

# Startups Success Prediction

## Using Ensemble Classification

MD. ABU AMMAR – 1821944642

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh

[abu.ammar@northsouth.edu](mailto:abu.ammar@northsouth.edu)

Sadia Afrin Tamanna - 1812030042

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh

[sadia.tamanna@northsouth.edu](mailto:sadia.tamanna@northsouth.edu)

Mostak Ahamed – 1821737042

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh

[ahamed.mostak@northsouth.edu](mailto:ahamed.mostak@northsouth.edu)

Tamalika Bakshi - 1812469042

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh

[tamalika.bakshi@northsouth.edu](mailto:tamalika.bakshi@northsouth.edu)

**Abstract:** In this era, we are determined to find the business performance whether it grows up well or not. The importance of startups for a dynamic, innovative, and competitive economy has already been acknowledged within the scientific and business literature. Especially for entrepreneurship, it is important to know if the startup would be a success or a failure. The purpose of this paper is to analyze startup behavior based on several variables, determine what variables affect startup success the most, then build a model that can predict the success of a startup using machine learning algorithms. The algorithm proposed in this paper will help to predict the success of a startup based on different types of attributes discussed in the dataset. A variety of methods will be used to determine the best model such as Adaboost classifier, RandomForest classifier, LGBM classifier, Decision Tree, SVM classifier, Extra Tree classifier, and voting classifier from Ensemble learning.

### 1 Introduction

Startup is a business that has just been established and grown with the support of digital services and has also become an important element of innovation systems and economies around the world. The startup ecosystem is growing rapidly and still needs a lot of funding to operate with a minimalist working group. So, it is very important for the VC to monitor the performance and performance of a startup so that it can be used as a consideration to decide whether to fund a startup to drive its growth or refuse to take part in the funding. To monitor startup performance, it is important to analyze what makes a startup successful and how to determine its success.

The problem will be solved through a Supervised Machine Learning approach by training a model based on the history of startups which were either acquired or closed. The trained model will then be used to make predictions on startups which are currently operating to determine their success/failure.

Their rapid growth rates, legerity in developing innovative business models, and state-of-the-art technologies, along with their fail fast and learn

management approach turn them into unquiet factors within the international economy, particularly since their business playground is frequently a world on. Because of the global representation, the success of startups prediction is not only beneficial for entrepreneurs but also provides huge advantages to companies behind those other stakeholders such as investors, shareholders, suppliers, and customers/clients.

## 2 Literature Review

The startup success prediction model is an essential tool for business vendors to make decisions based on several aspects. Because of the study of multiple links among data, predicting startup success is a tough process. The social media dataset is also used in the research to improve the prediction's performance. To improve prediction performance and analyze relevant parameters, several models were applied to various data sets.

Cacciollattia et al. [1] created a framework for analyzing strategic alliances and predicting startup performance. To evaluate the performance of the devised technique, 3913 UK high-tech firms engaged in social innovation were contacted. The study finds that scalability is required for companies to achieve both performance and social purpose goals. According to the findings, startups, unlike large corporations, operate in a unique manner and type of alliance that is not appropriate for company development. The feature selection approach may be used to analyze the aspects that contribute to a startup's success.

Balboni, et al. [2] created a business model to examine the major conceptual improvement of research and explanation of development processes in the sphere of technology and science. The model for the prediction procedure looked at the industry structure and marketing dynamics. Entrepreneurial variables, contextual factors, and strategic factors have all been utilized to study startup performance prediction. According to the report, new businesses must adapt their operations to the environment while also addressing internal consistency issues. The elements are analyzed using a random forest classifier to

efficiently evaluate the link between the components and a solution for the vendors.

Guzmán, et al. [3] used the Lussier model to examine the success and failure characteristics of 303 business data records. The information was gathered through a personal interview, and the factors were analyzed using machine learning techniques. According to logistic regression, there are three elements that influence startup performance. Professional guidance, attracting and retaining personnel, and partnering with success are the three elements. The created approach has a prediction accuracy of 66 percent, according to the experimental study. When examining a large number of components, logistic regression performs poorly.

## 3 Methodology

Our primary objective is to develop a model that can predict success of a startup company on any industry or any category it belongs. Since there are plenty of reason for a startup to be succeed there need to have a very robust method that can predict the success without any doubt several machine learning algorithm are very robust to the plenty of information or feature that will help to build a model that can help entrepreneur, businessman, and investor to invest their time, money and effort on it with a certain amount of relief with help of success prediction of a particular startup company.

Our research will be conduct as visualize on the fig(1) to build an effective model for startup success prediction. Initially we will visualize the raw data and it will pass through data preprocessing (describe in section 3) then the dataset will be split into features and lable(0 or 1binary classification), there will conduct two types of validation technique reason is mentioned in section 3.3 from where we will select the best model for Ensemble Classification and with Stratified K-fold validation technique we will validate our final model.

*Figure 1: Flow Chart of Research Methodology*

### 3.1 Dataset Acquisition and Description:

GMO obtained data from other vendors on a trial basis. The data includes industry trends, investment insights,

and individual company data. Since the data was acquired on a trial basis, it only contains information from companies known until 2012. For this project, only data from companies operating between 2005 and 2012 were used. Train/test data includes all companies

acquired / closed (using a label) within that period. After training the model, we predict whether startups working in the same time frame will be acquired/closed. The dataset 923 rows and 49 columns

Table 1: Some Important Attributes of Dataset

Feature Name	Description	Data Type
age_first_funding_year	Age of the company in years since it got its first funding.	float
age_last_funding_year	Age of the company in years since it got last funding.	float
relationships	It says how many relationships a startup has. For example a start up can have relationships with accountants, investors, vendors, mentors, etc.	int
funding_rounds	The rounds of funding that startups go through to raise capital.	int
funding_total_usd	The amount of total funding throughout the years.	int

milestones	It points in time along the company's timeline prior to the number of future events or goals.	int
age_first_milestone_year	The number of years science it reached its first milestone.	float
age_last_milestone_year	The number of years science it reached its last milestone.	float
state	Name of the state it belongs to.	object
industry_type	Type of the industry in which the startup belongs to.	object
has_VC	If the startup has any Venture Capital(VC) or not.	int
has_angel	If it has an angel investor (also known as a private investor, seed investor or angel funder) or not.	int
has_roundA	If the startup goes through round A or not.	int

has_roundB	If the startup goes through round B or not.	int
has_roundC	If the startup goes through round C or not.	int
has_roundD	If the startup goes through round D or not.	int
avg_participants	The number of average participants.	float
is_top500	Whether the startup belongs to the top 500 or not.	int
status(acquired/closed)	Tells about the startup success, whether it acquired its goal or got closed.	object

### 3.2. Data pre-processing

Pre-processing has been processed so that the dataset becomes more conducive for the machine learning algorithms to be applied on them. We can observe in the dataset that there are 49 features with several kinds of data type where some of them are categorical. If we need to include them into features, we have to convert them into numeric to fit it properly in machine learning algorithms. Fortunately, all the categorical data of this data set are irrelevant in respect of the feature so we will drop them from our final data frame that will be used to train ML algorithms. The next step we have searched for the duplicate entries it shows up there are no duplicate entries. Furthermore, we have analyzed that there exist several missing entries in 5 columns out of 49 columns.

	Null Values	% Missing Values
Unnamed: 6	493	53.412784
closed_at	588	63.705309
age_first_milestone_year	152	16.468039
age_last_milestone_year	152	16.468039
state_code.1	1	0.108342

Figure 2

The analysis report shows that total missing values are 1386.

#### 3.2.1 Handling Missing Values

We have further analysed these 5 columns and found that column "Unnamed: 6" is a column of information from a combination of column "city", "state\_code", and "zip\_code". So to handle the missing value of column "Unnamed: 6", we decided to remove the contents of the column Unnamed: 6 first then fill in the data, based on a combination of "city", "state\_code", and "zip\_code" columns.

Column "closed\_at" is a column where startup "Closed" so that the empty data should be a startup whose status is still "Acquired". To handle this, we fill the last date "31/12/2013" to all the missing entries.

Based on the results of the analysis obtained, the columns 'age\_first\_milestone\_year' and 'age\_last\_milestone\_year' have null values because the startup does not have milestones. This can be confirmed by looking at the 'milestones' column containing the data 0 must be accompanied by the null 'age\_first\_milestone\_year' and 'age\_last\_milestone\_year' columns. so we decided to fill that null column with a value of 0.

Column "state\_code.1" has missing value in line 515, the "state\_code" column and the "state\_code.1" column must be the same, so the "state\_code.1" column must be dropped.

We left zero missing values in the dataset after handling the missing values.

#### 3.2.2 Outliers Detection

We have selected some important numeric features ('age\_first\_funding\_year', 'age\_last\_funding\_year', 'age

'\_first\_milestone\_year', 'age\_last\_milestone\_year', 'funding\_total\_usd') that will play an important role to train ML models. We need to ensure that these features don't have any outliers so to ensure that we draw a box plot to see if there exist any outliers in these features.

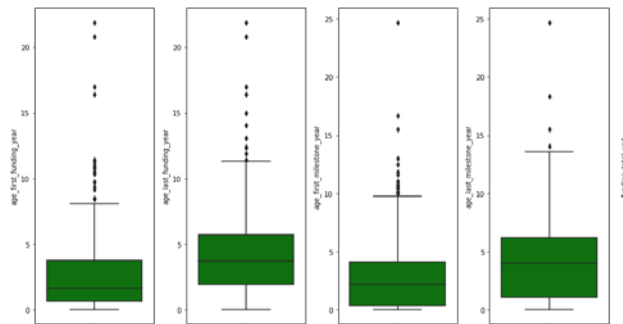


Figure 3

Visualizing the box plot we have assured that there exist outliers so to handle the outliers we take log transformation of funding and milestone year variable then we again draw the boxplot to ensure that our work has been paid off.

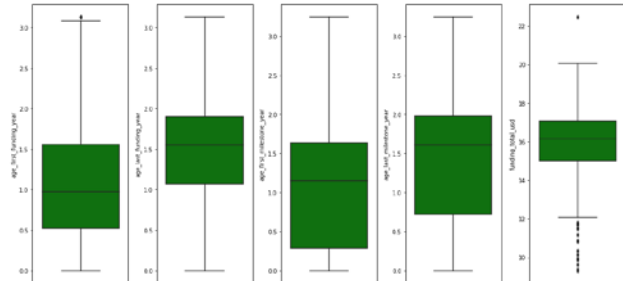


Figure 4

The boxplot shows that the outliers have been reduced very wisely

### 3.2.3 Feature Scaling

For some machine learning algorithms, we must have feature scaling otherwise the model will be biased to some particular feature so to avoid this biasness of the model we have to take all the values of all features in the range of 0 to 1. After analyzing the dataset column 'age\_last\_funding\_year', 'age\_first\_milestone\_year', 'age\_last\_milestone\_year', 'funding\_rounds', 'milestones', 'age', and 'avg\_participants' need to be scaled in the range of 0 to

1. We have used MinMaxScaler to do this. Min Max Scaler uses the equation (1) to scale any feature.

$$x_{Scaled} = \frac{x - x_{min}}{(x_{max}) - x_{min}} \text{ ----- (1)}$$

Initially we scaled only the train set value then on to the test set scaled value applied to avoid the miscalculation.

### 3.2.4 Feature Engineering

In order to make machine learning algorithms work well we have done feature engineering which is mainly the process of selecting, manipulating, and transforming raw data into features. This is an essential part of our data pre-processing to design and train better results.

Selecting and manipulating the 'closed\_at' and 'found\_at' column we have added a new feature 'Age' that represents the age of the startup company. We found the 'last\_date' of a startup from the 'closed\_at' column then to get the age of the startup we subtract the founded date of the company from the 'last\_date' of the company.

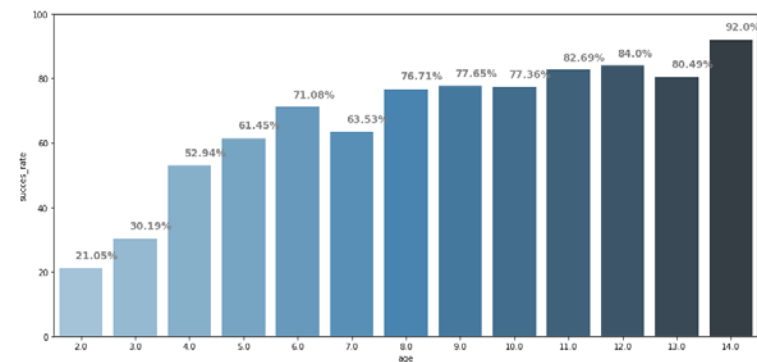


Figure 5

fig (5) represents a business insight from age: startups that have a lifespan of more than 4 years have a tendency to be successful startups (more than 52%). This analysis shows that 'Age' is an important feature for predicting the success of a startup.

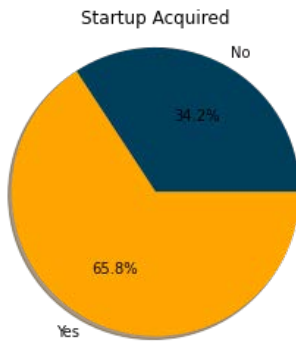
Furthermore, we have checked for distribution, there are some features with skewness distribution more than 2. Handling it with normalization. Column 'age\_first\_funding\_year', 'relationships', 'funding\_total\_usd' has skewed distribution more than 2 that's needed to be normalized which result to form

3 more column names with starting 'norm\_' then rest are respective name of each column that's been normalized.

After feature engineering we have got an increased number of columns from 49 to 53 columns as we have added 4 more columns with this process.

### 3.3 Class Imbalance

During the data visualization process, we have recognized class imbalance issues in the dataset. Fig (6) implies that the label-0 class has only 34.2% data compared with the label-1 with 65.8% data which clearly represent class label-0 as a minority class.



To combat this problem, we have chosen two ways: one is oversampling training data with a technique named RandomOverSampler which over-sample the minority class by picking samples at random with replacement, and another one is oversampling plus stratified k-fold cross validation technique.

### 3.4. Machine learning classifiers

#### 3.4.1 LGBM Classifier

LGBM is a gradient boosting framework that uses tree based learning algorithms. It uses two techniques, 1) Gradient based One Side Sampling (GOSS) and 2) Exclusive Feature Bundling (EFB) and it fulfills the limitations of histogram-based algorithm that is primarily used in all Gradient Boosting Decision Tree frameworks. These two techniques comprise together and thus gives a better result over other GBDT frameworks. The framework has faster training speed and higher efficiency. With lower memory usage it provides better accuracy. This framework is also capable of handling large-scale data. Moreover, this framework supports gpu learning. Even though this

framework is used widely, it is advised not to use this framework on small datasets as it is sensitive to overfitting and can easily be overfitted with small data. The most valuable aspect of this framework is that it focuses on accuracy. The equation used in GOSS technique is described below shortly.

Considering a dataset of  $\{x_1, \dots, x_n\}$ , where each  $x_i$  is a vector with dimension  $s$  in space  $X^s$ . In each iteration of gradient boosting, the negative gradients of the loss function with respect to the output of the model are denoted as  $\{g_1, \dots, g_n\}$ . In this GOSS technique, the training instances are ranked according to their absolute values of their gradients in descending order. After that,  $(\text{top-}a \times 100\%)$  instances with larger gradients are kept and we get an instance subset  $A$ . Then, for the remaining set  $A^c$  consisting  $(1-a) \times 100\%$  instances with smaller gradients, we further randomly sample a subset  $B$  with size  $b \times |A^c|$ . Finally, we split the instances according to the estimated variance gain at vector  $V_j(d)$  over the subset  $A \cup B$ .

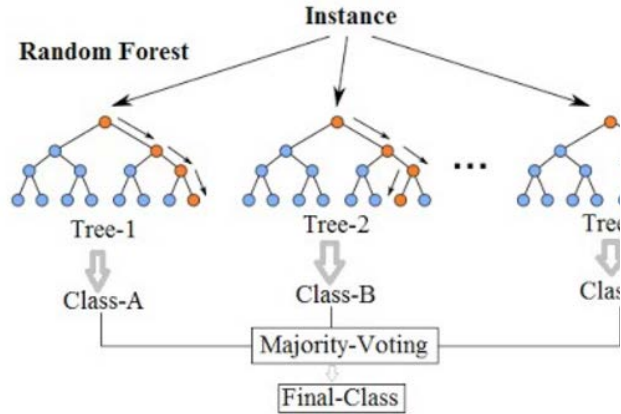
$$\bar{v}_j(d) = \frac{1}{n} \left\{ \frac{\left( \sum_{x_i \in A_l} g_i + \frac{1-a}{b} \sum_{x_i \in B_l} g_i \right)^2}{n_l^j(d)} + \frac{\left( \sum_{x_i \in A_r} g_i + \frac{1-a}{b} \sum_{x_i \in B_r} g_i \right)^2}{n_r^j(d)} \right\} \quad \text{-----}(2)$$

where  $A_l = \{x_i \in A : x_{ij} \leq d\}$ ,  $A_r = \{x_i \in A : x_{ij} > d\}$ ,  $B_l = \{x_i \in B : x_{ij} \leq d\}$ ,  $B_r = \{x_i \in B : x_{ij} > d\}$ , and the coefficient  $(1-a)/b$  is used to normalize the sum of the gradients over  $B$  back to the size of  $A^c$ .

#### 3.4.2 Random Forest Classifier

This classifier creates a lot of decision trees from a randomly selected subset of the training set. This classifier then uses averaging to enhance the prophetic accuracy and management overfitting. So, this classifier is basically a set of decision trees (DT) from a randomly selected subset of the training set and after that all the votes from different decision trees are collected to decide the final prediction. We used ensemble machine learning algorithms and its smart or glorious performance across regression predictive modeling problems. The basic idea of this classifier is shown below.





The training algorithm for random forests applies the general technique of bagging (bootstrap aggregating). a training set  $X = x_1, \dots, x_n$  with responses  $Y = y_1, \dots, y_n$ , bagging repeatedly ( $B$  times) selects a random sample with replacement of the training set and then it fits trees to targeted samples:

For  $b = 1, \dots, B$ : Sample, with replacement,  $n$  training examples from  $X, Y$ .

$$\sigma = \sqrt{\frac{\sum_{b=1}^B (f_b(x') - \hat{f})^2}{B-1}} \text{ -----(4)}$$

Here,  $B$ , number of trees(samples) is a free parameter. Different amounts of trees are used depending on the size and nature of the training set. The mean prediction error on each training sample  $x_i$  is done by using only the trees not having  $x_i$  in their bootstrap sample.

### 3.4.3 SVM Classifier

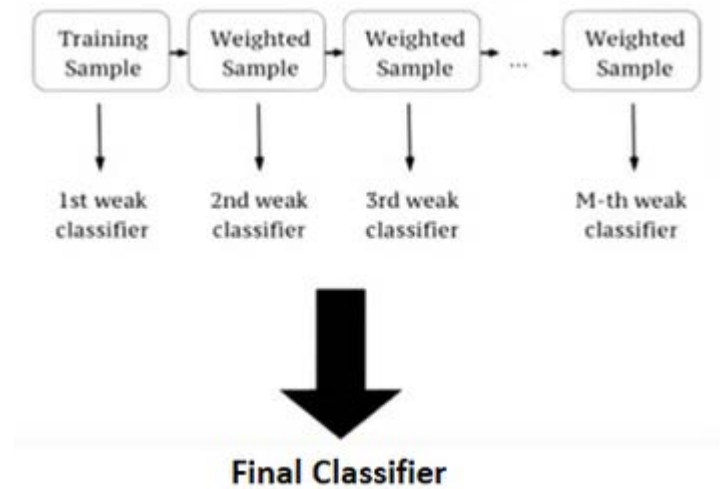
Support Vector Machine is a Supervised Machine Learning Algorithm which is used for classification and/or regression. The basic technique of this classifier is that it finds a hyper-plane that creates a boundary between the types of data and in the 2D space it's just a line. In SVM, each data item in the dataset is plotted in an  $N$ -dimensional space, where  $N$  is the number of attributes in the data. Then, the optimal hyper-plane to separate the data is determined. This classifier can only perform binary classification. Though it can only perform in binary classification, it can be used on multi-class problems. For that, a binary tree will have to be created for each of the classes of the data. The two results of each classifier will belong to that class OR. Radial Basis Function Kernel (RBF) is the default kernel used in this classifier. The equation used by RBF is as following-

$$k(x, x') = \exp(-\gamma ||x - x'||) \text{ -----(3)}$$

### 3.4.4 XGB Classifier

XGB is an implementation of Gradient Boosted Decision Trees. In this classifier, decision trees are created in sequential order. Weights are assigned to all the independent variables which are then fed into the decision tree that predicts the results. The wrong predicted weights of variables are increased and added to the second decision tree.

Then individual predictors ensemble to give a strong and more precise model. The basic idea of XGB is shown below.



The equation used by this classifier is shown below.

$$L(\phi) = \sum l(\hat{y}, y_i) + \sum \Omega(f_k) \text{ -----(5)}$$

$$\text{Where } \Omega(f) = \gamma T + \frac{1}{2} \lambda ||\omega||^2 \text{ -----(6)}$$

### 3.4.5 GradientBoosting Classifier

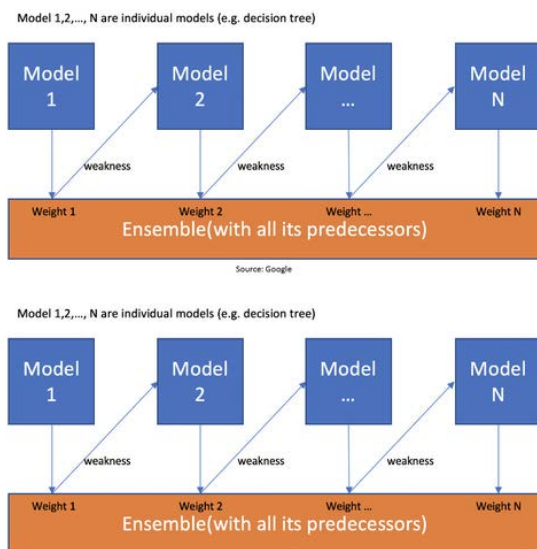
It is a boosting algorithm. Here, each predictor corrects its predecessor's error. Each predictor is trained using the residual errors of the predecessor as labels. The ensemble consists of  $N$  trees. The first tree is trained using the feature matrix  $X$  and the label is  $Y$ . The predictions labelled as  $Y_1$  are used to determine

the training set residual errors  $r_1$ . The second tree is then trained using the feature matrix  $X$  and the residual errors  $r_1$  of the first tree as labels. The predicted results  $r_1$  are then used to determine the residual  $r_2$ . This process repeats till  $N$  trees consisting the ensemble are trained. Each tree predicts a label and thoroughly the final prediction is determined. The equation used in this classification is shown below.

$$y(\text{pred}) = y_1 + (\eta * r_1) + (\eta * r_2) + \dots + (\eta * r_N)$$

### 3.4.6 AdaBoostClassifier

It is a boosting technique used as an Ensemble Method. Here, weights are re-assigned to each instance. The higher weights are assigned to incorrectly classified instances. Boosting is used to reduce bias as well as variance for supervised learning. It works on the basis of learners/stumps growing sequentially. Without the first learner, each learners are grown from previous grown learners. Thus, weak learners are strengthened. The basic idea of AdaBoost is shown below.



Initially, the first base learner/stump is created. Then, the following learners will grow. After all of them are grown, the Total Error (TE) will be calculated. After that the performance of the stump will be calculated. The equation used for this is shown below.

$$\text{Performance of Stump} = \frac{1}{2} \ln\left(\frac{1-TE}{TE}\right)$$

After determining the performance of stump, weights will be updated for incorrectly classified records. The equation used is shown below.

$$\text{New Sample Weight} = \text{Sample Weight} * e^{(\text{Performance})}$$

Then, the weights will be updated for correctly classified records. The equation is as follows.

$$\text{New Sample Weight} = \text{Sample Weight} * e^{-(\text{Performance})}$$

Finally, from the old dataset, a new dataset will be created. In this new dataset, the frequency of incorrectly classified records will be more than the correct ones. The new dataset will be created using and considering the normalized weights.

### 3.4.7 ExtraTreesClassifier

Extremely Randomized Trees Classifier (Extra Trees Classifier) is a form of ensemble learning approach that outputs a classification result by aggregating the outcomes of several de-correlated decision trees gathered in a "forest." It is conceptually identical to a Random Forest Classifier, with the exception of how the decision trees in the forest are constructed. The Extra Trees Forest's Decision Trees are all created from the original training sample. Then, at each test node, each tree is given a random sample of  $k$  features from the feature-set, from which it must choose the best feature to partition the data using some mathematical criterion. Multiple de-correlated decision trees are created as a result of this random sampling of characteristics.

$$\text{Entropy}(S) = \sum_{i=0}^c -p_i \log_2(p_i) \dots (7)$$



#### 4 Experiments and Results:

This section discusses the experimental methodology and corresponding result of it. As mentioned in the methodology we have used mainly 7 classification machine learning algorithms with two kinds of validation technique one is HoldOut Validation Approach or known as Train-test split and another one is Stratified K Fold Cross Validation Technique where k = 10 refers 10-fold. We will divide two sections for these two validation techniques and then a comparison will be done between these two techniques to select the best model to predict the success of a startup.

##### I. HoldOut Validation Approach

We have divided the dataset into two parts, train and test with a percentage of 70 and 30 respectfully. We have trained seven ML algorithms with this train set then evaluate each model separately. Performance of each model were evaluated in terms of confusion matrix, accuracy, precision, recall, F1-score, and area under the ROC curve.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \text{---(8)}$$

$$\text{Recall} = \frac{TP}{TP+FN} \text{-----(9)}$$

$$\text{Precision} = \frac{TP}{TP+FP} \text{-----(10)}$$

$$\text{F1-score} = \frac{2*\text{recall}*\text{precision}}{\text{recall}+\text{precision}} \text{-----(11)}$$

Here,

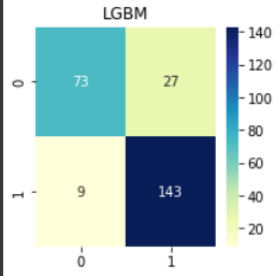
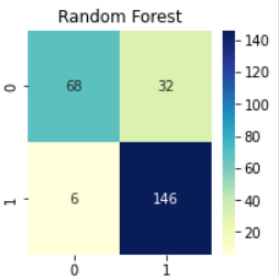
TP = Actual class is positive and predicted positive.

TN = Actual class is negative and predicted negative.

FP = Actual class is negative but predicted positive.

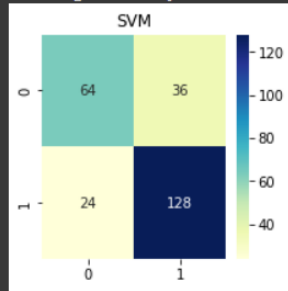
FN = Actual class is positive but predicted negative

Table 2:Models Evaluation

1.LGBM:					
Training Accuracy : 1.0					
Testing Accuracy : 0.8571428571428571					
					
	precision	recall	f1-score	support	
0	0.89	0.73	0.80	106	
1	0.84	0.94	0.89	152	
accuracy			0.86	258	
macro avg	0.87	0.84	0.85	258	
weighted avg	0.86	0.86	0.85	258	
-----					
ROC Curves = 0.8353947368421053					
Precision-Recall Curves = 0.9088401149933658					
2.RandomForest:					
Training Accuracy : 1.0					
Testing Accuracy : 0.8492063492063492					
					
	precision	recall	f1-score	support	
0	0.92	0.68	0.78	106	
1	0.82	0.96	0.88	152	
accuracy			0.85	258	
macro avg	0.87	0.82	0.83	258	
weighted avg	0.86	0.85	0.84	258	
-----					
ROC Curves = 0.8202631578947367					
Precision-Recall Curves = 0.9022802793500605					

### 3. SVM

Training Accuracy : 0.8508771929824561  
Testing Accuracy : 0.7619047619047619



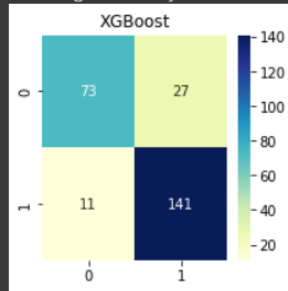
	precision	recall	f1-score	support
0	0.73	0.64	0.68	100
1	0.78	0.84	0.81	152
accuracy			0.76	252
macro avg	0.75	0.74	0.75	252
weighted avg	0.76	0.76	0.76	252

roc\_auc 0.7410526315789473

ROC Curves = 0.7410526315789473  
Precision-Recall Curves = 0.8589155816370193

### 4.XGBoost

Training Accuracy : 0.9486215538847118  
Testing Accuracy : 0.8492063492063492

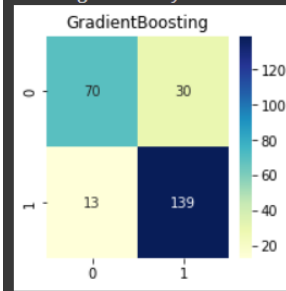


	precision	recall	f1-score	support
0	0.87	0.73	0.79	100
1	0.84	0.93	0.88	152
accuracy			0.85	252
macro avg	0.85	0.83	0.84	252
weighted avg	0.85	0.85	0.85	252

ROC Curves = 0.8288157894736843  
Precision-Recall Curves = 0.9052840434419382

### 5. GradientBoosting

Training Accuracy : 1.0  
Testing Accuracy : 0.8293650793650794

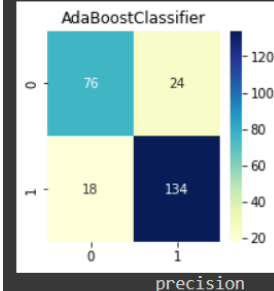


	precision	recall	f1-score	support
0	0.84	0.70	0.77	100
1	0.82	0.91	0.87	152
accuracy			0.83	252
macro avg	0.83	0.81	0.82	252
weighted avg	0.83	0.83	0.83	252

ROC Curves = 0.8072368421052631  
Precision-Recall Curves = 0.8942730964492098

### 6. AdaBoost

Training Accuracy : 0.8771929824561403  
Testing Accuracy : 0.8333333333333334



	precision	recall	f1-score	support
0	0.81	0.76	0.78	100
1	0.85	0.88	0.86	152
accuracy			0.83	252
macro avg	0.83	0.82	0.82	252
weighted avg	0.83	0.83	0.83	252

roc\_auc 0.8207894736842105

ROC Curves = 0.8207894736842105  
Precision-Recall Curves = 0.9005543923098887

## II. Stratified K Fold Cross Validation

The models were evaluated in terms of median and standard deviation of 10-fold. From fig(5) we can see that Random forest , Extra Trees, GradientBoosting, LGBMC, XGBC perform really well from this validation technique and also this validation technique

improved the accuracy than the HoldOut validation approach by comparing Table (2) with Table (3) we can conclude that. Hyper Parameters tuning applied on SVMC, AdaBoost and GradientBoost with GridSearchCV that's improved the result of these algorithm than before. Finally, we have got our 6 top algorithm that will be used as the estimator of Ensemble Classification. Voting Classifier from Ensemble classification has been used to build our final model with Stratified 10-Fold Validation technique which outperform all the model that's been build till now.

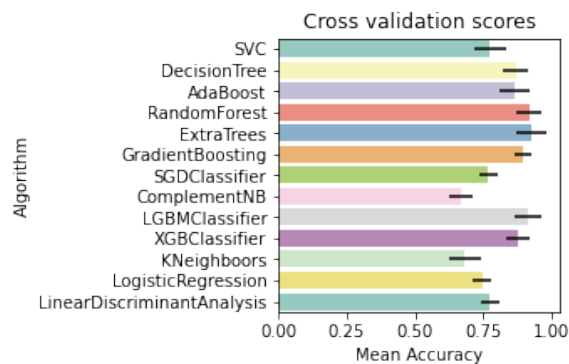


Figure 6: Cross validation score of all the models.

```
Fitting 10 folds for each of 112 candidates, totalling 1120 fits
Adaboost score: 0.8735443037974683
Fitting 10 folds for each of 28 candidates, totalling 280 fits
SVC classifier score is :0.867246835443038
Fitting 10 folds for each of 144 candidates, totalling 1440 fits
GradientBoosting score is :0.8985601265822784
```

Figure 7: Evaluation of SVC, AdaBoost, Gradient Boosting after Hyper parameters tuning with GridSearchCV

```
[0.85    0.875    0.925    0.8875   0.9125   0.925
 0.975    0.9375    0.98734177 0.94936709]

Results = 0.9224208860759493 +/- 0.040935905115090994
```

Figure 8: Result of voting classification of Ensemble Classifier.

## 5 Conclusion:

This paper looks into the startup dataset. A dataset detailing information about various attributes of a startup to be succeed or failure. There was a total of 923 instances and 49 attributes initially. After careful preprocessing, feature engineering, feature selection, machine learning algorithms were applied with two kind of validation technique and hyper parameters tuning we got some best classifiers to use in ensemble classification which ultimately gave the best model with the prediction of 92.2 %.

## References:

1. L. Cacciolatti, A. Rosli, J. L. Ruiz-Alba, and J. Chang, "Strategic alliances and firm performance in startups with a social mission", Journal of Business Research, vol. 106, 2020, pp. 106-117.
2. G. Bortoluzzi, M. Tivan, A. Tracogna, and F. Venier, "The Growth Drivers of Start-up Firms and Business Modelling: The First Step toward a Desirable Convergence", Management, 9, 2014.
3. J. Guzmán, and R. Lussier, "Success factors for small businesses in Guanajuato, Mexico", International Journal of Business and Social Science, vol. 6, 2015, pp. 1-7.