Towards Automated Venture Capital Screening

W.M.R. Shelton

This report is submitted as partial fulfilment
of the requirements for the Honours Programme of the
School of Computer Science and Software Engineering,
The University of Western Australia,
2017

Abstract

Venture Capital (VC) firms face the challenge of identifying a few outstanding investments from a sea of opportunities. The VC industry requires better systems to manage labour-intensive tasks like investment screening. Previous approaches to improve VC investment screening have common limitations: small, private datasets, a focus on early-stage investment, and basic feature sets. To address these limitations, a multi-stage VC investment screening system is presented. The core of the system is an optimisation process that generates a supervised learning classifier which is applied to data collected from large, public online databases (CrunchBase and PatentsView). The system is evaluated against three criteria: practicality, robustness, and versatility. The system satisfies each of these criteria. The system is practical in that it is near-autonomous. The system is robust in that it has only minimal variance in performance when trained on historical datasets. Finally, the system is versatile in that it addresses a large domain of investment prediction tasks with respect to forecast window, developmental stage and target outcome. This project also contributes a comprehensive empirical study of startup performance. The prior experience of a startup's advisors, executives and founders is found to be the greatest predictor of startup performance. Ultimately, this project makes steps towards automation in the VC industry.

Keywords: Venture Capital, Investment Screening, Machine Learning
CR Categories: I.5, J.1, J.4

Acknowledgements

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Venture Capital (VC) is financial capital provided to early-stage, high-potential, high-growth companies (startups). VC firms have funded many successful companies, such as Google, Apple, Microsoft and Alibaba. Unlike investors in the public markets, VC firms often take a more active role in managing their investments, providing expertise and advice in both managerial and technical areas. VC firms have two primary roles: as scouts, identifying the potential of new startups, and as coaches, helping startups realise that potential [5]. In these ways, VC is often critical to the success of startups, and therfore the commercialisation of new technologies more generally. Adoption of the Internet and inexpensive, ubiquitous computing has transformed the VC industry: companies require less funding to launch but more to scale in highly competitive markets [23]. There is now an impetus for VC firms to change the way they operate.

VC firms face the challenge of choosing a few outstanding investments from a sea of hundreds of thousands of potential opportunities. VC firms seek to make investments in companies that can provide a liquidity event that returns many times their investment value within the time frame of their fund. For startups, a liquidity event (also referred to as an 'exit') is either an Initial Public Offering (IPO) or an acquisition by a larger competitor. Most VC firms expect their investments to exit within 3-6 years, in accordance with their fund time-frame [22]. When compared to public market investors, this is a long-term investment strategy. However, few companies are capable of maturing from early-stage to exit at this pace. In addition, traditional metrics of performance (e.g. cash-flow, earnings) often do not exist or are unclear [44]. In summary, VC firms must select from a field of many investment candidates, where little information is available on each of them, and only a few will grow at a fast enough rate to be worthwhile – this is why the VC investment screening process is considered difficult.

The VC industry requires better systems and processes to efficiently manage labour-intensive tasks like investment origination and screening. Currently, investment opportunities are either referred or identified through technology scans (e.g. Google searches, patent searches). These manual search processes are time-

consuming for VC firms. Attempts in the literature to solve this problem have three common limitations: small sample size [1, 20, 16, 25, 54, 3, 52, 14], a focus on early-stage investment [6, 1, 12, 55, 14, 48], and a basic feature set [1, 3, 12, 14, 52, 20]. Although individual studies address some of these limitations, none synthesise the findings into software ready for use in industry. The popularity of online databases like AngelList and CrunchBase , which offer information on startups, investments and investors, is evidence of the VC industry's desire for better, more quantitative methods of assessing startup potential [35]. There is preliminary evidence that mining these data sources may address previous limitations and make VC investment screening more efficient and effective [48, 7].

We believe it is now possible to address previous limitations in this field and produce an improved VC investment screening system. Our system aims to identify startup companies that are likely to raise additional funding, become acquired or have an IPO (or some combination therof) in a given period of time. This system may assist VC firms to efficiently and effectively screen investment candidates. To be useful in this context, the system must meet the following criteria:

1. Practicality. The system must be practical for use in the VC industry and more efficient to use than manual investment screening. The system should be designed to operate with minimal user input and no assumed technical expertise. The system should also be designed to run in reasonable time.

2. Robustness. The system must be robust to changes over time. The system should be designed to have minimal variance in performance when training on datasets from different times so investors can trust its ability to make future-looking predictions. The system should also be designed to adapt to the quantum and type of data available from data sources over time.

3. Versatility. The system must be versatile in its ability to address a large domain of investment prediction tasks. The system should be designed to make accurate predictions for companies of different developmental stages (e.g. Seed, Series A), for different target outcomes (e.g. Acquisition, IPO) across different forecast windows (e.g. exit in two years, exit in four years).

This thesis is organised as follows:

- Chapter 2: Literature Review. We review the theoretical background of startup performance and VC investment and evaluate previous efforts in developing technologies for use in VC investment screening systems.

- Chapter 3: Design. We outline the design of our system architecture, propose a framework that guides our feature and data source selection, and describe the development of our adaptive classification pipeline process.
- Chapter 4: Evaluation. We perform a series of experiments to evaluate our system against three criteria: practicality, robustness and versatility.
- Chapter 5: Discussion. We discuss the merits and limitations of our project and their implications for investors and future research.

CHAPTER 2

Literature Review

In this chapter, we first review the startup investment literature to develop criteria to evaluate our Venture Capital (VC) investment screening system. We then turn our focus to determining the best techniques to use to create this system, which we break down into three intercorrelated areas: feature selection, data sources and classification algorithms.

1. Criteria Selection. VC firms review many potential investment candidates to short-list for investment. Traditional screening methods involve referral, networking and Internet search. These screening methods are highly time-consuming and subject to human selection biases. Based on our review, we believe that a superior system can be produced and should be assessed on the basis of its efficiency, robustness, and predictive power.

2. Feature Selection. VC is a key driver of startup development but our understanding of factors that influence VC firms' investment decisions and the subsequent performance of those investments is incomplete.

3. Data Sources. Startup performance is a multi-faceted problem and different data sources provide insights into different actors, relationships and attributes. Our review focuses on novel online data sources which have the potential to transform entrepreneurship and VC research. Preliminary evidence suggests that the online startup databases CrunchBase and AngelList are promising and likely to provide a comprehensive feature set that can form the basis of our system. Other sources like PatentsView, Twitter, LinkedIn, and PrivCo are considered.

4. Classification Algorithms. Predicting startup performance is a difficult problem for humans. After all, a high percentage of even VC-backed startups still fail. However, machine learning techniques have been recently used in other areas of finance (e.g. in the public markets) with some success. We cross-reference the characteristics of our intended dataset with

the characteristics of common supervised classification algorithms. Our analyses suggest that we should expect Random Forests, Support Vector Machines and Artificial Neural Networks to be most suitable for our system.

2.1 Criteria Selection

Venture Capital (VC) financing has lagged behind other forms of high finance (e.g. bond trading, loan applications, insurance) in adopting computational analytics to aid decision-making. Banks are now able to evaluate personal loan requests in minutes while VC firms take far longer to put together deals, sometimes months. While these are markedly different forms of finance (VC has a longer return period, larger investments, higher risk profiles), a more data-informed and analytical approach to venture finance is still foreseeable.

In this section, we provide an introduction into VC firm strategy and review the existing state of the VC investment process. We find that analytical tools are nascent and use of analytics in industry is limited. To date only a small handful of VC firms have publicly declared their use of computational analytical methods in their decision making and investment selection process. We explore why the use of data mining in the VC industry is limited and we develop criteria by which we can judge a VC investment screening system to be successful.

2.1.1 Venture Capital Industry

Early-stage investment is a key driving force of technological innovation and is vitally important to the wider economy, especially in high-growth and technology intensive industries (e.g software, medical and agricultural technologies). VC is a form of private equity, a medium to long-term form of finance provided in return for an equity stake in potentially high growth companies. Reported US VC investments in 2015 totalled US$60 billion [33].

Typically, VC firms are reliant on a small number of high-risk investments to produce outsized returns through successful exit events. A common rule-of-thumb is that given a portfolio of ten startup companies: three will fail entirely, three will remain active but will not be very profitable, three will be active and profitable, and one highly successful startup will provide the investor with a multiple return on all of the investments [48]. In comparison to other traditional investment classes, VC financing is heavily biased towards control at the expense of risk mitigation. Although VC firms tend not to take majority stakes in startups, they exert their influence through significant minority stakes, board membership,

their relative seniority to the company's founders, and through leveraging their business networks [17].

Despite VC firms', often significant, influence on the trajectory of their investments, they are still highly selective of the companies that they invest in. Although rarely reported, a small number of studies show VC investment rates vary between 1.5-3.5% of proposals considered [48]. Accordingly, traditional venture finance is a very labour intensive and time consuming process involving extensive due diligence on behalf of the investor [18]. The VC investment process involves several main stages: deal origination, screening, evaluation, structuring (e.g., valuation, term sheets), and post investment activities (e.g., recruiting, financing).

2.1.2 Venture Capital Systems

Early-stage investment is characterised by a large number of investment candidates, high degree of uncertainty; a lack of reliable data on company performance (particularly financial performance); and a high time-cost of undertaking due diligence. This makes for a complicated origination and screening process. While referral from trusted sources (e.g., entrepreneurs, accountants, lawyers, other investors) is often used to screen opportunities, as the cost of starting businesses dramatically decreases investors are faced with an increasingly large number potential businesses and investment opportunities to assess and evaluate. This has led to an "information overload" problem in venture capital.

Despite evidence that VC firms could benefit from increased use of data mining, it appears few are interested in advanced data analytics. Stone [48] interviewed Fred Wilson of Union Square Ventures who said: "We have not been able to quantify [startup potential]. We haven't even tried. Although I am sure someone could do it and they might be very successful with it. To us, the ideal founding team is one supremely talented product oriented founder and one, two, or three strong developers, and nothing else." Likewise, when asked, Chris Dixon of Andreessen Horowitz said: "I've seen a few attempts to do it quantitatively but I think those are often flawed because the quantitatively measurable things are either obvious, irrelevant, or suffer from over-fitting (finding patterns in the past that don't carry forward in the future)".

Similarly, while recently new software tools have been developed to assist VC firm, there is limited evidence of their adoption. Stone (2014) found that adoption of technology in the VC industry is generally limited to larger, later-stage investment firms [48]. Stone suggested that due to the overhead of implementing such software tools, the perceived benefits may be realised only by larger organi-

sations with larger prospective deal flow. Furthermore, we suggest that in field of finance largely defined by emphasising control at the expense of diversification, there may be also be psychological barriers for VC investors to rely or cede any form of control to technological systems.

2.1.3 Proposed Criteria

Based on our review of the VC industry and current VC origination and screening processes, we have developed criteria on which we can evaluate our proposed system.

1. Efficiency. Our system must be more efficient than traditional, manual investment screening by referral and technology scan (e.g. Google search, media, databases). This means that it needs to be able to provide enough information (observations and features) to meet similar levels of accuracy.

2. Robustness. Our system must be robust enough to be reliable over time. The system must provide a generalised, robust solution for investors that does not require significant technical knowledge to use, and is not over-fitted to a specific time-period or data source.

3. Predictive Power. Our system must be consistently accurate at identifying a variety of high-potential investment candidates. The system should be robust to different forecast windows (i.e. exit in three years from now) as VC firms make investment decisions with different periods so they can strategically manage the investment horizons of their funds.

2.2 Feature Selection

Our understanding of the factors that influence Venture Capital (VC) investment decisions and the subsequent performance of those investments is incomplete. We believe a diverse range of features is critical to developing accurate models of startup performance and investment decisions.

Prior work focuses on basic company features (e.g. the headquarters' location, the age of the company) for startup investment predictive models [6, 20]. Semantic text features (e.g. patents, media) [25, 55] and social network features (e.g. co-investment networks) [52, 12, 54] may also predict startup investment. We expect a model that includes semantic text and social network features alongside basic company features could lead to better startup investment prediction.

Ahlers and colleagues

Their framework has two factors: venture quality and level of uncertainty. The first factor is based on work by Baum and Silverman [5] that suggests key determinants of startup potential are human capital, alliance (social) capital, and intellectual (structural) capital. The second factor is based on investors' confidence in their estimation of startup potential.

Next, we must operationalise this conceptual framework into features that we can incorporate into our machine learning model. Table 2.1 shows a review of features tested in previous studies of startup investment. In Appendix C, we describe each of these features and outline theoretical and empirical evidence that justify their inclusion in our conceptual framework.

2.3 Data Sources

Predicting startup investment and performance is a complex and difficult task. There are many features that can influence startup investment decisions. Capturing the diversity of these features is critical to developing accurate models. Accordingly, this task will likely involve data collection from multiple data sources. Appropriate selection of these data sources is important because different data sources provide insights into different actors, relationships and attributes.

Previous studies in this field have been limited by data sources restricted in sample size. Many studies have samples of fewer than 500 startups [1, 20] or between 500 and 2,000 startups [25, 54, 3, 52, 14]. Only a few studies have used large scale samples (more than 100,000 startups), usually derived from CrunchBase or AngelList [43, 12]. Sample size is more critical to model development than the sophistication of machine learning algorithms or feature selection [10]. Startup databases (e.g. CrunchBase) and social networks (e.g. Twitter) offer data sets larger than those used in many previous studies. We expect data collected from these sources will lead to the discovery of additional features and higher accuracy in startup investment prediction.

In Table 2.2, we outline the characteristics of relevant data sources and how they could contribute to our chosen features. In this section, we describe desirable characteristics of data sources for this task, review potentially relevant data sources, and ultimately determine which data sources are most likely to suit the characteristics of this task.

Features	Results from Studies	
	Significant	Non-Significant
Startup Potential		
Human Capital		
Founder Capabilities	[6, 3, 20]	[43, 13]
Advisor Capabilities	[5]	[1, 3]
Executive Capabilities	[6, 3, 13]	[1]
Social Capital		
Strategic Alliances	[5]	-
Social Influence	[6, 3, 12, 54]	-
Structural Capital		
Patent Filings	[25, 26, 5]	[1, 20]
Investment Confidence		
Third Party Validation		
Investment Record	[1, 6, 14, 25, 13]	-
Investor Reputation	[3, 52, 26]	[25]
Media Coverage	[6]	[3]
Historical Performance		
Financial Performance	[6, 5]	-
Non-Financial Performance	[3, 20]	[25]
Contextual Cues		
Industry Performance	[43, 14, 20]	[6, 13]
Broader Economy	[6, 14, 25, 13, 26]	[43, 1]
Local Economy	[43, 6, 14, 20, 25]	-

Table 2.1: Features relevant to startup investment. We review thirteen empirical studies that investigate drivers of startup investment. For each study, we note whether included features have a significant effect on the startup investment model. We classify identified features according to our proposed conceptual framework.

2.3.1 Source Characteristics

Entrepreneurship research is transforming with the availability of online data sources: databases, websites and social networks. Entrepreneurship studies have historically relied on surveys and interviews for data collection. Measures of human capital (e.g. founders' capabilities), strategic alliances, and financial performance are difficult to capture elsewhere. However, the trade-off for access to these features is that surveys and interviews are time-consuming and costly to

Properties	Startup Databases		Social Media		Other Sources	
	CrunchBase	AngelList	LinkedIn	Twitter	PatentsView	PrivCo
Features						
Startup Potential						
Human Capital						
Founders' Capabilities	✓	✓	✓✓	✗	✗	✗
NED Capabilities	✓	✓	✓✓	✗	✗	✗
Staff Capabilities	✓	✓	✓✓	✗	✗	✗
Social Capital						
Social Influence	✓	✓✓	✓✓	✓✓	✗	✗
Strategic Alliances	✓	✓	✗	✗	✓	✗
Structural Capital						
Patent Filings	✗	✗	✗	✗	✓✓	✗
Investment Confidence						
Third Party Validation						
Investment Record	✓✓	✓✓	✗	✗	✗	✓
Investor Reputation	✓	✓✓	✓	✗	✗	✗
Media Coverage	✓✓	✓	✗	✓	✗	✗
Awards and Grants	✓	✗	✗	✗	✗	✗
Historical Performance						
Financial Performance	✗	✗	✗	✗	✗	✓✓
Non-Financial Performance	✓✓	✓✓	✓	✗	✗	✓
Contextual Cues						
Competitor Performance	✓	✓	✗	✗	✗	✗
Broader Economy	✓	✓	✗	✗	✗	✗
Local Economy	✓	✓	✗	✗	✗	✗
Ease of Use						
Cost Effective	✓	✓✓	✓	✗	✓✓	✗
Time Efficient	✓✓	✓✓	✗	✓✓	✓✓	✗
Accurate Data	✓	✓	✓✓	✓✓	✓✓	✓✓
Large Data Set	✓✓	✓✓	✓✓	✓✓	✓✓	✓

Table 2.2: Data sources relevant to startup investment. We reviewed six data sources commonly used in entrepreneurship research for their suitability for our startup investment task. We evaluated data sources for their ability to provide relevant features for our analyses and for their ease of use in data collection. We excluded offline sources from our analyses. Ratings are: ✗ = poor, ✓ = satisfactory, ✓✓ = good.

implement. While online surveys address some of these issues, it is still difficult to motivate potential participants to contribute. Online data sources like startup databases and social networks are efficient because collecting data is a secondary function of users interacting with these sources. Researchers can also collect data from these sources automatically and at scale. For these reasons, we only consider online data sources for inclusion in this study, specifically crowd-sourced startup databases (e.g. CrunchBase, AngelList), social networks (e.g. Twitter, LinkedIn), government patent databases (e.g. PatentsView) and private company intelligence providers (e.g. PrivCo). We review the characteristics of each of these data sources commonly used in entrepreneurship research in Appendix A.

2.3.2 Source Evaluation

Entrepreneurship and Venture Capital (VC) research is primed to take advantage of the availability of new online data sources. We evaluated relevant data sources for their suitability to predicting startup investment. Startup databases CrunchBase and AngelList provide the most comprehensive set of features. There are small differences between the features recorded by each. CrunchBase has slightly more coverage and tracks media better but lacks AngelList's social network. At least one startup database should be used and either are satisfactory. Of the other data sources we review, PatentsView is the most promising. PatentsView provides comprehensive patent information, though it could prove difficult matching identities to other sources. Other data sources are less promising because of access issues. LinkedIn cannot be easily collected now the API is deprecated. Twitter provides social network topology and basic profile information through its free API but does not provide access to historical tweets. Financial reports are too expensive for the purposes of this study.

2.4 Classification Algorithms

Predicting startup performance is a difficult problem for humans. Computational analytics have been heavily deployed in high finance and we believe there is scope for applying related techniques to improve upon investment decision making in the domain of venture finance. Machine learning is characterised by algorithms that improve their ability to reason about a given phenomenon given greater observation and/or interaction with said phenomenon. Mitchell provides a formal definition of machine learning in operational terms: "A computer program is said to learn from experience E with respect to some class of tasks T and performance

measure P if its performance at tasks in T, as measured by P, improves with experience E." [32].

Machine learning algorithms can be classified based on the nature of the feedback available to them: supervised learning, where the algorithm is given example inputs and desired outputs; unsupervised learning, where no labels are provided and the algorithm must find structure in its input; and reinforcement learning, where the algorithm interacts with a dynamic environment to perform a certain goal. These algorithms can be further categorised by desired output: classification, supervised learning that divides inputs into two or more classes; regression, supervised learning that maps inputs to a continuous output space; and clustering, unsupervised learning that divides inputs into two or more classes.

We evaluated common machine learning algorithms with respect to their suitability for predicting startup investment. In Table 2.3, we rank these algorithms by cross-referencing their assumptions and properties with the task characteristics. In the following sections, we describe the characteristics of the startup investment prediction task, review common machine learning algorithms, and determine which algorithms are most likely to suit the characteristics of this task.

2.4.1 Task Characteristics

Machine learning tasks are diverse. Our investigation into startup investment is a task that suits supervised machine learning algorithms. We will manipulate the data we collect into a single labelled data set. Startups will be labelled based on whether they are acquired or have had an IPO at a later time. The key objective of machine learning algorithm selection is to find algorithms that make assumptions consistent with the structure of the problem (e.g. tolerance to missing values, mixed feature types, imbalanced classes) and suit the constraints of the desired solution (e.g. time available, incremental learning, interpretability). In the following sections, we outline the characteristics of supervised learning tasks relevant to our startup investment prediction task.

2.4.1.1 Data Set Properties

While data sets can be pre-processed to assist with their standardisation, some types of data sets are still better addressed by particular algorithms. Data set properties like missing data, irrelevant features, and imbalanced classes all have an effect on classification algorithms. Data sets often have missing values, where no data is stored for a feature of an observation. Missing data can occur because

Criteria	Machine Learning Algorithms													
	NB		LR		KNN		DT		RF		SVM		ANN	
Data Set Properties	**2**		4		6		**2**		**1**		5		7	
Missing Values	✓✓	[29]	✓	-	✗	[29]	✓✓	[29]	✓✓	[49]	✓	[29]	✗	[29]
Mixed Feature Types	✓✓	[29]	✓✓	-	✓✓	[29]	✓✓	[29]	✓	[49]	✓	[29]	✓	[29]
Irrelevant Features	✗	[29]	✗	[30]	✓	[29]	✓✓	[29]	✓✓	[49]	✗	[29]	✗	[29]
Imbalanced Classes	✓✓	-	✓✓	-	✗	-	✗	[29]	✓	[49]	✓✓	[29]	✓	[29]
Algorithm Properties	**2**		**1**		5		5		**2**		5		**2**	
Predictive Power	✗	[10]	✓	[10]	✓	[10]	✗	[29]	✓✓	[10]	✓✓	[10]	✗✓	[10]
Interpretability	✓✓	[29]	✓✓	[30]	✗	[29]	✓✓	[29]	✓	[30]	✗	[29]	✗	[29]
Incremental Learning	✓✓	[29]	✓✓	-	✓✓	[29]	✓	[29]	✓	-	✓	[29]	✓✓	[29]
Overall	**2**		**2**		6		4		**1**		5		6	

Table 2.3: Evaluation of machine learning algorithms for startup investment prediction. We reviewed seven common supervised machine learning algorithms for their suitability for our startup investment task. We evaluated algorithms for their robustness to the structure of the data set and their appropriateness for the constraints of our implementation. We ranked the algorithms according to the sum of these measures (in each section and overall) and emphasised highly-ranked algorithms. Ratings are: ✗ = poor, ✓ = satisfactory, ✓✓ = good. Algorithms are: NB = Naive Bayes, LR = Logistic Regression, KNN = K-Nearest Neighbours, DT = Decision Trees, RF = Random Forests, SVM = Support Vector Machines, ANN = Artificial Neural Networks.

of non-response or due to errors in data collection or processing. Missing data has different effects depending on its distribution through the data set. Public data sets, like startup databases and social networks, are typically sparse with missing entries despite their scale. Therefore, robustness to missing values is a desirable property of our algorithm. Despite efforts to only include features that have theoretical relevance, machine learning tasks often include irrelevant features. Irrelevant features have no underlying relationship with classification. Depending on how they are handled they may affect classification or slow the algorithm. We expect irrelevant and non-orthogonal features in our data set because our proposed framework includes features that have not been thoroughly tested in the literature. Therefore, robustness to irrelevant features is a desirable property of our algorithm. Data sets are not usually restricted to containing equal proportions of different classes. Significantly imbalanced classes are problematic for some classifiers. In the worst case, a learning algorithm could simply classify every example as the majority class. Our data set is not dramatically imbalanced overall, but when looking at funding status for different funding rounds it is significantly imbalanced. Therefore, robustness to imbalanced classes is a desirable property of our algorithm.

2.4.1.2 Algorithm Properties

The desired properties of machine learning algorithms are related to the business problems that are being addressed. Predictive power, interpretability and processing speed are all desirable characteristics but involve trade-offs and must be prioritised. Predictive power is the ability of a machine learning algorithm to correctly classify new observations. Predictive power can be evaluated in many ways. As the data set is likely to have an imbalanced class distribution, we will evaluate predictive power based on balanced metrics like Area under the Receiver-Operator Curve and the F1 Score. If a model has no predictive power, the model is not representing the underlying process being studied. For this reason, predictive power is a desirable property of our algorithm. However, if multiple algorithms provide similar predictive power other selection criteria become significant. Interpretability is the extent to which the reasoning of a model can be communicated to the end-user. There is a trade-off between model complexity and model interpretability. Some models are a "black box" in the sense that data comes in and out but the model cannot be interpreted. For this study, it is a key objective that we improve our understanding of the determinants of startup investment. Therefore, interpretability is a desirable property of our algorithm. Finally, processing speed is another desirable property, especially when handling real-time data or when there is a need to run exploratory analyses on

the fly. In this case, processing speed is not critical because generally Venture Capital (VC) investment decisions are made over weeks and months, though there is some need for the data set to be updated with new information as it becomes available.

2.4.2 Algorithm Characteristics

Supervised machine learning are algorithms that reason about observations to produce general hypotheses that can be used to make predictions about future observations. Supervised machine learning algorithms are diverse, from symbolic (Decision Trees, Random Forests) to statistical (Logistic Regression, Naive Bayes, Support Vector Machines), instance-based (K-Nearest Neighbours), and perceptron-based (Artificial Neural Networks). In Appendix B, we describe each candidate learning algorithm, critique their advantages and disadvantages, and present evidence of their effectiveness in applications relevant to startup investment.

2.4.3 Algorithm Evaluation

We evaluated supervised learning algorithms for their suitability in startup investment prediction. While our evaluation gives us directionality of fit, we hesitate to discard algorithms based on our literature review. Algorithm selection is complex and preliminary testing will provide clarity as to which algorithms should be used. In addition, larger training sets and good feature design tend to outweigh algorithm selection [10]. With those concessions in mind, our findings suggest we expect Random Forests, Support Vector Machines and Artificial Neural Networks to produce the highest classification accuracies. An ensemble of these algorithms may improve accuracy further, though at the cost of computational speed and interpretability. We may expect Random Forests to outperform the other two algorithms due to robustness to missing values and irrelevant features and native handling of discrete and categorical data. However, Random Forests are not highly interpretable so Decision Trees and Logistic Regression may be preferable for exploratory analysis of the data set.

2.5 Research Gap

The Venture Capital (VC) industry requires better systems and processes to efficiently manage labour-intensive tasks like investment screening. Existing ap-

proaches in the literature to predict startup performance have three common limitations: small sample size, a focus on very early stage investment, and incomplete use of features. In addition, there is little evidence that previous research has been translated into systems that are able to assist investors directly. We conducted a literature review to determine how to address these limitations and produce a system that will assist VC firms in originating and screening investment candidates.

Firstly, we reviewed the business problem and developed three criteria that guided the evaluation of our system: efficiency, robustness and predictive power. Secondly, we developed a conceptual framework of predicting startup performance that incorporates determinants of startup potential and signals that influence investment confidence. This framework informs our feature selection. We then assessed potential data sources and found preliminary evidence that suggests that the startup databases CrunchBase and AngelList are promising and likely to provide a comprehensive feature set that can form the basis of our system. Finally, we reviewed supervised machine learning techniques applied to startup investment and other areas of finance. Our analyses suggested that we should expect Random Forests, Support Vector Machines and Artificial Neural Networks to be most suitable for our system.

Based on this literature review, we believed that it was possible to address previous limitations in this domain and produce an investment screening system that is efficient, robust and powerful. In the next chapter, we outlined the process by which we developed that system.

CHAPTER 3

Design

In this chapter, we explain the methodology used to fill the research gap identified in Chapter 2, thereby producing a system that identifies high-potential startup companies that are likely to receive additional funding or a exit in a given forecast window. Figure 3.1 depicts the overall system architecture, which is described in the following chapter. We evaluate the performance of this system in the next chapter.

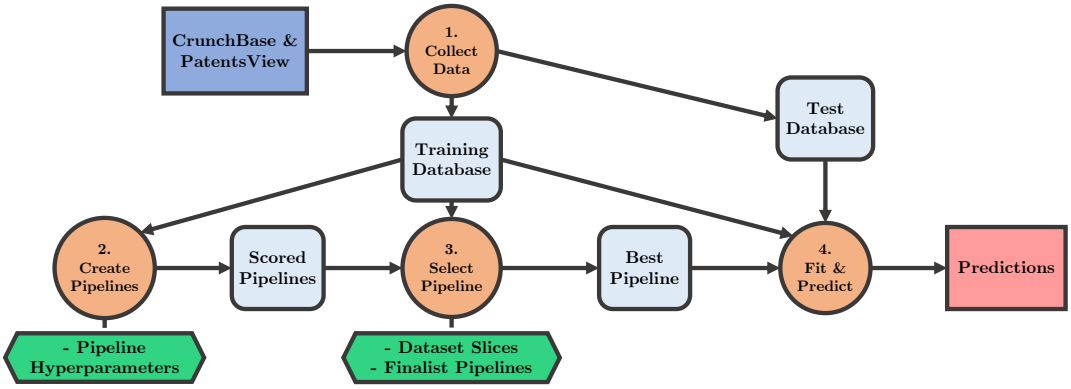


Figure 3.1: An overview of the system architecture proposed by this project, structured in four stages: data collection, pipeline creation, pipeline selection and prediction. Legend: dark blue square = input, yellow circle = process, light blue rounded square = intermediate, red square = output, green hexagon: iterative process / search space.

1. Dataset Preparation. We developed a conceptual framework of startup investment from the literature to guide our feature and data source selection. We selected CrunchBase, a startup database, as our primary data source, which we supplemented with patent filing records from PatentsView. We collected two database dumps from CrunchBase in September 2016 and April 2017, for training and testing respectively. The database dumps were

in the format of CSV files which we imported into a relational database (SQLite) and performed aggregation queries to create features suitable for classification. We then performed screening based on each company's developmental stage and age to ensure only relevant companies were included in the dataset. Finally, we explored the dataset and identified issues of sparsity, long-tailed distributions, imbalanced feature ranges, and non-orthogonality.

2. Pipeline Creation. We developed a processing pipeline framework that seeks to address the dataset issues identified during data collection. Our pipeline is based on the popular Python machine learning library Scikit-learn [36]. Pre-processing steps include imputation, transformation, scaling and feature extraction. Each pre-processing step has hyper-parameters that can be tuned (e.g. imputation strategy, number of components to extract) that affect the pipeline's classification performance. We also tested a number of common classification algorithms and their hyper-parameters, selected from our literature review in Chapter 2. We performed a search across the pipeline's hyper-parameters to generate candidate pipelines. The hyper-parameters that we found to have the most significant effect on the final performance of the pipelines were related to the classification algorithms.

3. Pipeline Selection. Our system evaluates and ranks the candidate pipelines and tests the best pipelines (finalist pipelines) over pseudo-historical datasets. This process ensures we select pipelines that are robust with respect to time. We prepared the dataset slices using a technique that filters records by their timestamps, effectively recreating historical datasets. We use Area under Precision-Recall (PR) Curve as our evaluation metric. We aggregate the results for each finalist pipeline across these dataset slices and rank the finalist pipelines on their overall performance, so we can select the best pipeline for further evaluation. We don't observe significant variance in the pipelines on aggregate against the dataset slices, but there is variance within the individual pipelines. Our results suggest that it is optimal to evaluate the top 3-5 candidate pipelines in this manner.

4. Model Fit and Prediction. Finally, our system applies the best pipeline to a training dataset to produce a fitted model. The model is then applied to a feature vector from a held-out test database, which generates a set of predictions which could, in practice, then be used by Venture Capital (VC) firms. We evaluate the accuracy of the models produced by our system with respect to a number of variables in the next chapter.

3.1 Dataset Preparation

In the previous chapter, we reviewed the literature concerning feature selection and data sources for entrepreneurship and Venture Capital (VC) investment. We identified that a wide range of features have been implicated as significant to startup performance, but concluded that the field is lacking a unified conceptual framework. We evaluated that the most promising primary data sources for this project are CrunchBase and AngelList, for their size, comprehensiveness and ease of access. We suggested PatentsView (the online database of the US Patent Office) could be a useful secondary data source for structural capital features.

In this section, we first discuss how we developed a conceptual framework that guided our feature and data source selection. Next, we describe the process of generating our primary datasets, which involves collecting data from CrunchBase and PatentsView, converting the relational databases into a format suitable for machine learning, and performing screening to ensure we only included relevant companies. This process is depicted in Figure 3.2. Finally, we describe the results of our exploratory analysis and identify dataset issues to be addressed in later steps.

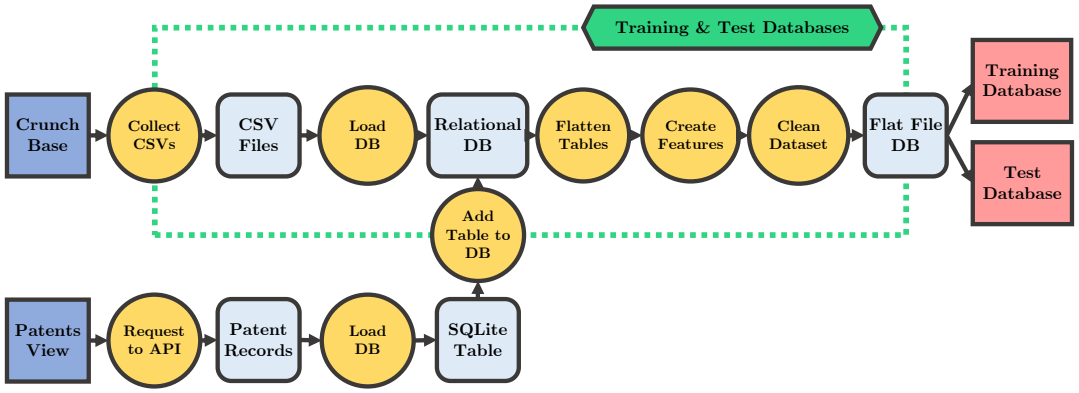


Figure 3.2: Data collection overview. Legend: dark blue square = input, yellow circle = process, light blue rounded square = intermediate, red square = output, green hexagon: iterative process / search space.

3.1.1 Feature Selection

We developed a conceptual framework that builds upon previous work to ensure that we include a comprehensive and relevant set of features in our investment

screening system. This framework builds on previous work by Ahlers and colleagues (2015), designed to model performance on equity crowd-funding platforms [1]. We sought to generalise Ahlers' framework [1] beyond equity crowd-funding. While the first factor of Ahlers' framework (venture quality) applies to startups generally, Ahlers defined their second factor (investment confidence) with respect to whether startups offer an equity share in their crowd-funding, and whether they provide financial projections. These features are specific to equity crowd-funding. We proposed an extension of Ahlers' framework that generalises and develops this second factor. We described investment confidence as a product of third party validation, historical performance and contextual cues. Our proposed framework is depicted in Figure 3.3.

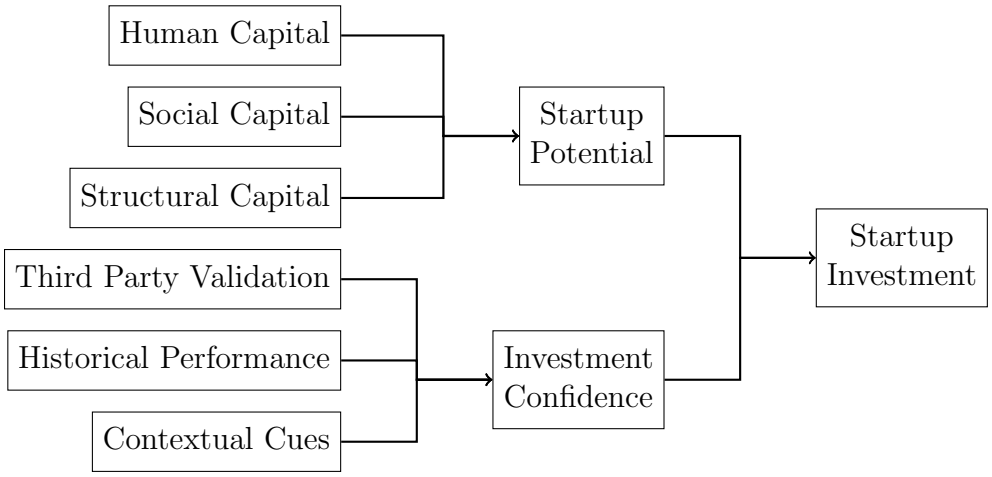


Figure 3.3: Proposed conceptual framework for startup investment. We adapt the framework proposed by Ahlers et al. [1], originally based on work by Baum and Silverman [5]. For an extended version of this framework, please refer to Figure C.1.

Feature selection is critical to the success of our proposed conceptual framework. In this section, we have built on the framework proposed by Ahlers et al. [1] in several ways. First, our framework generalises the "Investment Confidence" factor for startups seeking any type of investment (not just equity crowd-funding). Second, our framework has greater depth. Where Ahlers uses one or two features for each factor in their model (e.g. "% Non-executive board" represents "Social (alliance) capital"), we perform a review of many features employed in this area and perform a higher degree of classification. For example, in our proposed framework "Social (alliance) capital" is composed of "Social media influence", "Events influence" and "Strategic alliances", each of which will also be composed of several features (e.g. "Twitter followers", "Average Tweets per

day").

3.1.2 Data Collection

3.1.2.1 CrunchBase

We were granted an Academic License to collect data from CrunchBase. CrunchBase provides database access in a few formats that offer trade-offs in terms of accessibility and comprehensiveness: REST API, CSV Dumps, MS Excel Summary. We chose to use the CSV Dumps for this implementation of the system because they provided a good trade-off of ease of use and comprehensiveness of access. The Dumps provide a CSV file for each of CrunchBase's key endpoints (e.g. organizations, people, funding rounds) which can be loaded easily into relational databases (see Appendix D for the database schema). We downloaded CSV Dumps from CrunchBase on 09 September 2016 and 04 April 2017 which became our training and testing datasets, respectively.

3.1.2.2 PatentsView

We used PatentsView to obtain the patent filing records of each company in our CrunchBase dataset, focusing on information relating to dates, citations, and patent types. We matched the data sources on standardised company names (removing common suffixes, punctuation etc.) and using normalised Levenshtein distances. Although approximate matching introduces error, the volume of companies in the database is too high to be matched manually and there are no other identifying records. We stored the PatentsView data in a relation which we merged into our master and test databases.

3.1.3 Dataset Manipulation

To prepare the dataset for machine learning, we first flattened the relational database into a single file using SQL aggregation queries. We aggregated each relevant relation in turn, grouping by Company ID and then combined each aggregated table using left outer joins. Following this process, we used Python to convert tuples (e.g. Round Type and Round Date) and lists (e.g. Round Types) into dummy variables.

We performed preliminary screening on the primary dataset (N = 425,934) to ensure it only included relevant companies. We were interested in removing

traditional, non-startup businesses from the dataset (e.g. consulting firms, companies that will not take VC funding etc.). To do this, we explored two factors for each company: developmental stage and age. By developmental stage, we primarily refer to external funding milestones. These stages are associated with shifts in a startup company's functions and objectives and we also expect them to correlate with company age. Our dataset as grouped by startup developmental stage is depicted in Figure 3.4.

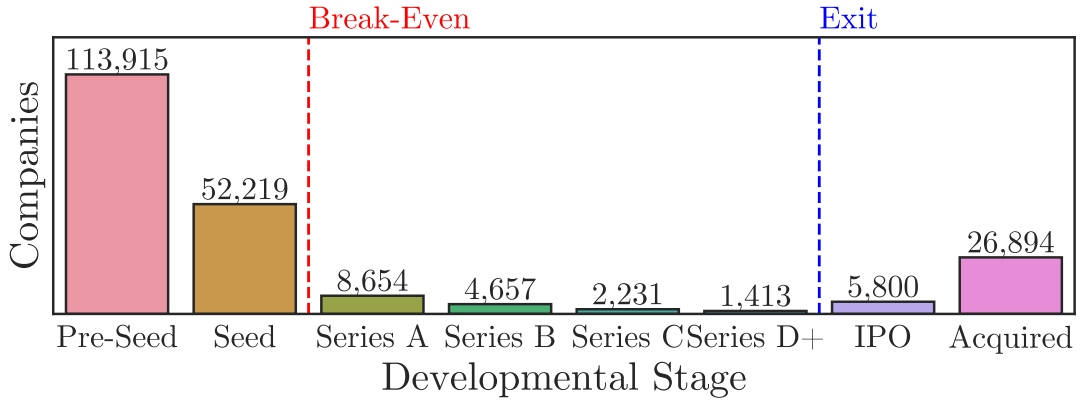


Figure 3.4: Companies grouped by stages of the startup development life-cycle.

After attempting to place the companies into development stages we are left with a large group of companies (the majority of the dataset) that have not raised funding and so can not be classified on that basis. We assume that companies that have not raised funding fall into two groups - those that intend to raise funding but have not had time to yet, and those that have chosen not to pursue funding and are unlikely to do so. We separated these groups by applying a cut-off equal to the 90th percentile of the age of companies in the Seed category, and excluded the older group from further analyses (N = 227,162, 53.3%). As we are only interested in companies that could theoretically seek investment, we also excluded Closed, Acquired and IPO groups from further analyses (N = 35,973, 8.4%).

Figure 3.5 depicts the ages of companies in the master dataset, grouped by developmental stage. While there is significant variability in age for each developmental stage, there is a broad positive relationship between age and developmental stage. Most pre-Series A companies are under five years old, and the majority of Series D+ funded companies are under 10 years old and the 75th percentile is at 15 years old. On this basis, we excluded companies that are over the 75th percentile of the age of companies in the Series D+ category (N =9,756, 2.2%). Overall, our preliminary screening steps reduced the dataset from 425,934

companies to 153,043 companies, a reduction of 64.1%.

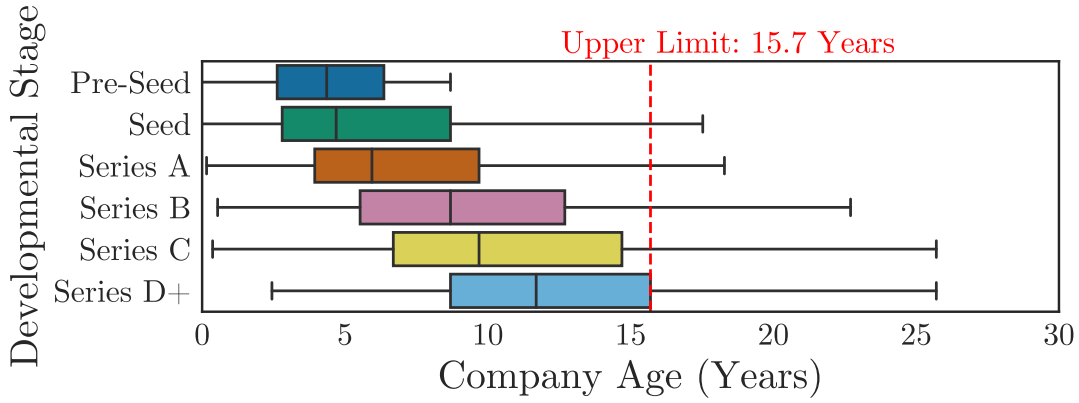


Figure 3.5: Company ages in years grouped by developmental stage. The dashed red line represents the 75th percentile of the age of companies in the Series D+ category (15.7 years).

3.1.4 Exploratory Analysis

3.1.4.1 Descriptive Statistics

Table D.1 presents the descriptive statistics for the dataset. The dataset is heavily skewed towards Pre-Seed companies (i.e. companies that were recently founded and have not raised any type of funding yet, 68.9%). These companies have few available features in comparison to companies at later developmental stages. We will investigate the impact of this sparsity on our predictions in Chapter 4. We are presented with a heterogeneous dataset: the interquartile ranges imply significant variability in all measures. We do not believe that this implies that the data has not been cleaned effectively, but rather, reflects that startup companies vary significantly in their traits.

CrunchBase's approach to industry classification is simplistic compared to other classification schemes (e.g., USSIC, NAICS, VentureSource) which generally have an industry hierarchy with tiers for broad industry sectors and subsectors providing further granularity. As a result, CrunchBase class labels include over represented and vague classes (e.g., "Software", "Internet Services") which could describe the majority of companies included in the database. In fact, "Software" and "Internet Services" account for 16.4% and 13.4% of all companies in the dataset respectively (see Figure 3.6). Despite these vague class labels, it is

Stage	Obs	Age (Years)		Funding Raised (USD, M)		Funding Rounds (N)		Patent Filings (N)		Available Features (N)	
	N	50th	75th	50th	75th	50th	75th	75th	90th	50th	75th
Pre-Seed	113,915	4.36	6.36	0.00	0.00	0	0	0	0	25	133
Seed	38,942	4.66	6.69	0.25	1.30	1	2	0	1	178	231
Series A	6,615	5.69	8.70	4.40	9.41	2	3	0	2	239	302
Series B	3,342	7.61	10.70	14.89	28.20	3	4	0	4	255	314
Series C	1,610	8.70	11.70	35.29	62.00	3	5	1	9	305	321
Series D+	998	9.70	12.70	74.39	130.8	5	7	4	19	319	330
Included	165,422	4.69	6.69	0.00	4.00	1	2	0	1	90	160

Table 3.1: Descriptive statistics grouped by developmental stage.

clear the dataset skews towards high technology startups, as opposed to biomedical, agricultural, or other technologies (which do not make the Top 10).

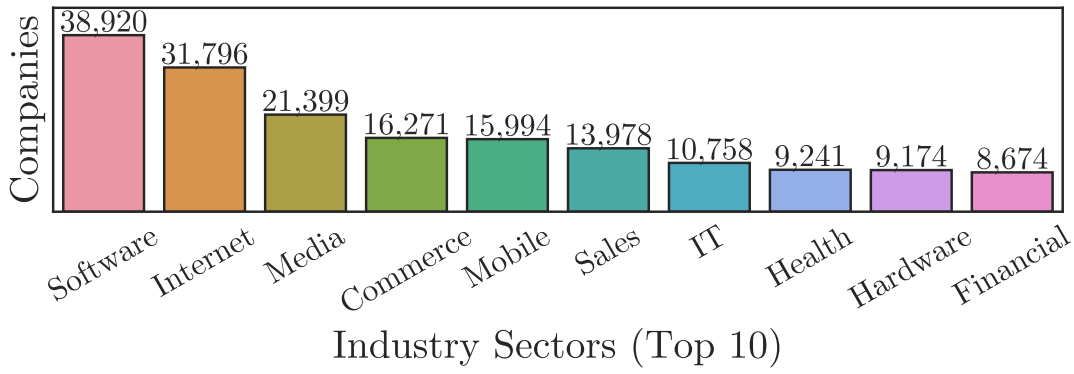
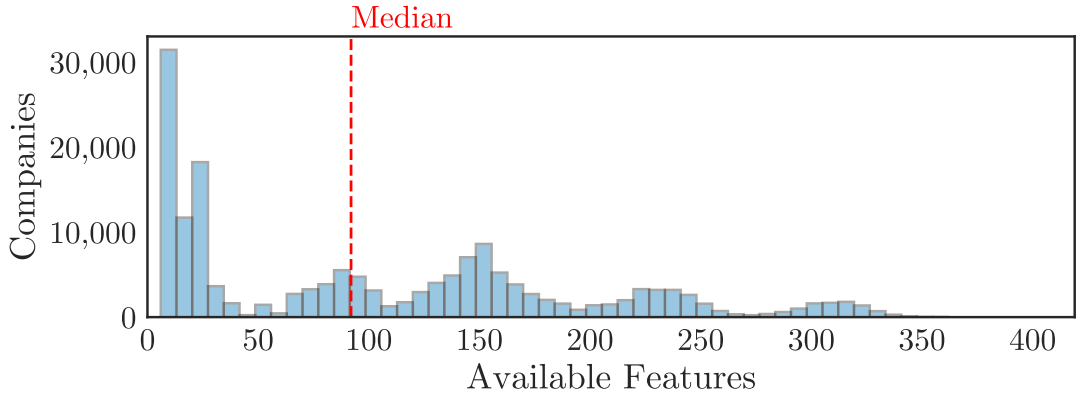


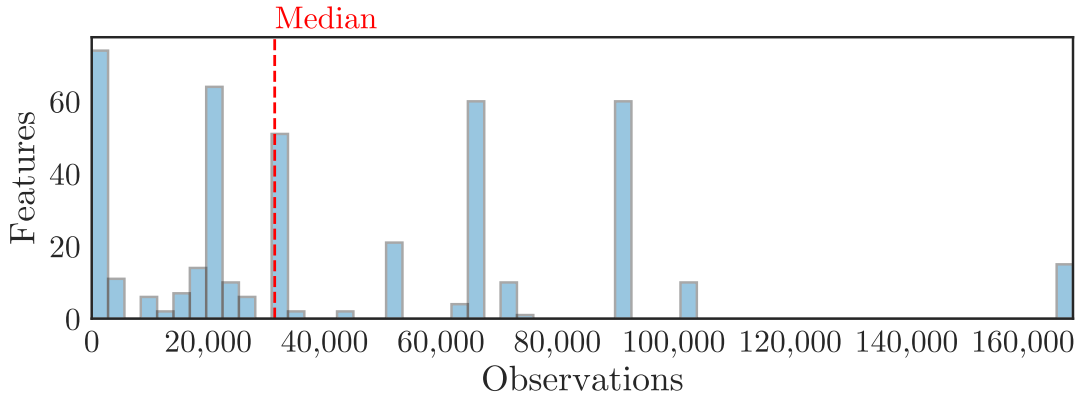
Figure 3.6: Companies grouped by industry sector. The 10 most common sectors are displayed. Source: Master dataset (c. September 2016).

3.1.4.2 Sparsity

First, we explored missing data in the dataset. We expected the dataset to be highly sparse because it primarily came from CrunchBase, a crowd-sourced database. As profiles are entered into CrunchBase piece-meal, it is not clear at face-value whether data (e.g. records of funding rounds) is missing or didn't occur. Figure 3.7 displays the distribution of missing data in the dataset, with respect to each feature and each observation. The multi-modal peaks of both distributions suggest that missing data across certain groups of features may be correlated with each other (e.g. all features derived from funding rounds).



(a) Distribution of missing data by company

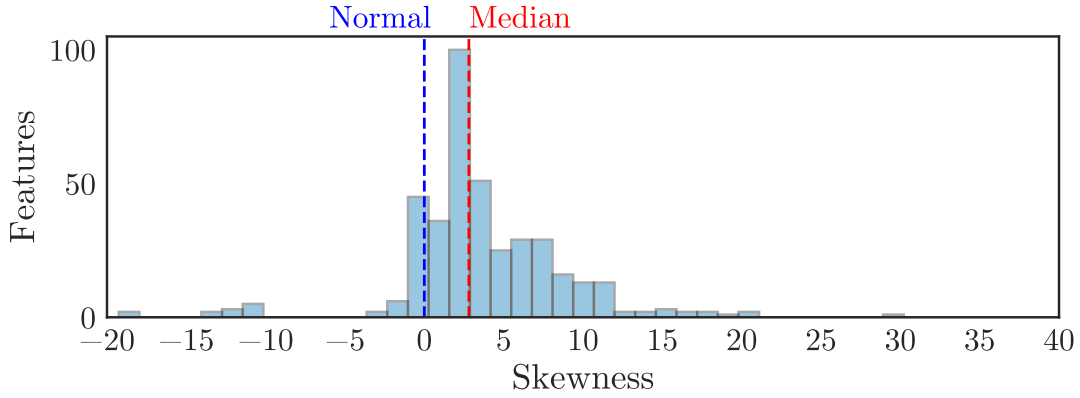


(b) Distribution of missing data by feature

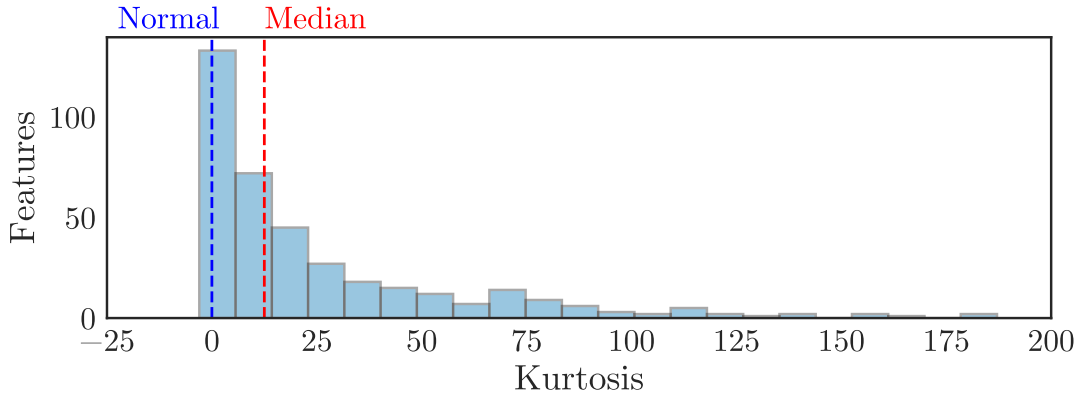
Figure 3.7: Distribution of missing data in master dataset (c. September 2016).

3.1.4.3 Normality

Next, we explored the distributions of features. Figure 3.8 shows the skewness and kurtosis of the features in our dataset. A feature is considered horizontally symmetrical if it has a skewness of 0 and considered highly skewed if its absolute skewness is above 1 [9]. The vast majority of our features are more skewed than this cut-off. Kurtosis is a measure of the distribution of variance in a feature. We use Fisher's measure of kurtosis, which has a normal value of 0. Our dataset has consistently higher kurtosis than normal which suggests that we have many extreme values in our dataset. These results in concert suggest that our dataset has many features that are positively skewed with long-tail distributions. This is intuitively what we might expect for features like "amount of funding raised".



(a) Skewness.



(b) Kurtosis.

Figure 3.8: Distribution of features in master dataset (c. September 2016).

3.1.4.4 Scale

Next, we explored the scaling and range of each of our features. Figure 3.9 shows the Interquartile Range (IQR) of each feature (transformed by log1p for ease of viewing). The distribution is extremely skewed, which shows that our features have a wide range of magnitudes. This may be an issue for our machine learning estimators and feature extractors, so we address this by applying a scaler in our pipeline.

3.1.4.5 Orthogonality

Finally, we explored the orthogonality of our features: the extent to which the variance of our features is unrelated. This is a less straight-forward measure. We

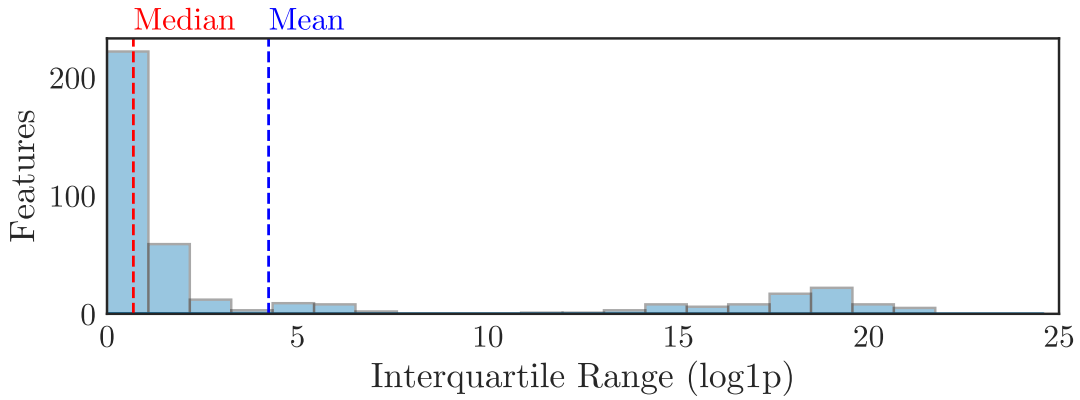


Figure 3.9: Distribution of interquartile ranges (transformed by log1p) in master dataset (c. September 2016).

explore pair-wise inter-correlations between our features and evaluate how many of the inter-correlations are above a particular correlation cut-off, as depicted in Figure 3.10. We use two correlation metrics: Pearson and Spearman. Pearson is more commonly used but Spearman is a ranked metric and may more accurately reflect our non-normal feature distributions. Although most features have relatively low inter-correlations ($\tilde{6}0\%$ below 0.2) there are still a considerable number that are highly correlated, so it might be efficient to remove these features using an unsupervised feature extractor prior to estimation.

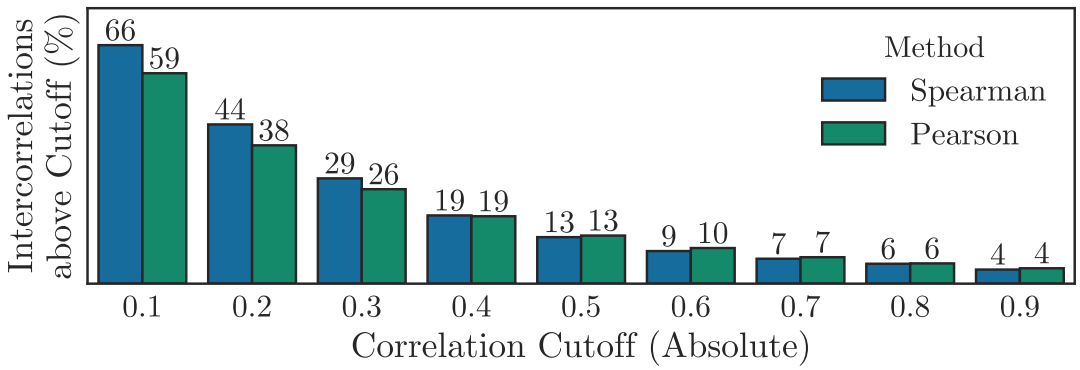


Figure 3.10: Distribution of inter-correlations.

3.2 Pipeline Creation

We developed a classification pipeline using the popular Python-based machine learning library Scikit-learn [36]. The classification pipeline construct allows us to easily search across hyper-parameters at each step in the pipeline (see Appendix E for hyper-parameter list). The following sections explore the testing of each hyper-parameter decision, and the selection of primary classifiers for the following steps. This process is depicted in Figure 3.11.

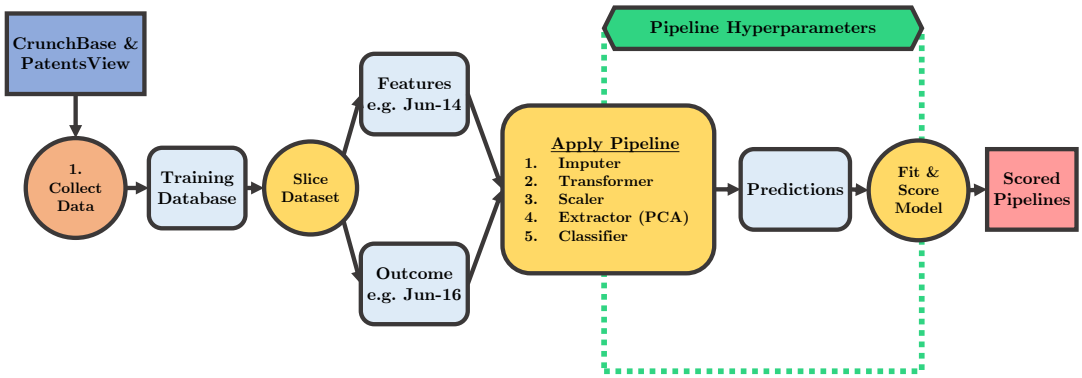


Figure 3.11: Pipeline creation overview. Legend: dark blue square = input, orange circle = system component, yellow circle = process, light blue rounded square = intermediate, red square = output, green hexagon: iterative process / search space.

3.2.1 Imputation

After reviewing the distribution of missing data, we decided to perform further investigation into imputation methods. Common imputation strategies include replacing missing values with the mean, median or mode of each feature. Figure 3.12 shows the distribution of mean, median and modes for each feature in the dataset. For the majority of features, all three measures of central tendency are equal to zero. This resolves the issue of distinguishing missing data from negative observations because, following imputation, all of these data points will map to zero. Figure 3.13 shows the receiver-operating characteristics of the different imputation strategies. As expected, all three imputation strategies produce similar results (within the margin of error).

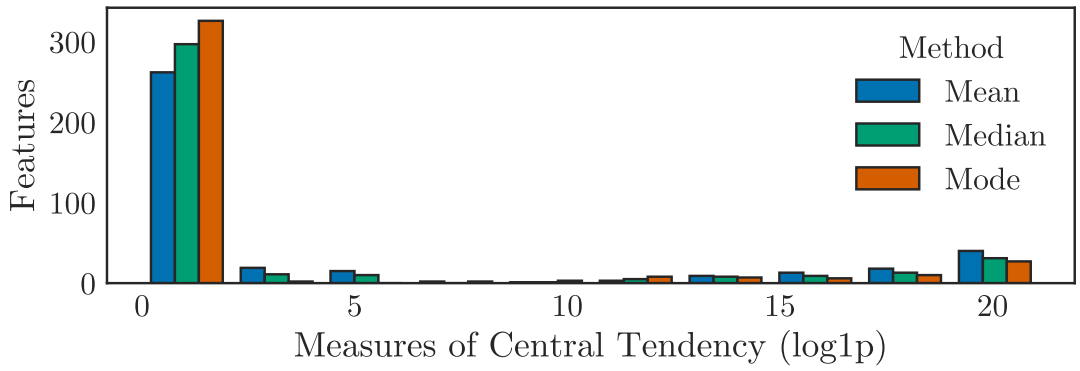


Figure 3.12: Distribution of measures of central tendency (mean, median and mode).

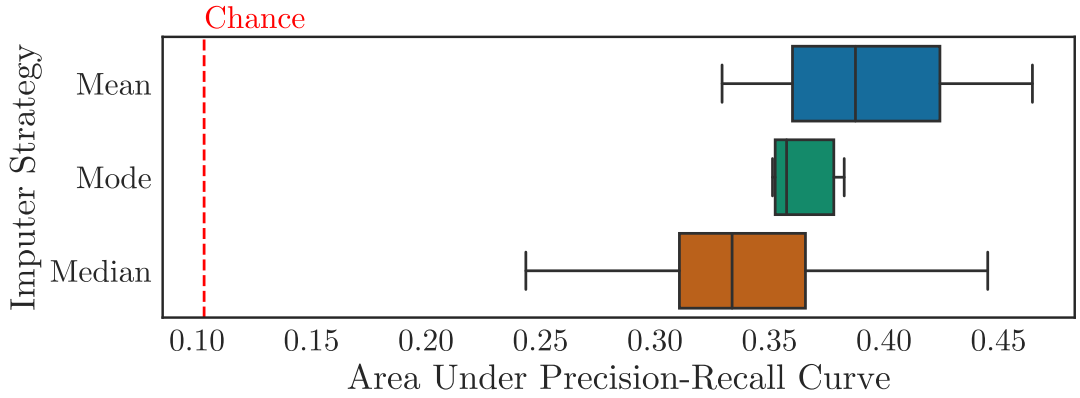


Figure 3.13: Area under Receiver Operating Characteristic (ROC) for different imputation strategies. Imputation strategies include replacing missing values with the most frequent (mode), median and mean value of each respective feature. Results presented are aggregated from hyper-parameter optimisation performed over entire classification pipeline (including all classifiers). Source: Features (Apr-12) and labels (Apr-14, 2 year forecast window) derived from Master dataset (c. Sep-16).

3.2.2 Transformation

While the classification algorithms we identified in the previous chapter are relatively robust to violations of normality, it may be beneficial to transform the data if the feature distributions are extreme. Figure 3.14 shows one of the key features, Total Funding Raised, under a number of different transformations. Like many features in our dataset, the distribution of Total Funding Raised is highly

skewed. The log transformation reduces this skewness (a normal distribution of non-zero values becomes visible) and square root transformation also reduces this skewness (to a lesser extent). The impact of these transformations is reduced by the extent of their zero-inflation. However, it is still reasonable to expect both of these transformations to improve the classification accuracy. Figure 3.15 shows the ROC of these different transformation functions. Both functions provide a small performance improvement, with the square root function narrowly best.

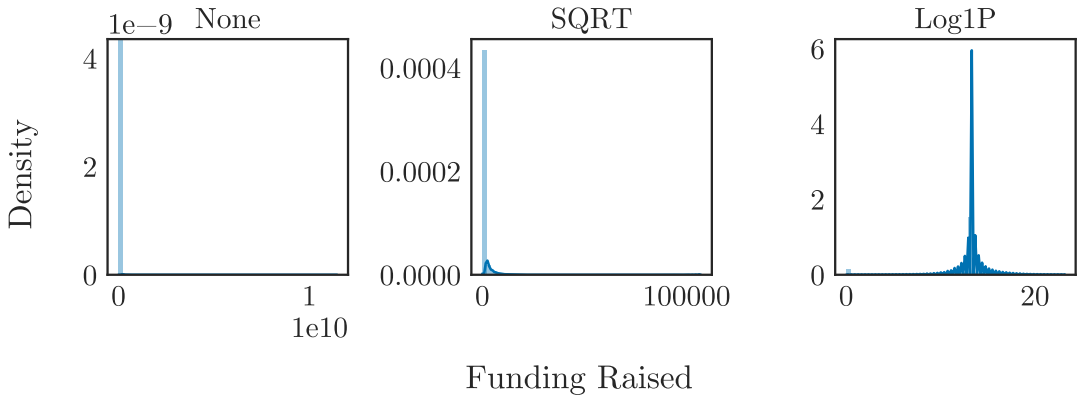


Figure 3.14: Funding raised transformed by functions.

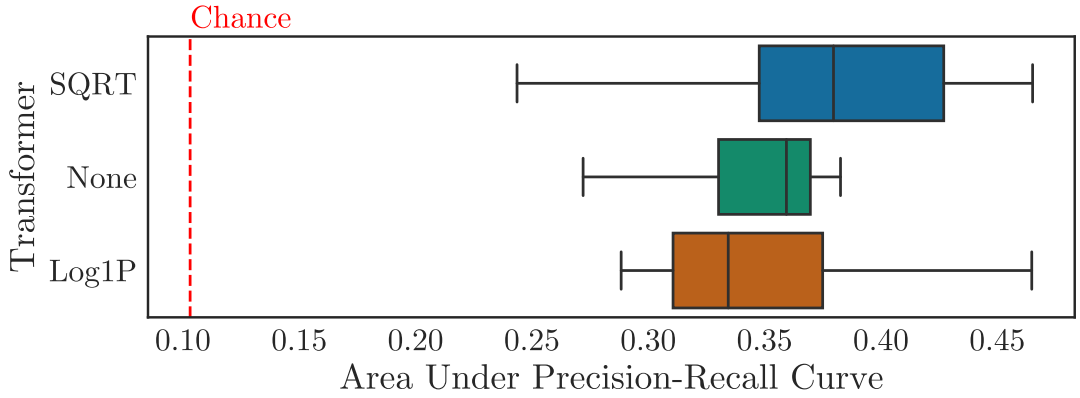


Figure 3.15: Area under ROC for different transformation functions. Transformations include: None (identity transformation), Log1p (natural logarithm of one plus the input array, element-wise), and Sqrt (the square root of the input array, element-wise). Results presented are aggregated from hyper-parameter optimisation performed over entire classification pipeline (including all classifiers).Source: Features (Apr-12) and labels (Apr-14, 2 year forecast window) derived from Master dataset (c. Sep-16).

3.2.3 Scaling

Standardisation of datasets is a common requirement for many feature extraction methods and machine learning estimators. Sci-kit learn provides three primary scaling functions: StandardScaler, RobustScaler and MinMaxScaler. RobustScaler is intended to alleviate the effect of outliers while MinMaxScaler is intended to preserve zero entries in sparse data - both of these are relevant properties for the dataset. Figure 3.16 shows the receiver-operating characteristics of the different scaling functions. MinMaxScaler and RobustScaler actually underperform the null condition while StandardScaler only performs on par with the null condition. This is unexpected but may be caused by the previously applied transformations.

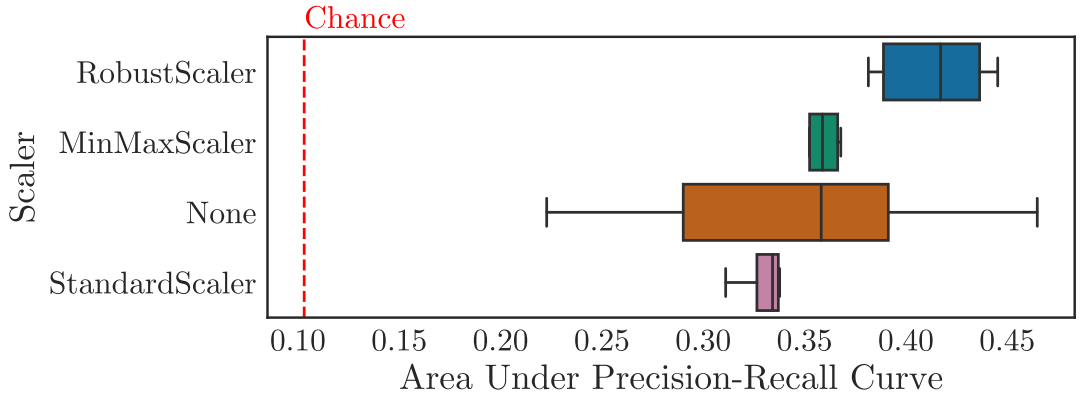


Figure 3.16: Area under ROC for different scaling functions. Scaling functions include: None, StandardScaler (mean: 0, variance: 1), RobustScaler (median: 0, IQR: 1) and MinMaxScaler (min: 0, max: 1). Results presented are aggregated from hyper-parameter optimisation performed over entire classification pipeline (including all classifiers).Source: Features (Apr-12) and labels (Apr-14, 2 year forecast window) derived from Master dataset (c. Sep-16).

3.2.4 Extraction

Feature extraction reduces high-dimensional data into lower-dimensional data in such a way that maximises the variance of the data. The most common approach to dimensionality reduction is Principal Component Analysis (PCA). PCA is a technique which takes a set of vectors and finds an uncorrelated coordinate system in which to represent these vectors [28]. This new co-ordinate system optimally distributes the variance, so that the first basis vector (eigenvector) has the largest

possible variance and all successive eigenvector have the largest possible variance given that they are strictly uncorrelated with the previous eigenvectors. The magnitude of each eigenvector (its eigenvalue) is displayed in Figure 3.17. The majority of explained variance is captured in the first 10 components, and the eigenvalues drop below 1 by 100 components - this suggests that these are reasonable values for further hyper-parameter search. Figure 3.18 shows the ROC for different numbers of extracted components. All curves produce similar classification results (within margin of error) which implies that we should extract between 1 – 20 components because it will provide us with more efficient computation.

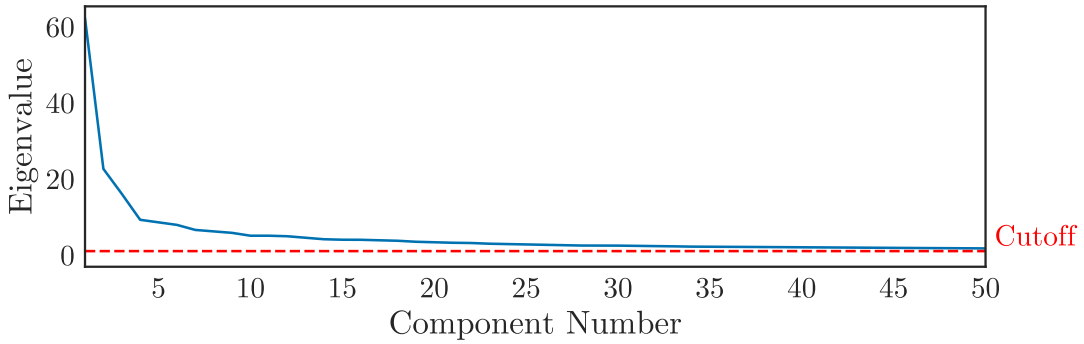


Figure 3.17: Eigenvalues extracted from PCA model. Horizontal line drawn at an Eigenvalue of 1 – this theoretically represents the 'contribution' of one original feature and is commonly used as an approximate threshold for included components. Source: Master dataset (c. Sep-2016).

While PCA is efficient at reducing features, the resultant components are not interpretable. Similarly, individual analysis of 400+ features is difficult to interpret. A compromise is to group the features using the conceptual framework we developed earlier from the literature review. The grouping approach applied weights to each individual feature that optimised the inter-correlations within each group. Given the highly skewed features, we use Spearman correlation which is robust to skewness because it is based on ranking. Figure 3.19 displays the inter-correlations between each factor from the conceptual framework. As we would expect, Investors and Funding features are highly correlated. While Investors captures the influence of previous investors, it also captures features like the size of an investor's past investments, which would likely correlate with the size of the investment they made in the target company. Interestingly, Founders features are positively correlated with all other features except for Advisors features which are negatively correlated with all other feature groups.

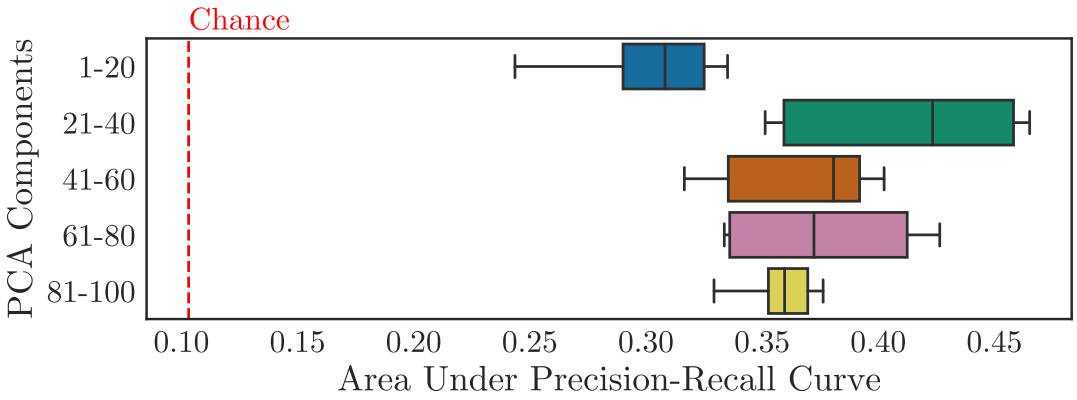


Figure 3.18: Area under ROC for different number of extracted components from PCA. Curves have been grouped by the quotient of the number of components divided by 20 to result in five ordered groups (e.g. Range [0, 19] becomes 0). Results presented are aggregated from hyper-parameter optimisation performed over entire classification pipeline (including all classifiers).Source: Features (Apr-12) and labels (Apr-14, 2 year forecast window) derived from Master dataset (c. Sep-16).

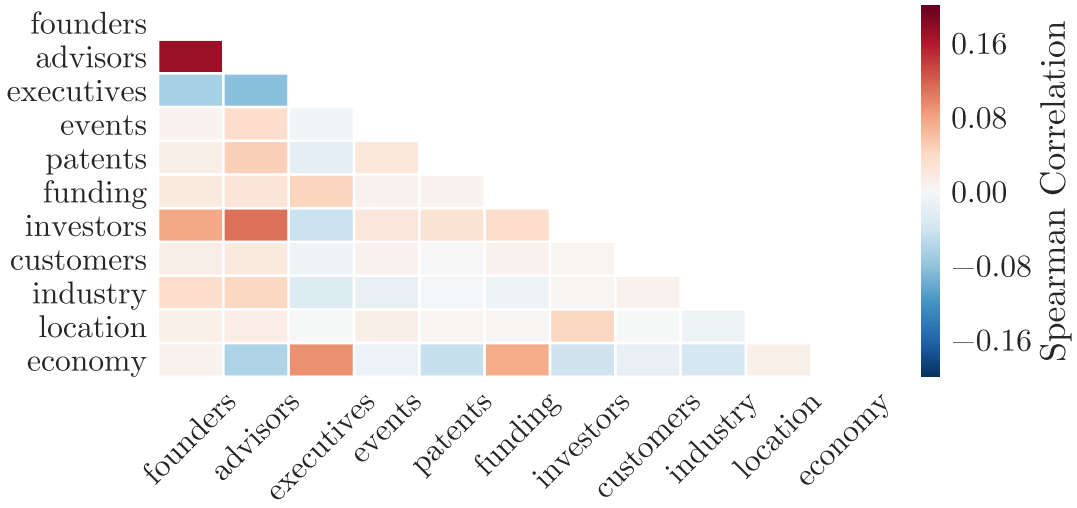


Figure 3.19: Inter-correlations of each factor from conceptual framework. Spearman ranking correlation is used. Individual features are grouped by applying weights that maximise the inter-correlations within each group from our conceptual framework (see Figure 3.3). Source: Master dataset (c. Sep-2016).

3.2.5 Classification Algorithms

The literature review we performed in the previous chapter revealed seven common supervised classification algorithms potentially suitable for application to this problem area. Our review suggested that Random Forests were most likely to provide a successful trade-off between predictive power, interpretability and time taken. We empirically tested each of these classifiers and compared their performance against a range of metrics, as displayed in Table 3.2. We report maximum as well as median recorded scores to ensure we didn't penalise algorithms that had unfavourable hyper-parameter search spaces.

	AUC PRC		AUC ROC		F1		MCC		Fit Time (s)	
Classifier	50th	Max	50th	Max	50th	Max	50th	Max	50th	75th
Logistic Regression	0.417	0.465	0.675	0.710	0.339	0.358	0.255	0.288	7.3	412.7
Random Forest	0.376	0.465	0.619	0.709	0.332	0.360	0.271	0.288	68.3	69.0
Decision Tree	0.388	0.429	0.651	0.659	0.305	0.314	0.212	0.224	15.3	16.8
Naive Bayes	0.354	0.367	0.623	0.638	0.303	0.321	0.212	0.239	8.6	26.8
K-Nearest Neighbors	0.335	0.353	0.532	0.565	0.131	0.226	0.137	0.210	8.5	20.8
Artificial Neural Network	0.320	0.335	0.517	0.523	0.072	0.096	0.111	0.140	9.1	21.0
Support Vector Machine	0.233	0.244	0.503	0.504	0.014	0.017	0.038	0.045	29.0	29.0
Total	0.357	0.465	0.623	0.710	0.300	0.360	0.209	0.288	15.3	29.0

Table 3.2: Overview of classification algorithm performance.

We take a closer look at the Precision-Recall (PR) curves for each classifier in Figure 3.20. While all classifiers perform better than chance, Logistic Regressions and Random Forests come out ahead, and Support Vector Machines and Artificial Neural Networks appear to under-perform. Delving into the cross-validated learning curves for each classifier (Figure 3.21) we see that Naive Bayes, Logistic Regression, Artificial Neural Networks and Support Vector Machines quickly converge, whereas Decision Trees, Random Forests and K-Nearest Neighbours require more observations to converge. This suggests that we might expect Random Forests to do better in final testing (as testing will not be cross-validated), as well as in the future as the dataset naturally grows.

3.3 Pipeline Selection

In the previous chapter, we developed a system that generated a cross-section of candidate pipelines with different hyper-parameters. In this step, we rank these candidate pipelines and evaluate the best pipelines (finalist pipelines) over a number of different dataset slices. This process, depicted in Figure 3.22, ensures

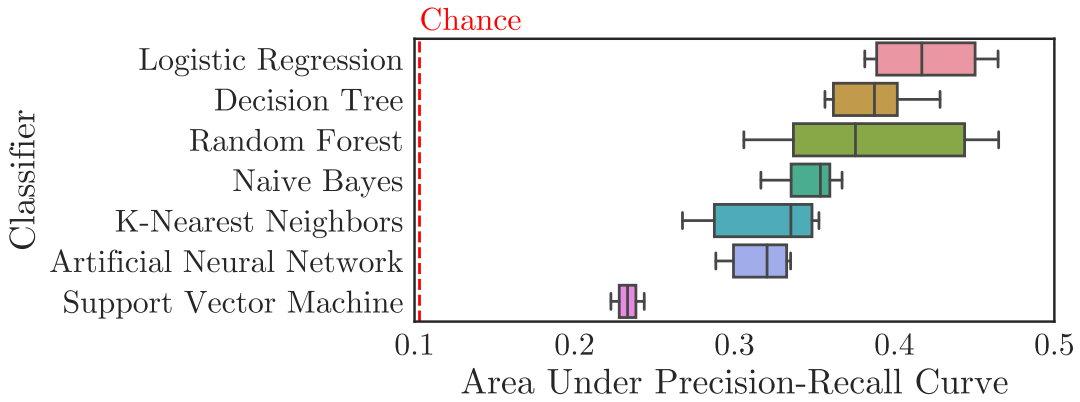


Figure 3.20: Area under ROC for different classification algorithms. All algorithms are implementations from the Sci-kit learn library. Results presented are aggregated from hyper-parameter optimisation performed over entire classification pipeline (including all classifiers).Source: Features (Apr-12) and labels (Apr-14, 2 year forecast window) derived from Master dataset (c. Sep-16).

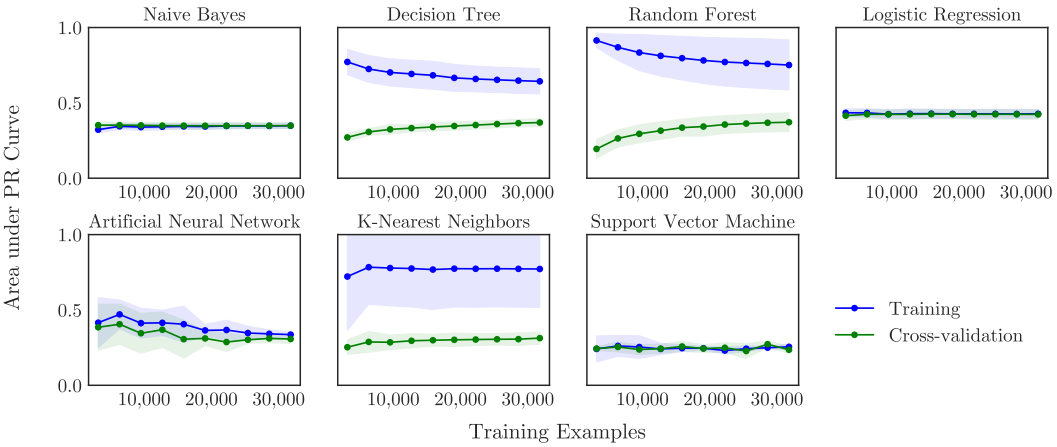


Figure 3.21: Learning curves by classification algorithms.

that our final pipeline is robust in their performance with respect to time. We aggregate the results for each finalist pipeline across these dataset slices and rank the finalist pipelines on their overall performance. Finally, we select the best pipeline.

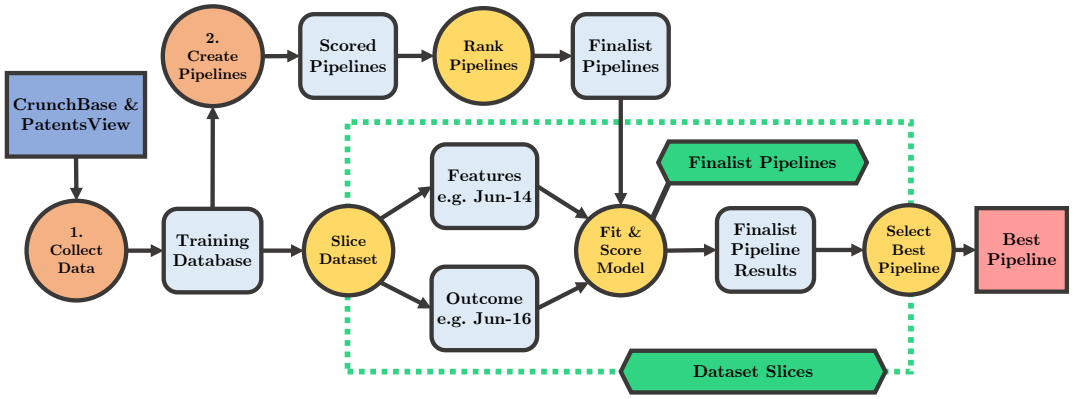


Figure 3.22: Pipeline selection overview. Legend: dark blue square = input, orange circle = system component, yellow circle = process, light blue rounded square = intermediate, red square = output, green hexagon: iterative process / search space.

3.3.1 Dataset Slicing

We developed a procedure for generating historical datasets from our CrunchBase and PatentsView data. CrunchBase provides created and last-updated timestamps for each record in their CSV-formatted dumps (and also in the JSON-formatted responses from their API). We took advantage of this to produce a system that reverse-engineers previous database states by filtering the current database by only records that were created by a given 'slice' date.

We performed preliminary testing of our reverse-engineering technique by comparing a historical CrunchBase database collected in December 2013 with a slice from our primary dataset collected in September 2016, as shown in Figure 3.23. While there are some differences, particularly in the IPO counts, we consider this to be satisfactory variance considering the 3-year time difference (i.e. perhaps some companies have been since removed from the database). The key relations for the purposes of our system are Companies, Funding Rounds and People, all of which had minor differences considering the size of these datasets.

Figure 3.24 shows company counts by startup development stage from different dataset slices. We limited our experiments to dataset slices from 2012-onwards because prior to 2012 the datasets become too small to use to make predictions (particularly given the class imbalance).

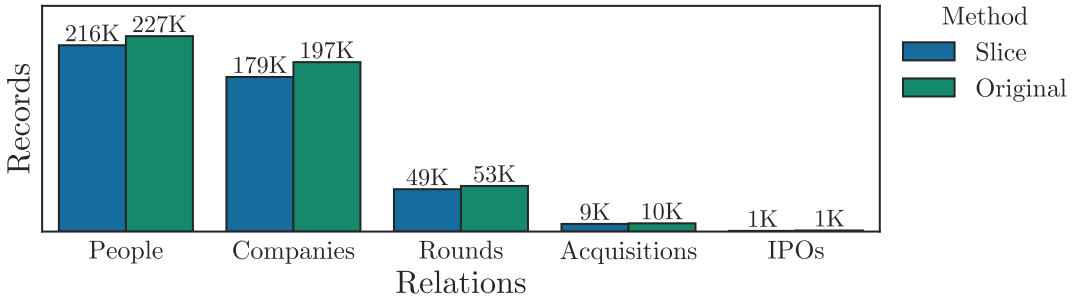


Figure 3.23: Dataset slice compared with original dataset.

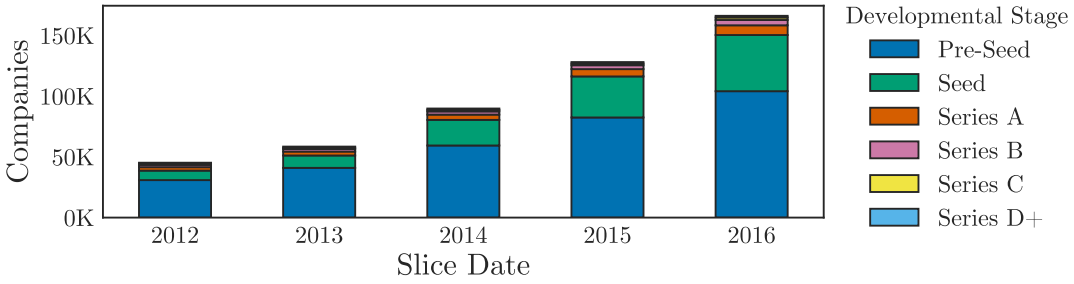


Figure 3.24: Dataset slice counts over time.

3.3.2 Evaluation Metrics

Next, we decided how to narrow our candidate pipelines down to finalist pipelines that we can evaluate further. There are a number of different metrics used to evaluate binary classifiers in machine learning. The most simplistic metric is accuracy but this is rarely used in practice because it gives misleading results in the case of imbalanced classes. Receiver Operating Characteristic (ROC) curves are perhaps the mostly commonly used evaluation tool in binary classification, and show how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples. The area under these curves gives a standardised result across a spectrum of decision thresholds. Precision-Recall (PR) curves are similar to ROC curves but instead map the trade-offs between precision and recall. They are less commonly used than ROC curves but have been shown to produce more accurate results for imbalanced classes than ROC curves [15]. Given our dataset is highly imbalanced (the positive class is approximately 10%) we decided to proceed with PR curves. We will also use this metric to determine which is ultimately the best of our finalist pipelines.

3.3.3 Finalist Pipeline Evaluation

Our hypothesis is that the performance of our classification pipelines may vary with respect to the date that the dataset was collected (in this case, sliced). To study this hypothesis, first we explored variance between the pipelines on aggregate against the slice dates, presented in Figure 3.25. We see little variance on this basis, and we don't observe a relationship between slice date and score.

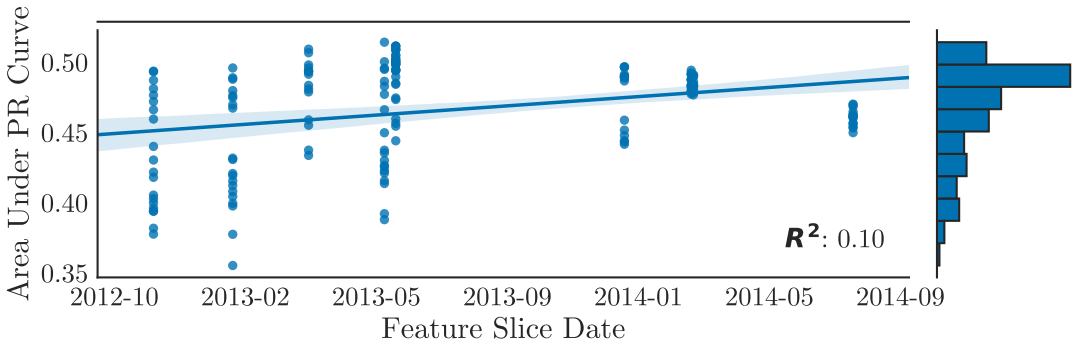


Figure 3.25: Pipeline performance by slice date.

Next, we study the variance within the individual pipelines, presented in Figure 3.26. Here, we can see there is significantly more variance in the scores. Although there is still a strong positive correlation between the pipelines initial ranking and their scores, we can see that there are some individual deviations. Importantly, the top-ranked pipeline from the first stage actually has a lower median score than the second-ranked pipeline. These results suggest that the top 3-5 pipelines should be evaluated in this manner to ensure that the best pipeline is selected. The candidate pipeline selected by this process is depicted in Table F. We adopted this pipeline configuration for our following experiments.

3.4 Model Fit and Prediction

Finally, we use our optimised pipeline to estimate a model and make predictions, as shown in Figure 3.27. Our system applies the best pipeline that was generated in the previous section to a training dataset, producing a fitted model. The model is then applied to a feature vector from a held-out test database, which generates a set of predictions which could, in practice, then be used by Venture Capital (VC) firms. We evaluate the accuracy of the models produced by our system with respect to a number of variables in the next chapter.

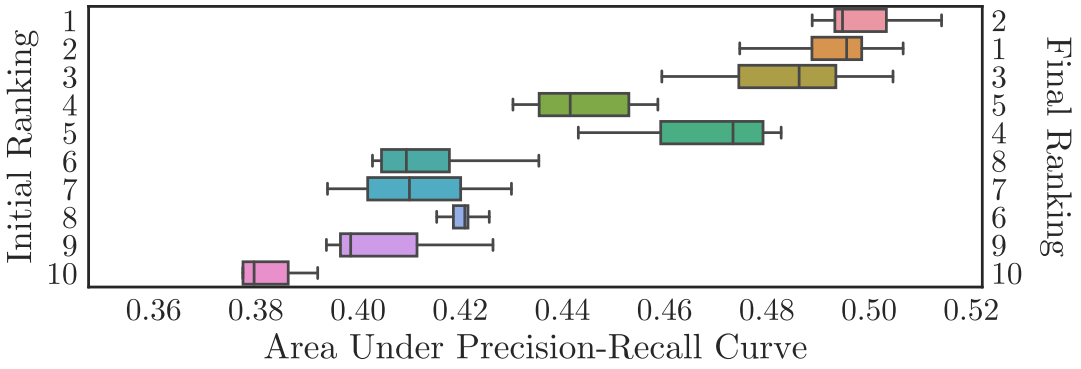


Figure 3.26: Overview of finalist pipeline performance.

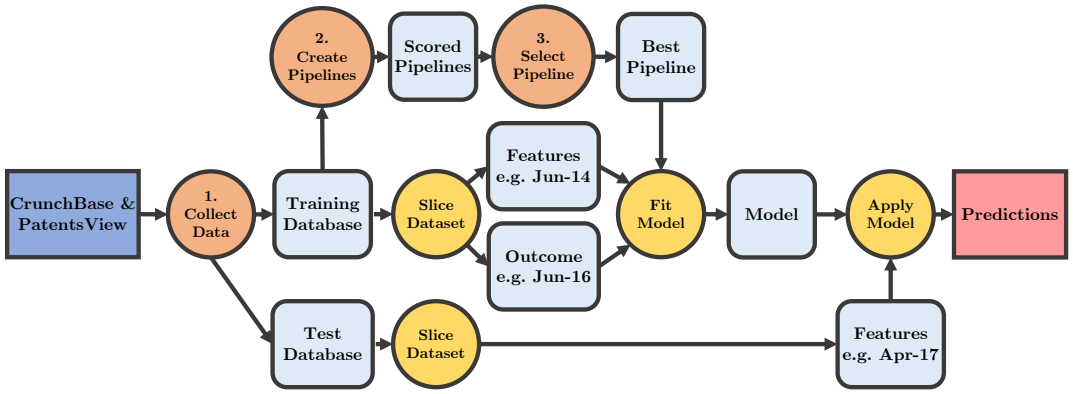


Figure 3.27: Model fit and prediction overview. Legend: dark blue square = input, orange circle = system component, yellow circle = process, light blue rounded square = intermediate, red square = output.

CHAPTER 4

Evaluation

We believe it is possible to produce a Venture Capital (VC) investment screening system that is efficient, robust and powerful. In Chapter 3, we described the development and structure of such a system. Our system identifies startup companies likely to receive additional funding or exit in a given forecast window. This system generates statistics and make recommendations that may assist VC firms to efficiently and effectively screen investment candidates. In this chapter, we first outline our experimental design, and then we evaluate models developed by our system against criteria of efficiency, robustness and predictive power.

1. Efficiency. We evaluated efficiency by exploring the learning curves of our classification techniques and whether there is sufficient data to produce reliable statistics. In some cases, our system could use smaller training sets without significant reduction in predictive power. We also explored the time profile of our system. An indicative implementation of our system takes 46 hours to run.

2. Robustness. We evaluated robustness by evaluating our models against multiple reverse-engineered historical datasets and measuring their variance. We found variance evaluated across all metrics to be low, with more variance over shorter forecast windows. When we explored the feature weights for each model developed on different historical datasets, we found slight variance.

3. Predictive Power. We evaluated our system's predictive power across different forecast windows, for startups at different stages of their development lifecycle, and for different potential target outcomes. We find that our system's performance is positively related to longer forecast window (for 2-4 years), later developmental stage (e.g. Seed, Series A), and breadth of target outcome (e.g. Exit, Acquisition).

4.1 Experimental Design

In the previous chapter, we produced a classification pipeline optimised with respect to its robustness over time. In our experimentation, we evaluated models produced by this pipeline against a held-out test dataset while varying a number of other factors. This evaluation process is depicted in Figure 4.1. The pipeline is fit to a training dataset. The model is applied to a test feature vector to produce predictions. We score these predictions against truth values derived from the held-out test database (collected in April 2017). This process is performed multiple times to evaluate the three primary criteria derived from our literature review: efficiency, robustness and predictive power. The configuration of the system during our experiments is detailed in Appendix F.

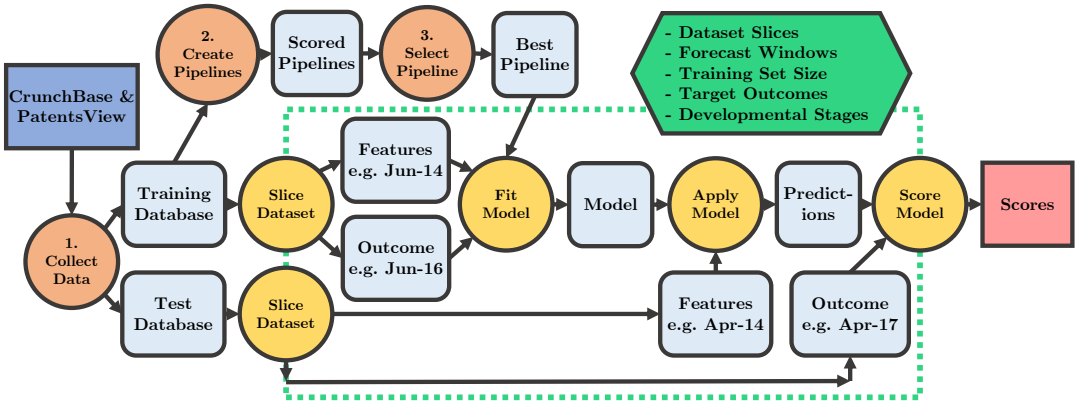


Figure 4.1: Pipeline evaluation overview. Legend: dark blue square = input, orange circle = system component, yellow circle = process, light blue rounded square = intermediate, red square = output, green hexagon: iterative process / search space.

4.1.1 Baseline Analysis

Before we evaluated our system, we performed preliminary analyses to determine the baseline trends and distributions of company outcomes in our database.

First, we looked at company outcomes by forecast window. We applied the same system of reverse-engineering time slices that we used in previous experiments on robustness, but this time we varied the time difference between the slice that provides our features and the slice that provides our outcome. We combined pair-wise datasets of each year from 2012-2016 inclusive and explored the proportion of companies that raised additional funding or exited.

Figure 4.2 shows how company outcome varies with respect to the forecast window (time between the observed features and the measured outcome). We observe a positive relationship between length of forecast window and company outcome. Few companies appear to have exited or raised funds over a period of less than 2 years so we will focus our experimentation on forecast windows of 2-4 years.

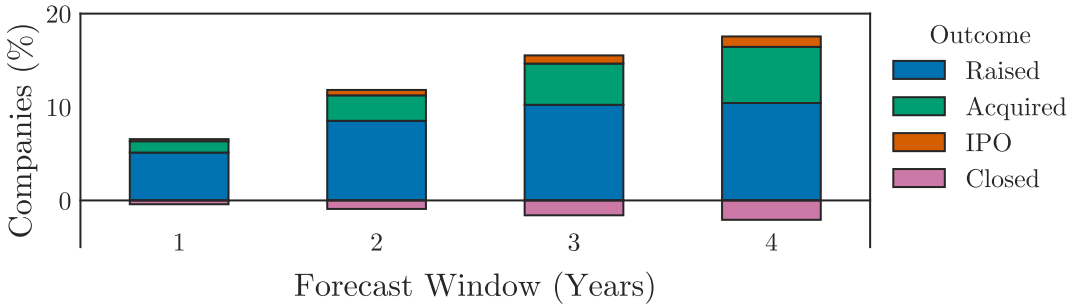


Figure 4.2: Outcomes by forecast window.

We also looked at how company outcomes vary with respect to development stage, shown in Figure 4.3. We see a broad positive relationship between developmental stage and likelihood of further funding rounds and exits, which we would expect as at each stage there is higher market traction and scrutiny from investors. The variance between the outcomes of different developmental stages suggested that in our experimentation we should investigate how our system predicts each stage independently, as well as in aggregate.

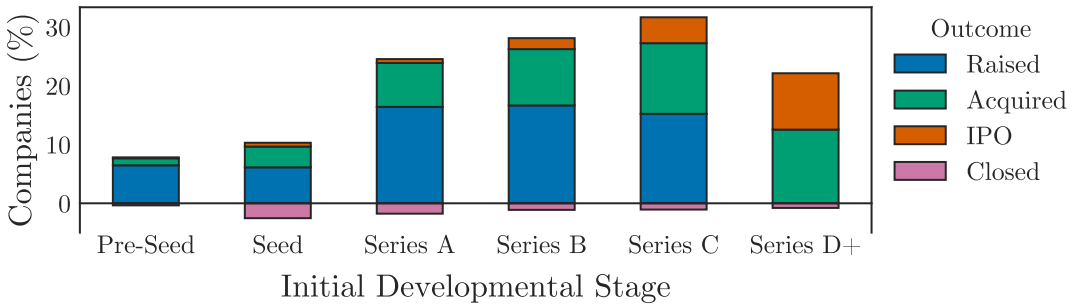


Figure 4.3: Outcomes by developmental stage.

4.1.2 Evaluation Metrics

While Area under the Precision-Recall (PR) Curve was used to guide the development of our system during pipeline optimisation, in evaluation of our system's

performance we primarily use F1 Scores. An F1 score is the harmonic mean of recall and precision at points on the PR curve. In this sense, the Area Under Curve (AUC) measure provides an overall evaluation of a classification system, whereas the F1 Score evaluates a set of predictions. For investment screening, we're more sensitive to classification performance for the positive class (companies that have been successful in raising further funding or achieving an exit), so thereafter, when we refer to F1 Score, we refer to the F1 Score for this class alone. We also present Matthews Correlation Coefficient (MCC) in some of our analyses. MCC is a measure of the correlation between the observed and predicted binary classifications. It should produce similar results to a macro-averaged F1 Score, incorporating the performance of both classes.

4.2 Efficiency

The Venture Capital (VC) industry requires more efficient forms of investment analysis, particularly in surfacing and screening. These processes are currently performed through referral, Google search, industry papers and manual search of startup databases. By its nature, our automated system should be more efficient than these methods. In this section, we assess how efficient our system is – in terms of data consumed and time taken – and look at whether we can further improve its efficiency.

4.2.1 Dataset Size

Learning curves allow us to evaluate how the bias and variance of a classification technique varies with respect to the amount of training data available. We investigated learning curves for our classification pipeline to determine whether smaller samples could achieve similar predictive power and reduce the system's computational demand. We applied 10-fold stratified cross-validation to split our dataset into 10 subsets of different sizes which we used to train the estimator and produce training and test scores for each subset size. The rate of convergence of our training and cross-validation curves implies whether our classification pipeline is over- or under-fitting our data for various sizes allowing us to select an optimal sample size.

Figure 4.4 shows the learning curves for forecast windows of 2-4 years. The maximum number of training examples is negatively related to the length of the forecast window because newer datasets have more examples. For a forecast window of 4 years the curves have converged, whereas for shorter forecast windows

there still seems to be some benefit to additional training examples. Much of the testing score improvement comes in the first 20,000 training examples, which suggests this pipeline is approaching optimal performance.

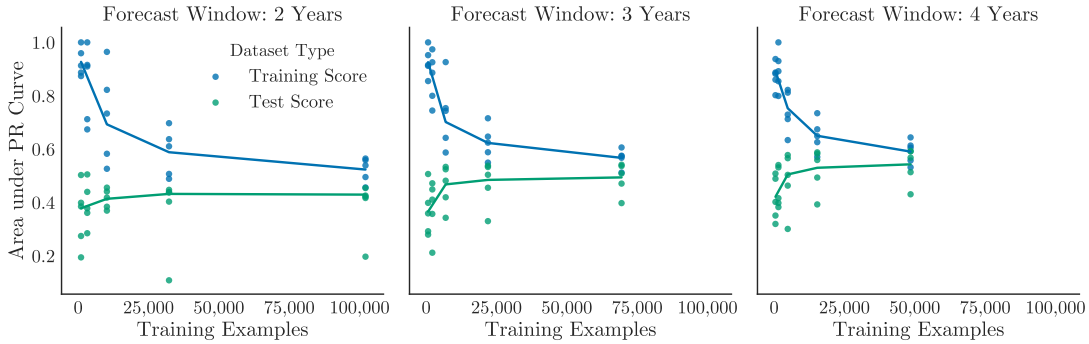


Figure 4.4: Learning curves by forecast window.

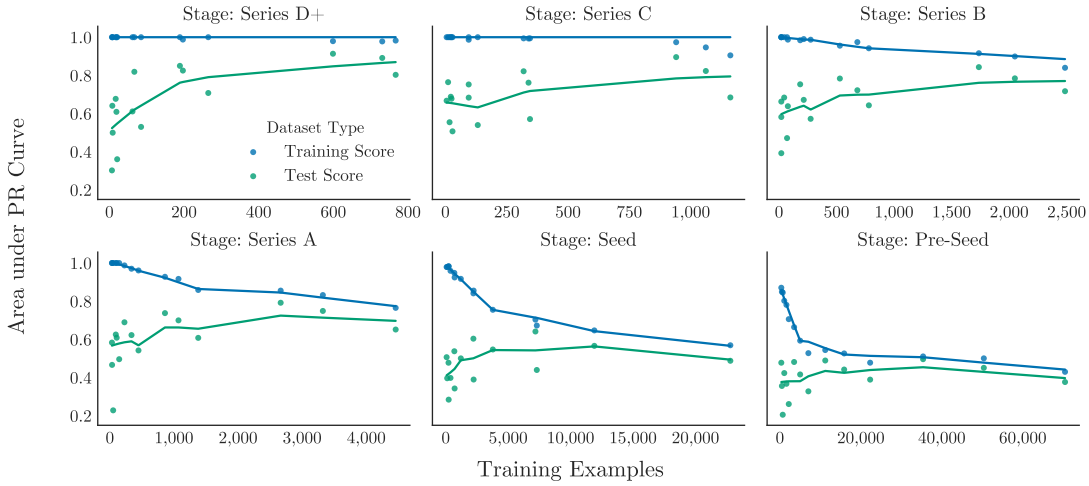


Figure 4.5: Learning curves by developmental stage.

The plots in Figure 4.4 are evaluated against our base target outcome, which we term "Extra Stage" (i.e. whether a company raises an additional funding round, is aquired or has an Initial Public Offering (IPO)). When our learning curves are split by components of this target outcome, we see that the efficiency of our system varies, as shown in Figure 4.6. We observe that predicting whether a company raises an extra round is the least data-intensive outcome, as it converges even over a forecast window of 2 years. In comparison, predicting company exits does not converge, even over a forecast window of 4 years. Our model has most difficulty predicting IPO exits, which are rare events in our dataset.

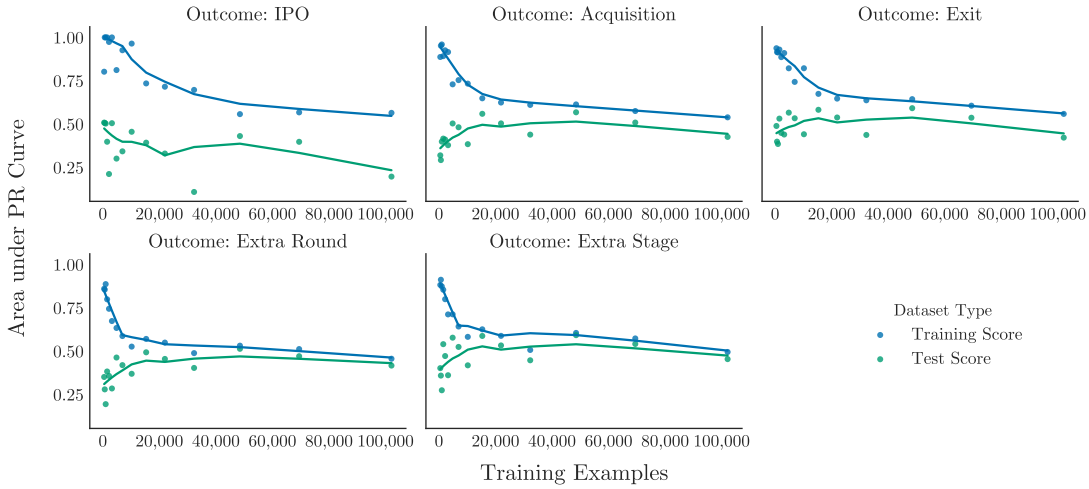


Figure 4.6: Learning curves by target outcome (column) and forecast window (row).

4.2.2 Time Profile

Unlike other forms of finance, like equity or derivatives trading, VC operates on a much longer timeframe – deals close over weeks, rather than minutes. This has two key disadvantages: VC firms have higher management costs because they spend more time screening investments and startup founders waste precious time negotiating with investors when they could be building their businesses. Automated systems could decrease the time taken to generate investment opportunities. We investigated the time profile of our system to determine whether it is practical for use in the VC industry.

An indicative time profile of the system is shown in Table 4.1. At the highest-level, this configuration of the program takes 46 hours to complete on a modern desktop PC. When we further break this time down by system component, the vast majority of time (84.8%) is taken up by the initial pipeline creation component. This time is due to the pipeline optimisation process - the model is fit and scored over 500 times on different classification algorithms and parameters. Scoring takes a long time because, in this case, it also involves generating learning curves for reporting, which is another cross-validated process.

Function	Cycle (s)	Cycles (N)	Time (s)	Time (m)	Time (h)
Generate Dataset (CV)	1,800	1	1,800	30	0.5
Prepare Feature Dataset	1,200	1	1,200	20	0.3
Prepare Outcome Dataset	180	1	180	3	0.1
Merge Datasets	360	1	360	6	0.1
Finalise Dataset	60	1	60	1	0.0
Fit and Score Model[1]	265	525	139,125	2,319	38.6
Fit Model	15	525	7,875	131	2.2
Score Model	250	525	131,250	2,188	36.5
Subtotal: Create Pipelines			140,925	2,349	39.1
Get Finalist Pipelines	5	1	5	0	0.0
Generate Dataset (CV)	1,800	5	1,800	30	0.5
Fit and Score Model[2]	265	75	19,875	331	5.5
Select Best Pipeline	5	1	5	0	0.0
Subtotal: Select Best Pipeline			21,685	361	6.0
Generate Dataset (Training)	1,800	1	1,800	30	0.5
Generate Dataset (Test)	1,800	1	1,800	30	0.5
Fit Model	30	1	30	1	0.0
Make Predictions	5	1	5	0	0.0
Subtotal: Fit and Make Predictions			3,635	61	1.0
Total			166,245	2,771	46.2

Table 4.1: System time profile.

4.3 Robustness

The Venture Capital (VC) industry is concerned that predictive models trained on historical data will not predict future trends and activity. This has been identified as a key barrier to the adoption of automated systems by the VC industry [48]. Therefore, it is critical that our system is shown to be robust in its performance with respect to time so investors can rely on its predictions.

We generated three models from datasets created from our training database from each year of 2012-2014 for forecast windows of 2 years (i.e. [2012, 2014], [2013, 2015], and [2014, 2016]) and evaluated each model against a dataset created from our test database (i.e. [2015, 2017]). We expected that if the factors that predict startup investment success through time are consistent, we would observe little difference between the performance and characteristics of these models.

Figure 4.7 shows the standard deviations of models trained on dataset slices from different years, against key evaluation metrics. We grouped by forecast

windows as later dataset slices cannot be tested with long forecast windows which would skew results along this dimension. Variance across metrics is low, with more variance over shorter forecast windows.

We explored the feature weights for each model in Figure 4.8. While there are some slight differences, the general trend is similar across all models. We will discuss the distribution of these feature weights in more detail in a following section.

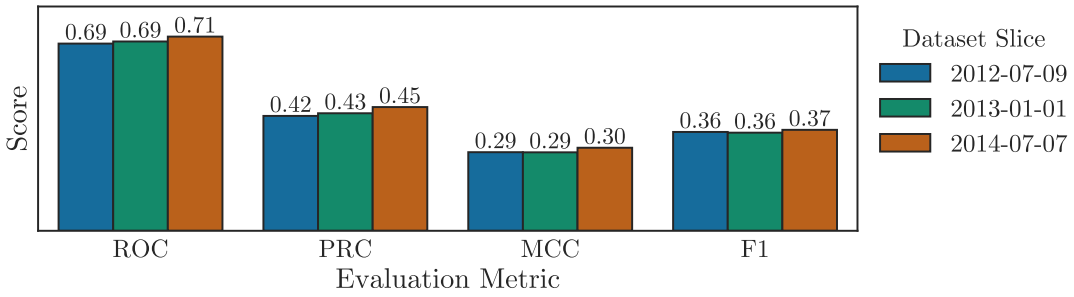


Figure 4.7: Performance variation by slice date.

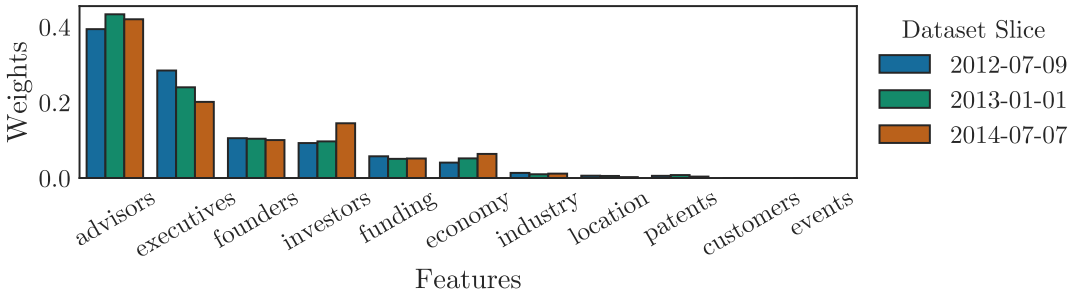


Figure 4.8: Feature weight variation by slice date.

4.4 Predictive Power

The system must be consistently accurate at identifying a variety of high-potential investment candidates. We evaluated the systems' predictive power based on its ability to predict over different forecast windows (e.g. 2-4 years), for target companies at different developmental stages (e.g. Seed, Series A etc.), and for different target outcomes (e.g. predicting additional funding rounds, being acquired, having an Initial Public Offering (IPO), or some combination thereof).

4.4.1 Forecast Windows

A forecast window is the period of time between when a prediction is made and when that prediction is evaluated (i.e. a prediction made in 2014 on whether a company would exit by 2017 is a forecast window of 3 years.) The Venture Capital (VC) industry raises funds with fixed investment horizons (3–8 years), so time to payback is a key component of VC investment decision-making and portfolio management. It is important we understand how the models and predictions produced by a VC investment screening system varies with respect to the length of these forecast windows.

Figure 4.9 shows model performance across a range of metrics, grouped by forecast window. We observe little difference in Area under the Receiver Operating Characteristic (ROC) curve across the forecast windows. However, across all three other metrics, there is a positive relationship between length of forecast window and model performance. The F1 Score shows the greatest improvement in performance over time (52.7%), compared to Area under the Precision-Recall (PR) curve (34.1%) and Matthews Correlation Coefficient (MCC) (11.6%).

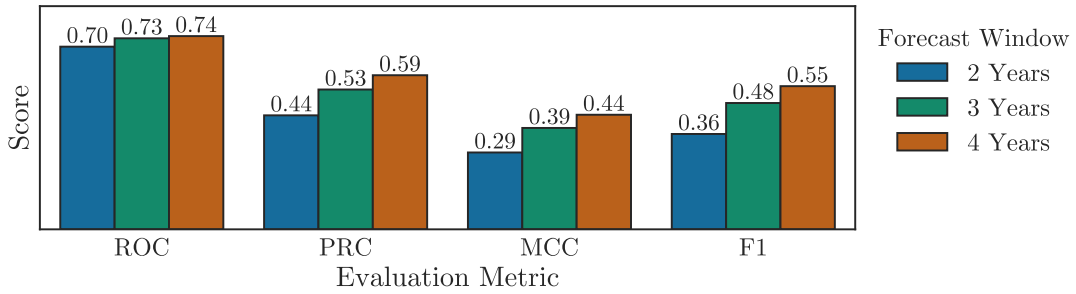


Figure 4.9: Performance by forecast window.

Figure 4.10 shows the standardised weights of features grouped using the conceptual framework proposed earlier in this paper, grouped by forecast window. First, we discuss the baseline distribution and then examine the variation in weightings with respect to forecast window. Advisors are the best predictor of startup investment success. Executives and founders are also important factors, and round out measures of human capital. The quality of investors that invest in a startup (assessed by their prior investments) is found to be more important than the quantum of investment raised by a startup. Local economy and industry factors are weak predictors, as are customers and social influence (in this case measured through participation at events). There is little difference between the weightings of each feature group with respect to forecast window. However, there are a few trends to point out: the importance of advisors increases over time,

and the importance of executives and the broader economy decreases over time.

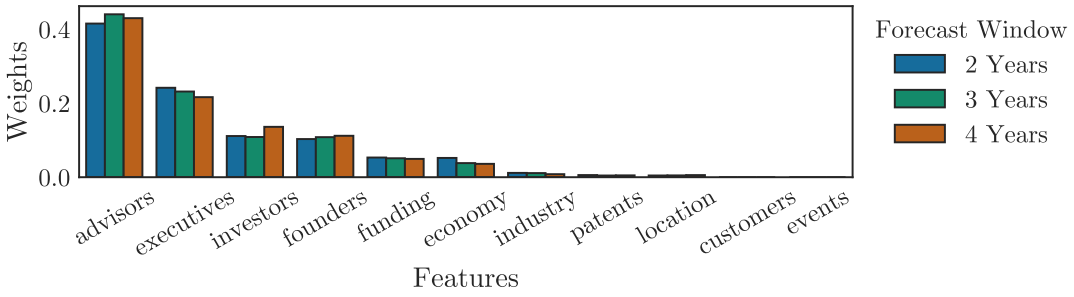


Figure 4.10: Feature weights by forecast window.

4.4.2 Development Stage

Startups can be classified into developmental stages by virtue of their external funding milestones. These milestones not only signal a change in the resources available to a startup, but also their functions and objectives, and in turn the type of investors that are interested in them as investment opportunities. In Chapter 3 we mapped the companies in our dataset to their developmental stages. In the following section, we evaluated how the system models and predicts the outcomes of companies at different developmental stages.

Figure 4.11 shows F1 Scores grouped by developmental stage and fit method. First, we exmaine the baseline distribution and then the variation in performance by fit method. Model performance has a positive relationship with developmental stage. The only deviation from this relationship is for Series D+. To understand this discrepancy better, we split the datasets into their developmental stages and fit the model onto each of these sub-datasets individually. This results in a broad performance improvement. This method has the least impact on Pre-Seed and the greatest impact on Series D+ companies.

Figure 4.12 shows the standardised weights of features, grouped by developmental stage. While a similar trend to Figure 4.10 is observed, there is more varation in weights than was observed when grouped by forecast window. Advisors are more important to earlier stage companies than late stage companies, investor track record and reputation becomes important as companies approach an exit (Series D+), executive and founder experience are important in pre-seed companies, as is broader economic outlook.

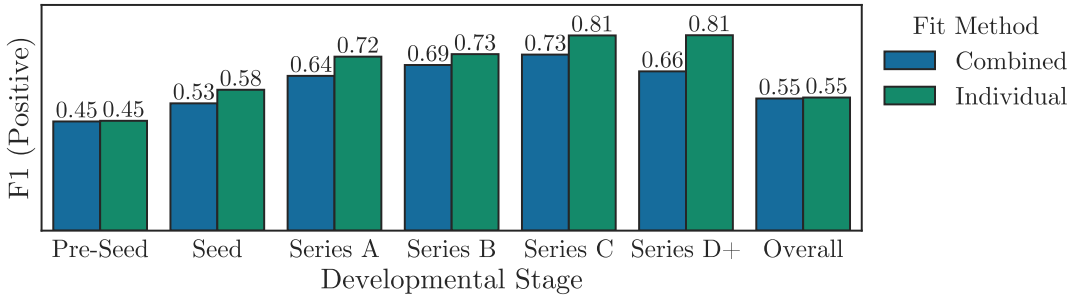


Figure 4.11: Performance by developmental stage.

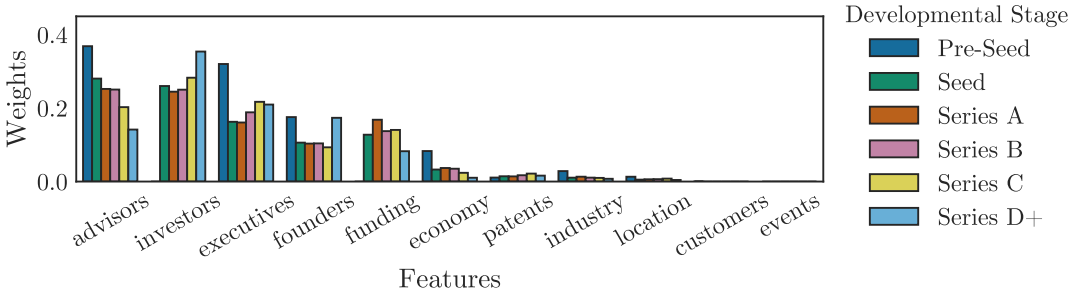


Figure 4.12: Feature weights by developmental stage.

4.4.3 Target Outcomes

Ultimately, VC firms seek rare investments that will return their invested funds many times over within an investment horizon of their fund (3-8 years). Funds are only returned to VC investors when startups have liquidity events (IPO, Acquisition). However, recently, many companies that are considered successful are delaying their liquidity events and seeking later-stage private funding (e.g. Uber). In this case, whether a company has raised additional funding rounds may be used as a proxy for investment success. Unless otherwise specified, we performed our previous analyses against our base target outcome, Extra Stage (i.e. whether a company raises an additional funding round, is aquired or has an IPO). In the following section, we explore whether the component outcomes (e.g. predicting IPOs) has an affect on our system's predictive power.

Figure 4.11 shows F1 Scores grouped by target outcome and forecast window. First, we examine the baseline distribution and then the variation in performance by forecast window. Our model is most accurate at predicting extra funding rounds and worst at predicting IPOs. As we observed in Figure 4.9, there is a positive relationship between length of forecast window and model performance.

This relationship has a similar magnitude across all target outcomes except for IPOs which improve much more when the forecast window is increased from 2 to 3 years.

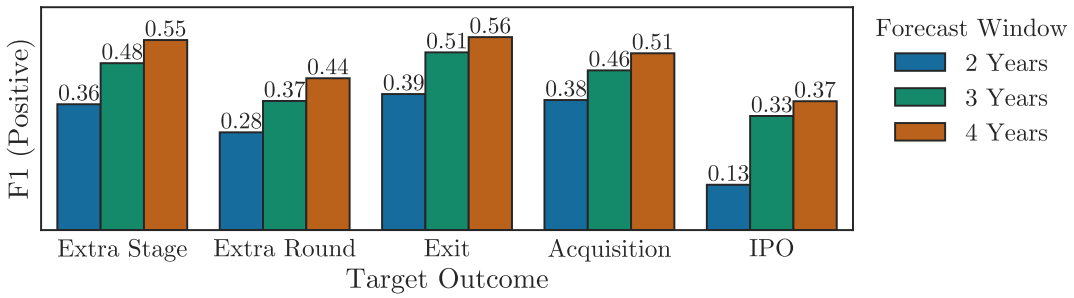


Figure 4.13: Performance by target outcome.

Figure 4.14 shows the standardised feature weight distribution, grouped by target outcome. Models of target outcomes produce considerable variance in feature weights. Exit and Acquisition have similar feature weights. Investors, Executives and Founders are key features for Exits and Acquisitions. In comparison, IPOs have more weighting towards Funding, Advisors and the Broader Economy. Extra Round is most strongly related to Investors and Funding and Extra Stage is most strongly related to Advisors.

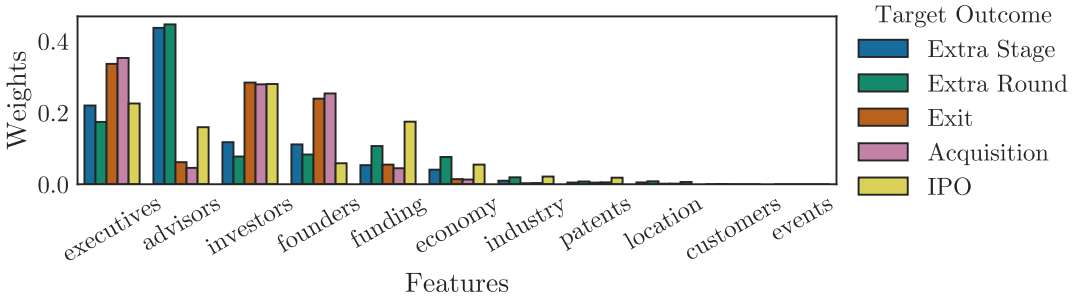


Figure 4.14: Feature weights by target outcome.

CHAPTER 5

Discussion

In previous chapters, we outlined how we designed and evaluated a novel data mining system for automated Venture Capital (VC) investment screening. In this chapter, we discuss the merits of this project with respect to our system's design, our system's performance, and its contribution to theory on startup investment.

1. System Design. We developed a data mining system that provides automated VC investment screening. Our system uses data collected from CrunchBase and PatentsView. We found that CrunchBase and PatentsView databases are large, comprehensive, and growing. However, CrunchBase records are sparse, long-tailed, and require cleaning. PatentsView features have minimal overall impact on the system's performance, but provide a significant improvement in prediction performance for companies at later developmental stages. Our system generates a classification pipeline optimised to the dataset. We found classification algorithm tuning had the greatest impact on performance during optimisation. In particular, Random Forests and Logistic Regressions were the most successful classifiers, Support Vector Machines and Artificial Neural Networks underperformed. We found that the top 3-5 pipelines generated from the optimisation process should be checked for their robustness over time. A limitation of our design is that the dataset slicing technique may introduce biases related to artificial completeness. Our system is semi-autonomous but could be made fully autonomous with further development.

2. System Performance. We evaluated the performance of our VC investment screening system. We found that our system's performance is robust with respect to historical datasets (for 2012-16), which makes it suitable for forward-looking predictions. We found that the performance of our system was better or comparable to previous results from the literature. Performance was positively related to longer forecast window (for a period of 2-4 years), later developmental stage (e.g Series C, Series D+), and breadth of target outcome (e.g. Exit). Our system's observed performance may have

been improved further if we had performed pipeline optimisation separately for each experiment. A limitation of our system is that some nuances of investment success are not fully captured (e.g. down-rounds, acqui-hires). However, we still believe our system has performance practical for use in investment screening.

3. Model Evaluation. We developed a conceptual framework for startup investment performance based on the literature. This framework guided our data source selection and feature creation. We evaluated the framework using our data mining system. We found that models of startup investment performance generated by our system are robust with respect to time (for 2012-16) and forecast window (over 2-4 years), and vary with respect to developmental stage (e.g. Seed, Series A) and target outcome (e.g. IPO, Acquisition). Our system does have some limitations. We were unable to represent all factors from our conceptual framework (e.g. financial information), we may have understated some factors by not using more complex features (e.g. temporal relationships), and we were unable to generate structural models. Despite these limitations, we believe this project has made contributions to models of startup investment performance.

5.1 System Design

We developed a data mining system that provides automated Venture Capital (VC) investment screening using data collected from large online databases, CrunchBase and PatentsView. This system provides a multi-stage pipeline optimisation process that can automatically adapt to changes in the dataset or prediction task over time.

5.1.1 Data Collection

Our system uses data collected from CrunchBase and PatentsView online databases. Both data sources are fairly new (CrunchBase gained critical mass in 2012 and PatentsView was formed in 2015) and neither have been studied frequently in the literature. These data sources offer a significant improvement in terms of size and variety of features over previous data sources used in this research field (e.g. surveys, interviews, closed datasets). However, we found CrunchBase was sparse, with many long-tailed features, and other abnormalities, and required considerable cleaning to remove irrelevant companies. We addressed these issues with a number of pre-processing steps in our classification pipeline. Unlike CrunchBase,

PatentsView is a government-regulated data source, so it has fewer data quality issues. The greatest issue we had with PatentsView was matching companies between PatentsView and CrunchBase because companies often use variations on their names or have sub-entities. However, PatentsView data still produced a broad performance improvement to our system.

Our system converts the data we collect into historical datasets by using timestamps. While this technique provides significant benefits to our system, it also raises some concerns. We performed preliminary testing of our dataset slicing technique using last-updated timestamps and realised this would remove too many recently-updated records from the dataset. Instead, we used created-at timestamps which retain more records, but at the cost of possibly using features not originally available. While the impact of this effect is mediated by the relational database structure (e.g. acquisition and investment records have separate timestamps), it may artificially inflate historical results for companies that have had many later edits to their records. We tried to evaluate the impact of this technique by comparing a CrunchBase database collected in December 2013 with a slice from our primary dataset collected in September 2016. We found there was only minimal variance in the number of records found in each relation. However, because the database schema changed dramatically between 2013 and 2016, we were unable to determine whether the completeness of each respective record is similar. As we collect more original database dumps, we will be able to better evaluate this technique.

The current implementation of our system downloads CSV-dumps of the CrunchBase data and connects to the PatentsView API. While both of these data collection methods are an improvement upon the status quo (manual data collection) they can be improved further. In the earlier stages of our project we developed a connector to CrunchBase's API that provided real-time access to their database. CrunchBase publishes a daily change-log which our connector could use to request only information from nodes that had changed. Although we abandoned this approach because of time constraints, it deserves further investigation. A fully time-stamped database produced in this manner would also allow for greater accuracy and analysis of temporal trends, and avoid the issues raised by our current dataset-slicing technique. Although our current method of collecting data from PatentsView is through their API, unlike CrunchBase PatentsView does not provide a change-log. This limitation means we have to run a full database sweep (approximately 10 hours) whenever we want to check our patent filing records are up-to-date. We hope PatentsView will add this functionality in the near-future.

5.1.2 Pipeline Optimisation

A key component of our system is the pipeline optimisation process. While previous studies in this field have applied a few specific classification algorithms, we developed a pipeline optimisation process with the aim of greater accuracy and re-calibration of the system as the dataset changes over time. Our pipeline optimisation process is divided into two steps: pipeline creation and pipeline selection.

Pipeline creation performs a broad search and evaluation of candidate pipelines with varying hyper-parameters. This search is performed across the pre-processing steps of the pipeline and also the classification algorithms. We found classification algorithm tuning had the greatest impact on performance during optimisation. It appears very little optimisation of the pre-processing steps were needed. In aggregate, the performance of the classification pipeline was not improved by the pre-processing steps. However, it is likely the effect of the pre-processing steps was also dependent upon the classification algorithm (e.g. Random Forests are more resilient to low orthogonality than Naive Bayes), which may have reduced the aggregate effect. Nonetheless, optimisation of the pre-processing steps should still improve the overall robustness of the optimisation process as the dataset and prediction tasks change.

Our literature review suggested Random Forests would be the most successful classifier, followed by Artificial Neural Networks and Support Vector Machines. We found Random Forests and Logistic Regressions performed best and Artificial Neural Networks and Support Vector Machines underperformed. Random Forests may have outperformed the other algorithms due to its robustness to missing values and irrelevant features. Learning curves also revealed that, unlike most of the other classifiers, Random Forests was least likely to converge early, which suggests with larger training sets it should perform better. Logistic Regression. It was surprising that Support Vector Machines and Artificial Neural Networks underperformed the other algorithms. However, these algorithms are far more difficult to accurately tune than these other algorithms, so it may reflect that our search process was too limited. In production, we could perform the search process over longer iterations which would likely result in better performance from these algorithms.

The second step in our system's pipeline optimisation process is pipeline selection. In this component, we rank the candidate pipelines generated previously and evaluate the best pipelines (finalist pipelines) over a number of different dataset slices. This process ensures our final pipeline is robust in their performance with respect to time. We don't observe significant variance in the pipelines on aggregate against the dataset slices, but there is variance within the individ-

ual pipelines. The former result suggests our pipelines produce models that are robust with respect to time (which is reinforced in the final evaluation). The latter result justifies this step in our process. In our preliminary evaluation of this test we selected the top ten candidate pipelines. Although there is still a strong positive correlation between the pipelines initial ranking and their scores, we can see there are some individual deviations. Importantly, the top-ranked pipeline from the first stage actually has a lower median score than the second-ranked pipeline. These results suggest it is optimal to evaluate the top 3-5 candidate pipelines in this manner.

5.1.3 Automation & Efficiency

A key benefit of our proposed VC investment screening system is that it will reduce the amount of manual effort required prior to the investment decision-making process. The implementation of the system described in this paper goes some ways to address this. Currently the system is semi-autonomous: it has little requirement for external input besides configuration of investment criteria (e.g. forecast window, developmental stage etc.), but still runs on-demand, rather than continuously.

An improved implementation of the system would run in the background continuously, scheduling components of the system to run as needed to ensure the results are always optimised. The system currently takes a total of 46 hours to complete. Most of this time is taken by the pipeline creation component, which performs a large search across potential pipeline hyper-parameters. However, when placed into production, this component could be run infrequently - perhaps once per year - to ensure the pipelines being used are still optimally suited for the dataset. The next component of the system, selecting the most robust pipeline, could occur more frequently - perhaps once every month - and the final component of the pipeline, making up-to-date predictions, could be evaluated every time new data is fed into the system (perhaps once per day) because it only takes an hour.

When we evaluated the learning curves of the pipeline selected by our system, we found our system's performance had converged for some target outcomes but for others (e.g. IPO, Acquisitions) it had yet to converge. This suggests our system could still benefit from a larger dataset. However, these results could also arise because our pipeline was optimised for predicting our base target outcome, and if the entire system was performed on the different target outcomes we might find other classification pipelines yield better performance. A key advantage of our system is that, as we collect more data over time, our pipeline optimisation process will adapt to the nature of the dataset and select classifiers with less bias

and more variance, so we may see Support Vector Machines or Artificial Neural Networks adopted by the system in the future.

5.2 System Performance

We evaluated the performance of our Venture Capital (VC) investment screening system across a range of datasets with different training dates, forecast windows, developmental stages, and target outcomes. In this section, we discuss the performance of our system across these domains, with comparison to previous systems from literature. We also discuss the limitations of our experimental design.

5.2.1 Time Slices

We evaluated the robustness of our system's performance by training the system on datasets of different dates from 2012-2014. Robustness with respect to time is a critical attribute for our system so VC firms can rely that models trained on historical datasets will be accurate into the future. We evaluated the performance of the system using a variety of evaluation metrics. Across all evaluation metrics, we observed little variance in performance. We observed that performance variance decreased as forecast windows became longer in duration. This seems reasonable as we would expect longer-term models might be based on more fundamental features that are less likely to vary with respect to time. This trend may also be due to greater variance in the dates of datasets with longer forecast windows. This is because we are restricted in how far back in time we can reverse-engineer data slices without reducing the training size by too much.

5.2.2 Forecast Window

We evaluated our system's performance against forecast windows of 2-4 years and a variety of evaluation metrics. We observe a positive relationship between performance and length of forecast window. This trend suggests that it is harder to predict when a startup company will raise a funding round or exit than whether it will do it at all. This seems like a reasonable proposition as non-performance related factors (e.g. finding an investor with the right fit, requiring extra funding to enter a new market) may influence the timing of each activity. In the future, it would be interesting to explore forecast windows closer in duration to a typical VC investment horizon (5-8 years). If we decompose the trend further, we see that there is a relationship between the magnitude of the performance improvement

and how sensitive the evaluation metric is to both the imbalanced nature of the dataset and our bias towards the positive class. Accordingly, F1 Scores show the greatest performance improvement and Area under the Receiver Operating Characteristic (ROC) curve show little trend.

5.2.3 Developmental Stage

We compared performance of our system across target companies of different developmental stages ranging from Pre-Seed through Series D+. We found a positive trend between later developmental stage and performance. This is possibly a product of later-stage companies having more complete feature vectors. Beckwith (2016) studied companies seeking equity crowd-funding (which maps to Pre-Seed in our classification system) and showed poor classification results even just predicting whether a company would raise the equity crowd-funding round, let alone exit at a later date [6]. Stone (2014) suggested that VC investment screening was simply not viable prior to Series A stage [48]. Bhat (2011) studied companies that had previously raised three VC rounds (Series C in our classification) and received comparable results to our system [7].

A discrepancy in the positive trend between developmental stage and performance is a slight decrease in our system's performance at Series D+. This decrease may be because the model is primarily predicting exits at this stage, rather than additional funding rounds, and exits are harder to predict. To investigate this discrepancy further, we split the datasets into their developmental stages and fit the model onto each of these sub-datasets individually. Pre-Seed companies make up most of our original dataset and we see the smallest improvement for this stage. However, for Series D+ we see a large improvement, which suggests the features that predict Series D+ performance vary from earlier stages. Overall, this stage-specific fit method results in a broad performance improvement. This is despite each model having significantly less observations to train on, which suggests that the underlying factors that influence startup investment performance for each stage are significantly different.

5.2.4 Target Outcome

Ultimately, our system seeks to identify startup investment opportunities that will return their invested funds many times over within an investment horizon of a VC's fund (typically 3-8 years). However, our dataset has little information about valuations at funding rounds or during acquisitions because valuation is considered sensitive and confidential. Instead we developed broader target

outcomes as rough proxies for the underlying success of the investment. These outcomes include raising additional funds, being acquired, having an IPO and combinations thereof. We evaluated each of these outcomes separately to determine their effect on our system's performance. As we would expect, our system is better at predicting more common events, so performs best at predicting additional funding rounds and worst at predicting IPOs. The system's poor performance on IPOs may also be due to non-performance related factors that affect IPO timing, like financial market conditions. Surprisingly, our system is actually better at predicting whether a company will exit, than whether it will exit or gain additional funding. This requires further research.

While our target outcomes provide a rough proxy for investment success, there are nuances that can't be captured by these outcomes. While most funding rounds are generally at higher valuations than the previous round, some funding rounds are not - these are termed 'down-rounds'. Likewise, although most acquisitions are performed at higher valuations, sometimes they are not - these are often termed 'acqui-hires'. As our publicly-sourced dataset has little information about valuations at funding rounds or during acquisitions, our system has little ability to distinguish between successful activity and down-rounds or acqui-hires. These discrepancies limit the performance of our system. In AppendixH we present four case studies that highlight the nuances of our system's performance. In future, applying sentiment analysis to media coverage of funding rounds, acquisitions or IPOs may indicate whether the activity was genuinely successful.

5.2.5 Experimental Design

Our experimental design involved evaluating the performance of the system across a range of variables, including size of training set, date of training set, duration of forecast window, company developmental stage, and target outcome. For each of these experiments, we manipulated these variables during the model fit and prediction step of our system design. However, to reduce the time taken by our experiments, we used the same optimised pipeline for each experiment (for the configuration, see AppendixF). This pipeline optimisation step takes the vast majority of time of our system (84.8%). By using a pipeline optimised for different objectives we are likely to have under-reported the performance of our system. In future research, it would be interesting to determine the extent to which the results of our pipeline optimisation changes with respect to these variables and the extent to which our results improve.

5.3 Model Evaluation

As a by-product of the evaluation of our system, we are also able to provide a comprehensive study of the determinants of startup investment performance. From our literature review, we developed a conceptual framework for startup investment performance, based on previous work by Ahlers and colleagues [1]. Our conceptual framework posited that startup investment decisions are based on two primary components: startup potential and investment confidence. We decomposed these components further into 15 factors that were identified in previous empirical studies in the literature. Many of these factors are evaluated by our system. Through our experimentation on the system, our system generated models that describe features associated with startup investment performance over time, with different forecast windows, developmental stages, and target outcomes.

5.3.1 Time Slices & Forecast Window

We evaluated our system by training it on a range of historical datasets from 2012-16 across varying forecast windows. We found that models generated by the system were robust to changes in training date, with a standard deviation of less than 1% of the total normalised feature weights. We found a similar trend with respect to forecast window, with little variance for periods of 2-4 years. Despite this, we previously observed a positive relationship between system performance and forecast window length. Together, these findings suggest that for a given set of independent variables (developmental stage and target outcome), our models of startup investment performance are stable over time. This is somewhat in contradiction to the widely held within the Venture Capital (VC) industry that the factors that influence startup investment performance change over time. Admittedly, our models do not perfectly predict startup investment success and this margin of error might be caused by dynamic factors not incorporated in our system. If not to a decision-making degree of performance, our results still suggest features that correlate with startup investment success are predictable and stable.

5.3.2 Developmental Stage

We evaluated our system with respect to the developmental stage of our target companies. We found considerable variation in models developed for companies of different stages. We find that Advisors are more important to earlier stage companies than late stage companies. This is a surprising and perhaps counter-

intuitive finding that has not been previously substantiated in the literature. One explanation is that successful startups recruit experienced, influential advisors earlier to fill in gaps in their experience but by the time these startups reach later-rounds, this advantage is neutralised by the effect of bringing in investors as advisors. We found that Investor track record becomes more important as companies approach an exit (Series D+). This may be due to top investors having more experience at going through exit processes, and leveraging their contacts to make the process easier. We found that Executive and Founder experience are particularly important in Pre-Seed companies. This is a factor that has previously been substantiated in the literature [6, 3]. At an early-stage of a company's development, there is little to rely on aside from the previous experience and skill-set of the founding team and staff. Finally, we found that Economic factors were most important at the Pre-Seed stage and have little effect at other stages. We would not expect economic factors to have a large effect on startup performance (at least compared to larger, more established companies) because startups are quite flexible, agile and target more niche markets. One explanation for the greater impact of Economic factors at the Pre-Seed stage is that perhaps more people leave established companies to launch startups when the economy is doing poorly, but these companies do not survive longer than the Pre-Seed stage. Overall, investigation into startup investment performance across the startup development life-cycle has previously been largely neglected. Most research in this field has been performed on either early-stage companies [6, 48, 55, 1], or much later-stage companies [7]. We believe that this is the first comprehensive study that takes a broader cross-sectional approach.

5.3.3 Target Outcome

We evaluated our system with respect to different target outcomes (funding rounds, acquisitions, Initial Public Offering (IPO)s and combinations thereof). We found that this experimentation generated the most amount of variance within our models of startup investment performance. We would expect to see broad similarities between the models that predict Acquisitions, IPOs and their combination (Exits). There are some similarities between the models for these outcomes, especially between Acquisitions and Exits which is probably because Acquisitions make up a large proportion of Exits in our database. Investors and Executives have similar weightings across all three outcomes. As mentioned in the previous section, we expected Investors to be a good predictor of Exits because Investors with a track record should be able to better prepare startups better for Exit, in terms of procedures and connections. IPOs are slightly more weighted towards economic factors which makes sense given that IPOs are gen-

erally more sensitive to market conditions. Extra Round is most strongly related to Investors and Funding. This is an interesting finding, which may suggest that funding momentum plays a role in startup investment success (i.e. investors are pre-disposed to investing in companies that have already received investment, despite their fundamental characteristics). Finally, and surprisingly, the model for Extra Stage is most strongly weighted towards Advisors. It is unclear why Extra Stage, which is a combination of all other target outcomes, would not have weightings that tend towards the mean of the models of the other outcomes. Perhaps the ability of a startup to attract high-quality Advisors is a proxy for a more fundamental level of company quality. Further research should investigate this in more detail. Previous studies have separated predicting funding round outcomes [6, 55, 1, 3] from predicting exits [7]. This separation in the literature is probably at least partially due to an interaction with developmental stage. Most studies of funding rounds focus on early-stage funding whereas exits tend to focus on later-stage companies. Our study bridges this divide in a way that is more holistic and thus more useful for investors.

5.3.4 Limitations

5.3.4.1 Missing Features

While CrunchBase and PatentsView provide features that cover much of our conceptual framework, there are factors we were unable to evaluate in this implementation of our system. Missing factors include: media coverage, social media influence, strategic alliances and financial performance. While it would likely be a significant factor in our models, we do not expect to be able to source financial information for our dataset in the future. In fact, it is a key benefit of our system that it can perform accurately without detailed financial information. The paucity of available financial data is what makes VC investment screening distinct from other fields of finance. Data to support the other missing factors may be easier to source in the future. The CrunchBase API provides an archive of media coverage on each startup company, so connecting directly to the CrunchBase API (rather than the CSV-dumps) would give access to this feature. Social media influence is more difficult, because historical records are hard to find. CrunchBase does track whether companies have social media profiles, but does not provide time-stamps for this information which limits our ability to create historical records. There are a number of Twitter historical data services, but these are expensive to use. Finally, strategic alliance information (e.g. with suppliers or universities) is not a typical feature that is recorded but could be engineered through textual analysis on media coverage. These features should be

investigated in future work that builds on our system.

5.3.4.2 Homogeneous Features

We incorporated features that covered a broad framework into our models, but the nature of these features was largely homogeneous. Most features were fairly basic: booleans, counts, summations, averages. This is in line with most of the prior work in this field that focused on basic company features (e.g. the headquarters' location, the age of the company) [6, 20]. However, there is preliminary research that has looked into using semantic text features (e.g. patents, media) [25, 55] and social network features (e.g. co-investment networks) [52, 54, 12] with some success. We expect a model that includes semantic text and social network features alongside basic company features could lead to better startup investment prediction. However, a disadvantage of making models with dynamically generated features is that it becomes difficult to train and test on different datasets because the features do not align. This would have limited our ability to test the robustness of our system in the fashion that we did, training on different datasets and then comparing the results. Finally, aside from evaluating our system's performance against historical datasets and for different forecast windows, we did not incorporate temporal relationships into our system. However, in other forms of finance, like equity markets, time-series analysis is relied on heavily. Perhaps future research can look at the temporal relationships between different startup activities (e.g. media coverage, funding rounds, IPOs etc.) and how this chain of activity might predict investment success.

CHAPTER 6

Conclusions

Our project's aim was to produce a Venture Capital (VC) investment screening system that is practical, robust and versatile. Our system uses online data sources and machine learning to identify startups that are likely to raise additional funding, become acquired or have an Initial Public Offering (IPO) (or some combination therof) in a given period of time. While this is a challenging problem even for experienced VC investors, our system achieved results that have practical application for VC firms.

6.1 Evaluation of Criteria

6.1.1 Practicality

Our system intends to replace manual investment screening (referral, Google search, industry papers, and manual search of startup databases). Our automated system should be more efficient than these methods because it requires less human input. We also evaluated our system's efficiency based on dataset size and time profile. An indicative implementation of our system takes 46 hours to run, which is reasonable for a process that is not time-sensitive in this industry. The majority of this time is due to the pipeline optimisation process. However, when placed into production, this could be run less frequently with minimal reduction in performance.

6.1.2 Robustness

Our system must be robust in its performance with respect to time so investors can rely on its predictions. The Venture Capital (VC) industry is concerned that predictive models trained on historical data will not accurately predict future trends and activity. This has been identified as a key barrier to the adoption

of automated systems by the VC industry [48]. We evaluated our system across a number of historical datasets, forecast windows, and even multiple evaluation metrics. We found variance across all evaluation metrics to be very low, with slightly more variance over shorter forecast windows. When we explored the feature weights for each model developed on different historical datasets, we found only slight variance. This suggests that our system produces highly robust models, suitable for forward-looking investment screening.

Our system is relatively robust to dataset size, because it optimises the classification pipeline based on the datasets available. In some cases, our system could use smaller training sets without significant reduction in predictive power. However, our system's ability to use smaller training sets was related to the length of the desired forecast window, and the breadth of the target outcome.

6.1.3 Versatility

Our system must be consistently accurate at identifying a variety of high-potential investment candidates. Unlike previous studies, we evaluated the systems' predictive power across a large problem domain, including predicting over different forecast windows (e.g. 2-4 years), for target companies at different developmental stages (e.g. Seed, Series A etc.), and for different target outcomes (e.g. predicting additional funding rounds). Forecast window has an impact on our system's performance. Our system produced F1 Scores of 0.36, 0.48 and 0.55 for forecast windows of 2, 3 and 4 years. Our system's performance has a positive relationship with developmental stage, producing F1 Scores ranging from 0.33 for Pre-Seed companies through to 0.62 for Series C companies. Finally, our system varies in its performance at predicting target outcomes, producing F1 Scores ranging from 0.51 for predicting additional funding through to 0.24 for predicting an Initial Public Offering (IPO). Where comparable, our system produced better or similar results to previous studied systems.

6.2 Future Work

6.2.1 Automation & User Interface

Our current implementation of the system is near-autonomous, but still requires manual scheduling. Future work could prepare this system for use in industry by developing a set-and-forget task scheduling system that optimises when different components of the system are run to ensure that the performance of the system

is always near-optimal. This task scheduling system would pair well with an improved CrunchBase data collection system that connects directly to the REST API rather than downloading CSV-dumps. The REST API collector could indicate how many changes have been made in the dataset since the last data collection, and feed that information to the task scheduler. This would allow the task scheduler to run an optimisation process that maximises performance improvement against number of dataset changes and time taken. In addition to task scheduling and API-based data collection, our system requires a basic user interface before it can be commercialised, allowing users to search through the database, filter based on the results of the models, and change configuration of the system.

6.2.2 Feature Set Improvement

While this project was likely the most comprehensive study of startup investment performance in the literature with respect to diversity of features analysed, there are improvements to our feature set to be made in future work. In our implementation of the system, we were unable to source data to represent all the factors of our conceptual framework. Missing factors included: media coverage, social media influence, strategic alliances and financial performance. These features have all previously been indicated to be associated with startup investment performance. In addition, the nature of our feature set was largely homogeneous – mostly basic features (e.g. number of funding rounds). We expect that future work that extends our feature set by including semantic text features (e.g. keyword analysis from patents, sentiment analysis from media coverage) and social network features (e.g. co-investment networks, spheres of influence) could lead to better startup investment prediction. Finally, it would be interesting to look further at the temporal relationships between different startup activities (e.g. media coverage, funding rounds, Initial Public Offering (IPO)s etc.) and how this chain of activity might predict investment success (e.g. Markov networks).

6.3 Summary

In this project we set out to create a Venture Capital (VC) investment screening system that met our criteria of practicality, robustness, and versatility, and we have indeed created such a system. The work required to achieve this project's goals was extensive, from reviewing the state of VC theory and machine learning as applied to this field, to developing systems that collect and manipulate data from CrunchBase and PatentsView, to developing an adaptive classification

pipeline process, and finally performing experiments that validate the ability of the system to meet our aforementioned criteria.

This project makes three primary contributions with implications for industry and research. First, our system is designed for the VC industry: it is near-autonomous, robust to changes in dataset and prediction task, and uses diverse features collected from large public online databases. Second, our system's performance is not only better or comparable to previous studies, it also addresses a far larger domain of investment prediction tasks with respect to forecast window, developmental stage and target outcome. Third, this project contributes an empirical study of models of startup investment performance more comprehensive than any found in the literature. Ultimately, this project makes steps towards automation in the VC industry.

APPENDIX A

Data Sources

A.1 Databases

Databases play a critical role in understanding the startup ecosystem, aggregating information about startups, investors, media and trends. Most startup databases are closed systems that require commercial licenses (e.g. CB Insights, ThomsonOne, Mattermark). CrunchBase and AngelList are two crowd-sourced and free-to-use alternatives. AngelLists primary function is as an equity crowdfunding platform but it has a data-sharing agreement with CrunchBase which results in significant overlap between the two sources. CrunchBase and AngelList provide free Application Program Interfaces (API) for academic use. Crawlers can be developed to traverse these APIs and collect data systematically. The advantages of crawlers are that they can selectively collect data from nodes with specific attributes, collect random samples, or traverse the data source indefinitely, updating entries as new data becomes available. CrunchBase also provides pre-formatted database snapshots which allows easier access to the data set. The crowd-sourced nature of CrunchBase and AngelList has advantages and limitations . The key advantages are that access to the databases is free and the dataset is relatively comprehensive. The limitations are that both CrunchBase and AngelList have relatively sparse profiles (i.e. limited depth), particularly for unpopular startups. Both CrunchBase and AngelList also have error-checking provisions (including machine reviews and social authentication) to prevent and remediate inaccurate entries but there is still a greater chance for error. Comparing CrunchBase and AngelList, CrunchBase tends to have more comprehensive records of funding rounds [12] and media coverage but AngelList also has a social network element where users can ‘follow each other - in a similar way to Twitter.

A.2 Social Networks

Social networks provide an interesting perspective into the process of opportunity discovery and capitalisation that characterises entrepreneurship. Two social networks studied in detail in entrepreneurship research are LinkedIn and Twitter. LinkedIn is a massive professional social network often used in studies of entrepreneurship for measures of employment, education and weak social links. These measures are difficult to collect elsewhere. In addition, LinkedIn can provide a measure of the professional influence of founders and investors. Unfortunately, as of May 2015, the LinkedIn API no longer allows access to authenticated users' connection data or company data [51], making it difficult to use for social network analyses. Twitter is a massive social networking and micro-blogging service which is studied in entrepreneurship research because it is used by founders, investors, and customers to quickly communicate and broadcast. Twitter is a directed network where users can follow other users without gaining their permission to do so. Twitter's public API provides access to social network topological features (e.g. who follows who) and basic profile information (e.g. user-provided descriptions). However, Twitter's API only provides Tweets published within the last 7 days and access to historical Twitter data requires a commercial license [38].

A.3 Other Sources

While startup databases and social networks provide a variety of information on startups, there are two important areas that they do not cover: patent filings and financial performance. Startups often file patents to apply for a legal right to exclude others from using their inventions. In 2015, the US Patents Office (USPTO) launched PatentsView, a free public API to allow programmatic access to their database. PatentsView holds over 12 million patent filings from 1976 onwards [42]. The database provides comprehensive information on patents, their inventors, their organisations, and locations. It may be difficult to match identities across PatentsView to other data sources because registered company names (as in PatentsView) are not always the same as trading names (as elsewhere). Finding other information on startups, like financial information, is difficult. Unlike public companies, private companies are not required to file with the United States Securities and Exchange Commission (or international equivalent). Proprietary databases provide some data on private companies but commercial licenses are prohibitively expensive and have poor coverage of early-stage companies. PrivCo is one of few commercial data sources for private company business and financial

intelligence. PrivCo focuses its coverage on US private companies with at least $50-100 million in annual revenues but also has some coverage on smaller but high-value private companies (like startups) [4].

APPENDIX B

Classification Algorithms

B.1 Naive Bayes

Naive Bayes is a simple generative learning algorithm. It is a Bayesian Network that models features by generating a directed acyclic graph, with the strong (naive) assumption that all features are independent. While this assumption is generally not true, it simplifies estimation which makes Naive Bayes more computationally efficient than other learning algorithms. Naive Bayes can be a good choice for data sets with high dimensionality and sparsity as it estimates features independently. Naive Bayes sometimes outperforms more complex machine learning algorithms because it is reasonably robust to violations of feature independence [29]. However, Naive Bayes is known to be a poor estimator of class probabilities, especially with highly correlated features [34]. Naive Bayes was used alongside Logistic Regression, Decision Trees and Support Vector Machines to predict success in equity crowdfunding campaigns on the AngelList data set [6]. None of these models performed well. The algorithm that best predicts startup investment was Naive Bayes with a Precision of .41 and Recall of .19, which means only 19% of funded startups were classified correctly by the model. The author suggests the poor performance of their algorithms is caused by features not captured in their data set relating to Intellectual Capital, Third Party Validation and Historical Performance. These features will be included in this study.

B.2 Logistic Regression

Regression is a class of statistical methods that investigates the relationship between a dependent variable and a set of independent variables. Logistic regression is regression where the dependent variable is discrete. Like linear regression, logistic regression optimises an equation that multiplies each input by a coeffi-

cient, sums them up, and adds a constant. However, before this optimisation takes place the dependent variable is transformed by the log of the odds ratio for each observation, creating a real continuous dependent variable on a logistic distribution. A strength of Logistic Regression is that it is trivial to adjust classification thresholds depending on the problem (e.g. in spam detection [19], where specificity is desirable). It is also simple to update a Logistic Regression model using online gradient descent, when additional training data needs to be quickly incorporated into the model (incremental learning). Logistic Regression tends to underperform against complex algorithms like Random Forest, Support Vector Machines and Artificial Neural Networks in higher dimensions [10]. This underperformance is observed when Logistic Regression is applied to startup investment prediction tasks [6, 7]. However, weaker predictive performance has not prevented Logistic Regression from being commonly used. Its simplicity and ease-of-use means it is often used without justification or evaluation [20].

B.3 K-Nearest Neighbours

K-Nearest Neighbours is a common lazy learning algorithm. Lazy learning algorithms do not produce explicit general models, but compare new instances with instances from training stored in memory. K-Nearest Neighbours is based on the principle that the instances within a data set will exist near other instances that have similar characteristics. K-Nearest Neighbours models depend on how the user defines distance between samples; Euclidean distance is a commonly used metric. K-Nearest Neighbour models are stable compared to other learning algorithms and suited to online learning because they can add a new instance or remove an old instance without re-calculating [29]. A shortcoming of K-Nearest Neighbour models is that they can be sensitive to the local structure of the data and they also have large in-memory storage requirements. K-Nearest Neighbours was compared to Artificial Neural Networks to predict firm bankruptcy [2]. K-Nearest Neighbours is attractive in bankruptcy prediction because it can be updated in real-time. By optimising feature weighting and instance selection, the authors improved the K-Nearest Neighbours algorithm to the extent that it outperformed the Artificial Neural Networks.

B.4 Decision Trees

Decision Trees use recursive partitioning algorithms to classify instances. Each node in a Decision Tree represents a feature in an instance to be classified, and

each branch represents a value that the node can assume. Methods for finding the features that best divide the training data include Information Gain and Gini Index [29]. Decision Trees are close to an "off-the-shelf" learning algorithm. They require little pre-processing and tuning, are interpretable to laypeople, are quick, handle feature interactions and are non-parametric. However, Decision Trees are prone to overfitting and have poor predictive power [11]. These shortcomings are addressed with pruning mechanisms and ensemble methods like Random Forests, respectively. Decision Trees were compared with Naive Bayes and Support Vector Machines to predict investor-startup funding pairs using CrunchBase social network data [31]. Decision Trees had the highest accuracy and are desirable because their reasoning is easily communicated to startups.

B.5 Random Forests

Random Forests are an ensemble learning technique that constructs multiple Decision Trees from bootstrapped samples of the training data, using random feature selection [8]. Prediction is made by aggregating the predictions of the ensemble. The rationale is that while each Decision Tree in a Random Forest may be biased, when aggregated they produce a model robust against over-fitting. Random Forests exhibit a performance improvement over a single Decision Tree classifier and are among the most accurate learning algorithms [11]. However, Random Forests are more complex than Decision Trees, taking longer to create predictions and producing less interpretable output. Random Forests were used to predict private company exits using quantitative data from ThomsonOne [7]. Random Forests outperformed Logistic Regression, Support Vector Machines and Artificial Neural Networks. This may be because the data set was highly sparse, and Random Forests are known to perform well on sparse data sets [8].

B.6 Support Vector Machines

Support Vector Machines are a family of classifiers that seek to produce a hyperplane that gives the largest minimum distance (margin) between classes. The key to the effectiveness of Support Vector Machines are kernel functions. Kernel functions transform the training data to a high-dimensional space to improve its resemblance to a linearly separable set of data. Support Vector Machines are attractive for many reasons. They have high predictive power [11], theoretical limitations on overfitting, and with an appropriate kernel they work well even when data is not linearly separable in the base feature space. Support Vector

Machines are computationally intensive and complicated to tune effectively (compared to Random Forests, for example). Support Vector Machines were compared with back propagated Artificial Neural Networks in predicting the bankruptcy of firms using data provided by Korea Credit Guarantee Fund [46]. Support Vector Machines outperformed Artificial Neural Networks, possibly because of the small data set.

B.7 Artificial Neural Networks

Artificial Neural Networks are a computational approach based on a network of neural units (neurons) that loosely models the way the brain solves problems. An Artificial Neural Network is broadly defined by three parameters: the interconnection pattern between the different layers of neurons, the learning process for updating the weights of the interconnections, and the activation function that converts a neuron's weighted input to its output activation. A supervised learning process typically involves gradient descent with back-propagation [40]. Gradient descent is an optimisation algorithm that updates the weights of the interconnections between the neurons with respect to the derivative of the cost function (the weighted difference between the desired output and the current output). Back-propagation is the technique used to determine what the gradient of the cost function is for the given weights, using the chain rule. Artificial Neural networks tend to be highly accurate but are slow to train and require significantly more training data than other machine learning algorithms. Artificial Neural Networks are also a black box model so it is difficult to reason about their output in a way that can be effectively communicated. Artificial Neural Networks are rarely applied to startup investment or performance prediction because research in this area typically uses small and low-dimensional data sets. As one author puts it "More complex classification algorithms - artificial neural networks, Restricted Bolzmann machines, for instance - could be tried on the data set, but marginal improvements would likely result." [6]. However, this study will address these issues so Artificial Neural Networks may be more competitive.

APPENDIX C

Feature Selection

We develop a conceptual framework relating startup potential and investor confidence to startup investment. We will operationalise this conceptual framework into features that can be incorporated into our machine learning model. To do this, we review features that have been tested in previous studies related to startup investment or performance. In the following sections, we describe each of these features and outline conceptual and empirical evidence that justify their inclusion in our conceptual framework. Figure C.1 depicts how these features can be incoprorated into our conceptual framework.

C.1 Venture Quality

C.1.1 Human Capital

Human capital is critical to early-stage startups that have limited resources and are changing constantly. Startups are composed of founders, non-executive directors (NED) that may be investors or advisers, and staff. Each of these parties makes a contribution to the human capital of the startup. The human capital of these parties can generally be categorised three ways: education, prior experience, and synergies as a team.

Founder Capabilities Founders play multiple roles in early-stage startups, driving many aspects of the business growth and development. Accordingly, the human capital of founders has been shown to affect startup investment success. In particular, education of founders is a key signal. The number of degrees attained by founders is predictive of success [6, 20], as is whether a founder has obtained an MBA [6]. In addition, past entrepreneurial experience seems to be a predictive factor [20] though there is some evidence to dispute this [43]. Finally, the number of founders seems to be correlated

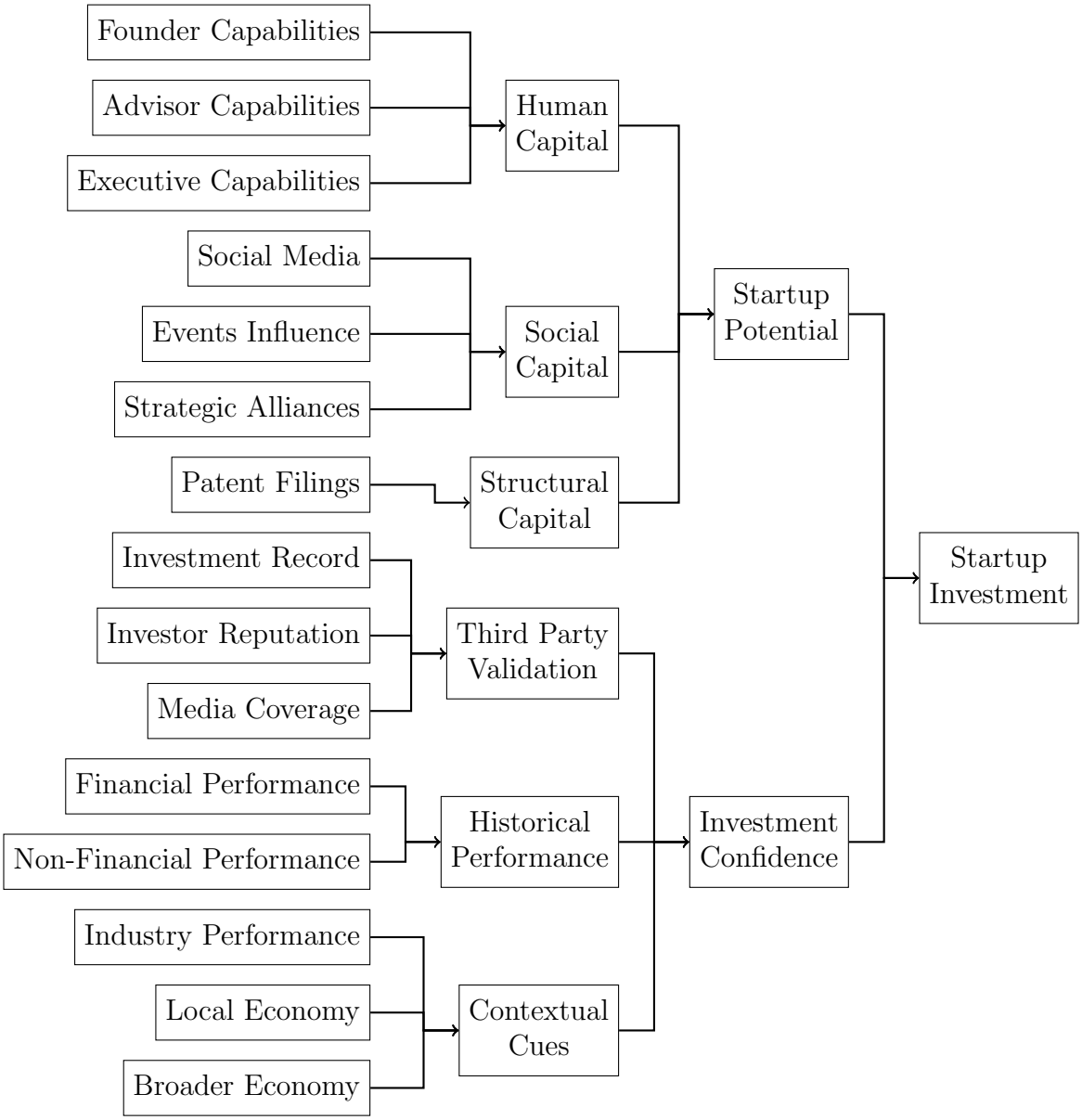


Figure C.1: Proposed conceptual framework for startup investment. This extended version of the framework includes features identified by empirical studies of startup investment. We adapt the framework proposed by Ahlers et al. [1], originally based on work by Baum and Silverman [5].

to startup success [6], though the underlying relationship may be more nuanced, and could be related to the distribution of team skillset.

Advisor Capabilities The boards of startups are smaller and have a higher concentration of ownership than those of well-established companies [27]. Startups lack corporate skills such as finance, human resources, information technology and legal expertise. Especially if founders are relatively inexperienced, they may look to the board to provide these skills. As a result, there is more overlap between governance and operational roles and directors may have greater influence on company performance through greater involvement in decision making [27]. Startups with more experienced directors are more successful at raising funds [5].

Executive Capabilities Founders play a key role in the very early stages of a startup and also in setting the culture for the organisation, but as the organisation grows more importance is given to the influence of employees. Measures like the number of current employees are broad representations of the startup's human capital and are correlated with subsequent startup investment [6, 3, 13]. Detailed analyses of executive human capital are not present in the literature but may be possible using data collected from sources like AngelList and LinkedIn.

C.1.2 Social Capital

Entrepeneurship revolves around opportunity discovery and realisation [45]. Opportunity discovery is only possible through the medium of social networks, so social capital is important. Social networks exist in many forms and contribute in different ways to social capital. These networks can be categorised in terms of the strength of their relationships: weak ties (e.g. social media) and strong ties (e.g. strategic alliances).

Social Media Startups use social media to communicate with other parties including their customers, potential customers, the media, potential employees, and potential investors. Social media activity can be proxy for a startup's social influence. Startups use different social media platforms for different purposes. Presence and engagement (e.g. number of followers, number of likes, number of posts) on Facebook and Twitter are predictive of startup investment success [12, 6]. These platforms are likely to capture customer or potential customer interactions, which is an indicator of

market adoption. In addition, the number of followers on AngelList predicts startup investment success [3], probably because it captures potential employees and investors' interest.

Event Influence

Strategic Alliances Strategic alliances with other companies or institutions have the potential to alter the opportunities that startups can access. Biotechnology startups that have links to industry partners are able to IPO more quickly and at higher market valuations [50]. Startups with more downstream (e.g. manufacturing), but not upstream (e.g. research and development), alliances obtain significantly more venture capital financing than startups with fewer such alliances [5].

C.1.3 Structural Capital

Structural capital is the supportive intangible assets, infrastructure, and systems that enable a startup to function. Intellectual property and their proxy, patents, are a key component of structural capital for newly-formed startups. Structural capital also includes processes and systems but these are less fully-formed in startups than more stable companies.

Patent Filings Many startups develop innovative technologies to help them capture a new market or better capture an existing market. Entrepreneurs protect their ideas through patent filings. Patents are an indicator of the technological capability of the startup. Patents and patent filings affect the survival and investment success of biotechnolgoy startups [5, 25]. However, there may not be as strong a relationship for non-biotechnology startups (e.g. software) [20, 1] This might be because factors like speed-to-market dominate the protective properties of patent filings in the quicker-moving high-technology sector.

C.2 Investment Confidence

C.2.1 Third Party Validation

By their nature, startups are optimsitic about the effectiveness of new technologies and business models. Founders are also highly invested in their startups and

therefore it is reasonable for investors to doubt their claims. Third party validation from credible sources like other investors, the media, and the government, may be factored into investors' decision-making process [26, 24].

Investment Record Intuitively, a track record of demand for investment is likely to be a strong signal of future likelihood of future investment. Average funding per round, number of investors per round, number of previous financing rounds and total prior funding raised all predict future likelihood of investment [1, 6, 14, 25, 13].

Investor Reputation Funding from reputable investors sends a clear signal to potential investors that a startup is likely to be of high quality. Investors may believe they require less due diligence because it has been performed by another investor. Startups that receive their initial funding round from a prominent investor are more likely to survive and receive higher valuations in initial public offerings [24]. Followers on AngelList and previous co-investors predict the likelihood of an investor's portfolio startups raising additional rounds successfully [3, 52].

Media Coverage Media coverage provides legitimacy and credibility to startups. Media attention for startups affects the perceived valuation of well-informed experts like venture capitalists [37]. This also translates to increased investment success [6]. There are a few possible explanations for this. First, media coverage signals public interest which might positively influence other stakeholders like customers, employees, etc. Second, new information become widely available which reduces perceived information assymetry.

C.2.2 Historical Performance

Startup performance is challenging to measure because there are no standardised reporting formats and the availability of data varies wildly. Capturing the multi-dimensionality of startup performance requires the use of multiple measures [53], however, most studies are only able to utilise simplistic performance metrics like survival time [39, 47, 21].

Financial Performance Despite being intuitive, there is little evidence of a relationship between startup financial performance and future investment success. This is because it tends to be difficult to access valid, accurate and complete financial performance measures (e.g. profit, revenue). This

information is considered by startups as private and confidential and unlike public companies, private companies are not required to make financial disclosures. Proprietary databases can provide some data on private companies but commercial licenses are expensive and have poor coverage of early-stage companies [4].

Non-Financial Performance With a paucity of financial information available, researchers have looked for other measures of startup performance. Survival time is the most commonly studied startup performance metric despite the coarseness of the measure [47, 3, 20]. There are a few possible explanations for this. One explanation is that startups have such a high failure rate and long time to profitability that many won't ever report any other meaningful performance metrics [41].

C.2.3 Contextual Cues

Startups do not exist in isolation but are rather a product of their context. Investors must consider the performance of a startup's competitors, their local economy and the broader economy when evaluating the reasonableness of signals of startup potential.

Industry Performance Startups are involved in almost every industry. However, startups across industries have very different requirements, trajectories and measure their performance in different ways. Comparing startups across industries does not necessarily provide a clear view as to whether the potential of a firm is remarkable, likely errant, or within normal ranges. Accordingly, industry classification has been found to be a key determinant of startup investment [43, 14, 20].

Local Economy Headquarters location is a key indicator of startup investment success [6, 14, 20]. A clear example of this effect is Silicon Valley, a location known for producing an outsized number of successful startups. Silicon Valley provides a focal point for engineering talent, previously successful entrepreneurs, and venture capital firms. Therefore, we might expect different signs of startup potential for Silicon Valley startups compared to those in locations where development and traction are more difficult to attain.

Broader Economy Although startups are less affected by broader economic trends than larger, well-established companies economic challenges have a knock-on effect for startup investment. The Global Financial Crisis led to a 20% decrease in the average amount of funds raised by startups per funding

round, disproportionately affecting later-stage funding rounds. Therefore, when comparing startups of different ages, these sort of shocks have key implications for assessing what is a normal trajectory. This may explain why the year a startup is founded can influence startup investment [14, 25].

APPENDIX D

Database Schema

Relation	Attributes
Acquisitions	Acquiree Name, Acquiree Country, Acquiree State, Acquiree Region, Acquiree City, Acquirer Name, Acquirer Country, Acquirer State, Acquirer Region, Acquirer City, Acquisition Date, Acquisition Price, Acquisition Price Currency, Acquisition Price (USD), Acquiree CB URL, Acquirer CB URL, Acquiree UUID, Acquirer UUID, Acquisition UUID, Created Timestamp, Updated Timestamp
Category Groups	Category Group UUID, Category Name, Category Group List
Competitors	Entity UUID, Competitor UUID, Created Timestamp, Updated Timestamp
Customers	Entity UUID, Customer UUID, Created Timestamp, Updated Timestamp
Event Relationships	Event UUID, Entity UUID, Event Type, Relationship to Event (Type), Relationship to Event (Detail), Created Timestamp, Updated Timestamp
Events	Event UUID, Event Name, Short Description, Started Date, Ended Date, Registration Details, Registration URL, Start Time, End Time, Venue Name, Venue Address, Location UUID, Cost, Description, City, Region, Country, Continent, Permalink, CB URL, Logo URL, Profile Image URL, Event Roles, Created Timestamp, Updated Timestamp
Funding Rounds	Company Name, Country, State, Region, City, Company Category List, Funding Round Type, Funding Round Code, Announced Date, Raised Amount, Raised Amount Currency, Raised Amount (USD), Target Money Raised, Target Money Raised Currency, Target Money Raised (USD), Post Money Valuation, Post Money Valuation Currency, Post Money Valuation (USD), Investor Count, Investor Names, CB URL, Company UUID, Funding Round UUID, Created Timestamp, Updated Timestamp
Funds	Entity UUID, Fund UUID, Fund Name, Started Date, Announced Date, Raised Amount, Raised Amount Currency, Created Timestamp, Updated Timestamp
Investment Partners	Funding Round UUID, Investor UUID, Partner UUID
Investments	Funding Round UUID, Investor UUID, Is Lead Investor
Investors	Investor Name, Primary Role, Website Domain, Country, State, Region, City, Investor Type, Investment Count, Total Funding (USD), Founded Date, Closed Date, CB URL, Logo URL, Profile Image URL, Twitter URL, Facebook URL, Investor UUID, Updated Timestamp
IPOs	Company Name, Country, State, Region, City, Stock Exchange Symbol, Stock Symbol, IPO Date, Opening Share Price, Opening Share Price Currency, Opening Share Price (USD), CB URL, IPO UUID, Company UUID, IPO UUID, Created Timestamp, Updated Timestamp
Jobs	Person UUID, Organization UUID, Started Date, Ended Date, Is Current, Job Title, Job Role, Is Executive Role, Is Advisory Role
Organizational Parents	Organization UUID, Parent Organization UUID, Relationship to Parent, Created Timestamp, Updated Timestamp
Organization Descriptions	Organization UUID, Description
Organizations	Company Name, Primary Role, Website Domain, Website URL, Country, State, Region, City, Zipcode, Address, Operating Status, Short Description, Category List, Category Group List, Number of Funding Rounds, Funding Total (USD), Founded Date, First Funding Date, Last Funding Date, Closing Date, Employee Count, Company Email, Company Phone, Facebook URL, CB URL, Logo URL, Profile Image URL, Twitter URL, Organization UUID, Created Timestamp, Updated Timestamp
Patents	Assignee UUID, Patent Date, Citations by Patents, Citations of Patents, Patent Type
People	First Name, Last Name, Country, State, City, CB URL, Logo URL, Profile Image URL, Twitter URL, Facebook URL, Primary Affiliation Organization, Primay Affiliation Title, Primary Organization UUID, Gender, People UUID, Created Timestamp, Updated Timestamp
People Descriptions	People UUID, Description

Table D.1: Relational database schema.

APPENDIX E

Pipeline Hyper-parameters

Imputer	[Mean, Median, Most Frequent]
Transformer	[None, numpy.log1p, numpy.sqrt]
Scaler	
None	
StandardScaler	With Mean: True, With STD: True
RobustScaler	With Centering: True, With Scaling: True, Quantile Range: (25, 75)
MinMaxScaler	Feature Range: (0, 1)
Extractor	Function: PCA, Components: In Range (1, 100), Whiten: False, SVD Solver: Auto
Classifier	
Naive Bayes	
K-Nearest Neighbours	Neighbors: In Range (5,20), Weights: [Uniform, Distance], Algorithm: Auto, Leaf Size: 30, Metric: Minkowski, Distance: Euclidean
Logistic Regression	C: In Range (1e-3, 1e6), Penalty: [L1, L2], Solver: Liblinear, Fit Intercept: True, Intercept Scaling: True, Class Weight: Balanced, Tolerance: 1e-4
DecisionTree	Max Depth: In Range (5, 20), Criterion: [Gini, Entropy], Class Weight: Balanced, Splitter: Best, Max Features: None, Min Samples Split: 2, Min Samples Leaf: 1, Min Impurity Split: 1e-7
RandomForest	Estimators: In Range (10, 100), Max Depth: In Range (5, 20), Criterion: [Gini, Entropy], Class Weight: Balanced, Max Features: SQRT(Features), Min Samples Split: 2,Min Samples Leaf: 1,Min Impurity Split: 1e-7, Bootstrap: True
Support Vector Machine	C: [1e-5, 1e6], Probability: True, Class Weight: Balanced, Tolerance: 1e-3, Kernel: Linear, Poly: Degree: 3, Gamma: 1/Features, Coef0: 0, RBF: Gamma: 1/Features, Sigmoid: Gamma: 1/Features, Coef0: 0
Artificial Neural Network	Hidden Layers: 1, Hidden Layer Size: 100, Activation Function: [Identity, Logistic, Tanh, Relu], Alpha: [1e-3, 1e6], Solver: Adam Beta1: 0.9, Beta2: 0.999, Epsilon: 1e-8, Batch Size: min(200, Samples), Max Iterations: 200, Tolerance: 1e-4, Initial Learning Rate: 1e-3

Table E.1: Pipeline hyper-parameter search space.

APPENDIX F

Experimental Configuration

Base Configuration	
Dataset Slices	9 (3 x Forecast Window)
Forecast Window	3 [2, 3, 4]
Classification Pipeline	
Imputer	Most Frequent
Transformer	SQRT
Scaler	MinMaxScaler
Extractor	None, PCA
Classifier	Algorithm: Random Forest, Classes: Balanced, Criterion: Entropy, Bootstrap: True, Estimators: 34, Max Depth: 8, Max Features: SQRT(Features)
Experiment 1: Time Frame	
Developmental Stage	All (Combined)
Target Outcome	Extra Stage
Training Set Size	100%
Experiment 2: Developmental Stage	
Developmental Stage	All (Combined), All (Individual)
Target Outcome	Extra Stage
Training Set Size	100%
Experiment 3: Target Outcome	
Developmental Stage	All (Combined)
Target Outcome	Extra Stage, Extra Round, Exit, Acquisition, IPO
Training Set Size	100%
Experiment 4: Training Set Size	
Developmental Stage	All (Combined)
Target Outcome	Extra Stage
Training Set Size	1.0%, 3.2%, 10.0%, 31.7%, 100%

Table F.1: Experimental configuration.

APPENDIX G

Classification Reports

Slice Date		N	%	Accuracy	Precision	Recall	F1
2012	Positive	11,717	9.7	-	0.28	0.52	0.36
	Negative	109,312	90.3	-	0.94	0.86	0.90
	Avg/Total	121,029	-	0.82	0.88	0.82	0.85
2013	Positive	11,713	9.7	-	0.27	0.55	0.36
	Negative	109,117	90.3	-	0.95	0.84	0.89
	Avg/Total	120,830	-	0.81	0.88	0.81	0.84
2014	Positive	11,717	9.7	-	0.27	0.60	0.37
	Negative	109,312	90.3	-	0.95	0.82	0.88
	Avg/Total	121,029	-	0.80	0.88	0.80	0.83

Table G.1: Classification report by slice date.

Forecast Window		N	%	Accuracy	Precision	Recal	F1
2 Years	Positive	35,147	9.7	-	0.27	0.56	0.36
	Negative	327,741	90.3	-	0.95	0.84	0.89
	Avg/Total	362,888	-	0.81	0.88	0.81	0.84
3 Years	Positive	37,169	14.3	-	0.40	0.62	0.48
	Negative	222,540	85.7	-	0.93	0.84	0.88
	Avg/Total	259,709	-	0.81	0.85	0.81	0.83
4 Years	Positive	30,279	18.0	-	0.49	0.62	0.55
	Negative	137,646	82.0	-	0.91	0.86	0.88
	Avg/Total	167,925	-	0.82	0.84	0.82	0.82

Table G.2: Classification report by forecast window.

Developmental Stage		N	%	Accuracy	Precision	Recall	F1
Pre-Seed	Positive	16,038	13.0	-	0.39	0.54	0.45
	Negative	107,760	87.0	-	0.93	0.88	0.90
	Avg/Total	123,798	-	0.83	0.86	0.83	0.84
Seed	Positive	5,472	21.7	-	0.55	0.62	0.58
	Negative	19,692	78.3	-	0.89	0.86	0.87
	Avg/Total	25,164	-	0.81	0.82	0.81	0.81
Series A	Positive	3,534	41.0	-	0.72	0.72	0.72
	Negative	5,082	59.0	-	0.81	0.80	0.80
	Avg/Total	8,616	-	0.77	0.77	0.77	0.77
Series B	Positive	2,682	48.9	-	0.81	0.66	0.73
	Negative	2,802	51.1	-	0.73	0.85	0.79
	Avg/Total	5,484	-	0.76	0.77	0.76	0.76
Series C	Positive	1,620	55.5	-	0.85	0.77	0.81
	Negative	1,299	44.5	-	0.74	0.83	0.78
	Avg/Total	2,919	-	0.80	0.80	0.80	0.80
Series D+	Positive	933	48.0	-	0.91	0.73	0.81
	Negative	1,011	52.0	-	0.79	0.93	0.86
	Avg/Total	1,944	-	0.84	0.85	0.84	0.83

Table G.3: Classification report by developmental stage.

Target Outcome		N	%	Accuracy	Precision	Recall	F1
Extra Stage	Positive	30,279	18.0	-	0.50	0.61	0.55
	Negative	137,646	82.0	-	0.91	0.86	0.89
	Avg/Total	167,925	-	0.82	0.84	0.82	0.83
Extra Round	Positive	21,588	12.9	-	0.34	0.61	0.44
	Negative	146,337	87.1	-	0.93	0.83	0.88
	Avg/Total	167,925	-	0.80	0.86	0.80	0.82
Exit	Positive	12,372	17.4	-	0.46	0.70	0.56
	Negative	155,553	92.6	-	0.97	0.94	0.96
	Avg/Total	167,925	-	0.92	0.94	0.92	0.93
Acquisition	Positive	10,566	16.3	-	0.40	0.71	0.51
	Negative	157,359	93.7	-	0.98	0.93	0.95
	Avg/Total	167,925	-	0.91	0.94	0.91	0.93
IPO	Positive	2,052	1.2	-	0.28	0.57	0.37
	Negative	165,873	98.8	-	0.99	0.98	0.99
	Avg/Total	167,925	-	0.98	0.99	0.98	0.98

Table G.4: Classification report by target outcome.

APPENDIX H

Case Studies

	Company			
	ChaCha	Doctor.com	Fab	Mixpanel
Feature (2013-04-09)				
Age (Years)	7.4	0.6	4.3	3.8
Funding Raised ($m)	92.0	0.0	171.0	12.0
Funding Rounds (N)	8	0	8	4
Developmental Stage	Series D+	Pre-Seed	Series C	Series A
Predicted Outcome	✓	✗	✓	✓
Outcome (2017-04-04)				
Age (Years)	11.4	4.6	8.3	7.8
Funding Raised ($m)	96.0	5.0	336.0	77.0
Funding Rounds (N)	9	3	11	5
Developmental Stage	Series D+	Series A	Acquired	Series B
Actual Outcome	✗	✓	✓	✓
Correct Prediction	✗	✗	✓	✓

Table H.1: Company profiles and predictions.

1. ChaCha is an Indiana-based mobile Q&A service, launched in 2005. ChaCha has a long and convoluted investment history. It raised its Series A round in 2006 backed by Jeff Bezos of Amazon, before raising Series B-F rounds in 2007-10 to total funds of $92m. However, ChaCha took on additional rounds at lower valuations in 2011 and 2013. Our system predicted that ChaCha would raise funds or exit within the period of April 2013-2017. ChaCha did not take on any additional rounds and eventually closed in 2016. Our system did not predict this outcome accurately. As our publicly-sourced dataset has little information about valuations at funding rounds (valuation is considered more sensitive than quantum raised), our system has little ability to distinguish between succesful funding rounds and down-rounds (where valuation drops).

2. Doctor.com is a New York-based marketing automation platform for medical practices, launched in 2012. Doctor.com entered a three-year healthtech startup accelerator run by GE and StartUp Health in mid-2013. Our system did not predict that Doctor.com would raise funds or exit within the period of April 2013-2017. However, Doctor.com raised a $5m Series A round from Spring Mountain Capital in Feb 2017. This was a difficult prediction problem for our system. There was very little information about Doctor.com in 2013 and the Series A funding round came very late in the forecast window.

3. Fab is a New York-based e-commerce startup, launched in 2009. Fab raised $171m according to CrunchBase records, from reputable investors like Andreessen Horowitz, Mayfield Fund and First Round Capital, and once was reportedly valued at more than $1 billion. Our system predicted that Fab would raise funds or exit within the period of April 2013-2017. Later in 2013, Fab completed a Series D round for $150m. In 2015, Fab was acquired by PCH, reportedly for only a sum of ˜$20m. In this case, our system was technically accurate: Fab both raised funds and completed an exit. However, this exit was not a success for investors.

4. Mixpanel is a California-based consumer analytics platform, launched in 2009. Mixpanel came out of famed startup accelerator Y-Combinator and raised $12m from Seed - Series A rounds up to 2012, from reputable Venture Capital (VC) firms and angels like Sequoia Capital, Andreessen Horowitz and Max Levchin. Our system predicted that Mixpanel would raise funds or exit within the period of April 2013-2017. Mixpanel went on to raise a $65m Series B round from Andreessen Horowitz in December 2014 that valued the ccompany at $865m. However, during 2016, MixPanel cut 20 staff (primarily in sales) as it restructures towards higher profitability. This was a good prediction by our system over this forecast window but MixPanel's long term outlook is unclear.