Startup Success Prediction – Final Report

Data Science Society Project
Team: Ariadni Papanikolaou, Robert Chen, Lan Nguy, Cathy Yang

1. Introduction

The high failure rate of new businesses makes predicting startup success an important task for the economy. For young companies, the journey from an initial idea to a successful outcome – such as being acquired or going public – is very risky. The main goal of this project is to use machine learning (ML) to build a reliable model that can estimate how likely a startup is to succeed.

This kind of prediction is especially valuable for investors, accelerators, and venture capital firms. They already understand the basic qualitative factors, such as the strength of the team or the quality of the idea. However, an AI model does not replace this human judgment. Instead, it supports it by providing a systematic, data-driven evaluation. In this project, "success" is defined as a binary outcome: a company either had a positive exit (acquisition or IPO) or closed down. The value of our model lies in offering a quantitative method to identify promising startups, going beyond intuition alone.

---

2. Dataset Description

The data used for this project comes from a public Kaggle dataset called "Startup Success Prediction." It contains information on a variety of startups and is well-suited for predictive modelling because it includes financial and operational metrics that real-world analysts often use.

Key characteristics of the dataset include:

- **Source:** Kaggle – Startup Success Prediction
- **Total Number of Startups:** 922
- **Initial Number of Features:** 48
- **Target Variable:** *status* (binary: acquired/IPO vs. closed)

An important aspect of the dataset is the imbalance in the target classes: successful startups are much fewer than closed ones. This imbalance is typical in success/failure prediction

tasks and influenced how we evaluated the models. The dataset also showed a clear geographical distribution, with San Francisco (14%) and New York (10%) being the main locations. Several variables also contained missing values (for example, location details), which required proper handling.

---

3. Data Preprocessing

The preprocessing of data comprises several sequential steps:

The first step concerns the treatment of missing values. Two distinct types of missingness appear in the dataset. The first arises from technical issues, such as the column named “Unnamed: 6,” which stores the geographic location and zip code together. It has missing values in 493 rows and is purely structural rather than informational, and is therefore filled using the mode to maintain consistency within its string-based format. Another type of missing value is due to the fact that the firms do not have certain kind of feature, such as closing dates (588 firms) and milestone years (152 firms). Since these features may still be relevant for modelling but cannot be left blank, they are imputed using the median, which provides a robust central tendency without being skewed by outliers.

A second step ensures that categorical features have corresponding dummy variables, as models only learn numbers. Core features include state code (geographical location), which is split into 5 dummy columns; category code (industry), which is split into 10 dummy columns; and some funding information that has already been stored as dummy columns. Ensuring consistent numerical encoding across these variables is necessary to avoid model misinterpretation

Thirdly, irrelevant text identifiers such as ID and zip code are dropped for model use.

Among the above three steps, feature selection and engineering occur simultaneously with data preprocessing, and the next step is to explain the reasoning of these steps before finally reaching a processed dataset.

---

4. Feature Engineering

A key variable in this project is the column “status” which includes only two values, namely “acquired” and “closed,” and is the core “y” that the project aims to estimate. This column is encoded using 1 to represent acquired (successful startups) and 0 to represent closed.

Apart from status, all other columns are the “x” features and can be grouped, according to the information they deliver, into three categories.

The first category consists of basic information such as IDs, names, labels, zip codes, etc. Most columns are in text form and cannot be read by models and are thus dropped. As a

result, 8 columns are left. Among these, founding and closing years provide firm's lifespan, while a top-500 indicator allows the model to differentiate firms with reputational or scale advantages. The remaining five dummy variables encode geographical information derived from state-level location data.

The second category includes funding and milestone information, which forms the largest component of the feature set. 16 variables are selected in total, including the years of first and last funding, ages at key funding moments, specific funding amounts, and six dummy variables representing different funding rounds. These features are essential because funding history is one of the strongest observable predictors of startup outcomes. Milestones secure substantial investment amounts and often signal investor confidence, operational performance, and general ability. For these reasons, most variables in this category are retained.

The third category describes industry information. Ten dummy variables represent ten industries are chosen and this is intuitive to see if certain industry is more friendly to startups.

After completing these selections, a total of 34 explanatory features remain, and the fully processed dataset consists of 923 rows and 35 columns, including the outcome variable.

---

5. Modeling Approach

The modelling processi involved building four classification models to predict startup success. The first model was **Logistic Regression**, used as a baseline linear classifier. Its main advantage is simplicity: it trains quickly, is easy to interpret, and provides a clear reference point for evaluating more advanced methods. However, its linear structure limits its ability to capture non-linear relationships or complex feature interactions.

To introduce non-linearity, a **Random Forest** model was developed. Random Forest is an ensemble of decision trees trained on bootstrapped samples of the data. This structure allows the model to learn more complex decision boundaries and detect interactions between variables. It is typically more flexible and accurate than Logistic Regression, though it requires greater computational resources due to the large number of trees.

A more advanced model, **XGBoost**, was then implemented. XGBoost is a gradient boosting algorithm that builds trees sequentially, with each new tree attempting to correct the previous errors. This boosting process enables the model to learn fine-grained patterns and handle both linear and non-linear relationships effectively. XGBoost also incorporates regularisation, which helps manage overfitting, but at the cost of increased complexity and longer training times.

Finally, an **XGBoost Tuned** model was created by adjusting hyperparameters such as learning rate, tree depth, and number of estimators. Hyperparameter tuning aims to improve the bias–variance trade-off and optimise performance for the specific dataset. This step substantially increases computational demand because many combinations must be evaluated, but it provides insight into how model behaviour changes under different configurations.T

---

6. Modeling Evaluation

The performance comparison across the four models shows a clear improvement as we move from linear to non-linear and then boosted methods. Logistic Regression performed the weakest overall, with accuracy **0.714** and F1-score **0.778**, likely due to its inability to learn non-linear relationships. Random Forest offered a strong improvement, particularly in recall (**0.84**), showing that tree-based methods are better at capturing more complex patterns. The XGBoost baseline continued this trend, achieving the highest overall F1-score (**0.8178**) and balanced precision–recall performance. The tuned version of XGBoost achieved the highest recall (**0.8583**) but showed only marginal improvements over the baseline, with a slight drop in precision.

The confusion matrices support these results. Logistic Regression produced the most false negatives, meaning it struggled to identify successful startups. Both Random Forest and XGBoost reduced these errors significantly, demonstrating stronger generalisation. Hyperparameter tuning improved recall but did not produce a major performance jump.

Overall, **XGBoost (baseline)** emerged as the best general-purpose model for this task, offering the strongest balance across all metrics and the most consistent classification performance.

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.714	0.782	0.775	0.778
Random Forest	0.75	0.79	0.84	0.81
XGBoost Baseline	0.7568	0.7953	0.8417	0.8178
XGBoost Tuned	0.7405	0.7687	0.8583	0.811

---

7. SHAP Interpretability

Although the XGBoost model achieved high accuracy, it can be difficult to interpret and is often seen as a “black box.” To address this, we used SHAP (SHapley Additive

exPlanations) to make the model's decisions understandable. SHAP explains how much each individual feature value contributes to the final prediction.

Global Feature Importance

The SHAP summary plot confirmed that the most influential features are those that show validation, progress, and financial support:

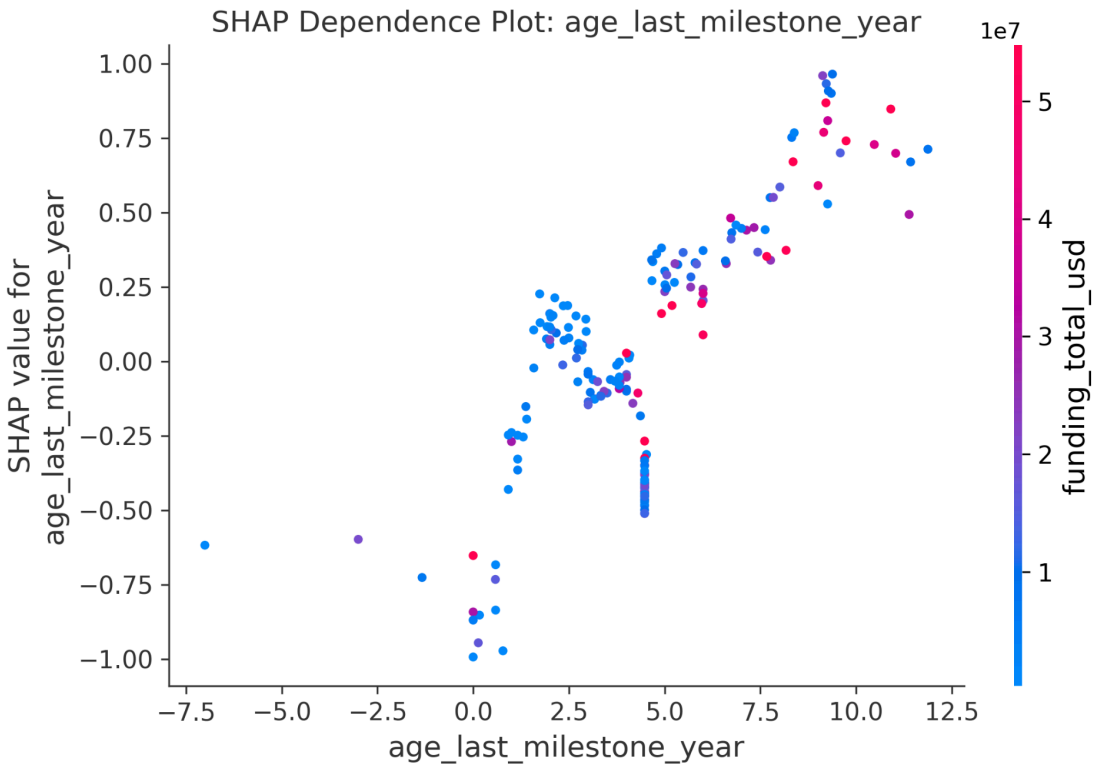
- **age_last_milestone_year:** The company's age (in years) at the time of its last milestone
- **milestones:** Total number of important milestones achieved
- **funding_total_usd:** Total funding received in USD

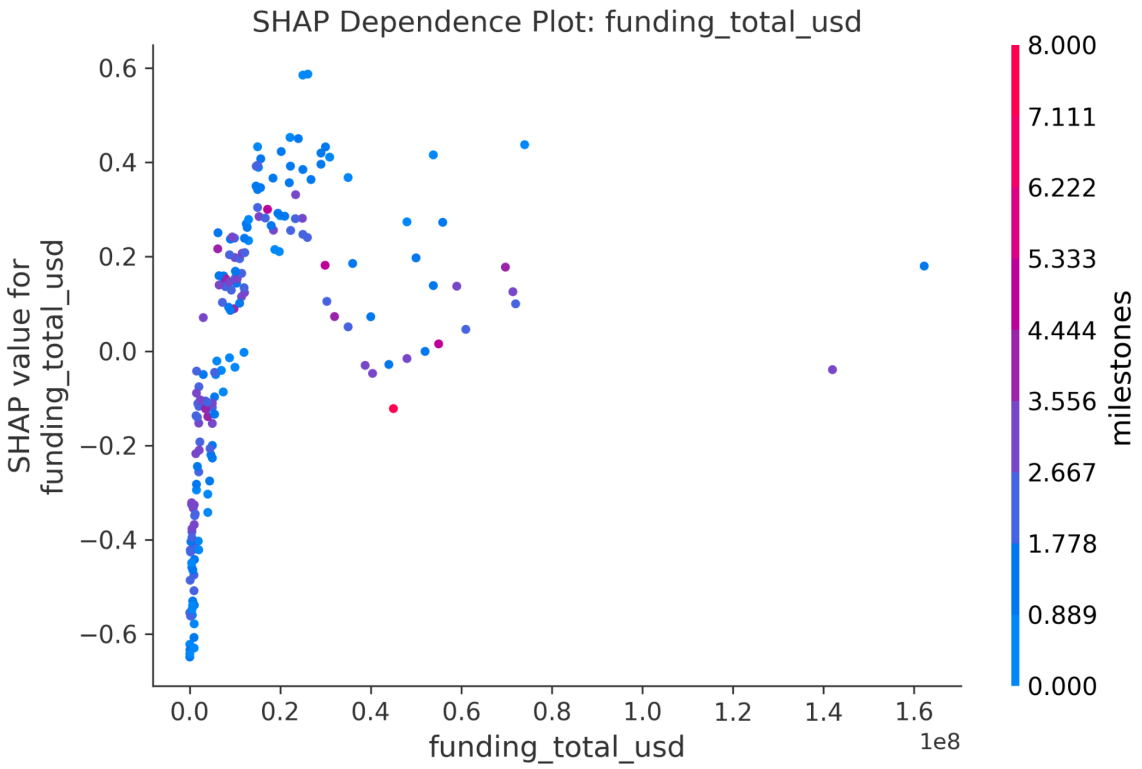
Insights from Dependence Plots

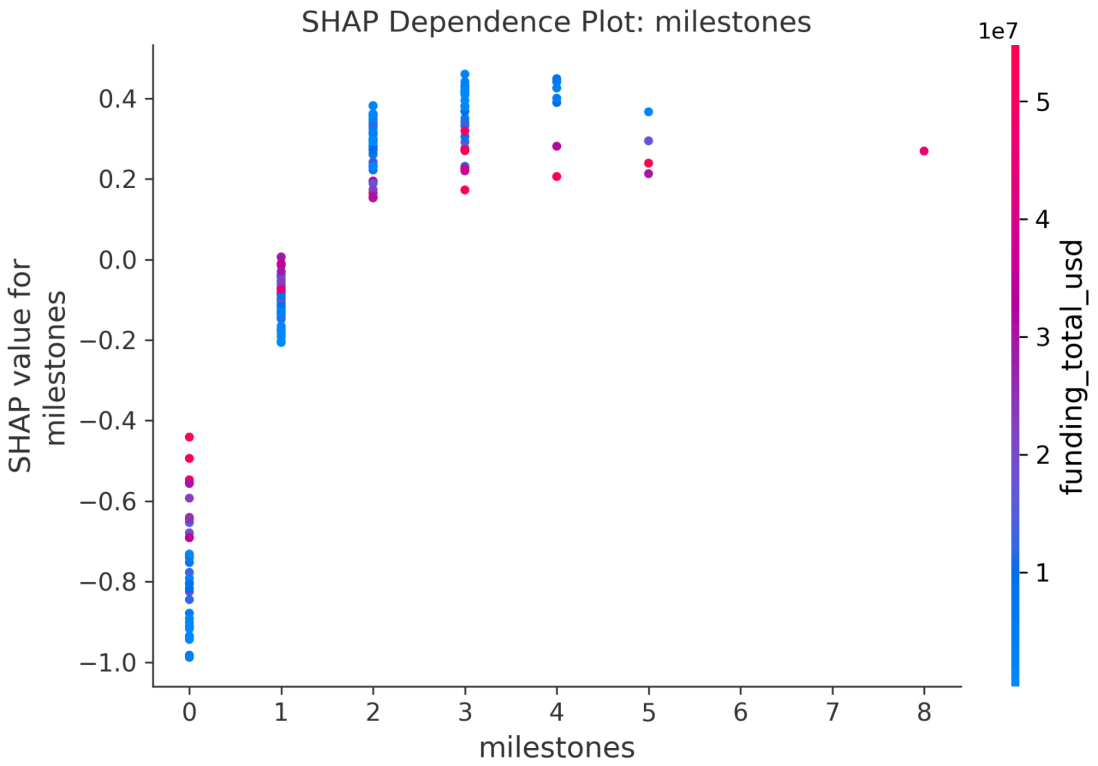
The detailed SHAP plots provided specific insights:

- **Milestones (Critical Threshold):** A clear success threshold appears at two milestones. Startups with zero or one milestone receive strongly negative SHAP values (predicting failure). From two milestones onwards, SHAP values increase significantly, suggesting a positive outcome. This confirms that reaching early major goals is crucial.
- **Age of Last Milestone (Endurance):** The strongest positive predictions occur when the last milestone was reached after five to six years or more. This suggests that endurance and steady development over several years are better indicators of success than achieving milestones very quickly. The model rewards companies that survive and continue progressing.

Overall, the SHAP analysis shows that the model learned similar patterns to those used by human experts: strong financial backing and continuous milestone achievement are key drivers of startup success.


SHAP Dependence Plot: age_last_milestone_year
SHAP value for age_last_milestone_year
age_last_milestone_year
funding_total_usd
1e7
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
5
4
3
2
1


SHAP Dependence Plot: funding_total_usd
SHAP value for funding_total_usd
funding_total_usd
1e8
milestones
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
8.000
7.111
6.222
5.333
4.444
3.556
2.667
1.778
0.889
0.000



---

8. Summary

In the project, an end-to-end Machine Learning pipeline is designed and implemented to predict the likelihood of startup success. Starting with a raw dataset of over 900 firms, a rigorous workflow comprising data cleaning, feature engineering, and exploratory data analysis is executed.

The project then progresses from baseline models, including Logistic Regression (captures linear interpretability) and Random Forest (captures non-linear relationships), to an advanced ensemble method, XGBoost (for hyperparameter tuning and maximum predictive power).

The evaluation identifies XGBoost as the strongest-performing model for this dataset. While Random Forest outperformed Logistic Regression with a higher F1-score (0.81 vs 0.778) and Recall (0.84 vs 0.775), XGBoost fine-tunes these predictions further.

To ensure the black-box issue remains transparent, SHAP analysis is used, and this reveals that success is not random but driven by specific, quantifiable behaviors:
- There is a strong positive correlation with success after the 5–6 year mark, which challenges the idea of rapid early wins.
- There is a critical threshold such that firms with 2+ milestones face a lower predicted risk of failure.
- More funding brings better opportunities for success, with around $20 million in total funding as an inflection point.

While the pipeline yields strong predictive results, there are several fields to enhance in terms of model and real-world applicability.

Firstly, future iterations could differentiate between funding rounds in more depth (Seed, Series A, Series B). The second is that survival analysis is suggested instead of binary classification, since startup success is time-dependent. Furthermore, an expanded dataset is suggested, with deeper tuning or neural networks explored, so that the text descriptions or other unstructured data that were dropped in this project could be fully used.

In terms of real-world application, it is suggested to deploy the inference pipeline via a web interface to make the model accessible to investors or founders, so that they can input startup characteristics and receive a real-time success probability score along with the top contributing risk factors.