Solent University

Faculty of Business, Law and Digital Technologies

MSc. Applied AI and Data Science

September 2022

Chukwuemeka Gabriel Agbo

Q15817253

"Predicting Start-up status using Funding and Sentimental data"

Supervisor : Dr. Femi Isiaq
Date of submission : September 2022

Abstract

In recent years, the prediction of start-up success has been an intriguing and difficult research topic. However, most of the research in this field has only used numerical variables from funding data to develop models, leaving out much of the important textual data from the wide range of social media sites. According to research, sentiments from social media activities can be used in improving future predictions.

Specifically, I examined a set of novel features to construct a flow within the machine learning framework. In improving previous studies further advances were made by adding the new features such as Twitter sentimental and profile data. I combined financial information from CrunchBase, one of the largest public databases for start-ups, with profile and tweet data from Twitter, one of the top and largest social media databases, to thoroughly examine, analyse and predict start-up status. A total of 2,613,123 tweets (the largest-scaled sentiment data ever recorded in the literature) from over 40,000 start-up Twitter profiles were analysed using eight specific machine learning algorithms which were all classifiers. Using Random Forest Classifier on the dataset, a True Positive Rate (TPR) of 95.5% and a False Positive Rate of 3.83% were achieved, which is the highest recorded with this data. The author's target of achieving low false negative was also attained where the false negative is 7.6%. The novel features, which emphasized the impact of social media presence/online legitimacy of a start-up, has proved important to the overall performance of the model by being among the most important features required in the final model showing the crucial role this type of data plays in predicting start-ups status.

In the end, I was able to demonstrate that a start-up status can be precisely predicted using a trained machine learning model with online legitimacy as a gauge of social acceptance based on tweets and profile data from Twitter and in combination with the start-up's funding data. In addition to attempting to predict start-up statuses, this study also contributes to the continuing debate about the significance of establishing credibility online and explains how machine learning approaches might be used in research in this area.

Contents

List of Figures

List of Tables

Chapter one

1.0 Introduction

Business start-ups are increasingly driving most country's economy. Start-up creation has increased exponentially over the past ten years in both the US and Europe. Understanding what makes these high-risk projects effective and, thus, alluring to investors and entrepreneurs, appears to be a pertinent topic. Here, the definition of success for a start-up is the occurrence that provides a sizable sum of money to the business's founders, investors, and early workers, specifically through Acquisition or an IPO (Initial Public Offering). Since the best targets are those with the potential to grow significantly soon, being able to predict success is a crucial competitive advantage for venture capitalists searching for investments. This allows investors to stay one step ahead of other competitors in funding viable business start-ups.

With the aim of developing a prediction model through supervised learning to precisely determine which start-ups are successful and which are not, secondary data accumulated by CrunchBase is explored and analysed. CrunchBase has been identified to be the largest structured database for start-ups in the world. In comparison to the current study, most studies on the prediction of business/company start-up success tend to concentrate solely on conventional measures presented by financial funding during the start-up early stage. As data technologies advance, it has become possible to manipulate data using data mining approaches and modern machine learning algorithms to define a more characterise reliable models by combining various niche and complex data, leading to highly dependable outcomes in data analysis.

In this study sentimental data was generated using archived tweets from twitter via API for each start-up listed in the dataset to provide results comparable with previous studies. The sentimental data created new features which focused on the impact of digital awareness on a company and proved pivotal to the overall performance of the model by being some of the most important feature to the final model showing the critical importance this type of data has on these start-ups.

1.1 Background

Related work on start-up prediction using funding data.

The term start-ups and entrepreneurship are used interchangeably in the modern economy because both lead to technological advancement, job creation, and economic growth. It is not unexpected that big towns are seeking to become the breeding place for innovative ideas, entrepreneurial talent, technology-driven firms, and venture capital money. Because it is essential in the funding of any start-up to understand the reasons behind investors and the techniques used by angel, venture capital, and private equity investors, researchers have invested great deal of time to find out controlling factors. This can be seen in the works done by Davila, A. et al (2003), Kortum, S. et al (2001) and Wong, A. (2002) in which these scholars do not seek to create a predictive model, instead, they concentrate on studying various financial elements surrounding a start-up. The significant amount of research of this kind examines either the rationale behind investors' decisions to offer or withhold funding capital or the objective causes of start-up failures. In essence, a start-up is a high-risk business that is still in its early stages of operation and frequently offers products or services related to technology (Ries, E., 2011).

Related work on start-up prediction using sentimental analysis.

With the advancement in technology, sentiment data is of crucial importance when working on a system that involves the public in general, because these are the target population which will give insights on what features of a solution is required and usable on a larger scale for the product. This, in my opinion, is crucial information that should be taken into account because prospective customers often turn to online presence as a resource when trying to understand the advantages of a new start-up's product. Additionally, the amount of information a start-up provides about its management, organisation, and products is seen as a major source of legitimacy (Shepherd and Zacharakis, 2003). Some marketing experts have noted that the regularity with which businesses use social media and the amount of information they share through these platforms are both favourably correlated with product sales (Clark and Melancon, 2013;

Reuber and Fischer, 2011). The effort that new businesses make to maintain relationships with and inform its stakeholders should, in the event that they do use social media as a method to establish online legitimacy, be an indication of their legitimacy.

Legitimacy, which is frequently referred to as the individual perception of a business, is closely related to credibility because it determines how a business stands in comparison to its competitors (Deephouse and Carter, 2005). It is anticipated that online credibility on Twitter is determined by not just the quantity of information provided, but also its content and tone. Thus, sharing sentimental content on Twitter can be utilised to establish clarity and reliability in order to acquire, uphold, and preserve legitimacy (Ashforth and Gibbs, 1990). According to the work done by Liew and Wang in 2016, their study shows that the performance of rising venture IPOs is substantially connected with the sentimental intensity from Twitter. Also, it is good to note that Twitter responses from the general public to new businesses show the unfiltered views and opinions of stakeholders, demonstrating the credibility of the platform (Etter et al., 2018). Previous research has demonstrated that the quantity of Twitter likes and the level of user involvement are reliable indicators of how well-established businesses communicate with their target population (Clark and Melancon, 2013; Kadam and Ayarekar, 2014). A reliable indicator of a new venture's online legitimacy is thus likely to be the number of engaged fans (twitter users) who actively share, like, or comment on the contents of the venture online.

Significant related research in this area of study is the work done by Xiang, G. et al, 2012, which deals with using data acquired from Crunchbase to train a classifier in order to predict a metric for firm success, in this case, Merger and acquisitions negotiations. This study has done justice, nonetheless, its limitations are not to be overlooked. First, it is constrained to nearly entirely using Crunchbase data, whereas my study supplements it with a substantial body of diverse and freely accessible data from both twitter and the web in general. Secondly, Xiang, G. et al. used overall data metrics for prediction, whereas in my study the model will be learning from much broader data which will be a sparsity from sentimental data.

1.2 Problem statement

For a given start-up that has received partial funding, predict whether it will succeed, acquired or fail in the nearest future using the provided funding data and complemented by sentimental data from twitter.

What inspired the research?

With the increasing number of start-ups yearly, and the need to identify start-ups with legitimate intentions for long term growth in the future, researchers have spent a lot of time trying to identify its controlling elements since it is crucial for start-up funding by investors, angel, venture capital, and private equity investors.

From review of past literature's insights, a measure of online legitimacy has been neglected to be incorporated in most research. This, as I believe and will prove in my research, is a vital data that should be considered because potential customers use online presence as reference for comprehending the benefits of a new start-up's product, the amount of information that a start-up offers about its products, management, and organisation which is seen as a main source of legitimacy (Shepherd and Zacharakis, 2003).

This model of start-up success has the flexibility to balance the investment risk-to-reward ratio for investors, which is a desirable quality. Typically, funding events—which may include major corporations like Apple, Amazon, Google, and others—are divided into rounds of ascending size, starting with the early angel and seed rounds, and continuing through series A/B/C and beyond. The more money a company has raised, the more established it is, and the more data there is on which to build a prediction. I will take into account businesses in this study that has already completed a specific sort of funding round (the trigger round) and make predictions about whether they will succeed or fail as a start-up.

1.3 Research question

For a given start-up, will it be possible to predict its future status using a machine learning model trained with the start-up partial funding and sentimental data?

1.4 Aim

The aim of this study is to train a machine learning model using funding and sentimental data of start-ups to predict if the start-up will either be successful, acquired or fail in the future.

1.5 Objectives

Previous works done on predicting start-up success/failure tend to focus more on the funding data. The main objective of this study is summarized as to bridge the gap between sentimental and funding data related to start-up businesses/companies. My objectives are laid out as follows:

- Exploratory data analysis of key features in the dataset, data transformation and feature engineering.
- Natural language processing of sentimental data from twitter, cross-plotting analyzed start-up funding data with sentimental data from twitter.
- Choose a predictive model by comparing several machine learning classification algorithms that will be suitable for the task and presenting experimental results from the models and drawing conclusions.
- Structure a cloud/SQL/NoSQL database that could hold fetched data and user inputs. Build a frontend Hybrid app that could serve a triple purpose of website, mobile app, and PC installable software to best present usage of the model to users.

1.6 Success metrics

In general, it's hard and arbitrary to define what constitutes a successful start-up. But for investors, who typically place a high value on returns on investment (ROI), the

concept of success is rather simple. Consequently, from the viewpoint of an investor, a good exit, such as an acquisition or continuity, is the gold standard.

I begin by taking into account the traditional, investor-based concept of success for the sake of this study: whether a start-up will be acquired or move forward in the operational stage. With the help of this concept of success, I will train a supervised learning machine classifier that will classify start-ups into three groups: successful, acquired, and failed. This classification is done utilising factors like geographic region, market type, funding type, and funding amount, among others. The supervised algorithm may accurately/closely predict, given specific characteristics, whether a start-up will eventually fall into any of these categories.

I would state that an F1-score and accuracy of 80% to 90% (which has very low false negative, and high true positive rate) would give me a confidence of achievement. Low percentage of false negative is vital in this study because predicting a start-up to be a failure in the future could trigger investors to withdraw from funding and may probably (high probability) cause the start-up to fail even when the start-up has positive intentions to succeed in the future.

For the evaluation criteria the Precision-Recall (PR) curve will be examined to provide a clear indication of performance quality. My interest lies in the low-recall region of the curve because in practise an investor will only be able to fund a percentage of the start-up.

1.7 Proposed artefact and societal impact

At the end of the study, the output artefacts would consist of a predictive model that is connected to an interactive GUI which users can utilize for start-up prediction.

The societal impact is focused on reducing the risk-to-reward ratio for both investors and the public. This is achieved by narrowing down to start-up businesses that has high legitimacy and intentions to increase economically in the future.

1.8 Methods

Multiple classifiers were used from which random forest classifier was selected to build and train a predictive model. This approach is based on classification and regression trees (CART) (Breiman et al., 1984).

In order to curb the problem of overfitting, random forest classifier creates n-number of random classification trees. The goal is to train a classifier for each subset by repeatedly randomly resampling the data. The overfitting of the data caused by various classifiers varies, therefore they are averaged on a wide scale (Liaw and Wiener, 2002). The random forest algorithm is very user-friendly in addition to being robust against overfitting because it only requires the researcher to determine two key parameters (the number of features to train each tree and the number of trees to train).

To avoid possible sample biases from our data collection process and to mimic the possibility of online validity of a start-up and predict it's status, a separate data collection method was used for the sentimental data from twitter. All start-up with found data from twitter is sliced and quantified separately for analysis.

Chapter two

2.0 Literature review

Definition and economic significance of start-ups

Start-up businesses are those that produce goods and venture into uncharted territory or markets. As a result, start-ups have high risks and are uncertain since a new service or product might not always appeal to its intended market and may need to undergo repeated modifications before finding its ideal market. In the end, a start-up is a business with steep risks and is currently in its early phases of growth and frequently offers products or services that are technology related (Ries, 2011).

These businesses are frequently started by the founders with their own money in an effort to profit from creating goods or services. Most of these businesses cannot be sustained over the long run without further funding from angel investors, venture capitalists, or any other investment firms due to little revenue or significant scalability expenses (as opposed to obtaining a loan from a bank). The most popular kind of start-up in the late 1990s was a "dotcom" company. Sadly, most start-ups technology failed between 1997 and 2001 during the "dotcom boom" due to serious errors in their business models, such as a lack of a reliable source of income (Geier, 2015).

Peter Thiel, the founder of PayPal, and a veteran businessman, defines a start-up as a company that develops vertical innovation rather than horizontal. In these terms horizontal innovation represents the process of globalisation of an existing technology (moving existing technology to regions that are yet to experience it), and vertical innovation represents the creation of technology that is not yet in existence.

According to Steve Blank, for every ten start-ups nine may fail, with the two most common reasons being that there is no market demand for a particular good or service and that businesses run out of money trying to maintain expenses or force the unwanted good or service on consumers. The main reasons why many random start-ups rise and fall according to Steve Blank is focused on four points; (a) Start-ups can now be built

for thousands rather than millions, (b) A higher resolute venture capital industry, (c) Entrepreneurship developing its own management science, (d) Speed of consumer adoption of new technology (Steve, 2006).

Defining start-ups success

Start-ups success is frequently characterised as a two-pronged approach, in which the business can either go public with an IPO (Public Initial Offering), enabling its investors to sell shares to the general public, or it can opt to being sold (acquired by larger companies) or merged with another business (M&A), in which case investors receive cash in exchange for their shares. The phrase "exit strategy" is frequently used to describe this procedure (Guo et. al. 2015).

The term "mergers and acquisitions" (abbreviated "M&As") refers to these transactions, which are crucial to corporate restructuring. Alam and Khan (2014) claim that a merger is a technique for combining two businesses into one (often with a new name) in order to boost profit and sales levels. This approach is more common in non-tech enterprises between businesses of comparable size and stature. Merger and acquisition are crucial for high-tech firms in particular since these sectors frequently employ M&As to acquire cutting-edge technologies or quickly increase their R&D capacities (Wei, Jiang, & Yang, 2009). "An acquisition occurs when one company buys another or when one business acquires a majority stake in another. A company that makes an effort to buy or merger with other business is referred to as an acquirer firm (Machiraju, 2007). The idea behind an M&A transaction is that two businesses have more value working together than they do alone. One of the most important corporate strategies for businesses to maintain their competitive edge is the merger of these two businesses (Machiraju, 2007; Xiang et al., 2012). According to the Thomson Reuters report, 2015 was the highest year ever for global M&A deals (Rogers, 2016).

An IPO is the first time a private company sells shares to the general public, according to Li & Liu (2010). Therefore, "going public" is a significant milestone in a company's life cycle. The businesses will expand into continuous growth as a healthy company during the post-IPO phases, get purchased prior to strong or weak operation, and be

removed from stock market list at the conclusion of their life cycle. If an IPO takes place, the business is given a stock market listing, which enables it to get more funding and ultimately allow shareholders to sell their shares to the general public. There is no ideal exit plan for a business because it strongly depends on a variety of variables, including the business' prosperity, the state of the financial markets, the flow of data among strategists, and the standard for other businesses' initial public offerings, and many more (Akerlof, Yellen, & Katz, 1970).

Being purchased or going public are typically viewed as successes for the business in the start-up ecosystem since they provide (significant) up-front cash to the start-up founders, shareholders, and early workers (Guo et al., 2015). Buying a smaller company's talent pool is one of the most popular motivations for start-ups and larger firms to purchase smaller businesses. In addition to purchasing technology, the parent firm also hires personnel. This process of acquisition offers a quick way to expand in markets with high competition (Marita Makinen, Haber, & Raymundo of Lowenstein Sandler, 2014).

Research on previous works on start-up success prediction

According to Ali-Yrkkö, Hyytinen, and Pajarinen (2005), Gugler & Konrad (2002), Meador, Church, & Rayburn (1996), the majority of study is based on predictions by analysing quantitative financial measures for big businesses, such as firm size, net income, total debt, and price to profit ratio. Some researchers have added managerial characteristics such as industry differences (Meador et al., 1996), administrative efficiency (Ali-Yrkkö et al., 2005; Meador et al., 1996), and source wealth (Meador et al., 1996). For the most part, Logistic Regression analysis was used in the creation of prediction models (Ali-Yrkkö et al., 2005; Gugler & Konrad, 2002; Meador et al., 1996; Ragothaman, Naik, & Ramakrishnan, 2003).

Wei, et al. created a set of features such as number of patents awarded to a company, number and impact of recent patents, and the company's technological quantity in order to classify a company's status using the Nave Bayes model. With a total of 2394 acquisitions, their findings show precision rates ranging from 42.93% to 46.43%. (Wei et

al., 2008). By omitting all other categories, such as managerial and financial qualities, they were only able to achieve a limited level of success in their attempt to anticipate M&As by using technological features.

Over the past decade, research on start-up failures and bankruptcy has also received a lot of attention (Xiang et al., 2012). In his research to predict bankruptcy, Professor Edward Altman, well known for creating the (Altman) Z-score, suggested a number of financial ratios as the elements of a multivariate analysis. By employing a set of 21 railroads that went bankrupt between the years of 1939 and 1970, Altman expanded his initial research. Altman specifically examined ratios like liquidity measures, capital adequacy and financial strength, profitability, and performance metrics achieving a highly accurate (accuracy of 97.7%) classification at one and two years preceding to the company's bankruptcy (Altman, 1983; Zhang & Zhou, 2004).

According to the work done by Ravisankar et al. in which they used 35 financial variables to analyse 202 companies listed on different Chinese stock exchanges using different machine learning algorithms, including Multilayer Feed Forward Neural Neural Network (MLFF), Support Vector Machines (SVM), Genetic Programming (GP), Group Method of Data Handling (GMDH), and Probabilistic Neural Network (PNN). The dataset included 101 fraudulent and 101 legitimate businesses. The Probabilistic Neural Network attained True Positive Rate of 98.09%, which is the highest among other machine learning algorithm used in identifying which businesses were fake (Ravisankar et al., 2011). Despite their impressive numbers, they make the assumption that there are major differences between fraudulent and non-fraudulent companies that could be easily identified in their learning task due to the use of a small sample of only 202 companies. Their strategy yields the best outcomes when compared, although the focus of their research is more on fraud prevention than on start-up status prediction.

In studying the behaviour of venture capitals and investors on start-ups, Eugene and Daphne Yuan (2012) employed social media network features and a supervised learning method to predict investment behaviour using financial dataset from CrunchBase. In order to forecast whether an Investor will invest in a start-up based on their social status, the researchers used a standard link problem to simulate investment

behaviour. They attained a True Positive Rate of 87.5% and 0.77% for the Area Under Curve using Decision Tree classifier as their machine learning model. Their study nevertheless indicates successful organisations even though it does not directly predict acquisitions (Eugene & Daphne Yuan, 2012).

Xiang et al. (2012) predicted start-up success by integrating the structured data from CrunchBase database with the use of sentimental data on scraped news from TechCrunch, using the same dataset but focusing on start-up acquisition and investments from venture capital. They use the Bayesian Network as their machine learning model, and their model's True Positive Rate varied between 60% and 79.8% for various start-up categories. They outperformed that of Wei et al. (2008), the previous researcher on this same study, who had precision rates of 42.9% and 46.4%. Additionally, this study's final dataset, which included 59,631 data points and more than 6,000 acquired start-up data points present in the dataset, was significantly larger than the 2,394 observations used by Wei et al (2008). They also demonstrated how their sentimental data component enhances final outcomes.

Except for studies that use the CrunchBase database, the majority of studies use limited, narrow datasets that, despite producing encouraging results, prevent them from expanding on their work. Additionally, the majority of works have a tendency to concentrate on funding aspects, which don't fully convey a company's state or acquisition possibility. Studies utilising the CrunchBase database also fail to fully utilise the data at their disposal by choosing not to create a variety of variables linked to the impact of venture capital, including the number of investors, rounds of investment, amount raised, and many more. To be fair to them, it must be acknowledged that some of the research work that is available presently might not have existed while they were studying.

Table 1: Other study review on predicting start-ups success

Author	Source	Summary	Metric
Ross Greg et al, 2021	CapitalVX: A machine learning model for start-up selection and exit prediction	Developed a machine learning model called CapitalVX (for "Capital Venture eXchange") to predict the outcomes for start-ups	Achieved 90% for a three-way classification model (80% for a four-way model)
Dellermann, D. et al 2021	Finding the Unicorn: Predicting Early Stage Start-up Success through a Hybrid Intelligence Method	Developed a model to predict the success of start-ups by combining the collective intelligence of humans (through questionnaires) and machines in a Hybrid Intelligence method.	Not specified
Antretter Torben et al, 2019	Predicting new venture survival: A Twitter-based machine learning approach to measuring online legitimacy	Using natural language processing, and machine learning to capture online information from twitter to predict entrepreneurial outcomes, such as survival. **Method**: random forest and gradient boosting	The accuracy across all survival rates is about 74%
Zhang, Q., 2017	Predicting Start-up Crowdfunding Success through Longitudinal Social Engagement Analysis	Using social media activity e.g., Facebook to generate validity for a start-up **Method**: decision tree, SVM, KNN	Using TP and TN for all models
Ang, Y. et al 2022	Using Machine Learning to Demystify Start-ups' Funding, Post-Money Valuation, and Success	Predict post-money valuation of start-ups across various regions and sectors, as well as their probabilities of success **Method**: XGBoost	Achieved an accuracy of 95%
Yin, D. et al 2021	Solving the Data Sparsity Problem in Predicting the Success of the Start-ups with Machine Learning Methods	Trained a model that predicts start-up success using data from CrunchBase. **Method**: The results suggest that LightGBM and XGBoost perform best	Achieve 53.03% and 52.96% F1 scores

Chapter three

3.0 Methodology

The methodology applied is a mixed method of both qualitative and quantitative approach which depicts a rigorous way in solving the defined problem. The steps include ***Data collection*** of mixed data (quantitative data from CrunchBase and qualitative data from twitter). ***Data visualization*** using graphical plots to understand the data being used; ***Data Selection and Preprocessing***, by defining important features from the entire CrunchBase dataset, cleaning, transforming, creating new features in the dataset, categorization and encoding; ***Experiment Setup***, which addresses the dataset's primary issue, imbalanced classes. A variety of machine learning classifier algorithms is tested to categorise the observations into acquired, operational and closed; ***Experiment Results***, in which I will display my findings and results interpretation. The flow design for my methodology approach is shown in figure 1.

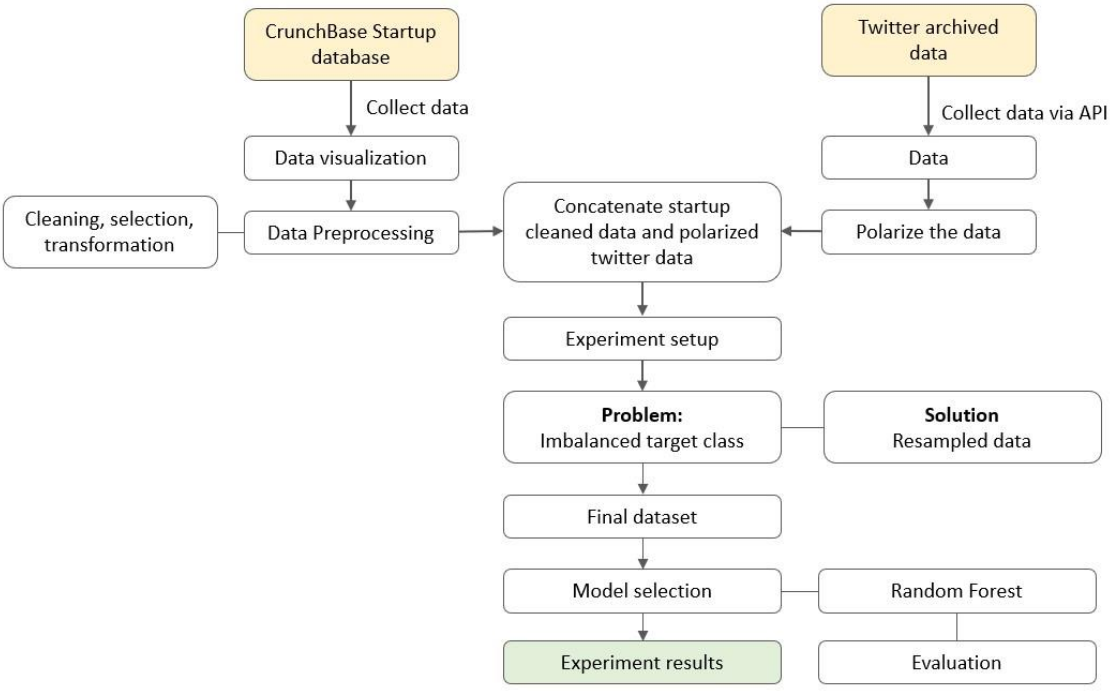


Figure 1: Flow design for the methodology

3.1 Data collection

The original source of the dataset used in this study comprises of start-up observations gathered from Crunchbase, a website that compiles information about start-up businesses. Crunchbase, which was created initially to monitor start-ups, now provides data about start-ups, venture capital firms, and businesses globally. Crunchbase obtains its data from a variety of sources, including from the start-up companies itself, venture programme, and internal data team. Also, tweets, which will be used for sentimental analysis, is collected via Application Programming Interface (API) using the date the start-up was found and the first funding date, along which the start-up's social media profile (with up to 13 features).

Also fetched 2,631,123 tweets related to the start-ups from twitter via Application Programming Interface (API) using the date the start-up was found and the first funding date, along which I also fetched the start-up's social media profile (having up to 13 features). From the raw data, I retained 46,560 observations from the funding data after filtering for empty fields, mismatching, or corrupted data, and finally down to 19,697 observations by focusing on data points related to my study by matching the start-ups with available tweets. This made it possible for me to keep a manageable quantity of data points and, as a result, to carry out analyses that are reliable. The final cleaned dataset of 19,697 observations (having a total of 34 features) which have available sentimental data is comparatively a portion of the total start-up population of 49,438, and I have considered this to be relatively representative because of the availability of the sentimental data for this portion. Programme codes showing how the tweets were collected from Tweeter can be found in Appendix 9.

The twitter dataset comprises of two parts: the start-ups profile (19,819 instances) and tweets from each start-up ranging from a minimum of 100 to maximum of 200 tweets (total of 2,633,258 instances).

For all start-up data found in the Twitter dataset, the funding data were first matched and merged with the Twitter profile dataset. A total of 19,687 profiles were merged

with the funding data, and a total of 2.6 million tweets were available within the start-up founding dates and first funding.

Using regular expression, each tweet was cleaned by removing hashtags, mentions and links. Polarization and subjectivity were also carried out immediately after running a single loop for cleaning each tweet, this way polarized and subjective data is extracted for all tweets belonging to a selected start-up and crunched into ten new features. Combining the funding data, twitter profile data, and crunched tweets resulted in data of 19,687 instances and 34 features (Appendices 6.2).

In order not to store excess tweets blob in the combined dataset csv file, the sentiment data was first crunched, then the numerical values (which has lower bytes) was stored for further usage.

3.2 Data understanding and Preprocessing

Data understanding in this study is done by visualizing features in the dataset using python libraries to plot graphs, count-plots, identify trends and plot other section maps using viable data grouping and slicing methods. Observations from the plots help identify in what area of the dataset needed cleaning. With preprocessing, functions were created to clean the data and prepare it for training models.

Despite its enormous size, the profiles in CrunchBase dataset have a lot of missing features and data values due to lack of validation means, as previously pointed out by Xiang (Xiang et al., 2012). Although the authors believe that the platform's validation were issues, as well as the fact that only well-known organisations and features were regularly assessed, the issue still exists and has gotten worse over the years. Due to CrunchBase free to edit feature, anyone can post and edit start-ups/business information without proper validation, and this in turn increased the number of sparse data.

Tackling the problem of sparsity head-on in this case may cause a loss of up to 45% of the data, most of which are valid datapoints as well. For this reason, my approach was to:

- check for features with highest number of missing data, then fill them with the most occurring data in that feature, except for dates.
- The dates, in this case the *founded_at* is filled with *first_funding_at* (dateTime). It is assumed that a company cannot be funded if it doesn't exist.
- The empty market type data is filled with others (objectType)

Feature Engineering

In this process new features were created from already existing features by extracting valuable information from them or by mathematical method.

The age of the start-ups, age of first and last funding, founded month, founded year, as well as the first and last funding year were created using existing date-time information in the dataset.

In order to perform a transformation for the funding-type I had to sum up the funding data into funding total for each start-up. Each of the of the funding type is transformed into a single feature, and any start-up without funding amount available during the transformation is labelled as "rounds".

Data selection

In data selection process, all unwanted variables where dropped and categorical features with exceeding numbers of 50 were grouped. The following steps were taken:

- First is dropping unwanted variables the contain string and objectType data.
- Dropping duplicates from the dataset.
- Ranking the country features into top 20. Any country that does not fall into the top 20 is grouped into "others". The top 20 countries selected was according to the global start-up ecosystem index, where the top countries with start-up was summed up into 20 countries (Berger, et al. 2016). The ranking rearrangement changes from year to year but remains within the top 20, where USA, UNITED KINGDOM, and CANADA remains at the top of the list. This has been ranked by money, good factors for management, and the availability of market for the

product or services. United States is confirmed to remain at the top all year round. The plotted sum of funding data (Appendix 4) using the state codes shows that at least the top 5 states which includes New Mexico (NM), Delaware (DE), Los Angeles (LA), Idaho (ID), and Vermont (VT) all belongs to United States.

- Grouping the markets type into top 20 as well and naming all other market type outside these top 20 as “others”. This is grouped according to the most occurring markets within the top 50 countries.
- Selecting all data that is above 1995. This slicing cuts down on the outliers outside the bracket of fast developing technology age.

Discretization

In this process I converted all funding types (excluding the rounds) to a maximum of 1, thereby converting the continuous variable to a discrete form. This will reduce the number of empty data with respect to funding. Before performing this process, I had to sum up the funding data into a funding total for each start-up.

Encoding

The first step in this process was encoding the target variable, status, by assigning integers to each unique string starting from 0 to 2. Next is encoding every other categorical variable in the dataset which includes market type, country code, and funding type.

3.3 Imbalanced data

Significant class disparity between the classes which represent the start-ups status presented another challenge when trying to develop an effective prediction model for the task at hand. When such imbalance difference in a dataset occurs and an algorithm is trained with it, the algorithm classifies more of the less represented class. In this case a trained model would have moderate to good accuracy, but accuracy is not a good metrics for evaluating a classification model at this point.

Handling of imbalanced data is done by using imbalance-learn library to perform oversampling. The library is an easy-to-use tool that requires passing the data to be oversampled to the selected sampling function. In this case I will be using Stratified Sampling, Synthetic Minority Over-Sampling Technique (SMOTE), and Adaptive Synthetic (ADASYN).

Stratified Sampling helps to remove sampling bias by splitting classes into strata based on shared features. A different probability sampling technique is used to randomly sample each subgroup once it has been split.

Synthetic Minority Over-Sampling Technique (SMOTE) is a strategy that involves oversampling the minority class. In this case rather of over-sampling with replacement, it will produce new synthetic datapoints of the class that is less represented in the dataset, which in this study is the "acquired" and "closed" classes.

According to Chawla who first introduced the strategy in machine learning, by taking each minority class sample and inserting synthetic samples along the line segments linking any/all the k-nearest neighbours, the minority class is oversampled. Randomly selected neighbours from the k-nearest neighbours are determined by the quantity of over-sampling necessary (Chawla et al., 2002).

Adaptive Synthetic (ADASYN) which, despite its similarities with SMOTE, create different numbers of samples based on estimates of the local distributions of the classes that would be oversampled. In other words, it generates synthetic data which are not copies of the minority classes, but instead a more difficult datapoints by learning what features made up the minority classes.

All results from the sampling methods used were combined and ranked by recall and accuracy (Appendix 3). Random forest classifier occurs at the top of the list with SMOTE as the imbalance oversampling method.

3.4 Model selection; Random Forest Classifier

In selecting a model to use, eight different classification models were first trained using the k-fold cross-validation method. In this study Random Forest Classifier was selected from these models as best fit for the dataset.

The broad category of ensemble-based learning techniques includes random forest classifiers. They have wide range of applications, are easy to deploy, low computational power, and have had great success in many classifications. A predetermined number of decision trees make up a random forest. A bootstrap selection from the training dataset is used to build each tree in the forest (Breiman, 2001). Assume that each feature is represented as M variables, a subset of F variables (where F < M) is randomly selected in each node as the random decision tree grows. To split the node, one of these F variables is chosen. Starting with an initial number of trees and iteratively increasing the number in a random forest, the list of important and unimportant features is constantly updated. A general pictographic representation of random forest is shown in figure 2. In general, it attempts to produce an uncorrelated forest of trees whose forecast by group is more accurate than that of any individual tree by using bagging and feature randomization while generating each individual tree (Oshiro, 2012).

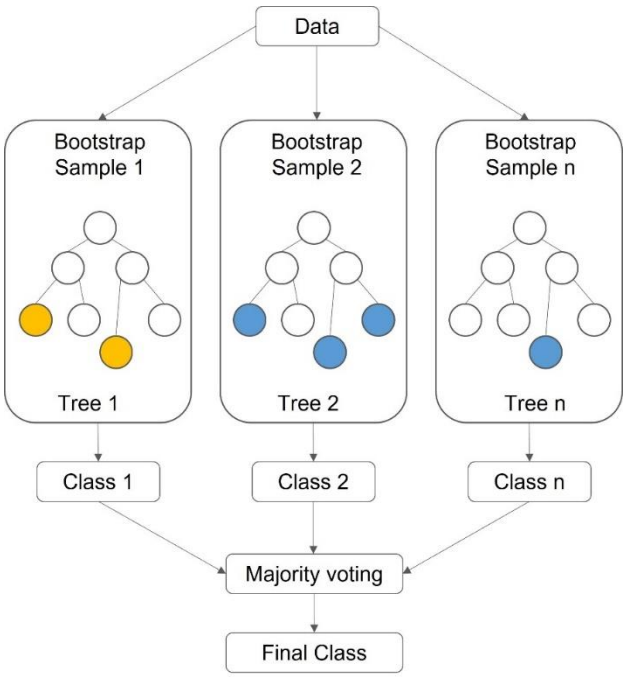


Figure 2: Random Forest Classifier bootstrapping method

3.5 Evaluation Metrics

Understanding the model that is to be evaluated is the first step in conducting a thorough evaluation of a machine learning model. In this study the classification model has multiple classes. A binary classification model (with classes of "positive" and "negative") may be used to understand the evaluation metrics for a multiclass classification model. The metrics evaluation is built from these four bases:

- ***True Positives (TP)**: Positive items that are correctly labelled positive.*
- ***False Positives (FP)**: Negative items that are wrongly labelled positive.*
- ***True Negatives (TN)**: Negative items that are correctly labelled negative.*
- ***False Negative (FN)**: Positive items that are wrongly labelled negative.*

In a multiclass classification problem, each class is in combination as a pair to form sets of binary classification problems. In this study, for instance, when considering the start-up class “operating”, a true positive occurs when an actual start-up is predicted to be a "operating". Any other prediction—whether it be “failure” or "acquired"—will be considered a false negative.

With the understanding of the metrics basis, I will briefly review the standard assessment criteria for machine learning classification models in this study and how they apply to the specific issue I am attempting to solve:

- **Accuracy**: items identified as true positive or true negative out of the total number of items — (TP+TN)/(TP+TN+FP+FN)
- **Recall**: items identified as true positive out of the actual positives — TP/(TP+FN)
- **Precision**: items identified as true positives out of the total identified positives — TP/(TP+FP).
- **F1-Score:** Average of the precision and recall taken into consideration, — (2 * precision * recall) / (precision + recall).
- **Specificity**: items identified as true negatives out of actual negatives — TN/(TN+FP)
- **False Positive Rate:** items wrongly identified as positive out of actual negatives — FP/(FP+TN).
- **False Negative Rate:** Items wrongly identified as negative out of the actual positives — FN/(FN+TP)

The proportion of correct predictions that are accurately identified in machine learning is determined by the true positive rate, otherwise known as sensitivity or recall (Wang et al., 2013). In this study, the true positive rate, also known as recall, is the proportion of all closed start-ups that were accurately categorised as such, whereas the false positive rate is the proportion of all acquired or operating start-ups that were declared successful.

To determine the performance of the algorithm I used confusion matrix from which values were extracted to calculate for True Positive Rate (TPR), False Positive Rate (FPR) and False Negative Rate (FNR). Representational image for multiclass confusion matrix as defined by Krüger 2016, is shown in figure 3.

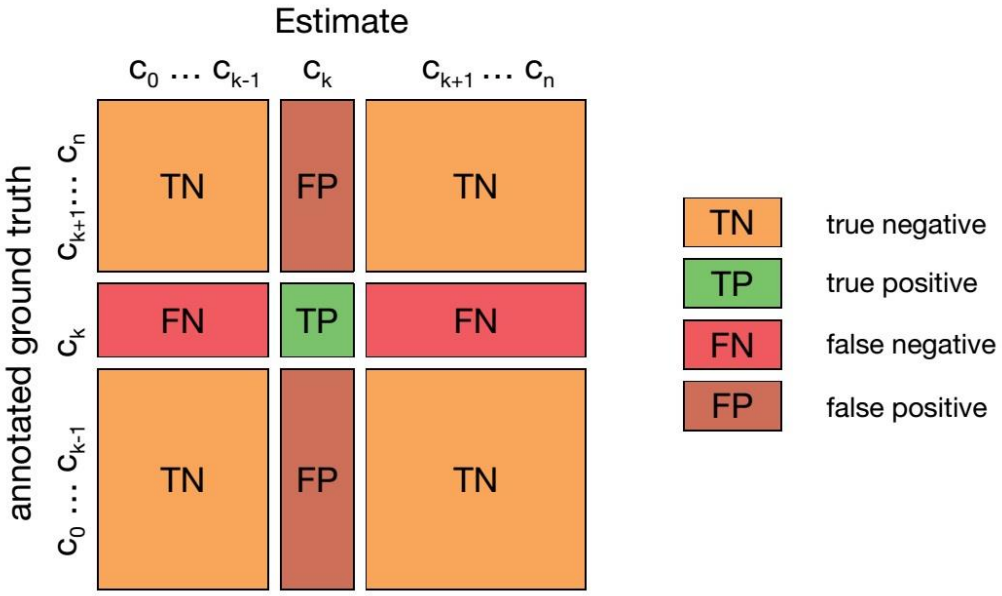


Figure 3: Confusion matrix for multiclass classification (Krüger 2016)

3.6 GUI Setup and Design Approach

The Graphic User Interface (GUI) which is used to present the result is programmed using python as backend to create an API, and ReactJS (a JavaScript language) at the frontend to build a Progressive Web App (PWA) which is installable in any device that can access a webpage. Figure 4 depicts a proper representation of the GUI design flow.

The backend API is built in python language using FLASK as its major library, and CORS library (Cross-Origin Resource Sharing). This helps the flow of data in JSON format (JavaScript Object Notation) between the backend and the frontend (Appendix 11).

React Apps are built in components. The design of the user interface (UI) used in this study is built in a way that each component performs a specific task in the sections of the user interface (UI). Few important sections of the program codes used for building the App and each component can be found in Appendices 12, and 13. The sample JSON data passed from the backend to the frontend can be found in Appendix 14.

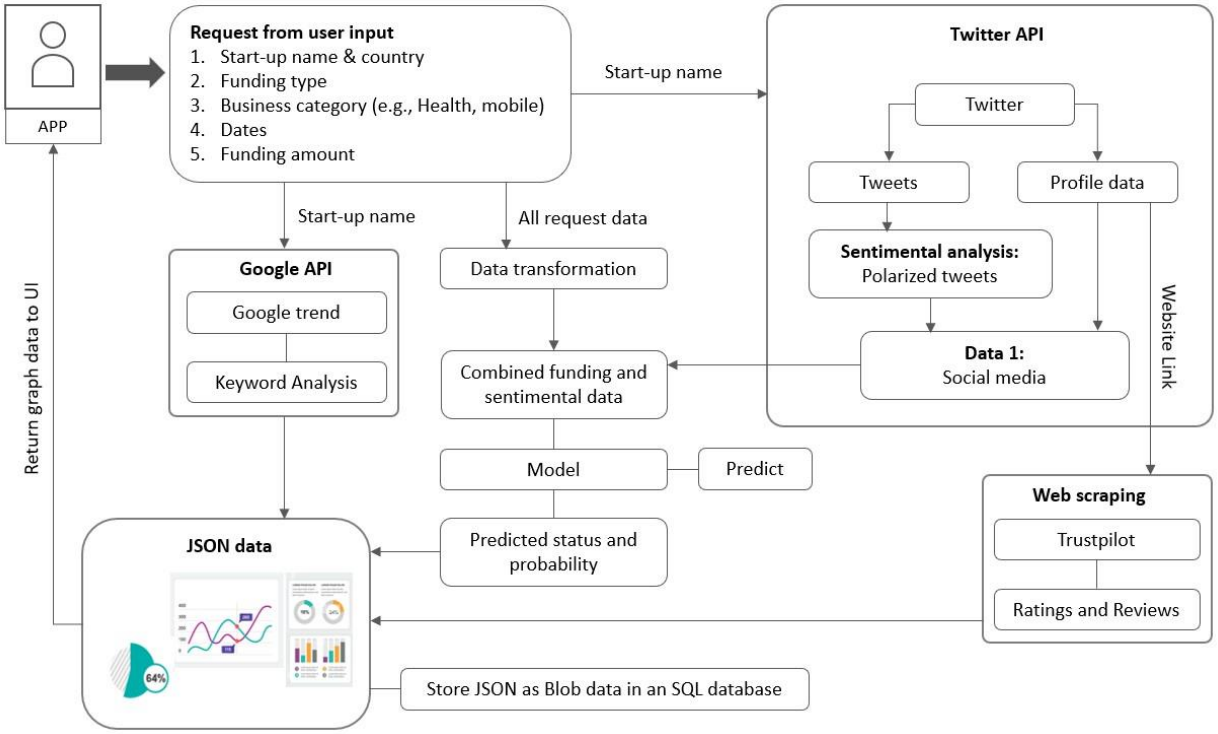


Figure 4: GUI design flow

Chapter four

4.0 Results

4.1 Data collection

The data which was collected from CrunchBase includes unprocessed information on start-ups and their funding rounds, dating back to 1915. The resultant dataset reflects a twenty-year period from 1995 to 2014, with a total of 49,438 observations of global start-ups funding instances with 39 features in total. Information on the categorical and numerical data is shown in table 1 and table 2. Information on the start-up twitter profile and tweet data is shown in table 3 and table 4.

Table 2: Summary of categorical data in the dataset

Features	count	Unique	Top occurring	Freq
permalink	49438	49436	/organization/treasure-valley-urology-services	2
name	49437	49350	Roost	4
homepage_url	45989	45850	http://spaceport.io	2
category_list	45477	16675	|Software|	3650
market	45470	753	Software	4620
status	48124	3	operating	41829
country_code	44165	115	USA	28793
state_code	30161	61	CA	9917
region	44165	1089	SF Bay Area	6804
city	43322	4188	San Francisco	2615
founded_month	38482	420	2012-01	2327
founded_quarter	38482	218	2012-Q1	2904

Table 3: Summary of Numeric data in the dataset

Features	Missing values	Unique values
funding_total_usd	0	15008
funding_rounds	0	17
founded_at	10885	3368
founded_year	10956	103
first_funding_at	10	3904
last_funding_at	6	3651
seed	0	3337
venture	0	9300
equity_crowdfunding	0	252
undisclosed	0	687
convertible_note	0	299
debt_financing	0	1872
angel	0	999
grant	0	532
private_equity	0	847
post_ipo_equity	0	239
post_ipo_debt	0	57
secondary_market	0	20
product_crowdfunding	0	176
round_A	0	2035
round_B	0	1269
round_C	0	740
round_D	0	458
round_E	0	225
round_F	0	110
round_G	0	32
round_H	0	5

Table 4: Start-ups Twitter profile data

Features	Missing values	Unique values
user	0	19760
date	0	19757
displayname	4	19698
description	3158	16601
followersCount	0	6186
friendsCount	0	3681
statusesCount	0	6915
listedCount	0	1308
favouritesCount	0	4410
linkUrl	4057	15680
profileImageUrl	3	18354
profileBannerUrl	7367	12408
verified	0	2
funding_name	0	19819
cleaned_name	1	19772

Table 5: Start-ups tweet data

Features	Missing values	Unique values
user	0	20307
date	0	2575131
tweet	0	2593201
retweets	0	1015
likes	0	2257
reply	0	412
quote	0	337
funding_name	0	19819
cleaned_name	200	19772

4.2 Data Visualization

Data visualisation is a crucial component of data analysis because it offers insights and reveals complex data structures that cannot be understood in any other way. In other words, data visualization helps us to best understand the data (Aisch, G., 2016).

A count-plot of the number of start-ups over the years (fig. 5) shows the increasing number of start-ups over time, with maximum spike in year 2012, and then decreased in year 2013 and 2014. Figure 6 shows that the funding of start-ups increase over the years which moves in direct response to the number of start-ups by year. This indicates that with increasing start-ups over time, there is also an increasing number of investors who are willing to take the risk on investment by investing in upcoming businesses which they have no proper way of verifying their potential returns.

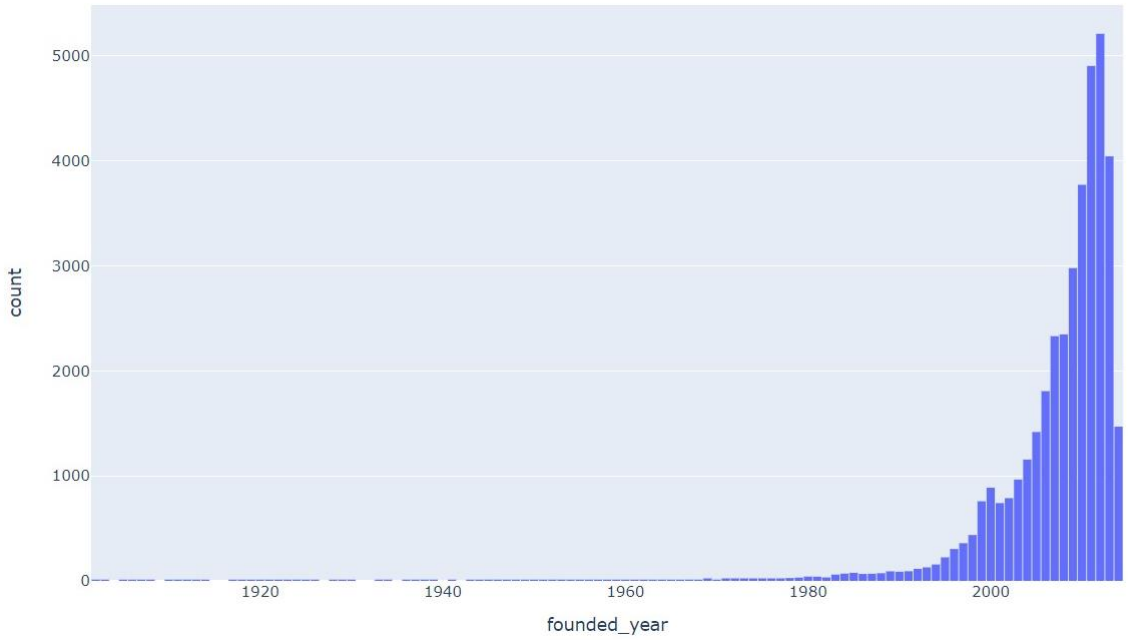


Figure 5: Start-up count plot over the years

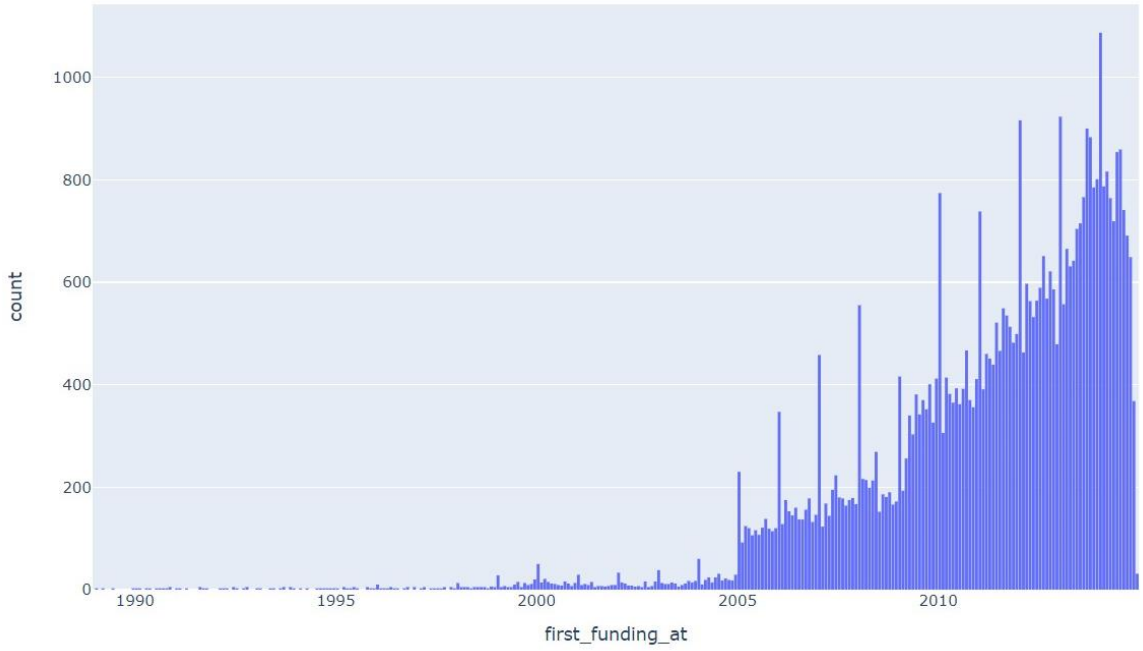


Figure 6: Start-up funding over the years

Figure 7 shows a count-plot for the top 50 unique country code (countries that occurred more frequently within the dataset) and the top 20 countries among these 50 countries with the highest success rate (fig. 8).

There are about 754 unique values for the different type of markets (categories) for the start-up data. All these market-types may not be representation of the category they belong to; some of the market-types are also not in existence. For this reason, I grouped and sorted the data according to the market-type occurrence. From the grouped data I selected the top 20 markets which have more occurrence across the dataset and also exist within the top 20 countries selected (fig. 9).

The status of start-ups across the selected markets where plotted (fig. 10), as well as the growth of the market with increase in number of years (fig. 11). With this plot (fig. 10) it is observed that all market-type has high success rate with SOFTWARE, BIOTECHNOLOGY, MOBILE, E-COMMERCE and HEALTH CARE taking the lead. CURATED WEB, SOFTWARE and MOBILE has the highest number of closed start-ups. This is further verified by the bar plot showing high rise of SOFTWARE, BIOTECHNOLOGY, and MOBILE over the years (fig. 11), and the high funding data for these top markets (fig. 12). Since the United States has been confirmed to have the highest number of start-ups, the count plot of market-types where plotted where the data exist within the United States (fig. 13), it was verified to have SOFTWARE, BIOTECHNOLOGY, and MOBILE still at the top of the list.

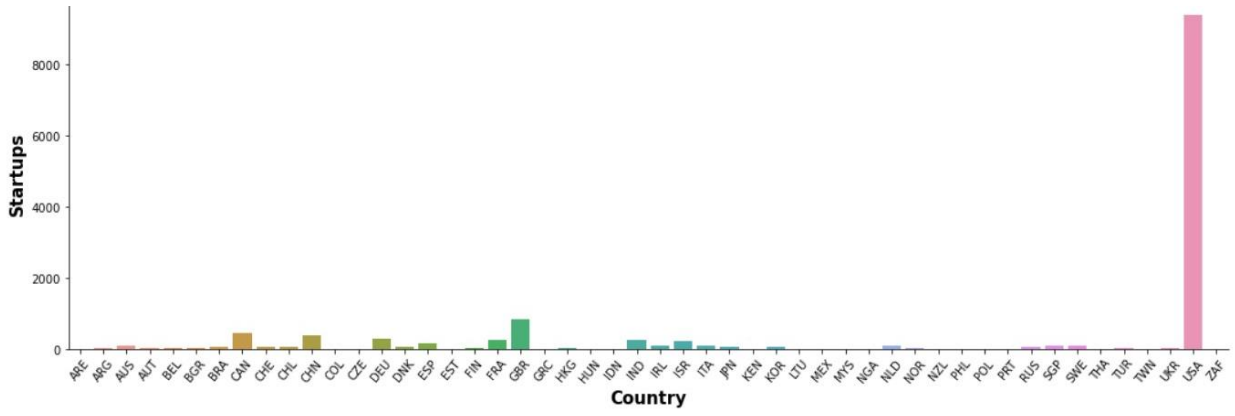


Figure 7: Start-up count-plot among 50 countries

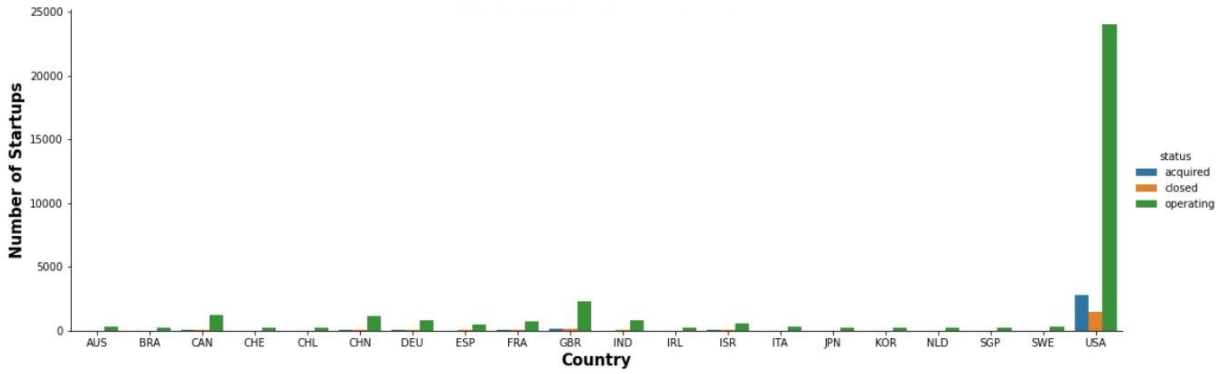


Figure 8: Start-up status count-plot in 20 countries

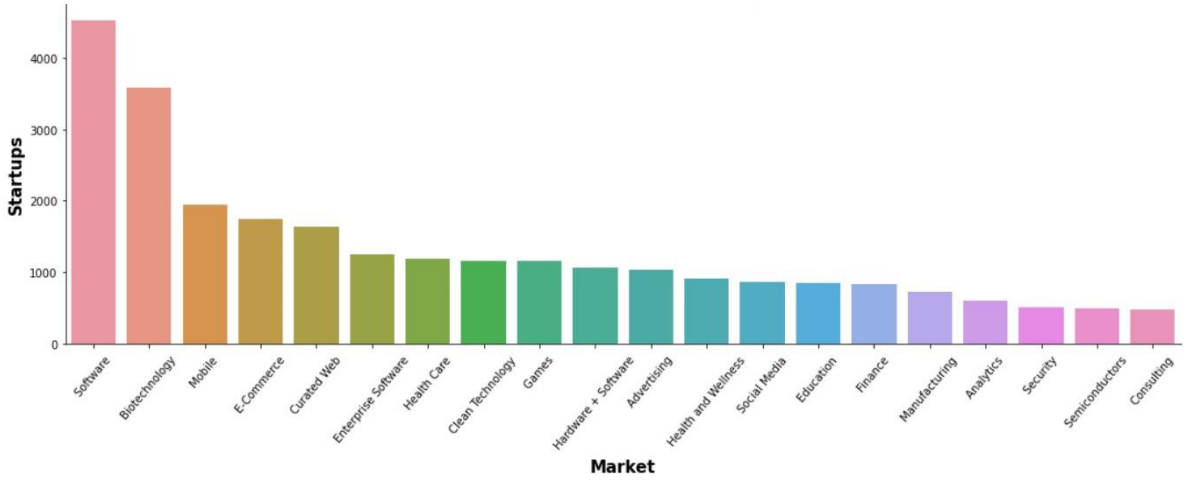


Figure 9: Start-up count-plot according to top 20 market-type

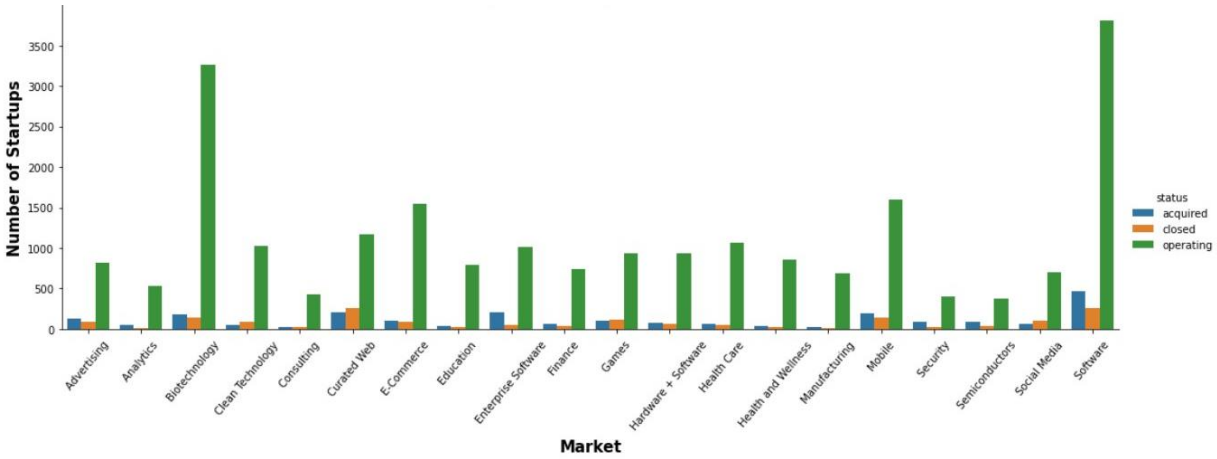


Figure 10: Start-up status according to top 20 markets

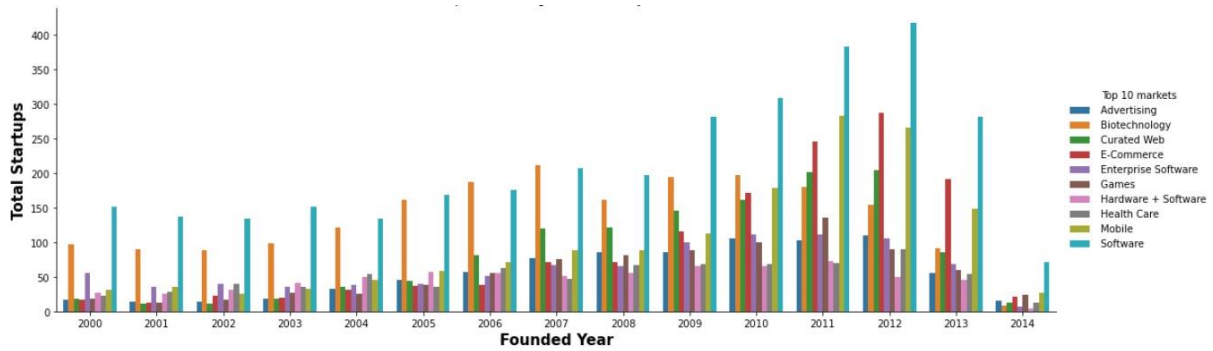


Figure 11: Top 10 start-up market-type growth over the years

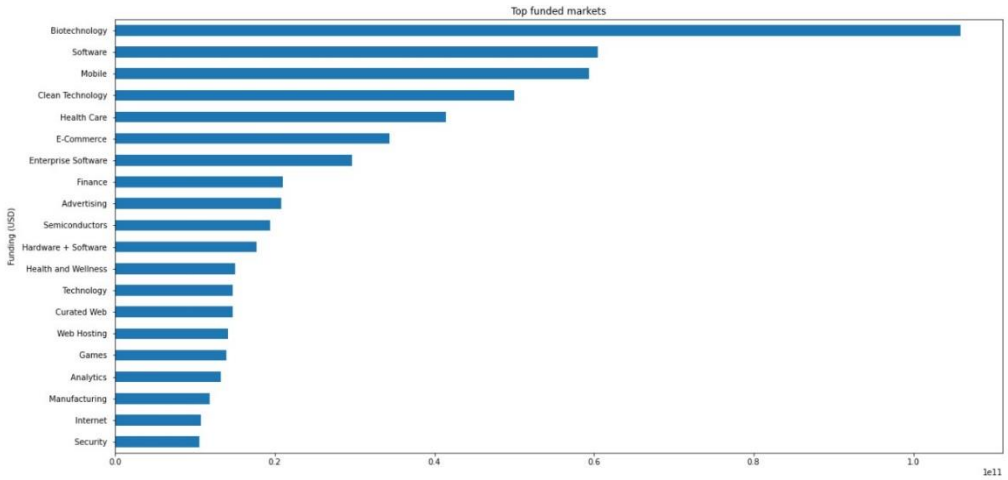
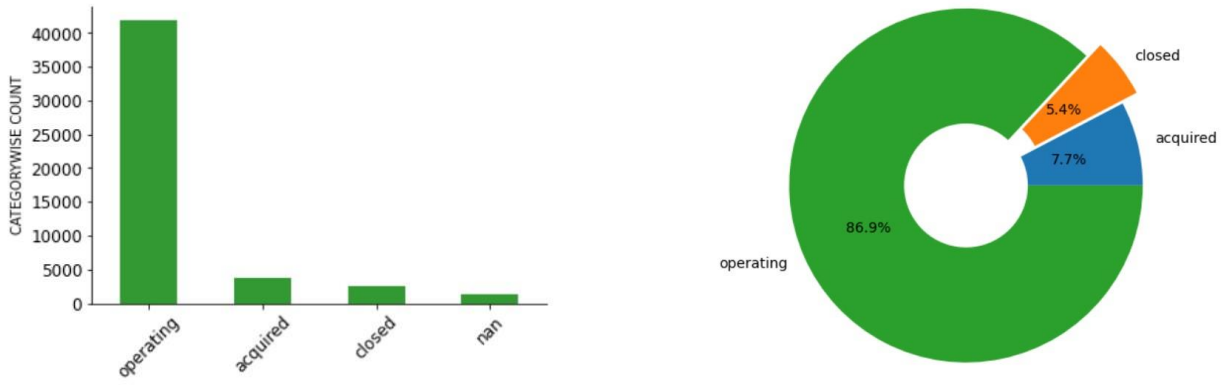


Figure 12: Start-up top funded market-types

Figure 13: Top 20 start-up market-type in the United States of America

Among the categorical variables is the status of the start-up which holds the classes to which all start-ups are classified accordingly. A bar plot of the status shows the variable to have four classes (fig. 14a) which includes operating (companies which are in continuation after funding), acquired (companies that are bought out or merged), closed (companies that are closed), and nan (meaning *Not a Number*). Selecting the first three valid classes and plotting them on a pie chart (fig. 14b) shows the imbalance state of the classes in the dataset.



(A) *(B)*

Figure 14: Start-up status (a) bar-plot of classes (b) pie-plot of valid classes

A box plot of the start-ups funding features in the dataset was plotted to identify features with outliers. Figure 15 shows the plotted box chart with all funding features showing widespread of values which contributes as outliers in the dataset.

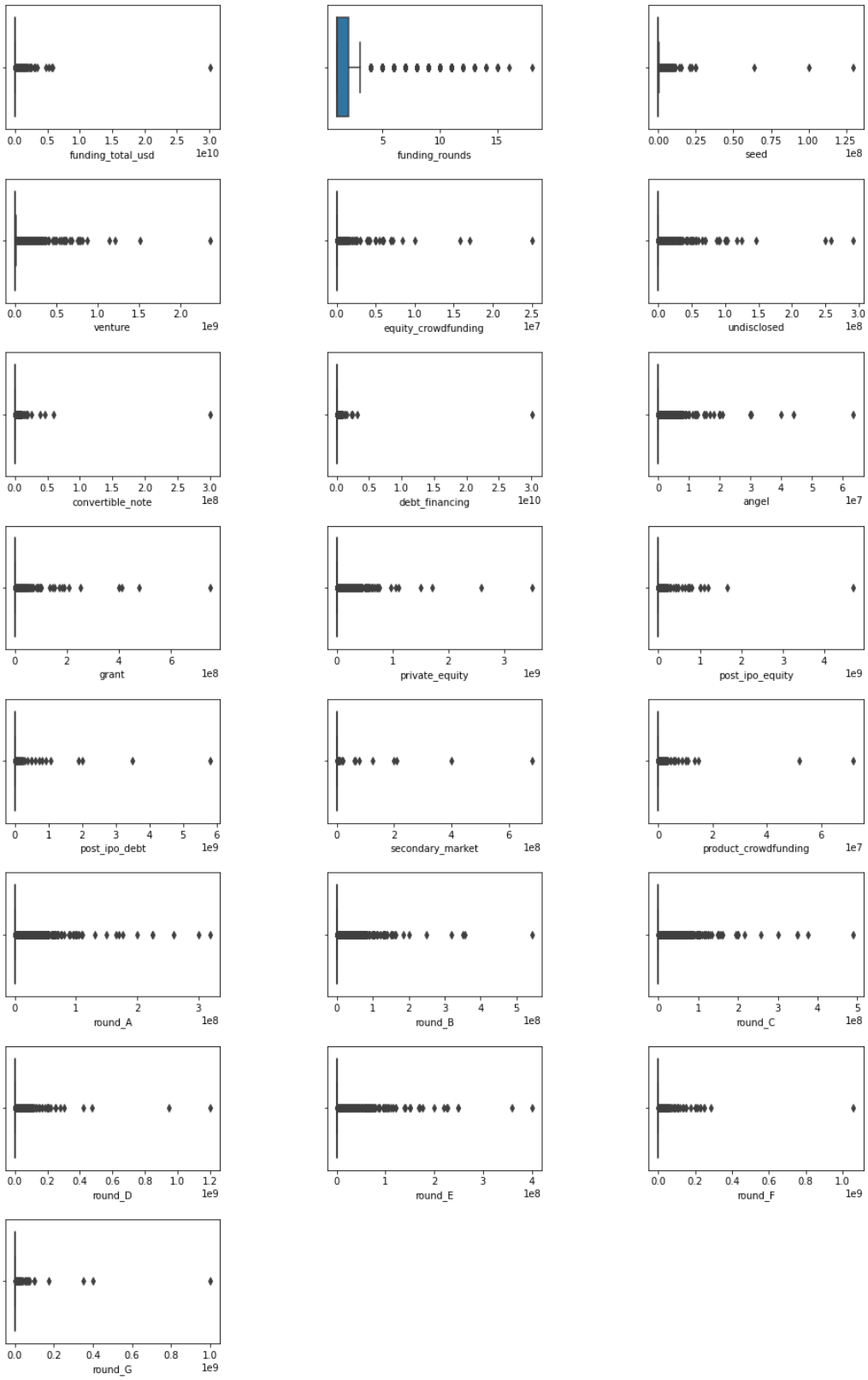


Figure 15: Box plot for start-ups funding features

4.3 Preprocessing and Experiment Setup

In preprocessing, challenges from the data sparsity and imbalance classes were solved. The results generated during this process are presented in tables. The proportion of target classes from start-up combined funding and Twitter data (combined data is in Appendix 2), which shows imbalance data is represented in table 5.

Table 6: Proportion of target classes from combined data.

Classes	Class Encode	Count	Percentage
Operating	2	17,348	88.12%
Acquire	1	1,474	7.49%
Closed	0	865	4.39%
	TOTAL	19,687	100%

The resulting data which proceeds solving the imbalance data by using stratified and oversampling methods (which includes Synthetic Minority Over-sampling Technique and Adaptive Synthetic) to train eight selected classification models is tabled in tables 6, 7, and 8. These results include the calculated recall, precision and F1-score for each model.

Table 7: Results of trained eight models after applying Stratified Sampling.

Model	Cross Val Mean	Cross Val Error	Recall	Precision	F1-Score
RandomForest	0.715340	0.019790	0.699790	0.674905	0.679113
GradientBoosting	0.716928	0.015241	0.691903	0.669474	0.675340
SVC	0.631058	0.000652	0.629338	0.397686	0.487386
LogisticRegression	0.631057	0.001962	0.628812	0.475496	0.490106
AdaBoost	0.622269	0.020238	0.605678	0.611668	0.608526
DecisionTree	0.624071	0.015773	0.598318	0.602243	0.600230
KNeighboors	0.552629	0.020732	0.534700	0.507760	0.518518
MultipleLayerPerceptron	0.486493	0.136294	0.511567	0.491406	0.498047

Table 8: Results of trained eight models after applying SMOTE.

Model	Cross Val Mean	Cross Val Error	Recall	Precision	F1-Score
RandomForest	0.798452	0.012036	0.750000	0.752054	0.750444
GradientBoosting	0.748214	0.014098	0.724167	0.724186	0.724034
AdaBoost	0.701071	0.011523	0.641111	0.641777	0.641383
DecisionTree	0.702500	0.018879	0.636389	0.637088	0.636664
KNeighboors	0.591786	0.014090	0.529167	0.527461	0.514823
LogisticRegression	0.457976	0.012384	0.450278	0.461733	0.449568
SVC	0.449762	0.009595	0.430000	0.410515	0.347622
MultipleLayerPerceptron	0.436667	0.041859	0.411111	0.408008	0.407680

Table 9: Results of trained eight models after applying ADASYN.

Model	Cross Val Mean	Cross Val Error	Recall	Precision	F1-Score
RandomForest	0.793249	0.009441	0.744972	0.746715	0.744064
GradientBoosting	0.743445	0.009136	0.723743	0.722278	0.721845
DecisionTree	0.688374	0.008765	0.627654	0.627031	0.627184
AdaBoost	0.690892	0.007912	0.625698	0.624367	0.624896
KNeighboors	0.561716	0.016195	0.497765	0.493856	0.481989
LogisticRegression	0.436493	0.011026	0.423743	0.432891	0.398290
SVC	0.438647	0.009954	0.422626	0.451115	0.336625
MultipleLayerPerceptron	0.413517	0.046367	0.389665	0.391537	0.382487

4.4 Experiment Result and Evaluation

Before training the selected model among the eight trained, feature importance of the of the variables in the training data is examined (fig. 16). The feature correlation was also carried out and the heatmap plotted (fig. 17). A selected Random Forest Classifier was trained and the confusion matrix result before optimization is displayed in figure 18, and the heatmap for the result in figure 19. Table 9 explains how the multiclass confusion matrix is classified. The model was run several times (43 times) to select a suitable estimator. Top 9 of the estimator results were plotted in table 10.

In getting the model best predicted results, and optimal threshold was determined using a precision-recall curve (fig. 20). Table 11 contains result of running the process several times to derive several calculated thresholds and F1-Scores. Using the threshold and model hyperparameters, the model prediction is optimized, and the confusion matrix result is printed (fig. 21). A single holdout dataset was predicted, and the result represented on a pie chart (fig. 22). All holdout dataset was then predicted, and the result compared with the start-up valid status and a confusion matrix for the holdout prediction is printed (fig. 23).

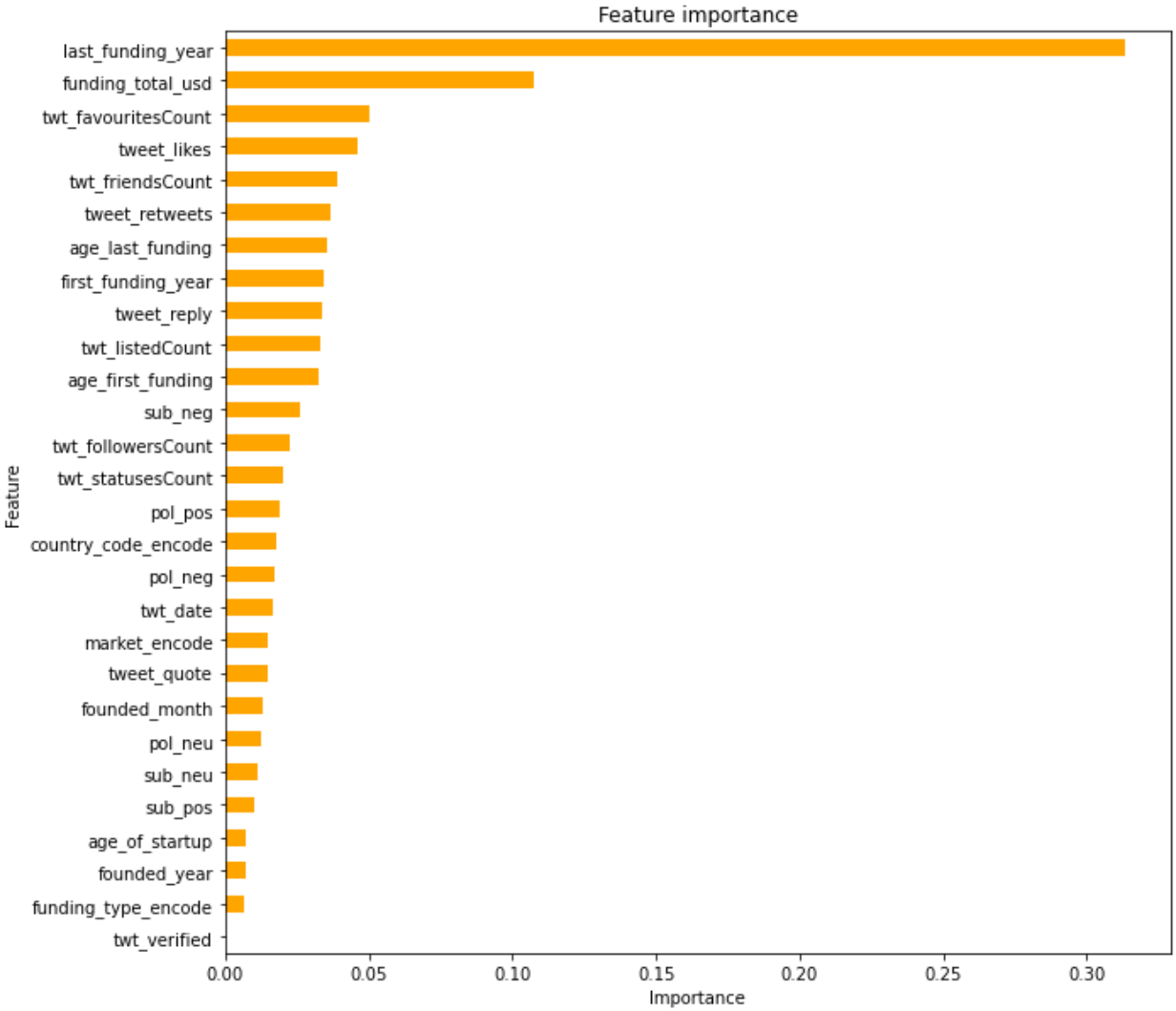


Figure 16: Feature importance of variables in the dataset

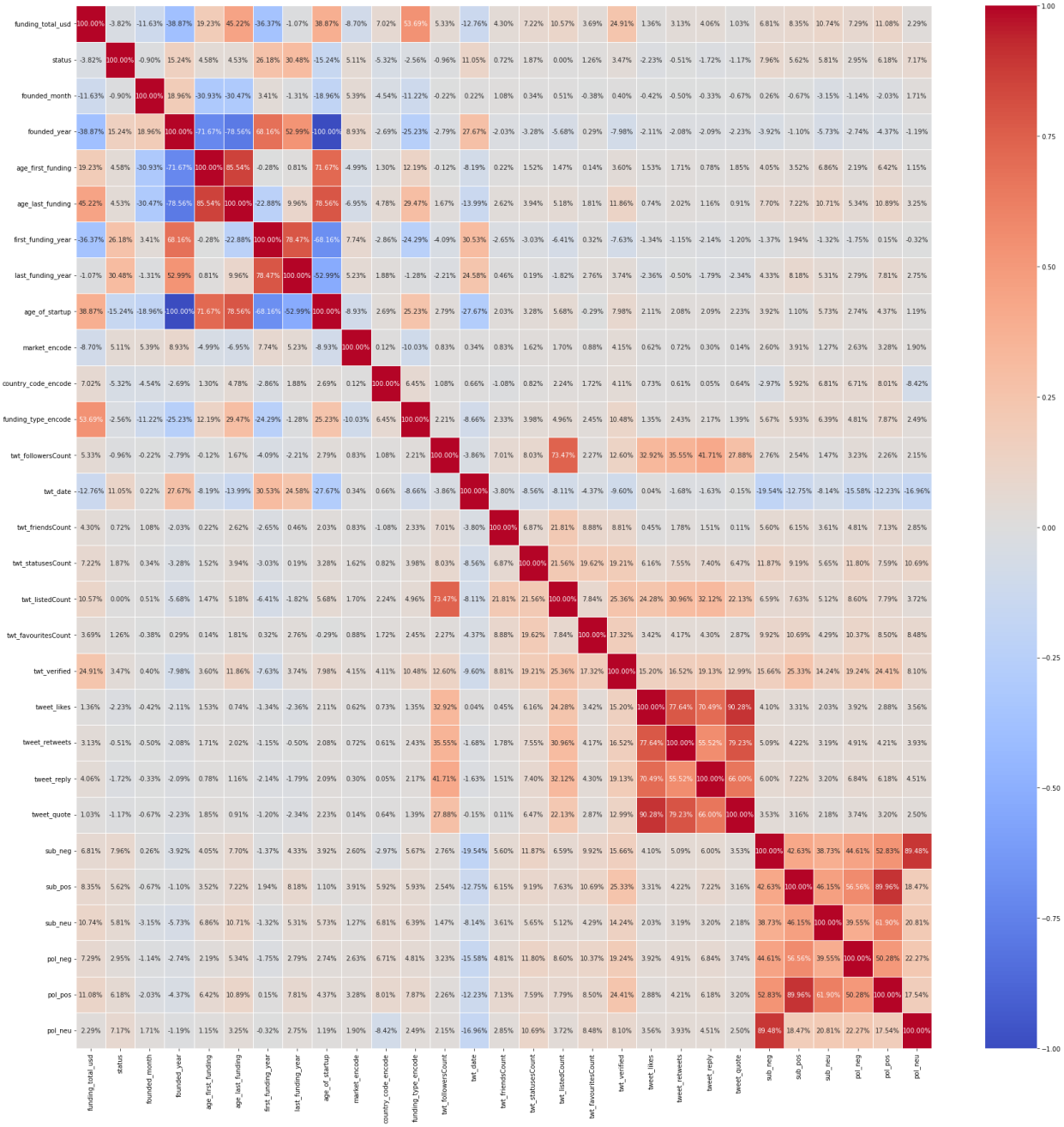


Figure 17: Correlation heatmap of the features

	precision	recall	f1-score	support
0	0.78	0.82	0.80	5088
1	0.76	0.75	0.75	5222
2	0.80	0.77	0.78	5304
accuracy			0.78	15614
macro avg	0.78	0.78	0.78	15614
weighted avg	0.78	0.78	0.78	15614

Figure 18: Confusion matrix before optimization

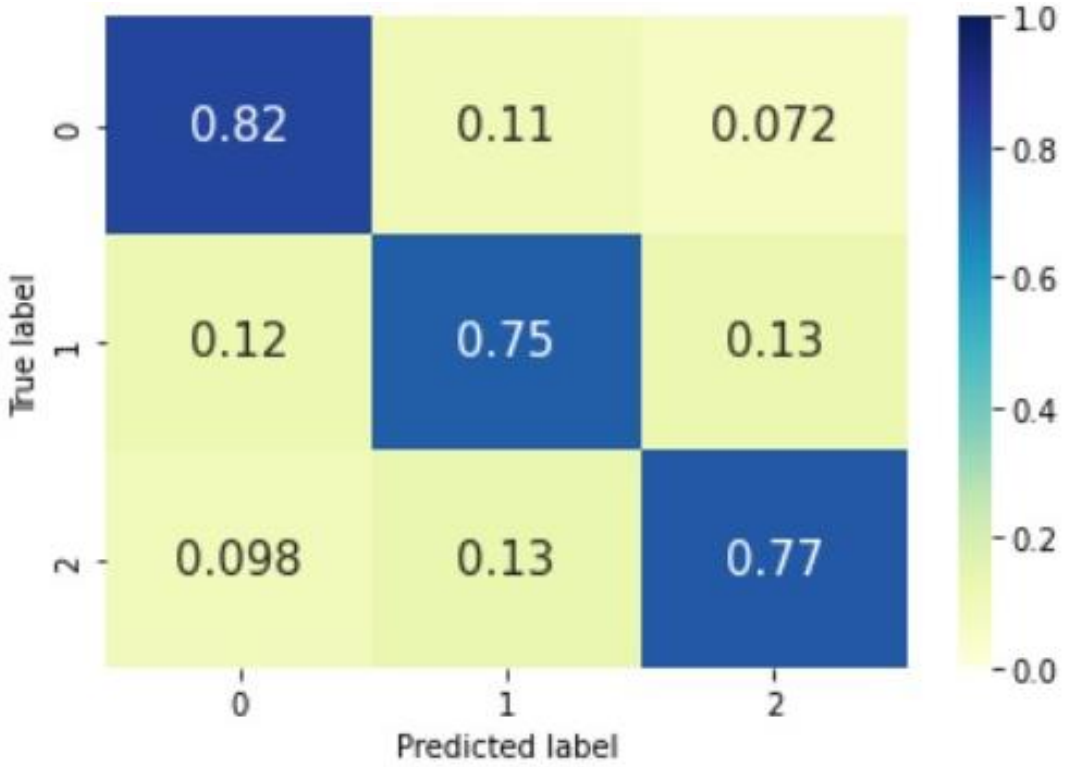


Figure 19: Confusion matrix heatmap

Table 10: Multiclass confusion matrix

Represents the target Vs the rest

		Predicted class		
	CLASSES	closed	acquire	operating
Observed class	closed	TP_C	FN	FN
	acquire	FP	TN_A	TN
	operating	FP	TN	TN_O

Table 11: Selecting nth-estimator value for Random Forest Classifier

n-estimator	Train result	Test result	Error difference
39	0.999698	0.926925	0.072773
40	0.999780	0.926092	0.073688
42	0.999753	0.926028	0.073725
27	0.999753	0.925708	0.074045
37	0.999753	0.925387	0.074365
33	0.999726	0.925195	0.074530
28	0.999726	0.925131	0.074594
41	0.999835	0.925067	0.074768
43	0.999726	0.924939	0.074786

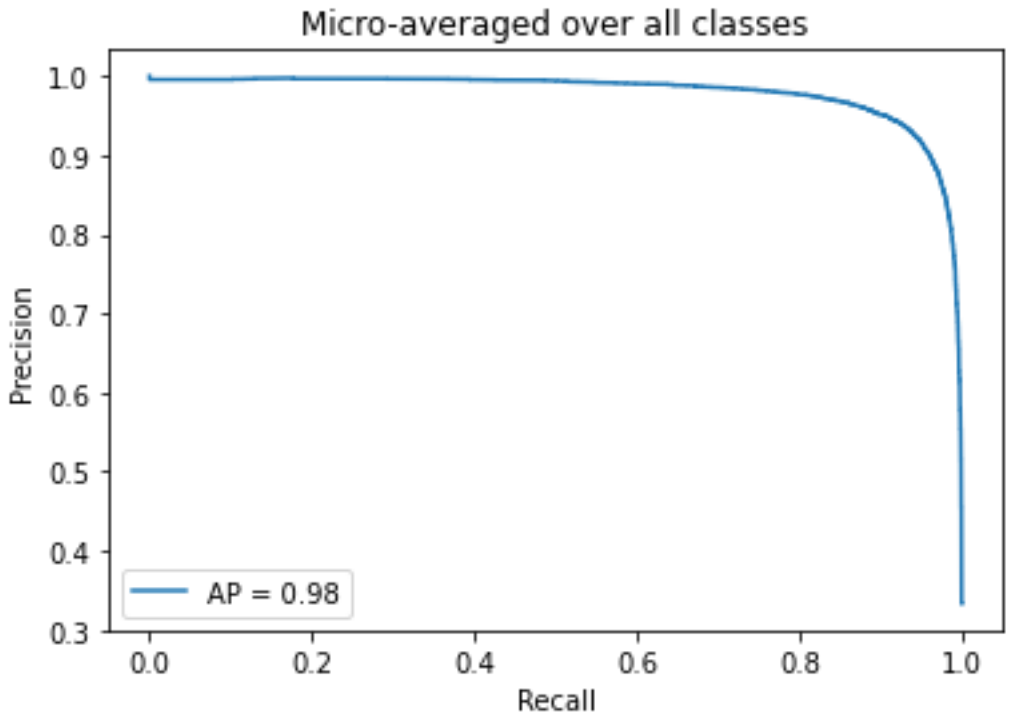


Figure 20: Precision-recall curve for finding Optimal threshold

Table 12: Selecting Optimal Threshold against best F1-Score

S/N	Threshold	F1-Score
1	**0.473**	**0.934**
2	**0.446**	**0.934**
3	**0.468**	**0.934**
4	**0.448**	**0.933**
5	**0.466**	**0.935**

	precision	recall	f1-score	support
0	0.93	0.96	0.95	5247
1	0.92	0.92	0.92	5099
2	0.94	0.91	0.92	5211
accuracy			0.93	15557
macro avg	0.93	0.93	0.93	15557
weighted avg	0.93	0.93	0.93	15557

Figure 21: Confusion matrix after optimization

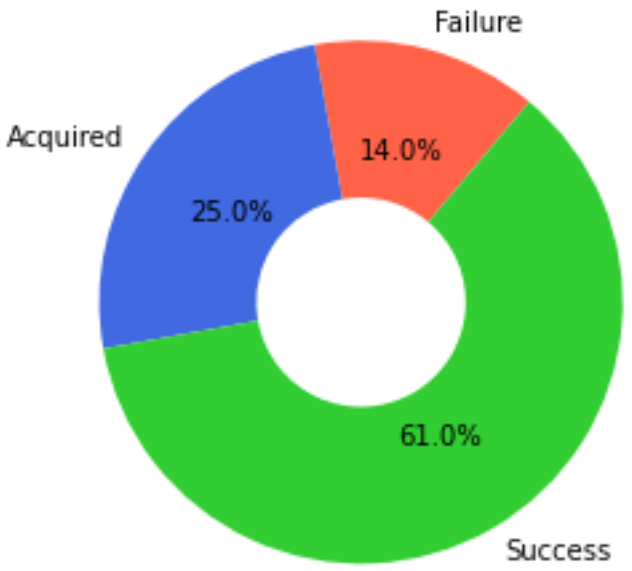


Figure 22: Plotting predicted probability values for a single datapoint

	precision	recall	f1-score	support
0	0.84	0.89	0.86	85
1	0.84	0.86	0.85	161
2	0.98	0.98	0.98	1723
accuracy			0.96	1969
macro avg	0.89	0.91	0.90	1969
weighted avg	0.96	0.96	0.96	1969

Figure 23: Confusion matrix on holdout validation data

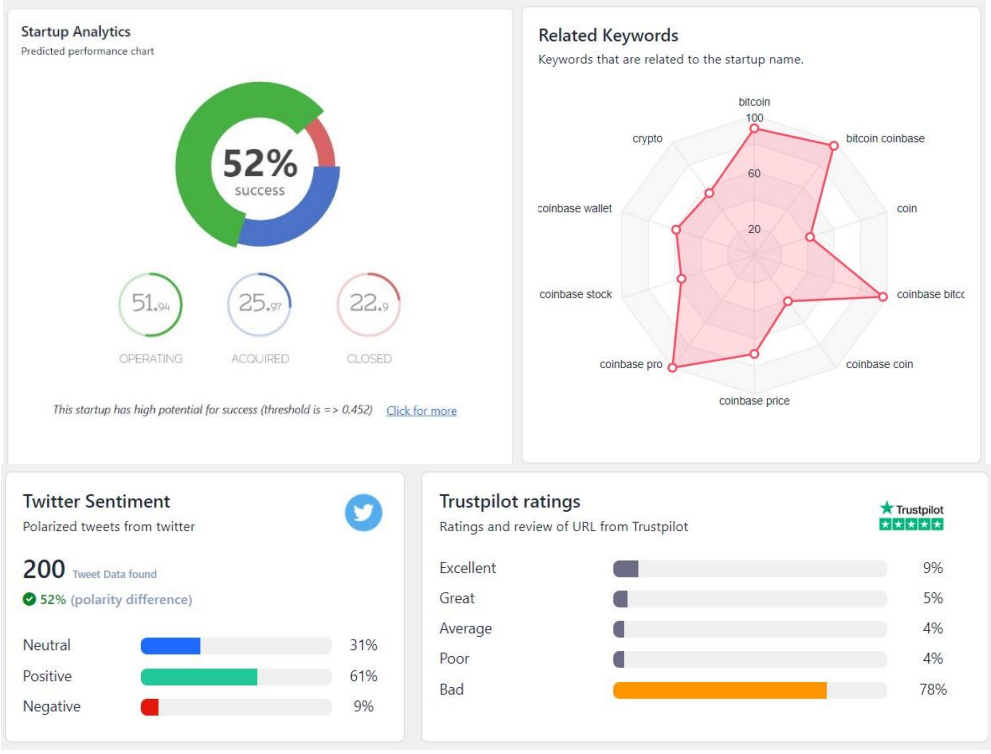


Figure 24: GUI App interface showing the first prediction

Chapter five

5.0 Discussion

From the proportion of target classes in the final data (after preprocessing, table 5), it is observed that the "**operating**" class covers 88.12% proportion of the data, and "**acquired**" having 7.49% proportion is the second largest, while "**closed"** data having 4.39% is the smallest class in the dataset. Imbalance in dataset is said to occur if the classes are not evenly or well distributed in the dataset (Chawla et al. 2002). Stratified Sampling, SMOTE, and ADASYN were used to resample the training dataset which is then used across the six selected models using k-fold cross validation. The cross validation means result, error, recall, precision and calculated F1-score for the three methods across the eight models were compared (tables 6, 7, and 8). From the list Random Forest Classifier has higher validation mean and F1-score. This is not to be unexpected as other researchers like Antretter Torben et. al. has selected Random Forest as best fit model (Antretter Torben et. al., 2019).

To determine the model performance confusion matrix was used (fig. 18 and 19), and the approach to decipher the confusion matrix for this study is represented in table 9. True positive rate, also known as recall, is the proportion of all closed start-ups that were accurately categorised as such, whereas the false positive rate is the proportion of all acquired or operating start-ups that were declared successful. False negative, which is not neglected, is the number of successful start-ups that are falsely classified as closed; the author aims to reduce this value to the barest minimum, but also maintaining high True Positive Rate (TPR). This has done justice to the limitation on the concise work of Zhang (Zhang, Q., 2017), where True positives were the evaluation aim using Facebook activities on trained models, and Ross Greg et. al. who developed CapitalXV using CrunchBase data focusing more on accuracy (Ross Greg et al, 2021). The use of sentiment data extracted from Twitter tweets has helped in improving model result positively, as against the use of funding data only (as seen in the works of Dellermann, D. et al 2021; Ang, Y. et al 2022; Yin, D. et al 2021).

Chapter six

6.0 Conclusion, Limitations and Recommendations

6.1 Conclusion

The major goal of the current study was to create a classification system using machine learning approach to predict the possibility of a start-up status to be success (which includes operating and acquired). It is considered that the goal was accomplished by creating a multi-class classifier to closely predict a start-up as successful (operating or acquired) or not successful (closed) having a True Positive Rate (TPR) of 95.5% and a False Positive Rate of 3.83%, the target of achieving low false negative was also achieved where the false negative is 7.6%.

The model has high accuracy of up to 92% in identify whether a start-up is actually classified as successful or achieved success through acquisition (precision) in addition to classifying the total number of successful start-ups in the dataset (TPR, recall). Among the eight classification models that were trained, Random Forest Classifier machine learning algorithm was used because it offers a quick, simple, and effective model with good outcomes.

In addition to creating a prediction model, this study has made a significant contribution to the direction of the research by offering a comprehensive analysis of the collected datasets and output outcomes. Predictions depend heavily on a start-up's online presence and validity data, such as that obtained from social media sites like Twitter. However, a significant finding of our work is the usefulness of taking a start-up's online presence in the form of quality social media profile into account; this has proven to increase predictions positively.

6.2 Limitation

Accessing high volume of start-up funding data comes at a high price. From CrunchBase database, these data contain more volumes of start-up data with multi-class targets that could substitute the synthetic data used in resampling.

This study is limited to the data collected from CrunchBase and Twitter only, whereas more data could be accumulated from other sources like Facebook, Google search, reddit and even TechCrunch. This limitation is tied to the second limitation, time constraint. It took the researcher four days to completely fetch 2,631,123 tweets from over 40,000 profiles on Twitter. It took another two days to completely crunch the data, combine with funding data and train eight models in 3 different ways (total of 24 trainings), with the PC clocking at full computation speed for those days without being turned off. With all this listed anyone would understand that there is no option for mistake at any stage or else the cycle will be repeated for the next four days, thereby loosing precious four days from the allocated 3 months for this research. Researchers with wider range of time may want to consider enriching their data from these sources as mentioned earlier.

6.3 Recommendation

Even though this study has addressed a number of the shortcomings of earlier studies on start-up success prediction, it has identified a number of areas for improvement. First, additional study should utilise the content of web pages that are found mentioning the keyword trend connecting to the start-up name in addition to tracking merely the source of start-up mentions online.

Secondly, just as data from Twitter has proven to increase prediction score, it is highly recommended that further research should focus on gathering social and web data rigorously via several means will enrich the training data and add meaning to the model learning path.

6.4 Future work and Possible Updates

If given more time the author will likely take on the recommendations as stated above and try to tackle the limitations where possible. Future upgrades will include gathering web interlinked data using Google Analytics API. Also, more social media data to firmly verify start-up online presence and legitimacy will be increased using Facebook API. SEMrush is notable a high standard recommendable SEO analyser, and the data from using its tools can also be gathered via its API in which I will aim to collect granular data related to keyword search queries and similarities that could increase possible ranking for the start-ups. Finally, will rearrange the design flow by building a pipeline which will execute sequential steps that do everything from data extraction and preprocessing to model training and deployment when batches of start-ups status are detected from CrunchBase and TechCrunch newsfeed.

References

Aisch, G., 2016, June. Data visualization and the news. In *Information+ Conference presentation*.

Akerlof, G.A., 1970. 4. The market for 'lemons': quality uncertainty and the market mechanism. *Market Failure or Success*, p.66.

Alam, A. and Khan, S., 2014. 1 STRATEGICMANAGEMENT: MANAGINGMERGERS &ACQUISITIONS.

Altman, E.I., 1983. Predicting corporate bankruptcy: the Z-score model. *Corporate Financial Distress: A Complete Guide to Predicting, Avoiding and Dealing with Bankruptcy*.

Ang, Y.Q., Chia, A. and Saghafian, S., 2022. Using Machine Learning to Demystify Startups' Funding, Post-Money Valuation, and Success. In *Innovative Technology at the Interface of Finance and Operations* (pp. 271-296). Springer, Cham.

Antretter, T., Blohm, I., Grichnik, D. and Wincent, J., 2019. Predicting new venture survival: A Twitter-based machine learning approach to measuring online legitimacy. *Journal of Business Venturing Insights*, *11*, p.e00109.

Ashforth, B.E. and Gibbs, B.W., 1990. The double-edge of organizational legitimation. Organization science, 1(2), pp.177-194.

Blank, S.G., 2006. The Four Steps to the Epiphany. lulu. com.

Breiman, L., 2001. Random forests. *Machine learning*, *45*(1), pp.5-32.

Breiman, L., Friedman, J., Olshen, R. and Stone, C., 1984. Cart. *Classification and Regression Trees*.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, pp.321-357.

Clark, M. and Melancon, J., 2013. The influence of social media investment on relational outcomes: A relationship marketing perspective. *International Journal of Marketing Studies*, *5*(4), p.132.

Davila, A., Foster, G. and Gupta, M., 2003. Venture capital financing and the growth of startup firms. Journal of business venturing, 18(6), pp.689-708.

Deephouse, D.L. and Carter, S.M., 2005. An examination of differences between organizational legitimacy and organizational reputation. Journal of management Studies, 42(2), pp.329-360.

Dellermann, D., Lipusch, N., Ebel, P., Popp, K.M. and Leimeister, J.M., 2021. Finding the unicorn: Predicting early stage startup success through a hybrid intelligence method. *arXiv preprint arXiv:2105.03360*.

Etter, M., Colleoni, E., Illia, L., Meggiorin, K. and D'Eugenio, A., 2018. Measuring organizational legitimacy in social media: Assessing citizens' judgments with sentiment analysis. *Business & Society*, *57*(1), pp.60-97.

Eugene, L.Y. and Yuan, S.T.D., 2012, August. Where's the money? the social behavior of investors in facebook's small world. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 158-162). IEEE.

Geier, B., 2015. What did we learn from the dotcom stock bubble of 2000. *Time. Com. http://time. com/3741681/2000-dotcom-stock-bust/.[Visto 15-05-17]*.

Grandini, M., Bagli, E. and Visani, G., 2020. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*.

Guo, B., Lou, Y. and Pérez-Castrillo, D., 2015. Investment, duration, and exit strategies for corporate and independent venture capital-backed start-ups. *Journal of Economics & Management Strategy*, *24*(2), pp.415-455.

Kadam, A. and Ayarekar, S., 2014. Impact of Social Media on Entrepreneurship and Entrepreneurial Performance: Special Reference to Small and Medium Scale Enterprises. *SIES Journal of Management*, *10*(1).

Kortum, S. and Lerner, J., 2001. Does venture capital spur innovation?. In Entrepreneurial inputs and outcomes: New studies of entrepreneurship in the United States. Emerald Group Publishing Limited.

Krüger, Frank. 2016. "Activity, Context, and Plan Recognition with Computational Causal Behaviour Models." ResearchGate, December. Accessed 2019-08-20.

Kuppuswamy, V. and Bayus, B.L., 2017. Does my contribution to your crowdfunding project matter?. Journal of Business Venturing, 32(1), pp.72-89.

Li, D., 2010. *The life cycle of initial public offering companies: a panel analysis of Chinese listed companies* (Doctoral dissertation, Salford: University of Salford).

Liaw, A. and Wiener, M., 2002. Classification and regression by randomForest. *R news*, *2*(3), pp.18-22.

Liew, J.K.S. and Wang, G.Z., 2016. Twitter sentiment and IPO performance: A cross-sectional examination. The Journal of Portfolio Management, 42(4), pp.129-135.

Machiraju, H.R., 2007. *Mergers, acquisitions and takeovers*. New Age International.

Makinen, M., Haber, D. and Raymundo, A., 2012. Acqui-hires for growth: planning for success. *Venture Capital Review*, *28*, pp.31-42.

Oshiro, T.M., Perez, P.S. and Baranauskas, J.A., 2012, July. How many trees in a random forest?. In *International workshop on machine learning and data mining in pattern recognition* (pp. 154-168). Springer, Berlin, Heidelberg.

Ravisankar, P., Ravi, V., Rao, G.R. and Bose, I., 2011. Detection of financial statement fraud and feature selection using data mining techniques. *Decision support systems*, *50*(2), pp.491-500.

Reis, E., 2011. The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses. Currency. New York: Crown Business, 27, pp.2016-2020.

Reuber, A.R. and Fischer, E., 2011. International entrepreneurship in internet-enabled markets. Journal of Business venturing, 26(6), pp.660-679.

Ries, E., 2011. *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Currency.

Rogers, R., 2016. Mergers and Acquisitions Review, Financial Advisors. *Thomson Reuters report*.

Ross, G., Das, S., Sciro, D. and Raza, H., 2021. CapitalVX: A machine learning model for startup selection and exit prediction. *The Journal of Finance and Data Science*, *7*, pp.94-114.

Shepherd, D.A. and Zacharakis, A., 2003. A new venture's cognitive legitimacy: An assessment by customers. *Journal of Small Business Management*, *41*(2), pp.148-167.

Thiel, P. and Masters, B., 2014. Zero to one, notes on start-ups, or how to build the future; crown business.

Wang, H., Zheng, H. (2013). True Positive Rate. In: Dubitzky, W., Wolkenhauer, O., Cho, KH., Yokota, H. (eds) Encyclopedia of Systems Biology. Springer, New York, NY. https://doi.org/10.1007/978-1-4419-9863-7_255

Wei, C.P., Jiang, Y.S. and Yang, C.S., 2008, December. Patent analysis for supporting merger and acquisition (m&a) prediction: A data mining approach. In *Workshop on E-Business* (pp. 187-200). Springer, Berlin, Heidelberg.

Wong, A.Y., 2002. Angel finance: the other venture capital. Available at SSRN 941228.

Xiang, G., Zheng, Z., Wen, M., Hong, J., Rose, C. and Liu, C., 2012. A supervised approach to predict company acquisition with factual and topic features using profiles and news articles on techcrunch. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 6, No. 1, pp. 607-610).

Yin, D., Li, J. and Wu, G., 2021. Solving the Data Sparsity Problem in Predicting the Success of the Startups with Machine Learning Methods. *arXiv preprint arXiv:2112.07985*.

Zhang, D. and Zhou, L., 2004. Discovering golden nuggets: data mining in financial application. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *34*(4), pp.513-522.

Zhang, Q., Ye, T., Essaidi, M., Agarwal, S., Liu, V. and Loo, B.T., 2017, November. Predicting startup crowdfunding success through longitudinal social engagement analysis. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 1937-1946).

Appendix 1

Link to dataset from CrunchBase and Data-World:

https://data.crunchbase.com/docs/daily-csv-export

https://data.world.com/datanerd/startup-venture-funding

Link to Project App:

https://ziplink.webprojectdev.com

QR-Code to Project App:

Appendix 2

Features of combined funding data and crunched tweets

S/N	Column	Missing values	Unique values
1	market	0	19
2	funding_total_usd	0	17094
3	status	0	3
4	country_code	0	20
5	founded_month	0	12
6	founded_year	0	20
7	age_first_funding	0	3201
8	age_last_funding	0	4159
9	first_funding_year	0	23
10	last_funding_year	0	23
11	age_of_startup	0	20
12	funding_type	0	14
13	market_encode	0	19
14	country_code_encode	0	20
15	funding_type_encode	0	14
16	name	0	19687
17	twt_name	0	19651
18	twt_followersCount	0	6159
19	twt_date	0	14
20	twt_friendsCount	0	3672
21	twt_statusesCount	0	6863
22	twt_listedCount	0	1308
23	twt_favouritesCount	0	4383

24	twt_verified	0	2
25	tweet_likes	0	1564
26	tweet_retweets	0	1021
27	tweet_reply	0	500
28	tweet_quote	0	309
29	sub_neg	0	223
30	sub_pos	0	184
31	sub_neu	0	91
32	pol_neg	0	102
33	pol_pos	0	218
34	pol_neu	0	208

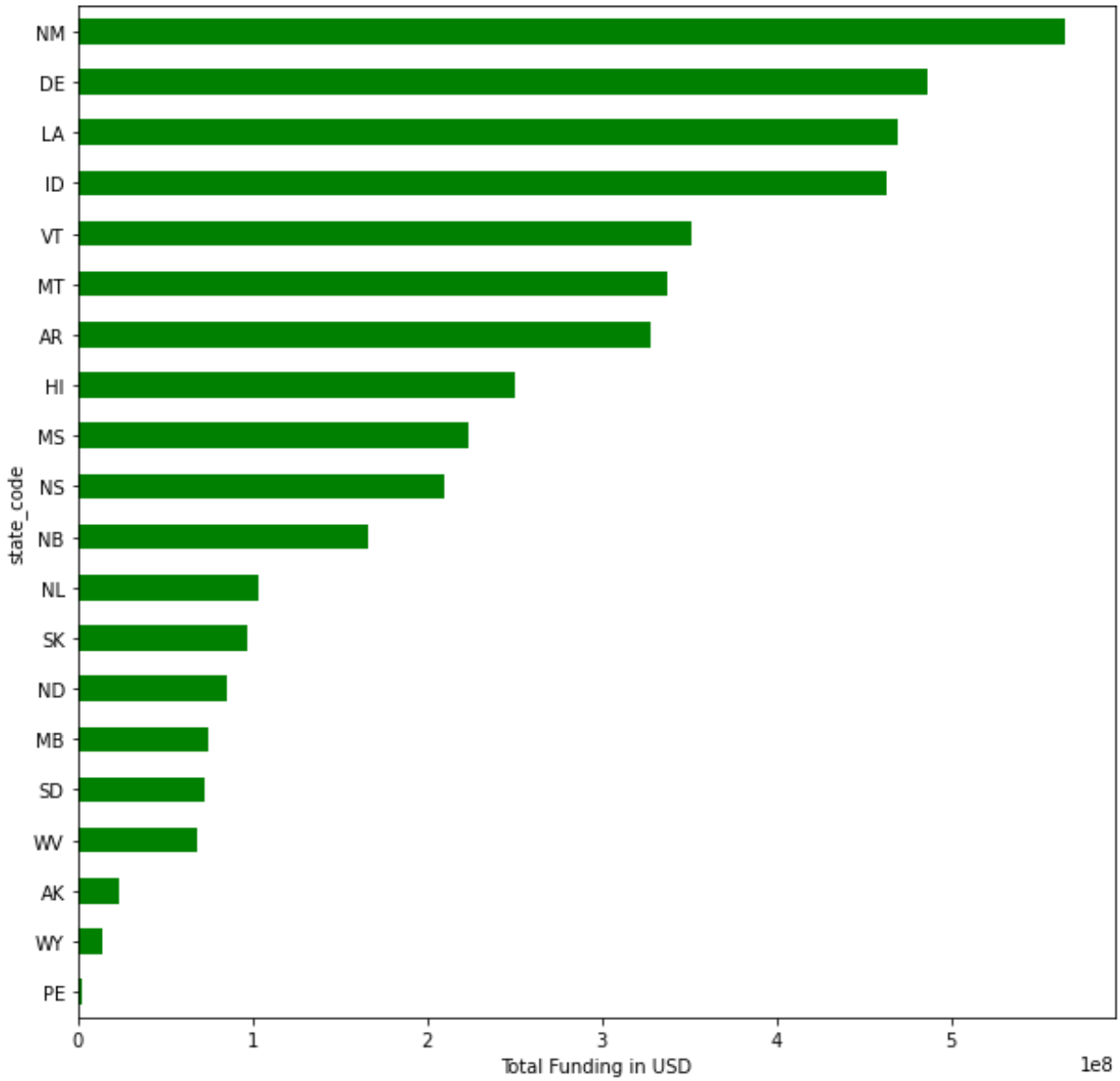
Appendix 3

Results of combined Sampling and trained models

Model	Sampling method	CrossValMeans	CrossValError	Recall	Precision	F1-Score
RandomForest	SMOTE	0.798452	0.012036	0.750000	0.752054	0.750444
RandomForest	ADASYN	0.793249	0.009441	0.744972	0.746715	0.744064
GradientBoosting	SMOTE	0.748214	0.014098	0.724167	0.724186	0.724034
GradientBoosting	ADASYN	0.743445	0.009136	0.723743	0.722278	0.721845
RandomForest	Stratified	0.715340	0.019790	0.699790	0.674905	0.679113
GradientBoosting	Stratified	0.716928	0.015241	0.691903	0.669474	0.675340
AdaBoost	SMOTE	0.701071	0.011523	0.641111	0.641777	0.641383
DecisionTree	SMOTE	0.702500	0.018879	0.636389	0.637088	0.636664
DecisionTree	ADASYN	0.688374	0.008765	0.627654	0.627031	0.627184
AdaBoost	ADASYN	0.690892	0.007912	0.625698	0.624367	0.624896
AdaBoost	Stratified	0.622269	0.020238	0.605678	0.611668	0.608526
DecisionTree	Stratified	0.624071	0.015773	0.598318	0.602243	0.600230
KNeighboors	Stratified	0.552629	0.020732	0.534700	0.507760	0.518518
KNeighboors	SMOTE	0.591786	0.014090	0.529167	0.527461	0.514823
MultipleLayerPerceptron	Stratified	0.486493	0.136294	0.511567	0.491406	0.498047
LogisticRegression	Stratified	0.631057	0.001962	0.628812	0.475496	0.490106
SVC	Stratified	0.631058	0.000652	0.629338	0.397686	0.487386
KNeighboors	ADASYN	0.561716	0.016195	0.497765	0.493856	0.481989
LogisticRegression	SMOTE	0.457976	0.012384	0.450278	0.461733	0.449568
MultipleLayerPerceptron	SMOTE	0.436667	0.041859	0.411111	0.408008	0.407680
LogisticRegression	ADASYN	0.436493	0.011026	0.423743	0.432891	0.398290
MultipleLayerPerceptron	ADASYN	0.413517	0.046367	0.389665	0.391537	0.382487
SVC	SMOTE	0.449762	0.009595	0.430000	0.410515	0.347622
SVC	ADASYN	0.438647	0.009954	0.422626	0.451115	0.336625

Appendix 4

Bar-plot of start-up funding data using state codes



Appendix 5

Programme code to building a simple database to store JSON as Blob

```
CREATE TABLE "analysis" (
	"id"	INTEGER NOT NULL UNIQUE,
	"name"	TEXT NOT NULL,
	"link"	TEXT NOT NULL UNIQUE,
	"data"	BLOB NOT NULL,
	"datetime"	TEXT,
	PRIMARY KEY("id" AUTOINCREMENT)
);
```

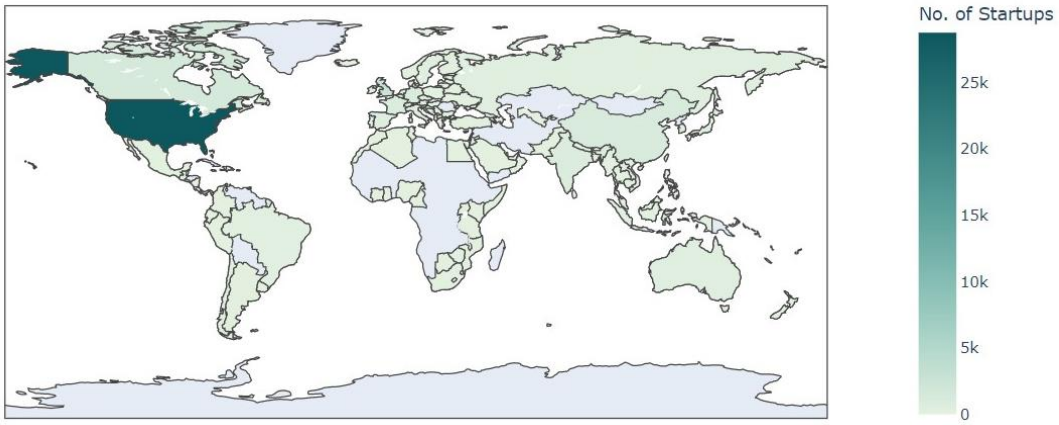
Appendix 6

Function for checking mission data

```
# function to check missing data
def check_missingData(df,columns=None):
    if columns==None:
        columns = df.columns
    dfvalues = pd.DataFrame(columns=['Column', 'Missing values',
                                     'Unique Features',
                                     'Percentage null'])
    for j, i in enumerate(columns):
        percentage = str(round((df[i].isnull().sum()/df.shape[0])*100,
        dfvalues.loc[j] = [i, df[i].isnull().sum(),
                           df[i].nunique(), f'{percentage}%']
    return dfvalues
```

Process for plotting a state/country map

```
fig_map = go.Figure()
fig_map.add_trace(go.Choropleth(locations=df['country_code'].value_counts().index,
                                z=df['country_code'].value_counts(),
                                colorscale=px.colors.sequential.Mint,
                                colorbar_title="No. of Startups"))
fig_map.update_layout(title_text="Distribution of startups in countries.",
                      title_x=0.5, title_font_size=20)
fig_map.show()
```



Appendix 7

Function to create bar plots of start-ups status against numerical continuous funding data

```
# checking for imbalance data using status and fundings

funding_col = ['seed', 'venture','equity_crowdfunding', 'undisclosed', 'convertible_note',
               'debt_financing', 'grant', 'private_equity', 'post_ipo_equity','post_ipo_debt',
               'secondary_market', 'product_crowdfunding', 'round_A','round_B', 'round_C',
               'round_D', 'round_E', 'round_F', 'round_G','round_H']

fig = plt.figure(figsize = (21,21))

for c,num in zip(funding_col, range(1,22)):
    ax = fig.add_subplot(7,3,num)
    x = np.random.rand()
    y = np.random.rand()
    z = np.random.rand()
    ax =  df3.groupby('status')[c].agg('mean').plot(kind='bar',alpha = .8, color=(x,y,z))

    ax.set_xlabel('')
    temp = 'Median "' + c.replace('_',' ') + '" value'
    ax.set_ylabel(temp)
    ax.set_title(c)
    ax.tick_params(axis='x',labelsize=12)
    ax.tick_params(axis='y')
    sns.despine()
    plt.xticks(rotation=45)

plt.tight_layout()
plt.subplots_adjust(hspace=0.7, wspace = 0.5)
```

Appendix 8

Feature Engineering

```
#To reduce the complexity, while processing the date variable fields, i converted the actual dates to
#relative dates, describing the distance from founded to first and last funding and then drop the founded date.
#Subtracting the newer date (funding start) from the older date (business created) will give the days in between
#This is then divided by 365 days to get number of years
#pd.Timedelta(days=365) is used to convert the 365 into a date-time format

startup_df = df.copy()

# startup_df['Age'] = (startup_df['first_funding_at']-startup_df['founded_at']).dt.days /365

startup_df['age_first_funding'] = (startup_df['first_funding_at']-startup_df['founded_at'])/pd.Timedelta(days=365)
startup_df['age_last_funding'] = (startup_df['last_funding_at']-startup_df['founded_at'])/pd.Timedelta(days=365)
startup_df['founded_month'] = pd.DatetimeIndex(startup_df['founded_at']).month
startup_df['founded_year'] = pd.DatetimeIndex(startup_df['founded_at']).year
startup_df['first_funding_year'] = pd.DatetimeIndex(startup_df['first_funding_at']).year
startup_df['last_funding_year'] = pd.DatetimeIndex(startup_df['last_funding_at']).year

current_year = dt.datetime.now().year
startup_df['age_of_startup'] = current_year - pd.DatetimeIndex(startup_df['founded_at']).year
```

```
# adding all funding to get funding total
startup_df["funding_total_usd"] = startup_df.iloc[:, 18:].sum(axis=1)

startup_df.head(2)
```

Function for removing outliers

```
# removing outliers in the dataset
# this will be commented out because remove outliers might drop inportant data
# instead i will use capping to cap the data
def remove_outliers(raw_df, features):
    index_list = []
    for ft in features:
        try:
            percentile25 = raw_df[ft].quantile(0.25)
            percentile75 = raw_df[ft].quantile(0.75)
            IQR = percentile75 - percentile25
            upper_limit = percentile75 + 1.5 * IQR
            lower_limit = percentile25 - 1.5 * IQR
            ls = raw_df.index[(raw_df[ft]<lower_limit) | (raw_df[ft]>upper_limit)]
            index_list.extend(ls)
        except Exception as e:
            print(f'Error for {ft} => {e}')
    index_list = list(dict.fromkeys(index_list))
    df_clean = raw_df.drop(index_list)
    return df_clean

data_outlier = startup_df.copy()

data_outlier = remove_outliers(data_outlier, numerics)
data_outlier.shape
```

```
# capping of numeric continuous features
for feature in numerics:
    percentile25 = startup_df[feature].quantile(0.25)
    percentile75 = startup_df[feature].quantile(0.75)
    IQR = percentile75 - percentile25
    upper_limit = percentile75 + 1.5 * IQR
    lower_limit = percentile25 - 1.5 * IQR
    startup_df[feature] = np.where(startup_df[feature] > upper_limit, upper_limit,
                                   np.where(startup_df[feature] < lower_limit, lower_limit,
                                            startup_df[feature]))
startup_df.shape
```

Appendix 9

Function for extracting and building tweet

```
def build_tweets(name, start, end):
    #print(name)
    query = f"(from:{name}) until:{end}-01-01 since:{start}-01-01"
    tweets = []
    user_details = {}
    limit = 200
    allt = []
    columns = ['user', 'date', 'tweet', 'retweets', 'likes', 'reply', 'quote']
    columns2 = ['user','date','displayname','description','followersCount', 'friendsCount', 'statusesCount', 'listedCount',
                    'favouritesCount', 'linkUrl', 'profileImageUrl', 'profileBannerUrl', 'verified']
    try:
        for tweet in sntwitter.TwitterSearchScraper(query).get_items():
            if len(tweets) == limit:
                break
            else:
                allt.append(tweet)
                data = [tweet.user.username, tweet.date, tweet.content, tweet.retweetCount, tweet.likeCount,
                        tweet.replyCount, tweet.quoteCount]
                tweets.append(data)
                if name not in user_details:
                    data2 = [tweet.user.username, tweet.user.created, tweet.user.displayname, tweet.user.description,
                             tweet.user.followersCount, tweet.user.friendsCount, tweet.user.statusesCount,
                             tweet.user.listedCount, tweet.user.favouritesCount, tweet.user.linkUrl,
                             tweet.user.profileImageUrl, tweet.user.profileBannerUrl, tweet.user.verified]
                    user_details[name] = data2
        df = pd.DataFrame(tweets, columns=columns)
        df2 = pd.DataFrame(list(user_details.values()), columns=columns2)
    except:
        df = pd.DataFrame()
        df2 = pd.DataFrame()
    return df, df2
```

Appendix 10

Function for sentimental analysis of the tweet data

```
def define_sentiment(tweet_value):
    if tweet_value > 0:
        return 'Positive'
    elif tweet_value == 0:
        return 'Neutral'
    else:
        return 'Negative'

def define_sub_sentiment(tweet_value):
    if tweet_value > 0.5:
        return 'Positive'
    elif tweet_value == 0.5:
        return 'Neutral'
    else:
        return 'Negative'


def clean_tweets(origin_name, cleaned_name):
    global df
    global tweetdf

    df_twitter = tweetdf[tweetdf['user']==cleaned_name][['tweet','retweets','likes','reply','quote']]
    new_df = comb_df[comb_df['name']==origin_name].copy()

    for _,row in df_twitter.iterrows():
        row['tweet'] = re.sub('http\S+', '', row['tweet'])
        # remove hashtags
        row['tweet'] = re.sub('#\S+', '', row['tweet'])
        # remove mention of people's nametag
        row['tweet'] = re.sub('@\S+', '', row['tweet'])
        # newline cleanup
        row['tweet'] = re.sub('\\n', '', row['tweet'])


    df_twitter['polarity'] = df_twitter['tweet'].map(lambda tweets: textblob.TextBlob(tweets).sentiment.polarity)
    df_twitter['subjectivity'] = df_twitter['tweet'].map(lambda tweets: textblob.TextBlob(tweets).sentiment.subjectivity)
    df_twitter['pol_result'] = df_twitter['polarity'].apply(define_sentiment)
    df_twitter['sub_result'] = df_twitter['subjectivity'].apply(define_sub_sentiment)
    # tweet_result = {'Negative': 18, 'Neutral': 36, 'Positive': 46}
    pol_result = df_twitter.groupby(['pol_result']).size()
    sub_result = df_twitter.groupby(['sub_result']).size()
    # fft["final"] = np.where(fft["sub_result"] == fft["pol_result"], fft["pol_result"], 0)
    new_df['tweet_likes'] = df_twitter['likes'].sum()
    new_df['tweet_retweets'] = df_twitter['retweets'].sum()
    new_df['tweet_reply'] = df_twitter['reply'].sum()
    new_df['tweet_quote'] = df_twitter['quote'].sum()

    if "Negative" in sub_result.keys():
        new_df['sub_neg'] = sub_result['Negative']
    else:
        new_df['sub_neg'] = 0
    if "Positive" in sub_result.keys():
        new_df['sub_pos'] = sub_result['Positive']
    else:
        new_df['sub_pos'] = 0
    if "Neutral" in sub_result.keys():
        new_df['sub_neu'] = sub_result['Neutral']
    else:
        new_df['sub_neu'] = 0

    if "Negative" in pol_result.keys():
        new_df['pol_neg'] = pol_result['Negative']
    else:
        new_df['pol_neg'] = 0
    if "Positive" in pol_result.keys():
        new_df['pol_pos'] = pol_result['Positive']
    else:
        new_df['pol_pos'] = 0
    if "Neutral" in pol_result.keys():
        new_df['pol_neu'] = pol_result['Neutral']
    else:
        new_df['pol_neu'] = 0

    return new_df
```

Appendix 11

Building backend API using FLASK and CORS

```python
from flask import Flask, request, jsonify
import pandas as pd
import appmodules as apm
import datetime as dt
from flask_cors import CORS
import sqlite3
import json

app = Flask(__name__)

# load data at the start of software...this will help reduce low processing time
df = apm.import_csv_data()
df_cleaned = apm.import_csv_data('startups_cleaned.csv')

"""
# if I want user to access all document route
CORS(app)
# if I want only me to access it from my app origin
cors = CORS(app, resources={r"/api/*": {"origins": "http://12.0.0.1:3000"}})

# to send back response you can either use
return jsonify(data=data, status=status)
------OR-------
return Response(dumps({'content': 'Hello world'}), mimetype='text/json')
# read more on mimetype data here: https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types
"""

# if I want user to only access the /api/ route
cors = CORS(app, resources={r"/api/*": {"origins": "*"}})

```

A section of the python code with calls the prediction path

```python

@app.route('/api/startup_prediction', methods=['POST'])
def startup_prediction():
    global df
    try:
        request_data = request.get_json()
        collect_data = {}
        # calculations
        current_date = dt.datetime.now()
        start_date = pd.to_datetime(request_data['startdate'], format='%Y-%m-%d')
        first_funding = pd.to_datetime(request_data['first_funding'], format='%Y-%m-%d')
        age_first_funding = (first_funding - start_date) / pd.Timedelta(days=365)
        market = request_data['category'].lower()
        market_encode = df[df['market'] == market]['market_encode'].values[0]
        funding_type = request_data['funding_type'].replace(" ", "_").lower()
        funding_type_encode = df[df['funding_type'] == funding_type]['funding_type_encode'].values[0]
        country_code_encode = df[df['country_code'] == request_data['country']]['country_code_encode'].values[0]

        collect_data['funding_total_usd'] = [float(request_data['funding_total_usd'])]
        collect_data['founded_month'] = [start_date.month]
        collect_data['founded_year'] = [start_date.year]
        collect_data['age_first_funding'] = [age_first_funding]
        collect_data['first_funding_year'] = [first_funding.year]
        collect_data['last_funding_year'] = [first_funding.year]
        # collect_data.append(current_date.year) #last funding year
        collect_data['age_of_startup'] = [current_date.year - start_date.year]
        collect_data['market_encode'] = [market_encode]  # market encode
        collect_data['country_code_encode'] = [country_code_encode]  # country encode
        collect_data['funding_type_encode'] = [funding_type_encode]  # funding type encode

```

Appendix 12

ReactJS Routes code for the App to build the Home and Search routes

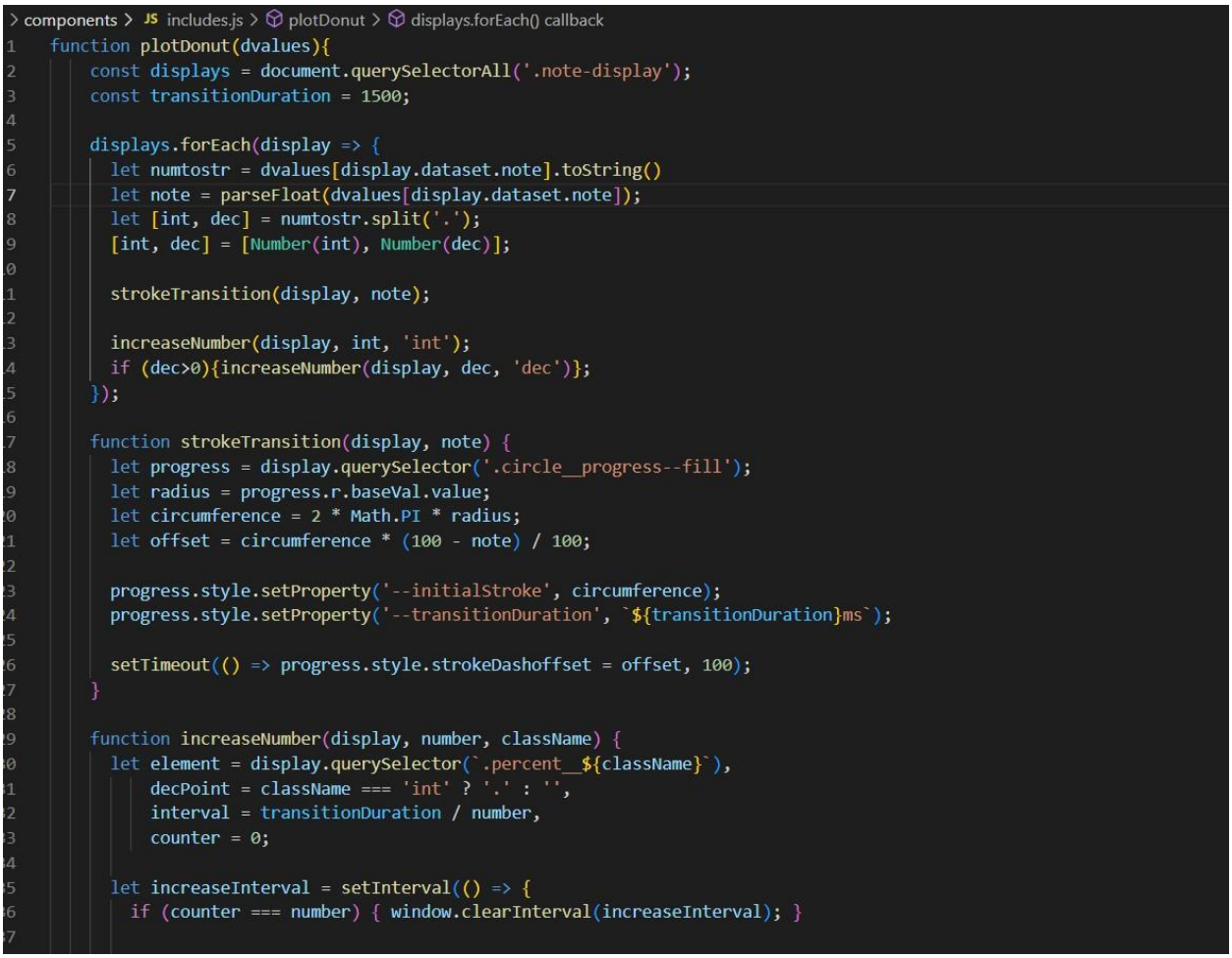
```
import { BrowserRouter as Router,Routes, Route, Link } from 'react-router-dom';
import Home from './Home';
import Search from './Search';



function App() {
  return (
    <Router>
        <Routes>
            <Route exact path='/' element={<Home />}></Route>
            <Route exact path='/search' element={<Search />}></Route>
        </Routes>
    </Router>
  );
}

export default App;
```

React **Home** component

```
return (
  <>
    <div id="app" className="app">
      <div id="content" className="app-content">
        <h1 className="page-header mb-3 text-center appTitle">
          Hello, <small style={{color:'#595959'}}>welcome back.</small>
          <small className='d-block' style={{color:'#595959', padding:'5px', maxWidth:'478px', margin:'0 auto', textAlign:'left'}}
          >This App aims to predict start-up's possible future status using trained data that is a combination of the start-up
          funding and online validity. <span className='appinstallation'> You can install <strong>App</strong> both on PC and
          Mobile <button type='button' className='btn btn-secondary btn-sm' onClick={()=>{installApp()}}>INSTALL APP <i
          className='fa fa-download'></i></button></span> </small>
        </h1>
        <div className="row">
          <Profile data={tempmem.data.profile} />
          <Formdata countryFlagHandler={countryFlagHandler}
          ipLocation={ipLocation} submitForm={SubmitForm} setData={setData} closePreferences={closePreferences} />
          <Donut data={tempmem.data.donutdata} clickInfo={clickInfo} getKeyByValue={getKeyByValue} />
        </div>
        <div className="display_on_reult" style={{display: "none"}}>
          <div className='row'>
            <Tweet data={tempmem.data.twt_data} />
            <Ratings data={tempmem.data.ratings} />
          </div>
          <Suggestions data={tempmem.data.recommend} />
          <Analytics data={tempmem.data.keywordsAnalytics} />
          <Barplot data={tempmem.data.bardata} />
          <Qrcode searchkey={tempmem.data.shortLink} profile={tempmem.data.profile} />
          <Footer />
        </div>
      </div>
    </div>
  </>
);
```

Appendix 13

Included helper functions



```
> components > JS includes.js > plotDonut > displays.forEach() callback
function plotDonut(dvalues){
    const displays = document.querySelectorAll('.note-display');
    const transitionDuration = 1500;

    displays.forEach(display => {
      let numtostr = dvalues[display.dataset.note].toString()
      let note = parseFloat(dvalues[display.dataset.note]);
      let [int, dec] = numtostr.split('.');
      [int, dec] = [Number(int), Number(dec)];

      strokeTransition(display, note);

      increaseNumber(display, int, 'int');
      if (dec>0){increaseNumber(display, dec, 'dec')};
    });

    function strokeTransition(display, note) {
      let progress = display.querySelector('.circle__progress--fill');
      let radius = progress.r.baseVal.value;
      let circumference = 2 * Math.PI * radius;
      let offset = circumference * (100 - note) / 100;

      progress.style.setProperty('--initialStroke', circumference);
      progress.style.setProperty('--transitionDuration', `${transitionDuration}ms`);

      setTimeout(() => progress.style.strokeDashoffset = offset, 100);
    }

    function increaseNumber(display, number, className) {
      let element = display.querySelector(`.percent__${className}`),
          decPoint = className === 'int' ? '.' : '',
          interval = transitionDuration / number,
          counter = 0;

      let increaseInterval = setInterval(() => {
        if (counter === number) { window.clearInterval(increaseInterval); }
```

Appendix 14

A proper representation of the JSON data passed from the backend to the frontend.


```
▼{bardata: {…}, donutdata: {…}, keywordsAnalytics: {…}, profile: {…}, ratings: Array(5), …}
  ▼bardata:
    ▶2010: {acquired: 11, closed: 16, operating: 96}
    ▶2011: {acquired: 8, closed: 12, operating: 110}
    ▶2012: {acquired: 4, closed: 2, operating: 113}
    ▶2013: {acquired: 0, closed: 1, operating: 67}
    ▶2014: {acquired: 0, closed: 0, operating: 32}
    ▶[[Prototype]]: Object
  ▼donutdata:
      msg: "This startup has high potential for success (threshold is => 0.452)"
    ▶values: {acquire: 0, failure: 0, success: 100}
    ▶[[Prototype]]: Object
  ▼keywordsAnalytics:
    ▶country_trend: {Andorra: 71, Antigua & Barbuda: 66, Benin: 67, Gibraltar: 73, Guam: 100, …}
    ▶related_keywords: {binance: 34, bitcoin: 96, coin: 42, coinbase bitcoin: 100, coinbase coin: 41, …}
    ▶rising_keywords: {binance vs coinbase: 36250, cardano: 80050, cardano coinbase: 77950, coinbase nft: 59150,
    ▶trend: {2017-09-30: 13, 2017-10-31: 30, 2017-11-30: 56, 2017-12-31: 234, 2018-01-31: 82, …}
    ▶[[Prototype]]: Object
  ▼profile:
      banner: "https://pbs.twimg.com/profile_banners/574032254/1646095309"
      displayname: "Coinbase"
      favourites: 1276
      followers: 5340634
      friends: 24
      img: "https://pbs.twimg.com/profile_images/1484586799921909764/A9yYenz3_normal.png"
      listed: 22725
      statuses: 4512
      username: "coinbase"
    ▶[[Prototype]]: Object
  ▶ratings: (5) [646, 356, 283, 317, 5817]
  ▼recommend:
    ▶Age: {data: 3, expected: 15, remark: 'Average'}
    ▶Followers: {data: 5340634, expected: 2041918, remark: 'Excellent'}
    ▶Friends: {data: 24, expected: 53238, remark: 'Poor'}
    ▶Funding: {data: 1000000, expected: 2256804032, remark: 'Poor'}
    ▶Likes: {data: 130452, expected: 95340, remark: 'Excellent'}
    ▶Tweets: {data: '> 200', expected: '> 200', remark: 'Excellent'}
    ▶Twitter age: {data: 10, expected: 6, remark: 'Excellent'}
    ▶URL: {data: 2, expected: 2, remark: 'Excellent'}
    ▶Verified: {data: 1, expected: 1, remark: 'Excellent'}
    ▶[[Prototype]]: Object
    shortLink: "coinbase166263926786D"
  ▼twt_data:
      likes: 130452
      negative: 17
      neutral: 62
      positive: 121
      retweets: 30904
      total_tweets: 200
    ▶[[Prototype]]: Object
  ▶[[Prototype]]: Object
```

Appendix 15

Approved ethics application for this project

Ethical clearance for research and innovation projects

Project status

Status

Approved

Actions

Date	Who	Action	Comments
09:31:00 29 July 2022	Femi Isiaq	Supervisor approved	
15:14:00 05 July 2022	Chukwuemeka Gabriel Agbo	Principal investigator submitted	

Get Help