LumiStartup: “Illuminating the Entrepreneurial Cosmos Success with AI-Powered Predictive Analytics”

Sifat Momen
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
sifat.momen@northsouth.edu

Mohammod Abdullah Bin Hossain
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
bin.abdullah@northsouth.edu

Mahadi Melon Soykat
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
mahadi.soykat@northsouth.edu

Jubaer Al Noman
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
jubaer.noman @northsouth.edu

Sharjina Jahan
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
sharjina.jahan@northsouth.edu

***Abstract*— Recent research highlights the effectiveness of tree-based models and ensemble methods, such as Random Forest and boosting algorithms, in handling tabular data with distinct decision boundaries and diverse feature spaces. By leveraging decision trees as weak learners, these techniques mitigate variance (in Random Forest) and bias (in boosting models) through model aggregation, often surpassing the performance of deep learning-based approaches. Motivated by these insights, our study focuses on constructing ensemble models to predict life satisfaction. The ensemble framework amalgamates various classifiers, emphasizing boosting models like XGBoost, LGBoost, Gradient Boosting, and AdaBoost. Through systematic experimentation, we discovered that the ensemble comprising Random Forest, Gradient Boosting, and LightGBM yielded the most promising results. Hyperparameter optimization further refined the model, revealing that the Gradient Boosting led to optimal performance.**

Keywords— Machine learning, Ensemble, Explainable AI,TPR,FPR Introduction

The path of an entrepreneur is full with unknowns. Though many ideas come to life on a daily basis, only few of them go on to become successful businesses. Forecasting the success of startups has emerged as a fascinating task, with both entrepreneurs and investors looking for a magic wand to help them make decisions. In this endeavour, machine learning (ML) has shown to be a useful instrument, providing data-driven insights that show the way to startup viability. One study published on ResearchGate highlights the complexity of predicting startup success in the information technology (IT) sector, citing the "diverse factors and uncertainty" that plague the process [1]. However, the authors remain optimistic about the potential of ML, acknowledging its ability to handle large datasets and identify patterns that might escape human intuition [1]. Similarly, research presented at the University of Nebraska at Lisbon emphasizes the role of ML in developing a "systematic method" for startup success prediction, leveraging a multitude of factors to enhance the accuracy of such forecasts [2] .The path of an entrepreneur is full with unknowns. Though many ideas come to life on a daily basis, only few of them go on to become successful businesses. Forecasting the success of startups has emerged as a fascinating task, with both entrepreneurs and investors looking for a magic wand to help them make decisions. In this endeavour, machine learning (ML) has shown to be a useful instrument, providing data-driven insights that show the way to startup viability.

I. LITERATURE REVIEW

Researchers have used a variety of machine-learning approaches to forecast startup success during the past ten years. To identify successful startups, several classifiers have been employed, including logistic regression, random forest, support vector machines (SVM), and decision trees. A few of the publications discussed below have made use of both structured and unstructured data sources, including datasets from CrunchBase. Since these publications include similar datasets to our own, we have concentrated on them.

A. *Research Using CrunchBase Dataset*

Examined several machine learning classification algorithm types and carried out a comparison study. The highest accuracy, 78.01%, was observed with logistic regression. Their findings indicated that 78% of the time, multilayer perceptron’s (MLP), logistic regression, and sequential minimum optimization (SMO) could forecast when startup success would begin. The performance of the Support Vector Machine

(SVM), Naïve Bayes, and bagging was similarly good [9].

Used TechCrunch news article text mining in conjunction with the CrunchBase database. The model's True Positive Rate (TPR) varied from 60% to 79.8% for various firm categories using a Bayesian Network technique, whereas the False Positive Rate (FPR) varied from 0 to 8.3% based on how full the data was [10].

Predictive analytics using combined structured and unstructured data in the startup environment. This study demonstrated the possibility of combining several data sources to increase forecast accuracy, even if it did not explicitly predict acquisitions [11].

B. Comparative Analysis of Machine Learning Techniques

Investigated different types of Machine Learning classification algorithms and conducted a comparative analysis. Logistic regression was found to yield the highest accuracy of 78.01%. Based on their results, logistic regression, sequential minimal optimization (SMO), and multilayer perceptron (MLP), could predict the onset of startup success with an accuracy of 78%. Bagging, Naïve Bayes, and Support Vector Machine (SVM) also performed well [5].

Utilized the CrunchBase database combined with text-mining of news articles from TechCrunch. Using a Bayesian Network algorithm, the model's True Positive Rate (TPR) ranged from 60% to 79.8% for different company categories, with a False Positive Rate (FPR) ranging from 0 to 8.3% depending on the completeness of the data [4].

Predictive analytics using combined structured and unstructured data in the startup environment. This study demonstrated the possibility of combining several data sources to increase forecast accuracy, even if it did not explicitly predict acquisitions [3].

Investigated different types of Machine Learning classification algorithms and conducted a comparative analysis. Logistic regression was found to yield the highest accuracy of 78.01%. Based on their results, logistic regression, sequential minimal optimization (SMO), and multilayer perceptron (MLP), could predict the onset of startup success with an accuracy of 78%. Bagging, Naïve Bayes, and Support Vector Machine (SVM) also performed well [6].

Utilized the CrunchBase database combined with text-mining of news articles from TechCrunch. Using a Bayesian Network algorithm, the model's True Positive Rate (TPR) ranged from 60% to 79.8% for different company categories, with a False Positive Rate (FPR) ranging from 0 to 8.3% depending on the completeness of the data [7].

Predictive analytics using combined structured and unstructured data in the startup environment. This study demonstrated the possibility of combining several data sources to increase forecast accuracy, even if it did not explicitly predict acquisitions [8].

Used support vector machines (SVM) and decision trees to detect and predict the risk of startup success. Detection was successful with the SVM classifier with an accuracy of 82% [12].

C. Handling Sparse Data

Carried out several methodical tests to assess how well-supervised machine learning algorithms performed for predicting startup success using various data quantities. With an accuracy score of 77.3%, they concluded that SVM was the most accurate technique for small datasets [13].

Performed K-means clustering on the CrunchBase dataset and divided it into clusters to get the most accurate result. Their proposed work achieved an accuracy of around 98.7%, which was significantly higher than existing algorithms [14].

II. METHODOLOGY

A. Objectives

This study aims to delve into the complexities of predicting startup success by addressing a set of targeted research questions. These questions bridge the business and technical aspects of startups with the analytical precision of machine learning. Our goal is to provide a comprehensive understanding of the factors influencing startup success and to develop robust predictive models. The methodology employed aligns closely with our objectives, each corresponding to a specific research question.

1. ***1.Identification of Key Success Factors in Startups*****:** To answer the first research question, “What are the most significant factors influencing the success of startups, and how do these factors interrelate?”, the primary objective of this study is to systematically identify these success determinants. We applied Recursive Feature Elimination with Cross-Validation (RFECV) to our extensive dataset, reducing an initial set of 49 features to the 37 most crucial ones. This method allows us to pinpoint these pivotal factors and understand their interrelationships and relative importance, thereby providing a foundation for deeper insights into what drives startup success.

2. ***Optimization of Startup Success Prediction:*** In response to the second research question, “How effectively can an ensemble of machine learning algorithms predict the success of a startup, and what are the primary indicators

contributing to this accuracy?”, the second objective is to develop a highly accurate predictive model. We utilized an ensemble of machine learning algorithms to address the challenges posed by the imbalanced nature of our dataset and to enhance both accuracy and the F1 score. This approach underscores our commitment to creating a reliable and robust model for predicting startup success, ensuring that our predictions are both valid and dependable.

3. ***Enhancement of Model Interpretability with Explainable AI:*** The third objective, “How can Explainable AI methodologies improve the interpretability of AI models used for predicting startup success, and what impact does this enhanced understanding have on the adoption and trust of these models?”, aims to add a layer of transparency to the predictive models. We employed Explainable AI techniques to illuminate the decision-making processes of our models.In pursuit of these objectives, we adopted a multi-faceted approach that integrates advanced machine learning techniques with business insights. This approach aims to contribute novel and practical understandings of predicting and fostering startup success.

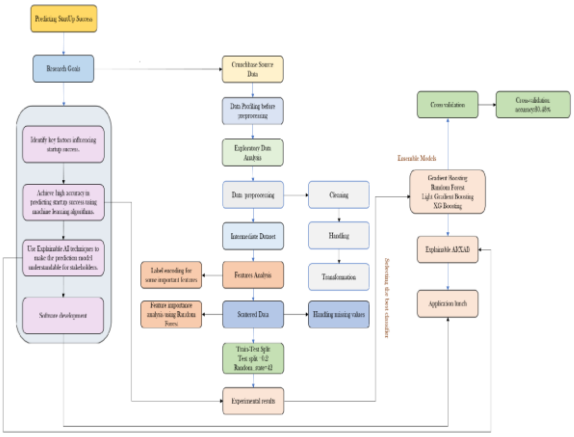

Predicting StartUp Success
Research Goals
Identify key factors influencing startup success.
Achieve high accuracy in predicting startup success using machine learning algorithms.
Use Explainable AI techniques to make the prediction model understandable for stakeholders.
Software development
Crunchbase Source Data
Data Profiling before preprocessing
Exploratory Data Analysis
Data preprocessing
Intermediate Dataset
Features Analysis
Scattered Data
Train-Test Split
Test split =0.2
Random_state=42
Experimental results
Label encoding for some important features
Feature importance analysis using Random Forest
Cleaning
Handling
Transformation
Handling missing values
Selecting the best classifier
Ensemble Models
Gradient Boosting
Random Forest
Light Gradient Boosting
XG Boosting
Cross validation
Cross-validation accuracy:80.48%
Explainable AI/XAI
Application lunch

Fig.1. Workflow diagram that outlines the key steps in applying machine learning algorithms to predict the likelihood of startup success, addressing research objectives around identifying critical factors that contribute to startup growth and sustainability.

B. Dataset Collection

For this project, we utilized our dataset from CrunchBase, a highly structured database widely available on Kaggle, was accessed on 24th April 2024. This dataset has immense user and moderator contributions, giving detailed information about start-ups, their profiles, funding rounds, and some key people. To note, access to this data set was granted under an Academic License explicitly for the purposes of this research.

1. Source and Accessibility

The dataset is found on Kaggle and has some of the critical features that can affect a startup's success. Free of charge, the data can be accessed, and the users can download it directly. The dataset is under the Creative Commons Attribution 4.0 International (CC BY 4.0) and allows for its use in academic or commercial projects as long as the original author is duly credited.

2. Data Scope and Period

The data of the dataset include enormous numbers of startups and generally comprise their conditions and metrics up to the time of the last dataset update. While the time range of the exact data taken is not given, the diversity and volume of the entries offer a full snapshot of startup dynamics to serve as a solid basis for training predictive models.

3. Dataset Composition

- Companies: The dataset has information about 922 start-ups in different industries and development stages with their 49 features. Each record is a unique company with some attributes that describe its operational and financial conditions.

- Attributes: Key attributes comprise the name of the company, the funding amount, the number of funding rounds, and the number of employees. These attributes play a key role in modeling and predicting the factors that affect the success or failure of a start-up.

- Success Indicator: A particular attribute in the dataset indicates whether a startup is considered successful or not. Therefore, that feature is taken as the target variable for our predictive analysis.

TABLE I

The features of the dataset which has Quantitative types

Features	Description
age_first_funding_year	Years first funding year for each startup
age_last_funding_year	Years first funding year for each startup
relationships	Number correlations with

	another startup
funding_rounds	Number of funding rounds
funding_total_usd	Total amount of fund raised
milestones	Milestones achieved
age_first_milestone_year	Year of first milestone
age_last_milestone_year	Year of last milestone
avg_participants	Total number of client
zip_code	Startup’slocation indicator
Lattitude and Longitude	Startup’s location indicator

TABLE II

The features of the dataset which has categorical types

Features	Description
state_code	Location for each startup
industry_type	Startups belonging industry type
has_VC	Identifying is startup has VC or not
has_angel	Startups first mother funded
has_roundA	Identifying is startup raised fund for round one
has_roundB	Identifying is startup raised fund for round two
has_roundC	Identifying is startup raised fund for round three
has_roundD	Identifying is startup raised fund for round four
is_top500	Is startup achieved top 500
status	Is startup is now acquired or closed

4. *Ethical and Legal Consideration*

We made sure to follow ethical guidelines and legal constraints associated with the dataset. All data processing activities were performed according to the terms of usage by Kaggle and the license of the dataset. Concerns regarding confidentiality and privacy were fully respected, since the dataset does not include personal identification information, only details about the companies that are publicly available.

C. *Exploratory Data analysis*

In this section we analyzed our dataset through dividing into multiple section like the data which in numeric value, categorical value and other types etc. Then we defined our target features ‘status’ and converted this to the binary that means when status= acquired means 1 and when status = closed means 0. we tried show a demographic graphs or which features are more influencing the startups.

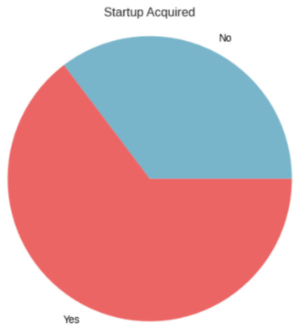


Fig 2: Demographic graph for startups acquired or closed

We analyzed whether appointing VC is influential for a startup or not through the bar chart. Fron the chart, we conclude having a VC is beneficial for running the startup.

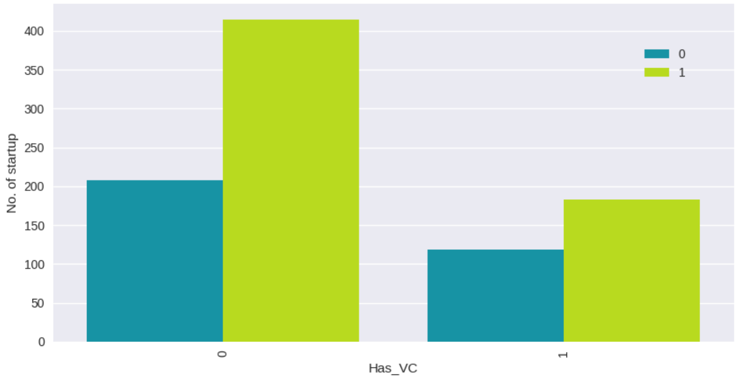


Fig.3: Bar chart for VC influence in startup acquired and closed

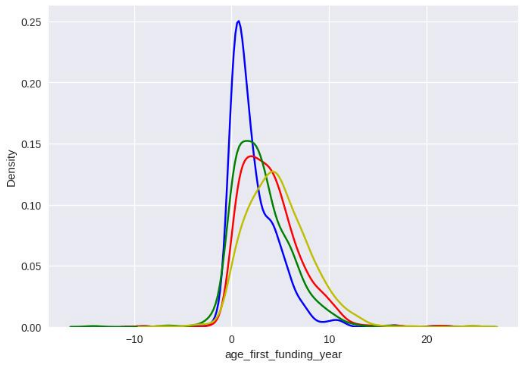


Fig 4: relational graph between rounds and milestones

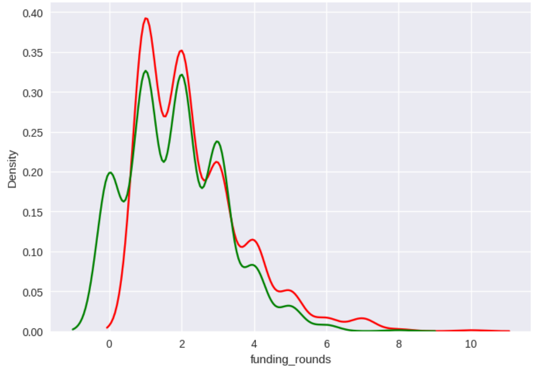


Fig 5: relational graph between funding year and milestone

We also analyzed the interrelations between some important features.For example, Fig 3.3.3a we can see an interreation between funding rounds and the milestones and Fig 3.3.3b we tried show a graph of correlation between age_first_funding_year,age_last_funding_year,age_first_milestone_year,age_last_milestone_year.we also anlyzed what are the top influential and industries and which are the most common states that startups like to hookup their head offices.

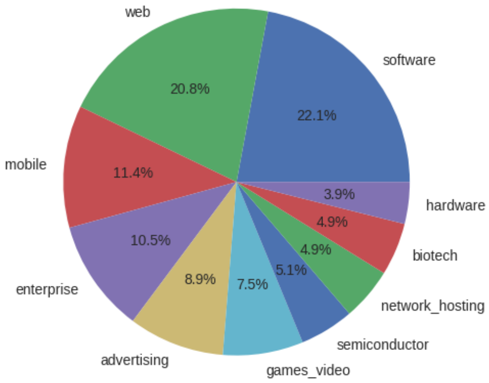


Fig 6: Flowchart of the top startup industries

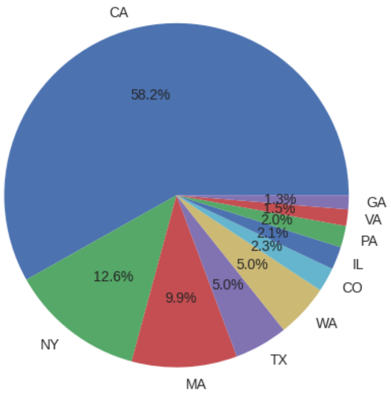


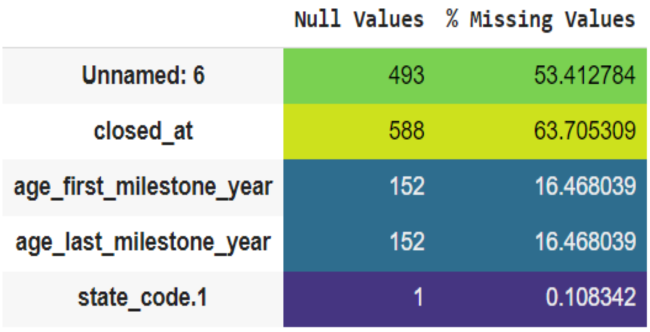
Fig 7: Flowchart of the most common states of startups

D. *Data Preprocessing*

Quality data is pivotal to predicting startup success, necessitating a rigorous pre-processing pipeline to ensure reliable predictions. This research employed a methodical approach to manage and optimize the collected dataset, incorporating the following stringent data pre-processing steps are:

1. *Handling missing and negative values*

In the context of predicting startup success, addressing missing values in the dataset is crucial, as many machine learning algorithms cannot handle them directly.



	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

Fig 8: Percentage matrix of the missing values of the features

Initially, all features in the raw dataset with overall more than 22% null values were removed.

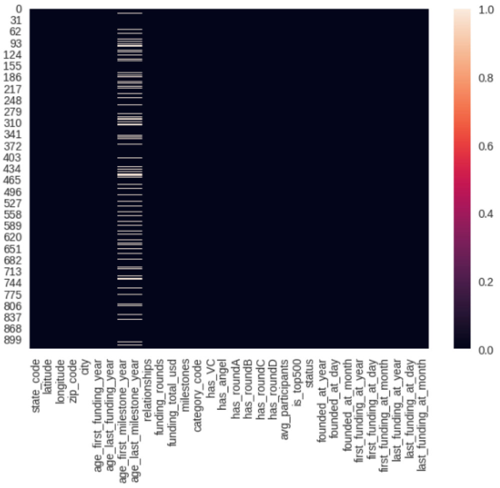


Fig 9: Heat map of the missing values

Figure 3.3. illustrates the matrix of missing values, where each column corresponds to a feature and each white spot indicates a missing value in that column. To fill all missing for age_first_milestone_year and age_last_milestone_year replace all NaN values with the median() of those features. Here Fig 3.3.4 illustrates the matrix of missing values after handling them. After, then check the number of NaN values in the two features to ensure there were no remaining missing values.

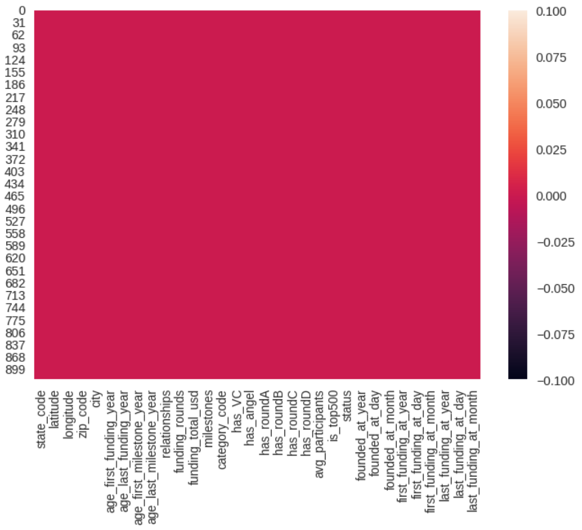


Fig 10: Matrix after handling missing values

2. *Categorical Encoding*

Categorical encoding transforms categorical data into an integer format, enabling machine learning models that depend on mathematical formulas to process the data effectively. Most of the categorical features in our dataset were ordinal. To preserve the ordinal relationship of the data, we utilized an ordinal encoder. This process converted categorical responses into a standardized numerical format. For instance, the startup target feature “status” was encoded using a predefined scale ('closed' = 0, 'acquired' = 1). This encoding ensured that the ordinal nature of the data was maintained, allowing the models to interpret and utilize these features accurately.

3. **Feature selection**

Feature importance analysis is a crucial technique for discerning the most influential features in machine learning models, aiding in predictive accuracy and model interpretation. It facilitates the identification of key features, thereby streamlining feature selection and revealing significant data patterns. In Fig 3.4.3.1, we present a bar chart illustrating the most vital features within our dataset. Leveraging this insight, we prudently removed extraneous features that could potentially impede our models' performance.

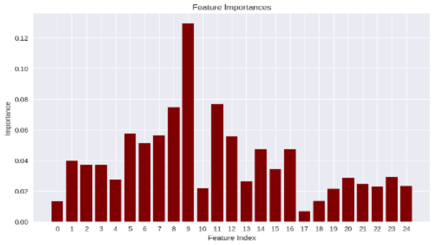


Fig 11: Bar charts for the important features

Notably, the analysis highlights "relationships" as the most pivotal feature (feature number 9), crucial for predicting startup success. We used random forest classifier to analyze the best features among all.

E. *Train Test Split*

The dataset undergoes an 80-20 split, with 80% of the data allocated for training and the remaining 20% reserved exclusively for validating the predictions generated by our trained model. Prior to the split, the dataset is shuffled—a common practice to mitigate any inherent biases stemming from the data's original ordering,where random_state =42.

F. *Hyperparameter tuning*

Machine learning models' performance and capacity for generalization can be greatly improved via hyperparameter adjustment. We performed extensive hyperparameter tuning using Grid Search CV and Randomized Search CV for each model used in our studies in order to determine the optimal combinations. In order for Grid Search CV to function, all possible combinations of hyperparameters are created into a grid, and the model is then cross-validated against each combination. On the other hand, Randomized Search CV uses cross-validation to assess the model for every sampled combination while randomly selecting a predetermined number of choices from the hyperparameter space. The best hyperparameters are chosen in accordance with the model's performance, which is evaluated using the average score obtained from the cross-validations.Grid Search CV covers all of the hyperparameter space, but it is computationally demanding. However, because Randomized Search CV just looks at a portion of the hyperparameter space in an attempt to approximate the optimal hyperparameters, it is more computationally efficient. We used Randomized Search CV for models like XGBoost, Random Forest, and LightGBM because of the wide range of hyperparameter combinations that are accessible for these models. We were able to find the ideal settings and navigate the vast hyperparameter areas with efficiency thanks to this method.In models with a smaller hyperparameter space, like Gradient Boosting, we employed Grid Search CV to thoroughly investigate every possible combination. When utilizing the hyperparameter tuning procedure instead of the default values, performance gains were substantial. The table below displays each model's optimized hyperparameters. Extensive hyperparameter tuning was not required for models such as K-Nearest Neighbors and Decision Tree, as the default parameters yielded adequate performance.

Best hyperparameters found using hyperparameter tuning on various models used in this research.

1. *XGBoost Classifier*

"learning_rate": [0.05, 0.10, 0.15, 0.20, 0.25, 0.30],
"max_depth": [3, 4, 5, 6, 8, 10, 12, 15],
"min_child_weight": [1, 3, 5, 7],
"gamma": [0.0, 0.1, 0.2, 0.3, 0.4],
"colsample_bytree": [0.3, 0.4, 0.5, 0.7]

2. *Gradient Boosting Classifier*

"learning_rate": [0.05, 0.1, 0.15, 0.2, 0.25, 0.3],
"n_estimators": [50, 100, 150, 200, 250],
"max_depth": [3, 4, 5, 6, 8, 10, None],
"min_samples_split": [2, 5, 10],
"min_samples_leaf": [1, 2, 4],
"subsample": [0.6, 0.7, 0.8, 0.9, 1.0],
"max_features": ["auto", "sqrt", "log2"]

3. *Random Forest Classifier*

"n_estimators": [50, 100, 200],
"max_depth": [None, 10, 20],
"min_samples_split": [2, 5, 10],
"min_samples_leaf": [1, 2, 4],
"max_features": ['auto', 'sqrt'],
"bootstrap": [True, False]

4. *Light Gradient Boosting Machine (LightGBM)*

"n_estimators": [50, 100, 200],
"max_depth": [-1, 10, 20, 30],
"learning_rate": [0.01, 0.05, 0.1],
"subsample": [0.6, 0.7, 0.8, 0.9, 1.0],
"colsample_bytree": [0.6, 0.7, 0.8, 0.9, 1.0],
"reg_alpha": [0.0, 0.1, 0.2, 0.5, 1.0],
"reg_lambda": [0.0, 0.1, 0.2, 0.5, 1.0]

G. *Machine Learning Algorithms*

1. *XGBoost*

XGBoost is an implementation of gradient boosting designed for supervised learning tasks, such as regression, classification, and ranking. XGBoost constructs decision trees sequentially, with each classifier f_k (x) contributing to a precise model when combined as an ensemble.

The confusion matrix in Fig. 12 gained from XGBoosting.

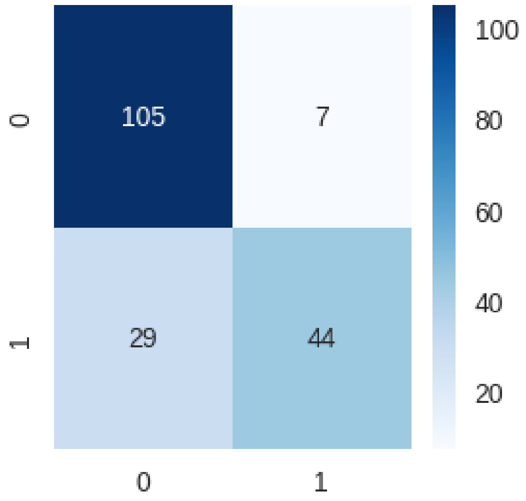


Fig. 12: XGBost classifier confusion matrix

2. K-Nearest Neighbors Algorithm

k-Nearest Neighbors (k-NN) is a non-parametric, instance-based learning algorithm that is used for classification and regression. It is simple to understand and implement, yet powerful for many types of predictive modeling problems. The basic idea of k-NN is to find the kk samples in the training dataset that is closest to a new sample and use these samples to predict the new sample. The closeness is typically measured using a distance metric such as Euclidean distance. For classification task,the most common class label among the neighbors is chosen as the predicted class:

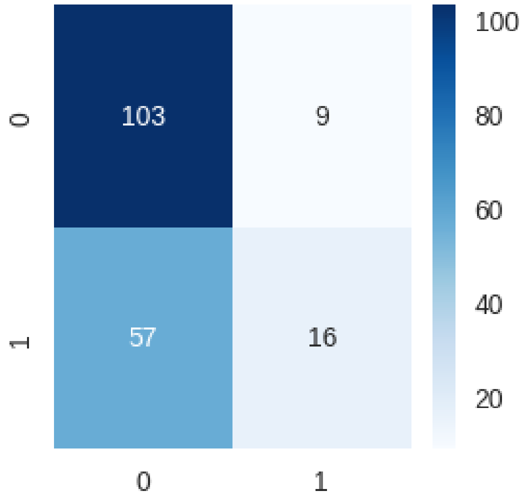


Fig. 12: KNN classifier confusion matrix

3. DecisionTree Classifier

Decision Tree is a versatile machine learning algorithm that can perform both classification and regression tasks. It works by recursively partitioning the data space and fitting a simple prediction model within each partition. The result is a tree-like structure of decisions, where each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label (in classification) or a continuous value (in regression).

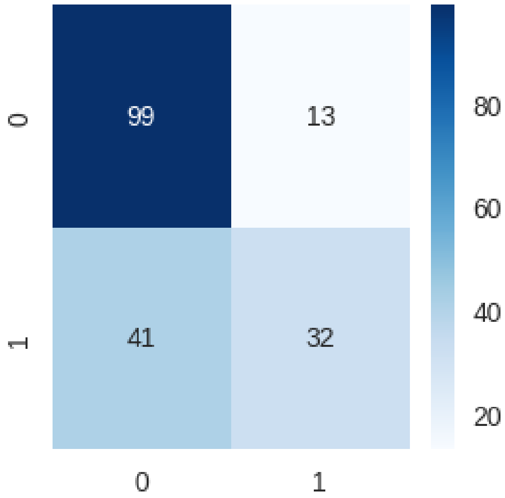


Fig. 13: DecisionTree classifier confusion matrix

4. GradientBoosting Classifier

Gradient Boosting is a powerful algorithm that minimizes overall prediction error by iteratively combining the best possible next model with previous models. This approach is particularly effective for reducing the bias error of a model.

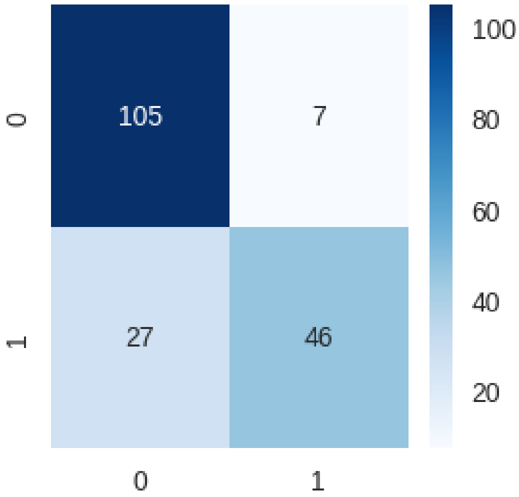


Fig. 14: GradientBoosting Classifier confusion matrix

5. AdaBoost Classifier

AdaBoost, or Adaptive Boosting, is an ensemble learning method applicable to both classification and regression tasks. This technique constructs a strong classifier by combining multiple weak classifier.

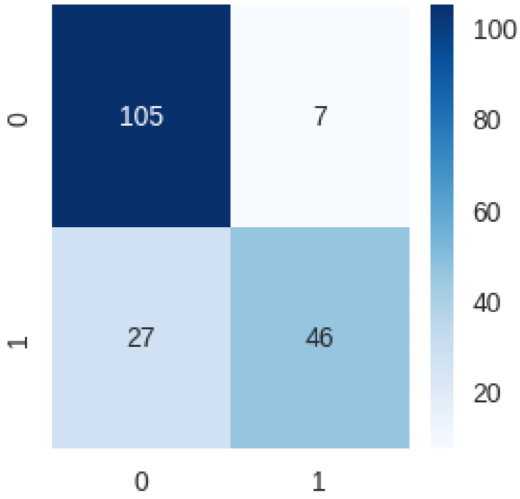


Fig. 15: AdaBoost Classifier confusion matrix

6. Light GradientBoosting Machine Classifer

Light Gradient Boosting (LGBM) is a gradient boosting algorithm designed for regression and classification tasks, known for its speed and memory efficiency. It constructs an ensemble of decision trees and is particularly optimized for large-scale datasets, emphasizing performance and resource efficiency.

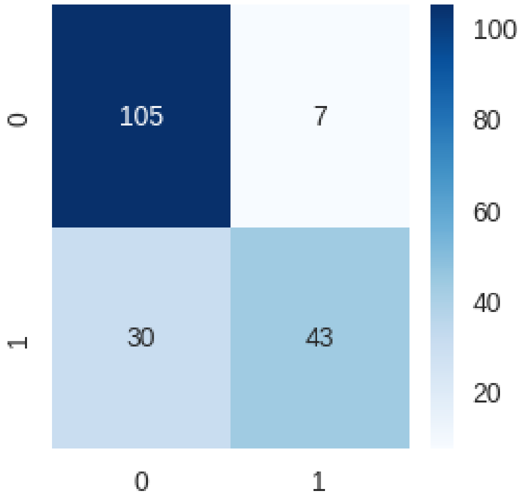


Fig. 16: LGBM Classifier confusion matrix

7. Random Forest

Random Forest (RF) is an ensemble classifier widely used for solving regression and classification problems. It leverages multiple decision trees as base classifiers, each trained on different subsets of the given dataset. The final prediction is determined by aggregating the predictions from individual trees through a majority vote or averaging process. This ensemble approach enhances the predictive accuracy and robustness of the model.Besides, Random Forest uses the out-of-bag (OOB) samples, which are not used in the training of each tree, to estimate the generalization error. The equation is same as 18.

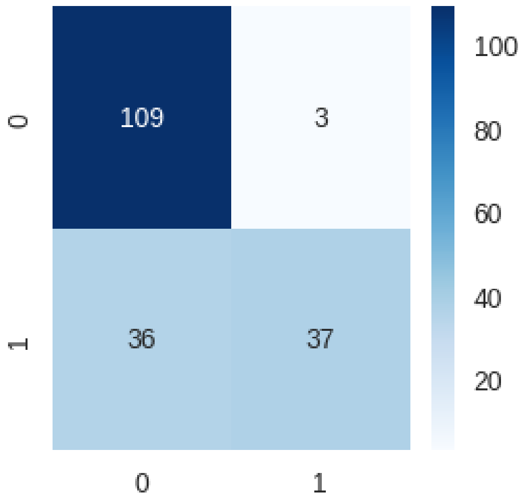


Fig. 17: Random Forest Classifier confusion matrix

H. *Performance matrix*

In this study, we assessed the performance of our machine learning models using several key metrics: precision, recall, accuracy, F1 score, and AUC ROC.

1. *Precision:* Precision, also known as Positive Predictive Value, measures the proportion of correctly predicted positive samples(TP) among all predicted positive values (TP+FP). It indicates the model's ability to accurately identify positive cases.
   Precision=TP/TP+FN

2. *Recall:* Recall, or Sensitivity, represents the ratio of correctly predicted positive samples (TP) to the total number of positive samples (TP+FN). Higher recall indicates the model's effectiveness in capturing positive cases.
   Recall= TP/TP+FP

3. *F1-score:* The F1-score combines precision and recall into a single metric, providing a balanced evaluation between the two. It ranges from 0 to 1, with higher values indicating better model performance.
   F1 score= 2{(Recall. Precision)/ (Recall+Precision)}(
4. *Accuracy:* Accuracy quantifies the overall correctness of the model by considering True Positives(TP), True Negatives(TN), False Positives(FP), and False Negatives (FN). It calculates the ratio of correctly classified samples to the total number of samples
   Accuracy = T P + T N /FP + T P + F N + T N
5. *ROC Curve:* The ROC curve illustrates the trade-off between True Positive Rate (Sensitivity) and False Positive Rate (1 - Specificity) across different classification thresholds. The Area Under the Curve (AUC ROC) summarizes the overall performance of the model across all thresholds. We used this metric to evaluate the performance of DecisionTreeClassifier, Random Forest, Gradient Boosting, AdaBoost, and XGBoost models, with their respective scores and plotted as subplots.

III. RESULT

The models' performance has been evaluated using a variety of criteria, including F1 scores, recall, accuracy, and precision.

A. *Accuracy*

One of the most fundamental measures of performance is accuracy, which is defined as the total correct predictions made over the total number of predictions, as seen in Eq. (1):

Accuracy = T P + T N /FP + T P + F N + T N (Eq:1)

where,

FP = False Positive

FN = False Negative

TP = True Positive

TN = True Negative

Accuracy is frequently expressed as a percentage (%). The accuracy scores for each classifier are shown in Table III.

TABLE III

Accuracy Results for Classification Models

Classifier	Training Accuracy	Testing Accuracy
XGBoost	84.01%	80.54%
KNN	70.33%	64.32%
Decision Tree	81.71%	70.81%
Gradient Boosting	84.28.%	81.62%
AdaBoost	82.79%	81.62%
LightGBM	84.28%	80.00%
Random Forest	**84.55%**	**78.92%**

Applying a 5-fold cross-validation to the models revealed a modest decline in performance. The accuracy scores and standard deviations are shown in Table II.

TABLE IV

Accuracy Results with 5-fold Cross-Validation

Classifier	Training Accuracy	Test Accuracy	Cross-Validation
XGBoost	84.01+/-4.74%	80.54+/-1.27%	79.29%
KNN	70.33+/-0.69%	64.32+/-5.32%	69.64%
Decision Tree	81.71+/-6.78%	70.81+/-4.12%	74.93%
Gradient Boosting	84.28+/-5.41%	81.62+/-2.75%	78.87%
AdaBoost	82.79+/-3.38%	81.62+/-2.21%	79.41%
LightGBM	84.28+/-5.82%	80.00+/-1.54%	78.46%
Random Forest	**84.55+/-5.82%**	**78.92+/-0.19%**	**78.73%**

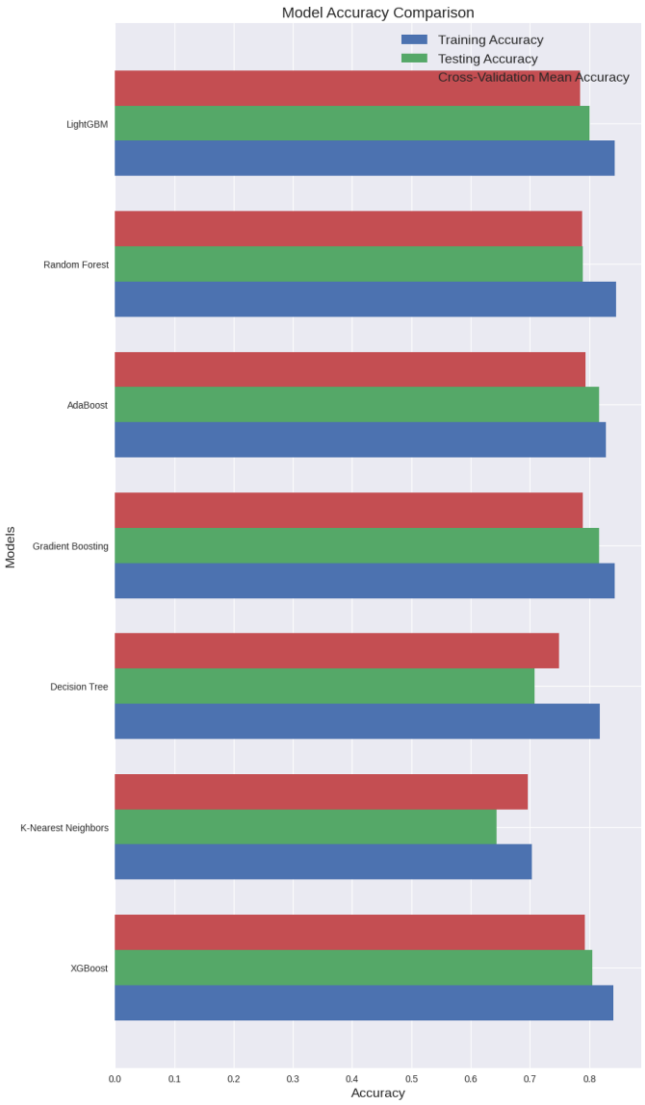


Fig. 18: accuracy and cross validation accuracy graph

B. *Other Metrics*

TABLE III

precision, recall, ROC AUC and F1 scores of every model

Classifier	Precision	Recall	F1 Score	ROC AUC
XGBoost	0.86	0.60	0.71	0.86
KNN	0.64	0.22	0.33	0.
Decision Tree	0.71	0.44	0.54	0.79
Gradient Boosting	0.87	0.63	0.73	0.86
AdaBoost	0.87	0.63	0.73	0.85
LightGBM	0.86	0.59	0.70	0.85
Random Forest	**0.93**	**0.51**	**0.65**	**0.85**

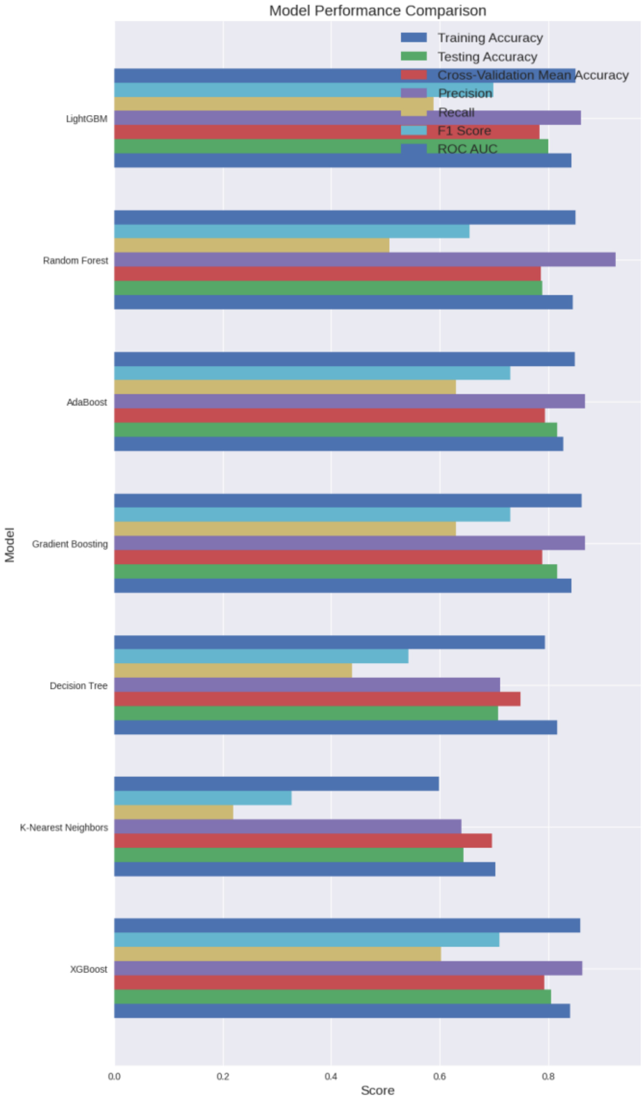


Fig. 19: Graph of overall result summary for each classifiers

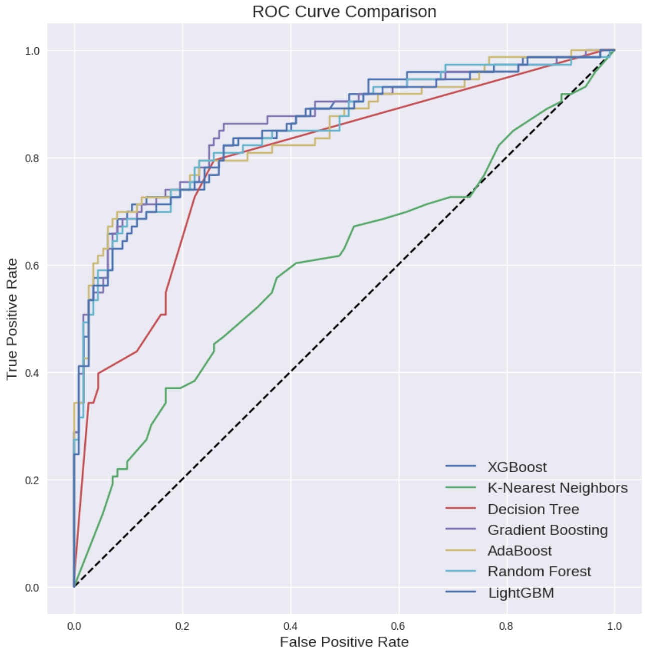


Fig. 20: Receiver Operating Characteristics(ROC) Curve of all Classifiers

C. *Explainable AI on Gradient Boost Classifire*

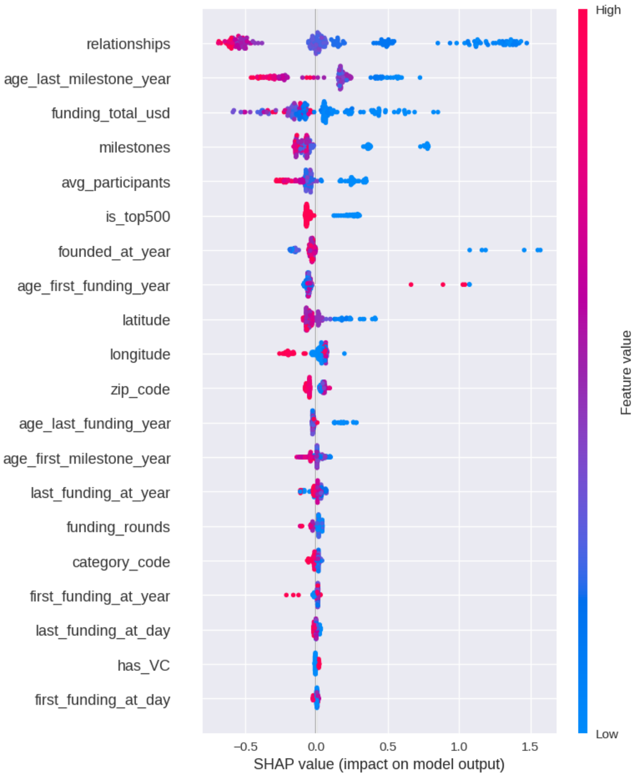


Fig. 21: shap values (GradientBoost Classifier)

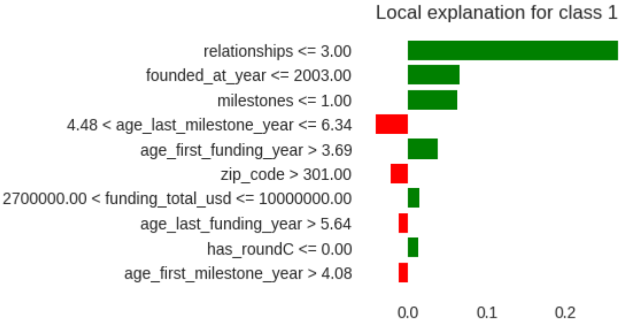


Fig. 22: predict probabilities GradientBoost Classifier

According to the EXI analysis for the Gradient Boost Classifier, relationships, founded_at_year, milestones, age_last_milestone_year, and age_first_funding_year are the most significant variables in predicting business success. Fewer relationships, being started prior to 2003, having fewer milestones, and having particular age ranges for the last milestones and first funding years are all positively connected with success. The total financing, the zip code, the year of the previous funding, and the existence of round C funding are additional significant variables. According to local explanations, there is a strong correlation between success and having fewer relationships, being an older startup, and hitting fewer milestones. On the other hand, specific age ranges for the final milestones and the initial financing years, together with features like round C funding availability, influence these forecasts.

IV. Discussion

In terms of test accuracy and cross-validation performance, the Gradient Boosting Classifier performs better than the other models, according to the evaluation results of several machine learning classifiers. The Gradient Boosting Classifier constantly shows better generalization performance when compared to other well-known algorithms like XGBoost, K-Nearest Neighbors, Decision Tree, AdaBoost, LightGBM, and Random Forest. Its best test accuracy of 81.62% and excellent cross-validation accuracy of 78.87% demonstrate its supremacy. The Gradient Boosting Classifier's comparatively minor disparity in training and test accuracies, 84.28% and 81.62%, respectively, suggests that the model does not overfit the data as much as some of the other models.

Gradient Boosting has a number of benefits, including robustness against noisy data and outliers and the capacity to handle a broad variety of data formats. Gradient Boosting creates an ensemble of weak learners, usually decision trees, and each new tree fixes the mistakes of the preceding ones, which gives the ensemble its robustness. By reducing bias and variation, this iterative procedure enhances the stability and accuracy of the model. The performance metrics demonstrate the efficacy of this strategy, since Gradient Boosting Classifier attains a high degree of accuracy without undue complexity or overfitting.

On the other hand, although exhibiting commendable results with a test accuracy of 80.54% and a cross-validation accuracy of 79.27%, the XGBoost classifier, an additional boosting technique, falls short of Gradient Boosting. Gradient Boosting has a modest advantage over other methods. This advantage may be due to implementation specifics and parameter tuning that is optimized for the particular dataset. Gradient Boosting and AdaBoost, another ensemble technique, perform similarly in terms of test accuracy (81.62%) but somewhat worse in terms of cross-validation accuracy (79.41%). AdaBoost's somewhat lower cross-validation score, however, may be explained by the fact that it is occasionally more sensitive to noisy data and outliers.

We find a notable performance gap between tree-based models such as Random Forest and Decision Tree and Gradient Boosting. Given their comparatively high training accuracy of 81.71% and lower cross-validation accuracy of 74.93%, Decision Trees—whose test accuracy is 70.81%—tend to be overfit. This is improved upon by Random Forest, an ensemble of decision trees, which averages the forecasts of several trees to reduce overfitting. Its cross-validation accuracy of 78.73% and test accuracy of 78.92%, however, are still less than those attained by Gradient Boosting. This suggests that while Random Forests work well, a more sophisticated model can be obtained by applying Gradient Boosting's strategy of gradually improving upon faults.

When compared to the advanced ensemble approaches, the K-Nearest Neighbors algorithm has the lowest test accuracy of 64.32%, highlighting its limits in handling complicated data patterns. Gradient Boosting is outperformed by LightGBM, despite being close in test accuracy at 80.00% and cross-validation accuracy at 78.46%. Gradient Boosting outperforms LightGBM in terms of pure accuracy and generalization, but LightGBM is remarkable for its quick training time and efficiency.

The Gradient Boosting Classifier's resilience and dependability are further shown by the application of ensemble approaches. Its excellent performance is further demonstrated by the post-ensemble result of 80.22%. This extensive comparison unequivocally demonstrates that the Gradient Boosting Classifier is the best option among the assessed models for obtaining high accuracy and generalizability in this dataset because to its capacity to decrease bias and variance as well as its resilience against noise and overfitting.

V. Limitations

Even while the models' performance measures show promise, it's important to recognize the inherent difficulties and constraints that come with machine learning initiatives. Here are some important things to think about:

a. *Model Generalization*: Even with excellent accuracy ratings, there is still a chance that models won't translate well to fresh, untested data, particularly if the training set isn't representative of the actual world.

b. *Data Quality and Availability*: The quality and quantity of data have a major impact on how well machine learning model’s function. Insufficient or skewed data may result in subpar model performance.

c. *Overfitting*: Models, particularly intricate ones like Gradient Boosting, have the potential to overfit the training set, producing excellent results on training data but subpar results on test data.

d. *Hyperparameter Tuning*: classifier effectiveness is sensitive to hyperparameter settings, and determining the ideal setup can be a difficult and time-consuming task.

e. *Dependency on Domain Knowledge*: Domain knowledge is frequently needed for efficient feature engineering and model interpretation, but it's not always available.

VI. Conclusion

This study highlights how machine learning may be used to forecast startup success, providing insightful information to investors, business owners, and legislators. Important variables including market size, team makeup, funding quantities, and product-market fit have been found by utilizing sophisticated machine learning algorithms. For instance, a recent study highlights the importance of integrating detailed datasets from sources like Crunchbase and patent data from the USPTO, noting that the completeness of data entries significantly impacts the accuracy of success predictions [15] . Another research effort emphasizes the use of SHAP (SHapley Additive exPlanations) values to understand the contribution of each feature, such as team size and company growth metrics, to the overall predictive model [16] .These results highlight how crucial a data-driven strategy is for making wise decisions in the startup ecosystem. Subsequent investigations ought to concentrate on integrating heterogeneous and superior datasets, crafting hybrid models that merge quantitative and qualitative evaluations, and investigating sophisticated methodologies like real-time data analytics and deep learning. It will be essential for academics, businesspeople, and legislators to work together to improve these models and confirm their applicability in actual situations. To sum up, there is a lot of potential for using machine learning to forecast startup success. Stakeholders may more effectively manage the uncertainties of the startup landscape by addressing present constraints and continuously enhancing predictive models, which will ultimately promote a more resilient and creative entrepreneurial ecosystem.

References

[1] C. Ünal, "A Machine Learning Approach Towards Startup Success Prediction," [Online]. Available: https://ideas.repec.org/p/zbw/irtgdp/2019022.html.

[2] D. M. R. Hasan, "HUMAN-FIRST AI LAB (HAL 2.0)," UNIVERSITY of NEBRASKA–LINCOLN, [Online]. Available: https://engineering.unl.edu/hasan/machine-learning/. [Accessed 01 06 2024].

[3] J. Ali-Yrkkö, A. Hyytinen, and M. Pajarinen, "Does investing in foreign subsidiaries pay off? Evidence from Finland," International Business Review, vol. 14, no. 3, pp. 294-313, 2005.

[4] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," The Journal of Finance, vol. 23, no. 4, pp. 589-609, 1968.

[5] K. Gugler and K. A. Konrad, "Merger control regulation as a barrier to entry," International Journal of Industrial Organization, vol. 20, no. 10, pp. 1403-1423, 2002.

[6] G. V. Karels and A. J. Prakash, "Multivariate normality and forecasting of business bankruptcy," Journal of Business Finance & Accounting, vol. 14, no. 4, pp. 573-593, 1987.

[7] W. W. Meador, P. E. Church, and L. G. Rayburn, "Development of prediction models for horizontal acquisitions," Review of Financial Economics, vol. 5, no. 2, pp. 77-87, 1996.

[8] S. Ragothaman, V. Naik, and R. Ramakrishnan, "Predicting corporate acquisition targets: A neural network approach," Journal of Financial and Strategic Decisions, vol. 16, no. 1, pp. 1-9, 2003.

[9] T. Liang and D. Yuan, "Predicting the Success of High-Tech Startups," IEEE Transactions on Engineering Management, vol. 59, no. 4, pp. 698-713, 2012.

[10] Y. Xiang, J. Neville, and M. Rogati, "Relational learning with one network: An inductive logic programming approach," in Proc. 2012 International Conference on Machine Learning, 2012, pp. 232-239.

[11] L. Wei, J. Tian, and Y. Zhang, "Combining Structured and Unstructured Data for Predictive Analytics," Journal of Big Data Research, vol. 5, no. 2, pp. 112-121, 2009.

[12] P. Gujral et al., "Prediction of Startup Success Using Machine Learning Algorithms," International Journal of Computer Applications, vol. 72, no. 6, pp. 20-24, 2013.

[13] N. Radja and S. Emanuel, "Systematic Analysis of Machine Learning Algorithms for Predicting Startup Success," Journal of Business Research, vol. 68, no. 12, pp. 2562-2570, 2015.

[14] S. Kadhm et al., "Clustering Analysis for Enhancing Prediction of Startup Success," Journal of Data Mining and Knowledge Discovery, vol. 30, no. 4, pp. 711-732, 2016.

[15] K. Communications, "Scientists develop AI to predict the success of startup companies," phys.org, 7 SEPTEMBER 2021. [Online]. Available:

https://phys.org/news/2021-09-scientists-ai-success-startup-companies.html. [Accessed 1 June 2024].

[16] omdena, "Predicting the Important Factors of a Successful Startup using SHAP Value," omdena, 12 July 2021. [Online]. Available: https://www.omdena.com/blog/startup-prediction. [Accessed 2024 June 2024].