NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

Predicting Start-up Success with Machine Learning

Francisco Ramadas da Silva Ribeiro Bento (M2013022)

Dissertation presented as partial requirement for obtaining the Master's degree in Statistics and Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

2017

Predicting Start-up Success with Machine Learning

Francisco Ramadas da Silva Ribeiro Bento (M2013022)

MGI

NOVA Information Management School

Universidade Nova de Lisboa

Predicting Start-up Success with Machine Learning

By

Francisco Ramadas da Silva Ribeiro Bento (M2013022)

Dissertation presented as partial requirement for obtaining the Master's degree in **Information Management**, with a specialization in **Business Intelligence**

Advisors / Co Advisors: Professor Doutor Roberto Henriques / Mestre João Ferreira Loff

November 2017

ACKNOWLEDGEMENTS

I would like to dedicate this work to my family and friends who always believed in me and gave me strength to finish this project in times where time seemed unavailable.

To my advisor Professor Roberto Henriques for the insights, reviewing my work and letting me present this study. I was not easy I know. Thank you.

To João, who co-advised my work. A friend and co-founder of the most challenging professional experience of my life. For the technical support and long hours dedicated into understanding my stubborn view of this study. Thank you.

To my mother and father who always pushed me into finishing what I start. It would not be possible without you. Thank you.

To my friend Guilherme, with whom I share not only one of the greatest friendships but an amazing professional adventure which challenges us every day. Thank you for the patience, motivation and the long hours working alone while I wasn't available. I look forward to returning the effort with interest.

To Constança who, in times of pressure, always pushed me by giving motivation to conciliate all my duties. You rock. Thank you.

To Gonçalo for all the discussion – you made me challenge everything. I firmly believe in your potential. Your curiosity, knowledge and intrinsic view of technical problems will take you as far as you want. Thank you.

ABSTRACT

Start-ups are becoming the motor that moves our economy. Google, Apple, or more recently Airbnb and Uber are companies with tremendous impact in worldwide economy, social interactions and government. Over the past decade, both in the US and Europe, there has been an exponential growth in start-up formation. Thus, it seems a relevant challenge understanding what makes this type of high-risk ventures successful and as such, attractive to investors and entrepreneurs. *Success* for a start-up is defined here as the event that gives a large sum of money to the company's founders, investors and early employees, specifically through a process of M&A (Merger and Acquisition) or an IPO (Initial Public Offering). The ability to predict *success* is an invaluable competitive advantage for venture capitals on the hunt for investments since first-rate targets are those who have the potential for growing rapidly soon, which ultimately, allows investors to be one step ahead of competition.

We explored the world's largest structured database for start-ups – provided by the website CrunchBase.com, with the objective of building a predictive model, through supervised learning, to accurately classify which start-ups are successful and which aren't. Most of the studies regarding the prediction of processes of M&A or an alternative definition of a company's success tend to focus on traditional management metrics provided by financial reports and thus using a low number of observations compared with the present study. As technologies of information evolve it became possible to achieve highly reliable results in data analysis by manipulating it with complex machine learning algorithms or data mining techniques to define features and characterize robust models.

Further developments on previous studies such as the development of new features and a new definition for the target variable were applied. Using Random Forests on our dataset, a general model (as including all categorical features) achieved a True Positive Rate (TPR) of 94%, which is the highest recorded with this data source, and a False Positive Rate (FPR) of 8%. The author also generated models per each category of a company to provide results comparable with previous studies the values achieved ranged between 61% and 96% compared with 44% and 80%. As a novelty, models for each of the five geographical regions selected (all from USA) are provided, with TPRs ranging between 90% and 96%. The new features, focused on the impact of venture capital in a company, proved pivotal to the overall performance of the models by being some of the most important to the final models showing the critical importance this type of investment has on these ventures.

Keywords

Start-up, Mergers and Acquisitions (M&A), IPO, data analysis, machine learning, venture capital, true positive rate (TPR), false positive rate (FPR).

INDEX

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

"A start-up can be defined as a human institution created to develop new products and/or services under extreme uncertainty conditions."

(Ries, 2011)

Start-ups are booming everywhere as more colleges, governments and private companies invest and stimulate people to pursue their ideas throughout these ventures. Companies are raising millions with ease and achieving *unicorn status* (i.e., a one-billion-dollar valuation) in a matter of years. Slack, a messaging app, achieved it after operating for 1.25 years (Kim, 2015). Examples like Uber and Airbnb are changing societies in such impactful ways that regulation had to be created to keep pace with a new reality. Start-ups are having such impact that, ultimately it becomes every investor's ambition to be part of a large acquisition such as Facebook acquiring WhatsApp (another messaging app) for nineteen billion dollars which allowed Sequoia (a Venture Capital fund) to have a 50x return on investment (Neal, 2014). But there is a catch, start-ups are companies with an estimated 90% probability of failure, which means a lot of investments without proper returns (Patel, 2015).

Predicting the success of a start-up is commonly defined as two-way strategy that makes a large amount of money to its founders, investors and first employees, as a company can either have an IPO (Initial Public Offering) by going to a public stock market (i.e. Facebook *going public*, allowing everyone to invest in the company by buying shares being sold by its insiders in the U.S stock market) or, be acquired by or merged (M&A) with another company (i.e. Microsoft acquiring LinkedIn for $26B) where those who have previously invested receive immediate cash in return for their shares. This process is often denominated as an exit strategy (Guo, Lou, & Pérez-Castrillo, 2015). This study will therefore, consider both an IPO (Initial Public Offering) and a process of M&A (Mergers & Acquisitions) as the critical events that classify a start-up as successful.

With a focus on how a start-up or an investor could explore all this knowledge for a better decision making in investment strategy and monetary gain, the study intends, by applying data mining and machine learning techniques, to create a predictive model that has as the dependent variable a label to classify whether a start-up is (already) successful or not.

Improved areas of our society are already being improved by the application of machine learning. From healthcare, where by applying segmentation and predictive modelling it is possible to identify different types of treatment (from preventive to life-style changes) for a patient or even diagnose him (Raghupathi & Raghupathi, 2014), to marketing personalization where companies benefit from knowing as much as possible from their clients to create customer-centric experiences all around. Fraud detection, financial services, insurance and even smart cars are all industries creating value, in a short to medium term, through the application of machine learning (Marr, 2016). It is possible to bring similar advantages to investors in start-ups, by giving these players information about which start-ups are closer to a successful event in their near future they can better choose where to put their chips and have higher returns on their investments.

To generate the predictive model, three supervised machine learning algorithms were tested: Support Vector Machines, Logistic Regression and Random Forests. All these algorithms fit the characteristics of the dataset (147 features and more than 140 000 observations), provide a fast and

simple technical implementation. The creation of a predictive model to explain this specific phenomenon is an excellent indicator of how the level of exploitation of Data Mining techniques allows analysts to extract the full potential of the available data to reach all proposed goals. Being able to accurately classify if a start-up had this event in its progress is not only incredibly valuable for all the players in the start-up world (entrepreneurs, angels and investors of Venture Capital) but also, the application of different techniques and features to build models with higher predictive accuracy represents a step forward to not only the academic literature but also the industry.

Although there are a lot of studies about predicting processes of M&A, most focus on financial and managerial features with Logistic Regression being the most common predictive algorithm used (Ali-Yrkkö, Hyytinen, & Pajarinen, 2005; Altman, 1968; Gugler & Konrad, 2002; Karels & Prakash, 1987; Meador, Church, & Rayburn, 1996; Ragothaman, Naik, & Ramakrishnan, 2003). There is still space for an approach focused on venture capital (or other type of investment) features and different machine learning algorithms to company acquisition and with a platform as rich as CrunchBase it is an interesting challenge to explore and compare achieved results with previous approaches (Liang & Daphne Yuan, 2012; Xiang et al., 2012).

Considering the improvements achieved with the current approach, from 61% to 96% compared with 44% to 80% for different company's categories and an overall TPR (True Positive Rate) of 94%, it is important to reinforce the advancements achieved in this study.

The following dissertation is divided in three sections: first section explores the study relevance and its importance, the objectives, a literary review of the thematic including previous studies of company acquisition and an overview of baseline articles. Secondly, the process to generate a final dataset from CrunchBase data. This includes pre-processing, creation of new variables, problems faced and its solutions. Finally, the application of machine learning algorithms to generate the proposed predictive model through supervised learning – the experiment setup, its results and final conclusions.

1.1. OBJECTIVES

The present work has as the main objective, the development of a predictive model to classify a start-up/company as successful or not (binary classification).

The most recent works, such as *A Supervised Approach to Predict Company Acquisition with Factual and Topic Features Using Profiles and News Article,* on predictive analysis (using the same dataset) on start-up success rate shows there is room for improvement. Previous studies tend to focus primarily on managerial features and often overview the impact of financial features related with funding (specially from Venture Capital funds). It is intended to bridge this gap by creating funding-oriented features with good predictive impact in classifying successful companies. Additionally, there is room to improve the quality of the sample by being more selective with companies or by better treating the amount of sparse data which is characteristic of this dataset.

The present dissertation will also test different machine learning algorithms for its learning task in the generation of said predictive model (Xiang, Zheng, Wen, Hong, & Rose, 2012).

1.1.1. Technical Objectives

During the process, the author expects to achieve several technical objectives:

During a first phase of Data Analysis, a full **understanding of the CrunchBase database** is expected, followed by the process of **Data cleaning** (missing values, duplicates, redundant data). Having a full database ready to be filtered it fundamental to define the **scope of data to be used** in the model and to be able to do an **explorative analysis** of key features. **Transformation of data** will be made by defining and creating new features which will generate the final dataset to be used in the learning task.

Followed by a second phase consisting on the Experiment Setup and its Results, where the **experiment** will be **set up** by applying different machine learning algorithms to generate the best possible model through supervised learning to try to outperform current state of art. The algorithms tested are Logistic Regressions (LR), Support Vector Machines (SVM), and Random Forests (RF). **Experiment results and conclusions** will be presented and discussed.

2. LITERATURE REVIEW

The literature review will identify major themes related to the subject of this study. Also, similarities and differences in previous studies will be considered to enrich the project's content to make it unique and innovative for the start-up scene.

2.1. START-UP ECOSYSTEM

2.1.1. Start-up definition & Growing Importance

Start-ups are companies that make products that venture to an area or market in ways that haven't been done before. This makes start-ups risky and unpredictable as a new product or service may not work among its apparent users and may require constant adjustments before it gets product/market fit. Ultimately, a start-up is a high-risk company that is in the first stage of operations and commonly related to technology as a product or a service (Ries, 2011).

These ventures are often initially bootstrapped by their entrepreneurial founders as they attempt to capitalize on developing a product or service. Due to limited revenue or high scalability costs, most of these small-scale operations are not sustainable in the long term without additional funding from venture capitalists (in opposition to getting a bank loan). In the late 1990s, the most common type of start-up was a known as "dotcom". As access to the internet expanded and computers took on an increasingly important part in people's daily lives, venture capital became extremely easy to obtain during that time due to an excitement among investors speculating on the emergence of new types of businesses with a market penetration never seen before. Unfortunately, between 1997 and 2001, in a crisis known as the "dotcom bubble", most of these technological start-ups went bankrupt due to major oversights in their business plans, such as a lack of a sustainable revenue (Geier, 2015).

Founder of PayPal, Chairman of Palantir and serial entrepreneur Peter Thiel, defines a start-up as a creator of vertical innovation and not horizontal. Being vertical innovation the technology that hasn't been created before and horizontal innovation the process of globalization, bringing existing technology to places that don't have it yet. Thiel is also a firm believer that a start-up must aim to create a monopoly in a niche market and only then expand to new markets (Thiel & Masters, 2014). Thiel contributes with the most extreme point-of-view on the definition of a start-up, clearly focused on the characteristics of technological ventures. This is demonstrated by his views of exponential growth and market positioning (as a monopoly ruler) only possible in this field and at an early stage.

Y Combinator founder Paul Graham puts it more simply in his essay: "A start-up is a company designed to grow fast". Also, contrary to technology mogul, Peter Thiel, Graham doesn't believe technology is essential to start-ups. To him, only companies with fast growth matters. Quoting, "Being newly founded does not in itself make a company a start-up. Nor is it necessary for a start-up to work on technology, or take venture funding, or have some sort of exit. The only essential thing is growth" (Graham, 2012). By explicitly including non-technological companies as start-ups, Graham gives the more realistic definition to today's businesses as even small ventures can have cash flow positive operations in short term without having to raise debt allowing them to focus on expanding as fast as possible and thus being deemed start-ups.

Steve Blank, author of Four Steps to the Epiphany, although with a similar definition as Graham, defines it differently adding an important notion of scalability: a start-up is an organization aiming for a repeatable and scalable business model for a limited period. Once a start-up finds its model, it ceases to be a start-up (Blank, 2006).

It is also important to understand the difference between start-ups and traditional small businesses, according to "The Global Start-up Ecosystem Ranking 2015" developed by Compass.co (formerly Start-up Genome) in partnership with CrunchBase (the source of data used in the present work), a traditional small business has the odds of financial succeeding for the first two years of around 75%, inversely, a start-up has a 75% chance of failing. Nonetheless, an auto-shop or a laundry will hardly reach a Fortune 500 market capitalization but there are hundreds of start-ups in that league (Hermann, Gauthier, Holtschke, Bermann, & Marmer, 2015).

It is a *game* of higher risk, higher returns. Knowing the high risk and huge percentage of start-ups that fail, but also the exponential growth in start-up formation in US and Europe as well as its growing importance in national economies, it seems a valuable challenge to quantitatively study a phenomenon that is challenging so many people around the globe (Hermann et al., 2015; Williams, 2015).

Neil Patel (2015) from Forbes reports that nine out of ten start-ups will fail with most common factors for it, being first, the lack of market need for a specific product or service and secondly, companies that run out of capital (Griffith, 2014).

So, why are we seeing start-ups everywhere? Even countries are promoting its creation. See Portugal, a country with a massive debt (Eurostat, 2016) launched a program with fifteen different incentives for both investors and entrepreneurs in 2016 to further develop the ecosystem (Matias, 2016).

Focused on technological start-ups Steve Blank, proposes four reasons for the phenomenon in his book "The Four Steps to the Epiphany" (Blank, 2006):

- Start-ups can now be built for thousands rather than millions: With a decrease in cost of product development by a factor of 10 over the last decade (Hermann et al., 2015), it is now cheaper than ever to build technology. Access to tools, open-source code, cheaper servers, and an ever-growing community of developers contributing to the dissemination of technology around the globe allows everyone to build, test and share its products. The highest representation of this fact is WhatsApp, which was bought for more than $19 billion dollars and had sixteen employees (Neal, 2014).

- A higher resolute venture capital industry: When Venture Capital (VC)[1] were required to spend millions of dollars on an investment, they had to make a small number of big bets. However, with the cost of technology being less expensive every year it has created an opportunity for other types of investors: angels, accelerators and micro-VCs. These entities, with smaller checks can make a whole lot of small bets and help a larger number of start-ups. This lifeline for small start-ups allows them to not look for additional outside funding until later stages of development.

[1] Financial entities which make investments in venture businesses (start-ups - high risk, high returns)

- Entrepreneurship developing its own management science: When the first wave of Information Era venture-backed software companies began in the 1970's, many entrepreneurs applied its knowledge of Management Science created by Henry Ford and his peers. However, especially after the huge dotcom bubble burst in the final years of the nineties, many entrepreneurs began to realize start-ups were a different reality with a different rule set. Forty years after the beginning of the modern start-up era, Steve Blank with "The Four Steps to the Epiphany" and Eric Ries with "The Lean Start-up" laid the foundation for a new management science for start-ups, which has come to be known as the Lean Start-up Movement. Overtime "entrepreneurs have become significantly better at creating start-ups." (Hermann et al., 2015)

- Speed of consumer adoption of new technology: As internet became universally accessible, start-ups could be - from day one - what Steve Blank calls, a "micro-multinational" and people from all over the world can access products from the opposite end of the planet without any inconvenience (Blank, 2006). Google and Facebook prove that location is, probably, meaningless. Even the business conjuncture changed as big companies are now willing to try cheaper, faster and more elegant technologies from emerging start-ups. For example, Slack, the fastest ever company to achieve a billion-dollar valuation (becoming a *unicorn* in 1.25 years) is a three-year-old start-up sensation who managed to get customers like Airbnb, Buzzfeed, eBay, Expedia, NASA and Salesforce through very cheap software and a refined product ("Slack: Customer Stories," 2017).

Not only has "the ease of global access to users and customers (…) and the increasing speed of technological adoption by consumers and businesses enabled start-ups to grow at a significantly faster rate" but also the access to up-to-date data and data-mining techniques gave entrepreneurs access to more knowledge than ever, avoiding mistakes of the past and correctly assessing what are the fundamental features (KPIs) for their companies. For an investor, a more data-driven decision process (as supported by data-mining and machine learning) lowers risks, which in the end represents higher returns on investments (Hermann et al., 2015).

2.1.2. Success for Start-ups: IPOs and M&As

The success of a start-up is commonly defined as a two-way strategy as a company can either have an IPO (Public Initial Offering) by going to a public stock market, allowing its shareholders to sell shares to the public, or be acquired or merged (M&A) with another company where those who have previously invested receive immediate cash in return for their shares. This process is often designated as an exit strategy (Guo, Lou, & Pérez-Castrillo, 2015).

Mergers and acquisitions are usually referred to as M&As and play an important part of corporate restructuring. According to Alam & Khan (2014), a merger is the strategy of joining two companies to form one single company (usually under a new name) to increase the profit and sales level and is, in non-tech companies, more frequent between entities of the same size and stature. M&A activities are especially critical for high-tech industries, because they often use M&As to acquire state-of-the-art technologies or rapidly expand their R&D capabilities (Wei, Jiang, & Yang, 2009). "An acquisition refers to a situation where one firm acquires another and the latter ceases to exist. An acquisition occurs when one company takes controlling interest in another firm (…) A firm that attempts to acquire or merge with another company is called an acquiring company" (Machiraju, 2003).

The rationale behind a process of M&A is that two companies are of more value together than as separate entities. This consolidation of two companies is a critical corporate strategy for companies to preserve their competitive advantages (Machiraju, 2003; Xiang et al., 2012).The Thomson Reuters report, shows 2015 as the biggest year ever in worldwide M&A deals with a $4.7 trillion in total business. A 42% percent rise from 2014, beating the former record of $4.4 trillion in 2007 (Rogers, 2016). The understanding of mergers and acquisitions is of great importance in today's world where newspapers tell stories of such taking place around the globe. (Alam & Khan, 2014) Ultimately, an M&A prediction can help start-ups assess their possibility of being acquired or merged and who are the possible bidder companies (Wei et al., 2009). Using CrunchBase data to corroborate this trend, there were 4 589 acquisitions in 2015 and 7 899 in 2016 which represents a 72% increase.

According to Li & Liu (2010), "An IPO is the first sale of stock by a private company to the public. Therefore, 'going public' is an important event over the life cycle of a company. In the post-IPO stages, the companies will be evolved into continued growth as a healthy company, get acquired before a strong performance or weak performance, and be delisted from the stock market at the end of its life cycle.".

When an IPO occurs, the venture obtains a stock market listing enabling the company to receive additional financing and allowing insiders to eventually sell their shares to the public.

There is no optimal exit strategy for a company as it heavily depends on multiple factors, such as the profitability of the company, the financial market conditions, the trade of information between insiders, the benchmark of other companies' IPO, among many others (Akerlof, Yellen, & Katz, 1970).

In the start-up ecosystem either one of these events is usually considered a success for the company, being acquired or *going public*, as it brings (large) amounts of immediate money to its founders, investors and early employees (Guo et al., 2015). One of the most frequent reasons for start-ups/companies to acquire smaller companies is to *buy* its talent pool. Not only the parent company is buying technology but also hiring its employees. This type of acquisition is commonly called,

acquihiring and provides a fast strategy to grow in competitive markets (Marita Makinen, Haber, & Raymundo of Lowenstein Sandler, 2014).

2.2. DATA ANALYSIS

"We are drowning in information and starving for knowledge."

– Rutherford D. Roger

2.2.1. Data Mining

We are currently living in a society where all our business, scientific and government transactions are computerized but also in a world where digital devices, social media and bar codes are generating data. *Data scientists* have been facing a challenge to rapidly increase our ability to generate and collect data through new techniques and automated tools, aiming to transform the ever-growing databases into useful information and most importantly, knowledge (Han & Kamber, 2006; Kantardzic, 2003).

Ian Witten and Eibe Frank define Data Mining as the process of extracting implicit and previously unknown information with potential use from a dataset (Witten, Frank, & Eibe, 2000). By building programs that look through databases, there is the potential to find strong patterns which, if found, will be able to generalizable complex problems and make accurate predictions on future data. Witten and Frank provide an example, *the weather problem,* to illustrate how by using only a set of four features – *outlook, temperature, humidity* and *windy,* one can find a pattern and predict if there are conditions to play outside. Through a simple set of rules, they can accurately classify an observation as a place with conditions to play outside or not (Witten et al., 2000). Machine learning provides the technical basis for data mining. It is used to extract information from the raw data in databases. The process of discovering patterns in data must be automatic or semiautomatic (which happens more frequently), and the discoveries must be "meaningful" in that they lead to some advantage. Since both terms are frequently associated, It is also important to understand machine learning as the mathematical algorithms used to create models and Data Mining as the entire process of knowledge extraction (which may or may not have machine learning techniques in its process) (Witten et al., 2000).

Berry & Linoff have a more business-centric definition, defining data mining a collection of technological tools and techniques required to support companies by providing useful knowledge. Their rational revolves around the notion that companies need to make decisions based on data (informed decisions) as opposed to assumption-based ones (uninformed decisions) and that companies need to measure all results which will always be beneficial to the business (Berry & Linoff, 2004). Christopher Clifton, with a similar definition, considers data mining as an interdisciplinary subfield of computer science with the overall goal of extracting information from large volumes of data, discovering patterns and transforming it into understandable knowledge.

Data mining is widely used in business, scientific research and even government security, since it combines methods from machine learning and statistics with database management to analyze data. Traditionally, data mining and the knowledge extraction were performed manually, however, the dissemination and increasing power of computer technology has dramatically increased data collection, storage, and manipulation ability. As datasets have grown in size and complexity, direct "hands-on" data analysis has increasingly been augmented with indirect, automated data processing, aided by newest discoveries in computer science, such as neural networks, cluster analysis, genetic

algorithms (1950s), decision trees and decision rules (1960s), and support vector machines (1990s) (Christopher Clifton, 2009; Kantardzic, 2003).

To make sense of data and aiming to address the problem of data overload, data scientists came up with a process concerned with the development of methods and techniques to standardize the application of Data Mining – Knowledge Discovery in Databases (or, in more recent approaches, Data). It is defined by Fayyad, Piatetsky-Shapiro and Smyth as the application of specific data-mining methods for pattern discovery and knowledge extraction. Jiawei Han and Micheline Kamber added, more recently, the notion that this data can be provided by different sources such as multiple databases, data warehouses, web or any data stream. The original definition of a KDD process is a 5-step framework that every Data Mining problem should follow: (1) Selection, data into target data; (2) Pre-processing, target data into processed data; (3) Transformation, processed data into transformed data; (4) Data Mining, transformed data into patterns[2]; (5) Interpretation/Evaluation; interpretation of patterns into knowledge (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).

While this definition is considered the standard for KDD, Jiawei Han and Micheline Kamber propose a more modern approach: (1) Data cleaning, removing noise, outliers, missing values; (2) Data integration, combining different data sources; (3) Data selection, retrieving relevant data from the database; (4) Data transformation, data is transformed as new features are created; (5) Data mining, mathematical algorithms are applied to extract meaningful patterns; (6) Evaluating results; (7) Knowledge presentation, where visualization and knowledge representation techniques are used to present results (Han & Kamber, 2006).

Applications of data-mining can be seen in healthcare as data mining is becoming increasingly essential in this field. Evaluating **treatment effectiveness** by comparing causes, symptoms and courses of treatment to the outcomes of patient groups treated with different drug regimens for the same disease allows to determine which treatments work best and are most cost-effective for each group (Kudyba, 2014). Also, to aid **healthcare management**, data mining applications can be developed to better identify chronic disease states and high-risk patients, design appropriate interventions, and reduce the number of hospital admissions and claims (Chye Koh & Tan, 2011). Other applications of data mining in healthcare are detection of fraud, customer relationship management and, even, predictive medicine.

Marketing also attracts a lot of development in this field. The most common application of data mining in marketing is through segmentation, which by analyzing customer databases allows the definition of different customer groups and even forecast their behavior. The amount of data gathered has so much potential that one time, Target (a US retailer), segmented a young woman as pregnant even before the father knew about the pregnancy (Hill, 2012). Another marketing application of data mining is through market-basket analysis systems, which find patterns in customers consumption habits (Fayyad et al., 1996). This allows a better management of stock, and distribution of shelve space in supermarkets.

[2] Data Mining is a step in the KDD process that consists on applying data analysis and discovery algorithms to produce patterns (or models) over transformed data. Classification (as in the present study), regression or clustering are examples of common data analysis. The data-mining component of the KDD process often involves repeated iterative application of data-mining methods.

2.2.2. Machine learning

Over the last 50 years, machine learning evolved from the efforts of scientists like Arthur L. Samuel exploring whether machines could learn to play games like checkers (Samuel, 1962) to a broad discipline taught in scientific schools all over the world and to be applied in all our interactions with technology. With computational power rapidly increasing over the past few decades it became possible to use these techniques in more practical ways than before. Using technologies like regressions and support vector machines, Google created PageRank, Google News and even Gmail spam classifier in its way to become one of the most powerful companies in the world. These algorithms became easy to distribute making new applications that rely on these techniques, more and more common (Beyer, 2015).

Kirk Borne, Principal Data Scientist at Booz Allen, clearly defines "machine learning as the basis set of mathematical algorithms that learn the models that describe the patterns and features in data" and "data mining as the application of those algorithms to make discoveries from large data sets" ("Artificial Intelligence and Machine Learning: Top 100 Influencers and Brands," 2016; Onalytica, 2016).

Tom M. Mitchell, Department Head of machine learning at Carnegie Mellon University in his "The discipline of machine learning", starts his exploration on the thematic by defining the question the field of machine learning seeks to answer:

"How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?"

The answer is broad as machine learning covers learning tasks ranging from autonomous robots, to the data mining of consumer records to predict their behavior, to search engines that automatically learn its users' preferences but idea is: machine learning, a natural outgrowth of the intersection of Computer Science and Statistics, is the ability to make a machine learn something through experience (data) and original settings (algorithms and its parameters) (Mitchell, 2006).

Rob Schapire, formerly the Professor of computer science at Princeton University and currently at Microsoft, defines ML very simply as: "machine learning studies computer algorithms for learning to do stuff". Machine learning is the capacity of telling a computer how to complete a task, make accurate predictions or even learn on how to act properly upon a determined scenario. It always starts with previous observed data and a set of instructions on how to analyze it. "So in general, machine learning is about learning to do better in the future based on what was experienced in the past", Rob Schapire adds (Schapire, 2008).

We live in a world where machine learning applications are present in (almost) every sector of our daily lives:

- Banks and other businesses in the financial industry who use it primarily to identify investment opportunities, or help investors know when to trade. Using data mining can also identify clients with high-risk profiles, or pinpoint warning signs of fraud (Schapire, 2008).
- Websites promoting items you might like based on previous purchases or searches based on our previous behavior. More recently, Text Mining is also being used to compare a user review score with his review text (Loff, 2016).

- Uber, Lyft and other car sharing services use these algorithms to make routes more efficient or to predict potential problems to increase profitability. Even self-driving cars need machine learning to predict accidents or optimize routes ("10 Million Self-Driving Cars Will Be On The Road By 2020 - Business Insider," 2016; NGUYEN, 2015).
- The health industry uses it as tool to help medical teams carry out pattern recognition of damaged tissue (structural health monitoring) to correctly diagnose patients. And more recently, wearable devices use sensors to monitor people's health in real time (Farrar & Worden, 2012).

Machine learning can be divided in four different categories: supervised, unsupervised, semi supervised and reinforcement learning. Being supervised and unsupervised learning the most widely used.

Supervised learning algorithms make predictions based on a set of examples. A **supervised learning algorithm** is, having *x* input variables and an output variable *y*. The algorithm learns to map the function (*y*=f(x)) and can (correctly) predict/classify any new output *y* after getting new input data *x*. The possible answers from the output are known. All data is labelled, and the algorithms learn to predict the output from the input data. Supervised algorithms can be grouped into regression and classification problems: A **regression** function is a type of model when the output variable is a real value, i.e., 88, 130, 0%. A **classification** function generates models where the output is a category, i.e., "red"/ "blue", "acquired"/ "not acquired".

Unsupervised learning algorithm is when we only have input variables/features and no output (target variable). It is in the learning process that the algorithm will discover and classify possible outcomes. Here, we don't know the possible answers. As all data is unlabeled, the algorithm should learn to create patterns from the input data. Typically, unsupervised learning can be grouped into clustering and association analysis. A **clustering** problem is the discovery of groups with heterogeneous characteristics between them and homogeneous characteristics between the observations of each group. A frequent application of cluster techniques is in the segmentation of clients for a company (marketing). An **association** rule problem is when you want to discover n rules that describe large portions of data, such as people that acquire A also tend to buy B (usually used in supermarket chains) (Aggarwal, 2015; Berry & Linoff, 2004; Han & Kamber, 2006; Kantardzic, 2003; Mitchell, 2006;).

Frequently mistaken with machine learning, Data Mining is the set of different techniques to produce knowledge from data. It can involve statistical inference and machine learning algorithms to identify patterns in large datasets. Machine learning on the other hand is the specific set of mathematical algorithms running through computers to understand the structure of data being analyzed (Christopher Clifton, 2009). Machine learning can be defined as the set of methods and techniques used to discover patterns in data, it is a step in a broader discipline which is Data mining. An in-depth exploration of the topic is present in 2.2.1. Data Mining.

Being machine learning the ability to make computers *learn* through past information to provide present or future context, it is natural to see the potential for company acquisition studies using these techniques. We now have an immense historic information regarding acquisitions, IPOs, investment and others, that should be explored. Both supervised and unsupervised learning techniques can provide value in this field. For example, through a regression or a classification

problem it is possible to predict success (as in the present study), while through segmentation (unsupervised learning), one would be able to differentiate companies automatically and in ways not always obvious. The possibilities are endless, and it is up those working in this field to provide the most value from the available data.

2.3. PREVIOUS RESEARCH ON ACQUISITION PREDICTION

Most research focused on predictions by analyzing common quantitative financial variables for corporate companies as firm size, market to book value ratio, cash flow, debt to equity ratio and price to earnings ratio (Ali-Yrkkö, Hyytinen, & Pajarinen, 2005; Gugler & Konrad, 2002; Meador, Church, & Rayburn, 1996). With some adding managerial features as industry variations (Meador et al., 1996), management inefficiency (Ali-Yrkkö et al., 2005; Meador et al., 1996) and resource richness (Meador et al., 1996). Most of the analysis methods used to build M&A prediction models have been Logistic Regressions (or Multinomial Logistic Regressions) (Ali-Yrkkö et al., 2005; Gugler & Konrad, 2002; Meador et al., 1996; Ragothaman, Naik, & Ramakrishnan, 2003).

Hyytinen and Ali-Yrkko reported "how multinomial logic estimations show that if a Finnish firm owns a number of patents registered via the European Patent Office (EPO), the patents increase the probability that the firm is acquired by a foreign firm.". The authors took under consideration other variables for their model as firm size, cash flow ratio to total assets and ROI (return on investment) to simulate managerial performance (Ali-Yrkkö et al., 2005). A relevant finding in Hyytinen and Ali-Yrkko's work is that size (as a logarithmic function of total assets owned) matters. The larger the firm, the more likely it is acquired. However, their sample of 815 Finnish companies is too small to test with more powerful techniques.

Wei, et al., also studied the importance of patents a company has and its importance supporting Merger and Acquisitions prediction. Through a Naïve Bayes model to classify a company as whether the candidate target company would be acquired or merged by the bidder company or not, they defined a set of features such as number of patents granted to a company, number and impact of recent patents and the company's technological quantity. Their results, with a set of 2394 acquisitions, vary between a precision rate of 42.93% and 46.43% (Wei et al., 2009). Although making a relevant step in predicting M&A's by including technological variables they limited their results by excluding all other categories such as management and financial features.

ACTARGET is a tool to classify firms into acquisition and non-acquisition target categories and uses discriminant analysis and rule induction in its model. They developed the tool with a database of 97 acquired and 97 non-acquired firms, achieving 81.3% of the acquisitions and 65.6% non-acquisition companies as correctly classified (Ragothaman et al., 2003). Although promising, the small dataset and the use of only eight financial features limit their results.

There has also been a large focus on studies about business failures and bankruptcies over the last fifty years (Xiang et al., 2012). Professor Edward Altman, best known for the development of the (Altman) Z-score, proposed several financial ratios as the features of a multivariate statistical analysis in his study to predict bankruptcy. Altman extended his first study into the prediction of railroad bankruptcies in America by using a set of 21 railroads that went bankrupt between the years 1939-1970. Specifically, Altman with a five-variable model using multiple discriminant analysis, analyzed ratios like, common liquidity measures, solvency and leverage measures, and profitability measures plus efficiency indicators with a very accurate classification at one and two years prior to bankruptcy (achieving an accuracy of 97.7%) (Altman, 1968; Zhang & Zhou, 2004).

More recently Ravisankar et al., used six machine learning algorithms, Multilayer Feed Forward Neural Network (MLFF), Support Vector Machines (SVM), Genetic Programming (GP), Group Method

of Data Handling (GMDH), Logistic Regression (LR), and Probabilistic Neural Network (PNN) to understand the differences between a set of 202 companies listed in various Chinese stock markets, using 35 financial features. The dataset consisted of 101 non-fraudulent companies and 101 that were. Their Probabilistic Neural Network outperformed all other classifiers with a True Positive Rate of 98.09% predicting which companies were fraudulent (Ravisankar et al., 2011). Their numbers are impressive but the use of a small sample of 202 companies and a lack of exploratory analysis of the features used allows the assumption that significant differences between fraudulent and non-fraudulent companies exist and would be "easily" distinguished in their learning task. Their approach has the highest results analyzed but the scope of their investigation is not specifically company acquisition but more oriented to fraud prevention.

Investments behavior of venture capital firms and other investors in start-ups is also a subject of study. Liang & Daphne Yuan (2012) used the CrunchBase dataset to predict investor behavior using social network features and a supervised learning approach. They modelled the investment behavior through a classic link problem as they compare every pair of *Investor* and *Company* to predict if the *Investor* will invest in a *Company* based on how similar or different in terms of their social relationship. As of May 2012, their dataset comprised 89'370 companies and 28'108 investment rounds. Using Decision Trees as their learning algorithm, they achieved a TPR (True Positive Rate) of 87.53% with an AUC (Area Under Curve) of 0.77%. Although not directly predicting acquisitions their study still signals successful companies (Liang & Daphne Yuan, 2012) .

Using the same dataset but with a focus on start-up acquisition and investments from venture capital, Xiang et al. (2012), predicts company acquisition combining both the structured data from CrunchBase database and the application of text-mining on scrapped news from the website TechCrunch. Their model's TPR ranges between 60% and 79.8% for different company's category using Bayesian Network (BN) as their machine learning algorithm. FPR (False Positive Rate) ranging between 0 and 8.3% over categories with less missing values in the CrunchBase corpus. Their result is much better than the previously state-of-art article, Wei et al. (2009), who achieved a precision rate of 42.9% and 46.4%. Also, their final dataset consisted on 59 631 observations and with more than 6 000 acquisitions, this study far exceeded the 2 394 cases analyzed by Wei et al. (2012). Additionally, they proved that their text-mining component improves overall results.

Except for studies using CrunchBase database, most have small and specific datasets for the task at hands, and although achieving promising results, the nature of the data limits expansions on their work. Also, most works tend to focus on managerial features which doesn't tell the full scope of a company's status or potential to be acquired. Studies using CrunchBase database also do not take full potential of the data available opting for not creating several features related with the impact of venture capital such as number of investors, rounds of investment, amount raised among many others. In their defense, it must be said that some of the information available today might not been available at the time of their studies.

Authors	Title	Year	ML	Results	Baseline
Ragothaman, S., Naik, B., & Ramakrishnan, K.	Predicting corporate acquisitions: An application of uncertain reasoning using rule induction	2003	discriminant analysis and rule induction	81.3% for acquired companies; 65.6% for non-acquired companies	nan
Ravisankar, P., Ravi, V., Raghava Rao, G., & Bose, I.	Detection of financial statement fraud and feature selection using data mining techniques.	2011	Probabilistic Neural Network	98% True Positive Rate - fraudulent companies	nan
Wei, C. P., Jiang, Y. S., & Yang, C. S.	Patent analysis for supporting merger and acquisition (M&A) prediction: A data mining approach.	2009	Naive Bayes	~ 45% Precision Rate	nan
Liang & Daphne Yuan	Investors are Social Animals: Predicting Investor Behavior using Social Network Features via Supervised Learning	2012	Decision Trees	87% True Positive Rate	Partly
Xiang et al.	A Supervised Approach to Predict Company Acquisition with Factual and Topic Features Using Profiles and News Articles on TechCrunch	2012	Bayesian Network	69.4% (average) True Positive Rate	X

Table 1 - Previous studies on company acquisition

3. METHODOLOGY

The methodology here applied (Figure 12 – Methodology Overview) mirrors a loose interpretation of Knowledge Discovery in Databases (KDD) approach (Fayyad et al., 1996): **(1) Selection** of data to be processed by defining relevant tables from the entire structured CrunchBase database; **(2) Preprocessing**, by cleaning, Selecting and Transforming data. At this stage we deal with missing values, outliers, discretization, and other common problems. An explorative analysis is made before further transformations; **(3) Experiment Setup,** where evaluation metrics are defined, and the two major problems of the dataset - Sparsity and Imbalanced target classes, are dealt with. Both these problems are only addressed at this stage. Several machine learning algorithms are chosen to test a binary classifier to *classify* the observations as either "successful" or "not-successful"; **(4) Experiment Results**, where we draw conclusions and interpret results.

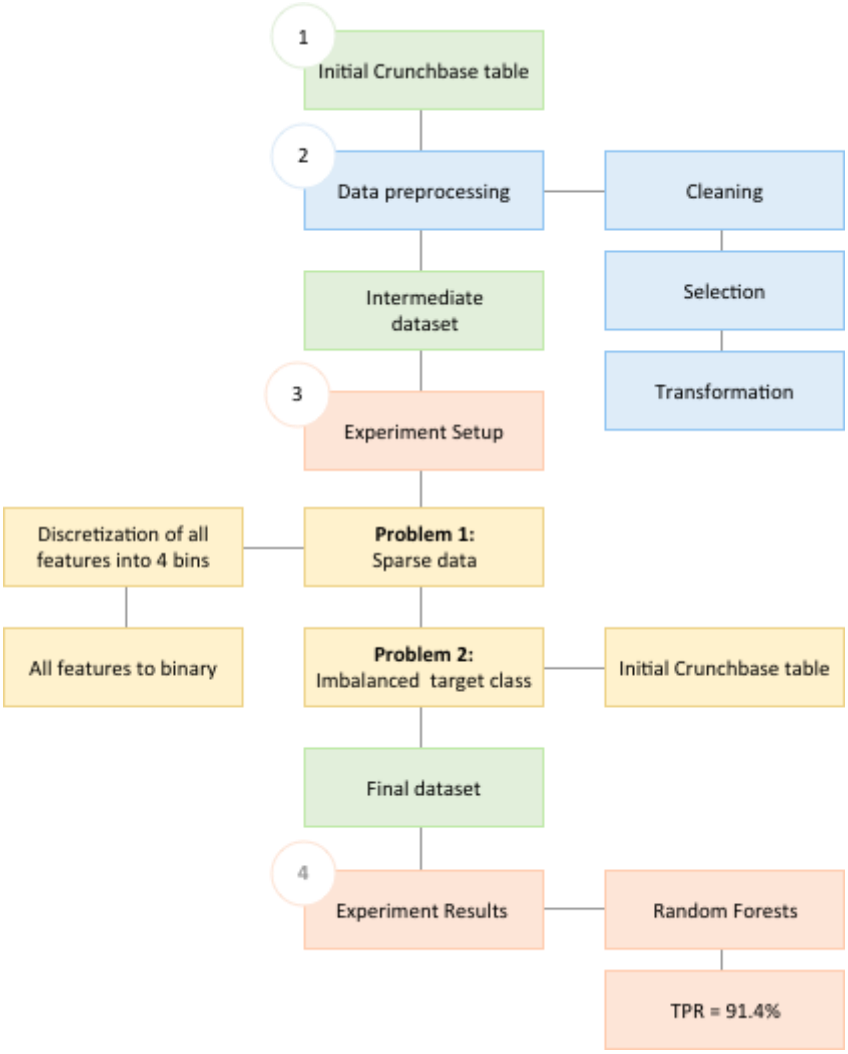


Figure 1 - Methodology Overview

3.1. DATA COLLECTION AND SELECTION (CRUNCHBASE CORPUS)

The data used in the present work is the entire structured database from the website CrunchBase.com and was acquired on 2017-01-23. The data is created and edited by its users and moderators. It is important to disclaim that the access to the data was given to the author for an *Academic License* and to be used exclusively for the present work.

Despite being community based, especially for small-to-medium companies, its value is not to question. It has been an invaluable resource for multiple different companies as venture capitals, consulting companies (Deloitte, Oliver Wyman), marketing and sales platforms (Engagio, Datanyze) and previous academic studies ("Customer Stories | Crunchbase Data Solutions," 2017; Liang & Daphne Yuan, 2012; Yuxian Eugene & Daphne Yuan, 2012). The website is a referenced database for all start-up ecosystem and investors in general.

As of 2017-01-23, the database from CrunchBase consisted of 20 tables in CSV (comma-separated-values) files:

Name	*Observations*	*Selected*
organizations	495 798	•
people_descriptions	Na	
people	422 032	•
organization_descriptions	Na	
jobs	996 453	•
competitors	520 137	•
funding_rounds	153 412	•
customers	304 323	•
investments	237 668	•
investors	50 319	•
events	33 211	
acquisitons	35 532	•
investment_partners	44 525	•
ipos	11 986	•
schools	10 891	
event_relationships	7 717	
org_parents	6 942	•
funds	5 611	•
category_groups	737	
awards	38	

Table 2 - CrunchBase Database

Note: To produce a dataset for the training task, only tables marked 'Selected' column will be used.

People_descriptions, *organization_descriptions* are descriptive tables of people and organizations and do not have pertinent information for this work. The tables *events, event_relationships, schools* and *awards* are also out of scope and do not possess relevant information. All the selected tables will provide data converging in *organizations* table therefore acting as support tables.

3.2. DATA PRE-PROCESSING

"If there is much irrelevant and redundant information present or noisy and unreliable data, the knowledge discovery during the training phase is more difficult".

(Kotsiantis, Kanellopoulos, & Pintelas, 2006)

Data pre-processing can often have a critical impact on general performance of a supervised machine learning task. The process will follow ***general changes*** (as transversal to all thirteen tables in-use) and ***changes made to the organizations*** table as it is where all relevant information converges, ultimately becoming the training dataset of the task at hands. Due to the nature of the data and problem the priority is understanding its interdependence and not minimizing correlations.

The data pre-processing consists in a 3-step process:

- **Data cleaning**, where the author aims to remove all redundant and irrelevant information from the database as well as duplicates, missing values and outliers. The explanation of this process is divided between specific changes in the 'Organizations' table and general changes made transversely in all tables;
- **Data selection,** where the context of the study (i.e., social-demographic criteria) is defined to filter which data will be taken into the final dataset and
- **Data transformation,** consisting on the process of creating new variables or aggregating data from different tables into organization's table.

3.2.1. Data Cleaning

"Data cleaning is a time-consuming and labor-intensive procedure but one that is absolutely necessary for successful data mining." (Witten et al., 2000)

Feature	*Description*	*Type*
uuid	Organization's unique id	Nominal
company_name	Organization's name	Nominal
primary_role	Company, group, investor, school	Categorical
permalink	Link to access organization information on CrunchBase	Nominal
domain	Organization's domain	Nominal
homepage_url	Organization' URL	Nominal
country_code	Country (i.e., USA – United States)	Categorical
state_code	USA's states (i.e., CA – California)	Categorical
region	Country's region	Categorical
city	Country's city	Categorical
zipcode	Organization's zip code	Nominal
address	Organization's address	Nominal
status	Operating, closed, acquired or IPO	Categorical
short_description	Short description	Nominal
category_list	Subcategories of an organization	Nominal
category_group_list	General categories of an organization	Nominal
funding_rounds	Total funding rounds	Ordinal
Funding_total	Total amount raised	Interval
funding_total_usd	Total amount raised in US dollars.	Interval
Founded_on	Foundation date	Datetime
first_funding_on	Date of first funding	Datetime
last_funding_on	Date of last funding	Datetime
closed_on	Organization's date of closure	Datetime
employee_count	Quantity of employees (in interval categories)	Categorical
email	Organization's email	Nominal
phone	Organization's phone	Nominal
facebook_url	Organization's Facebook URL	Nominal
linkedin_url	Organization's LinkedIn URL	Nominal
cb_url	Organization's CrunchBase URL	Nominal
logo_url	Organization's logo URL	Nominal
profile_image_url	CrunchBase's profile image of organization	Nominal
twitter_url	Organization's twitter URL	Nominal
created_at	Date of instance creation (timestamp)	Datetime
updated_at	Date of last edit (timestamp)	Datetime

Table 3 – Organizations' Table

The first step of pre-processing consists on treating all the ***irrelevant*** and ***redundant*** information present in tables. As a free-to-edit database with multiple purposes, the CrunchBase dataset has several columns (features) and instances (observations) whose context don't match the objective of predicting a start-up's success.

From the 'Organizations' table:

- Deleted *region*, *city*, *zip_code*, *address* as they provide too much granularity.
- Deleted *domain, homepage_url, cb_url, facebook_url, linkedin_url, logo_url, twitter_url, profile_image_url, short_description; name, email and phone* as irrelevant features.
- Deleted *funding_total* (as we only need *funding_total_usd*, a *standardized* version in U.S dollars allowing comparisons between the funding of companies from different countries in the same currency.)
- Deleted *category_list* (a subgroup comprising 689 unique values to (sub) categorize an organization, as a specific methodology (3.2.3.1) was applied to define a single category for an organization, this column ceased to be relevant).

It is also important to evaluate the redundancy of certain observations by looking for the presence of ***duplicates***:

General changes:

- Only a few duplicate instances were found in the database and all were removed.

The second step consists on eliminating ***noisy*** or ***unreliable*** data being the two most common cases of inconsistencies**, Missing Values** and **Outliers**. A Missing value (or missing data) is a variable that has no data value stored in an observation. Missing values are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data. "Most machine learning methods make the implicit assumption that there is no particular significance in the fact that a certain instance has an attribute value missing: the value is simply not known." (Witten et al., 2000).

Although they may occur for several reasons (such as malfunctioning measurement equipment, changes in variable definition during data collection), the most probable cause for missing values in this dataset is simply because the creator of the profile didn't put all the information on the database's profiles, hence making it incredibly difficult to separate sparse data from missing values in the current context.

Following the premise, "as the amount of data decreases, the rate of increase in accuracy grows" (Kotsiantis et al., 2006), the following instances in the *organizations* table were simply deleted as the features may still have relevant information for the predictive task.

From the 'Organizations' table:

- Deleted instances with missing values for *primary_role*, *status*, *country*, *category_*group, and *founded_date*;

Outliers are excessively deviating values from the scale of the feature. (Kotsiantis et al., 2006) An example of an outlier found in the dataset can be an extremely high "total funding in USD" (i.e. *total_funding_usd* = $B 30000, read, thirty thousand billion dollars*)* of a specific company, probably due to an incorrect conversion from a different currency by the user who edited the organization's page. Observations with excessively deviating values were deleted.

From the 'Organizations' table:

- Deleted outliers for *funding_total_usd, #funding_rounds*.

Another type of inconsistent data can be ***misspellings*** or ***contradictory values***, especially due to the crowdsource nature of the in-use database. Wrong dates or presence of letters in numerical features are examples of frequently present inconsistent data.

General Changes:

- Deleted instances where *closed_on* is before *founded_on*, generating negative age (in #years).

3.2.2. Data Selection

Before further advancements in the experiment setup of the dataset it is important to contextualize what will be the subject of study and filter data. Due to the context of the present the study only the companies from the United States were selected to be part of the training dataset. They were categorized as: CA (California), NY (New York), TX (Texas), MA (Massachusetts) and Other (consisting on the remaining states):

- Being part of American website of tech news TechCrunch, it means every article has referenced data from the database. More quality of information and media coverage means more revisions and curated information;
- Platform's only language is English which limits the input of some features by foreign users like correct values for currencies;
- Similar strategy used by Xiang et al., (2012), although they used regions instead of states. Also, CrunchBase only started to export international start-ups and other profiles in 2014 (Lennon, 2014);
- California is, historically and currently, the most important place for tech companies worldwide (Weller, 2016);
- Top 5 most-represented states.

usa_state_code	COUNT	%	SUCCESSFUL COMPANIES	TARGET RATIO
MA	4 609	5%	922	20%
TX	4 967	6%	695	14%
NY	9 926	11%	1 191	12%
CA	27 291	32%	4 912	18%
Other	39 795	46%	5 969	15%
All	86588	100%	13690	16%

Table 4 – US states used in dataset

Companies with founding date between 1985 and 2014: Although some of this companies can't be considered start-ups anymore due to their advanced age without a *success* event, they were at some point and some had events as funding rounds who potentially brought them closer to success, so these companies will stay in the dataset. A similar strategy was used by Xiang et al., (2012). The rationale behind this decision relies on the assumption that companies need time to mature and show results. At the same time, we cover the Dot-com bubble in 1997 & World crisis 2008.

Companies with at least one review of its profile 90 days after its creation: Users and moderators can both review company profiles. By only having access to *date of creation* and *date of last modification*, the author filtered companies with at least a 90-day difference between the two. This transformation allows two main advantages: it limits the number of fake and incomplete company profiles and guarantees that profiles were subject to a review in a 90-day period.

Companies with category: To try to compare results with previous publications and being a category of a company something that influences, among other factors, its average age of success the author chose to only take companies with a category to further analysis; The company's category reflects both its industry as well as if it is a tech company or not.

After all the previous filters were applied, the dataset is comprised of 86 588 observations.

3.2.3. Data Transformation

Transforming data can be summarized as "the application of mathematical modification to the value of a variable" to extract more value than in its original state (Osborne, 2002). In the present dissertation, the data transformation process can be divided in two successive phases:

1st) Changes in original data;

2nd) New variables created.

3.2.3.1. Changes in original Data

These changes are applied in a consistent way to all variables.

Organizations table:

- *Employee_count* from categorical to ordinal: Discretized categories, nulls became "0"; (50-200) became 1; or (201-205) became 2 (to a total of 9 categories). (Annex 8.4.1)
- Transformed all dates in age in #years (i.e., *date_founded* to *age_yrs* or *date_closed* to *age_closed*).
- **Category:** All companies were classified into one of 32 categories: *commerce, commerce(Tech), communications(Tech), education, education(Tech), entertainment, entertainment(Tech), financial, financial(Tech), government, government(Tech), hardware(Tech), healthcare, healthcare(Tech), information tech(Tech), internet services(Tech), lifestyle, lifestyle(Tech), manufacturing, manufacturing(Tech), media, media(Tech), mobile(Tech), mobility, mobility(Tech), realEstate, realEstate(Tech), sciences(Tech), security(Tech), software(Tech), utilities&energy, utilities&energy(Tech).* This process allows, for example, the possibility to distinguish two *media* companies like TechCrunch (tech) and Wall Street Journal (non-tech).

We now detail the process used to determine each company's category. Originally organizations had between 1 up to 14 categories selected from a 46-unique value list. It varies between "software", "hardware", "manufacturing", "energy", etc. Categories are merged into a column separated by the symbol "|"and sorted from A-Z. A simplistic method consisting on (1st) generalizing the 46-unique list into 20 newly created categories; (2nd) replacing each of the 46-unique values in each observation for the respective of the new category; (3rd) evaluate the *mode* of each category in each observation; (4th) attribute the category most represented (ties were always attributed to the less frequent to generate a more balanced distribution).

The new categories are: *realEstate, manufacturing, entertainment, lifestyle, hardware, education, mobile, mobility, internet services, software, financial, media, commerce, information tech, healthcare, utilities&energy, sciences, communications, government, security*.

The applied method had the intent of creating heterogeneous and representative categories in the simplest way possible. As previously stated, there was also a sub category column *(category_list*) but it consisted on 689 unique values, so it was not considered and was deleted. During this process, it became possible to categorize a company as a tech company and an

additional binary feature was created (*isTech*). Ultimately, both the category and the feature "isTech" were merged and every organization fall into one of the following 32 categories: *commerce, commerce(Tech), communications(Tech), education, education(Tech), entertainment, entertainment(Tech), financial, financial(Tech), government, government(Tech), hardware(Tech), healthcare, healthcare(Tech), information tech(Tech), internet services(Tech), lifestyle, lifestyle(Tech), manufacturing, manufacturing(Tech), media, media(Tech), mobile(Tech), mobility, mobility(Tech), realEstate, realEstate(Tech), sciences(Tech), security(Tech), software(Tech), utilities&energy, utilities&energy(Tech). This process allows, for example, the possibility to distinguish two media companies like TechCrunch (tech)* and *Wall Street Journal (non-tech).*

3.2.3.2. New Variables

Using the information present in the other tables the following variables were created in the organization's table (queries can be consulted in annex 8.4). Note that most of data in the database is either missing or sparse, which affects the selection of machine learning algorithms and might require special treatment to lower its level. Since the distinction between the two is dubious, the percentage of missing/sparse data per variable is defined as **Sparsity Level**. Further transformations were made to promote the best results for the learning task. An in-depth explanation of the strategy used can be read in 3.3.2.

Name	*Description*	*Sparsity Level (Avg=69%)*	*Average*
roundD	The company did a round D	98.8%	
roundC	The company did a round C	95.9%	
roundB	The company did a round B	92.5%	
roundA	The company did a round A	88.8%	
VentureCapital	Has venture capital (with missing values)	60.7%	
isTech	Is a tech company	0.0%	
target	The company was acquired by other or went to a public stock market (IPO)	0.0%	
roundD_raised_amount	Raised amount of Round D	98.8%	$40.449.855
roundC_raised_amount	Raised amount of Round C	96.0%	$21.162.205
roundB_raised_amount	Raised amount of Round B	92.9%	$14.968.688
roundA_raised_amount	Raised amount of Round A	89.7%	$7.640.412
investment_per_round	Total US Dollars invested per round of investment	61.5%	$9.161.589
funding_total_usd	Total funding in US dollars	61.5%	$21.646.508
roundD_age	Company's age when it did its round D	98.8%	6,5
total_investmen	Total number of investments made by	98.5%	2,7

ts	the company		
customer_count	Number of customers	98.3%	6,0
ipo_age	Company's age when it went to a public stock market	96.4%	7,7
roundC_age	Company's age when it did its round C	95.9%	5,4
total_acquisitions	Number of acquisitions made by the company	93.9%	2,2
competitor_acquired_ipo	Number of competitors either acquired or IPO'd	93.1%	2,0
roundB_age	Company's age when it did its round B	92.5%	4,1
roundA_age	Company's age when it did its round A	88.7%	3,3
competitor_count	Number of competitors	87.8%	3,2
age_Acquired	Company's age when acquired	87.1%	9,2
success_age	Age of company when it got acquired or went to a public stock market (IPO)	84.3%	8,6
top500_investor	Number of top500 investors in the company (Top 500 by number of investments made by investor)	81.2%	3,5
investors_per_round	Number of investors per round	70.0%	2,3
total_exp_founders_years	Total experience of founders in years	70.0%	9,5
age_first_funding_year	Company's age when it received first funding	53.9%	3,2
funding_rounds	Number of funding rounds	53.9%	2,1
totalFounders	Number of founders	41.4%	1,7
total_experience_jobs_years	Total experience of total jobs in the company in years	28.8%	12,0
totaljobs	Total jobs of the company	23.9%	5,3
age_yrs	Actual age in years	0.0%	10,3

employee_count_ordinal	Ordinal feature to classify interval of employees of company (Where, 0 = Missing Values and 10 = [1001;100000] employees)	0.0%	1,8
category_general	One of thirty-two different categories of the company	0.0%	
usa_state_code	One of five different states (CA, NY, MA, TX, Other)	0.0%	

Table 5 - Variables created from other tables

3.2.4. Dataset Breakdown

All companies in the dataset belong to one of five USA' states: CA – California, NY – New York, MA – Massachusetts, TX – Texas and Other (consisting on the rest of North American states). A similar approach was used by Xiang et al., (2012). Yuxian Eugene & Daphne Yuan, (2012) and Liang & Daphne Yuan, (2012) which also used data from US-only start-ups (have also used CrunchBase database).

According to the data, the state of California is where it is fastest to achieve success as a company, taking an average of 7 years to either be acquired or to go to the U.S stock market (thus being designated successful for the present study). The most famous region for tech companies in the world is Silicon Valley, home of many of the world's biggest tech companies – including headquarters of 39 companies of Fortune 1000 (109 for the entire California state). Silicon Valley became a global synonym for leading high-tech research and companies and accounts for one third of the venture capital investment in the U.S which attracts an enormous amount of technology workers to the region. ("Fortune 1000 Companies List for 2016 - Geolounge," n.d.) With an environment that heavily promotes entrepreneurship and with a lot of capital to invest in, companies in the state of California develop faster comparing with the rest of the U.S. 'Other' states' companies have an average of 9,7 years before achieving success compared to 7,1 years in California, which proves how attractive the state is to fund a company.

The state of California also has one of the highest percentage of successful companies with 18% of successful companies to Massachusetts, 20%. However, companies in the state of Massachusetts take around 1,8 years more to achieve success. New York also boasts an interesting average age of success for its companies (7,6 years) but has a lower percentage of success cases (14%). Texas has a better percentage of successful companies than New York but at a lower average age of success with 9,4 years.

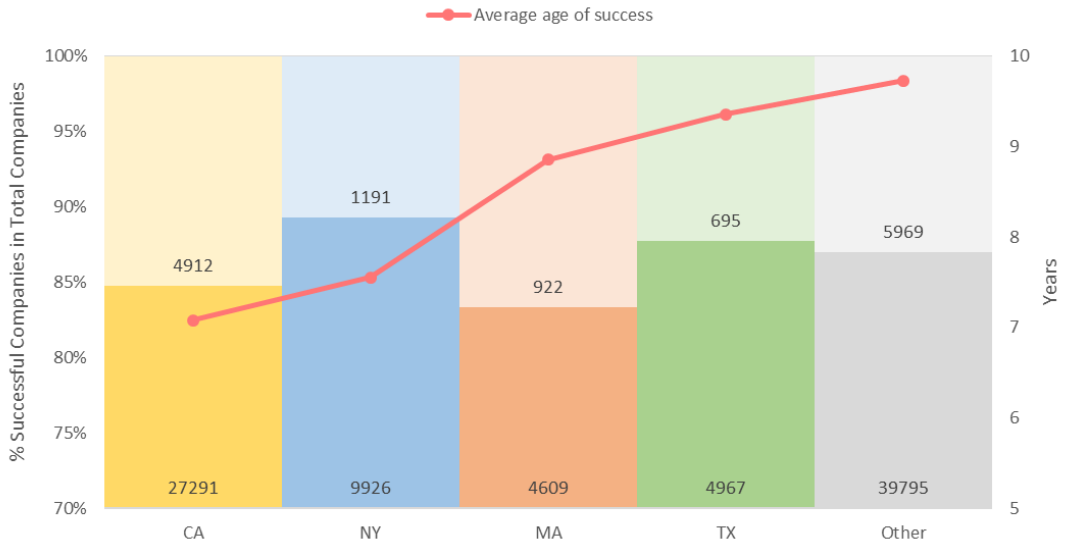


Figure 2 - Average age of success per state

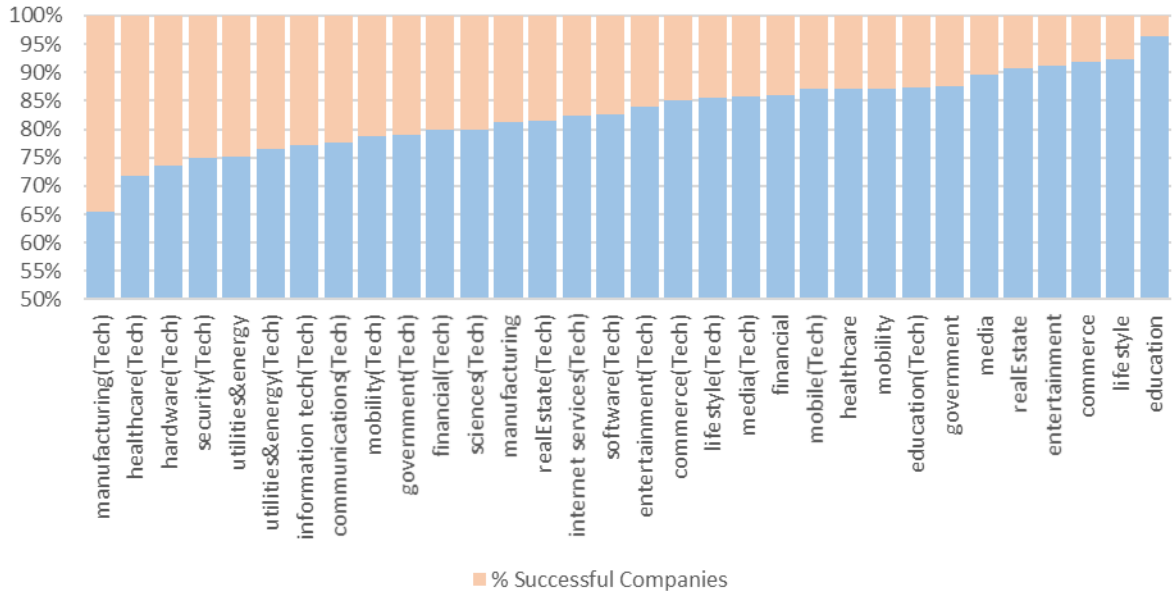


Figure 3 – Percentage of successful companies per category

Another categorical feature present in the dataset is the general **category** of a company. A company is categorized in one of thirty-two different labels (Figure 3). An overview of the percentage of successful companies per categories allows two immediate conclusions: of the top-10 categories with most successful companies (Figure 3), 9 are tech companies, being *'energy&utilities'* the only non-tech category represent and of the top-10 least successful, 9 are non-tech. Second, there is a large difference between the most successful category – *'manufacturing(Tech)'*, 35%, and the least, *'education'*, 4%. (Full table can be consulted in annex 8.1 – Table 20).

It is also important to understand how different categories of a business influence its average age of success as companies in industries like 'healthcare' take more time to implement, test and regulate its products/services than a tech company in entertainment like a video game company which would take much less time to develop, test and launch products. According to the dataset (Figure 4), the fastest way to achieve success in the top 10 most represented categories is by founding a tech company in New York in the category of *'commerce'* (such as an e-commerce or an online marketplace) as it will find success, as previously defined, in less than 6 years. On average, 'entertainment (Tech)' is the fastest category to achieve success in all states (7 years) and California, as previously mentioned, the fastest state. In the opposite direction, the slowest way is by founding a *'healthcare'* company in New York as it takes, on average, more than 13 years to find success. *'Healthcare'* is also overall category average, the slowest category to achieve success with 10,6 years.

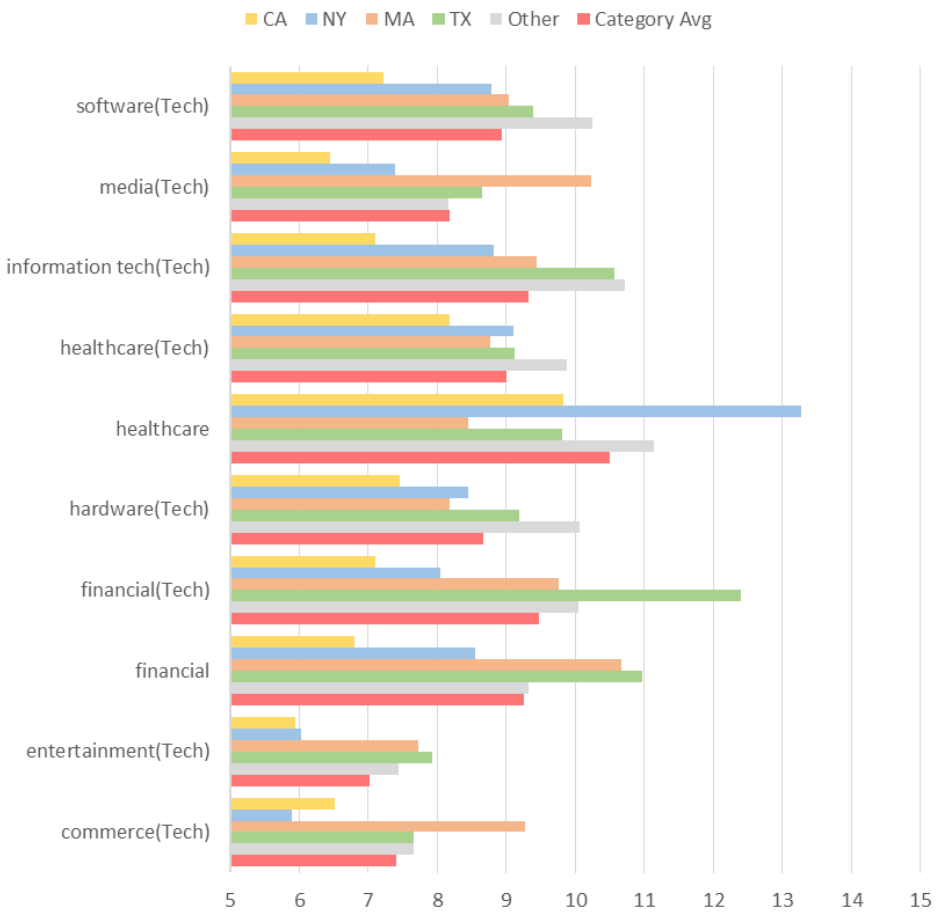


Figure 4 - Average age of success per TOP-10 category in each state

Of the 10 categories in analysis, it is possible to compare both 'healthcare' and 'healthcare (tech)' as well as financial and financial (tech), with both tech categories averaging a faster overall age of success than its non-tech counterparts (Figure 4).

With 76,5% of all successful companies in the dataset being tech companies it is important to understand the importance of financing through venture capital and how raising money from Venture Capital funds (instead of borrowing money from a bank) allows companies to develop faster and, ultimately, achieving success faster. Venture capital is a type of financing that investors provide to start-ups/companies where money is exchanged by a percentage of equity (or ownership stake) of the company. This type of financing mechanism has as premise the high growth potential of said start-up but also the higher risk associated.

Due to the nature of technology, with faster scalability than a traditional non-tech business, Venture Capital is often more associated to tech companies.

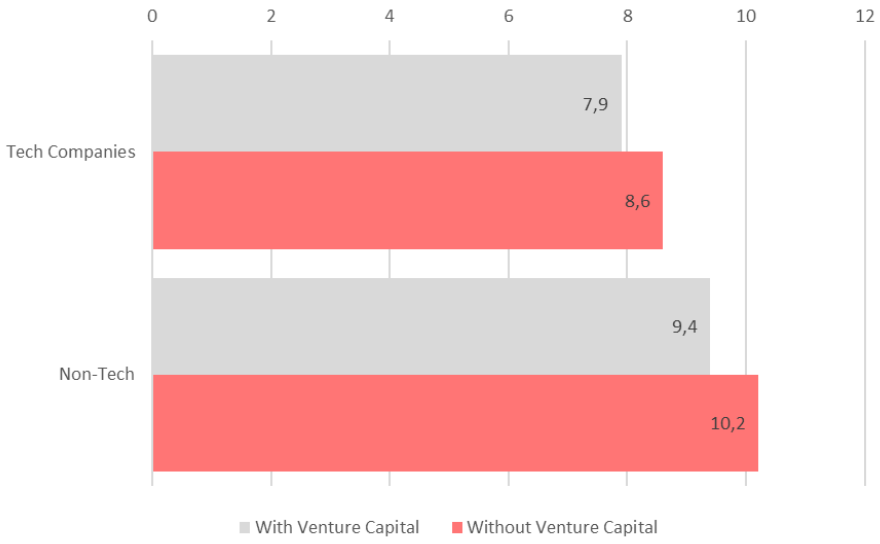


Figure 5 – Average impact of VC in Tech and non-Tech companies

A tech company with raised money through Venture Capital (VC) achieves success, on average, 7 months earlier (7,9 months overall) than without Venture capital and a non-tech with VC funding achieves it 8 months earlier (in 9,4 years). M&A activities are especially critical for tech companies as they often use M&As to acquire state-of-the-art technologies or rapidly expand their R&D capabilities (Wei et al., 2009).

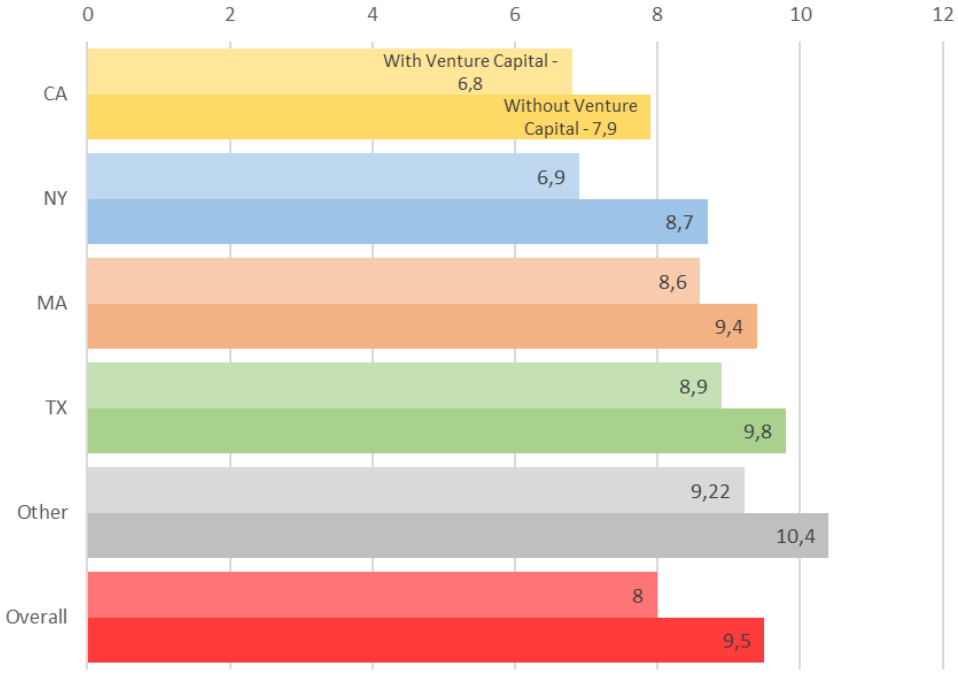


Figure 6 - Impact of VC in each state (tech and non-tech)

State-wise, California benefits from a heavily venture capital-centered environment and a start-up with VC money achieves success 1,1 year faster than without VC.

Overall, VC allows companies to achieve success 1,5 years faster than without Venture capital. Not only is it faster but it becomes more probable to find success as 65,4% percent of the successful companies had some type of Venture capital invested in (Figure 7).

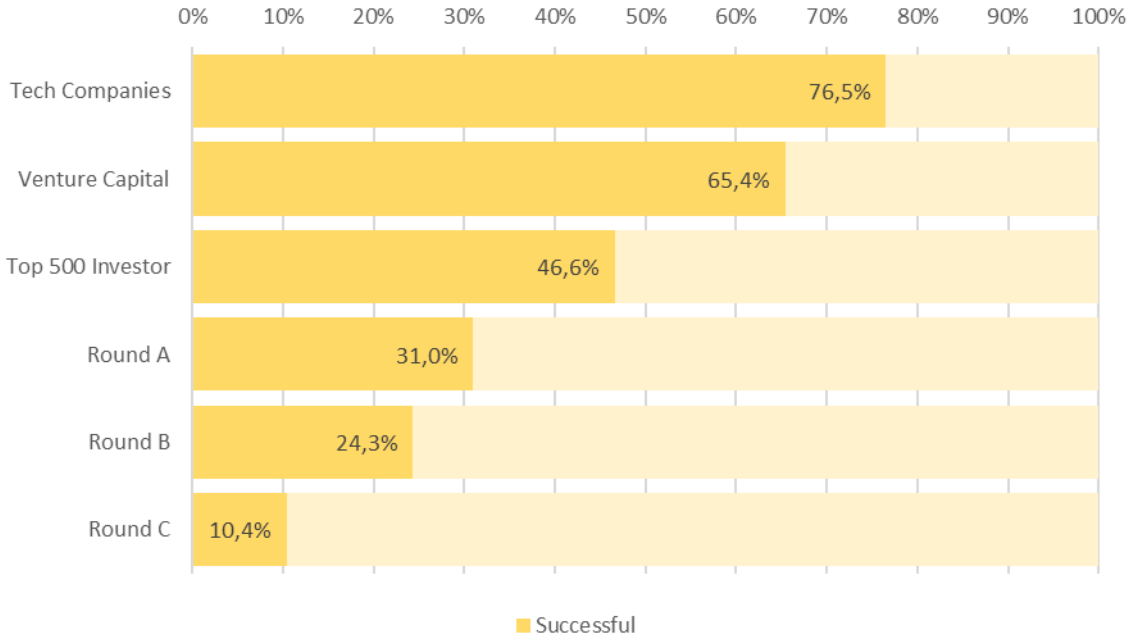


Figure 7 - Impact of VC and Top500 Investors in successful companies

A financing round of Venture capital through a fund usually consists on a first round called Seed, a second round called round A, a third round called B, and so on. The present study considered Venture capital as any type of round (including seed) and the consequent rounds separately (round A, B, C, D). Of all the successful companies in analysis, 31% has a round A, which allows a company to accelerate its development and hire more resources in less time. Only 10,4% of all successful companies achieve a round C, proving how hard it is to grow a business to this stage (Figure 7).

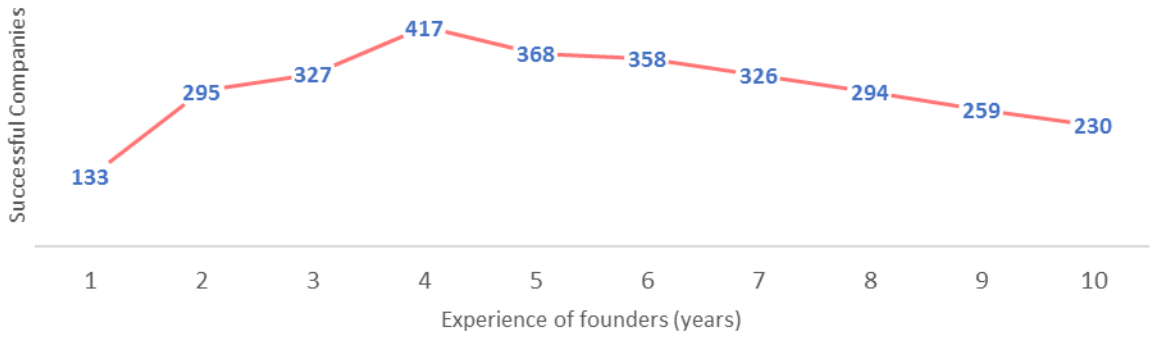


Figure 8 - #successful companies per #experience of founding team (in years)

One fact that should be considered to understand what influence the success of a company should be the prior experience of the founding team. As Figure 8 illustrates, founding teams with 4 years of cumulative experience are almost four times more frequent than successful companies with one or less years of experience. Curiously, the trend isn't upwards as expected as more years of experience don't always mean more successful companies although it can be assumed that more experience is always better than 1 year. The above trend can be related with the nature of the database in question, which, as a free-to-edit database would require more reviews and effort from the platform's users in people, companies and jobs profiles than a company with less information to add/edit. Also, we assume people are less likely to be involved in the start-up scene as they get older due to the high risk attached to it.

The average founding team is composed by 1,8 members and 46,6% of all successful companies includes a Top 500 Investor (by number of investments) which proves how a company becomes credible by having an expert investing in it.

An analysis of the funding level of a company allows us to understand that there are more successful companies as the total amount of funding (either through venture capital or other financing mechanism) grows. It can be assumed that more money allows companies to develop faster.

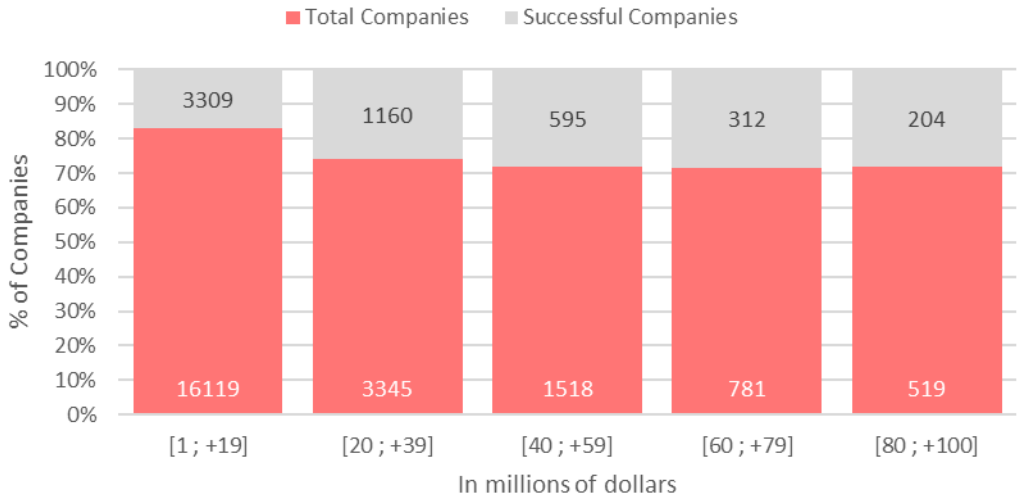


Figure 9 - Funding Total (in millions of dollars) per company

Evaluating the funding level of a company and its relation with success one can assess that more funding means more probability of success. Of all 16'119 companies with a funding between 1 and <20 million dollars the percentage of successful companies is 20,5% while those with funding between 20 and <40 million dollars have a successful rate of 34,7%. Interestingly from a funding of 40 million dollars upwards the success rate is almost constant at 39,3%. As companies raise more money and become more valuable they also become harder to be bought as less players in the market have the money to buy them out (acquire them).

Summarizing, founding a start-up in California raises chances of success and lowers the time needed to achieve it. California is also the most tech-savvy city in the world, culturally and historically (Weller, 2016). Of the top-10 most successful categories to find a company by % of successful

companies, 8 are tech, meaning it is easier to be successful in technology (Figure 3). Success also comes 1,5 years faster for tech-companies (Figure 5). Founding an e-commerce or marketplace (tech) in New York is the fastest way to achieve success, overall – less than 6 years. (Figure 4). Venture capital also allows a faster growth, and success by approx. 1,5 years comparing to companies without this type of investment (both for tech and non-tech) (Figure 5, Figure 7). Companies where founders have a cumulative of 4 years of experience between founders (with an average of approximately 2 founders) also show more success. The relative inexperience of founders is probably compensated by higher levels of energy and availability compared with founders with more experience (Figure 8). Finally, more funding means higher chance success but after a total funding of, at least, $20 million the % of successful companies is almost constant (Figure 9).

3.3. EXPERIMENT SETUP

3.3.1. Evaluation Metrics

The classifier will have as its main evaluation metrics, **True Positive Rate (TPR)** and **False Positive Rate (FPR)**. Not only are those standard for most binary classification tasks but they were also used in the work considered as state of art. Also, by using the same metrics we can perform a statistical comparison between the two approaches for the same problem.

True Positive Rate (TPR = TP / (TP+FN)) or **Recall** can be defined as the percentage of all the successful companies correctly identified as successful. On the other side, **False Positive Rate (FPR = FP / (FP+TN))** can be understood as the percentage of all unsuccessful companies classified as successful. As an easy to understand metric, **TPR** clearly tells the predictive capability of the crucial aspect under study - classifying companies as successful with the features and methodology in-use.

Confusion Matrix	0, (Predicted Negative)	1, (Predicted Positive)
0, (Actual Negative)	True Negative (TN), company classified as not successful and it is not successful	False Positive (FP), company classified as successful and it is not successful
1, (Actual Positive)	False Negative (FN), company classified as not successful and it is successful	True Positive (TP), company classified as successful and it is successful

Table 6 - Confusion Matrix

Precision will be shown as a support metric and can be defined as, “percentage of all successful companies correctly classified”. Although this metric is not the one used to compare results with previous studies it supports how well our instances are classified.

Precision = (TP+TN) / (TP+FP+TN+FN)

As typical measure used in statistics to evaluate binary classifiers, ROC (Receiver Operating Characteristic) curve is a graphic plot that illustrates the predictive capability of a model by plotting both cumulative TPR and FPR at different thresholds. The area under the ROC curve (AUC) is a standard metric taken from the ROC curve as it clearly shows the trade-off between both main evaluation metrics, TPR and FPR. AUC measures the discrimination of the model, meaning, its capability to correctly classify instances of the test. Using the present context, the area under the curve is the percentage of randomly drawn pairs of observations (one with target=1 and one with target=0), for which the test correctly classifies both observations in said pair of observations. As reference for future results, a AUC between 90% and 100% is an excellent result while results between 50 and 60% are considered failures. (Hanley & McNeil, 1982)

3.3.2. Problems with the dataset and solutions used

3.3.2.1. Sparsity of the dataset

The first problem found in the present analysis was the sparsity of the CrunchBase database. As previously stated by Xiang et al., "despite its huge magnitude, the CrunchBase corpus is sparse with many missing attributes in the profiles" (Xiang et al., 2012). Although the authors argue that the age and maintenance of the platform were a problem as it was too young, as well as the fact that only popular entities and features were frequently reviewed, five years later the problem persists, and it has grown into a bigger one. Due to its free-to-edit nature, anyone can create companies and fill its data without much control. This fact allied with its growing popularity creates an exponential growth in sparse data as more incomplete profiles are created than reviews are made. Also, it is reasonable to state that the rate of new companies created is much faster than investment rounds, acquisitions or IPO's profiles are added/happen. After the data pre-processing process, which consisted on cleaning, transforming and selecting data, the sparsity level of the dataset was of 75%, which was too high.

Despite the ability of the machine learning algorithms used in this work to deal well with sparse data, the ambiguity in what is sparse and what is missing value in the present context motivated us to solve the problem in the following two steps:

- A **binary feature** was created to support features with missing data. These binary features meant to signal whether the observation had value in the feature. For example, *number_of_investors* became supported by an additional *hasInvestor* feature. By doing this it is possible to create a dataset without missing data and at the same time *awards* companies with more information.

- All the sparse data in interval and ordinal features were imputed with "0" (zero) – this proved the most cost-efficient way of dealing with sparse data without too much loss of information (compensated by the creation of the support feature).

The following binary variables were added to the dataset:

Original Variable	*Binary Variable*	*Description*	*Average*
competitor_acquired_ipo	hasSuccessfulCompetitor	The company has at least one successful competitor	6.9%
competitor_count	hasCompetitor	The company has competitor	12.2%
customer_count	hasCustomers	The company has customers	1.7%
funding_rounds	hasFundingRound	The company has at least one funding round	44.3%
funding_total_usd	hasFunding	The company has funding	38.4%
investment_per_round, investors_per_round	hasInvestmentIn	The company has received any form of investment	43.2%

roundA	has_roundA	The company raised money a series A round	13.3%
roundB	has_roundB	The company raised money in a series B round	8.8%
roundC	has_roundC	The company raised money a series C round	4.1%
roundD	has_roundD	The company raised money a series D round	1.2%
top500_investor	hasTop500investor	The company has at least one top 500 investor	21.0%
total_acquisitions	hasAcquired	The company has acquired another company	6.1%
total_exp_founders_years	has_founder_experience	The founders have experience	15.1%
total_experience_jobs_years	hasExperience	The company has experience	71.1%
total_investments	hasInvested	The company has invested in another company	1.5%
totalFounders	hasFounder	The company has founder(s)	58.6%
totaljobs	hasJobs	The company has at least one job in its profile	76.1%
VentureCapital	hasVC	The company has any form of venture capital	42.3%

Table 7 - Binary features created (Support)

Although the first step was enough to create a non-sparse dataset there is still a lot of value to be taken from interval and ordinal variables as a company with $500'000 in investment cannot be compared with one with $10'000'000 although both have value 1 in said binary feature.

- A **discretization** of all features into a maximum of 4 bins using equal frequency instead of equal-width binning to ensure the missing values imputed with "0" and high values wouldn't have too much weight in the newly created bins. For example, the feature *funding_rounds* which had values ranging between 1 and 24 is discretized in 4 different interval values: [-inf-1.5], [1.5-2.5], [2.5-3.5], [3.5- inf].

- Also, at the cost of more training time, all features were transformed into **binary features**. Although there is no theoretical advantage on doing this transformation, the results were higher in models with it. Our rationale was: "In a random forest, each node is split using the best among a subset of predictors randomly chosen at that node" (Liaw & Wiener, 2002), and by breaking features into a max of four bins and then turning them into (a max of four) independent binary features would allow a broader number of ***m*** random features available to make subsets from. This would result in *trees* where the split is made in a specific value of the feature which may or may be not present in said subset of features. Trees can be formed based on a high/low value of a feature and not the feature itself. This minimizes the correlation between features by allowing more combinations of features in each *tree.* As for Logistic Regression and Support Vector Machines, this transformation would have no negative impact on the algorithms output (although making SVM much slower to compute).

- Categorical features, '*usa_state_code*' and '*category_general*', are converted to numerical by expanding the categorical feature with n possible values into n binary features. For example, 'usa_state_code' with 'CA' has its values transformed in 'usa_state_code=CA' with '1' for all the observations from California and '0' for all the other observations. The multicollinearity created by this transformation is not a problem with Random Forests.

With the present transformations a unique dataset in which the algorithms can train from at the same time is built, allowing a quick comparison of results. These transformations were made in Weka – an open source software that easily allows, among other features, the manipulation of the dataset. A full table with the final features generated after this transformation can be consulted in annex **8.3 Final features.**

Original Variable	*Bins (in binary features)*	*Discretized Feature*
competitor_acquired_ipo	4	competitor_acquired_BIN-X
age_first_funding_year	4	age_first_funding_year_BIN-X
competitor_count	4	competitor_count_BIN-X
customer_count	4	customer_count_BIN-X
funding_rounds	4	funding_rounds_BIN-X
funding_total_usd	4	funding_total_usd_BIN-X
investment_per_round	4	investment_per_round_BIN-X
investors_per_round	4	investors_per_round_BIN-X
roundA_age	4	roundA_age_BIN-X
roundA_raised_amount	4	roundA_raised_amount_BIN-X
roundB_age	4	roundB_age_BIN-X
roundB_raised_amount	4	roundB_raised_amount_BIN-X
roundC_age	4	roundC_age_BIN-X
roundC_raised_amount	4	roundC_raised_amount_BIN-X
roundD_age	4	roundD_age_BIN-X
roundD_raised_amount	4	roundD_raised_amount_BIN-X
top500_investor	4	top500_investor_BIN-X
total_acquisitions	4	total_acquisitions_BIN-X
total_exp_founders_years	4	total_exp_founders_years_BIN-X
total_experience_jobs_years	4	total_experience_jobs_years_BIN-X
total_investments	4	total_investments_BIN-X
totalFounders	4	totalFounders_BIN-X
totaljobs	4	totaljobs_BIN-X
category_general	36	category_general=x
usa_state_code	5	usa_state_code=x

Table 8 - Discretized features

3.3.2.2. Imbalanced Classes

"A dataset is imbalanced if the classes are not approximately equally represented."

(Chawla, Bowyer, Hall, & Kegelmeyer, 2002)

Another problem faced when trying to create a good predictive model for the task at hands was the large class imbalance between successful and non-successful companies. After pre-processing, only 16,8% of the dataset consisted on successful companies. Most machine learning algorithms work best when the number of observations of each class is equal because when there is such disparity between classes the algorithms tend to classify the lowest represented class as the opposed. In the present study, if all observations were marked negative (unsuccessful) the model would still achieve around 83% of Accuracy, which would be a better score than most models published in predicting success of a company (Wei et al., 2009; Xiang et al., 2012).

Not only is "Accuracy" a dangerous metric to evaluate the quality of a model with a large imbalance of classes (ROC curve is more adequate) but also the problem of class imbalance can be tackled using different strategies such as over sampling the lowest represented class or under sampling the largest.

SMOTE (Synthetic Minority Over-Sampling TEchnique) is a technique that consists in an oversampling of the minority class. Meaning it will create new synthetic instances of the lowest represent class (in this case, successful companies) rather than by over-sampling with replacement. Frequently used in fraud detection this technique was first introduced by Chawla. "The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the ***k*** minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen." (Chawla et al., 2002)

After dealing with the sparsity, SMOTE was applied to our dataset before testing the different machine learning algorithms. With an increase of 400% of synthetic instances classified "successful" with 5-nearest neighbors, the classes become balanced:

Before SMOTE	**Number of observations**	%
0	72 398	84%
1	14 190	16%
	86 588	100%

After SMOTE (400%;5 n neighbors)	**Number of observations**	%
0	72 398	51%
1	70 950	49%
	143 348	100%

Table 9 - Number of observations with SMOTE

The complete algorithm (pseudo-code) can be consulted in appendix, Figure 13.

3.3.3. Machine learning algorithms

In the present work, we have a **binary classification** task - the target feature is either classified as "1" (for successful companies) or "0" (for not-successful companies). It is a type of **supervised learning**, a method of machine learning where the output categories are predefined. It is important to choose not only the algorithm that better fits the problem but also one which adapts well to the characteristics of the dataset:

- 158 features
- 143 348 observations
- No-sparse data as it consists only of binary features (some algorithms cannot process observations with missing values)

Different learning algorithms make different assumptions about the dataset and have different purposes. When testing the following algorithms, we intended to test its data with ML models that not only fit the nature of dataset but are also easy to understand and implement. It is equally important to test algorithms used with this dataset in previous works.

In the following sections, we will do an overview of each tested machine learning algorithm – **Logistic regression (LR)**, **Support Vector Machines (SVM)** and **Random Forests (RF)**, aiming to understand the logic, advantages, disadvantages and applications of this techniques.

LR and **SVM** were previously tested by Xiang et al. (2012) to predict company acquisition using the same dataset. As a novelty, we are adding **RF** since its application fits the nature of our problem.

3.3.3.1. Logistic regression

Although the problem under study is a classification one (output is a category) and not a regression (where the output is continuous), **Logistic regression (LR)** is a modelling technique where the dependent variable (target), usually but not necessarily, takes binary values – "0" or "1", "not successful" or "successful". This allows the technique to be applied in classification problems in machine learning.

LR works the same way as a linear regression: in layman language, by multiplying each input by a coefficient, summing them up, and by adding a constant to each feature thus assuming there is a linear decision boundary that divides (classifies) its instances. In linear regression, the output is very straightforward: if we are predicting someone's weight, our output is simply someone's predicted weight. In logistic regression, however, the output is the log of the odds ratio (Hosmer & Lemeshow, 2000). Taking Ian Witten and Eibe Frank's example (2.2.1) – *the weather problem*, if we are predicting if it will rain tomorrow, the odds ratio is the odds that it will rain tomorrow divided by the odds that it won't rain. In logistic regression the outcome has limited number of possible values, either it will rain (Y=1) or it won't (Y=0) as it estimates the *probability* of an event and transforms it in a categorical format (for example, "0" if *probability* < 0.5 or "1" if *probability* > 0.5) (Hosmer & Lemeshow, 2000). It then finds the linear classification by making assumptions that P(Y=1|X), like the inverse logit function applied to a weighted sum of our features. In machine learning, the coefficients of the logistic regression algorithm must be estimated from the training data, with **LR** this is done using maximum-likelihood estimation (Jason Brownlee, 2016). Again, using *the weather problem* for context, the best model would have coefficients predicting values very close to "1" (as it will rain) for the default class and values very close to "0" (as not raining). The rationale for maximum-likelihood in logistic regression is that a search procedure looks for values for the coefficients that minimize the error in the probabilities predicted by the model (Jason Brownlee, 2016).

In machine learning, **LR** is one of the simplest and fastest algorithms to train and implement and usually used as a starting point for many classification problems. Since it has low variance it is less prone to over fit making it very adequate for binary classification problems with a clear separation of classes. Other advantages are its capability to not make assumptions about distributions of classes in feature space and its ability to be extended to multiple class classification problems instead of binary.

The major disadvantages of **LR** are the limited outcome which may limit its application if specific contexts and how prone they can be to overfitting if trained in datasets with many correlated features (Howbert, 2012). **LR** supports problems with high dimension data, say 100 000 observations for 500 000 features.

Being one of the most popular machine learning algorithms, it is possible to find applications of Logistic Regression in virtually any field of study – from fraud classification to classification of potential clients in marketing campaigns. For example, a recent study by Karan & Kumar (2016), applies **LR** in predicting bankruptcy for companies using financial ratios in a dataset consisting on 500 samples based on 41 attributes with an Area under the ROC curve of 96.7%.

3.3.3.2. Support Vector Machines

Manning et al. defines **Support Vector Machines (SVM)** in a very simplistic and concise way: "**SVMs** are inherently two-class classifiers (…) An **SVM** is a kind of large-margin classifier: It is a vector-space-based machine learning method where the goal is to find a decision boundary between two classes that is maximally far from any point in the training data (possibly discounting some points as outliers or noise)." The maximal decision boundary is the *hyperplane* that separates the classes and has the largest distance between border-line data points (support vectors) (Boser, Guyon, & Vapnik, 1992). With reference, a case in biomedicine involving cancer cells, Statnikov et al. wrote that if such a hyperplane, "does not exist, the data is mapped into a much higher dimensional space ("feature space") where the separating decision surface is found". This non-linear classification of the feature space is enabled by a kernel function, i.e. a measure based upon the distribution of similarities between a given data point and other data points around it. This ability to model problems which are not linearly separable is an advantage over **LR.**

SVM is different from other ML algorithms. It can handle high dimensional data/dataset (with large number of features) **SVM** but is better suited for situations where the number of observations is not too large as it requires long training time due to being very memory-intensive (Manning, Raghavan, & Schütze, 2009; Statnikov, 2011). It is particularly popular in text-classification problems (Statnikov, 2011) where problems with high-dimensionality are common.

A common disadvantage of **SVM** is the interpretability of results. Using the current dataset as an example, **SVMs** cannot represent the score of all companies as a simple parametric function of all features, since its dimension is very high. The weights of our features are not constant and thus the marginal contribution of each feature to the score is variable. Using a Gaussian kernel each company has its own weights according to the difference between the value of their own features and those of the support vectors of the training data sample (Auria & Moro, 2008).

Additionally, **SVM** maximizes the "margin" and thus relies on the concept of "distance" between different points (Boser et al., 1992). This means it is more suited for problems with more numerical features than categorical since for the latter the concept of "distance" isn't applicable.

3.3.3.3. Random Forest

Random Forest (RF) is a collection of **Decision Trees (DT)**. Contrary to **LR** or **SVM** (although in a highly multidimensional space) **RF** does not expect linear features. In its simplest form, it can be thought of using bagging on multiple tree classifier (Leo Breiman, 2001). However, since it is not possible to build multiple trees on the same data as it will get the same results, randomness of two types is introduced: each tree is built on slightly different rows, sampled with repetitions from the original (bagging), and each tree (or in some cases each branch decision) is built using a randomly selected subset of columns. The point of **RF** is to prevent overfitting which it does this by creating the random subsets of features and building smaller (shallow) trees using the subsets.

According to Andy Liaw and Matthew Wiener, the algorithm can be summed up in the following steps: (1) Draw *n tree* bootstrap samples from the original data; (2) For each of the bootstrap samples, grow an unpruned classification or regression tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample m of the predictors and choose the best split from among those variables. This sample of *m* predictors minimizes the correlation between the classifiers in the ensemble (Gislason, Benediktsson, & Sveinsson, 2006); (3) Predict new data by aggregating the predictions of the *n* trees (i.e., majority votes for classification, average for regression) (Liaw & Wiener, 2002).

The way the algorithm handles the Bias-variance trade-off, a central problem in supervised learning, is one of the main advantages of **Random Forests** - although its bias is the same of a single Decision Tree, its variance decreases as we increase the number of trees which also decreases the chances of overfitting. Other advantages are the fact that it runs efficiently on large datasets, handles thousands of input features without feature deletion, gives estimates of what variables are important to the classification, processes missing data and even maintains high accuracy when this proportion is large (Leo Breiman, 2001).

The main disadvantage of **Random Forests** compared with a simple **Decision Tree** is its interpretability as it is hard to see the relation between a dependent variable and the rule set created. A **Random Forest** must be a predictive tool and a descriptive one. It is easy to see its features importance but that might not be enough when the objective of the study is to understand the relationship between dependent and independent variables.

Several applications of Random Forests have been made in multiple fields of study proving it is one of the most popular machine learning algorithms these days. Cutler et al., showed in 2007 how Random Forests achieved higher accuracy than other commonly used classifiers using data on multiple ecological scenarios such as invasive plant species presence in California or nest sites for cavity nesting birds in Utah, USA, while Gislason et al., applied the classifier on geographic data to explore land cover problems. In a completely different direction. Lariviere and Vandenpoel used Random Forests to understand customer behavior and how to improve customer retention rate and profitability using data from a large European financial services company, ultimately finding that past customer behavior is what matters the most to generate repeat purchasing and favorable profitability evolutions (Cutler et al., 2007; Gislason et al., 2006; Lariviere, Vandenpoel, & D, 2005).

3.3.4. Baseline

With a focus on start-up acquisition and investments from venture capital, Xiang et al. (2012), predicts company acquisition using CrunchBase's database through supervised machine learning algorithms. This structured database is the world's greatest reference in this field and at the time of study, January 10th, 2012, had profiles for 81 219 companies, 107 274 persons, 7 328 financial organizations, 3 955 service providers, 25 895 funding rounds and 6 173 acquisitions – after preprocessing, their final dataset consisted on 59 631 observations. With more than 6 000 acquisitions, this study far exceeded the 2 394 cases analyzed by Wei et al (2009).

They designed two types of features:

22 Factual features (basic, financial and managerial features) based on CrunchBase's profiles:

Basic: 1: #employees, 2: company age (months), 3: number of milestones in the CrunchBase profile, 4: number of revisions on the company CrunchBase profile, 5: number of TechCrunch articles about the company, 6: number of competitors, 7: number of competitors that got acquired, 8: headquarter location, 9: number of offices, 10: number of products, 11: number of providers

Financial: 12: number of funding rounds, 13: number of investments by the company, 14: number of acquisitions by the company, 15: number of venture capital and private equity firms investing in the company, 16: number of people with financial background investing in the company, 17: number of key persons in the company with financial background, 18: number of investors per funding round, 19: amount of investment per funding round

Managerial: 20: number of companies founded by founders of the target company, 21: number of successful companies by founders, 22: founder experience (months)

Topic Features from an extraction of articles from news website TechCrunch

Using Latent Dirichletian Allocation (LDA) as their text-mining algorithm to build five topic features, they scraped 38 617 tech news for 5 075 companies during December 2011.

Bayesian Network (BN) outperformed Support Vector Machines (SVM) and Logistic Regression and was used as the primary learning algorithm as "it could better represent the probabilistic relationships between features via local conditional dependencies, which is more robust that simple linear classifiers". They defined True Positive (TP) and False Positive (FP) as the main classification metrics.

Their model achieved a TP between 60% and 79.8% with FP between 0 and 8.3% over categories with less missing values in the CrunchBase corpus. Their result is much better than the previously state-of-art article, (Wei et al., 2009) who achieved a precision rate of 42.9% and 46.4% for a data set of 2 034 companies.

Further improvements suggested by the authors include, defining more features to alleviate discrepancies in some unsuccessful companies, include more financial features traditionally used in acquisition prediction (price to earnings ratio, return on average asset, etc.) and include IPO as a positive class label (treating IPO as an acquisition). As for the topic features, they suggest including different sources as Twitter, Quora and Wikipedia could offer more suitable text data for the task.

Some of the suggestions were considered during the present dissertation such as including IPOs as a part of the positive class label and the creation of different features to gap the distance between successful and not-successful companies.

The article will be partly used as the baseline for the present dissertation as it shares the same data source (although older and with less data) and objective of predicting whether a start-up/company will be acquired or not. The present results will also be compared with (Liang & Daphne Yuan, 2012) and their study, *Investors Are Social Animals: Predicting Investor Behaviour using Social Network Features via Supervised Learning Approach,* who predicted if an investor would invest in a particular company based on their social network. Although not directly predicting success or company acquisition but rather what makes an investor invest in a specific company one can interpret their model as what does a company have to have, feature-wise, to make an investor, who always seeks for success, to invest in. The fact that the present study tries to find the features to explain what makes a company successful and the fact the authors used CrunchBase database and supervised learning techniques makes the study's results worthy of comparison.

3.4. EXPERIMENT RESULTS

3.4.1. Evaluating Learning Algorithms

During a first stage of evaluation to see which ML algorithm better fits our problem, the accuracy of several algorithms was calculated. Although, accuracy alone is not recommended to evaluate and choose a ML algorithm (especially if there is a problem of class imbalance), at this stage, the classes are balanced, and accuracy is used as a quick and easy to interpret metric.

Due to the nature of the dataset, with many correlated features, **Logistic Regression (LR)** and **Support Vector Machines** (with a linear kernel) were tested. These algorithms were previously tested by Xiang et al. in the paper considered baseline allowing a good platform to compare results.

Using a 10-fold cross-validation in a sample of 25% of the full dataset the following accuracies were calculated:

- Logistic Regression: 0.928 (0.0015)
- SVM (linear SVC): 0.928 (0.0014)
- Random Forests: 0.931 (0.0029)

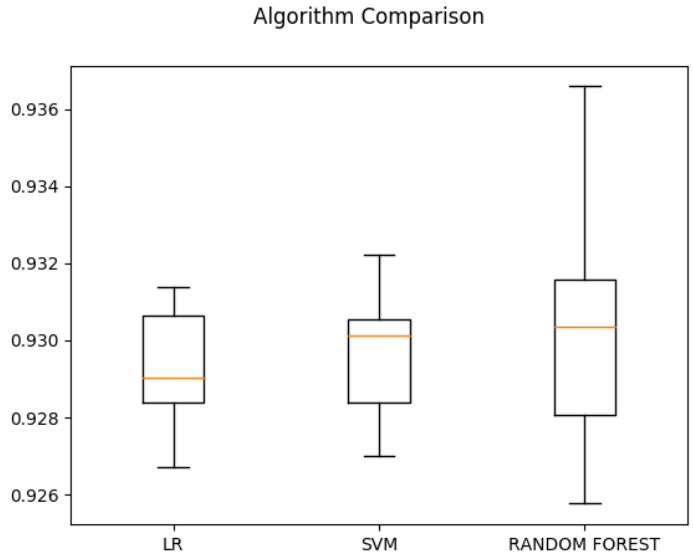


Figure 10 - Accuracies for **LR**, **SVM** and **Random Forests**

All the tested machine learning algorithms proved to be worthy contenders as all achieved a very high overall accuracy. Random Forests has both the highest accuracy and variance (as expected) as shown in the *Figure 10.* Being accuracy the sum of all true positives and true negatives over the total observations it provides a good and fast metric to pre-evaluate the performance of an algorithm in the current learning task (with balanced classes). **SVM** (with a linear kernel) and **LR** also achieved a very high overall classification accuracy, due to the correlation between the features in the dataset and thus being able to generate a linear separation in the feature space. But accuracy alone doesn't reflect the main metric used in this study, True Positive Rate (Recall).

3.4.2. Choosing the Learning Algorithm

An in-depth analysis over the algorithm's predictions using a split of a training set (70%) and a test set (30%) over the full dataset generated the following classification reports for an aggregated model with all categorical features included:

		Precision	Recall	f1-score	Support
Logistic Regression	0	0.891	0.966	0.931	21748
	1	0.962	0.890	0.924	21257
	Avg/Total	0.930	0.928	0.928	43005
SVM (LinearSVC)	0	0.896	0.971	0.932	21748
	1	0.968	0.885	0.925	21257
	Avg/Total	0.932	0.929	0.928	43005
Random Forest (Nº Trees=50;)	0	0.940	0.924	0.933	21748
	1	0.924	0.941	0.932	21257
	Avg/Total	0.933	0.932	0.932	43005

Table 10 - Classification Reports

Random Forest (built with 50 trees and the number of max features as the square root of the #features) is the chosen algorithm due to having the highest **True Positive Rate** (Recall) and its good balance between precision and recall. The **False Positive Rate**, 7.8% which is not an improvement over some previous studies still allows interesting interpretations, as explored further. **LR** and **SVM** (with a linear kernel) also achieve similar results, as expected, since they only differ in the loss function — SVM minimizes hinge loss while logistic regression minimizes logistic loss. **SVM** is also expected to perform marginally better than **LR** (Pedregosa, 2013).

Algorithm	Precision	True Positive Rate	False Positive Rate	Area under curve (ROC)
Random Forests (Nº Trees=50)	92.4%	94.1%	7.8%	93.2%

Table 11 - Random Forests' final output

Comparing all the tested algorithms, we can assess very good AUC scores for the three machine learning algorithms as shown in Figure 11. Area Under ROC is very high which suggests significant differences between *successful* and *non-successful* companies as it calculated based on the probability of random pairs of observations with different classification labels being both correctly classified. Although marginal, Random Forests still achieves a better score compared with the other models.

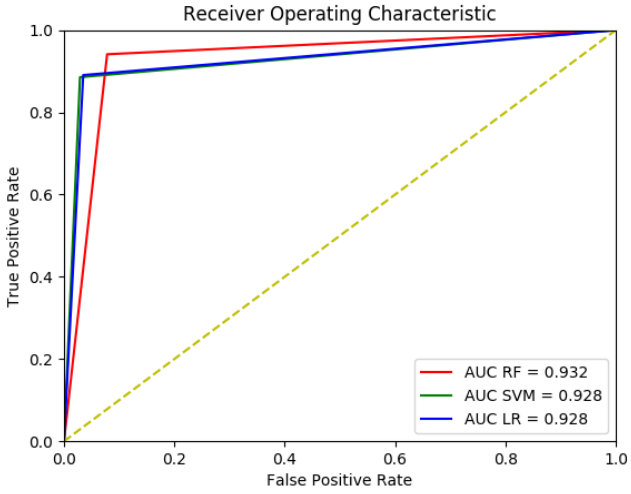


Figure 11 - ROC Curve

RF builds a more robust model than any of the other algorithms. Logistic Regression proves to be a good solution as there is a predictable linearity to the data – for example, companies with features like 'hasFundingRound', 'hasInvestmentIn' and/or 'has_top500_investor' (which are all binary) would be closer to success than those without any information in these features. **SVM** due to its capability to classify the observations in a multidimensional feature space also provides excellent results although at the cost of a higher time to train. Random Forests was ultimately chosen as it can discover more complex dependencies, like features that matter more for a specific company's category than others. Although the current dataset consists of only binary features without any missing value, Random Forest would also run through categorical data, data with large outliers and sparse data.

3.4.3. Feature Importance

Random forests are among the most popular machine learning methods thanks to their relatively good accuracy, robustness and ease of use. On the downside, RF lose interpretability when compared to single decision trees which can be recovered through **feature importance**: mean decrease impurity and mean decrease accuracy. Every node in the decision trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. The measure based on which the (locally) optimal condition is chosen is called impurity. For classification, it is typically either Gini impurity or information gain/entropy and for regression trees it is variance. Thus, when training a tree, it can be computed how much each feature decreases the weighted impurity in a tree. For a forest, the impurity decrease from each feature can be averaged and the features are ranked according to this measure.

The following output shows the top-20 out of 158 most important attributes based on average impurity decrease (and number of nodes using that attribute):

	Feature	*Average Impurity Decrease*	*Nr Nodes using Feature*
1	*hasVC='(0.5-inf)'*	0.33	26 878
2	*hasTop500investor='(0.5-inf)'*	0.31	24 123
3	*top500_investor='(2.5-5.5]'*	0.3	10 347
4	*top500_investor='(-inf-1.5]'*	0.3	14 543
5	*has_roundA='(0.5-inf)'*	0.3	23 888
6	*investors_per_round='(-inf-1.5]'*	0.3	23 071
7	*top500_investor='(5.5-inf)'*	0.29	6 408
8	*age_first_funding_year='(-inf-0.5]'*	0.29	16 653
9	*investors_per_round='(3.5-inf)'*	0.29	12 474
10	*usa_state_code=CA*	0.29	41 728
11	*funding_rounds='(-inf-1.5]'*	0.29	25 536
12	*age_first_funding_year='(1.5-3.5]'*	0.29	17 497
13	*funding_rounds='(1.5-2.5]'*	0.29	18 192
14	*top500_investor='(1.5-2.5]'*	0.29	9 836
15	*funding_rounds='(3.5-inf)'*	0.29	12 603
16	*investors_per_round='(2.5-3.5]'*	0.28	11 443
17	*investors_per_round='(1.5-2.5]'*	0.28	15 560
18	*age_first_funding_year='(0.5-1.5]'*	0.28	16 037
19	*funding_total_usd='(3500002.5-16248967.5]'*	0.28	16 368
20	*employee_count_ordinal='(2.5-inf)'*	0.28	22 174

Table 12 - Feature importance (Average impurity decrease)

Of the 20 features only two are non-related with investment. Having any type of Venture Capital invested in proves to be what is most important to classify a successful company as well as having top investors. Interestingly, having funding rounds matters but the class which matters the most of this feature is the one representing the interval with lowest number (1st bin of the discretized feature *funding_rounds*). The features created proved to have good explanatory capacity and thus allowing a good classifier of successful companies.

3.4.4. Evaluation by state and category

	TPR	FPR	Instances
Other	94%	8%	21 051
CA	94%	10%	14 766
NY	90%	10%	4 996
TX	96%	19%	3 282
MA	96%	23%	3 212

Table 13 - FPR & TPR per state

A new exploration is made by generating a model per each state contemplated in the dataset. The results are better for the most well represented states as California (CA) and 'Other' as both have much more observations for the learning task. 'Other' achieved results consistent with the general model with 94% TPR and 8% FPR while California had a slightly worse FPR of 10%. This analysis is useful to understand how important the size of the sample is to predict success of companies in our model. Massachusetts for example hit a very high 23% of false positives (with 96% of true positives, though) in a sample of 3 212 companies. Also, the fact that the categorical feature isn't present in the dataset as an independent variable allows us to assess its impact by comparing both TPR and FPR with the previously defined model.

Category	TPR	FPR	Instances
other(non-tech)	96%	4%	15 350
financial	94%	10%	6 499
lifestyle, entertainment, media(non-tech)	94%	9%	15 845
healthcare	91%	13%	5 797
sciences&education(Tech)	88%	13%	6 818
healthcare(Tech)	87%	8%	8 703
commerce	86%	4%	6 556
hardware(Tech)	86%	14%	6 897
software(Tech)	86%	16%	22 679
lifestyle, entertainment, media(Tech)	85%	18%	21 523
financial(Tech)	82%	16%	5 322
commerce(Tech)	76%	12%	5 096
other(Tech)	61%	4%	16 263
Average	86%	11%	14 3348

Table 14 - FPR & TPR per category

To have a more balanced number of observations per category, a new transformation was done to reduce into 13 unique categories. It is divided in 10 tech-categories and 3 non-technological, which follows the distribution of the dataset, consisting of 70% of tech companies. The category other(Tech) consisting on tech companies from communications, government, manufacturing, mobility, real estate, security and utilities & energy with the third higher represent class (16 263 observations) achieved a performance of 61% of TPR which is very low. On the other hand, the same categories but as non-tech companies achieved the best performance levels with 96% TPR and a 4%. This could mean that the categories composing other(Tech) have a much higher variance and heterogeneity between them thus making it harder for the model to classify them together. Also, for

'other' (non-tech) there are fewer positives in total which due to the *linearity* of our dataset makes it easier to classify as positive.

Although high, the False Positive Rate hides valuable information. As a false positively classified instance, it is reasonable to affirm that said company has enough value to be classified as successful by our model, which can be interesting for a financial analyst or an investor looking for suitable companies to invest in. The companies deemed successful but that really aren't are the companies.

4. CONCLUSIONS

The main objective of the present study was to generate a model to classify successful companies or start-ups. By building a binary classifier to *classify* a company as **successful** or **not-successful** with a **True Positive Rate (TPR) of 94.1%** and a **False Positive Rate of 7.8%** with 92.2% of Precision and an AUC of 93.2% it is assumed that the objective was achieved. It is the highest reported using data from CrunchBase. The model can classify with high efficiency not only the total of successful companies in the dataset (TPR, recall) but also, from all the successfully-classified which are successful (Precision). The machine learning algorithm used is Random Forests which provides a fast and easy to interpret and implement model with positive results. It provided better results than Support Vector Machines and Logistic Regression. Both the alternative models were chosen due to their potential to fit in the size and nature of our dataset as a linear relation was expected.

During the experiment setup, a transformation on all features was tested - after discretizing all features into a maximum of 4 interval bins, every feature was transformed into 4 new binary features, each assuming one bin. I.e., feature "*hasTop500investors*" which was comprised of values between 1-4 was transformed into four features "*hasTop500investor_BIN-X"*. This transformation, although not theoretically advantageous, decreased, the general model's FPR by 1% and TPR by 0.5% (at the cost of higher time to compute). This transformation allows the *trees* of the Random Forest to pick features which are specific values of the feature – allowing the model to *learn* from more specific *information* (through a higher number of combinations between features).

	TPR	*FPR*	*AUC*
RF without binary features	93.6%	8.8%	92.5%
RF with binary features	94.1%	7.8%	93.2%

Table 15 - Output with and without binary features

To provide comparable results with previous studies, using CrunchBase data to predict company acquisition or investment behavior, the present study is comprised of a general model which contemplates both the category and U.S state of a company and a model per category. Also, a model per each geographic region analyzed in our final dataset was provided. This approach is new and provides a new geographic baseline over the differences in company success over the states of California, New York, Massachusetts, Texas and all the other U.S states. The 'Other' states category achieved 94% TPR and 8% FPR and California 94% TPR with 10% FPR. States like Massachusetts and Texas, due to a much smaller number of observations (3 282 and 3 212 respectively), achieved worse results of FPR with 23% for Massachusetts and 19% for Texas, although both with 96% of TPR.

A Xiang et al. approached the problem by publishing performances of predictions per category, achieved TPR ranging from 44% to 79.8% with Bayesian Networks. It should be noted that their best performances were achieved in categories with a higher number of observations while the ones in the present study didn't always follow that behavior. The model achieved TPRs ranging between 61% and 96%. Area under ROC is also higher than theirs – 93.2% vs 88%. Ultimately, the present study benefited from a larger dataset in some categories which proved essential to achieve higher results,

the different treatment over the sparsity of the dataset and the creation of artificial observations to balance target class proved instrumental to achieve our results. FPR is higher (7.8% vs 2.2% in technological categories) but the interpretation of it holds important information as explored further ahead. Over all categories, their FPR varies between 0% and 3% while the present one is, on average, higher with results ranging between 4% and 18%.

Comparing the performance of the developed model with Liang & Daphne Yuan (*"Investors Are Social Animals"*), who aimed to build a model that explained how social relationships could impact an investor's decision at the time of investing, the study's Random Forests achieved a TPR of 94.1% versus theirs 89.6% achieved with SVM. Also, FPR is 7.8% compared with 33.4% for SVM which is considerably better. Liang & Daphne Yuan report a FPR of 5% for Naïve Bayes model although with a TPR of 54.8%. Category-wise, their published TPR ranges from 51% to 91% for Naïve Bayes while ours range between 61% and 96%. (Liang & Daphne Yuan, 2012)

The general model with all the categorical features aggregated achieved a better TPR than any model published so far using data from CrunchBase, proving to be very useful in predicting successful companies.

The FPR of the general model (7.8%) should be subject of analysis. Although false-positively classified one can assume these companies possess enough financial and managerial value to be classified as **successful**, which is an information of utmost value for an investor. Again, one can assume that these companies are close to what is understood as success in the present study, either an IPO or an M&A process. The model is predicting success for a start-up or a company by classifying it as successful although it still isn't.

The presented model provides better performance metrics (TPR) than its baseline studies. By applying some of its recommended changes – adding IPOs and new features, while doing some less orthodox transformations and applying Random Forests the author provides novelty to the study of company acquisition. It also possesses an interesting predicting potential with the treatment of the companies who fall into the false positive category, thus showing potential to achieve *success* as it is defined in the present study.

5. RECOMMENDATIONS FOR FUTURE WORKS

The exploration of companies here classified as false positives as it can be interpreted as companies showing enough potential to be considered successful according to the author's own definition of success – thus becoming of utmost value for those investing in future successful ventures.

The author suggests the application of different algorithms to the same data source and simpler transformations to the dataset than those applied here to achieve similar results.

Also, by providing an easy to use API, CrunchBase database could be turned into an operations tool which could be of use to funds, investors and all the other players operating in this space. Insights on this data as predictive models or segmentation are all explorations possible through the available data.

6. REFERENCES

10 Million Self-Driving Cars Will Be On The Road By 2020 - Business Insider. (2016). Retrieved February 1, 2017, from http://www.businessinsider.com/report-10-million-self-driving-cars-will-be-on-the-road-by-2020-2015-5-6

Aggarwal, C. C. (2015). *Data Mining: The Textbook*. *Springer International Publishing*. http://doi.org/10.1007/978-3-319-14142-8

Akerlof, G. A., Yellen, J. L., & Katz, M. L. (1970). The market for Lemons: Quality uncertainty and the market and the market mechanism. *The Quarterly Journal of Economics*. http://doi.org/488 500

Alam, A., & Khan, S. (2014). STRATEGIC MANAGEMENT: MANAGING MERGERS & ACQUISITIONS. *International Journal of BRIC Business Research*, *3*(1).

Ali-Yrkkö, J., Hyytinen, A., & Pajarinen, M. (2005). Does patenting increase the probability of being acquired? Evidence from cross-border and domestic acquisitions. *Applied Financial Economics*, *15*(14), 1007–1017. http://doi.org/10.1080/09603100500186978

Altman, E. I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *Journal of Finance*, *23*(4), 589–609. http://doi.org/10.2307/2978933

Artificial Intelligence and Machine Learning: Top 100 Influencers and Brands. (2016). Retrieved January 31, 2017, from http://www.onalytica.com/blog/posts/artificial-intelligence-machine-learning-top-100-influencers-and-brands/

Auria, L., & Moro, R. A. (2008). Support Vector Machines (SVM) as a Technique for Solvency Analysis, (August). Retrieved from www.diw.de

Berry, M. J. a., & Linoff, G. S. (2004). *Data mining techniques: for marketing, sales, and customer relationship management*. *Portal.Acm.Org*. Retrieved from http://portal.acm.org/citation.cfm?id=983642

Beyer, D. (2015). The Future of Machine Intelligence, Perspectives from Leading Practitioners.

Blank, S. G. (2006). *The Four Steps to the Epiphany*.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory - COLT '92* (pp. 144–152). http://doi.org/10.1145/130385.130401

Breiman, L. (1996). Bagging predictors. *Machine Learning*. Retrieved from http://www.springerlink.com/index/L4780124W2874025.pdf

Breiman, L. (2001). RANDOM FORESTS. Retrieved from http://www.math.univ-toulouse.fr/~agarivie/Telecom/apprentissage/articles/randomforest2001.pdf

Breiman, L., Friedman, J., Stone, C., & Olshen, R. (1984). Classification and regression trees.

Chawla, N. V, Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. Retrieved from https://www.jair.org/media/953/live-953-2037-jair.pdf

Christopher Clifton. (2009). Data Mining.

Chye Koh, H., & Tan, G. (2011). Data Mining Applications in Healthcare. *Journal of Healthcare Information Management —*, *19*(2). Retrieved from https://pdfs.semanticscholar.org/433a/57b382c528c78395e317d9fee008fb8ed9de.pdf

Customer Stories | Crunchbase Data Solutions. (2017). Retrieved October 17, 2017, from

https://about.crunchbase.com/customers/

Cutler, D. R., Edwards, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2007). RANDOM FORESTS FOR CLASSIFICATION IN ECOLOGY. *Ecology*, *88*(11), 2783–2792. http://doi.org/10.1890/07-0539.1

Farrar, C. R., & Worden, K. (2012). *Structural Health Monitoring: A Machine Learning Perspective - Charles R. Farrar, Keith Worden - Google Livros*. Wiley. Retrieved from https://books.google.pt/books?hl=pt-PT&lr=&id=2w_sp6lersUC&oi=fnd&pg=PP11&dq=machine+learning+health&ots=E1vmyBFsvo&sig=Mavuhd4Aq5DqiafMeP8nhHmyPOg&redir_esc=y#v=onepage&q=machine learning health&f=false

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, *17*(3), 37. http://doi.org/10.1609/aimag.v17i3.1230

Fortune 1000 Companies List for 2016 - Geolounge. (n.d.). Retrieved May 23, 2017, from https://www.geolounge.com/fortune-1000-companies-list-2016/

Geier, B. (2015). What Did We Learn From the Dotcom Stock Bubble of 2000? Retrieved from http://time.com/3741681/2000-dotcom-stock-bust/

Gislason, P., Benediktsson, J., & Sveinsson, J. (2006). Random forests for land cover classification. *Pattern Recognition Letters*. Retrieved from http://www.sciencedirect.com/science/article/pii/S0167865505002242

Graham, P. (2012). Startup = Growth. Retrieved February 1, 2017, from http://www.paulgraham.com/growth.html

Gugler, K., & Konrad, K. a. (2002). Merger Target Selection and Financial Structure, (2001), 1–25.

Guo, B., Lou, Y., & Pérez-Castrillo, D. (2015). Investment, Duration, and Exit Strategies for Corporate and Independent Venture Capital-backed Start-ups.

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques*. *Soft Computing* (Vol. 54). http://doi.org/10.1007/978-3-642-19721-5

Hanley, J. A., & McNeil, B. J. (1982). The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology*, (143). Retrieved from http://pubs.rsna.org/doi/pdf/10.1148/radiology.143.1.7063747

Hermann, B. L., Gauthier, J., Holtschke, D., Bermann, R. D., & Marmer, M. (2015). The Global Startup Ecosystem Ranking 2015. *The Startup Ecosystem Report Series*, (August), 1–156.

Hill, K. (2012). How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did. Retrieved October 17, 2017, from https://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/#5b90f7766686

Ho, T. K. (1995). Random Decision Forests. Retrieved from http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf

Kantardzic, M. (2003). *Data mining: concepts, models, methods, and algorithms*.

Kim, E. (2015). Fastest startups to $1 billion valuation - Business Insider. Retrieved August 21, 2017, from http://www.businessinsider.com/fastest-startups-to-1-billion-valuation-2015-8/#1-slack-is-the-fastest-growing-enterprise-software-ever-11111114

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data Preprocessing for Supervised Leaning. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE*, *I*.

Kudyba, S. (2014). *Big Data, Mining, and Analytics: Components of Strategic Decision Making - Stephan Kudyba - Google Books*. Retrieved from

https://books.google.pt/books?id=nuoxAwAAQBAJ&pg=PA287&lpg=PA287&dq=1.+Kincade,+K.+(1998).+Data+mining:+digging+for+healthcare+gold.+Insurance+%26+Technology,+23(2),+IM2-IM7.&source=bl&ots=6U-iwqjGtd&sig=U6DxjqzMopUQOGdl-yCyO9BIuJs&hl=en&sa=X&ved=0ahUKEwi

Lariviere, B., Vandenpoel, & D. (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications*, *29*(2), 472–484. http://doi.org/10.1016/j.eswa.2005.04.043

Lennon, M. (2014). CrunchBase Data Export Now Includes International Startups, Investors -. Retrieved October 20, 2017, from https://about.crunchbase.com/blog/crunchbase-data-export-now-includes-international-startups-investors/

Li, D., & Liu, J. (2010). The Life Cycle of Initial Public Offering Companies: A Panel Analysis of Chinese Listed Companies.

Liang, E., & Daphne Yuan, S.-T. (2012). Investors Are Social Animals: Predicting Investor Behavior using Social Network Features via Supervised Learning Approach.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*. Retrieved from https://www.researchgate.net/profile/Andy_Liaw/publication/228451484_Classification_and_Regression_by_RandomForest/links/53fb24cc0cf20a45497047ab/Classification-and-Regression-by-RandomForest.pdf

Loff, J. (2016). Using factorization machine to predict ratings from reviews text alone.

Machiraju, H. . (2003). *Mergers, Acquisitions and Takeovers*.

Manning, C. D., Raghavan, P., & Schütze, H. (2009). An Introduction to Information Retrieval. *Cambridge University Press*. Retrieved from https://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf

Marita Makinen, B., Haber, D., & Raymundo of Lowenstein Sandler, A. P. (2014). Acqui-Hires for Growth: Planning for Success. *Lowenstein Sandler PC*. Retrieved from https://www.lowenstein.com/files/Publication/6118b183-d40e-4a5e-8b95-58236c063a10/Presentation/PublicationAttachment/ea0af508-0319-4e3a-86dd-6452b1b6f15d/AcquiHires for Growth.pdf

Marr, B. (2016). The Top 10 AI And Machine Learning Use Cases Everyone Should Know About. Retrieved October 16, 2017, from https://www.forbes.com/sites/bernardmarr/2016/09/30/what-are-the-top-10-use-cases-for-machine-learning-and-ai/#2c5c482b94c9

Meador, A. L., Church, P. H., & Rayburn, L. G. (1996). Development of Prediction Models for Horizontal and Vertical Mergers. *Journal of Financial and Strategic Decisions*, *9*(1).

Mitchell, T. M. (2006). The Discipline of Machine Learning.

Neal, R. W. (2014). WhatsApp Investors Make Billions From Facebook Acquisition: Sequoia Capital Sees 50x Return On $1.3 Billion Investment. Retrieved August 21, 2017, from http://www.ibtimes.com/whatsapp-investors-make-billions-facebook-acquisition-sequoia-capital-sees-50x-return-13-billion

NGUYEN, T. (2015). ETA Phone Home: How Uber Engineers an Efficient Route - Uber Engineering Blog. Retrieved February 1, 2017, from https://eng.uber.com/engineering-an-efficient-route/

Onalytica. (2016). ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING.

Osborne, J. W. (2002). Notes on the Use of Data Transformation. - Practical Assessment, Research & Evaluation, *8*(6).

Pedregosa, F. (2013). Loss Functions for Ordinal regression. Retrieved July 25, 2017, from http://fa.bianp.net/blog/2013/loss-functions-for-ordinal-regression/

Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare: promise and potential. http://doi.org/10.1186/2047-2501-2-3

Ragothaman, S., Naik, B., & Ramakrishnan, K. (2003). Predicting corporate acquisitions: An application of uncertain reasoning using rule induction. *Information Systems Frontiers*, *5*(4), 401–412. http://doi.org/10.1023/B:ISFI.0000005653.53641.b3

Ravisankar, P., Ravi, V., Raghava Rao, G., & Bose, I. (2011). Detection of financial statement fraud and feature selection using data mining techniques. *Decision Support Systems*. http://doi.org/10.1016/j.dss.2010.11.006

Ries, E. (2011). The Lean Startup. *Working Paper*, 1–28. http://doi.org/23

Rogers, R. (2016). *MERGERS & ACQUISITIONS REVIEW MERGERS & ACQUISITIONS REVIEW Fairness Opinion Rankings*.

Samuel, A. L. (1962). SOME STUDIES IN MACHINE LEARNING USING THE GAME OF CHECKERS.

Schapire, R. (2008). COS 511: Theoretical Machine Learning (Princeton).

Statnikov, A. (2011). *A Gentle Introduction to Support Vector Machines in Biomedicine: Theory and methods*. Retrieved from https://www.google.com/books?hl=en&lr=&id=cxs8DQAAQBAJ&oi=fnd&pg=PP1&dq=+Gentle+Introduction+to+Support+Vector+Machines+in+Biomedicine&ots=TuaSDurkM_&sig=PErWxr2J8SLRFuGRp0iBF5L6YIY

Thiel, P., & Masters, B. (2014). *Zero to One*. Crown Business. Retrieved from www.crownpublishing.com

Wei, C. P., Jiang, Y. S., & Yang, C. S. (2009). Patent analysis for supporting merger and acquisition (M&A) prediction: A data mining approach. *Lecture Notes in Business Information Processing*, *22 LNBIP*, 187–200. http://doi.org/10.1007/978-3-642-01256-3_16

Weller, C. (2016). The 25 most high-tech cities in the world - Business Insider. Retrieved October 20, 2017, from http://www.businessinsider.com/the-most-high-tech-cities-in-the-world-2016-6/#25-washington-dc-1

Witten, Frank, & Eibe. (2000). Data Mining: Practical Machine Learning Tools and Techniques, Second Edition.

Xiang, G., Zheng, Z., Wen, M., Hong, J., & Rose, C. (2012). A Supervised Approach to Predict Company Acquisition with Factual and Topic Features Using Profiles and News Articles on TechCrunch. *Artificial Intelligence*, 607–610.

Yuxian Eugene, L., & Daphne Yuan, S.-T. (2012). Where's the Money? The Social Behavior of Investors in Facebook's Small World. http://doi.org/10.1109/ASONAM.2012.36

Zhang, D., & Zhou, L. (2004). Discovering golden nuggets: Data mining in financial application. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*. http://doi.org/10.1109/TSMCC.2004.829279

7. APPENDIX

7.1. SMOTE PSEUDO-CODE

Algorithm *SMOTE*(T, N, k)
Input: Number of minority class samples T; Amount of SMOTE $N\%$; Number of nearest neighbors k
Output: $(N/100)$ * T synthetic minority class samples

1. (∗ *If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd.* ∗)
2. **if** $N < 100$
3. **then** Randomize the T minority class samples
4. $T = (N/100) * T$
5. $N = 100$
6. **endif**
7. $N = (int)(N/100)$ (∗ *The amount of SMOTE is assumed to be in integral multiples of 100.* ∗)
8. k = Number of nearest neighbors
9. $numattrs$ = Number of attributes
10. $Sample[\,][\,]$: array for original minority class samples
11. $newindex$: keeps a count of number of synthetic samples generated, initialized to 0
12. $Synthetic[\,][\,]$: array for synthetic samples
(∗ *Compute k nearest neighbors for each minority class sample only.* ∗)
13. **for** $i \leftarrow 1$ **to** T
14. Compute k nearest neighbors for i, and save the indices in the $nnarray$
15. Populate(N, i, $nnarray$)
16. **endfor**

Populate(N, i, $nnarray$) (∗ *Function to generate the synthetic samples.* ∗)
17. **while** $N \neq 0$
18. Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest neighbors of i.
19. **for** $attr \leftarrow 1$ **to** $numattrs$
20. Compute: $dif = Sample[nnarray[nn]][attr] - Sample[i][attr]$
21. Compute: gap = random number between 0 and 1
22. $Synthetic[newindex][attr] = Sample[i][attr] + gap * dif$
23. **endfor**
24. $newindex{+}{+}$
25. $N = N - 1$
26. **endwhile**
27. **return** (∗ *End of Populate.* ∗)
End of Pseudo-Code.

Figure 12 - SMOTE algorithm (pseudo-code)

7.2. RANDOM FORESTS – HOW IT WORKS

A **Decision Tree** is a set of rules used to classify data into categories - it looks at all the variables in the dataset, determines which are most important (using, for example, the information gain with each feature) and then comes up with a tree-shaped scheme of decisions which best partitions the dataset. The *tree* is created by splitting data up by variables and then counting the frequency to see how many are in each bin after a split. Using the following sample example (used to classify hypothetical companies with generic features):

Name	*Number of Investors*	*California*	*Is Successful*

A	20	1	1
B	18	1	1
C	16	0	1
D	17	0	1
E	17	1	0
F	12	1	0
G	2	0	0
H	3	0	0

Table 16 – Decision Tree sample data (example)

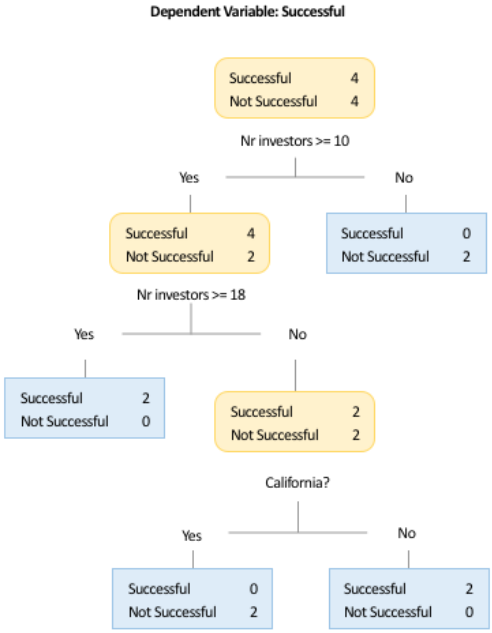


Figure 13 – Decision Tree sample tree (example)

First, the model checks if a company has less than 10 investors. If so, they're classified as **not successful**. If not, it sees if the number of investors is over 17. If so, they're **successful companies**. If not, the last partitions ask if the company has its headquarters in California. With these three questions, one can use a company's location and number of investors to classify all the observations as successful or not. I.e., if a company has more than 17 investors, the model predicts that it is successful. This is a very simplistic and convenient representation of a Decision Tree, used here to explain how the process goes. In real-world, the dataset would have successful companies with very few investors or none. Also, the above example doesn't consider the application of this technique algorithmically, having no definition for settings such as stopping criteria, pruning method or purity measure. In real world, the model would have to be optimized to make the most possible correct predictions.

Introduced by Ho in 1995 and later further developed by Leo Breiman and Adele Cutler (who combined it with the meta-algorithm, "bootstrap aggregating - *bagging*" (Leo Breiman, 1996)) and trademarked the algorithm), a **Random Forest** is an ensemble classifier using multiple **Decision Tree** models. It can be used for both **classification** and **regression** problems and its accuracy and variable importance information is provided with the results. Its trees are grown using binary divisions of random features. (Leo Breiman, 2001; Ho, 1995). Its main characteristic, the randomness of the algorithm, is divided in two levels:

- At **observation** level, as each one of the decision trees gets a random sample of the bootstrapped (with replacement in the original data) training data (bagging). Meaning, each of these trees will be trained independently on ***n'*** randomly chosen rows out of ***n*** rows of data, achieving different results in terms of predictions.

- At **feature** level, not all ***M*** features (columns) are passed into the training of each decision tree. Random number of features, ***m***, will be used to form the decision trees. The value of ***m*** is held constant during the forest growing process.

The trees are independent and identically distributed, in the sense that they are all fit on different re-samplings of the data and different random subsets of features.

Traditional CART (Classification and Regression Trees) (L Breiman, Friedman, Stone, & Olshen, 1984), assumes a binary representation where each tree is grown to the largest extent as there is no pruning (by default) and after all the decision trees are built, the results from each are taken and a final classification is given through voting (in classification problems) or averaging (for regressions). Given any new input, the tree passes the information by evaluating said input starting at the root node of the tree. Creating a CART model involves not only selecting input variables but also split points on those features until a suitable tree is built.

The selection of which features to use, and its split point is chosen through a greedy algorithm aiming to minimize a cost function - this is a procedure where all the values are lined up and different split points are tried and tested using a cost function. The split with the best cost (lowest cost because the objective is to minimize) is selected. The selection of the best split is usually carried out by impurity measures. The impurity of the parent node should be decreased by the split. In classification problems, the standard cost function/impurity measures, Gini (G), provides an indication of the "purity" of the nodes by testing how *mixed* the training data assigned to each node is. A node that has all classes of the same type (perfect class purity) will have G=0, where as a G that has a 50-50 split of classes for a binary classification problem (worst purity) will have a G=0.5. The algorithm proposed by Breiman and Cutler in 2001 added an additional layer to bagging while, in standard trees (CART), "each node is split using the best split among all variables. In a random forest, each node is split using the best among a sub- set of predictors randomly chosen at that node." (Leo Breiman, 2001; Liaw & Wiener, 2002)

The next step of the algorithm it to know when to stop, this is called the *Stopping Criterion*. The most common way to stop the procedure is to use a minimum count on the number of training instances assigned to each node. If the count is less than some minimum, then the split is not accepted, and the node is taken as a final node. The count of training members is tuned to the dataset defining how specific to the training data the tree will be. Too specific (i.e., a count of 1) and the tree will overfit

the training data and likely have poor performance on the test set. Although the stopping criterion is critical as it strongly influences the performance of the algorithm, one way to further lift the model's performance is to *prune* the tree after the learning. As the complexity of a **Decision Tree** is defined by the number of splits in the tree it makes simpler trees more favorable as they are easier to understand – also making them less likely to overfit. The fastest pruning method is to work each leaf node in the tree and evaluate the effect of removing it using a holdout test-set. Nodes are removed only if it results in a drop in the overall cost function on the entire test set, when no further improvements can be made the pruning stops.

8. ANNEXES

8.1. DATA ANALYSIS

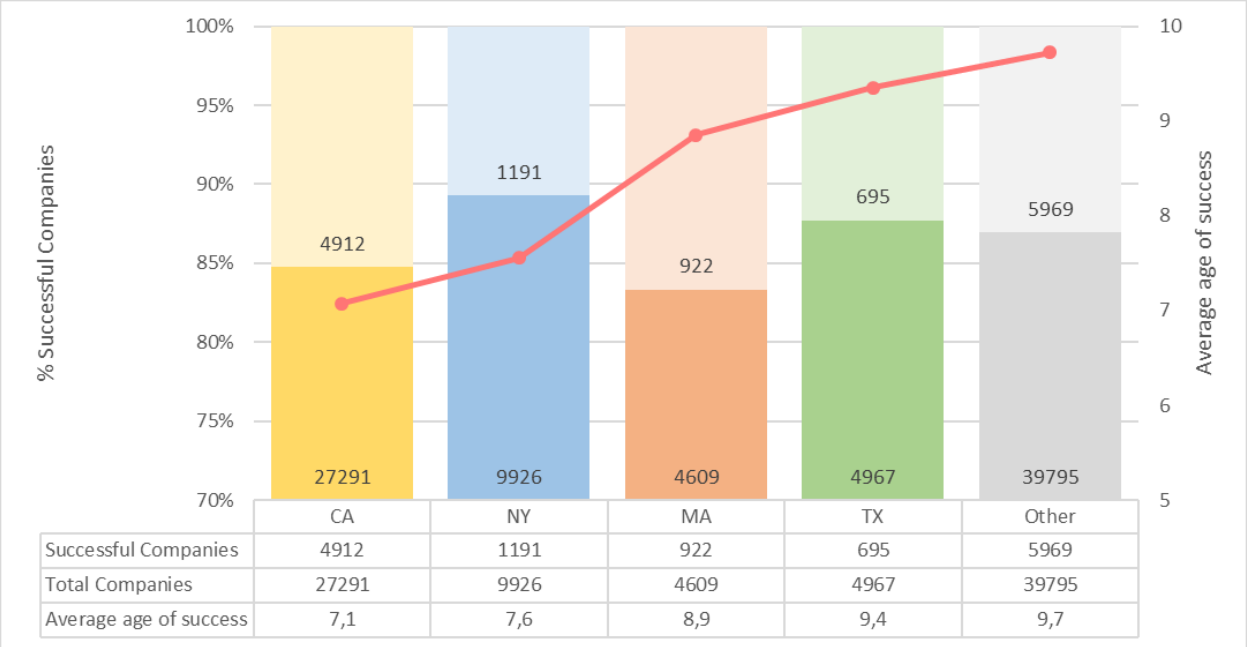


Figure 14 - Average success age full table

usa_state_code	COUNT	%	SUCCESSFUL COMPANIES	TARGET RATIO
MA	4609	5%	922	20%
TX	4967	6%	695	14%
NY	9926	11%	1191	12%
CA	27291	32%	4912	18%
Other	39795	46%	5969	15%
All	86588	100%	13690	16%

Table 17 - Successful companies per state full table

Row Labels	CA	MA	NY	Other	TX	(blank)	Grand Total
commerce	134	17	62	169	21		403
commerce(Tech)	203	34	69	213	16		535
communications(Tech)	110	15	16	72	10		223
education	10	7	5	24	0		46
education(Tech)	83	22	25	110	17		257
entertainment	145	15	66	117	13		356
entertainment(Tech)	592	70	166	346	48		1222
financial	106	25	84	375	29		619
financial(Tech)	236	54	86	318	46		740
government	8	1	4	28	4		45
government(Tech)	25	10	4	59	4		102
hardware(Tech)	487	115	44	423	71		1140
healthcare	133	33	20	298	37		521
healthcare(Tech)	414	184	54	621	53		1326
information tech(Tech)	394	96	65	388	69		1012
internet services(Tech)	173	38	21	163	20		415
lifestyle	54	12	21	137	10		234
lifestyle(Tech)	95	5	33	80	14		227
manufacturing	38	5	9	107	12		171
manufacturing(Tech)	177	37	9	151	21		395
media	137	13	64	171	17		402
media(Tech)	386	60	163	340	54		1003
mobile(Tech)	146	19	30	109	13		317
mobility	16	0	11	66	9		102
mobility(Tech)	43	6	7	56	6		118
realEstate	21	4	10	54	15		104
realEstate(Tech)	29	5	9	38	4		85
sciences(Tech)	90	35	16	173	13		327
security(Tech)	31	9	8	57	8		113
software(Tech)	324	76	56	522	56		1034
utilities&energy	46	17	13	181	118		375
utilities&energy(Tech)	65	13	10	88	45		221
(blank)							
Grand Total	**4951**	**1052**	**1260**	**6054**	**873**		**14190**

Table 18 - Successful companies - State vs Category

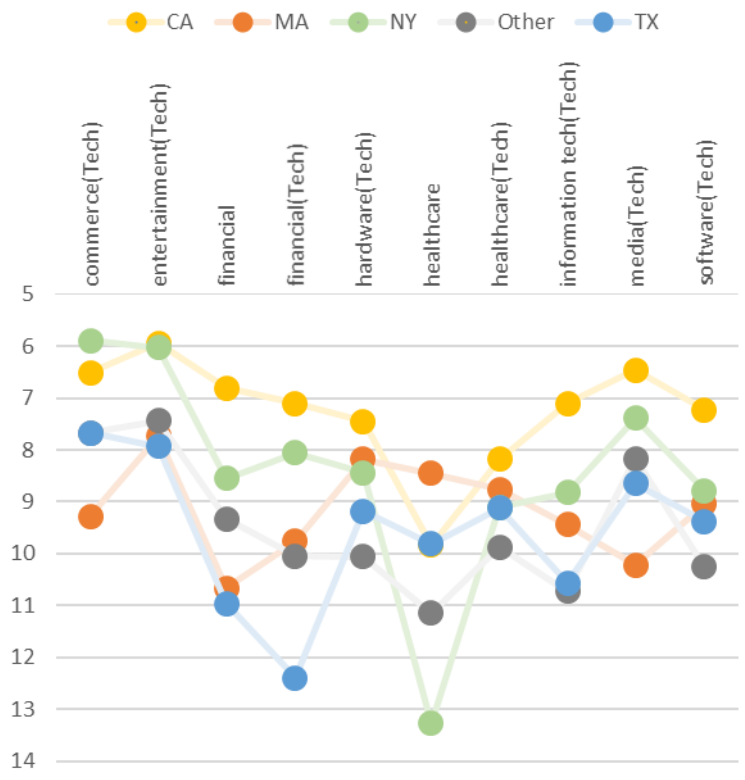
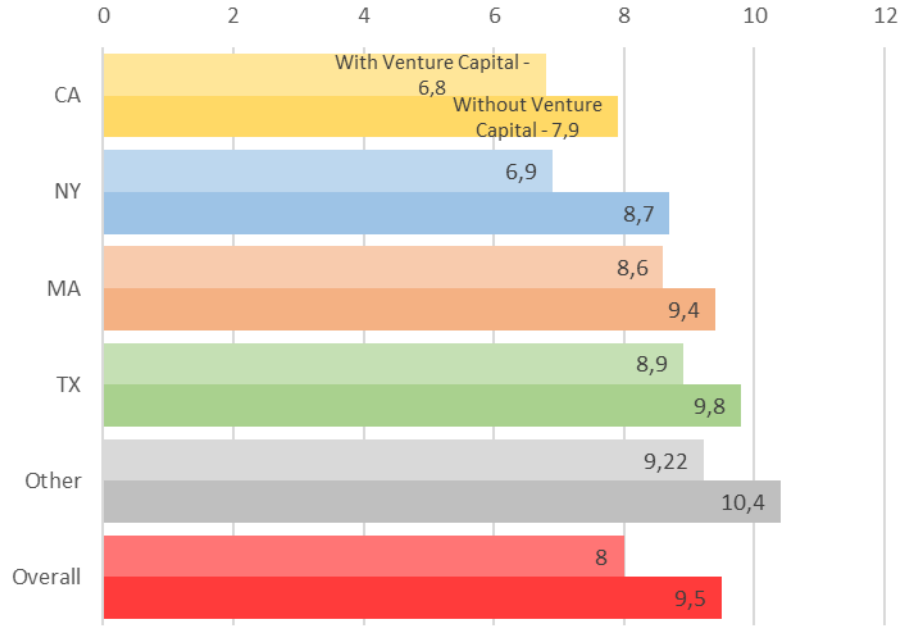
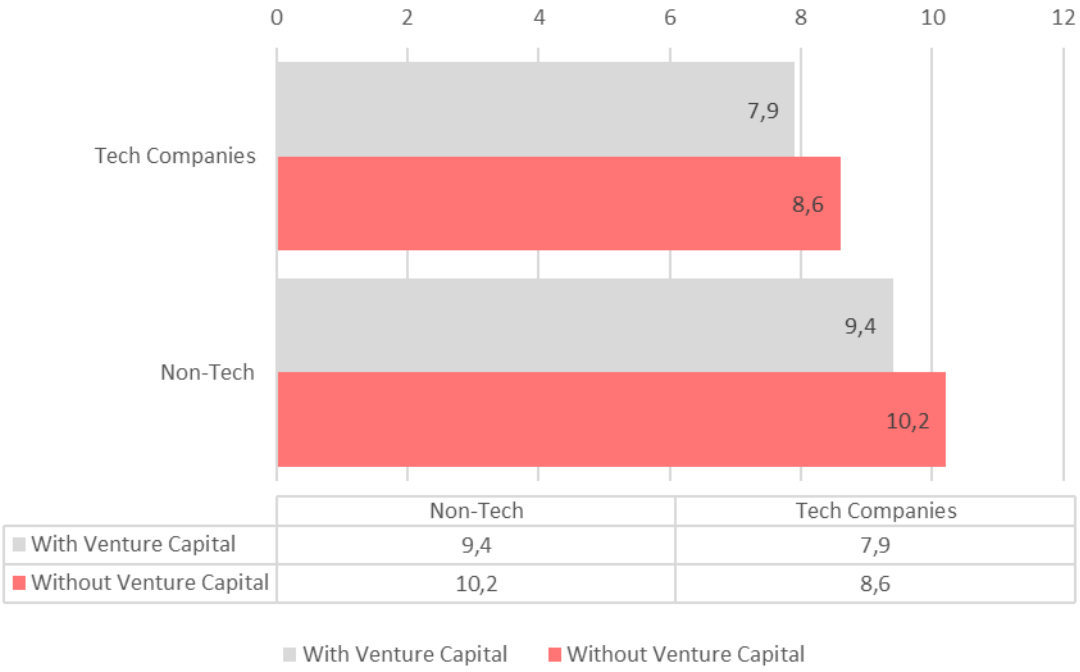


Figure 15 - Average age of success (Top 10 categories)



	Overall	Other	TX	MA	NY	CA
With Venture Capital	8	9,22	8,9	8,6	6,9	6,8
Without Venture Capital	9,5	10,4	9,8	9,4	8,7	7,9

Table 19 - Impact of VC per state



	Non-Tech	Tech Companies
With Venture Capital	9,4	7,9
Without Venture Capital	10,2	8,6

Table 20 - Impact of VC in Tech vs non-Tech

General Category	*Total Companies*	*Successful Companies*	*% Successful Companies*	*Rest of companies*
manufacturing(Tech)	1 144	395	35%	65%
healthcare(Tech)	4 699	1326	28%	72%
hardware(Tech)	4 296	1140	27%	73%
security(Tech)	450	113	25%	75%
utilities&energy	1 505	375	25%	75%
utilities&energy(Tech)	943	221	23%	77%
information tech(Tech)	4 450	1012	23%	77%
communications(Tech)	1 001	223	22%	78%
mobility(Tech)	557	118	21%	79%
government(Tech)	485	102	21%	79%
financial(Tech)	3 674	740	20%	80%
sciences(Tech)	1 630	327	20%	80%
manufacturing	911	171	19%	81%
realEstate(Tech)	457	85	19%	81%
internet services(Tech)	2 369	415	18%	82%
software(Tech)	5 912	1034	17%	83%
entertainment(Tech)	7 662	1222	16%	84%
commerce(Tech)	3 579	535	15%	85%
lifestyle(Tech)	1 581	227	14%	86%
media(Tech)	7 036	1003	14%	86%
financial	4 418	619	14%	86%

mobile(Tech)	2 444	317	13%	87%
healthcare	4 047	521	13%	87%
mobility	793	102	13%	87%
education(Tech)	2 027	257	13%	87%
government	359	45	13%	87%
media	3 835	402	10%	90%
realEstate	1 132	104	9%	91%
entertainment	3 994	356	9%	91%
commerce	4 882	403	8%	92%
lifestyle	3 041	234	8%	92%
education	1 275	46	4%	96%
	86 588	14 190		

Table 21 - Successful companies per category

Name	*Description*	*Sparsity Level (Avg=69%)*	*Average*	*Min*	*Max*	*Type*
roundD	The company did a round D	98,8%				binary
roundC	The company did a round C	95,9%				binary
roundB	The company did a round B	92,5%				binary
roundA	The company did a round A	88,8%				binary
VentureCapital	Has venture capital (with missing values)	60,7%				binary
isTech	Is a tech company	0,0%				binary
target	The company was acquired by other or went to a public stock market (IPO)	0,0%				binary
roundD_raised_amount	Raised amount of Round D	98,8%	$40.449.855	$12.500.000	$14.000.000.000	continuous
roundC_raised_amount	Raised amount of Round C	96,0%	$21.162.205	$76.265	$793.500.000	continuous
roundB_raised_amount	Raised amount of Round B	92,9%	$14.968.688	$19.208	$542.000.000	continuous
roundA_raised_amount	Raised amount of Round A	89,7%	$7.640.412	$7.000	$500.000.000	continuous
investment_per_round	Total US Dollars invested per round of investment	61,5%	$9.161.589	$1.000	$2.581.256.716	continuous
funding_total_usd	Total funding in US dollars	61,5%	$21.646.508	$1.000	$11.457.450.000	continuous
roundD_age	Company's age when it did its round D	98,8%	6,5	1	25	discrete

total_investments	Total number of investments made by the company	98,5%	2,7	1	178	discrete
customer_count	Number of customers	98,3%	6,0	1	87	discrete
ipo_age	Company's age when it went to a public stock market	96,4%	7,7	1	31	discrete
roundC_age	Company's age when it did its round C	95,9%	5,4	1	26	discrete
total_acquisitions	Number of acquisitions made by the company	93,9%	2,2	1	211	discrete
competitor_acquired_ipo	Number of competitors either acquired or IPO'd	93,1%	2,0	1	23	discrete
roundB_age	Company's age when it did its round B	92,5%	4,1	1	29	discrete
roundA_age	Company's age when it did its round A	88,7%	3,3	1	29	discrete
competitor_count	Number of competitors	87,8%	3,2	1	55	discrete
age_Acquired	Company's age when acquired	87,1%	9,2	1	31	discrete
success_age	Age of company when it got acquired or went to a public stock market (IPO)	84,3%	8,6	1	31	discrete
top500_investor	Number of top500 investors in the company	81,2%	3,5	1	37	discrete
investors_per_round	Number of investors per round	70,0%	2,3	1	39	discrete
total_exp_founders_years	Total experience of founders in years	70,0%	9,5	1	102	discrete
age_first_funding_year	Company's age when it received first funding	53,9%	3,2	0	31	discrete
funding_rounds	Number of funding rounds	53,9%	2,1	1	24	discrete
totalFounders	Number of founders	41,4%	1,7	1	19	discrete
total_experience_jobs_years	Total experience of total jobs in the company in years	28,8%	12,0	0	5859	discrete
totaljobs	Total jobs of the company	23,9%	5,3	1	2645	discrete

age_yrs	Actual age in years	0,0%	10,3	3	32	discrete
employee_count_ordinal	Ordinal feature to classify interval of employees of company (Where, 0 = Missing Values and 10 = [1001;100000] employees)	0,0%	1,8	0	9	discrete
category_general	One of thirty-two different categories of the company	0,0%				nominal
usa_state_code	One of five different states (CA, NY, MA, TX, Other)	0,0%				nominal

Table 22 - Features before transformations (Full table)

8.2. FINAL MATRIX

Confusion Matrix	*0*	*1*
0	22649	2030
1	1283	22776

Table 23 - RF's General model confusion matrix

8.3. FINAL FEATURES

#	Feature name	Type
1	*usa_state_code=CA*	Binary
2	*usa_state_code=Other*	Binary
3	*usa_state_code=NY*	Binary
4	*usa_state_code=TX*	Binary
5	*usa_state_code=MA*	Binary
6	*category_general=mobility(Tech)*	Binary
7	*category_general=hardware(Tech)*	Binary
8	*category_general=entertainment(Tech)*	Binary
9	*category_general=commerce*	Binary
10	*'category_general=information tech(Tech)'*	Binary
11	*category_general=healthcare(Tech)*	Binary
12	*category_general=utilities&energy(Tech)*	Binary
13	*category_general=financial*	Binary
14	*category_general=education*	Binary
15	*category_general=utilities&energy*	Binary
16	*category_general=commerce(Tech)*	Binary

17	*category_general=media(Tech)*	Binary
18	*category_general=media*	Binary
19	*category_general=realEstate(Tech)*	Binary
20	*category_general=financial(Tech)*	Binary
21	*category_general=entertainment*	Binary
22	*category_general=education(Tech)*	Binary
23	*category_general=healthcare*	Binary
24	*category_general=lifestyle*	Binary
25	*category_general=mobility*	Binary
26	*category_general=communications(Tech)*	Binary
27	*category_general=lifestyle(Tech)*	Binary
28	*category_general=software(Tech)*	Binary
29	*category_general=manufacturing(Tech)*	Binary
30	*'category_general=internet services(Tech)'*	Binary
31	*category_general=manufacturing*	Binary
32	*category_general=government*	Binary
33	*category_general=government(Tech)*	Binary
34	*category_general=mobile(Tech)*	Binary
35	*category_general=security(Tech)*	Binary
36	*category_general=realEstate*	Binary
37	*category_general=sciences(Tech)*	Binary
38	*'age_first_funding_year=\'(-inf-0.5]\''*	Binary
39	*'age_first_funding_year=\'(0.5-1.5]\''*	Binary
40	*'age_first_funding_year=\'(1.5-3.5]\''*	Binary
41	*'age_first_funding_year=\'(3.5-inf)\''*	Binary
42	*'employee_count_ordinal=\'(-inf-0.5]\''*	Binary
43	*'employee_count_ordinal=\'(0.5-1.5]\''*	Binary
44	*'employee_count_ordinal=\'(1.5-2.5]\''*	Binary
45	*'employee_count_ordinal=\'(2.5-inf)\''*	Binary
46	*'competitor_count=\'(-inf-1.5]\''*	Binary
47	*'competitor_count=\'(1.5-2.5]\''*	Binary
48	*'competitor_count=\'(2.5-4.5]\''*	Binary
49	*'competitor_count=\'(4.5-inf)\''*	Binary
50	*'competitor_acquired_ipo=\'(-inf-1.5]\''*	Binary
51	*'competitor_acquired_ipo=\'(1.5-2.5]\''*	Binary
52	*'competitor_acquired_ipo=\'(2.5-3.5]\''*	Binary
53	*'competitor_acquired_ipo=\'(3.5-inf)\''*	Binary
54	*'customer_count=\'(-inf-1.5]\''*	Binary
55	*'customer_count=\'(1.5-4.5]\''*	Binary
56	*'customer_count=\'(4.5-8.5]\''*	Binary
57	*'customer_count=\'(8.5-inf)\''*	Binary
58	*'funding_rounds=\'(-inf-1.5]\''*	Binary
59	*'funding_rounds=\'(1.5-2.5]\''*	Binary
60	*'funding_rounds=\'(2.5-3.5]\''*	Binary
61	*'funding_rounds=\'(3.5-inf)\''*	Binary
62	*'investors_per_round=\'(-inf-1.5]\''*	Binary

63	*'investors_per_round=\'(1.5-2.5]\''*	Binary
64	*'investors_per_round=\'(2.5-3.5]\''*	Binary
65	*'investors_per_round=\'(3.5-inf)\''*	Binary
66	*'top500_investor=\'(-inf-1.5]\''*	Binary
67	*'top500_investor=\'(1.5-2.5]\''*	Binary
68	*'top500_investor=\'(2.5-5.5]\''*	Binary
69	*'top500_investor=\'(5.5-inf)\''*	Binary
70	*'investment_per_round=\'(-inf-499900]\''*	Binary
71	*'investment_per_round=\'(499900-2083484]\''*	Binary
72	*'investment_per_round=\'(2083484-8001666.5]\''*	Binary
73	*'investment_per_round=\'(8001666.5-inf)\''*	Binary
74	*'funding_total_usd=\'(-inf-631250]\''*	Binary
75	*'funding_total_usd=\'(631250-3500002.5]\''*	Binary
76	*'funding_total_usd=\'(3500002.5-16248967.5]\''*	Binary
77	*'funding_total_usd=\'(16248967.5-inf)\''*	Binary
78	*'total_investments=\'(-inf-1.5]\''*	Binary
79	*'total_investments=\'(1.5-2.5]\''*	Binary
80	*'total_investments=\'(2.5-5.5]\''*	Binary
81	*'total_investments=\'(5.5-inf)\''*	Binary
82	*'total_acquisitions=\'(-inf-1.5]\''*	Binary
83	*'total_acquisitions=\'(1.5-2.5]\''*	Binary
84	*'total_acquisitions=\'(2.5-4.5]\''*	Binary
85	*'total_acquisitions=\'(4.5-inf)\''*	Binary
86	*'total_experience_jobs_years=\'(-inf-0.5]\''*	Binary
87	*'total_experience_jobs_years=\'(0.5-6.5]\''*	Binary
88	*'total_experience_jobs_years=\'(6.5-15.5]\''*	Binary
89	*'total_experience_jobs_years=\'(15.5-inf)\''*	Binary
90	*'totalFounders=\'(-inf-1.5]\''*	Binary
91	*'totalFounders=\'(1.5-2.5]\''*	Binary
92	*'totalFounders=\'(2.5-3.5]\''*	Binary
93	*'totalFounders=\'(3.5-inf)\''*	Binary
94	*'total_exp_founders_years=\'(-inf-4.5]\''*	Binary
95	*'total_exp_founders_years=\'(4.5-7.5]\''*	Binary
96	*'total_exp_founders_years=\'(7.5-12.5]\''*	Binary
97	*'total_exp_founders_years=\'(12.5-inf)\''*	Binary
98	*'totaljobs=\'(-inf-1.5]\''*	Binary
99	*'totaljobs=\'(1.5-2.5]\''*	Binary
100	*'totaljobs=\'(2.5-5.5]\''*	Binary
101	*'totaljobs=\'(5.5-inf)\''*	Binary
102	*'roundA_age=\'(-inf-1.5]\''*	Binary
103	*'roundA_age=\'(1.5-2.5]\''*	Binary
104	*'roundA_age=\'(2.5-4.5]\''*	Binary
105	*'roundA_age=\'(4.5-inf)\''*	Binary
106	*'roundA_raised_amount=\'(-inf-2497500]\''*	Binary
107	*'roundA_raised_amount=\'(2497500-4999999.5]\''*	Binary
108	*'roundA_raised_amount=\'(4999999.5-8220000]\''*	Binary

109	*'roundA_raised_amount=\'(8220000-inf)\''*	Binary
110	*'roundB_age=\'(-inf-1.5]\''*	Binary
111	*'roundB_age=\'(1.5-2.5]\''*	Binary
112	*'roundB_age=\'(2.5-4.5]\''*	Binary
113	*'roundB_age=\'(4.5-inf)\''*	Binary
114	*'roundB_raised_amount=\'(-inf-5600375]\''*	Binary
115	*'roundB_raised_amount=\'(5600375-10000000.5]\''*	Binary
116	*'roundB_raised_amount=\'(10000000.5-17990805]\''*	Binary
117	*'roundB_raised_amount=\'(17990805-inf)\''*	Binary
118	*'roundC_age=\'(-inf-3.5]\''*	Binary
119	*'roundC_age=\'(3.5-4.5]\''*	Binary
120	*'roundC_age=\'(4.5-6.5]\''*	Binary
121	*'roundC_age=\'(6.5-inf)\''*	Binary
122	*'roundC_raised_amount=\'(-inf-8021874.5]\''*	Binary
123	*'roundC_raised_amount=\'(8021874.5-14900000]\''*	Binary
124	*'roundC_raised_amount=\'(14900000-25000008]\''*	Binary
125	*'roundC_raised_amount=\'(25000008-inf)\''*	Binary
126	*'roundD_age=\'(-inf-4.5]\''*	Binary
127	*'roundD_age=\'(4.5-5.5]\''*	Binary
128	*'roundD_age=\'(5.5-7.5]\''*	Binary
129	*'roundD_age=\'(7.5-inf)\''*	Binary
130	*'roundD_raised_amount=\'(-inf-19850000]\''*	Binary
131	*'roundD_raised_amount=\'(19850000-26349003]\''*	Binary
132	*'roundD_raised_amount=\'(26349003-43100000]\''*	Binary
133	*'roundD_raised_amount=\'(43100000-inf)\''*	Binary
134	*VentureCapital*	Binary
135	*'age_yrs=\'(-inf-5.5]\''*	Binary
136	*'age_yrs=\'(5.5-8.5]\''*	Binary
137	*'age_yrs=\'(8.5-14.5]\''*	Binary
138	*'age_yrs=\'(14.5-inf)\''*	Binary
139	*'isTech=\'(0.5-inf)\''*	Binary
140	*'has_founders_experience=\'(0.5-inf)\''*	Binary
141	*'hasFunding=\'(0.5-inf)\''*	Binary
142	*'hasInvestmentIn=\'(0.5-inf)\''*	Binary
143	*'hasAcquired=\'(0.5-inf)\''*	Binary
144	*'hasCustomers=\'(0.5-inf)\''*	Binary
145	*'hasExperience=\'(0.5-inf)\''*	Binary
146	*'hasInvested=\'(0.5-inf)\''*	Binary
147	*'hasCompetitor=\'(0.5-inf)\''*	Binary
148	*'hasSuccessfulCompetitor=\'(0.5-inf)\''*	Binary
149	*'hasFounder=\'(0.5-inf)\''*	Binary
150	*'hasTop500investor=\'(0.5-inf)\''*	Binary
151	*'hasJobs=\'(0.5-inf)\''*	Binary
152	*'hasFundingRound=\'(0.5-inf)\''*	Binary
153	*'hasVC=\'(0.5-inf)\''*	Binary
154	*'has_roundA=\'(0.5-inf)\''*	Binary

155	*'has_roundB=\'(0.5-inf)\''*	Binary
156	*'has_roundC=\'(0.5-inf)\''*	Binary
157	*'has_roundD=\'(0.5-inf)\''*	Binary
158	*target*	Binary

Table 24 – Final features

8.4. PYTHON SCRIPTS

8.4.1. General Model

```
# Load libraries
import pandas
import sklearn
from pywFM import FM
from pandas.tools.plotting import scatter_matrix
import matplotlib.pyplot as plt
import numpy as np
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.svm import LinearSVC
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from sklearn.metrics import auc
from itertools import cycle
from sklearn import metrics
from sklearn import svm, datasets
from sklearn.metrics import roc_curve, auc
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
from scipy import interp
from ggplot import *
from statsmodels.compat import pandas as pd
from pandas import *

URL = "/Users/franciscobento/Google
Drive/Tese/Data/crunchbaseworld/datasets/crunchbase_final_Discretized_SMOTE.csv"

dataset2 = pandas.read_csv(URL, sep=",")
# shape
print(dataset2.shape)

# head
print(dataset2.head(20))

# Split out validation dataset
array = dataset2.values
X = array[:, 3:155].tolist()  # load the dataset
Y = array[:, 156].tolist()  # define the target variable (dependent variable) as Y
names = list(dataset2.columns.values)

# python doesnt accept strings although random trees does.
# in order to face this problem all categorical features were turned in binary features
# (i.e.category is now a group of 32 variables and state_code is a group of 5)
validation_size = 0.33
seed = 7
scoring = 'accuracy'
```

```
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, Y, test_size=validation_size,
                                                                    random_state=seed)

# Spot Check Algorithms
models = []
models.append(('LR', LogisticRegression()))
models.append(('SVM', LinearSVC(dual=False)))
models.append(('RANDOM FOREST', RandomForestClassifier(n_estimators=50)))

# Evaluate each model in turn
results = []
names = []
for name, model in models:
    kfold = model_selection.KFold(n_splits=10, random_state=seed)
    cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print (msg)

# Compare Algorithms
fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()


# # Make predictions on validation/test dataset
# LogisticRegression
print("LogisticRegression")
lr = LogisticRegression()
lr_model = lr.fit(X_train, y_train)
lr_predictions = lr.predict(X_test)  # classify X_test
print (accuracy_score(y_test, lr_predictions))
lr_confusion = confusion_matrix(y_test, lr_predictions)
# row, column
TP = lr_confusion[1, 1]
TN = lr_confusion[0, 0]
FP = lr_confusion[0, 1]
FN = lr_confusion[1, 0]
# Classification Error: Overall, how often is the classifier incorrect?
classification_error = (FP + FN) / float(TP + TN + FP + FN)
print('Classification Error: ' + str(classification_error))
#
# Sensitivity/Recall/TPR: When the actual value is positive, how often is the prediction correct?
sensitivity = TP / float(FN + TP)
print('Sensitivity/TPR: '+ str(sensitivity))
#
lr_roc_score = roc_auc_score(y_test, lr_predictions)
lr_fpr, lr_tpr, lr_threshold = metrics.roc_curve(y_test, lr_predictions)
lr_roc_auc = metric.auc(lr_fpr, lr_tpr)
print('LR ROC AUC SCORE: ' + str(lr_roc_score))
print(classification_report(y_test, lr_predictions))

# SVM
svm = LinearSVC()
svm_model= svm.fit(X_train, y_train)
svm_predictions = svm.predict(X_test)  # classify X_test
print(accuracy_score(y_test, svm_predictions))
print(confusion_matrix(y_test, svm_predictions))
print(classification_report(y_test, svm_predictions))
svm_fpr, svm_tpr, svm_threshold = metrics.roc_curve(y_test, svm_predictions)
svm_roc_auc = metric.auc(svm_fpr, svm_tpr)
#
# # Random Forest
rForest = RandomForestClassifier(n_estimators=50, max_features='sqrt')
rf_model = rForest.fit(X_train, y_train)
rf_predictions = rForest.predict(X_test)  # classify X_test
print(accuracy_score(y_test, rf_predictions))
print(confusion_matrix(y_test, rf_predictions))
print(classification_report(y_test, rf_predictions))
```

```
rf_fpr, rf_tpr, rf_threshold = metrics.roc_curve(y_test, rf_predictions)
rf_roc_score = roc_auc_score(y_test, rf_predictions)
rf_roc_auc = metric.auc(rf_fpr, rf_tpr)
#
print "Features sorted by their score:"
print sorted(zip(map(lambda x: round(x, 3), rForest.feature_importances_), names),
             reverse=True)
# # plot ROC CURVE
import matplotlib.pyplot as plt
plt.title('Receiver Operating Characteristic')
plt.plot(rf_fpr, rf_tpr, 'r', label = 'AUC RF = %0.3f' % rf_roc_auc)
plt.plot(svm_fpr, svm_tpr, 'g', label = 'AUC SVM = %0.3f' % svm_roc_auc)
plt.plot(lr_fpr, lr_tpr, 'b', label = 'AUC LR = %0.3f' % lr_roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1],'y--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

Figure 16 - Python script (General model)

8.4.2. Model per state/category

```
# Load libraries
import pandas
import sklearn
from pywFM import FM
from pandas.tools.plotting import scatter_matrix
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.svm import LinearSVC

URL = "/Users/franciscobento/Google Drive/Tese/Data/crunchbaseworld/datasets/CB_FINAL_CATEGORY.csv"
# Change file to CB_FINAL_STATE.csv to generate model per state

dataset2 = pandas.read_csv(URL, sep=";")
# shape
print(dataset2.shape)

print(dataset2.head(20))

grouped_category = dataset2.groupby('new_category', sort=False).size().order(ascending=False)
print grouped_category


unique_category = dataset2['new_category'].unique().tolist() # Change variable to usa_state_code for
model per state
print(unique_category)

for val in unique_category:
    dataset3 = dataset2.loc[(dataset2['new_category'] == val)] # Change variable to usa_state_code for
model per state

    array = dataset3.values
```

```
    X = array[:, 1:121].tolist()
    Y = array[:, 122].tolist()

    # Split out validation dataset
    validation_size = 0.33
    seed = 7
    X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y,
test_size=validation_size, random_state=seed)

    # Test options and evaluation metric
    scoring = 'accuracy'
    # Spot Check Algorithms
    models = []
    models.append(('RANDOM FOREST', RandomForestClassifier(n_estimators=50)))

    # Evaluate each model in turn
    print(val)
    results = []
    names = []
    for name, model in models:
        kfold = model_selection.KFold(n_splits=10, random_state=seed)
        cv_results = model_selection.cross_val_score(model, X, Y, cv=kfold, scoring=scoring)
        results.append(cv_results)
        names.append(name)
        msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
        print (msg)


    print("RandomForest")
    rForest = RandomForestClassifier(n_estimators=50)
    rForest.fit(X_train, Y_train)
    predictions40 = rForest.predict(X_validation)
    print(accuracy_score(Y_validation, predictions40))
    print(confusion_matrix(Y_validation, predictions40))
    print(classification_report(Y_validation, predictions40))
```

Figure 17 - Python script to generate models per state/category

8.5. SQL Queries

8.5.1. Discretization of employee_count:

```
update organizations org
set employee_count_ordinal  =

case WHEN Org.employee_count  = '(-)' then 0
WHEN Org.employee_count  = '(1-10)' then 1
WHEN Org.employee_count  = '(11-50)' then 2
WHEN Org.employee_count  = '(51-100)' or Org.employee_count  = '(51-200)'  then 3
WHEN Org.employee_count  = '(101-250)' then 4
WHEN Org.employee_count  = '(201-500)' or Org.employee_count  = '(251-500)'  then 5
WHEN Org.employee_count  = '(501-1000)' then 6
WHEN Org.employee_count  = '(1001-5000)' then 7
WHEN Org.employee_count  = '(5001-10000)' then 8
WHEN Org.employee_count  = '(10001-1000000)' then 9

    else 0 end;

select * from organizations;

select distinct employee_count_ordinal from organizations order by 1 desc;

|
```

8.5.2. Age of Acquisition and IPO

```
#To correctly calculate age
#Acquisitions
update organizations
INNER JOIN (
    select acquiree_uuid, acquired_on
    from acquisitions_ready_to_go
) x ON uuid = x.acquiree_uuid
SET organizations.acquired_on = x.acquired_on;

#IPO
update organizations, ipos_ready_to_go
set organizations.ipo_on = ipos_ready_to_go.went_public_on
where organizations.uuid = ipos_ready_to_go.company_uuid;

```

8.5.3. Set Tech companies and final category (1 per company)

```
# isTech
update organizations, organizations_xl_only_categories
set organizations.isTech = organizations_xl_only_categories.isTech
where organizations.uuid = organizations_xl_only_categories.uuid;

# category
update organizations, organizations_xl_only_categories
set organizations.category = organizations_xl_only_categories.category
where organizations.uuid = organizations_xl_only_categories.uuid;

#
```

8.5.4. Number of customers per company:

```
update organizations, customers_count
set organizations.customer_count = customers_count.customer_count
where customers_count.entity_uuid = organizations.uuid;

```

8.5.5. Investors per funding round, Average investors per round, Average investment (in dollars) per funding round:

```
#Investors per funding round
UPDATE organizations
INNER JOIN(
    SELECT
        company_uuid,
        avg(investor_count)AS avgInv
    FROM
        funding_rounds_ready
    GROUP BY
        company_uuid
)x ON uuid = x.company_uuid
SET organizations.investors_per_round = x.avgInv;

# Total average of investors per round where count > 1 (INFO)
SELECT
    avg(investors_per_round)AS investorRound
FROM
    organizations
WHERE
    investors_per_round > 0

#avg Investment per funding round
UPDATE organizations
INNER JOIN(
    SELECT
        company_uuid,
        avg(raised_amount_usd)AS avgInvestment
    FROM
        funding_rounds_ready
    GROUP BY
        company_uuid
)x ON uuid = x.company_uuid
SET organizations.investment_per_round = x.avgInvestment;

# Total avg investment per funding round
SELECT
    avg(investment_per_round)AS investmentRound
FROM
    organizations
where investment_per_round > 0
```

8.5.6. Number of founders, has founder, number of months of experience (sum of jobs), founders experience (sum of jobs), total number of jobs:

```
#1) Total Founders (includes CEO, FOUNDER)
update organizations
inner join (
select j.org_uuid, sum(j.isFounderCEO) as total
from jobs2_ready as j
GROUP BY org_uuid
) x on uuid = x.org_uuid
set organizations.totalFounders = x.total

#2) has founder
update organizations
inner join (
select org_uuid, sum(isFounderCEO) as total
from jobs2_ready as j
GROUP BY org_uuid
) x on uuid = x.org_uuid
set organizations.hasFounder = 1

#2.1 set Has_Founder = 0 when total_founders = 0 as it can include sums of 0 in 2)
update organizations
set hasFounder = totalFounders
where totalFounders = 0;

# cleaning columns
update organizations set totalFounders=0 where totalFounders is null

#3) Experience of jobs
update organizations
INNER JOIN (
    select jobs.org_uuid, sum(jobs.age_months) as total
    from jobs2_ready as jobs
    group by org_uuid
) x ON uuid = x.org_uuid
SET organizations.total_experience_jobs = x.total;
```

```
#2.1 set Has_Founder = 0 when total_founders = 0 as it can include sums of 0 in 2)
update organizations
set hasFounder = totalFounders
where totalFounders = 0;

# cleaning columns
update organizations set totalFounders=0 where totalFounders is null

#3) Experience of jobs
update organizations
INNER JOIN (
    select jobs.org_uuid, sum(jobs.age_months) as total
    from jobs2_ready as jobs
    group by org_uuid
) x ON uuid = x.org_uuid
SET organizations.total_experience_jobs = x.total;

#4) Founders experience
update organizations
inner join (
select jobs.org_uuid, sum(jobs.age_months) as total
from jobs2_ready as jobs
where jobs.isFounderCEO = 1
group by jobs.org_uuid
) x ON uuid=x.org_uuid
SET organizations.total_experience_founders = x.total

#5 Total jobs
update organizations
inner join (
select j.org_uuid, count(person_uuid) AS total
from jobs2_ready AS j
group by j.org_uuid
) x ON uuid=x.org_uuid
SET organizations.totaljobs = x.total
#
```

8.5.7. Number of competitors, was competitor acquired or IPO

```
# number of competitors
update organizations, competitors_count_unique_values
set competitor_count = nr_competitors
where competitors_count_unique_values.entity_uuid = organizations.uuid;

# is competitor acquired or ipo?
update competitors_without_duplicates_ready, organizations
set competitor_acquired_ipo = target
where organizations.uuid = competitor_uuid;

# number of competitors that got acquired (organizations)
update organizations
INNER JOIN (
    select entity_uuid, SUM(competitor_acquired_ipo) as total
    from competitors_without_duplicates_ready
    group by entity_uuid
) x ON uuid = x.entity_uuid
SET organizations.competitor_acquired_ipo = x.total;
```

8.5.8. Round A, B, C, D: has round, date of round, raised amount

```
#Has Venture
update organizations
inner join (
    select company_uuid, has_seed_angel_venture
    from funding_rounds_hasVenture
) x on uuid = x.company_uuid
set organizations.hasVentureCapital = x.has_seed_angel_venture

#ROUND A
# has
update organizations
INNER JOIN (
    select company_uuid, funding_round_A
    from funding_rounds_ROUND_A
) x ON uuid = x.company_uuid
SET organizations.roundA = x.funding_round_A;
#date
update organizations
INNER JOIN (
    select company_uuid, announced_on
    from funding_rounds_ROUND_A
) x ON uuid = x.company_uuid
SET organizations.roundA_date = x.announced_on;
#raised amount
update organizations
INNER JOIN (
    select company_uuid, raised_amount_usd
    from funding_rounds_ROUND_A
) x ON uuid = x.company_uuid
SET organizations.roundA_raised_amount = x.raised_amount_usd;

#ROUND B
#has
update organizations
left JOIN (
    select company_uuid, funding_round_B
    from funding_rounds_ROUND_B
) x ON uuid = x.company_uuid
SET organizations.roundB = x.funding_round_B;
```

```
#date
update organizations
left JOIN (
    select company_uuid, announced_on
    from funding_rounds_ROUND_B
) x ON uuid = x.company_uuid
SET organizations.roundB_date = x.announced_on;
#raised amount
update organizations
left JOIN (
    select company_uuid, raised_amount_usd
    from funding_rounds_ROUND_B
) x ON uuid = x.company_uuid
SET organizations.roundB_raised_amount = x.raised_amount_usd;

#ROUND C
#has
update organizations
INNER JOIN (
    select company_uuid, funding_round_C
    from funding_rounds_ROUND_C
) x ON uuid = x.company_uuid
SET organizations.roundC = x.funding_round_C;
#date
update organizations
INNER JOIN (
    select company_uuid, announced_on
    from funding_rounds_ROUND_C
) x ON uuid = x.company_uuid
SET organizations.roundC_date = x.announced_on;
#raised amount
update organizations
INNER JOIN (
    select company_uuid, raised_amount_usd
    from funding_rounds_ROUND_C
) x ON uuid = x.company_uuid
SET organizations.roundC_raised_amount = x.raised_amount_usd;
```

```
SET organizations.roundC_raised_amount = x.raised_amount_usd;

#ROUND D
# has
update organizations
INNER JOIN (
    select company_uuid, funding_round_code
    from funding_rounds_ROUND_D
) x ON uuid = x.company_uuid
SET organizations.roundD = x.funding_round_code;
# date
update organizations
INNER JOIN (
    select company_uuid, announced_on
    from funding_rounds_ROUND_D
) x ON uuid = x.company_uuid
SET organizations.roundD_date = x.announced_on;
# raised amount
update organizations
INNER JOIN (
    select company_uuid, raised_amount_usd
    from funding_rounds_ROUND_D
) x ON uuid = x.company_uuid
SET organizations.roundD_raised_amount = x.raised_amount_usd;


```

8.5.9. Number of top500 investors (by investments made)

```
#Top 500 investors
select uuid
from investors_ready as i
order by investment_count desc
limit 500

#Temp table of top500
CREATE TEMPORARY TABLE t1 (
select uuid
from investors_ready as i
order by investment_count desc
limit 500
);

#All investments by top 500 investors
select *
from investments_ready_to_go
inner join
(
select uuid
from investors_ready as i
order by investment_count desc
limit 500
) x on investments_ready_to_go.investor_uuid = x.uuid

#update column #investor_top500 per funding round
update funding_rounds_ready
left join (
select inv.funding_round_uuid as fr_uuid, count(investor_uuid) as total
from investments_ready_to_go as inv
where inv.investor_uuid in
(
select i.uuid
from t1 as i
) group by fr_uuid
) x on funding_rounds_ready.funding_round_uuid = x.fr_uuid
set investor_top500_count = x.total;
```

```
38
39  # 1 - funding rounds; sum de investor_top_500 por company
40  # 2 - organizations; where uuid = funding rounds company_uuid (DONE)
41  update organizations
42  left JOIN (
43      select rounds.company_uuid, sum(rounds.investor_top500_count) as total
44      from funding_rounds_ready as rounds
45      group by company_uuid
46  ) x ON uuid = x.company_uuid
47  SET organizations.top500_investor = x.total;
48
49  select uuid
50  from organizations
51  where top500_investor > 0;
52
```

8.5.10. Total acquisitions & total investments per company

```
1  update organizations
2  INNER JOIN (
3      select acquirer_uuid, count(acquirer_uuid) as total
4      from acquisitions_ready_to_go
5      group by acquirer_uuid
6  ) x ON uuid = x.acquirer_uuid
7  SET organizations.total_acquisitions = x.total;
```

```
1  update organizations
2  INNER JOIN (
3      select investor_uuid, count(investor_uuid) as total
4      from investments_ready_to_go
5      group by investor_uuid
6  ) x ON uuid = x.investor_uuid
7  SET organizations.total_investments = x.total;
```