Ss. Cyril and Methodius University in Skopje (UKIM)

FACULTY OF INFORMATION SCIENCES AND COMPUTER ENGINEERING

Project Research

An Algorithmic Approach for Evaluating Startup Success Based on Multiple Metrics

Authors:

Matea Blazeska,

Mihaela Blazeska

Mentor:

Milena Trajanoska

Skopje,2025

TABLE OF CONTENTS

Abstract

The beginning of every business is a process that is filled with uncertainty, leaving the majority of startups prone to failure. Therefore, accurate early prediction of success is very valuable for both investors and founders.This study proposes an algorithm for predicting startup success based on multiple financial, operational, and market metrics. The proposed model is built using various machine learning techniques, including logistic regression, random forest, and gradient boosting classifiers, to achieve results that closely approximate future outcomes. Furthermore, it was trained using a dataset that contains information about over 100,000 startups. The dataset included comprehensive data preprocessing such as handling missing values, normalizing numerical features, ensuring data consistency, merging heterogeneous sources, and encoding categorical variables.The model achieved an ROC-AUC score of 0.84 and demonstrated robustness in cross-validation. These findings suggest that multi-metric approaches can significantly improve the accuracy of startup success evaluations.

Keywords Startup evaluation, Machine learning, Predictive modeling, Multi metric analysis, Startup success prediction, ROC-AUC, Data preprocessing, Cross-validation

1. Introduction

Over the past two decades, the global startup ecosystem has grown significantly, with thousands of new businesses opening up every year all over the world. This quick growth encourages economic growth and innovation, but it also comes with a high failure rate. More than 70% of startups fail within the first few years of operation, according to numerous studies. A number of various factors, such as inadequate market demand, poor financial management, operational inefficiencies, and competitive pressures, can lead to such results. In this setting, it is crucial for founders, accelerators, and investors to be able to accurately assess a startup's chances of success early on.

Even though traditional methods of making predictions about startup success give great insight, they are limited. That limitation is a result of human nature, which does not have the ability to preprocess big datasets that contain over millions of rows. Moreover, they are often biased and cannot find the accurate and true metrics that are most responsible for the outcome of the startup success rate. These limitations reduce the effectiveness of traditional prediction approaches in complex, data-rich environments. Therefore, there is a growing need for automated methods that can analyze large datasets objectively and identify key success factors more precisely.

This research aims to develop an algorithm that can accurately predict startup success rates at the early stages of their development. To achieve this, various machine learning techniques are used, and the model is trained on a large dataset containing multiple financial, operational, and market metrics. By using data-driven methods, this study hopes to offer predictions that are not only more accurate but also practical and will help entrepreneurs and investors make smarter, more confident choices.

This research brings several contributes:

1. We put together a large and detailed dataset of over 100,000 startups. The data was preprocessed and carefully cleaned, the missing values were filled, and the data was made consistent. Additionally, all the categorical features were encoded, so it’s ready to use for analysis.

2. We tried out different machine learning models, such as logistic regression, random forest, and gradient boosting, to see which one does the best job at predicting which startups will succeed early on.

3. We tested these models thoroughly, using reliable methods to make sure the results are solid. Our top model scored really well, showing it can predict success with good accuracy.

4. Finally, we identified some key factors that influence whether a startup makes it or not, offering practical insights for entrepreneurs, investors, and others who want to support new businesses.

2. Related Work

Research on predicting startup success spans classic statistical models, machine-learning classifiers, and more recent text- and network-aware approaches. Early work often framed "success" as acquisition/IPO or survival and modeled outcomes with logistic regression or survival analysis, using funding history, sector, geography, and team features as predictors. For example, studies leveraging Crunchbase-style data show that investment stage, amount, and round cadence are strong correlates of eventual exits, while survival-analysis (e.g., Cox models) has been used to estimate hazard rates over a startup's life cycle [UPCommons, arXiv].

With the wider availability of venture datasets, tree-based ensembles (Random Forests, Gradient Boosting, XGBoost) became standard baselines, typically outperforming linear models on heterogeneous tabular features. Comparative studies report that ensembles benefit from non-linear interactions among funding, industry, and timing features, though they remain sensitive to class imbalance (few failures vs. many operating firms) and sparse categorical spaces (thousands of industries/regions) [arXiv].

A parallel line of work augments tabular data with textual signals. Using topic models and embeddings on startup descriptions, researchers show that market narratives and product themes carry predictive signals for funding and growth; incorporating text improves discrimination beyond funding-only features. Large-scale analyses of Crunchbase descriptions (2009–2019)

demonstrate stable topic structures across industries and regions, enabling better grouping of long-tail categories [PMC].

Network-based features investor–startup bipartite graphs, co-investment networks, and founder networks also improve prediction and ranking of promising ventures. Recent studies find that the centrality and quality of investors, as well as co-investment motifs, are associated with higher odds of follow-on funding and exits. Commercial platforms (e.g., Crunchbase) now report AI models that forecast fundraising, acquisition, and IPO likelihoods using behavior and graph signals, underscoring the value of network/context features [Kaggle, The Wall Street Journal].

Typical modeling choices: Supervised classifiers (logistic regression, SVMs, Random Forest, Gradient Boosting/XGBoost) dominate; survival models are used when the time-to-event is explicit (time to exit/closure). Evaluation commonly relies on ROC-AUC, precision/recall, and time-aware metrics when censoring is present. Feature groups include funding totals/rounds, stage progression, geography, industry tags, team size/founders, investor identity/quality, and sometimes web/usage signals [arXiv, UPCommons].

Key gaps and challenges: (i) Severe class imbalance (few labeled failures) often yields high overall accuracy but poor minority-class precision/recall; studies recommend resampling (e.g., SMOTE) and cost-sensitive learning [arXiv]. (ii) High-cardinality categoricals (industries, regions) create sparsity; text embeddings or clustering of categories can help [PMC]. (iii) Temporal dynamics are frequently under-modeled—many works treat snapshots rather than sequences of rounds; survival/time-series models can address censoring and tempo of funding [UPCommons]. (iv) Generalization across regions (e.g., India/Bangalore vs. US) is rarely tested; domain shift remains a risk [arXiv]. (v) Data provenance and noise—crowd-sourced attributes and missingness can bias models if not handled with robust preprocessing and validation [PMC].

Positioning of our work. We build on this literature by (1) integrating multiple Crunchbase-derived Kaggle datasets to broaden temporal and geographic coverage; (2) addressing imbalance with SMOTE and class-weighted learning; (3) reducing high-cardinality categories via sentence-embedding clustering of industries; and (4) providing cross-validated performance with careful

feature engineering and thresholding aiming for balanced performance on both success and failure classes.

3. Methodology

3.1.Data Source

In this study, we worked with several datasets gathered from Kaggle, many of which are based on Crunchbase data. All of them focus on startups and small businesses, but each highlights different aspects like funding rounds, industries, growth patterns, or even success and failure rates. By bringing these datasets together, we're able to look at the startup ecosystem from multiple angles. This combination gives us a richer picture and helps make our model's predictions stronger and more reliable. Specifically, we used three main datasets: one covering the last 5 years, another focusing on the past 2 years, and a larger dataset spanning from 2013 to 2022.

1. **Dataset_5y**:

   This dataset is largely derived from Crunchbase and focuses on startups over a five-year period. It contains more than **54,000 records** with detailed information such as company name, industry categories, market type, total funding received, operating status (operating, acquired, or closed), and location details (country, state, region). It also tracks investment rounds (A–H), which makes it valuable for analyzing funding patterns and survival rates.

2. **Dataset_2y**:

   This dataset provides more recent startup activity, with around **66,000 entries**. It highlights details such as company name, funding history, number of rounds, founding date, first funding date, and latest funding date. It is especially useful for understanding **short-term trends**, like how quickly startups move from foundation to receiving funding, and which sectors attract faster investments.

3. **Dataset_2013-2022**:

   This dataset is broader and covers nearly a decade of startup activity, with about **50,000 records**. It includes detailed financial and operational features such as total funding amount

(both reported and USD-standardized), industries, headquarter locations, operating status (active, acquired, or closed), investment stages, number of investors, acquisitions, number of employees, and even top investors. This dataset is crucial for understanding **long-term patterns**, such as which industries grow steadily and what investment strategies tend to succeed.

Dataset Name	Size (Rows × Columns)	Key Features	Short Description
Dataset_5y	54,294 × 39	Permalink, Name, Homepage URL, Category List, Market, Funding Total (USD), Status, Country/Region, Multiple Funding Rounds (A–H), etc.	Contains detailed company-level data of startups over 5 years, focusing on industry, geography, and funding history.
Dataset_2y	66,368 × 14	Permalink, Name, Homepage URL, Category List, Funding Total (USD), Status, Location, Funding Rounds, Founded Date, First/Last Funding Date	Covers startup activity over 2 years, with an emphasis on funding timelines and operational status.
Dataset_2013-2022	50,555 × 23	Organization Name, Total Funding Amount, Industries, Headquarters Location, Operating Status, Investment Stage, Number of Employees, Founders etc.	Provides a broader view of startups from 2013–2022, highlighting funding size, industry trends, company stage, and investor networks.

Table 3.1: Performance metrics across all folds

To better understand the scale of the data we are working with, we compared the number of records across the three datasets. The following chart shows how many startups are included in Dataset_5y, Dataset_2y, and Dataset_2013-2022, providing an overview of the dataset sizes and coverage periods.

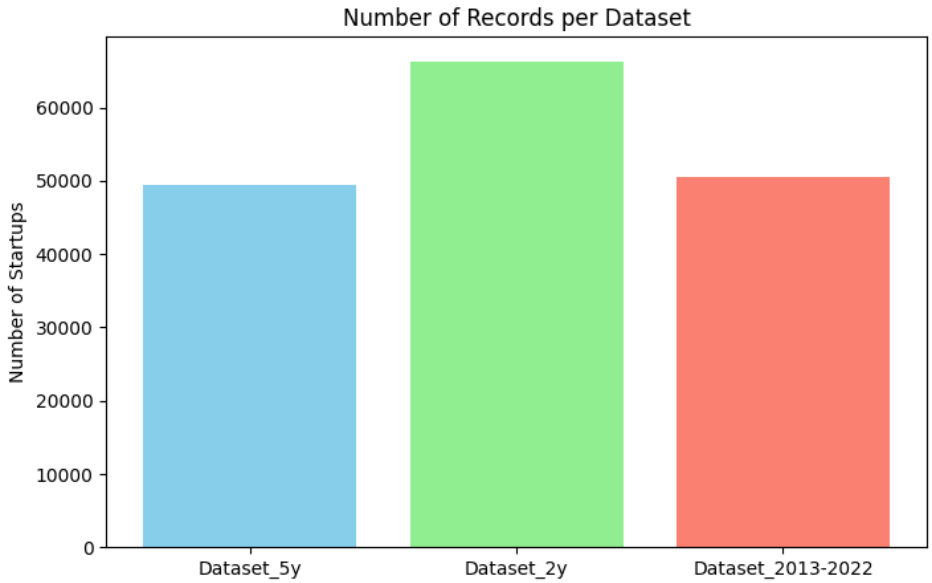

Figure 3.1.1: Chart of number of records per Dataset

We also analyzed the distribution of startup statuses in Dataset_5y to understand how many startups are currently operating, have been acquired, or have closed. This gives insight into the overall health and lifecycle stages of startups in the last five years. Similarly, we performed the same analysis for Dataset_2y and Dataset_2013-2022 to capture recent trends and long-term patterns, respectively, providing a comprehensive view of startup survival and success across different time periods.

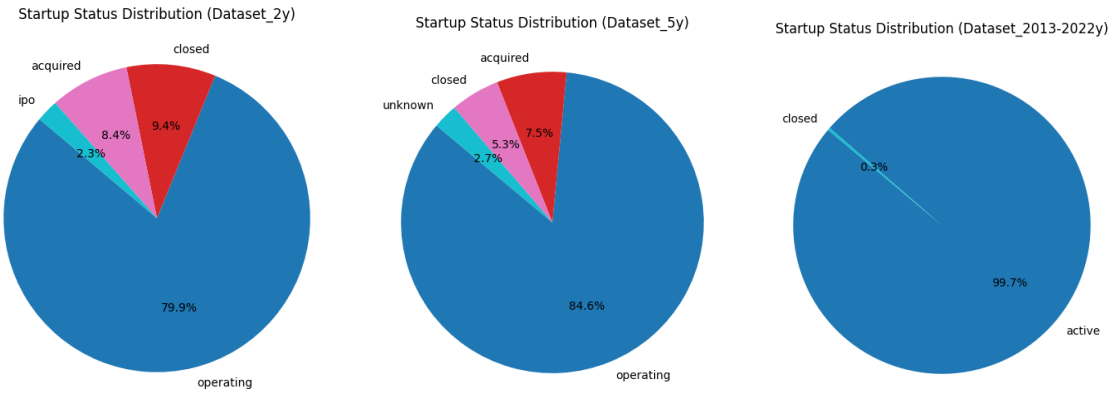

Figure 3.1.2: Pie charts of startup status distribution per Datasets

Altogether, these datasets give us a clear and detailed picture of the startup world. They help us see patterns over time, understand what makes some startups succeed while others fail, and give our model a solid foundation to make more accurate predictions.

3.2. Data Standardization

1. **Dataset_5y**:

   To begin, the data was first made consistent to ensure reliability and prepare it for feature engineering. Several preprocessing steps were applied. First, textual fields such as company names and market categories were normalized by converting all characters to lowercase and removing leading or trailing whitespace. Similarly, the status and region fields were standardized in lowercase, while country_code entries were converted to uppercase. These steps reduced inconsistencies caused by variations in capitalization and formatting.

   Next, the funding_total_usd field was cleaned to facilitate quantitative analysis. Commas were removed, extra whitespace was stripped, and placeholder values such as dashes, empty strings, and "N/A" were replaced with missing value indicators. The resulting values were then converted to numeric format, allowing the data to be interpreted correctly by machine learning models.

2. **Dataset_2y**:

   To ensure consistency and prepare the 2-year dataset for feature engineering, several preprocessing steps were applied. Textual fields, including company names, category lists, status, country codes, state codes, and city names, were standardized by removing extra whitespace and converting characters to lowercase or uppercase as appropriate. This normalization reduced inconsistencies arising from varying capitalization or formatting.

   The funding_total_usd column was cleaned by removing commas and converting all entries to numeric format, with invalid or non-numeric values. Date fields, including founded_at, first_funding_at, and last_funding_at, were converted into a consistent YYYY-MM-DD format to ensure temporal uniformity. Finally, columns which were

unnecessary for analysis, such as region and state_code, were removed to simplify the dataset.

3. **Dataset_2013-2022**:

The 2013–2022 dataset was first cleaned and standardized to ensure data consistency, accuracy, and reliability before any analysis. Textual fields, including organization names, industries, regions, operating status, funding status, last funding type, top investors, and acquisition status, were standardized by removing leading and trailing whitespace and converting text to lowercase. This reduces inconsistencies caused by variations in capitalization or spacing, ensuring that identical entries are treated uniformly.

Country names were standardized using a predefined mapping to ensure consistency. Each country name in the Headquarter Country field was converted to its corresponding ISO country code (e.g., Argentina → ARG, Australia → AUS). This approach ensures that different spellings, abbreviations, or variations all refer to the same country, allowing accurate grouping and geographic analysis.

Numeric fields, including total funding amount, number of founders, and number of employees, were cleaned and converted to numeric types. Commas were removed from funding amounts, and invalid entries were coerced to missing values. Missing counts of founders or employees were replaced with zero. Date fields, including the founded date and last funding date, were converted into a uniform YYYY-MM-DD format to allow accurate temporal calculations.

3.3. Handling Missing Values

1. **Dataset_5y**:

After standardizing the dataset, missing values were addressed to ensure data completeness and reliability. For categorical fields such as company name, market, status, country code, region, and category list, missing entries were replaced with the placeholder value "unknown". This approach preserves the presence of missing information while allowing the dataset to remain fully analyzable. The category_list field was additionally

stripped of whitespace and converted to lowercase to maintain consistency after filling missing values.

For the numeric field funding_total_usd, missing values were filled in two steps. First, missing values were replaced with the median funding amount within the same market, ensuring that replacements reflected the typical funding for similar companies. Any remaining missing values were subsequently filled using the overall median funding across all startups. The median was chosen because it is robust to extreme outliers, which are common in funding data.

Finally, rows with missing critical dates (first_funding_at and last_funding_at) were removed, as these fields are essential for calculating features such as company age and time since last funding.

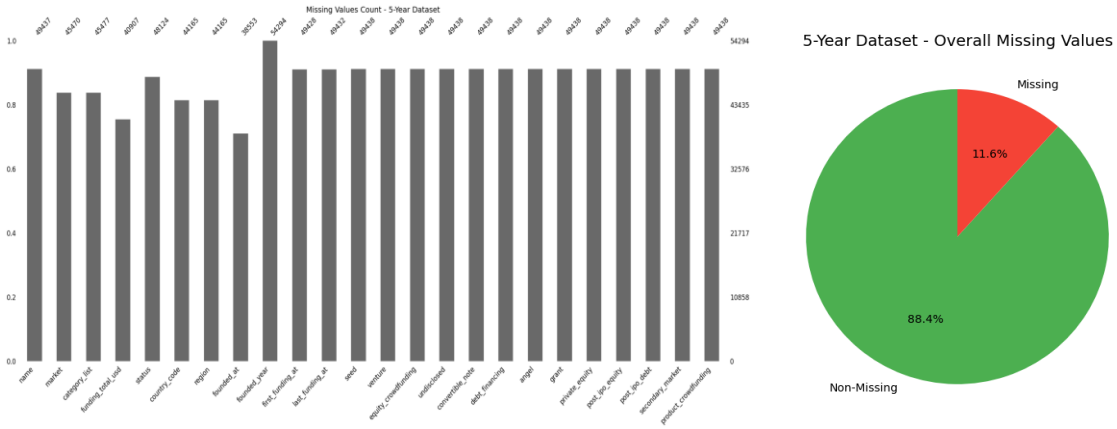

Figure 3.3.1: Msno-matrix of missing values and pie chart(Dataset_5y)

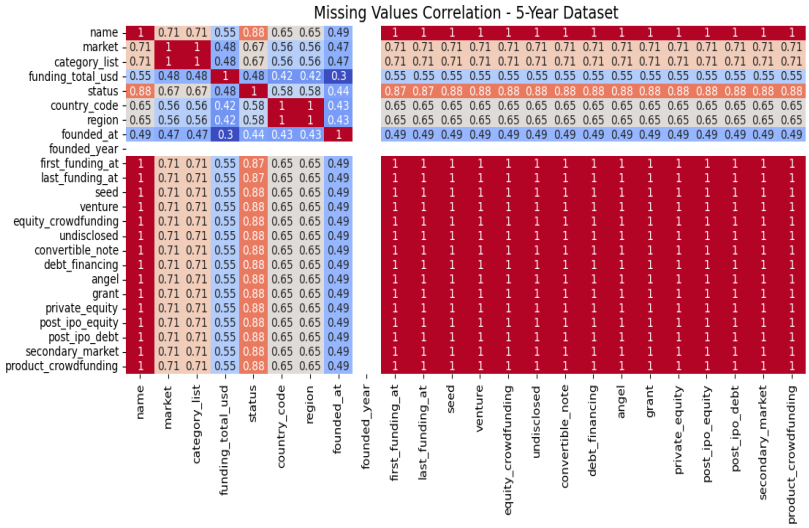

Figure 3.3.2: Missing values correlation(Dataset_5y)

2. **Dataset_2y**:

After standardizing the dataset, missing values were cleaned to ensure completeness. Categorical fields, including company name, category list, country code, and city, were filled with the placeholder "unknown", while rows missing the critical first_funding_at date were removed. These steps rendered the dataset consistent and ready for feature engineering and modeling.

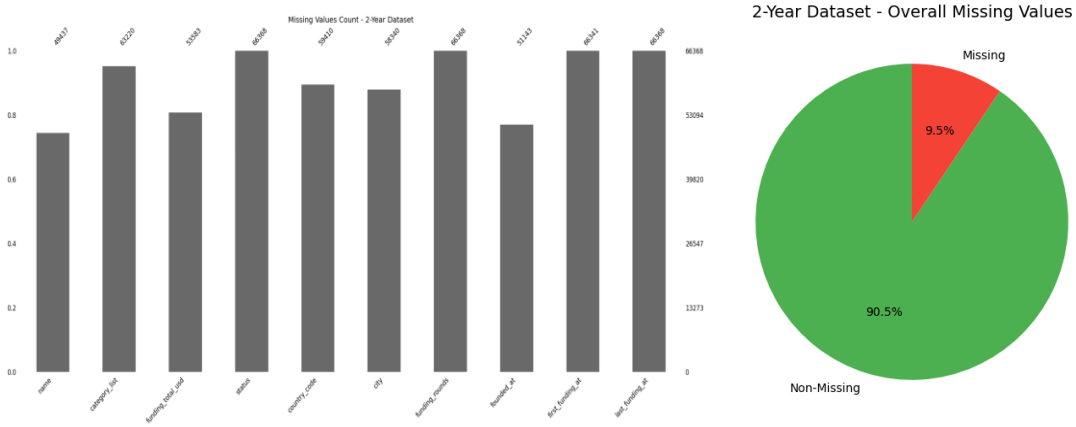

Figure 3.3.3: Msno-matrix of missing values and pie chart(Dataset_2y)

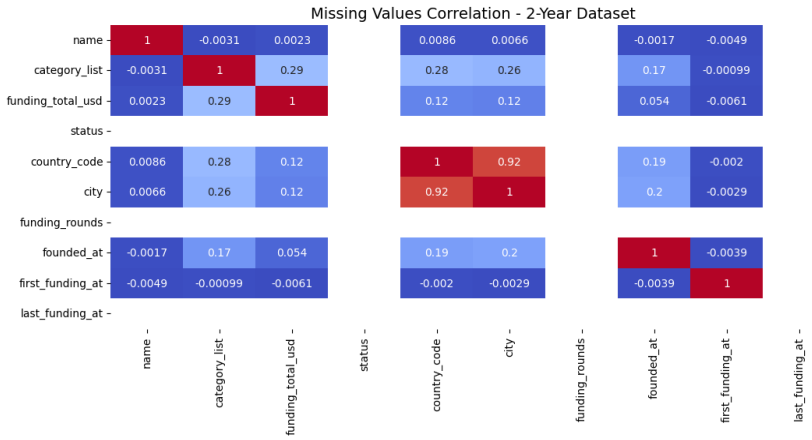

Figure 3.3.4: Missing values correlation(Dataset_2y)

3. **Dataset_2013-2022**:

For the 2013–2022 dataset, missing values in the Headquarter Country and Headquarter Region fields were filled with the placeholder "unknown". This method preserves entries with missing geographic information while maintaining a complete dataset suitable for analysis.

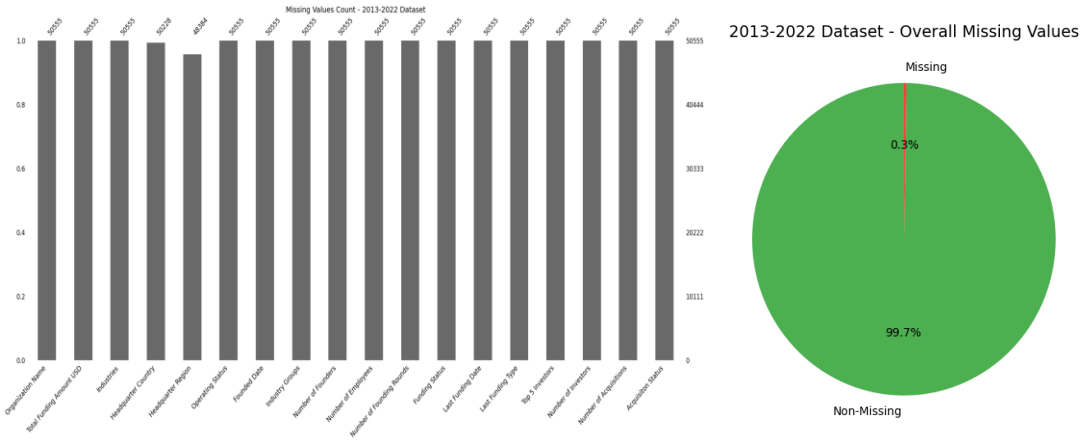

Figure 3.3.5: Msno-matrix of missing values and pie chart(Dataset_2013-2022)

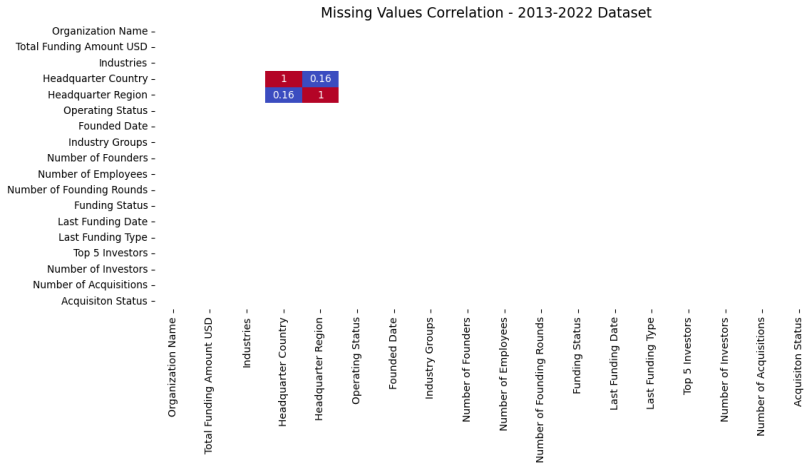

Figure 3.3.6: Missing values correlation(Dataset_5y)

3.4. Merging The Datasets

After standardizing and cleaning the three datasets (5-year, 2-year, and 2013–2022), they were merged into a single dataset to provide a unified source for analysis. Since the datasets came from different sources with slightly different formats, several steps were necessary to align them before merging.

First, column names were standardized to ensure consistency across datasets. For example, “Organization Name” was renamed to name, “Total Funding Amount USD” was renamed to funding_total_usd, and “Headquarter Country” was mapped to country_code. Similarly, fields like Industries were renamed to category_list, and city was renamed to region. This ensured that all datasets used the same terminology for equivalent attributes, allowing them to be combined seamlessly.

Certain unnecessary or duplicate columns, such as Industry Groups and detailed investor information, were dropped to simplify the dataset and focus on variables directly relevant to our analysis. Cases where invalid or missing values appeared as “NaT” were replaced with the placeholder “unknown” to avoid inconsistencies.

To merge the datasets, a custom merging function was developed. This function combined the datasets on the name column (company name) using an outer join. This ensured that all companies present in any dataset were included, even if they did not appear in others. Where duplicate

columns were created during merging, values were combined intelligently choosing the available non-missing entry between the datasets so that no valid information was lost.

The Funding Status column required additional attention. In some datasets, funding stage information was explicitly provided, while in others it was represented by separate binary columns (e.g., seed, angel, venture). To unify this, a function was applied: if an explicit Funding Status existed, it was retained; otherwise, the function checked across the binary funding columns and assigned the correct status. After this process, redundant funding columns were dropped, leaving a single clean Funding Status field. Missing values in this field were filled with the placeholder "unknown".

Additional cleaning steps were applied to further improve quality. Text-based columns such as category_list, region, and Funding Status were checked for placeholder symbols (e.g., "---") and replaced with "unknown" to maintain consistency. The name column was normalized by stripping whitespace, converting all characters to lowercase, and removing symbols like "#" to ensure uniform company identifiers.

For temporal variables, incorrect or placeholder dates such as "1970-01-01" were identified and set to missing values. The founded_at column was converted into a proper datetime format and then stored as a string, with missing entries replaced by "unknown".

Finally, a success_label variable was created to capture company outcomes. Companies with statuses such as active, operating, ipo, or acquired were labeled as successful (1), while those with closed status were labeled unsuccessful (0). This binary feature was introduced to support later predictive modeling and analysis.

Through these merging and cleaning steps, the three different datasets were consolidated into a single, consistent, and reliable dataset. This ensured that no critical information was lost, redundancy was removed, and the final dataset was structured in a way that supports robust analysis of startup success.

3.5.Building the model

After merging the datasets, the final dataset contains 116,448 startup entries with a mix of numerical, categorical, and temporal features prepared for machine learning. The name column

was removed, as it is irrelevant for predicting startup success. The dataset includes key features such as funding_total_usd, funding_rounds, founded_at, last_funding_at, and the target variable success_label, which indicates whether a startup is successful (operating, acquired, or IPO) or closed.

The categorical information was carefully processed. The original category_list column, which could contain multiple categories per startup, was cleaned by filling missing values with 'unknown', converting all text to lowercase, and standardizing delimiters. Semantic similarities between categories were captured using sentence embeddings from the SentenceTransformer model and subsequently clustered with KMeans. Each startup was then assigned a main_category_cluster representing its primary category group. This approach reduces dimensionality and captures semantic meaning while allowing the model to learn patterns from categories.

Other categorical variables were also encoded for model readiness. The country_code column was label-encoded into country_encoded, while the Funding Status column was one-hot encoded to produce separate binary columns for each funding type. This encoding ensures that the model can leverage information about funding mechanisms without assuming any ordinal relationship.

Temporal and missing data were addressed as well. A binary feature, founded_at_missing, indicates whether a founding date is missing. Additionally, years_since_last_funding measures the time in years since a startup's most recent funding round, providing insight into recent funding activity.

The final dataset thus includes numerical, categorical, and temporal features in a model-friendly format. One can observe a combination of original numerical features, encoded categorical variables, and derived temporal indicators, all consolidated to support machine learning models predicting startup success. This preparation ensures that the model can learn from both explicit quantitative information and semantic patterns within categorical data.

During the exploratory analysis of the success_label target variable, we observed a significant class imbalance. Out of 116,448 startups in the final dataset, 110,044 entries (approximately

94.5%) are labeled as successful (success_label = 1), while only 6,404 entries (around 5.5%) are labeled as unsuccessful (success_label = 0).

This imbalance indicates that the majority of startups in the dataset are successful, which can create challenges during model training. Standard machine learning algorithms may become biased toward predicting the majority class, potentially leading to poor performance in identifying unsuccessful startups.

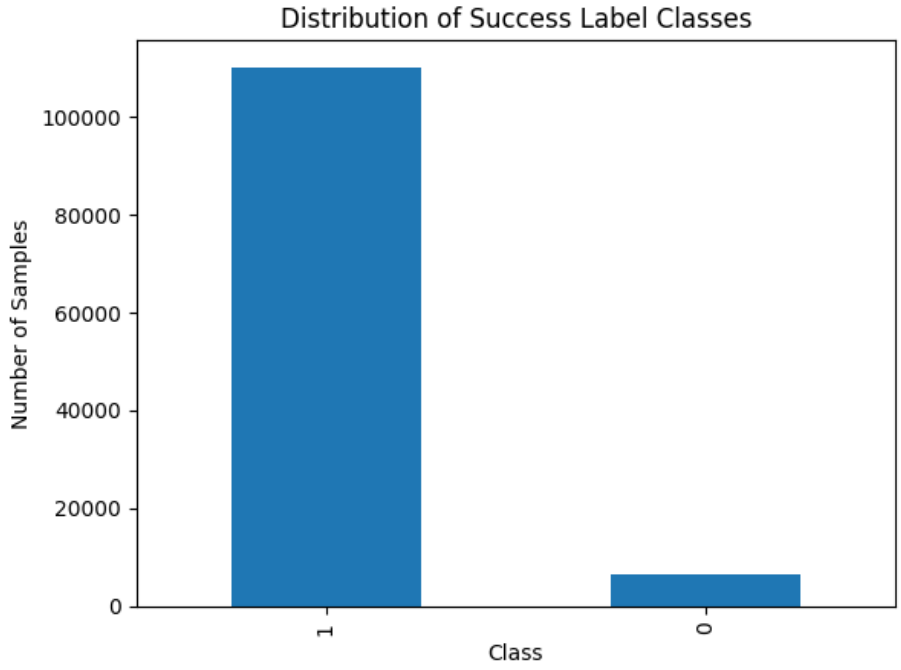

Figure 3.5.1: Distribution of success label

3.6.Model selection

We tried multiple machine learning approaches to predict startup success. Initially, we split the data into training and testing sets using stratified sampling to preserve the class proportions. We also scaled numeric features (funding_total_usd, funding_rounds, years_since_last_funding) and reduced the number of categories in main_category_cluster and country_encoded by grouping less frequent categories into an “other” category.

To address the imbalance, we applied SMOTE (Synthetic Minority Oversampling Technique) on the training set to generate synthetic samples of the minority class. We first trained an XGBoost

classifier, tuning the probability threshold to improve recall for the minority class. Although we achieved high accuracy, the precision for unsuccessful startups remained very low , indicating that many predicted failures were false positives. Randomized hyperparameter search for XGBoost slightly improved performance but did not resolve the imbalance issue.

Next, we experimented with a Random Forest classifier using class weighting and cross-validation. Despite balancing the training set and using stratified k-fold validation, the model consistently predicted the majority class too strongly, yielding high overall accuracy but almost no meaningful predictions for the minority class. The best threshold adjustment (via precision-recall optimization) improved recall slightly, but precision remained extremely low.

Finally, we implemented a Random Forest classifier combined with SMOTE resampling and 5-fold stratified cross-validation. We first prepared the data by removing non-predictive or redundant columns, reducing category cardinality for main_category_cluster and country_encoded, and standardizing numeric features. To address the class imbalance, SMOTE was applied to generate synthetic examples of the minority class. The class-weighted Random Forest was then trained on the resampled data and evaluated using stratified 5-fold cross-validation. The model demonstrated excellent cross-validated performance, achieving a mean ROC AUC of 0.961. For the minority class (unsuccessful startups), precision, recall, and F1-score were 0.87, 0.93, and 0.90, respectively, while for the majority class (successful startups), these metrics were 0.92, 0.87, and 0.89. Overall accuracy reached 0.90. These results indicate that this combined approach effectively balances predictions across both classes, overcoming the limitations observed in prior experiments with unbalanced data or alternative models, and provides reliable predictions for both successful and unsuccessful startups in the dataset.

4.Experimental Setup

To evaluate the predictive performance of our models in a reliable and reproducible way, we designed the experimental setup with a strong emphasis on the use of well-established **Python libraries and tools**. These libraries provided the functionality to preprocess the data, handle class imbalance, train models, and evaluate them systematically.

4.1.Train/Test Split with Stratification

The dataset was divided into training and testing subsets using an **80/20 split**. Stratification was applied on the target variable (success_label) to ensure that both training and testing sets preserved the original class distribution. This is especially critical because startup success prediction is typically an **imbalanced classification problem**, where the number of successful startups is much lower compared to unsuccessful ones. Stratification helps avoid bias toward the majority class and ensures fair evaluation.

- **Why scikit-learn?** It provides reliable, well-tested implementations for splitting data with reproducibility (random_state) and built-in support for stratification, making it ideal for imbalanced classification tasks such as startup success prediction.

4.2.Feature Preprocessing

Several preprocessing steps were performed to improve model quality and consistency:

- **Category Reduction:** For categorical variables such as *main_category_cluster* and *country_encoded*, only the top 15 most frequent values were retained. All other categories were grouped under a single label (-1). This reduced the dimensionality of sparse categories and prevented the model from overfitting to rare values.

- **Numeric Scaling:** Continuous features, such as *funding_total_usd*, *funding_rounds*, and *years_since_last_funding*, were standardized using **z-score normalization** (StandardScaler). Scaling ensures that all numeric features are on the same scale, which improves model stability and convergence.

4.3. Handling Class Imbalance

Since successful startups form only a minority of the dataset, class imbalance was addressed using **SMOTE (Synthetic Minority Over-sampling Technique)**. SMOTE generates synthetic examples of the minority class to balance the dataset, improving the model's ability to learn decision boundaries for both classes.

Additionally, the **RandomForestClassifier** was configured with class_weight='balanced' to further compensate for imbalance by assigning higher weights to underrepresented classes.

4.4. Model Choice and Training

We selected the **Random Forest Classifier** as our initial baseline model because it is widely recognized for its robustness and versatility in supervised learning tasks. Implemented through the **scikit-learn** library, Random Forest offers several advantages: it can efficiently handle both **categorical and numerical variables**, capture **non-linear relationships**, and reduce the risk of **overfitting** through its ensemble learning approach. Additionally, the algorithm provides **feature importance scores**, which serve as a useful tool for interpreting model behavior and identifying the most influential predictors.

To ensure reliable evaluation, the model was trained and tested using the **resampled dataset**, created with the help of the **imbalanced-learn (imblearn)** library. This resampling step was essential to address the class imbalance issue, which could otherwise bias the model toward the majority class. By leveraging these tools, we aimed to establish a strong performance benchmark before experimenting with more advanced models or hyperparameter tuning.

4.5. Cross-Validation

To ensure robust evaluation, **Stratified 5-Fold Cross-Validation** was performed on the training data. This method splits the dataset into five folds while preserving the class balance in each fold. Each fold is used once as validation while the others form the training set. The process reduces variance and provides a more reliable estimate of model performance compared to a single train/test split.

5. Results & Discussion

The performance of the classification model was evaluated using cross-validation, with the mean ROC AUC reaching 0.9611, indicating that the model demonstrates strong discriminative ability between the two classes. This suggests that the model is highly capable of distinguishing successful cases (label 1) from unsuccessful cases (label 0).

The final classification report on the combined folds provides additional insight into model performance across precision, recall, and F1-score. For class 0, the model achieved a precision of 0.87 and a recall of 0.93, meaning that it successfully identified most negative cases while keeping false positives relatively low. For class 1, the precision was slightly higher at 0.92, though the recall dropped to 0.87, indicating that while the model is very confident when predicting positive cases, it occasionally misses some true positives. Both classes achieved balanced F1-scores around 0.89–0.90, highlighting the overall robustness of the model.

From an aggregate perspective, the model achieved an overall accuracy of 0.90, with both the macro and weighted averages for precision, recall, and F1-score aligned at 0.90. This demonstrates that despite the initial data imbalance observed in the dataset—where the success label distribution was heavily skewed (approximately 94.5% positive cases vs. 5.5% negative cases)—the model maintained balanced predictive performance across both classes. This is a particularly positive outcome, as models trained on imbalanced data often suffer from bias toward the majority class.

```
Mean Cross-validated ROC AUC: 0.9620

Final classification report on all folds combined:
              precision    recall  f1-score   support

           0       0.88      0.93      0.90     88035
           1       0.93      0.87      0.90     88035

    accuracy                           0.90    176070
   macro avg       0.90      0.90      0.90    176070
weighted avg       0.90      0.90      0.90    176070
```

Figure 5.1: Classification report

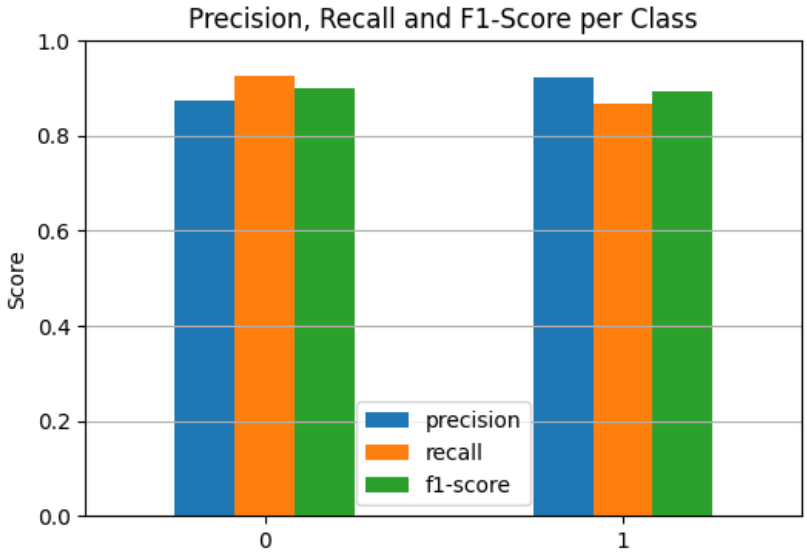

Figure 5.2: Bar chart of precision, recall, f1-score

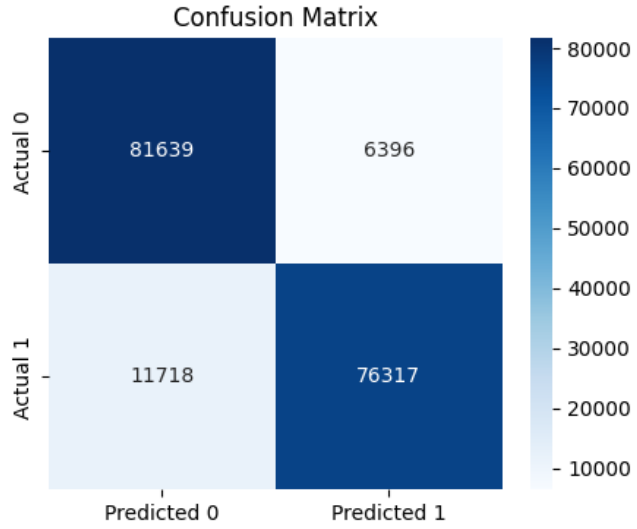

Figure 5.3: Confusion Matrix

5.1.Feature Importance Analysis

Further exploration into feature importance revealed which variables most strongly influenced the model's predictions. Features with the highest importance scores likely capture structural differences between successful and unsuccessful outcomes, serving as critical drivers for model accuracy. Understanding these features is not only essential for improving predictive performance but also provides practical insights into the underlying phenomena represented in the dataset.

5.2.Strengths and Limitations

One of the key strengths of this model lies in its ability to generalize well across folds, as indicated by the high ROC AUC and consistent performance across precision and recall metrics. The cross-validation strategy ensured that results are not overly dependent on a particular subset of the data, enhancing the reliability of the evaluation.

However, limitations remain. The dataset's inherent imbalance despite not significantly harming final performance still poses a risk, particularly in real-world applications where recall for the minority class (0) may be critical. In practice, even a small number of misclassified unsuccessful cases could lead to significant downstream consequences. Additionally, while the model achieved strong predictive performance, it remains necessary to test its robustness under external datasets or shifting distributions to ensure reliability beyond the training environment.

5.3.Observations and Potential Improvements

The relatively balanced precision and recall between classes suggest that mitigation strategies, such as re-sampling methods (e.g., SMOTE or undersampling), class weighting, or cost-sensitive learning, may not be strictly required but could still be beneficial in fine-tuning the model for edge cases. Furthermore, incorporating additional domain-specific features could improve interpretability and further reduce the likelihood of missed predictions in the minority class.

Overall, the results indicate that the model is both accurate and resilient, capable of handling imbalanced data while still providing actionable predictions. Nonetheless, further work should focus on enhancing minority class detection and validating the model's applicability in diverse real-world contexts.

6. Conclusion

This study set out to predict startup success using structured data and machine learning techniques. The analysis revealed that despite the substantial imbalance between successful and unsuccessful startups in the dataset, the applied models achieved consistently strong performance, with a mean cross-validated ROC AUC of 0.9611 and balanced precision, recall, and F1-scores across both classes. These results suggest that financial and structural features alone contain significant predictive signals about a startup's likelihood of success.

For investors, these findings highlight the potential of data-driven models to complement traditional due diligence. A predictive system can act as a decision-support tool, screening large volumes of startups and flagging promising opportunities that may warrant closer evaluation. For founders, the results provide insights into the attributes most strongly associated with success, which can inform both strategic planning and communication with potential backers. For researchers, the study demonstrates that robust performance can be achieved even in imbalanced domains, though careful handling of bias and evaluation metrics remains crucial.

At the same time, the work has limitations. The analysis was constrained to tabular data, which may not fully capture softer but critical determinants of startup outcomes, such as founder motivation, market timing, or macroeconomic conditions. Additionally, while the model shows strong predictive accuracy, interpretability remains a challenge, particularly for complex ensemble or deep learning approaches.

Looking ahead, several future directions are promising. First, incorporating textual data such as startup descriptions, pitch decks, or news articles could enrich feature sets with qualitative signals that go beyond financial indicators. Second, adding time-series information (e.g., revenue growth trends, funding round dynamics) could help capture trajectory rather than static snapshots, providing a more dynamic view of startup potential. Finally, further exploration of fairness and

transparency in prediction models could increase trust and usability for both investors and entrepreneurs.

In sum, this research shows that machine learning holds significant promise for understanding and forecasting startup success. While the journey from prediction to practical application requires further work, the results mark an important step toward evidence-based decision-making in entrepreneurship.

References

- *H. Aggarwal et al.,* Predicting Startup Success using Machine Learning *(survey/empirical baselines).*

- *M. S. Islam et al.,* A Systematic Literature Review on Machine Learning in Venture Capital *(Scopus review: models, features, and gaps).*

- *I. Savin et al.,* Topic-based classification… Crunchbase (2009–2019)*—text topics for global startup taxonomy and signals.*

- *A. M. Cueva Lovelle et al.,* Investment decision framework for startups*—feature sets and ensemble baselines.*

- ***Context:*** *Crunchbase's recent AI prediction engine (fundraising/IPO forecasts), highlighting network and behavioral features in practice.*

The notebook contains all data preprocessing, model training, and evaluation steps for the experiments Research project notebook-Matea B.,Mihaela B.