Utilizing Base Machine Learning Models to Determine Key Factors of Success on an Indian Tech Startup

Dishan Bhattacharya *

Posted Date: 4 January 2024

doi: 10.20944/preprints202401.0368.v1

Keywords: machine learning; startups; indian tech industry; responsible ai

Article

Utilizing Base Machine Learning Models to Determine Key Factors of Success on an Indian Tech Startup

Dishan Bhattacharya

Downingtown STEM Academy, 335 Manor Ave, Downingtown, PA 19335;
24dbhattacharya@student.dasd.org

Abstract: Startups are playing increasingly influential roles in the technology sector of the world. India has been a rapidly growing economy and hosts over one hundred thousand total startups. Investors have been increasingly investing in the Indian technology sector. However, the Indian startup ecosystem is different to the American startup ecosystem and requires a separate analysis to determine important influences for their success. Through this research with responsible machine learning, entrepreneurs will be empowered to better understand how to successfully raise their company. I collected data from Crunchbase and defined a successful startup as one who has acquired another company, was acquired, or went public. I used Random Forest, XGBoost, LightGBM, and CatBoost to predict the success of the startups. To determine the most important factors, I used the feature importance tools provided by the models. I compared these results and found that the time taken between the founding and first funding of the company, commonly referred to as seed lag, was the most pivotal factor to every model's prediction of success.

Keywords: machine learning; startups; indian tech industry; responsible ai

Introduction

The aim of this project is to equip entrepreneurs with the tools to successfully build a startup in the Indian technology industry. It is also for startups operating within this demographic to better understand factors of success. Investors will also benefit when determining which startups to invest in. In analyzing startup prediction, utilizing past research is very helpful to create effective and accurate models. The preprocessing of the data is similar to previous literature but can also provide new insight into important features that greatly affect the models. Many of the challenges in this field of study arise while preprocessing the data to ensure validity of the results.

Background

India has been an explosive region for tech startups and hosts the 3rd most startups in the world. I want to learn more about how tech startups can contribute to the economic state of my home country and how various factors such as the number of funding rounds can affect the success of the startup. India and China host more than 60% of startup funding, according to CrunchBase. Thousands of startups in the tech sector have come out of India in recent years, providing venture capitalists with numerous opportunities to invest in an emerging market. In 2023, there is a combined valuation of all Indian startups upwards of $500 bn. It is expected that there will be more than approximately 180,000 startups and over 22,000 startup investors in India by 2030 [1]. As the country's population gains access to the internet, it is proportionally expected that they will have greater access to resources to build startups.

High failure rates make startups incredibly risky to invest in but this model can provide investors in confidence for better ROI. Success prediction can provide valuable insights and allow startups to understand what to prioritize. By understanding how a startup can succeed in the domestic industry, entrepreneurs can gain insight into benchmarks startups should meet to be sustainable in a global ecosystem.

Initially, I explored KNearestNeighbors, Random Forest Classifiers, and Support Vector Classifiers. These models yielded accuracy levels ranging from the 0.4 to 0.6 range. This demonstrates that the models used must be tuned to improve the accuracy of the model. It also signified numerous improvements to the data must be made during the data cleaning and preprocessing stages. The Random Forest Classifier performed the best, leading me to look into other similar models. When studying research literature regarding financial data and startup success, often a name which arises is XGBoost, a supervised learning model [2]. Upon further exploration, I learned that XGBoost's open-source model is a regularizing gradient boosting framework often used to measure the success of a startup by continuously adding new trees and combining them with previous trees to finalize a stronger prediction [3].

Dataset

I pulled data from Crunchbase regarding company information as well as funding information for the company. I pulled the top 1000 companies from each year on Crunchbase between 1-1-2005 and 12-31-2019 and funding data for the top 1000 transactions within the same time frames. All of the tags used are Hardware, Information Technology, Software, Internet Services. Artificial Intelligence, and Messaging and Telecommunications. I then concatenated all of the company information files together and the funding information files together. There was no instance in which I did not collect all of the transactions in the specified time frame. I pulled the companies by year until there were more than 1000 startups each year. Then I pulled between each time there were 1000 startups registered, which was the maximum number of companies for which data could be pulled. This meant I had pulled as many of the recorded transactions available on Crunchbase between 2005 and 2019 in the specified demographic. A limitation of my dataset is that I did not account for every startup within this demographic; the explosive growth in startups within the Indian tech industry meant there were years where over 1000 startups were registered on the first of January, the start of the year.

Through analyzing past research, I set my definition of successful startup to be one which has either made acquisitions, was acquired, or went public. I determined if a company went public if they had data in their IPO column. When pulling data, I extracted dozens of features. However, an abundance of features can lead to overfitting. Thus, the model must be simplified to perform better when predicting on new data. To validate the assumption that I had too many features, I ran baseline classifier models using Random Forest Classifier, XGBoost, LightGBM, and CatBoost.

In order to have more accurate results, I underwent thorough preprocessing to eliminate redundant or otherwise unnecessary data. A limitation of many of the models I used is that they are unable to process missing data. To handle this, I created graphs to visualize where data was missing and if there were any correlations between if there was existing data or not in the numerous features. I did this through the *missingno* library. From these visualizations, I was able to determine if I could remove redundant features which did not contribute to whether a company was successful or not. For example, I removed the features 'Stock Symbol URL,' 'Stock Symbol,' and 'Stock Exchange.' My definition of success was built upon if there was data in specific columns. The mentioned columns were only filled if the company went public, or succeeded. By deleting these columns, I ensured that I would only be analyzing data preceding the success of a company. I also removed columns like 'Headquarters Location' and 'Headquarters Location.' Because all of the companies are located in the same country, the region was the same for all companies. There is a contention to be made that the city in which a startup is founded affects its success; however, analyzing this would require me to geographically map all of the unique points and create heatmaps for frequency per city. This could create bias towards major cities. Another limitation is that they are unable to process data in strings or floats. To bypass this, I encoded all of my data to integers. Many of the floats in my dataset were already equivalent to integers and those which were not were rounded to the nearest integer. The only strings I encoded were the funding type, the range of the number of employees, and the revenue

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2024 doi:10.20944/preprints202401.0368.v1

range. An alternative I could have pursued is to average the number of employees and amount of revenue in the range, which would have showcased a logarithmic scale.[1]

When training and testing the data, I employed a 0.7/0.3 training/test split. Because my data is unbalanced, closely reflecting the low likelihood of a startup succeeding in the real world, I wanted to ensure that I could test as much data as possible while maintaining a large training dataset.

Methodology

This project will determine the important factors of success by using different gradient boosting models. Success is defined as if a company was acquired, made acquisitions, or became public through an IPO. In this project, I explored Random Forests, XGBoost, LightGBM, and CatBoost.

Random Forest

Random Forest classification models use multiple decision trees and ensemble learning methods to obtain a final classification [4]. It uses bagging and feature randomness in its construction of decision trees and has three primary hyperparameters (node size, number of trees, and number of features). Using Random Forests will enable me to reduce the overfitting in my models due to the incorporation of uncorrelated trees in the decision model.

XGBoost

Recognized by CERN as a strong classifier model, XGBoost uses ensemble learning and multiple learners to create a stronger result from the culmination of several models [5]. XGBoost differs from similar models due to its regularization processes, which prevent overfitting by using Lasso and Ridge Regression (L1 and L2). XGBoost also uses a weighted quantile sketch, dictating it to be effective at processing and evaluating weighted data.

LightGBM

LightGBM, another gradient boosting framework, boasts greater efficiency and speed for large datasets compared to XGBoost [6]. The framework uses Gradient-based One Side Sampling; the model gains more information through larger gradients and randomly drops smaller gradients to keep as much information as possible. It also uses a number of parameters to control binning and the depth to which the model goes from branches and leaves; these include max_depth and max_bin.

CatBoost

CatBoost also uses gradient boosting with categorical data. It is different to other popular gradient boosting models as it uses ordered encoding to encode categorical features and bypasses many steps in preprocessing necessary to run most models [8]. It uses the data processed before it to calculate a substitute value for the feature. CatBoost also uses symmetric trees, meaning that each decision node in the same depth level uses the same split condition.

Evaluation Metrics

Each model uses a different measure to determine the importance of the feature to the model.

Below is the method of determination for each model used. Gini importance is the mean change in Gini score [8]. Gain importance is the relative impact of the feature on each tree of the model. Below is a generated image to illustrate which measure I used in my models.

[1] https://github.com/DishanBhattacharya/startup_pred/blob/main/GitHub%20Notebooks/12345.csv, file with dataset used for models

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2024 doi:10.20944/preprints202401.0368.v1

Classifier Used and Corresponding Calculation Method

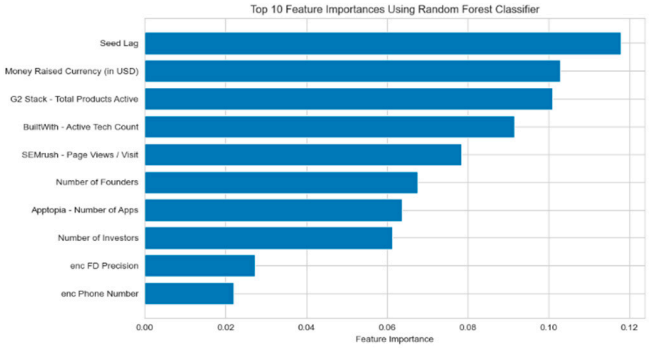
Random Forest	Gini Importance
XGBoost	Gini Importance
LightGBM	Gain Importance
CatBoost	Gini Importance

[9][10][11][12]

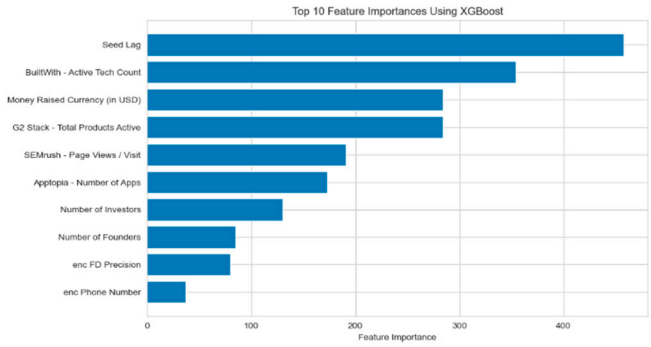
Results and Discussion

I determined the importance of the features through the built-in functions in the models. The results are shown below.

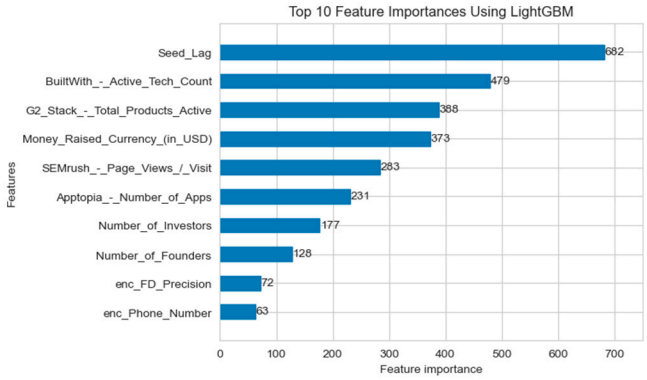
Random Forest



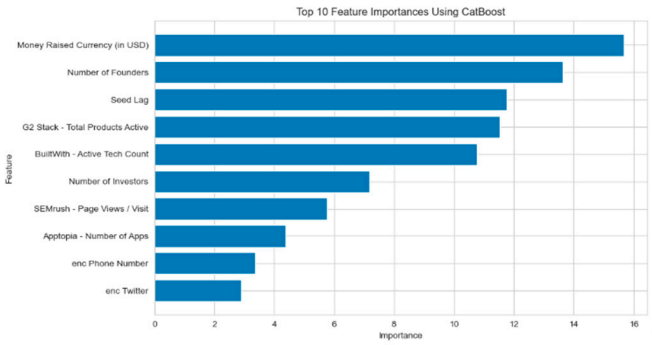
XGBoost



LightGBM



CatBoost



[2]

Many of the most important features are similarly considered by all four of the tested models, thus increasing the validity of these results. In accordance with previous literature predicting the success of a startup, seed lag is maintained as the most important feature for all models but CatBoost, where the money raised during the funding round is the most important feature. Other commonly important features include the number of investors, and the total money raised during the funding round. Surprisingly, the number of founders showed a sizable effect, indicating the efficacy of having the number of founders falling under a minimized range.

---

[2] https://github.com/DishanBhattacharya/startup_pred/blob/main/GitHub%20Notebooks/paper.ipynb, file used to determine results

The results from using CatBoost showed the greatest deviation from other models. The model highlights that the money raised from the funding is the most important feature affecting the prediction of the model. This is most similar to the results from the Random Forest Classifier, which places this feature as the second most important for prediction. The base XGBoost and LightGBM models produced the most similar results; the same features were in the top 10.

In consideration of potential sources of error, there can be errors from how I sourced my data. In addition, while I attempted to standardize the measurement of data, the different models used different measurement systems to determine the importance of the features.

Conclusions

Three of the four models dictated that the time taken from the formation of the company to the first time said company is funded is the most important factor in determining the outcome of a startup in the Indian technology sector, whether it succeeds or fails. The results of this can influence how Indian technology startups can foster a better environment to align themselves to prioritize important pieces of their company which can thus stimulate success. While none of the one-hot encoded features were present in any of the models' top features, further exploration can be made looking into their effect on the performance on the base models.

Acknowledgments: Thank you to Tyler Poore for assisting me through the processes to more effectively analyze the data and providing engaging support.

References

1. Inc 42. (2023, August 15). *The State Of Indian Startup Ecosystem Report 2023*. Inc42 Media. https://inc42.com/reports/the-state-of-indian-startup-ecosystem-report-2023/
2. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. https://arxiv.org/pdf/1603.02754.pdf
3. AWS. How XGBoost Works - Amazon SageMaker. (n.d.). Docs.aws.amazon.com. https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost-HowItWorks.html
4. IBM. (n.d.). What is Random Forest? | IBM. Www.ibm.com. https://www.ibm.com/topics/random-forest
5. Understanding the Math behind the XGBoost Algorithm. (2018, September 6). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/
6. LightGBM (Light Gradient Boosting Machine). (2020, July 15). GeeksforGeeks. https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/
7. How CatBoost algorithm works—ArcGIS Pro | Documentation. (n.d.). Pro.arcgis.com. https://pro.arcgis.com/en/pro-app/latest/tool-reference/geoai/how-catboost-works.htm
8. Codeacademy Team. (n.d.). Feature Importance. Codecademy. https://www.codecademy.com/article/fe-feature-importance-final
9. Dubey, A. (2018, December 15). Feature Selection Using Random forest. Medium. https://towardsdatascience.com/feature-selection-using-random-forest-26d7b747597f
10. CatBoost Documentation. (n.d.). get_feature_importance. Catboost.ai. Retrieved January 2, 2024, from https://catboost.ai/en/docs/concepts/python-reference_catboostclassifier_get_feature_importance
11. Machine Learning Mastery, & Brownlee, J. (2016, August 30). Feature Importance and Feature Selection With XGBoost in Python. Machine Learning Mastery. https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/
12. LightGBM Feature Importance and Visualization. (2023, October 13). GeeksforGeeks. https://www.geeksforgeeks.org/lightgbm-feature-importance-and-visualization/