Prospect To Prosperity: Unraveling Startup Success with ML Predictions

BANA 273 | Final Project Report

Team 9B

-

Husain Mandliwala

Nikhil Koka

Peiqi Huang

Tanaya Ranade

Wei-Shiang Wang

Executive Summary

This project aims to utilize machine learning models and processes for unraveling the likelihood for startups to acquire success. Considering the vast startup landscape in today's industry, there are a wide range of factors that influence success or failure and the same has been visually depicted through exploratory data analysis. In order to process such huge variability in the market effectively, this project builds three classifier models in order to understand whether machine learning models can be effectively used to predict success. Initially, the sourced dataset was cleaned for inconsistencies and reformatting needs. Thereafter, the Naive Bayes, Decision Tree and Logistic Regression models were built and optimized with preprocessing measures like cross-validation and resampling techniques. Through an extensive model evaluation, first within models built under each classifier and then amongst the best of them, an optimal model was chosen based on measures of overall accuracy, F1 score and stratified accuracy for predicting success or failure. In conclusion, the logistic regression model is chosen as it provides the most optimum scores on all key areas of performance resulting in 78.7% accuracy, 78.8% F1 or reliability, 82.7% accuracy for predicting success and 73.3% accuracy for predicting failure. Lastly, the report also provides multi-perspective insights and recommendations for executives and investors based on the models' results as well as additional analyses.

Table of Contents

Introduction

Market Context

A startup, initiated by an entrepreneur, is a venture aimed at exploring, refining, and confirming a scalable economic model. According to data from the Small Business Administration (SBA), entrepreneurship encompasses all new businesses, including self-employment and those not intending formal registration. In contrast, startups specifically denote emerging enterprises with aspirations for significant growth beyond a single founder. Research by the U.S. Bureau of Labor Statistics (BLS) indicates that startups face considerable uncertainty, with high rates of failure, but a minority among them manage to achieve success and wield significant influence. Approximately 20% of new businesses fail during the first two years of being open, 45% during the first five years, and 65% during the first 10 years. This challenging nature of startups is typically heavily influenced by factors such as market dynamics, competition, and financial constraints.

Notably, certain startups attain unicorn status, signifying privately held companies valued at over US$1 billion. According to data compiled by CB Insights, as of the latest available figures in 2022, there were over 800 unicorns globally, with industries such as technology, finance, and healthcare prominently represented. This signifies a remarkable achievement in the competitive landscape of startups. Hence, startups play a pivotal role in the business ecosystem, with their trajectory marked by uncertainty, high failure rates, but also the potential for significant success and influence, as demonstrated by the emergence of unicorn startups.

Problem Statement Development

The objective of this project is to predict whether a startup, currently operating in the market, would be successful or not based on a number of diverse factors. A startup will be classified as 'successful' if either of the three events have occurred, signifying that it is able to generate considerable revenues and profits:

- The startup underwent a Merger and Acquisition by a parent company
- The startup achieved unicorn status
- The startup achieves public status by offering an IPO.

Thus, the problem statements that this project aims to resolve are as follows:

- Which machine learning model can best predict whether a startup will be successful or not?
- How can the machine learning models predicting startup success be optimized for most accurate performance?
- Which factors are important to consider while predicting the success of a startup?

Methodology

Dataset Acquisition & Description

The dataset used for the purpose of this project is titled 'Startup_Success_Prediction_Dataset' and has been sourced from Kaggle. The dataset contains 922 unique rows and 49 columns, each row representing a distinct startup. There are missing values in several columns, such as "Unnamed: 6," "Closed At," "Age First Milestone Year," and "Age Last Milestone Year." The missing values in the "Closed At" column are attributed to the ongoing acquisition status of the company. Key information about the basic details and crucial funding details for each startup has been included in the dataset.

Data Cleaning Measures

While the dataset is very well organized, it needed certain cleaning procedures.

Dealing with Missing & Null Values

Records with missing or NA values in the attributes for age_first_milestone_year and age_last_milestone_year were replaced with 0.

Reformatting Data

The 'unnamed 0' column was an identifier and was renamed as the s_no denoting serial number.

The values for age_first_milestone_year, age_last_milestone_year, age_first_funding_year, age_last_funding_year and founded_year were converted from string to DateTime index formats for easier processing.

Dummy Encoding

Dummy encoding was conducted for the categorical variables of status (target variable), founding city, founded_year, first_funding_year, and last_funding_year.

Removing Irrelevant Data

The following attributes were removed from the modeling dataset since they were either irrelevant for predicting the success status of the startup, duplicated information or in an incompatible format for modeling.

Removed Attributes: Unnamed, latitude, longitude, zip code, id, company name, labels, state code (already represented via state dummy variables), closed at (not relevant to predicting success), category code (already represented through category dummy variables), object_id, last funding at and, founded_at (already represented through founding city).

Post data cleaning, the final dataset for modeling contained 922 rows and 302 features, including dummy encoded attributes. The table below lists the dependent variable as well as some key independent variables chosen after data cleaning.

Variable Name	Description
status	**Dependent Variable**: converted to binary 1 = acquired / successful & 0 = closed / failure
age_first_funding_year	Age of the startup when it received its first funding
age_last_funding_year	Age of the startup when it received its last funding
age_first_milestone_year	Age of the startup when it achieved its first funding
age_last_milestone_year	Age of the startup when it achieved its last funding
funding_total_usd	Total amount raised by the startup (in USD)
funding_rounds	Number of funding rounds successfully passed by the startup
Categories	Dummy variables for industry category of the startup
States	Dummy variables for founding states of the startup
Cities	Dummy variables for founding cities of the startup
Average Participants	Average employees of the startup over the period
is_top_500	Has the startup featured as a top 500 company?

Table 1. Key dependent & independent variables derived after data cleaning process

Exploratory Data Analysis: Insights & Implications

Considering the vast dimensionality of the chosen dataset in terms of the attributes to be evaluated for predicting whether a startup will be successful or not, this report conducts an exploratory data analysis to understand the overall trends in the data. Such analysis will guide the decision for selecting and building the machine learning models for success prediction.

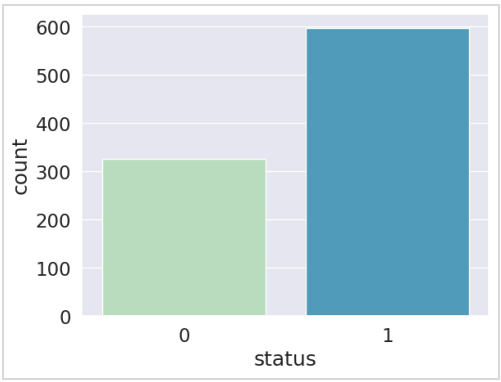

Figure 1. Class Distribution for Status - Is the Startup Successful or Not?

Figure 1 above establishes a preliminary classification understanding regarding the target variable. Specifically, it shows that the number of successful startups in this dataset is more (65% distribution) than the number of startups that failed (35% distribution), implicating an imbalance.

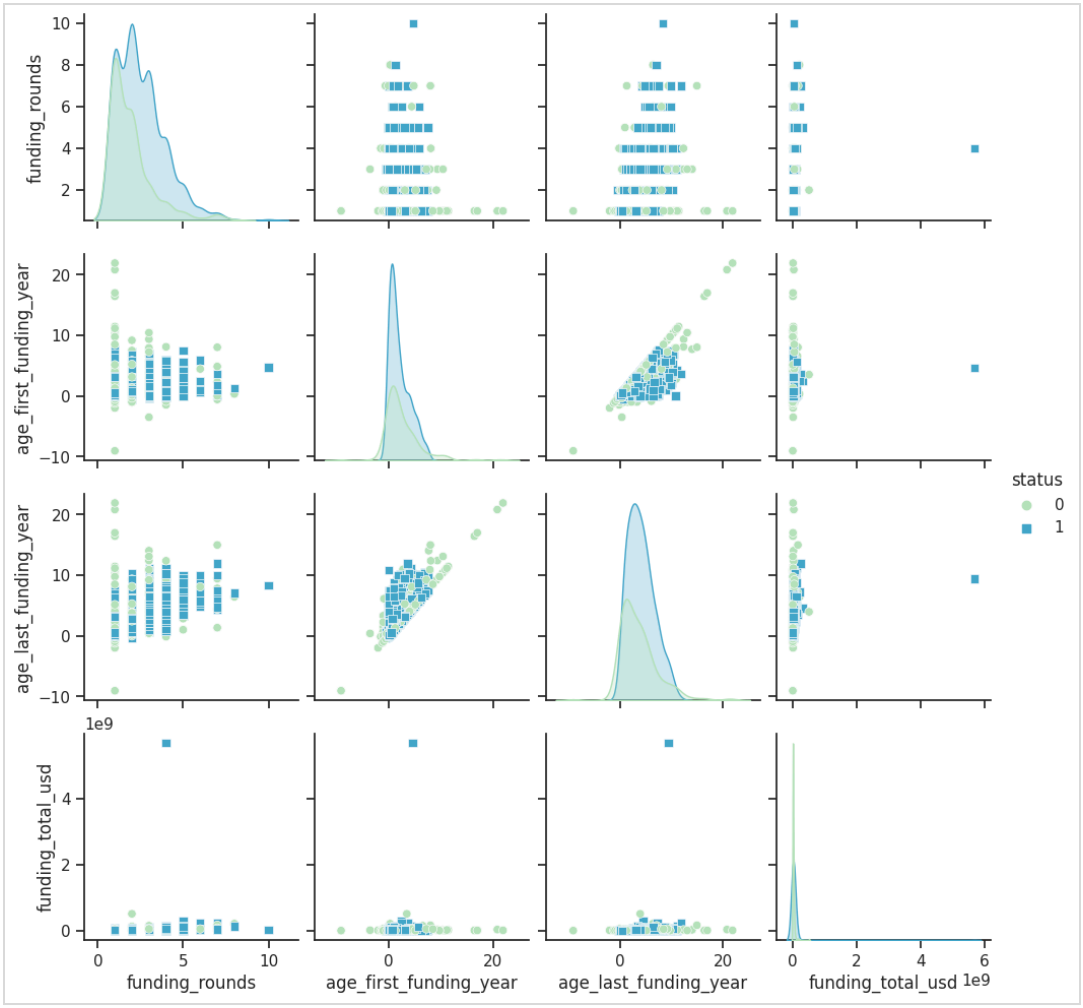

Figure 2. Pairplot between key numeric independent variables

In Figure 2 above, an overview of relationships has been depicted between the key independent variables present in the dataset, especially those that are of numeric nature. While this graph lays a great overall perspective on the pairwise relationships in this data, further in-depth exploration must be conducted to better understand the data.

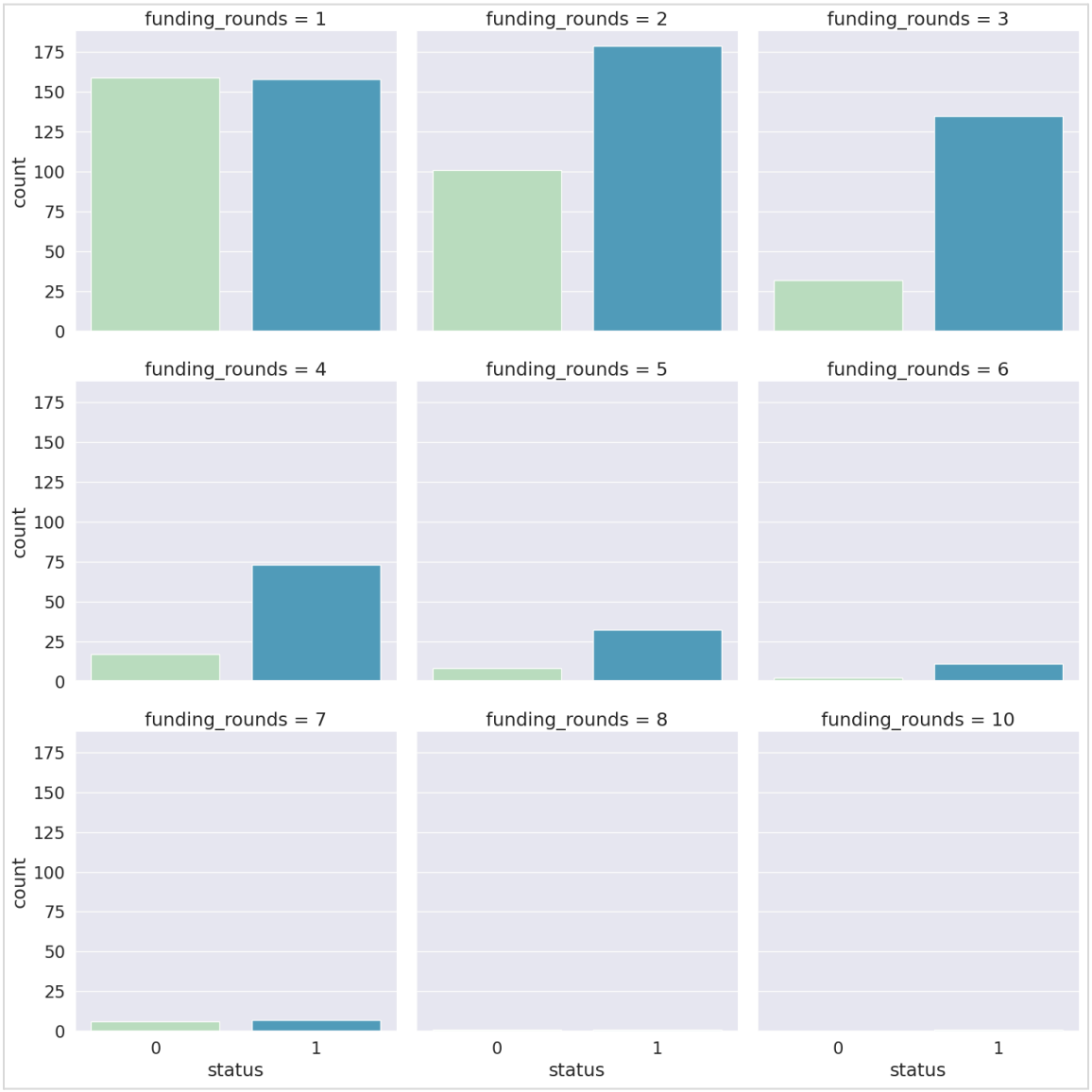

Figure 3. Is the number of funding rounds relevant for evaluating startup success?

A matrix of bar charts from Figure 3 above helps in delineating whether there can be any relationship between the chance of success for a startup and the number of funding rounds it passes through. Naturally, it can be assumed that a more successful startup would pass through more funding rounds. However, that might not be the case or a generalizable trend as seen from Figure 3 above. This is because considering the classification imbalance, there is a relatively decent number of startups that have persisted through multiple funding rounds, till 7 at the most.

Yet, they had to be closed and thus, failed at succeeding as a new company. Thus further analysis must be conducted as the number of funding rounds alone cannot predict startup success.

Figure 4. Age of startups during their first funding vs last funding - did this impact success?

Figure 4 compares successful versus unsuccessful startups on the parameters of their respective ages when they received their first versus last funding. For startups that failed, a majority of them received their first funding between ages of 0-2 years and they received their last funding between 0-5 years of founding the startup. This can indicate that the initial years are extremely crucial for any startup to become successful. Aligning with this insight, the graphs for successful startups show that they received their first funding between ages of 0-4 years and their last funding between ages of 1-10 years of founding the startup. This shows that successful startups were able to persist for longer in the market and within almost a decade of being founded, they were able to either get acquired, list publicly or make massive revenues.

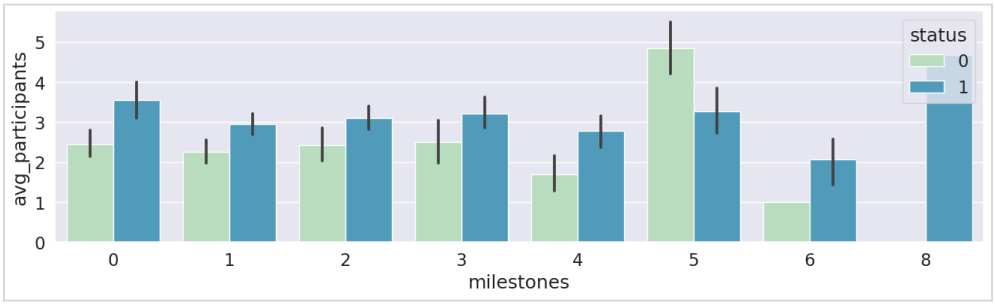

Figure 5. Does a bigger company always imply more success?

Lastly, Figure 5 evaluates whether a startup's size could be related to the number of milestones it achieves. The graph shows that for most successful startups, the organization size remained stable even as they achieved more milestones. However, there was an erratic pattern among failed startups, especially for the later milestones where there were sudden rises and falls in the company size. Thus, it's interesting to see and further analyze how all these factors will come together to predict the success of a startup based on individual, technical and market factors.

Analytical Models

In this project, 3 central models have been deployed for creating predictions and classifications for success determination of startups, namely

1. Naive Bayes Model
2. Decision Tree Model
3. Logistic Regression Model

All models utilized a 70-30 train-test split, wherein 70% of the overall data was randomly selected for machine learning while 30% of the overall data, remaining after training, was considered for model validation.

1. Naive Bayes Model

The initial Naive Bayes classifier underwent training on a dataset that resulted from meticulous data cleaning. What distinguishes this model from decision tree and logistic regression is the deliberate transformation of 'funding_total_usd' into 10 bins, a strategic measure aimed at alleviating complexities stemming from an excessive number of categories.

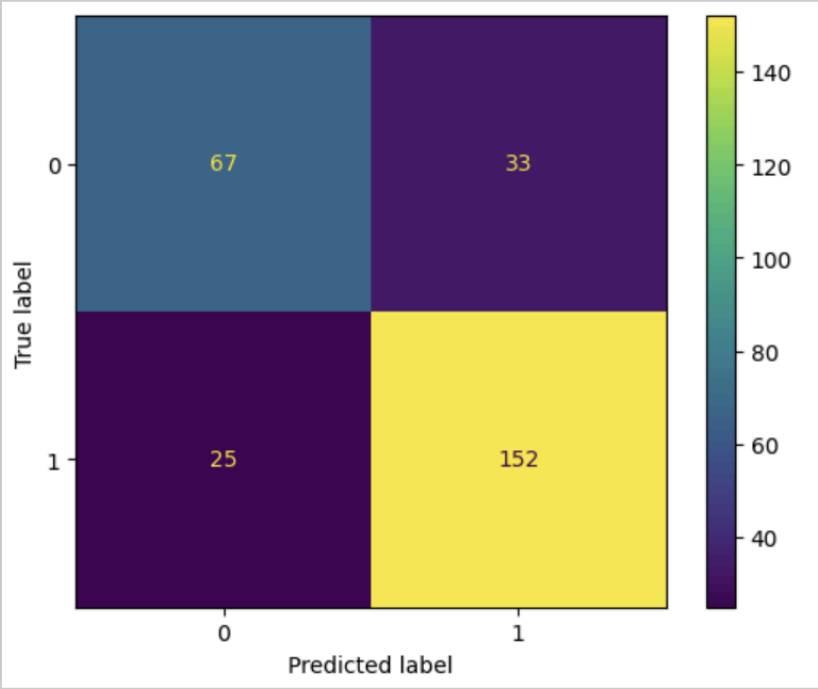

Figure 6. Confusion matrix for initial Naive Bayes classifier before data preprocessing

Accuracy	Accuracy (success)	Accuracy (failure)	F1 score
79%	86%	67%	80%

The respective measures of performance for the unprocessed Naive Bayes classifier, respectively, have decent metrics. However, the stratified accuracy for predicting startup success needs to be

improved. Furthermore, the confusion matrix shows that the model is overfitting the data and hence, certain data processing measures can be taken to tackle these issues.

Data Pre-Processing Measures

1. Grid Search CV (hyperparameter optimization)

To improve model accuracy and address class imbalance, a Grid Search approach was applied to fine-tune the hyperparameters of the Multinomial Naive Bayes classifier. The alpha parameter was specifically targeted, with a grid of values like [0.1, 0.5, 1.0, 1.5, 2.0]. Grid Search systematically explores hyperparameter combinations, ensuring a more exhaustive search and preventing overfitting. A 10-fold cross-validation strategy was employed during the Grid Search to robustly evaluate model performance while maintaining class distribution balance.

The accuracy metric guided the Grid Search, aiming to find the optimal hyperparameter configuration. The resulting model, with the best hyperparameters, was then evaluated on the entire dataset. The analysis included accuracy, confusion matrix and F1-score to assess the tuned model's effectiveness

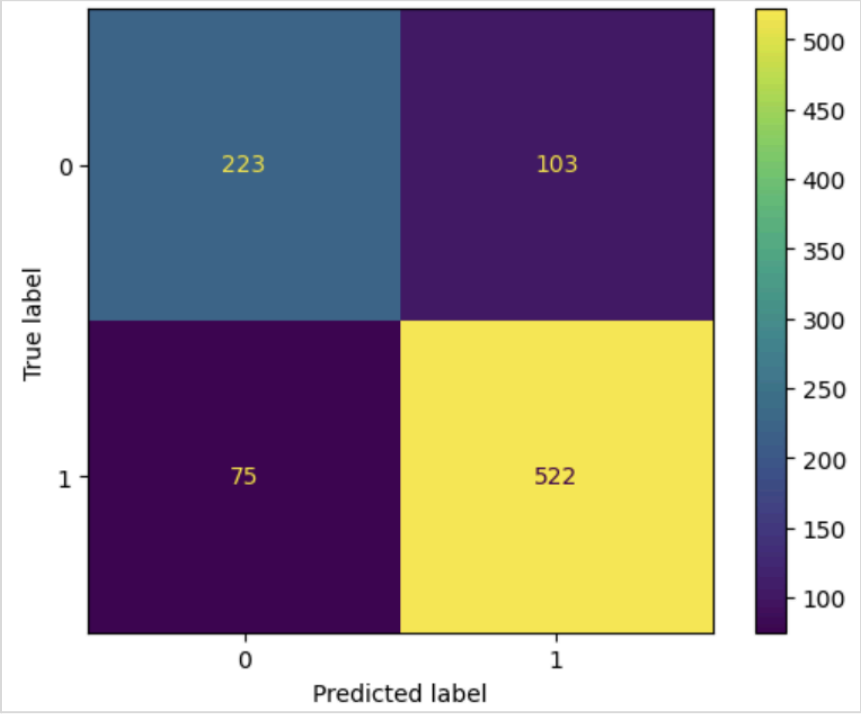

Figure 7. Confusion matrix for initial Naive Bayes classifier after data preprocessing using Grid Search cross-validation

Best Accuracy	Accuracy (success)	Accuracy (failure)	F1 score
78%	68%	87%	85%

The cross-validation results show notable changes post-optimization. Following Grid Search, overall accuracy dropped slightly to 78%. Notably, success accuracy decreased to 68%, while failure accuracy increased to 87%. Despite a small dip in overall accuracy, the model's adjustment appears to have shifted focus towards improving failure predictions, at the expense of a decrease in success accuracy. These changes may reflect a fine-tuning of the model's balance between successful and failed cases to achieve a more even performance.

2. SMOTE

In this dataset, 65% of the data is classified as 'successful,' while 35% of the data is classified as 'failure.' Therefore, SMOTE is used to balance distribution of samples across different classes.

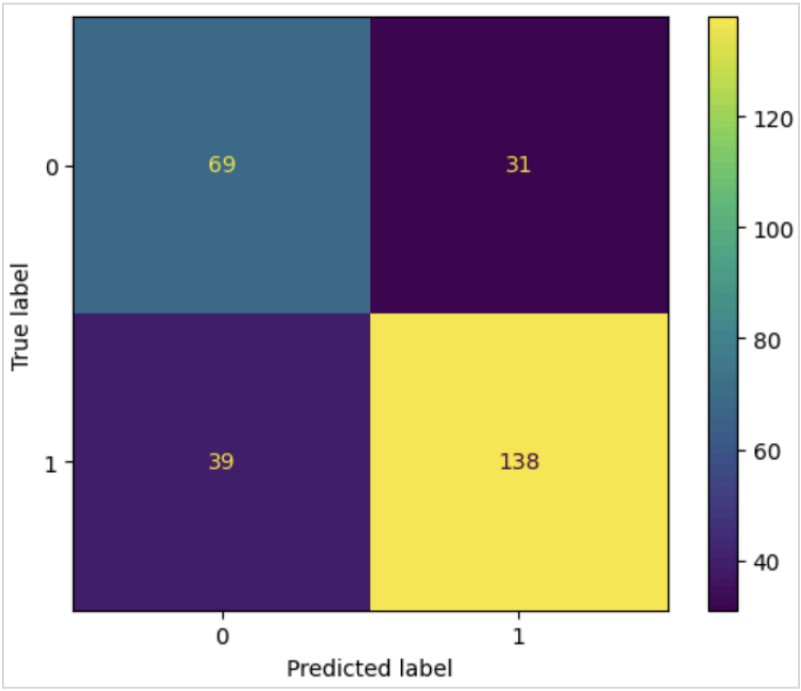

Figure 8. Confusion matrix for Naive Bayes classifier after data preprocessing using SMOTE

Accuracy	Accuracy (success)	Accuracy (failure)	F1 score
69%	78%	67%	80%

After this method of data pre-processing, the accuracy, stratified accuracy and F1 measures further declined. Meanwhile, the issue of data overfitting was also not satisfactorily resolved, as is seen from the confusion matrix above.

Intra-Model Performance Evaluation

Based on the following table, it can be observed that neither cross-validation nor SMOTE has significantly improved the accuracy or reliability of this model. Furthermore, analyzing the respective confusion matrices also indicates that the overfitting issue was not sufficiently resolved. Thus, between the three models developed within the Naive Bayes classifier, the one derived post stratified KFold cross-validation provides the best performance in terms of accuracy as well as F1 measure.

Model	Accuracy	Accuracy (Success)	Accuracy (Failure)	F1 Score
Unprocessed	79%	86%	67%	80%
Cross-validation	**78%**	**84%**	**65%**	**84%**
SMOTE	69%	78%	67%	80%

Table 2. Intra-model comparison for Naive Bayes classifier before and after data preprocessing

Feature Importance

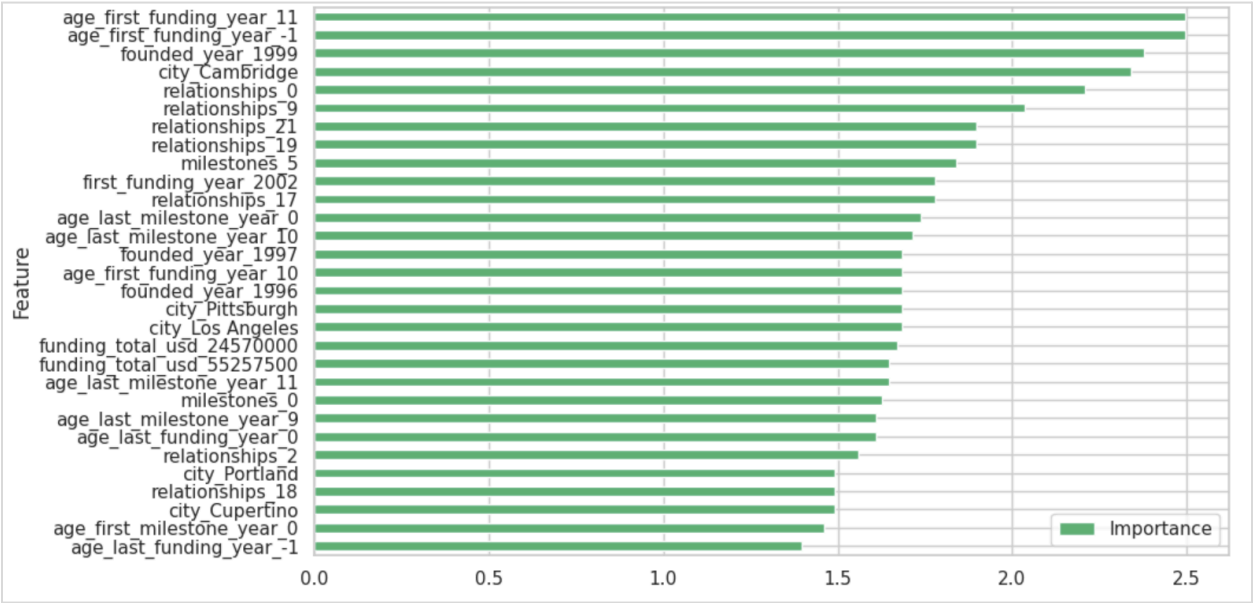

Figure 9. Feature importance by Naive Bayes classifier after data preprocessing

After data resampling, log probabilities were used to calculate feature importances. The most important features are the company's age, where the first funding year is 11, followed by the age when the first funding year is -1, and the founding year is 1999, as shown in the figure. Thus, according to predictions by the Naive Bayes classification, the time of founding of the startup as well as its age when it receives its first funding are crucial factors in determining startup success.

2. Decision Tree Model

- *Model Performance: Before Data Pre-Processing*

Before conducting any preprocessing steps, we executed a decision tree model on the raw dataset, configuring it to a depth of 4. The pivotal attribute for the first split within this unprocessed data was 'age_last_milestone_year,' showcasing an entropy value of 0.934 for its initial branch.

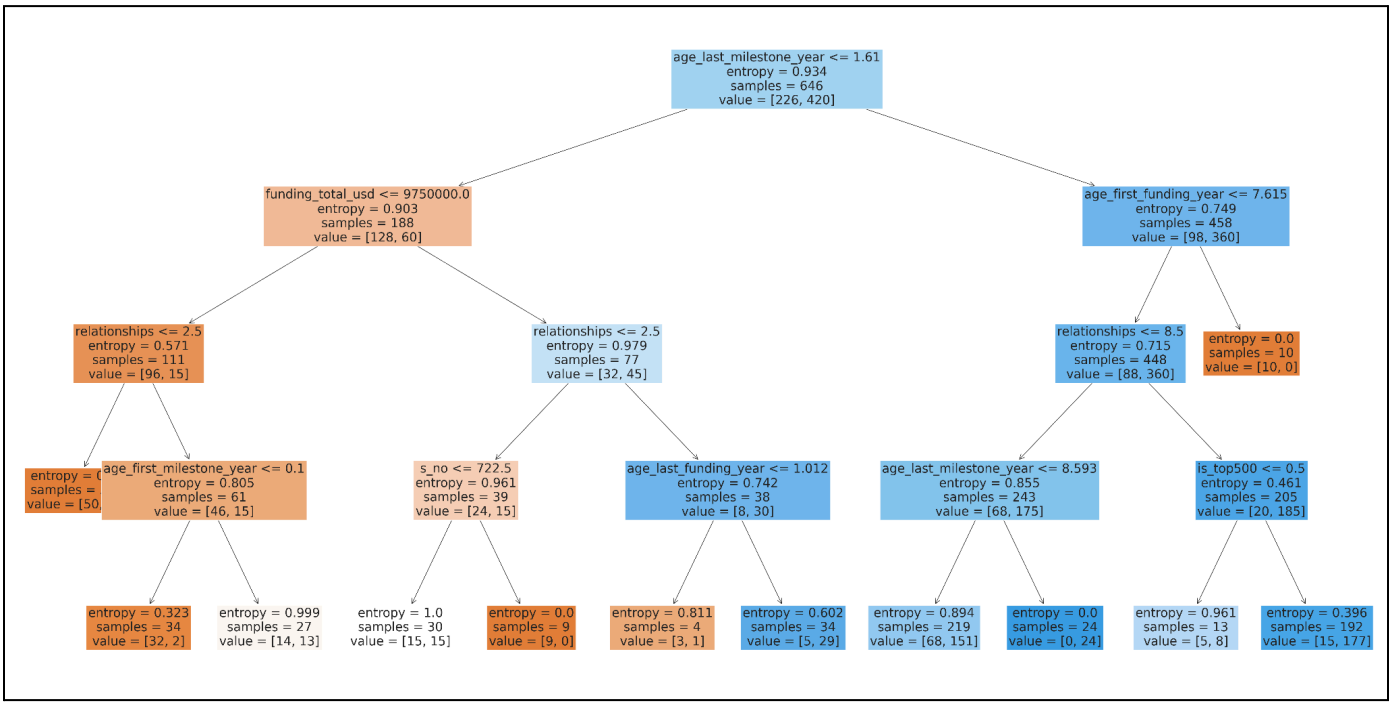

Figure 10. Decision Tree for initial Decision Tree classifier before data preprocessing

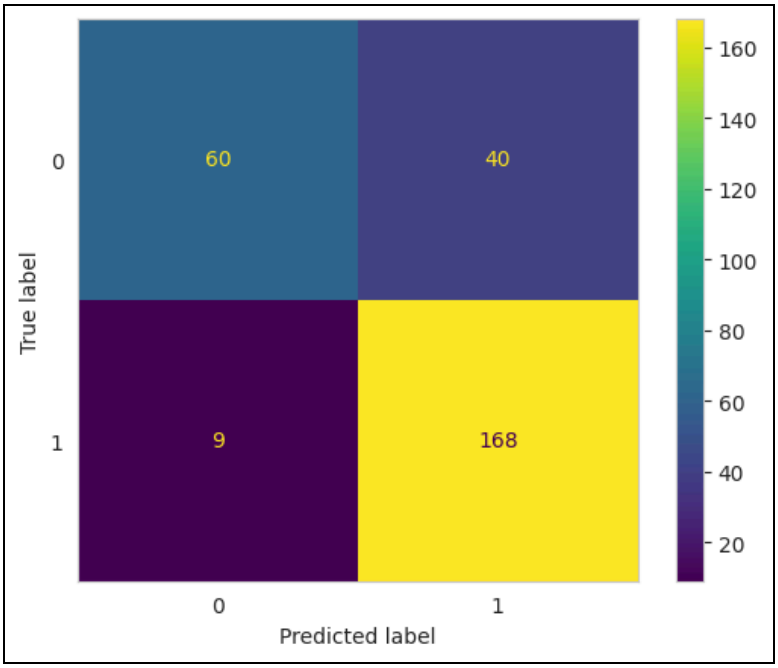

Figure 11. Confusion Matrix for initial Decision Tree classifier before data preprocessing

Accuracy	Accuracy (success)	Accuracy (failure)	F1 score
82%	95%	60%	83%

From the above figure, we can infer that while the overall accuracy is 82%. The F1 score provides a more balanced evaluation, indicating a good overall performance of the model. But from the confusion matrix we can conclude that there are high chances of data overfitting.

- *Model Performance: After Pruning*

During the process, the model involves growing the tree to its maximum depth and then removing branches that do not provide significant improvement in impurity.

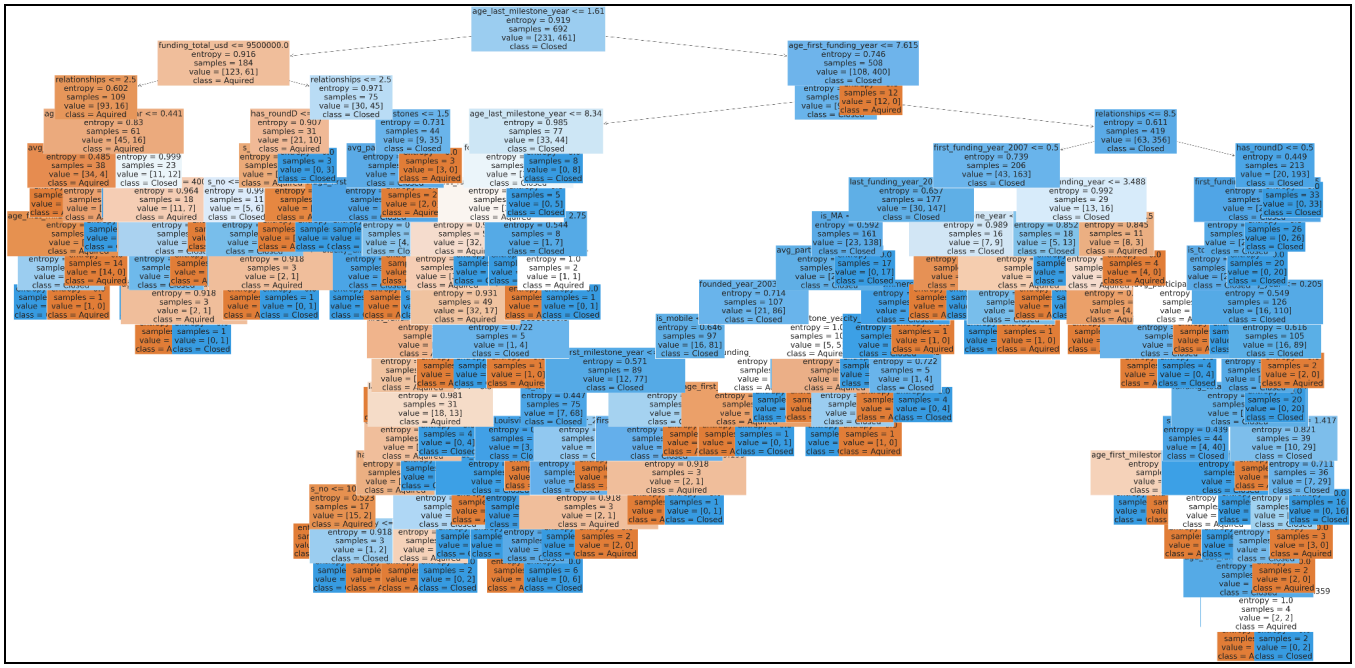

Figure 12. Decision Tree for initial Decision Tree classifier after data pruning

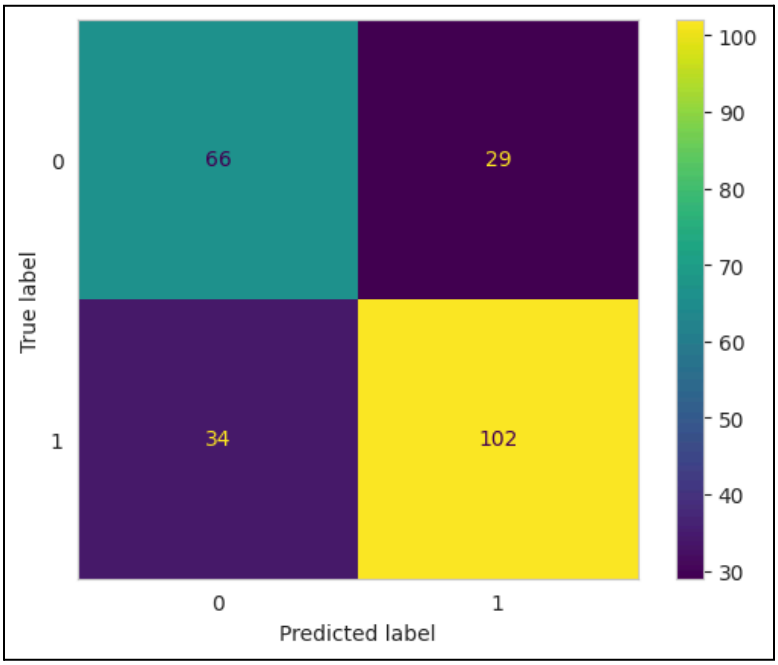

Figure 13. Confusion Matrix for initial Decision Tree classifier after data pruning

Accuracy	Accuracy (success)	Accuracy (failure)	F1 score
73%	75%	69%	73%

Given these metrics, the model demonstrates moderate performance. The overall accuracy and F1 score are at a reasonable level, suggesting that the model is making a good trade-off between precision and recall. However, the model, post-pruning, appears to provide balanced performance, but suggests overfitting

- *Model Performance: Using GridSearch*

We ran the model further using Grid searching through a predefined hyperparameter grid and evaluating the model's performance for each combination of hyperparameter values.

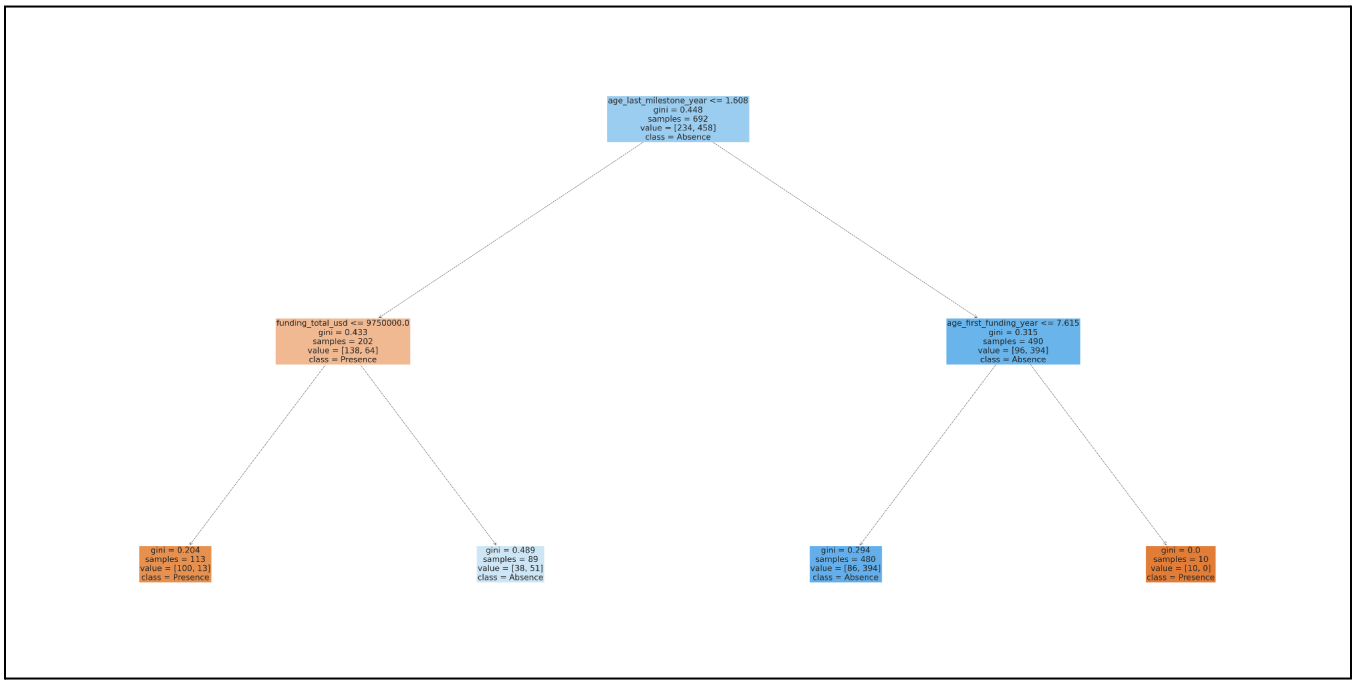

Figure 14. Decision Tree for initial Decision Tree classifier using GridSearch

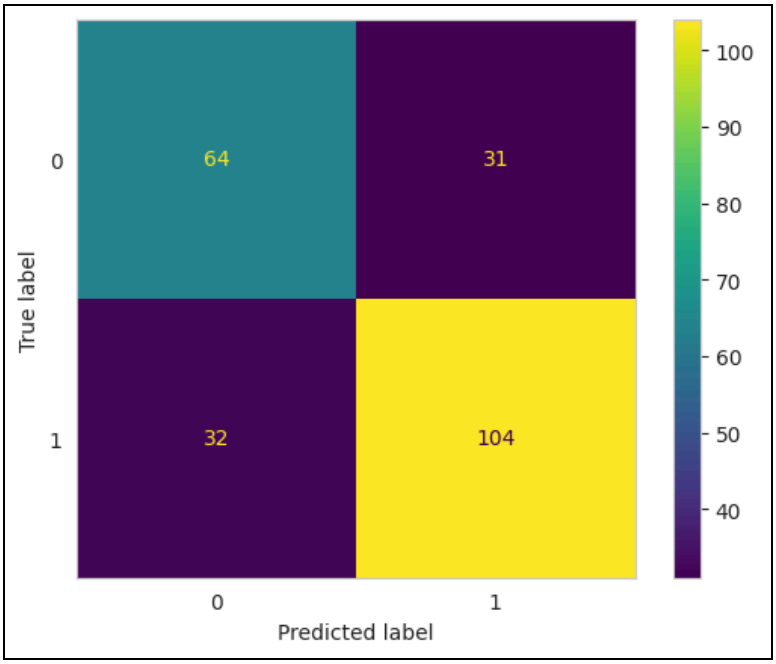

Figure 15. Confusion Matrix for initial Decision Tree classifier using GridSearch

Accuracy	Accuracy (success)	Accuracy (failure)	F1 score
72%	76%	67%	72%

3. Logistic Regression Model

The first logistic regression model was run on the dataset resulting from the initial data cleaning process. The overall dataset was split into 70% training sample and 30% testing sample. This model uses a 'liblinear' solver with a random_state of 42 and reports the following results.

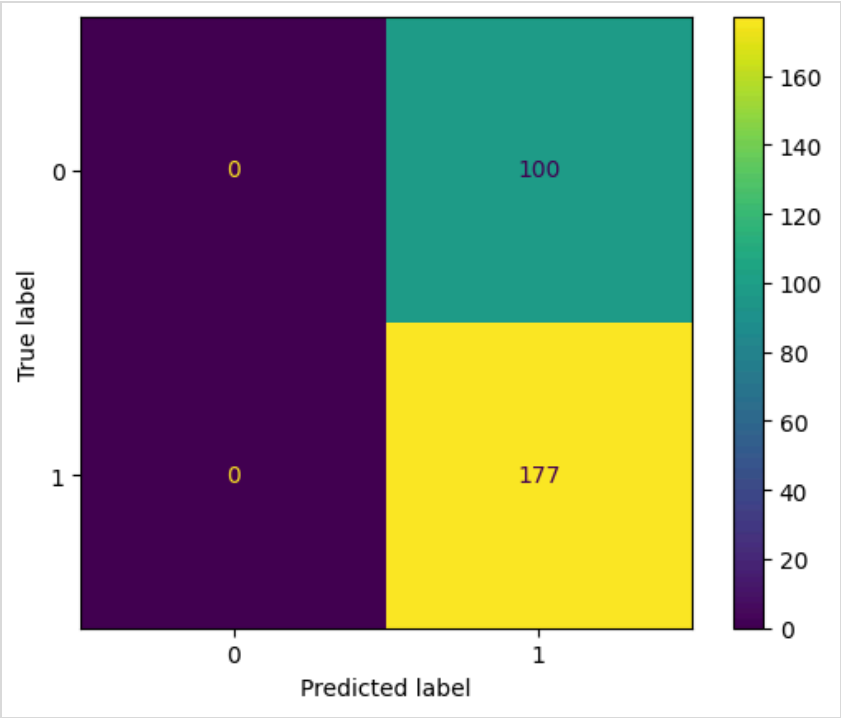

Figure 16. Confusion matrix for initial logistic regression model before data preprocessing

Accuracy	Accuracy (success)	Accuracy (failure)	F1 score
64%	100%	0%	78%

From the above figure it is evident that the logistic regression model, by itself, is not competent enough to accurately predict the success of startups. This is because it has very low measures of accuracy and F1. Furthermore, it is grossly overfitting the data as is evident from the confusion matrix as well as the measures of stratified accuracy. Thus, this model certainly needs to undergo preprocessing to improve performance.

Data Pre-Processing Measures

1. Grid Search CV (hyperparameter optimization)

Using the Grid Search cross-validation method, hyperparameter optimization was conducted to derive the most optimum logistic regression model, on the basis of accuracy. Based on results from this Grid Search CV, the following hyperparameters were selected and applied to the logistic regression model on the training data sample:

1. Solver = 'liblinear'
2. Random state = 42
3. Penalty = 'l1' or lasso regression

4. C = 1

Lastly, this optimized logistic regression model was run and measured for accuracy and F1 scores, as reported below.

Performance Measure	Before Hyperparameter Tuning	After Hyperparameter Tuning
Accuracy	64%	75.8%
Accuracy (success)	100%	85%
Accuracy (failure)	0%	59%
F1	78%	76.3%

Table 3. Examining comparative accuracy and F1 scores before and after hyperparameter optimization using Grid Search cross-validation

From the above table it is evident that hyperparameter optimization has contributed significantly to improving the overall accuracy of the logistic regression model as it grows from 0.64 to 0.76. While the F1 score slightly reduces after hyperparameter tuning from 0.78 to 0.76, these parameters are optimum for maintaining high model accuracy and reliability simultaneously.

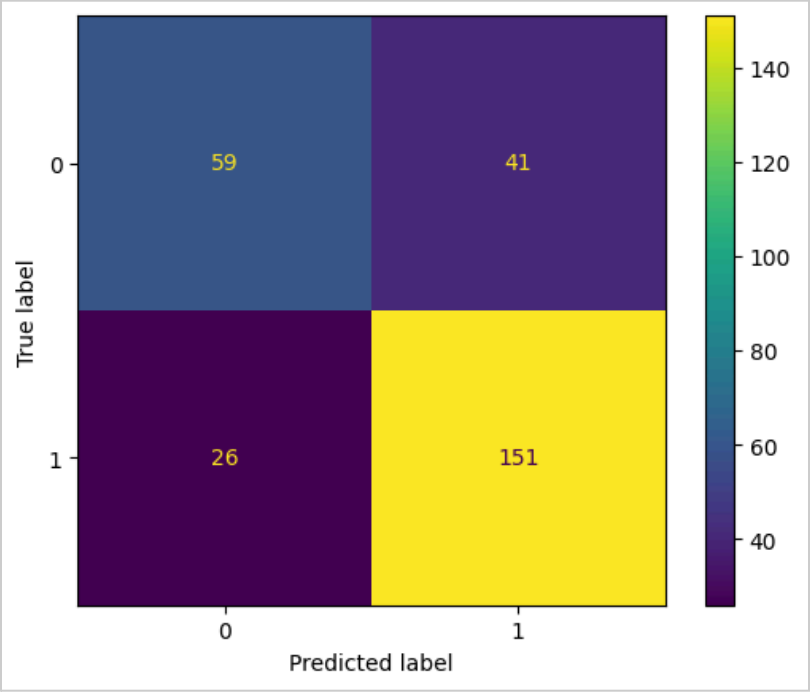

Figure 17. Confusion Matrix for Logistic Regression Model on testing data after hyperparameter tuning

Through the confusion matrix as well as the stratified accuracy measures above it is clear that hyperparameter tuning has considerably enhanced the performance of the model leading to more well-distributed predictions. Mainly, it has reduced the problems of data overfitting and improved measures of accuracy, especially stratified accuracy.

2. SMOTE

SMOTE, which stands for Synthetic Minority Over-sampling Technique, is a technique used in machine learning to address the class imbalance problem, particularly in classification tasks. Class imbalance occurs when one class (usually the minority class) has significantly fewer examples than the other class (majority class), leading to biased model training and potentially poor performance of the minority class. In the current dataset, there is a slight imbalance between the success status classes for startups wherein almost 65% of the data is classified as 'successful' while 35% of the data is classified as 'failure'. Thus, by utilizing the SMOTE resampling technique, such imbalance can be accounted for by balancing the training dataset and accuracy can further be improved. By providing more examples of the minority class, SMOTE helps the model learn the underlying patterns and features associated with that class, resulting in increased sensitivity and better classification performance. In this step of data processing, a 70-30 sampling strategy is maintained while splitting into the training and testing data. The parameters for the logistic regression model here have been derived from the hyperparameter optimization done in the previous step to further improve accuracy. Thus, the SMOTE re-sampling has yielded the following results:

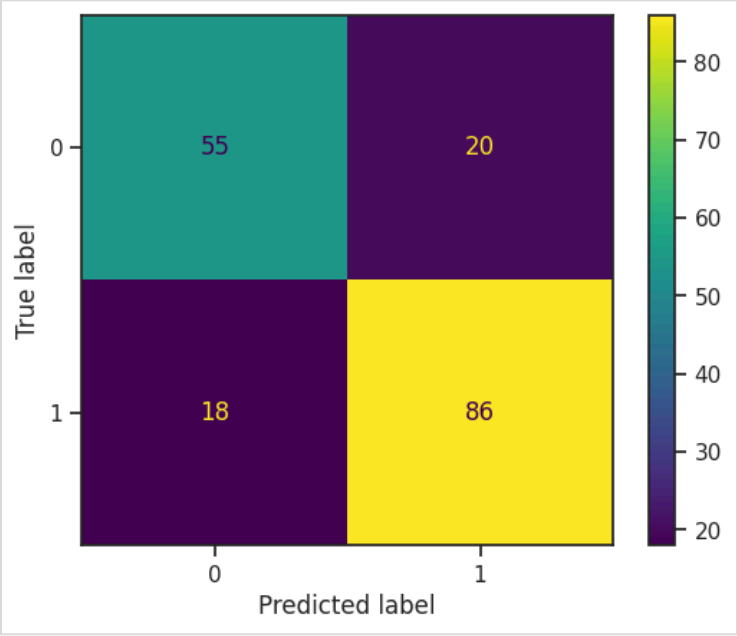

Figure 18. Confusion Matrix for Logistic Regression model on testing data after SMOTE resampling

Accuracy	Accuracy (success)	Accuracy (failure)	F1 score
78.7%	82.7%	73.3%	78.8%

The above table and confusion matrix depicts that the SMOTE resampling technique has further contributed to improving model performance by growing accuracy, reliability as well as ensuring more balanced prediction distribution, thus reducing overfitting, as compared to the previous model derived after hyperparameter tuning.

Intra-Model Performance Evaluation

Model	Accuracy	Accuracy (Success)	Accuracy (Failure)	F1 Score
Unprocessed	64%	100%	0%	78%
Grid Search CV	75.8%	85.31%	59%	76%
SMOTE	**78.7%**	**82.7%**	**73.3%**	**78.8%**

Table 4. Intra-model comparison for logistic regression models built before & after data preprocessing

From the Table above it is evident that the logistic regression model that employs the SMOTE resampling strategy provides the most optimum performance with the highest measures of accuracy, stratified accuracy and F1 scores. Furthermore, even the resulting confusion matrix for the SMOTE model is well-balanced in terms of its predictions versus true values. One of the main reasons for the optimum performance of this model is that it builds upon improvements from the previous model, using hyperparameter optimization, to increase its prediction accuracy.

Insights & Implications

Inter-Model Comparison: Investor Perspective

Model	Accuracy	Accuracy (Success)	Accuracy (Failure)	F1 Score
Naive Bayes	**78%**	**84%**	**65%**	**84%**
Decision Tree	**72%**	**76%**	**67%**	**72%**
Logistic Regression	**78.7%**	**82.7%**	**73.3%**	**78.8%**

Table 5. Inter-model comparison for predicting startup success

Lastly, this project conducted a performance evaluation between the best models identified within each of the three classifiers - Naive Bayes, Decision Tree and Logistic Regression - based on measures of overall accuracy, F1 score, and stratified accuracy. Since the main question this project aimed to uncover is 'Which machine learning model can best predict the likelihood of a startup to succeed?' it is equally important to consider values of stratified accuracy as well as overall accuracy.

Furthermore, considering the perspective of investing stakeholders, it is crucial to establish that the model can be accurate and reliable for predicting the likelihood of success as well as failure for a new startup. This is due to the extensive monetary, time and effort-based investments that are involved in funding startups through multiple rounds of investments.

Thus, the logistic regression model was chosen as the most optimal model given that it provides the best simultaneous results on measures of overall accuracy and reliability as well as stratified accuracies for success and failures for startups. Furthermore, it also satisfactorily resolves the issue of data overfitting, observed by other models. This will ultimately help investors get a clearer understanding of which startups might be a promising investment avenue and which ones could be avoided for retrieving maximum returns.

Managerial Insights: Executive Stakeholders' Perspective

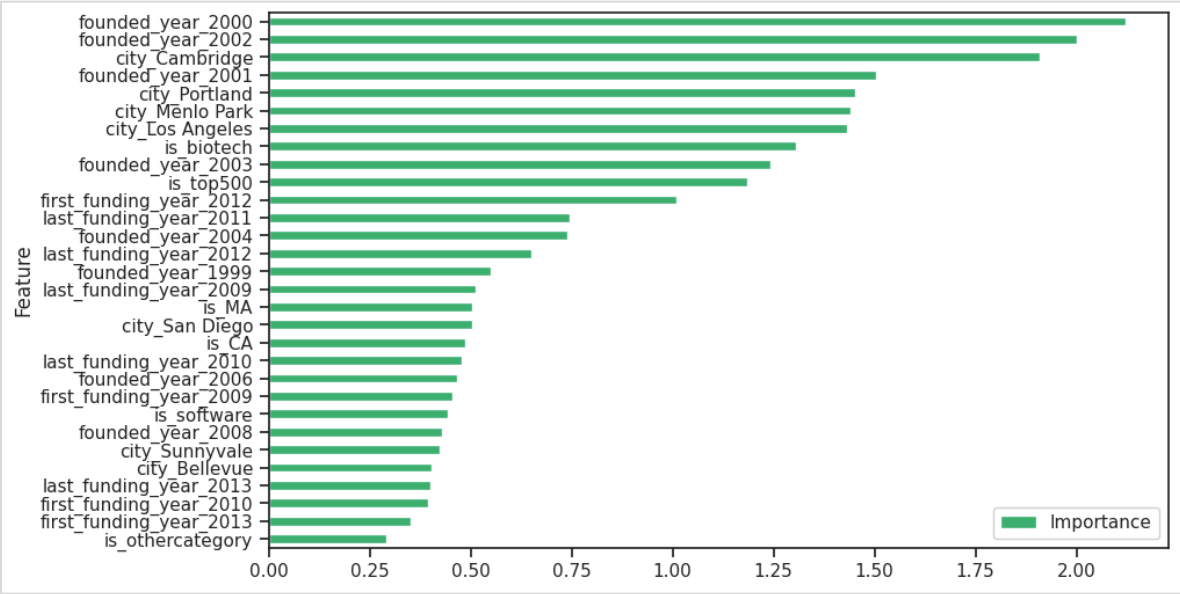

Figure 19. Ranking feature importance in predicting startup success using logistic regression

After conducting data preprocessing and model evaluation for choosing the most optimal model for predicting startup success, the SMOTE logistic regression model was used to derive the most important features that affect this success. According to the figure above, the model predicts that

if startups were founded between the years 2000 to 2003, they are likely to achieve success. Thus, executives and managers can try to understand market conditions, funding opportunities and trends and other environmental circumstances occurring in those years to replicate or bring about success in their startups today. Another factor that is seen to have high importance is the city of operations of the startup, specifically startups situated in Cambridge, Portland, Menlo Park and Los Angeles seem to have greater chance for success. Again, executives can map the markets, sizes and demographics, geographies as well as business opportunities in such cities to understand what might make them more lucrative than others - and try to capitalize on similar factors within their operational locations or think about shifting operation centers. Lastly, two interesting factors which were ranked relatively high on importance for prediction were whether the company is in the biotechnology industry and if it is featured as a top 500 company. Thus, new entrepreneurs can aim to develop solutions and services in the biotech industry and aim to grow their company into the top 500 list for increased chances of success.

Thus, the main objective of 'predicting startup success' was successfully accomplished by this project using an array of machine learning models and processes for optimizing performance. In conclusion, it was found that the logistic regression model might work most effectively in such predictions, given the nature of the data used. It can also be fruitful in delineating which factors are most important while making such predictions. The report concludes with multi-perspective recommendations and insights for various stakeholders involved in the startup industry.

Project Insights

Overall, this project was an extremely beneficial channel for implementing and refining our newly learned skills in machine learning and modeling. A few key learnings that we had from this project were:

- Importance of exploratory data analysis in identifying key variables and initial patterns present in the dataset - its benefits in deciding what should be further analyzed and how.
- Differentiating between various pre-processing techniques and truly understanding their use case in real world settings.
    - One mistake that we previously made during our model building process was to use the K-fold cross-validation strategy as a way to process data for improving model accuracy without realizing that it's actually a technique to split data into train-test samples and is primarily used to compare model evaluations rather than improving performance.
    - This allowed us to instead use the grid search cross-validation method for hyperparameter optimization, helping us create stark distinctions between the two methods and improve our understanding of the subject.
- Highest accuracy doesn't always equal best performance - this was one of the biggest learnings & helped us focus on examining model performance from various perspectives.