Data Mining Analysis of Start-Up Growth Patterns

Course: Data Mining
Mentors: prof. dr. J. Jaklič, prof. dr. U. Godnov

Authors: Neli Perme , Eva Strašek, Minho Choi, Ivan Ulasau

2025/2026

Introduction

The collapse of the dot-com bubble in the early 2000s represented a pivotal moment in the evolution of the global startup ecosystem. The subsequent period gave rise to a markedly different entrepreneurial environment, shaped by increased regulatory oversight, more cautious investment strategies, and a pronounced shift toward sustainable revenue models and long-term value creation. Startups founded in the aftermath of this market correction were compelled to operate under heightened performance expectations, making the post–dot-com era a particularly relevant and analytically rich timeframe for studying startup success and failure.

The project explores the use of data analytics and predictive modelling to examine the eventual outcomes of startups established after the dot-com bubble burst. Drawing upon a curated dataset of post–dot-com ventures, the study seeks to classify startups into three distinct outcome categories: those that were shut down, those that remain operational, and those that were acquired. Among these outcomes, acquisition is widely regarded as the most favourable scenario, as it often signifies market validation, strategic relevance, and successful value creation. Contemporary mergers and acquisitions within the startup ecosystem—such as Microsoft's acquisition of Activision Blizzard and Salesforce's acquisition of Slack—underscore how innovative startups can evolve into critical assets for larger corporations seeking technological advancement, market expansion, or competitive advantage.

Business Problem Definition

Investing in early-stage startups involves substantial uncertainty, as many ventures fail while only a limited share achieve successful exits. For investors, the key challenge is therefore not only to identify high-potential startups, but also to **avoid investments with a high risk of failure**. Making such decisions is difficult when relying solely on qualitative assessments or limited financial information.

The business problem addressed in this project is to determine whether **observable startup characteristics** can be used to predict investment outcomes and support data-driven investment screening. Specifically, the objective is to classify startups into **risky investments** (startups that eventually close) and **positive outcomes** (startups that are acquired), where acquisition is treated as a completed and successful exit.

By framing the problem as a binary classification task, the analysis focuses on outcomes that are directly relevant to investors. The resulting predictive models aim to reduce exposure to failing startups while maintaining the ability to identify promising acquisition opportunities. In this way, the project demonstrates how data mining techniques can provide practical decision support in the venture capital and startup investment context.

To address the predictive challenge, the project adopts a dual analytical framework that integrates both Python-based data science methodologies and RapidMiner's AI-driven modelling environment. Python is employed for data preprocessing, exploratory analysis, feature engineering, and the implementation of machine learning algorithms through a code-centric approach. In parallel, RapidMiner is utilised to construct visual workflows that facilitate model development, evaluation, and comparison without extensive programming. This combined methodology enables a comprehensive assessment of predictive performance while highlighting the strengths and limitations of each analytical paradigm.

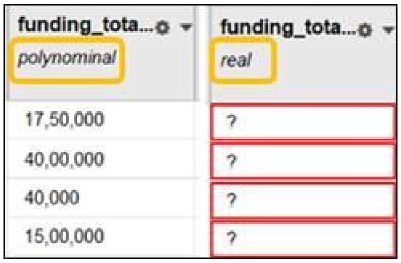
The report further examines a range of startup attributes—including funding dynamics, industry classification, temporal factors, and operational characteristics—to determine their relative influence on startup outcomes. By systematically evaluating these features across multiple models, the project aims to identify patterns that distinguish successful ventures from those that fail or stagnate.

The remainder of this report is structured as follows: an overview of the dataset and preprocessing techniques is presented first, followed by a detailed discussion of the predictive models developed in Python and RapidMiner. The report then compares model performance and interpretability before concluding with insights into the key determinants of startup success in the post–dot-com entrepreneurial landscape and potential directions for future research.

Data Preparation

Import Data and Initial Data Inspection

- The **Skip Comments (#)** option was disabled during data import because some startup names contained the **# character** (e.g., *#waywire*), which would otherwise cause those records to be incorrectly ignored.
- The **funding_total_usd** attribute was initially detected as **polynominal** due to inconsistent formatting, such as **comma-separated values** and **embedded spaces**. Converting it directly to a numeric type at import would have produced many missing values, so it was cleaned first and parsed later.



funding_tota... polynominal	funding_tota... real
17,50,000	?
40,00,000	?
40,000	?
15,00,000	?

Filter Examples and Rationale

- **founded_at > 01/01/1992**
  Startups founded from **1992 onward** were included to improve comparability, as firms founded earlier often operated under different market and financing conditions. In the early 1990s, **venture capital practices** became more standardized and **structured funding stages** became more common.
- **Status does not contain operating**

Startups with an **operating** status were excluded to align the dataset with the business objective of distinguishing **clear outcomes**. The analysis focuses on **acquired** versus **closed** firms, where acquisition indicates an exit and closure indicates failure, while operating firms remain unresolved at the time of observation.

Replace and Parse Number

- The **funding_total_usd** attribute was cleaned using **Replace operators** to remove **commas** and **spaces** and then converted from **polynominal** to **numeric** using Parse Numbers, enabling quantitative analysis.

Generate Attributes

· Outcome was defined as **0** for **closed** and **1** for **acquired**, treating acquisition as a **completed exit**; operating firms were excluded in the filtering step to avoid unresolved outcomes.

- The **firm_age** attribute was generated to capture startup **maturity**. Firm age is a relevant factor in modeling startup **outcomes**, as firms that have existed for longer periods have moved beyond early stages of **uncertainty** and are more likely to reach **stable states** or achieve successful exits.

Select Attributes

- After generating the **outcome** variable, the original **status** attribute was removed from the feature set to prevent **data leakage**, ensuring that the model does not use information directly related to the target label during training or evaluation.

Split Data

- The dataset was split into **70% training** and **30% testing** data using a **0.7 / 0.3** ratio to evaluate model performance on unseen data while retaining sufficient observations for training.

SMOTE Upsampling

- **SMOTE Upsampling** was applied **after data splitting** and **only to the training set** to address class imbalance between acquired and closed startups and to avoid inflated test performance.
- The **number of neighbours** was set to **3** to generate synthetic samples based on nearby minority instances while avoiding excessive smoothing.

- The **minority class** was detected automatically using the **auto detect minority class** option.
- The **equalize classes** option was enabled to balance the target classes during model training.
- **Normalization** was applied to support distance-based calculations during the upsampling process.

Modeling Processes and Results

Modeling Objective and Evaluation Logic

The objective of the modeling phase is to support **investment decision-making** by predicting whether a startup represents a **risky investment (closed = 0)** or a **positive outcome (acquired = 1)**.

From an investor's perspective, not all errors are equally costly:

- Misclassifying a *closed* startup as *acquired* may lead to financial losses
- Misclassifying an *acquired* startup as *closed* may lead to missed opportunities

For this reason, model evaluation focuses especially on:

- Recall for closed startups (class 0) -ability to detect risky investments
- Precision for acquired startups (class 1) -confidence that predicted "good" investments are truly attractive from investors perspective

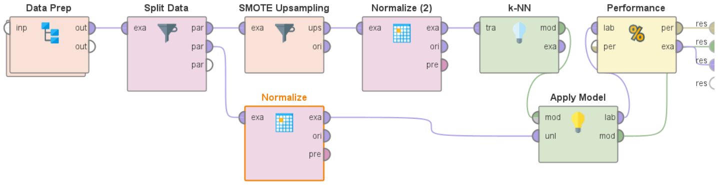
RapidMiner Modeling

All RapidMiner models were trained and evaluated using a consistent workflow, including assignment of the target /label variable (outcome: acquired/closed), a 70/30 train–test split, and SMOTE upsampling applied only to the training data.This standardized setup ensured a fair and comparable evaluation of model performance on unseen data. Extensive hyperparameter testing was conducted for each model; however, only the most relevant configurations and final results are presented for clarity. The results are summarized in the following table.

Model	Accuracy	Class_Recall_0_0	Class_Recall_1_1	Cl. Precision_0_0	Cl. Precision_1_1
k-NN	72.40%	70.59%	73.58%	63.50%	79.34%
Random forest	75.36%	84.41%	61.46%	77.08%	71.97%
Decision Tree	74.48%	52.74%	88.64%	75.14%	74.23%
Gradient Boosted Trees	72.72%	66.94%	76.49%	64.96%	78.03%

k-Nearest Neighbors (k-NN)

The k-Nearest Neighbors (k-NN) algorithm was implemented as a baseline distance-based classifier. Because k-NN relies on distance calculations, normalization was applied to all numerical features, the train and test branch. Without normalization, variables such as *funding amount* would dominate the distance metric and distort similarity calculations.Although k-NN achieved reasonable overall accuracy, its recall for closed startups was relatively weak. This means the model struggled to consistently identify risky investments, which limits its usefulness for investors compared to other models.



As k-NN does not learn explicit decision rules and is sensitive to noise, it serves mainly as a **benchmark** rather than a final decision-support model.

Decision Tree

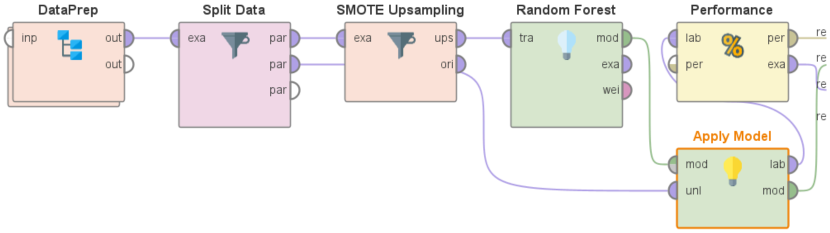
The Decision Tree model achieved an accuracy of approximately 74%.
It showed very high recall for acquired startups (≈ 88%), meaning that most successful exits were correctly identified. However, recall for closed startups was close to random guessing (≈ 52%), indicating that nearly half of the risky investments were incorrectly classified as positive outcomes. From a business perspective, this is problematic, as it exposes investors to avoidable risk.

Random Forest – Preferred RapidMiner Model

The **Random Forest** model provided the **best balance between risk detection and predictive reliability** among the RapidMiner models:

- Accuracy around **75%**
- **Recall for closed startups improved to ≈ 84%**, significantly reducing undetected risky investments
- **Precision for acquired startups remained high**, meaning predicted positive outcomes were relatively trustworthy

From an investor's perspective, this balance is crucial; higher recall for closed startups reduces the likelihood of investing in failing companies and adequate precision ensures that predicted "good" investments are credible.Due to this balanced performance and robustness, **Random Forest is selected as the preferred RapidMiner model** for investment screening.



Gradient Boosted Trees

Gradient Boosted Trees were tested as an additional ensemble method. The model achieved accuracy comparable to Decision Trees but showed no clear advantage over Random Forest in identifying risky investments.Given its higher complexity and limited interpretability, and the strong performance already achieved by Random Forest, Gradient Boosted Trees were not further optimized and are treated as an exploratory comparison.

Python-Based Models and Cross-Validation EVA I JUST PUT THIS HERE SO WE CAN IMAGINE THE STRUCTURE