work, several machine learning models were deeply investigated and six models were finally selected due to their suitability to the related application as well as their high performance demonstrated in many previous works in literature [10], [13], [21]. These models are Artificial Neurons Networks, Support Vector Machines (SVM), Random Forests, Bagging, Stacking, and Gradient Boosting**.**

4.2.1 Artificial Neurons Networks (ANNs)

ANNs [22] are computer models inspired by the human brain that are made up of numerous linked and consecutive layers. A single layer is made up of a collection of artificial cells known as nodes. Those nodes are linked to the successive layer by connections that show their influence on the next connected layer node. As shown in Figure 3, the initial layer is known as the input layer because it pushes data into the network. The middle levels are known as hidden layers, while the final layer is known as the output layer. ANNs' basic architectures are also referred to as Feed-Forward Networks or Multi-Layer Perceptrons. More advanced topologies can be investigated in the context of deep learning models [23].

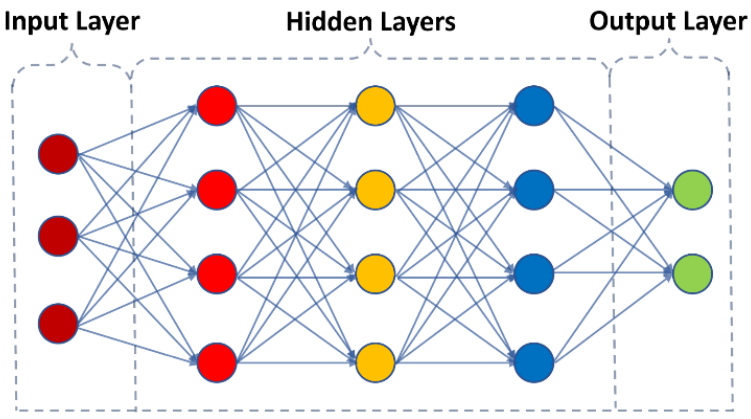

Figure 3: The topology of ANNs [21].

4.2.2 Support Vector Machines (SVMs)

Support vector machines (a.k.a. SVMs) have been widely and effectively utilized to tackle regression and classification issues [24]. This powerful technique is founded on two basic ideas that Vapnik [25] explicitly united in 1995. The maximum-margin hyperplane concept is the first (cf. Figure 4). Its goal is to determine the best hyper-plane that divides the classes by the greatest margin. If the data is linearly distinct, this is a standard quadratic-optimization issue. Nonetheless, data are typically not linearly separable. The second important concept, represented by the kernel function, provides the answer by changing the input space to a larger-dimensional space, in which a linear separator is highly expected to be found. In this manner, the SVM paradigm effectively solves the nonlinear classification issues.

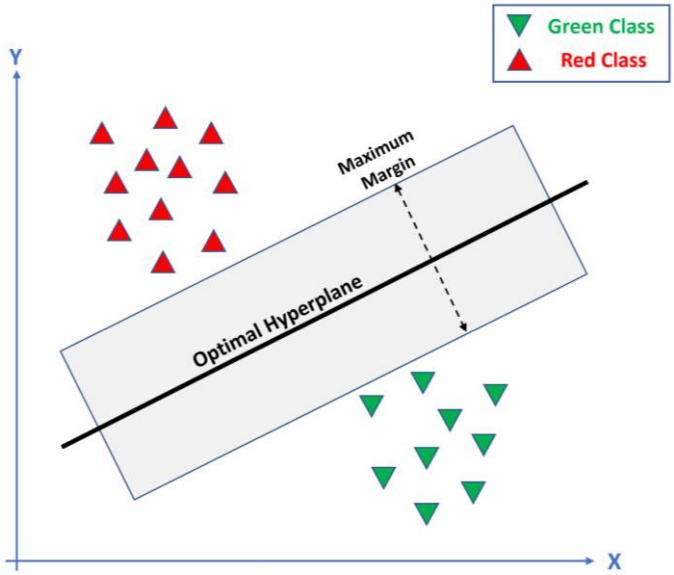

Figure 4: Overview of the SVM algorithm.

4.2.3 Random Forests

Random Forests (a.k.a. RF) [26] is also an extensively used prediction approach. Because it generates a list of decision trees [27] and integrates their numerous classification judgments using a voting method (as shown in Figure 5), it is considered as part of the Ensemble Learning techniques [28]. It is an attractive technique since it preserves the benefits of decision trees while avoiding over-fitting concerns. In comparison with traditional decision tree models, RF produces more consistent and accurate recognition outputs.

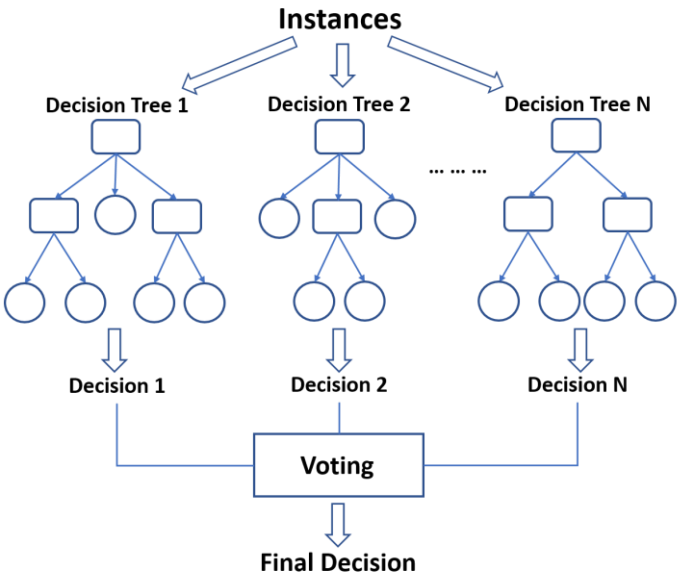

Figure 5: Overview of the Random Forests approach [21].

4.2.4 Bagging

Bootstrap aggregation, often known as Bagging, is a special aggregation of many estimators or classifiers (as shown in Figure 6). It is considered a type of ensemble methods. Actually, bagging produces a large number of estimators and every single estimator is considered to be a poor one. But, when the numerous estimators are joined, they build together powerful models. Indeed, many estimators are permitted to fit the training inputs and by employing the ensemble of learned classifiers, any bias may be handled, which ends in efficient classification scores. This approach is also suitable for unbalanced data since it reduces variance and thereby overfitting issues [29].

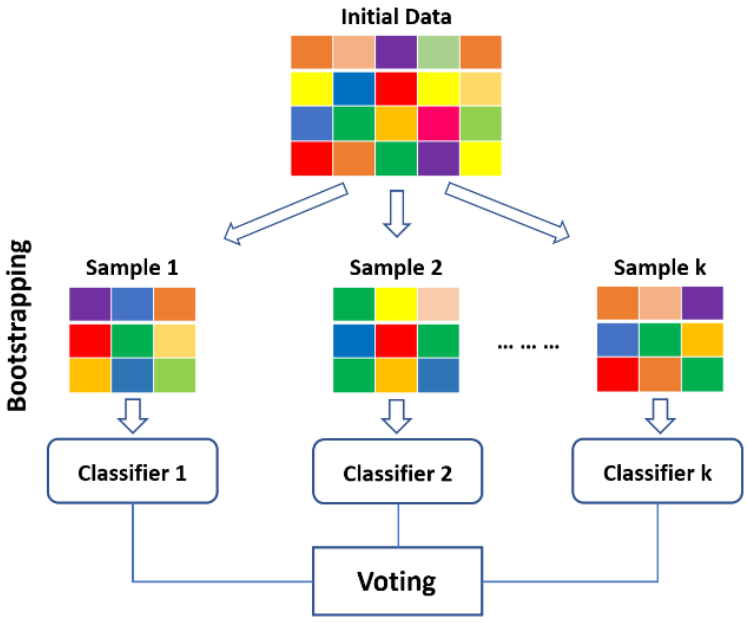

Figure 6: Overview of the Bagging approach [21].

4.2.5 Stacking

Stacking [30] refers to the technique of training a machine learning model using the results of numerous different learning algorithms as inputs. Once all input models have been trained using the available data, the combiner algorithm is taught to provide a final prediction utilizing all of the predictions of the other models as supplementary information. Although a logistic regression classifier is often used as the combiner, stacking may use any other effective algorithm.

4.2.6 Gradient Boosting

Gradient Boosting is a machine learning approach that generates a prediction tool from a set of relatively weak classifiers, most often decision trees [31]. It is primarily dependent on the boosting strategy. Fitting the data to a beginning model is the first step in the boosting approach. Next, a secondary model is built that focuses on correctly forecasting events where the first model fails. It is projected that the combination of these two models will outperform any model alone. This method is then repeated multiple times. Every succeeding model attempts to solve the shortcomings of the enhanced ensemble of all previous models. Briefly, gradient boosting is a kind of machine learning boosting. In the next subsection, we provide the primary metrics used to assess the aforementioned classification techniques.

4.3 Classification Metrics

In this subsection, we suppose a binary classification problem (Positive/Negative). The first essential tool for a data scientist to asses a binary classifier is the confusion matrix. This matrix has the advantage to measure the quality of a classifier in a simple and effective manner as shown in Table 1. For a binary classifier, it so composed basically of four statistics which are TP, FN, FP, and TN. The letter P refers to Positive, N refers to Negative, TP to True Positive, FN to False Negative, FP to False Positive, and TN to True Negative.

Table 1: Confusion matrix.

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

From the confusion matrix, four metrics can be easily computed which are accuracy, precision, recall, and F1-score. The accuracy represents the ratio of samples that were correctly classified (cf. formula 1). For a specific class, the precision rate represents the ratio of accurately classified instances from all detected ones in that class (cf. formula 2). For the recall rate of a specific class, it represents the ratio of accurately classified

instances from all existing ones in that class (cf. formula 3). Since precision and recall are inversely related, a better approach is to compute the harmonic mean of these two metrics as presented in formula 4 which represents the F1-score.

In addition to the last four metrics, another popular approach is widely used to visually evaluate the performance of a binary classifier. This approach is the Receiver Operating Characteristic also known as the ROC curve [32]. Graphically, the ROC curve is represented by plotting the rate of true positives ("TPR: True Positive Ratio") as a function of the rate of false positives ("FPR: False Positive Ratio"). The more the curve moves away upwards from the line y=x (which represents chance), the better the model performs. Beyond the visual comparison between two models, there is a metric called AUC ("Area Under the Curve") which represents the area under the ROC curve. When AUC=1, the model is perfect while when AUC=0.5, the model is identical to chance (line y=x). For information, the metrics that were presented (accuracy, recall, precision, etc.) can also be calculated for a multi-class classifier. In the next section, we present the computed experiments related to our proposed approach.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1-score = 2 \times \frac{precision \times recall}{precision+recall} \quad (4)$$

5. Experiments

5.1 Dataset

Previous work has collected data from many sources such as LinkedIn [33], Crunchbase [34], [35], Tech Crunch [3], and Twitter [17]. In our work, we adopted a Kaggle dataset that contains almost 840 startup information. Please note that in this dataset, a startup is considered successful if it concludes an M&A deal. Initially, the dataset provided a set of 48 features that describe each, a particular aspect of the company [36]. But before any modeling, we applied a preprocessing step to prepare the dataset for the machine learning models [37]. This step is recommended and even mandatory for several classifiers. For instance, preprocessing operations involved replacing missing values, deleting redundant or unnecessary information, removing noise such as incorrect values, etc. These operations are important to attenuate the bias from the learning process. As a result, the newly prepared dataset included nearly 35 features that describe the following information:

- Age and location of the company.

- Activity sector (software, e-commerce, biotech, etc.).

- Information about the timing of fundraising.

- Information about the fundraising rounds (seed, round A, B, C, etc.).

- Information about the total fundraising amounts.

- Information about fundraising participants (venture capitalists, business angels, etc.).

- Relationships in the market with other companies.

- Information about the achievements of the company.

As previously mentioned, these features will be used in conjunction with the already-known fate of the startups to train the selected machine learning models. The next subsection provides further information on technical and implementation details.

5.2 Implementation

Practically, our experiments were carried out using a PC outfitted with an Intel Core (i7-8550) processor and 16GB of RAM. Preprocessing operations, the training of the models, their empirical testing, and assessment metrics, were all implemented with Python and well-known data science libraries such as Pandas, LightGBM, and Scikit-learn. Moreover, a 5-cross validation strategy was used to limit overfitting concerns and to ensure model consistency. Furthermore, for all algorithms, we tested many hyperparameters to find optimal performance. For instance, for ANNs, we adopted an architecture with two hidden layers enclosing each, 70 nodes. For SVM, the C parameter (a.k.a. the regularization parameter) was set to 100. The optimal number of estimators was found to be 100 for Random Forests and 50 for the Bagging Approach. For stacking, SVM et Decision Trees were chosen as initial estimators while Logistic Regression was adopted as the combiner classifier. For Gradient Boosting, we adopted the implementation of the LightGBM library which takes the Decision Tree as a base model. The results of all these models are discussed in the next section.

6. Results and Discussion

We remind that six machine learning models were finely selected and adapted to our prediction task. These models are Artificial Neural Networks, Support Vector Machines, Random Forests, Bagging, Stacking, and Gradient Boosting. To evaluate these models, a 5 cross-validation methodology is applied and many metrics were computed. These metrics are mainly accuracy, F1-score, AUC as well as training and testing time. These statistics are shown respectively in Table 2, Table 3, Table 4, and Table 5. In addition, the first three metrics are summarized in Figure 7.

Table 2: Accuracy of all models.

Classifier	Accuracy (Standard Deviation)
ANNs	82.96 (±1.86)
SVMs	82.95 (±2.38)
Random Forests	82.96 (±2.56)
Bagging	81.65 (±2.42)
Stacking	84.98 (±2.29)
Gradient Boosting	**85.46** (±2.73)

Table 3: F1-score of all models.

Classifier	F1-score (Standard Deviation)
ANNs	82.57 (±1.81)
SVMs	82.57 (±2.34)
Random Forests	82.35 (±2.57)
Bagging	80.90 (±2.64)
Stacking	84.55 (±2.30)
Gradient Boosting	**85.19** (±2.83)

Table 4: AUC of all models.

Classifier	AUC (Standard Deviation)
ANNs	86.49 (±0.90)
SVMs	87.40 (±1.96)
Random Forests	87.33 (±2.95)
Bagging	86.70 (±3.20)
Stacking	88.12 (±3.22)
Gradient Boosting	**89.08** (±2.48)

Table 5: Training and Testing times for all models.

Classifier	Training Time (sec)	Testing Time (sec)
ANNs	2.214	0.004
SVMs	1.012	0.049
Random Forests	1.214	0.082
Bagging	11.012	1.209
Stacking	2.318	0.043
Gradient Boosting	**0.791**	0.008

The statistics shown in Table 2, Table 3, and Table 4 represent the average result of each classifier, followed by their standard deviation, all computed from cross-validation iterations.

The first evaluation metric is accuracy (cf. Table 2). The best accuracy rates were outputted by the gradient boosting algorithm (85.46%) followed by the stacking algorithm (84.98%). The least performant algorithm was the bagging algorithm with an accuracy rate of 81.65%. The rest of the classifiers (i.e. ANNs, SVMs et random forests) have recorded accuracy rates of around 82.96%. The great performance of the gradient boosting algorithm can be explained by two main reasons.

First, the gradient boosting approach, since it is part of ensemble methods, has the advantage to combine the decisions of many sequential classifiers. Second, our gradient boosting algorithm was developed using the LightGBM [38] library which provides many benefits that enhance accuracy compared to other implementations [39]. Indeed, LightGBM is a gradient-boosting framework that relies on decision tree-boosting algorithms. It divides the tree according to its leaves, whereas other boosting algorithms

divide the tree according to its depth or levels. This leaf-wise strategy in Light GBM reduces more loss than the level-wise strategy, ending in a significantly higher degree of accuracy that is seldom obtained by any of the current boosting techniques [39].

Furthermore, compared to other approaches, LightGBM can train faster because it employs a novel approach known as histogram-based optimization [39] that minimizes the amount of data necessary to build each tree. This is what explains the great running times of gradient boosting compared to other models (as shown in Table 5). While the training time of the rest of the models ranges between 1.012 and 11.012 (sec), the recorded training time of the LightGBM approach has not exceeded 0.791 (sec).

Similar results were recorded for F1-score and AUC metrics as shown in Table 3 and, Table 4. For the F1-score, the best performances were produced respectively by gradient boosting (85.19%) and stacking (84.55%). Other classifiers' rates ranged between 80.90% and 82.57%. Comparable outcomes were also observed for the AUC metric. The highest AUC values were achieved by the gradient boosting algorithm (89.08%) and the stacking algorithm (88.12%). To synthesize our findings, these great performance rates, especially for gradient boosting and stacking, confirm the relevance of our developed approach in predicting startup outcomes.

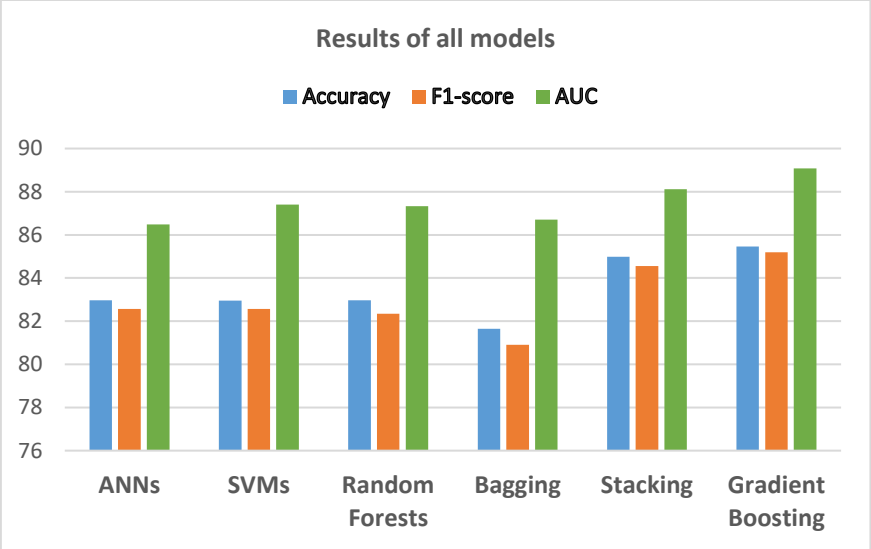

Figure 7: Accuracy, F1-score, and AUC metrics for all models.

Since gradient boosting implementation using LightGBM has generated the best results, this library was also used to analyze feature importance in predicting startup success. Indeed, feature importance reveals how much each input variable adds to the model's prediction. Essentially, it assesses the degree to which a certain variable is beneficial for the present forecasting model. In our case, the feature importance study has revealed important weights in the prediction model for specific variables such as the age of the startup, the total amount of raised funds, fundraising timing, the achievements timing of the startup, as well as their relationships in the surrounding ecosystem. For example, the greater the quantity of money raised, the more likely the

business will succeed, and vice versa. Another example is that the more links the firm has with other partners, the more likely the company is to prosper.

To conclude, we do not claim any direct causation, however, all these identified factors may be seen as highly potential variables that should be extensively explored when building strong predictive models for startup success [40].

7. Conclusion

In this study, we proposed a special approach for forecasting the success of startups using machine learning and historical data. The data on over 840 businesses have been meticulously preprocessed to derive 35 attributes that each define a distinct component of the startups under study. Several computational models based on machine learning techniques were constructed and evaluated using a cross-validation strategy. The primary purpose is to forecast the startup's performance, particularly in terms of mergers and acquisitions. Various models, specifically Artificial Neural Networks, Support Vector Machines, Random Forests, Bagging, Stacking, and Gradient Boosting, have been utilized. Overall, the findings were quite interesting, as the top model (gradient boosting-based algorithm) performed great predictions with respective rates of 85.46%, 85.19%, and, 89.08% for accuracy, F1-score, and, AUC. In addition, a feature importance study was conducted and revealed important scores for attributes related to age, fundraising, achievements, and, startup professional network. In future work, we intend to investigate larger datasets in order to enhance the learning process and thus enhance prediction outcomes. Additionally, with larger data sets, Deep Learning models may be an appealing alternative to traditional models due to their superior performance in a variety of disciplines. Furthermore, we intend to explore local startup data, especially data from the gulf and the middle east ecosystems.

References

[1] M. Bangdiwala, Y. Mehta, S. Agrawal, and S. Ghane, *Predicting Success Rate of Startups using Machine Learning Algorithms*. 2022. doi: 10.1109/ASIANCON55314.2022.9908921.

[2] S. L. Chaudhari and M. Sinha, “A study on emerging trends in Indian startup ecosystem: big data, crowd funding, shared economy,” *International Journal of Innovation Science*, vol. 13, no. 1, pp. 1–16, Jan. 2021, doi: 10.1108/IJIS-09-2020-0156.

[3] A. Krishna, A. Agrawal, and A. Choudhary, “Predicting the Outcome of Startups: Less Failure, More Success,” in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Barcelona, Spain, Dec. 2016, pp. 798–805. doi: 10.1109/ICDMW.2016.0118.

[4] M. Cantamessa, V. Gatteschi, G. Perboli, and M. Rosano, “Startups’ Roads to Failure,” *Sustainability*, vol. 10, no. 7, Art. no. 7, Jul. 2018, doi: 10.3390/su10072346.

[5] G. Ross, S. Das, D. Sciro, and H. Raza, “CapitalVX: A machine learning model for startup selection and exit prediction,” *The Journal of Finance and Data Science*, vol. 7, pp. 94–114, Nov. 2021, doi: 10.1016/j.jfds.2021.04.001.

[6] C. Ünal, “Searching for a Unicorn: A Machine Learning Approach Towards Startup Success Prediction,” masterThesis, Humboldt-Universität zu Berlin, 2019. doi: 10.18452/20347.

[7] A. Pisoni and A. Onetti, “When startups exit: comparing strategies in Europe and the USA,” *Journal of Business Strategy*, vol. 39, no. 3, pp. 26–33, Jan. 2018, doi: 10.1108/JBS-02-2017-0022.

[8] D. Yin, J. Li, and G. Wu, “Solving the Data Sparsity Problem in Predicting the Success of the Startups with Machine Learning Methods.” arXiv, Dec. 15, 2021. Accessed: Nov. 03, 2022. [Online]. Available: http://arxiv.org/abs/2112.07985

[9] B. Yankov, P. Ruskov, and K. Haralampiev, “Models and Tools for Technology Start-Up Companies Success Analysis,” *Economic Alternatives*, no. 3, p. 10, 2014.

[10] A. Mihoub, H. Snoun, M. Krichen, R. B. H. Salah, and M. Kahia, “Predicting COVID-19 Spread Level using Socio- Economic Indicators and Machine Learning Techniques,” in *2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, Nov. 2020, pp. 128–133. doi: 10.1109/SMART-TECH49988.2020.00041.

[11] A. Mihoub, O. B. Fredj, O. Cheikhrouhou, A. Derhab, and M. Krichen, “Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques,” *Computers & Electrical Engineering*, vol. 98, p. 107716, Mar. 2022, doi: 10.1016/j.compeleceng.2022.107716.

[12] A. Mihoub, “A Deep Learning-Based Framework for Human Activity Recognition in Smart Homes,” *Mobile Information Systems*, vol. 2021, Sep. 2021, doi: 10.1155/2021/6961343.

[13] S. M. Qaisar, A. Mihoub, M. Krichen, and H. Nisar, “Multirate Processing with Selective Subbands and Machine Learning for Efficient Arrhythmia Classification,” *Sensors*, vol. 21, no. 4, Art. no. 4, Jan. 2021, doi: 10.3390/s21041511.

[14] S. Ragothaman, B. Naik, and K. Ramakrishnan, “Predicting Corporate Acquisitions: An Application of Uncertain Reasoning Using Rule Induction,” *Information Systems Frontiers*, vol. 5, no. 4, pp. 401–412, Dec. 2003, doi: 10.1023/B:ISFI.0000005653.53641.b3.

[15] A. N. Thirupathi, T. Alhanai, and M. M. Ghassemi, “A machine learning approach to detect early signs of startup success,” in *Proceedings of the Second ACM International Conference on AI in Finance*, Virtual Event, Nov. 2021, pp. 1–8. doi: 10.1145/3490354.3494374.

[16] D. Dellermann, N. Lipusch, P. Ebel, K. M. Popp, and J. M. Leimeister, “Finding the Unicorn: Predicting Early Stage Startup Success Through a Hybrid Intelligence Method.” Rochester, NY, Dec. 10, 2017. doi: 10.2139/ssrn.3159123.

[17] T. Antretter, I. Blohm, D. Grichnik, and J. Wincent, "Predicting new venture survival: A Twitter-based machine learning approach to measuring online legitimacy," *Journal of Business Venturing Insights*, vol. 11, p. e00109, Jun. 2019, doi: 10.1016/j.jbvi.2018.e00109.

[18] G. Xiang, Z. Zheng, M. Wen, J. Hong, C. Rose, and C. Liu, "A Supervised Approach to Predict Company Acquisition with Factual and Topic Features Using Profiles and News Articles on TechCrunch," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 6, no. 1, Art. no. 1, 2012, doi: 10.1609/icwsm.v6i1.14306.

[19] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," in *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, Amsterdam, The Netherlands, The Netherlands, 2007, pp. 3–24. Accessed: Feb. 25, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?id=1566770.1566773

[20] İ. Uysal and H. A. Güvenir, "An overview of regression techniques for knowledge discovery," *The Knowledge Engineering Review*, vol. 14, no. 4, pp. 319–340, Dec. 1999, doi: 10.1017/S026988899900404X.

[21] S. Zidi, A. Mihoub, S. Mian Qaisar, M. Krichen, and Q. Abu Al-Haija, "Theft detection dataset for benchmarking and machine learning based classification in a smart grid environment," *Journal of King Saud University - Computer and Information Sciences*, May 2022, doi: 10.1016/j.jksuci.2022.05.007.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[23] M. Z. Alom *et al.*, "A State-of-the-Art Survey on Deep Learning Theory and Architectures," *Electronics*, vol. 8, no. 3, Art. no. 3, Mar. 2019, doi: 10.3390/electronics8030292.

[24] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004, doi: 10.1023/B:STCO.0000035301.49549.88.

[25] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[26] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[27] S. B. Kotsiantis, "Decision trees: a recent overview," *Artif Intell Rev*, vol. 39, no. 4, pp. 261–283, Jun. 2011, doi: 10.1007/s10462-011-9272-4.

[28] R. Polikar, "Ensemble learning," *Scholarpedia*, vol. 4, no. 1, p. 2776, 2009, doi: 10.4249/scholarpedia.2776.

[29] B. Ghojogh and M. Crowley, *The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial*. 2019.

[30] S.-A. Alexandropoulos, C. Aridas, S. Kotsiantis, and M. Vrahatis, “Stacking Strong Ensembles of Classifiers,” 2019, pp. 545–556. doi: 10.1007/978-3-030-19823-7_46.

[31] T. Hastie, R. Tibshirani, and J. Friedman, “Boosting and Additive Trees,” in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, T. Hastie, R. Tibshirani, and J. Friedman, Eds. New York, NY: Springer, 2009, pp. 337–387. doi: 10.1007/978-0-387-84858-7_10.

[32] D. M. Powers, “Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation,” Dec. 2011, Accessed: Sep. 04, 2019. [Online]. Available: https://dspace.flinders.edu.au/xmlui/handle/2328/27165

[33] B. Sharchilev, M. Roizner, A. Rumyantsev, D. Ozornin, P. Serdyukov, and M. de Rijke, “Web-based Startup Success Prediction,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, New York, NY, USA, 2018, pp. 2283–2291. doi: 10.1145/3269206.3272011.

[34] K. Żbikowski and P. Antosiuk, “A machine learning, bias-free approach for predicting business success using Crunchbase data,” *Information Processing & Management*, vol. 58, no. 4, p. 102555, Jul. 2021, doi: 10.1016/j.ipm.2021.102555.

[35] J. Arroyo, F. Corea, G. Jimenez-Diaz, and J. A. Recio-Garcia, “Assessment of Machine Learning Performance for Decision Support in Venture Capital Investments,” *IEEE Access*, vol. 7, pp. 124233–124243, 2019, doi: 10.1109/ACCESS.2019.2938659.

[36] “Startup Success Prediction.” https://www.kaggle.com/search?q=startup+success+in%3Adatasets+sortBy%3Adate (accessed Dec. 04, 2022).

[37] B. Ghojogh *et al.*, “Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review.” arXiv, May 07, 2019. doi: 10.48550/arXiv.1905.02845.

[38] “Welcome to LightGBM’s documentation! — LightGBM 3.3.3.99 documentation.” https://lightgbm.readthedocs.io/en/latest/index.html (accessed Nov. 15, 2022).

[39] G. Ke *et al.*, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” in *Advances in Neural Information Processing Systems*, 2017, vol. 30. Accessed: Nov. 15, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[40] A. Hjerpe, “Computing Random Forests Variable Importance Measures (VIM) on Mixed Continuous and Categorical Data,” 2016.