CLASSIFYING THREE TIERS OF SUCCESS IN CROWDFUNDING WITH MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING

by

Sze Wing Wong

A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computational Sciences and Informatics

Committee:

Robert Axtell
Dr. Robert L. Axtell, Committee Chair

WG Kennedy
Dr. William G. Kennedy, Committee Member

Dr. Igor Griva, Committee Member

O Gkountouna
Dr. Olga Gkountouna, Committee Member

Jason M Kinser
Dr. Jason Kinser, Department Chairperson

Donna M. Fox
Dr. Donna M. Fox, Associate Dean, Office of Student Affairs & Special Programs, College of Science

Dr. Fernando R. Miralles-Wilhelm, Dean, College of Science

Date: 12|01|2021

Fall Semester 2021
George Mason University
Fairfax, VA

Classifying Three Tiers of Success in Crowdfunding with Machine Learning and Natural Language Processing

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

by

Sze Wing Wong
Master of Science
University of New Mexico, 2009
Bachelor of Science
New Mexico Institute of Mining and Technology, 2008

Director: Robert L. Axtell, Professor
Department of Computational Sciences and Informatics

Fall Semester 2021
George Mason University
Fairfax, VA

DEDICATION

To my wonderful parents and cousin Iris, who have always unconditionally loved and supported me.

ACKNOWLEDGEMENTS

First and foremost, I want to express my gratitude to Dr. Robert Axtell, my dissertation advisor, for his mentorship and assistance. Without his tremendous support, I would not be able to complete this dissertation. I would like to thank Dr. William Kennedy for his guidance and all the detailed feedback throughout this dissertation. Dr. Kennedy, I am extremely grateful for all of your assistance, time, patience, and invaluable advice in helping me make the most of my Ph.D. experience. I would like to thank my committee members, Dr. Igor Griva and Dr. Olga Gkountouna, for their support and assistance to complete this journey. I would also like to thank Dr. Edward Wegman for the stimulating discussion and the encouragement to pursue my research interest.

In addition, I would like to express my gratitude to my colleagues and friends, particularly Christie Woo, Grace Kong, Khai Nguyen, and Khang Nguyen, for their crucial help and discussion to this dissertation.

I want to thank Stephon for his unwavering love and for always being there for me. Stephon, thank you for always getting my morning coffee ready and reminding me how much you can’t wait for my Ph.D. to finish. Last, I must thank my parents and other family members, for instilling in me all the wisdom and moral ideals I need to face life's challenges. They are constantly reminding me that I am capable and strong enough to accomplish anything with dedication and hard work.

Twelve years ago, I would never have considered pursuing another graduate degree, let alone a Ph.D. It all started with an innocent conversation with a dear friend in 2016. Thank you for your inspiration, Dana!

After all, everything is possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

CLASSIFYING THREE TIERS OF SUCCESS IN CROWDFUNDING WITH MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING

Sze Wing Wong, Ph.D.

George Mason University, 2021

Dissertation Director: Dr. Robert L. Axtell

With the continued growing adoption of crowdfunding for raising capital, much research and numerous studies have been done using econometric analyses and traditional machine learning models to identify key indicators and to classify success in campaigns. These findings are informative and hold beneficial insights for both entrepreneurs and investors. However, in recent years, the increasing number of successful campaigns far exceeds the number of failed campaigns. Therefore, previous studies focusing on binary classification are no longer sufficient to capture different levels of emergent success in crowdfunding. This dissertation examines three tiers of success in reward-based campaigns of the 2019 Kickstarter data to gain insights into the evolved crowdfunding phenomenon. How I demonstrate new key indicators are identified by exploiting campaign information relates to "people" versus "products" with hierarchical multiple and ordinal logistic regressions. In conjunction, I adopt Binary

Particle Swarm Optimization (BPSO) for feature selection to classify campaign success. The BPSO improved Extreme Gradient Boosting (XGBoost) classifier shows favorable model performance in multiclass classification. Most importantly, the interest in this research extends beyond using categorical and numeric features but also fuses multiple textual information with natural language processing (NLP) and deep neural networks (DNNs) for multiclass classification. The proposed multimodal Long Short Term Memory (LSTM) concatenates BPSO selected metadata with the project's pitch and the creator's biography text yields the best performing multiclass classification accuracy of 71.04% after tuning. However, the BPSO improved Extreme Gradient Boosting (XGBoost) classifier achieves the highest accuracy of 74.61% overall. These impactful findings allow entrepreneurs and researchers to gain further insights and to optimize the cyber marketing space effectively.

1. INTRODUCTION

1.1 Definition of Crowdfunding

According to the definition from J. Pedersen et al. [1], crowdsourcing is a community based model mobilized by "people-centric web technologies to solve individual, organizational, and societal problems using a dynamically formed crowd of people who respond to an open call for participation." Crowdsourcing is a rising practice in many businesses and academia communities; it allows the public to participate the process and contribute to efforts or in monetization, to shape development and optimize resources for achieving the final goals. Therefore, it is essential to understand the various types of crowdsourcing formation, considering crowdsourcing is still a newly adopted practice. Crowdsourcing formation comes in different forms. Four categories are characterized by Hoi et al. [2] as follow:

1. Crowd voting is a common crowdsourcing method that combines and leverages public citizens' opinions or judgement for results. For example, foodies' ratings on Yelp allows customers and business owners for ranking and sorting reviews of businesses before trying out food services.

2. Crowd solving is a crowdsourcing method that allows the public to contribute expertise and wisdoms for solving some industrial problems or specific tasks. For example, on Challenge.gov website, numerous of U.S federal agencies have recruited

public efforts to assist on research ideas and problem-solving methodologies by rewarding monetary prize for best submission. Similar to a data science project platform such as Kaggle website (http://www.kaggle.com).

3. Crowd searching is another form of crowdsourcing to seek assistance from citizens to participate a search for lost and missing items or persons.

4. Crowdfunding is a special kind of crowdsourcing method to collaborate and pool interest from the public for achieving a goal. However, unlike other crowdsourcing types that rely mainly on the efforts from the public for goal accomplishment as mentioned above, crowdfunding is seeking a direct monetary contribution instead. Crowdfunding itself has various subcategories of funding portals for entrepreneurs to raise capitals, and for investors to invest or donate to be part of a business idea. For example, Kickstarter (http://www.kickstarter.com) is a crowdfunding platform provides startups or capital seekers to advertise their business ideas and reward investors with the finished products or other items, in exchange for monetary investment from citizens.

Crowdfunding has gained an increasing popularity and attracted massive attentions in recent years, especially in the micro financing domain of the modern days. Therefore, many economists and other researchers are motivated to understand the role of crowdfunding offerings as being part of the “financial ecosystem” for raising capitals, and its impact from the economic and the social interconnection. Hence, this dissertation aims to explore crowdfunding and the success factors in raising funds. The next section details background on crowdfunding to help understand some of the objectives in this research.

1.2 Types of Crowdfunding

One of the common obstacles to converting a great idea into a useable product or an actual business service at the first step in entrepreneurship, is to have enough cash flow and capital. According to an article from American Express [3], some of the traditional methods for entrepreneurs and small business owners to raise capital are:

- Business loans through banks,
- Government grants application,
- Angel investors or venture capitalists for equity exchange, and
- Donations from others or bootstrap financing (self-funded).

However, all the listed methods above are not easy to acquire and can be quite cumbersome for capital seekers and investors due to numbers of challenges:

- Various requirements and numerous meetings for entrepreneurs from different potential investors,
- No centralized access for both investors and entrepreneurs,
- Participation is limited to accredited investors who earn $200,000 and have a net worth that exceeded $1,000,000,
- High standard prerequisite in capital seekers credits history for applications,
- Lengthy application process,
- Limited exposure to potential investors due to geography, and

- High equity exchange causing dilution in business ownership.

As a result, many have turned to a more creative financial instrument such as crowdfunding web portals to attract public investors for raising capitals.

There are well over 1.25 million crowdfunding campaigns in the U.S. since 2019, and the estimated total number of campaigns may surpass 1.3 million by 2023 [4]. With the rising popularity of using crowdfunding for many projects, it is necessary to introduce the four main categories of this financing instrument to define the characteristics of different campaigns. Typically, project initiators will choose one of these categories to launch their ideas:

1. Donation-Based Crowdfunding is a not-for-profit crowdfunding approach to raise capitals or asking for resources to support a cause in exchange of appreciation for generosity, without any rewards. The funding targets are not only limited in monetization, but it could also be other resources for projects or time investment for charitable events. Some of the popular donation-based crowdfunding sites or social media platforms would be GoFundMe (http://www.GoFundMe.com) and Crowdrise (http://www.crowdrise.com).
2. Rewards-Based Crowdfunding is a common crowdfunding method to seek monetary investment for creating a product or providing a service, in exchange of a reward, such as a delivery of the early version of the manufactured products or an early viewing of a new film production to investors. Some of the popular rewards-based crowdfunding efforts take place on web or social media platforms are Kickstarter and Indiegogo

(http://www.indiegogo.com). These platforms allow capital seekers to advertise and campaign their ideas, prototypes or products with videos, pictures, textual details, or personal biographies as part of their capital raising campaigns to individuals.

3. Equity-Based Crowdfunding is an approach very similar to the traditional capital formation for seeking monetary investment from investors in exchange of company's shares or equity as financial return. However, instead of processing underwriting and valuation to seek capitals from banks or other ventures, it raises capitals through public platforms to allow interested individuals to crowdfund as part of the shareholders at the companies. Some of the popular equity-based crowdfunding efforts on web or social media platforms are Wefunder (http://www.Wefunder.com), Fundme (http://www.fundme.com), and Seedinvest (http://www.seedinvest.com) websites. Unlike the other types of crowdfunding, equity-based crowdfunding involves an actual transaction, the selling securities. With that key difference, the crowdfunding intermediaries are subject to the compliance of the Regulation Crowdfunding (Reg CF) under the regulation of the U.S. Securities and Exchange Commission (SEC). Reg CF was adopted in 2015 under the Securities Act of 1933 and the Securities and Exchange Act of 1934 to implement the requirements of Title III of the Jumpstart Our Business Startups ("JOBS") Act during the President Obama Administration[5]. The JOBS Act is intended to protect financial investors and capital seekers at the

same time to provide “opportunity to participate in the early capital raising activities of start-up and early-stage companies and businesses”[6]. Not only the intermediaries of capital seekers will be regulated by the JOB Act; the investors are also subjected to the maximum amount of $107,000 in investment during a 12-month period based on individual income and net worth [6].

4. Debt-Based Crowdfunding is a lending model from crowds to fund a project by raising capitals in the form of loans to project initiators without partaking of banks, in exchange, “the rewards are normally the interest and the payback after the lending period” [7] at the interest rate and the principal amount that were first agreed on. This crowdfunding model is also called peer-to-peer (P2P) lending. The partial benefit for this micro lending is not only to help project initiators crowdfund their capitals at a better agreement of interest rates, but also allows investors to profit from a better interest rate of return on their cash assets than a regular bank account. This type of lending can be registered as securities and traded in secondary market; therefore, debt-based crowdfunding could be subjected under the regulation of the SEC compliance as well. Some of the popular debt-based crowdfunding web or social media platforms are Prosper Marketplace (http://www.prosper.com) and Lendingclub (http://www.lendingclub.com). Debt-based crowdfunding can also be used for campaigning to fund individuals for home improvement loans or other purposes, and may not necessarily fund a business or a broad-based

project. Microlending is a very different crowdfunding model with different objectives and fixed returns than reward-based or equity-based in comparison, since the return of profits to investors is more predictable and the risk is much lower. In addition, other non-profit debt-based platform like Kiva (http://www.kiva.com), is a microlending platform with zero interest repay for investors, but more focusing on the support towards good cause to help small businesses.

1.3 Research Objectives

The research interest for this dissertation is to inform and guide project creators to construct and optimize their campaigns before launching their ideas through crowdfunding. For this purpose, my aims are to examine and analyze features that were influential to crowdfunding success when campaigns first launched, and to develop classification models using relevant features at the beginning of the funding period. However, much of the research in crowdfunding included many dynamic features that were updated throughout the funding duration. Therefore, this dissertation addressed the pre-launching aspect of the projects. Another main concern with much of the existing research in reward-based crowdfunding is that there is a lack of distinction between "people" or "product" features used in research, and to further explore their impacts on project success. In addition, most of the research work in crowdfunding is only interested in success or failure outcomes where some of the 'super' successful campaigns that raised over 1.5 times more than their pledged goal should be better understand in their strategies as well. Last, there is also an inadequate number of studies to combine textual information with categorical and numeric features when adopting machine learning algorithms to leverage language modeling for classification. The following objectives are explored and answered in my research for additional understanding in the success of crowdfunding campaigns:

1) What are the indicators and their association strength in crowdfunding campaign success?

   The first objective used empirical study on a class of attributes of products and creators to study the factors drive campaign for different levels of success. A hierarchical multiple regression and a hierarchical ordinal logistic regression were used to assess not only the significance of each factor, but also the strength of association on the crowdfunding success. Indicators were identified and discussed that utilize background information on creators or on team formation attract more investors to fund the campaigning products, other than just using information of products alone in campaigns.

2) How much competitive advantage in the classification power for campaign outcomes by using features that are creator-specific and product-specific?

   The second objective used both 'product' and 'people' features of the campaigns to classify campaign success through machine learning models. Integrated with the knowledge learned from the results in first objective, Extreme Gradient Boosting (XGBoost) model was selected with the best performing classification among a series of other models. Followed by Particle Swarm

Optimization to perform feature selection, its classification power was further improved.

3) Will the addition of textual information with other variables improve classification performance?

By utilizing textual information with meta data from a campaign in a machine learning model, previous literatures have proven there are statistical advantages in model performance. The implemented solution used a representation model of textual information such as creator's biography, or blurbs, and combining with a set of auxiliary features from the campaign to enhance the classification power. With the implementation of NLP and DNNs using TensorFlow Keras architecture, four different DNNs were compared against the Naïve Bayes baseline model. Moreover, different combinations of information to feed in the DNNs were also examined.

1.4 Dissertation Outline

The rest of the dissertation is organized as follows: Chapter 2 describes data framework and identifies campaign success indicators. Chapter 3 discusses details on using different machine learning models and optimization techniques for creating the best performing model in crowdfunding classification. Chapter 4 explores the adoption of several natural language models and uses the textual information of the campaigns to extend the auxiliary features from the developed model in Chapter 3. Chapter 5 concludes result findings and limitations in this research, along with several future approaches to extend this study.

2. FINDING KEY INDICATORS OF CROWDFUNDING CAMPAIGNS FOR ALL LEVELS OF SUCCESS

2.1 Introduction

To understand crowdfunding as a financial instrument and its influence in the financial market, reward-based crowdfunding has produced a great public interest. This crowdfunding method attracts many research communities to explore further for a glimpse of the crowdfunding ecosystem and its social mechanisms. One of the most globally used reward-based crowdfund platforms is Kickstarter (www.kickstarter.com). Therefore, Kickstarter data will be used in this study to answer the research objectives for a better understand of crowdfunding.

Kickstarter was launched on April 28, 2009, and it built a strong project community since then [8]. It has over 19 million project backers and facilitated more than $5.6 billion funding, which has greater than 197,000 projects were successfully funded across all types of project categories [8] as in March, 2021. There are 16 project categories which include Art, Comics, Crafts, Dance, Design, Fashion, Film & Video, Food, Games, Journalism, Music, Photography, Publishing, Technology, Theater, and undefined. Kickstarter adopts the ‘all-or-nothing’ approach where creators will only get the funding if their project’s funding goal is reached. Similarly, project backers will only be charged for their pledge credits if each of their backed projects is successfully funded when the funding campaign ends. The payment scheme is that Kickstarter charges a flat

5% fee on the total amount raised, only if projects are funded successfully where funding goals are reached; otherwise, no fees will be collected from creators. Moreover, there is a 3% processing fee and a flat $0.20 per pledge will be charged if pledges are more than $10; otherwise, a 5% processing fee and a flat $0.05 per pledge will be charged if pledges are less than $10. The aims of Kickstarter are to provide an affordable marketing avenue for creators while constructing a center stage for campaign publicity to reach more investors for promotion. This method is popularly used by many small businesses in comparing to other marketing channels. To fully elevate the use of Kickstarter platform or other reward-based crowdfunding platforms, several exploratory studies have been documented in literatures to examining the characteristics of crowdfunding platforms and their mechanisms.

2.2 Related Work

There are limited number of early studies done in assessing successful crowdfunding campaign before 2014, considering the subject itself was and still is a novel practice with possible challenges in data extraction. In one of the previous studies reported by Baldwin and Von Hippel [9] in 2009, conducted empirical research on the competitiveness between the model of single user individuals (or firms) and the model of open collaborative innovation for design cost, communication cost, production cost, and transaction cost. Their results suggested that with the progress of technology used, open collaborative innovation, would be a strong viable model for producer due to certain advantages in the transaction and design cost, if the communication cost was low [9]. Later on, in 2011, Hemer [7] investigated multiple crowdfunding platforms to analyze the interface design and the systematic description of the main characteristics in crowdfunding projects. However, due to the lack of sufficient data, the findings were not clear but speculated that there was a high concentration focus on projects in the music and performing arts sectors in crowdfunding [7].

With the shifting paradigm in funding businesses using internet sites and mobile applications as technology evolved, more attraction and closer studies began to emphasize campaign features of crowdfunding in 2013. Mollick [10] conducted a study on Kickstarter data where he analyzed different key features in projects, and measured the quality of campaign through logistic regression. His results suggested that crowd founders who made frequent updates, had fewer spelling or grammatical errors, more videos postings, and larger social network would have positive influence on investors for a successful campaign [10]. Crosetto and Regner [11] also expanded on some other previous reward-based crowdfunding research by using Startnext sample, a German reward-based crowdfunding platform, to include more key features in their regression analyses, such as word count, blog entries, image count, and video count. They found evidence that blog entries, images, and videos were also highly correlated to campaign success; the same suggestion also goes to higher presence of other pre-selling rewards. They also found that large goal amounts and long durations of campaigns were negative indicators in project success [11]. Another study from Frydrych et al. [12] addressed the importance of organizational legitimacy and resources assembly processed on funding successful rate, where projects created from teams or pairs had a higher chance to succeed than projects from individuals. They also suggested that "a longer fund-raising period might expose an uncertain narrative for the project, resulting in decreasing support for the project" [12].

Similarly, Fernandez-Blanco et al. [13] provided a crowdfunding exploration using Kickstarter data as well. Instead, they extracted 13 key features to conduct k-means clustering for six main groups and found significant differences in success rate based on non-parametric Kruskal-Wallis test [13]. They also concluded pledges, comments, updates, and backers were key indicators in project success, particularly in music and arts categories [13]. A more recent study in 2017 from Zhao et al. [14], the authors used Indiegogo data where this reward-based platform allows either a "fixed goal" or "flexible goal" for creators to choose. They tracked across the crowdfunding timeline and created a two-layer regression model to predict funding amount and its perks for a given day [14].

For other types of crowdfunding, such as equity-based projects, there are number of existing reports as well. For example, Ralcheva and Roosenboom [15] examined the data of the UK equity-based crowdfunding platforms, Crowdcube and Seedrs, where their findings offered evidence that age of companies could negatively associate in project success.

A series of papers and research have analyzed the influence of factors and campaigns outcomes over time. Despite that previous research can only be served as an initial step towards a more profound understanding of the campaign features and the impact on funding success. Many questions remained unanswered. There are other critical features of creators and team formation that may potentially affect investors' interest for campaign success that are not fully explored. Moreover, results show that images and videos postings correlate higher funding success, are consistent among reports and demonstrate the key elements for advertising. However, a more specific question is whether images and videos postings of products yield a higher funding rate, or images and videos postings relate to creators play an integral part of funding success as well. These key questions and notions are not addressed in the literature. To fill these gaps, the first research objective is identifying other key indicators and their associated strength in crowdfunding campaign success.

2.3 Data

2.3.1 Data Source

Kickstarter data is the main concentration of this research objective, and it is also employed to answer other research objectives (discussed in Section 1.3) for the rest of this dissertation. The data source is from the Web Robots [16] site, where a regular web crawler is used to collect the meta data of the Kickstarter projects into files for public downloads. The data for this research was downloaded in May of 2020. This raw set of data consists of 204,625 rows with 38 columns. Prior to the scope of data sample, a yearly rate of successfully funded Kickstarter campaigns across all countries is illustrated in Figure 2.1. Year of 2019 seems to have the latest projects which comprises of 36,044 projects, and the percentage of successfully funded campaigns is quite encouraging with an uptick to 78.13% since the drop in 2014. Part of the reason for this increase is the rise of popular use in crowdfunding for raising capital at the Kickstarter platform with a well-established community of creative projects since 2009.

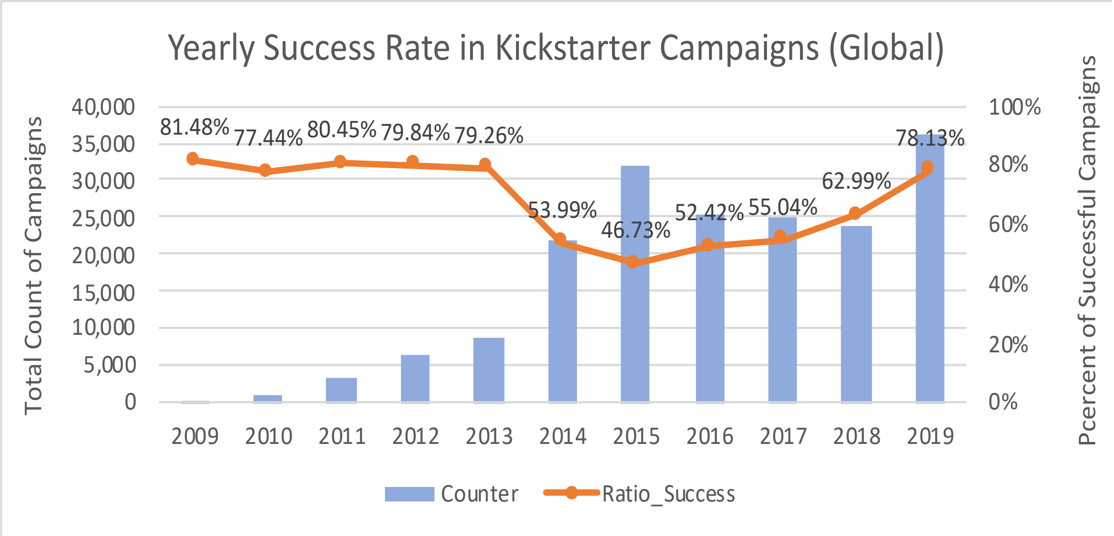


Figure 2.1: The percentage of all successful campaigns since 2009 for Kickstarter. Each column shows the total count of Kickstarter campaigns (excludes all the suspended campaigns).

To explore 2019 further, the geographical distribution of creators from different countries that used Kickstarter to launch their projects and the relative success rate is presented in Table 2.1. Hong Kong had the highest ratio of successfully funded projects and Austria had the least in 2019. However, U.S. had the most submitted projects and way exceeded the other listed countries with 21,499 projects where 78.96% projects reached the funded goal. Therefore, U.S. is a prominent figure in crowdfunding with much representation to display some of the influential factors for attractive campaigns.

Table 2.1: Sorted list of countries and frequency counts of Kickstarter projects that were launched in 2019 (excludes all the suspended campaigns). Sorted by the highest percentage of successfully funded projects.

COUNTRY	TOTAL COUNT	SUCCESSFULLY FUNDED PROJECTS (%)
HONG KONG	668	88.77%
SINGAPORE	298	82.89%
UK	5026	81.99%
CANADA	1890	80.85%
LUXEMBOURG	15	80.00%
AUSTRALIA	1013	79.27%
US	21499	78.96%
NEW ZEALAND	179	78.77%
FRANCE	741	76.52%
JAPAN	253	76.28%
DENMARK	197	74.62%
BELGIUM	152	73.68%
GERMANY	908	73.68%
NORWAY	86	73.26%
SWEDEN	347	72.91%
NETHERLANDS	305	72.46%
SWITZERLAND	185	66.49%
SPAIN	608	65.95%
IRELAND	137	65.69%
ITALY	584	59.42%
MEXICO	843	58.36%
AUSTRIA	110	57.27%

2.3.2 Sample Design and Data Collection

Since there is a sufficient crowdfunding market in the U.S., the scope of this research concentrated on all the U.S. campaigns launched in 2019. Other Kickstarter campaigns from other countries were not considered to avoid extra translation or interpretation requirement. In addition, to avoid premature campaigns or non-serious capital seekers, campaigns with less than a $1000 raised goal or campaigns with more than $500,000 were excluded. Duplicated campaigns (with same IDs) or suspended campaigns were also eliminated in the sample. Last, campaigns with any invalid nor inactive URL site were also removed.

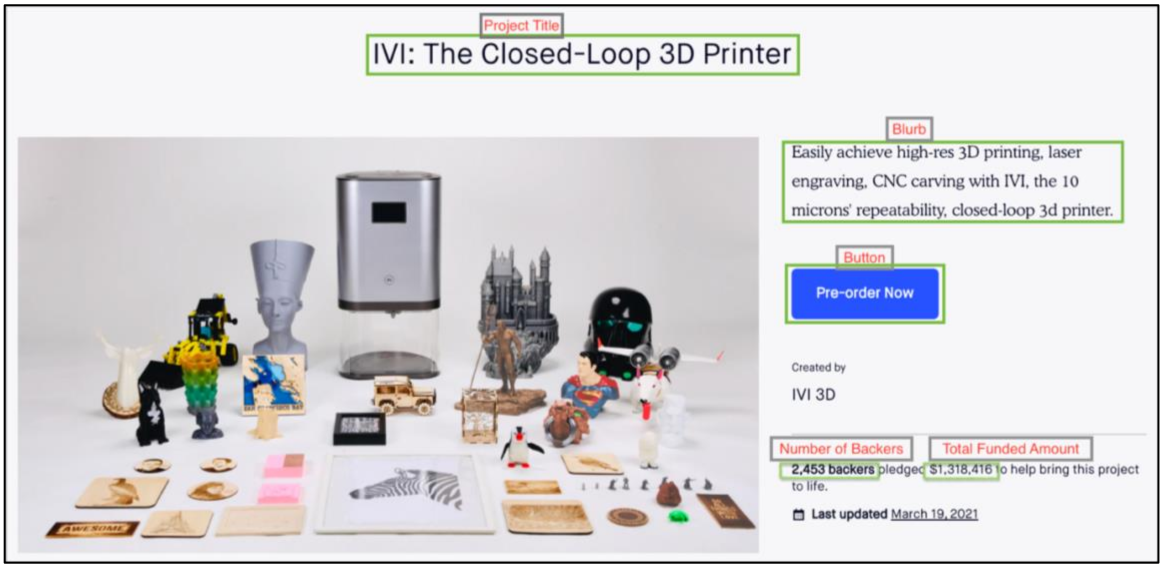
The downloaded Kickstarter data set has 38 columns as mentioned, and the data contains a mix of integer, floating-point, character, string, Boolean, and null fields. Some columns have XML unstructured data or html encoded strings. The majority of columns are not usable and only a handful of columns were applicable to a limited extent. To build useful data set, I first parsed out the URLs, project titles and their project ids from the downloaded data to catalog a list of campaigns. With the 11,410 campaigns in the final sample, the next step was to collect relevant and valuable data of each campaign through additional web scraping and manual data collection to construct my own customized dataset and features.

To automate the data extraction process by looping through a large catalog of URLs, I developed a webscraping script in R to locate all pertinent elements of the campaigns. Multiple R packages were used to parse and construct for different data types. Several notable packages used were, including, but not limited to, RSelenium, rvest, and xml2. These packages are crucial for interacting with Document Object Model (DOM) elements in browsers that contain JavaScript or jQuery rendered objects, reading in html or xml documents, and parsing specific elements either using CSS or XPath selectors from source codes. To run RSelenium, I set up a Docker container environment to deploy my developed program on top of the local operating system. A general outline for using docker and RSelenium for interacting with DOM elements can be found in Appendix A.4.

Majority of the desirable features from campaigns were parsed out through this automation method. However, some features of the campaigns were not possible to extract through automation and required human judgement to categorize properly during the data extraction process for data mining. For instance, when identifying any description of timeline or future planning from the project creators, it required human judgement to ingest information from the campaign in order to collect the data. Another challenge during data mining was to classify videos or images into the appropriate groups. To identify pictures that were product relevant or creator relevant properly, a human judgement was also involved during the classification process.

2.3.3 Key Variables and Data Preprocessing

The general structure of a Kickstarter campaign has three parts: the project spotlight page (top), a dynamic banner (middle), and the main content (rest of the page). In addition, for creator's background information, through clicking the creator's embedded link in Figure 2.2, more information would be available. An example of Kickstarter campaign's structure is illustrated in Figure 2.2 through Figure 2.4, where the highlighted elements in green were extracted either through webscraping automation or manually collected as described in Section 2.3.2. The overall objective was to explore what indicators have influence on crowdfunding success, and to show entrepreneurs how to strategize their campaigns for launching. Therefore, only variables that were displayed and available when first launched were relevant and employed as input data. On the other hand, some of the dynamic features that were updated during the funding period would not be relevant and were not part of the data collection, such as the live counts of FAQ, updates, comments, and total backers.



IVI: The Closed-Loop 3D Printer

Easily achieve high-res 3D printing, laser engraving, CNC carving with IVI, the 10 microns' repeatability, closed-loop 3d printer.

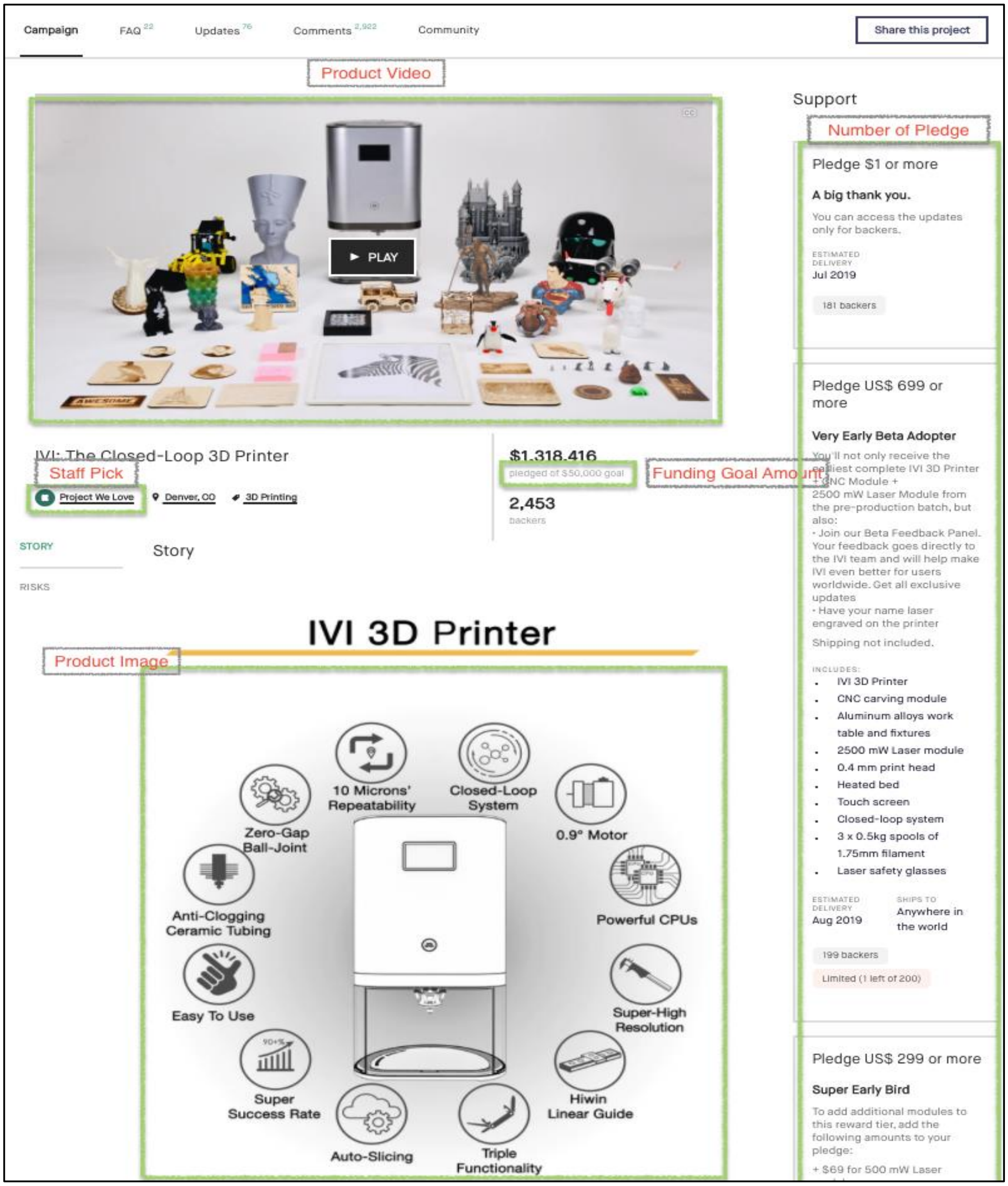
Created by

IVI 3D

2,453 backers pledged $1,318,416 to help bring this project to life.

Last updated March 19, 2021

Figure 2.2: An example of the project spotlight page locates at the top of Kickstarter campaign. All highlights in green were extracted into data features as tagged in red.



IVI: The Closed-Loop 3D Printer

$1,318,416
pledged of $50,000 goal

2,453
backers

Support

Pledge $1 or more

A big thank you.

You can access the updates only for backers.

ESTIMATED DELIVERY
Jul 2019

181 backers

Pledge US$ 699 or more

Very Early Beta Adopter

You'll not only receive the earliest complete IVI 3D Printer + CNC Module +
2500 mW Laser Module from the pre-production batch, but also:
• Join our Beta Feedback Panel. Your feedback goes directly to the IVI team and will help make IVI even better for users worldwide. Get all exclusive updates
• Have your name laser engraved on the printer

Shipping not included.

INCLUDES:
- IVI 3D Printer
- CNC carving module
- Aluminum alloys work table and fixtures
- 2500 mW Laser module
- 0.4 mm print head
- Heated bed
- Touch screen
- Closed-loop system
- 3 x 0.5kg spools of 1.75mm filament
- Laser safety glasses

ESTIMATED DELIVERY
Aug 2019

SHIPS TO
Anywhere in the world

199 backers

Limited (1 left of 200)

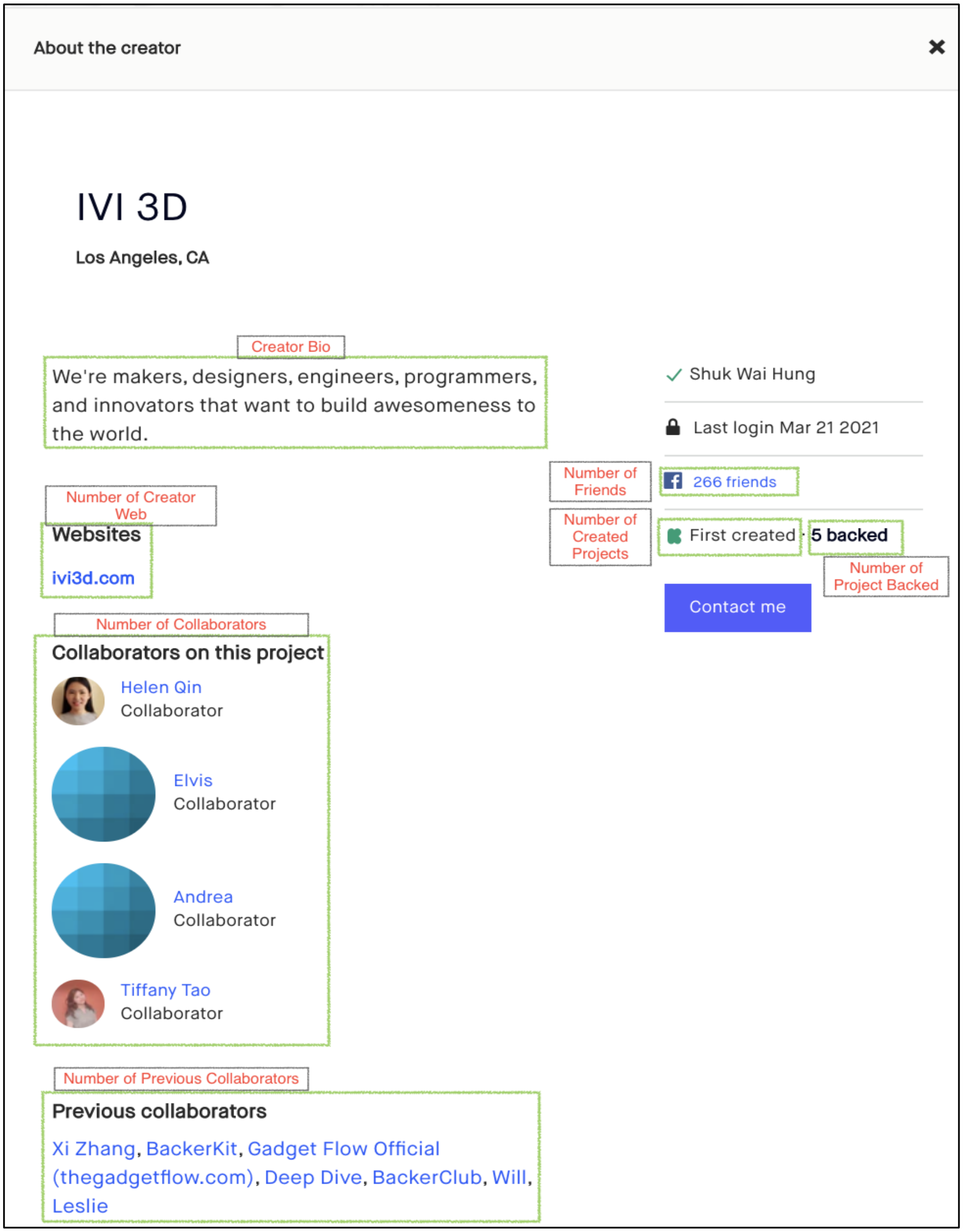
Pledge US$ 299 or more

Super Early Bird

To add additional modules to this reward tier, add the following amounts to your pledge:

+ $69 for 500 mW Laser

Figure 2.3: A continued example of the dynamic banner with the live counts of FAQ, Updates, and Comments, and the main content of Kickstarter campaign. All highlights in green were extracted into data features as tagged in red.



About the creator ✖

IVI 3D

Los Angeles, CA

We're makers, designers, engineers, programmers, and innovators that want to build awesomeness to the world.

Websites

ivi3d.com

Collaborators on this project

Helen Qin
Collaborator

Elvis
Collaborator

Andrea
Collaborator

Tiffany Tao
Collaborator

Previous collaborators

Xi Zhang, BackerKit, Gadget Flow Official (thegadgetflow.com), Deep Dive, BackerClub, Will, Leslie

✓ Shuk Wai Hung

Last login Mar 21 2021

266 friends

First created · 5 backed

Figure 2.4: An example of project creator's background when clicked on creator's name in the Kickstarter campaign. All highlights in green were extracted into data features as tagged in red.

The finalized data set after parsing on 11,410 URLs comprises 40 variables. A data dictionary in Appendix A.1 details a full list of variables with definitions. Data processing and engineering techniques were applied to number of variables for the purpose of computational numeration as shown below:

- Converted string variables to word counts:

Table 2.2: List of string variables and their corresponding variables for word counts.

Variable	**Data Type**		**Engineered Variable**	**Data Type**
Bios	String	⬄	Bios_wdct	Integer
Blurb	String	⬄	Blurb_wdct	Integer
Name	String	⬄	Name_wdct	Integer

- Converted creator associated variables to Boolean values:

Table 2.3: List of variables and their corresponding variables for binary representation.

Variable	**Data Type**		**Engineered Variable**	**Data Type**
Badge	String	⬄	Is_backerfav	Boolean
Year_Exp	Float	⬄	Is_exp	Boolean
Num_Members	Integer	⬄	Is_members	Boolean
Degree	String	⬄	Is_degree	Boolean

In general, all Boolean variables are encoded into a binary representation as defined in Table A.1. If Boolean value is True, then the binary encoding value equals to 1 where it indicates information was presented in the campaign; if Boolean value is False, then the binary encoding value equals to 0 where it indicates information was not presented in the campaign.

- Success Rate:

This variable is computed with the ratio of the total raised amount to the funding goal. Using this ratio can standardize the fund-raising performance of each project by comparing how much the total raised fund is under or above the expectation as one of the metrics to evaluate funding success.

- Final Class:

The original class variable, Orig_Class, from the downloaded data classified projects into a binary class, either a campaign was 'failed' or 'successful' in raising funds. In this research, my aim is to look beyond the two separate classes and to investigate the exceptionally successful campaigns. Campaigns are divided into three classes: 'failed', 'successful', and 'super successful'. The class was assigned as a result of the computed ratio of total amount raised to the funding goal, Success Rate variable. Projects were assigned to 'failed' if the Success Rate value is less than 1. Similarly, projects are assigned to 'successful' if the Success Rate ratio value is greater than or equal to 1 but less than 1.5. Furthermore, the rest of the projects are assigned to 'super successful' if the Success Rate ratio value is greater than or equal to 1.5, since these projects represents a superior fund-raising performance by exceedingly more than twice the funding goal.

2.4 Methods

To investigate which influential indicators pose any impact on crowdfunding success, two regression models were applied, hierarchical multiple regression and hierarchical ordinal logistic regression. Both exploratory hierarchical models were chosen to acquire understanding of how one set of variables affect and associate with the variability of the dependent variable. The motivation of using hierarchical regression is to learn the complex relationship among all the combinations of variables and their strength of association, though not so much on establishing the best model to predict in this case. Another advantage of using this framework is that it will provide statistical evidence on the potential classification power of the independent variables as indicators. The regression models are implemented using a Python Statsmodels module for statistical inferential metrics.

2.4.1 Hierarchical Multiple Regression

For hierarchical multiple regression, instead of introducing all independent variables into the model simultaneously, it requires two-stage process in order to evaluate the additive significance of the independent variables. First, a set of control variables was defined and used in the first stage to form a restricted model. Subsequently, the rest of the independent variables were added to form the full model in the second stage for evaluation on the dependent variable. Here is the outline of the implemented first and second stage of the hierarchical multiple regression model:

$$F_{restricted}(Success\ Rate) = \beta_0 + \beta_1 Goal + \beta_2 Duration + \beta_3 Number\ of\ Pledge\ Options + \beta_4 Staff\_Pick + \beta_5 Cat\_Type + \epsilon \tag{2.1}$$

Where $F_{restricted}(Success\ Rate)$ is a transformed dependent variable, β_0 is a constant, and ϵ is an error term.

$$F_{full}(Success\ Rate) = \beta_0 + \beta_1 Goal + \beta_2 Duration + \beta_3 Number\ of\ Pledge\ Options + \beta_4 Staff\_Pick + \beta_5 Cat\_Type + \beta_6\ Number\ of\ Creator\ Web + \beta_7 Number\ of\ Created\ Proj + \beta_8 Num\_Collabs + \beta_9 Friends + \beta_{10} Button\_Flag + \beta_{11} Video\_Creators + \beta_{12} Img\_Creators + \beta_{13} Video\_Prod + \beta_{14} Img\_Prod + \beta_{15} Plan\_Timelne + \beta_{16} Is\_backerfav + \beta_{17} Is\_exp + \beta_{18}\ Is\_degree + \beta_{19} Is\_member + \beta_{20} Bios\_wdct + \beta_{21} Blurb\_wdct + \epsilon \tag{2.2}$$

Where $F_{restricted}(Success\ Rate)$ is a transformed dependent variable, β_0 is a constant, and ϵ is an error term.

2.4.2 Transformation for Hierarchical Multiple Regression

To evaluate success rate (the ratio of total raised amount to the funding goal) as a continuous dependent variable, four assumptions of the multiple regression were tested [17]: 1) Error terms are normally distributed, 2) Linearity exists between independent variables and dependent variables, 3) Homoskedasticity for a constant variance in error terms, 4) Independence exists and no correlation shows in error terms.

Before any four assumptions were tested, Figure 2.5 was plotted and it shows there is a violation of linearity in success rate. Additionally, the distribution of success_rate is also highly right-skewed with a long tail, as described in the density plot of Figure 2.5 where the mean of success_rate is larger than its median. Hence, different techniques of transformation were explored to improve linearity and reduce skewness for a ‘more’ normal distribution. A skewness is computed with the Fisher-Pearson coefficient of skewness using scipy.stats.skew module.

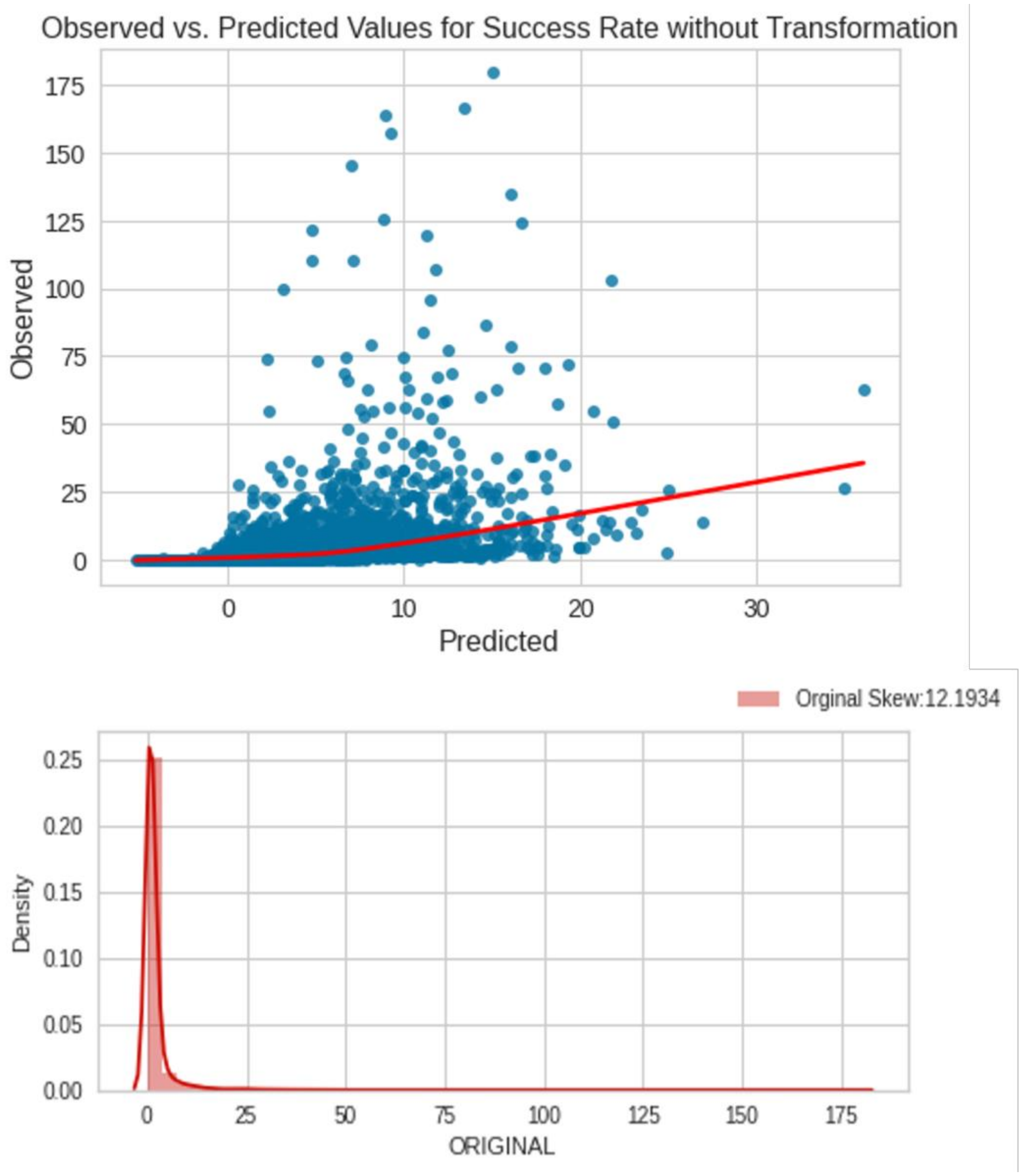


Figure 2.5: The scatterplot of observed success rate and predicted* success rate before transformation (top), and the density plot of success rate (bottom).

*Term “Predicted” is used as a generalization of all projects in regression model and terminology, not for a specific project.

These are the transformation attempted on the regression model, $y = \beta_0 + \beta_i X + \epsilon$.

- Logarithm:

$$\text{Log Linear: } \text{Log}(y') = \beta_0' + \beta_i' X + \epsilon' \tag{2.3}$$

$$(\text{Log} + 1) \text{ Linear: } \text{Log}(y' + 1) = \beta_0' + \beta_i' X + \epsilon' \tag{2.4}$$

$$\text{Log} - \text{Log: } \text{Log}(y') = \beta_0' + \beta_i' \text{ Log}(X) + \epsilon' \tag{2.5}$$

- Inverse square root:

$$1/\sqrt{}(y') = \beta_0' + \beta_i' X + \epsilon' \tag{2.6}$$

- Box-cox power function:

$$y'(\lambda) = \begin{cases} \dfrac{y^{\lambda} - 1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log y, & \text{if } \lambda = 0 \end{cases} \tag{2.7}$$

Where λ is the optimal value of power between -5 to 5, and y>0.

- Yeo-Johnson power function:

$$y'(\lambda) = \begin{cases} \dfrac{(y_i + 1)^{\lambda} - 1}{\lambda}, & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y_i + 1), & \text{if } \lambda = 0, y \geq 0 \\ \dfrac{-[(-y_i + 1)^{(2-\lambda)} - 1]}{(2 - \lambda)}, & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y_i + 1), & \text{if } \lambda = 2, y < 0 \end{cases} \tag{2.8}$$

Where λ is the optimal value of power between -5 to 5, and y>0.

By visualizing the transformed distribution of success rate in Figure 2.6, log transformation yields a better suited result since it improved skewness from 12.19 to -1.46 with a unimodal distribution along with a fairly centered spread. However, the distribution of success rate indicates the possible existence of underlying bimodality, which it doesn't come as a surprise since the nature of the project success is measured in binary classification. Chapter 3 will further discuss on using nonlinear or machine learning models to better fit the data for classification. As far as finding key indicators in success rate, regression is still an adequate approach to shed light on the relationship among variables.

After the chosen log transformation applied to the dependent variable, the restricted and the full model's R-squared were used as the metric for goodness-of-fit, and the Analysis of Variance (ANOVA) test was followed to compare the first and the second order models for the significance of having additional terms with F-statistics. The hypotheses of the F-test are defined:

H_0 = The fit of restrictive model is sufficient to explain the variability of the dependent variable.

H_1 = The fit of full model is significantly better than the restrictive model to explain the variability of the dependent variable.

And the

$$F\ statistic = \frac{\frac{RSS_1 - RSS2}{k_2 - k_1}}{\frac{RSS_2}{n - k_2}} \tag{2.9}$$

Where RSS_1 is the sum of squares of residual errors for model 1 and RSS_2 is the sum of square of residual errors for model 2. K_1 is the number of parameter terms in model 1 and K_2 is the number of parameter terms in model 2, and n is the number of observations.

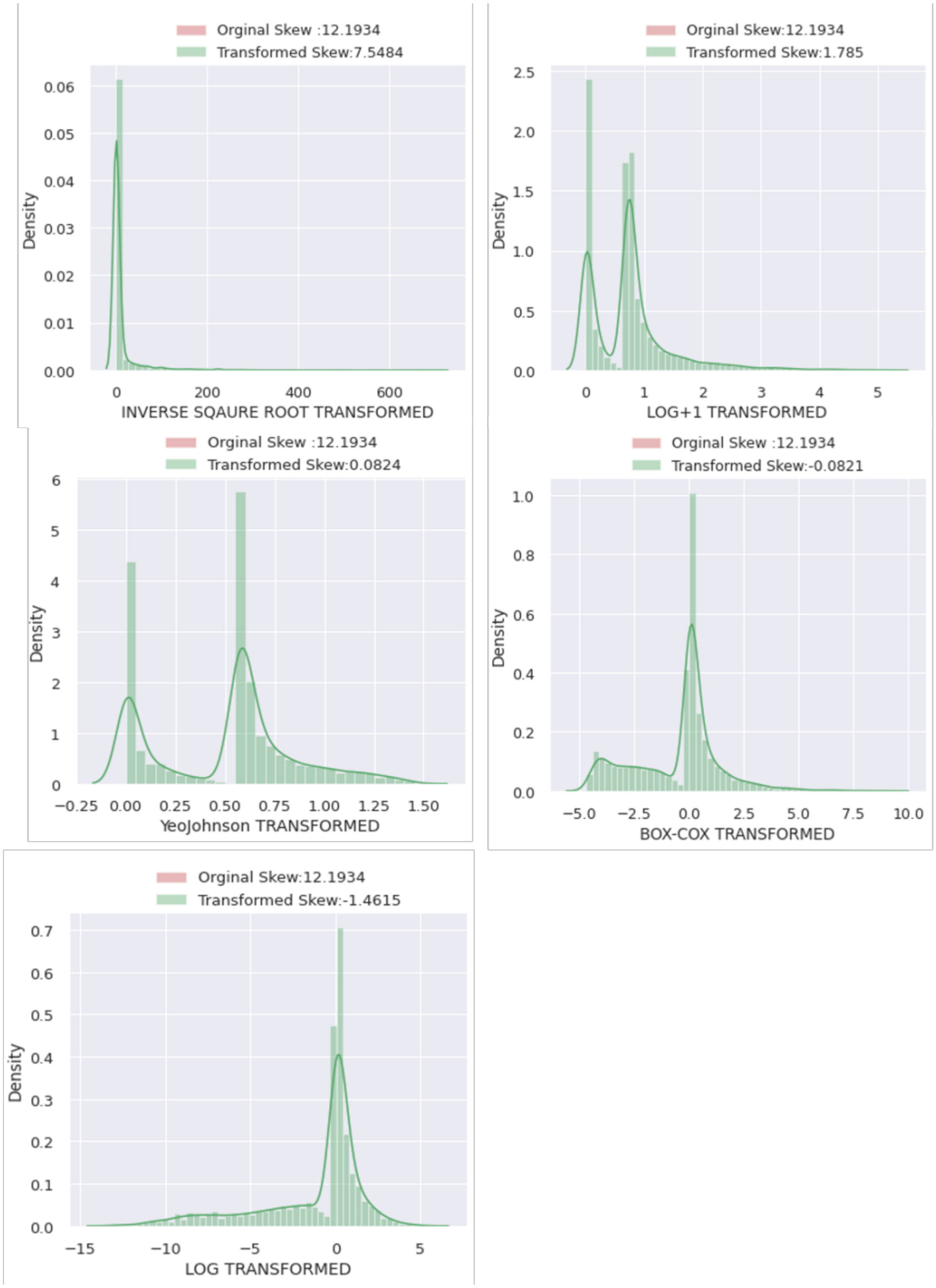


Figure 2.6: The density plots of transformed success rate, sorted by skew value in descending order.

2.4.3 Hierarchical Ordinal Logistic Regression

For hierarchical ordinal logistic regression, similar to the described hierarchical multiple regression model, a two-stage process was applied. Control variables and independent variables were presented in the outlined first and second stage of the hierarchical logistic regression model:

$$F_{restricted}(Final_{Class} = i) = \beta_0 + \beta_1 Goal + \beta_2 Duration + \beta_3 Number\ of\ Pledge\ Options + \beta_4 Staff\_Pick + \beta_5 Cat\_Type + \epsilon \qquad (2.10)$$

where $\mathrm{F(Final\_Class)} = \ln\left(\frac{\mathrm{P(Final\_Class=i)}}{1-\mathrm{P(Final\_Class=i)}}\right)$ is the log of odds for $\mathrm{Final_{Class}} = \mathrm{i}$ or the probability of a final class is converted as $\mathrm{P(Final\_Class} = \mathrm{i)} = \frac{e^{(\beta_0+\beta' x)}}{\left(1+e^{(\beta_0+\beta' x)}\right)}$ and i = 1, 2, or 3 as the ordinal classification for project success. β_0 is a constant, and ϵ is an error term.

$$F_{Full}(Final\_Class = i) = \beta_0 + \beta_1 Goal + \beta_2 Duration + \beta_3 Number\ of\ Pledge\ Options + \beta_4 Staff\_Pick + \beta_5 Cat\_Type + \beta_6\ Number\ of\ Creator\ Web + \beta_7 Number\ of\ Created\ Proj + \beta_8 Num\_Collabs + \beta_9 Friends + \beta_{10} Button\_Flag + \beta_{11} Video\_Creators + \beta_{12} Img\_Creators + \beta_{13} Video\_Prod + \beta_{14} Img\_Prod + \beta_{15} Plan\_Timelne + \beta_{16} Is\_backerfav + \beta_{17} Is\_exp + \beta_{18}\ Is\_degree + \beta_{19} Is\_member + \beta_{20} Bios\_wdct + \beta_{21} Blurb\_wdct + \epsilon \qquad (2.11)$$

Where $\mathrm{F(Final\_Class)} = \ln\left(\frac{\mathrm{P(Final\_Class=i)}}{1-\mathrm{P(Final\_Class=i)}}\right)$ is the log of odds for $\mathrm{Final_{Class}} = \mathrm{i}$ or the probability of a final class is computed as $\mathrm{P(Final\_Class} = \mathrm{i)} = \frac{e^{(\beta_0+\beta' x)}}{\left(1+e^{(\beta_0+\beta' x)}\right)}$ and i = 1, 2, or 3 as the ordinal classification for project success. β_0 is a constant, and ϵ is an error term.

The dependent variable is a discrete categorical variable to represent class assignment of failed, successful, or super successful projects. Since the class assignment is encoded in an ordinal sequence in accordance with the success rate, ordinal logistic regression was employed as the most suitable logistic model. Further, to test the goodness-of-fit, Pseudo R squared value of both the restricted and full logistic models from each stage were compared. Followed with the likelihood ratio test to test the significance of having additional terms in the full model, similar to the ANOVA test used in multiple regression. The hypotheses of the likelihood ratio test are same as the hypotheses as the ANOVA test. The likelihood ratio (LR) test statistic is:

$$\text{LR statistic} = -2\ln\left(\frac{\text{L(Model 1)}}{\text{L(Model 2)}}\right) = 2(\text{loglik(Model2)} - \text{loglik(Model 1)}) \quad (2.12)$$

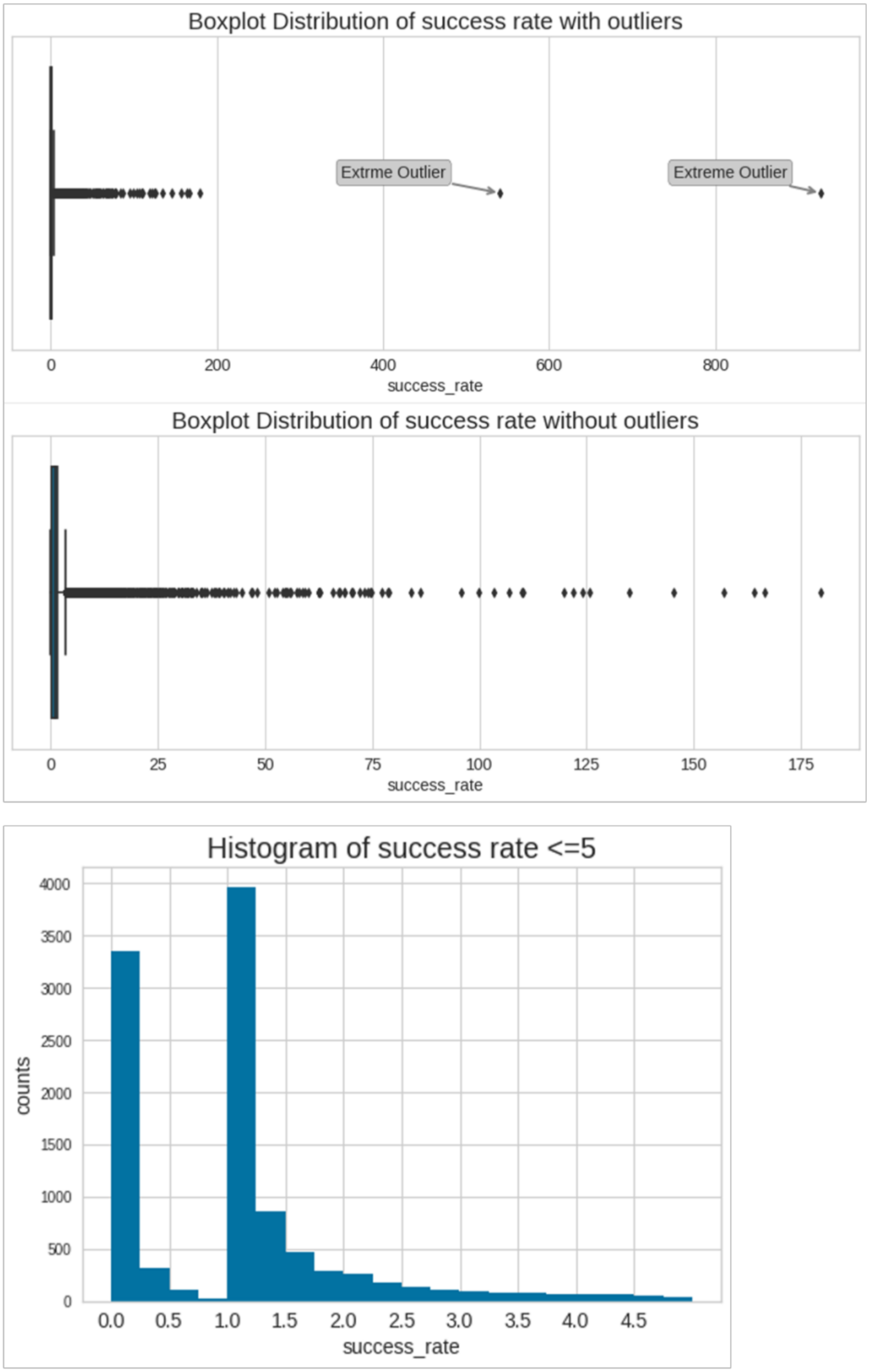
where L(Model*) is the likelihood for model 1 and model 2, loglik(Model*) is the natural log of the likelihood of the respective model. The test statistic follows the chi-squared distribution with the degrees of freedom is the number of constraints or free parameters.

2.5 Results

2.5.1 Descriptive Statistics

The final sample comprised of 11,410 projects with funding goal between $1,000 and $500,000 while excluding all suspended projects. An exploratory data analysis was conducted to gain insights on data distribution and outliers' detection. Although it is not always necessary to remove outliers but two extreme outliers with success rate (the ratio of the total raised amount to the funding goal) of 540.54 and 926.57 in Figure 2.7 were removed to better represent the population since success rate is one of the primary dependent variables in modeling. A statistical summary of main variables used is followed in Table 2.4. The average success_rate is 2.17 while 75% of projects have success_rate with less than 1.46. The average funding goal is $17,211 with half of projects less than $6,500; the average pledged amount is $27,139 with half of projects less than $4,524. Only 25% of projects have campaigns marketed for more than 40 days and the average duration is 35 days; there are 9 pledge options for investors to choose from, and creators have 445 Facebook friends in network on average. About 50% of projects have at least 1 listed relevant website, and more than 50% of projects are created by creators who have created at least a prior project. In addition, more than 50% of projects are created by creators who have backed a project previously. On average, at least one video and one picture of either creators or products is shared on campaign websites. The average number of words for blurb (or project's pitch) and creator bios are 17 and 71 respectively. Projects include future planning or product timeline 50% of the time; only 25% and 15% of projects will mention creators' education background and relevant experience respectively. In addition, 11% of projects are created by someone who earned a Kickstarter badge of backer's favorite status; 76% of projects include some background information on team formation or memberships. For geographic distribution, Figure 2.8 presents the 2019 demographic distribution of campaigns in the U.S where California had the most launched Kickstarter campaigns with 2,304 projects, and New

York came second with 1,304 projects. South Dakota had the least projects launched with only 14 campaigns.


Boxplot Distribution of success rate with outliers

Boxplot Distribution of success rate without outliers

Histogram of success rate <=5

Figure 2.7: The distribution of the 2019 Kickstarter campaigns success rate before (top) and after (middle) two extreme outliers were removed. A histogram (bottom) of success rate with raised funds less than 5 times of pledged goal.

Table 2.4: Summary statistics of the main variables used.

variables	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
Success_Rate	2.17	7.09	0.00	0.10	1.06	1.46	179.50
Usd_Pledged ($)	27139	184298	1	862	4524	13417	12143436
Goal ($)	17211	39052	1000	3000	6500	15000	500000
Duration (in days)	35	13	1	30	30	40	98
Number Of Pledge Options	9	7	0	5	8	11	101
Number Of Creator Web	2	2	0	0	1	2	20
Number Of Created Proj	3	6	0	1	1	2	59
Number Of Project Backed	15	46	0	0	1	7	984
Friends	445	1018	0	0	0	349	5000
Num_Collabs	1	2	0	0	0	1	23
Number Of Precollabs	1	1	0	0	0	0	1
Video_Creators	1	1	0	0	1	1	13
Img_Creators	2	3	0	0	0	1	48
Video_Prod	1	2	0	0	0	1	21
Img_Prod	12	16	0	1	6	16	142
Blurb_Wdct	17	7	1	12	17	21	31
Bios_Wdct	71	86	0	22	45	85	1557
Staff_Pick*	0.15	0.35					
Plan_Timeline*	0.50	0.50					
Is_Backerfav*	0.11	0.32					
Is_Exp*	0.25	0.43					
Is_Degree*	0.15	0.35					
Is_Member*	0.76	0.43					

*encoded in binary representation

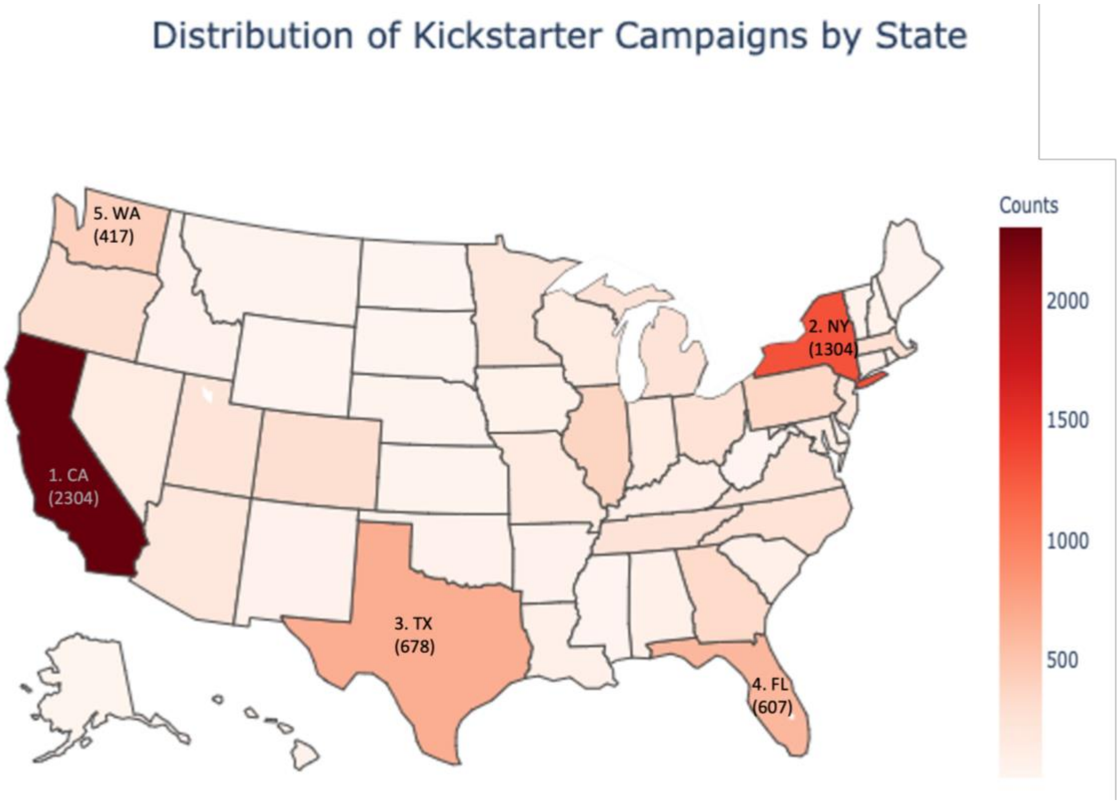


Figure 2.8: The demographic distribution of the 2019 Kickstarter campaigns in the U.S.

Another crucial dependent variable is the project success classification. As mentioned, this research has extended beyond the binary outcomes but divided project success into three main classes: failed, successful, and super successful. Figure 2.9 shows the overview of all three classes. Furthermore, the distribution of project types by project success was also explored as presented in Figure 2.10, where Technology category has the greatest number of projects and dance category has the least number of projects. All categories contain at least one or more super successful projects, including dance category with 3 super successful projects.

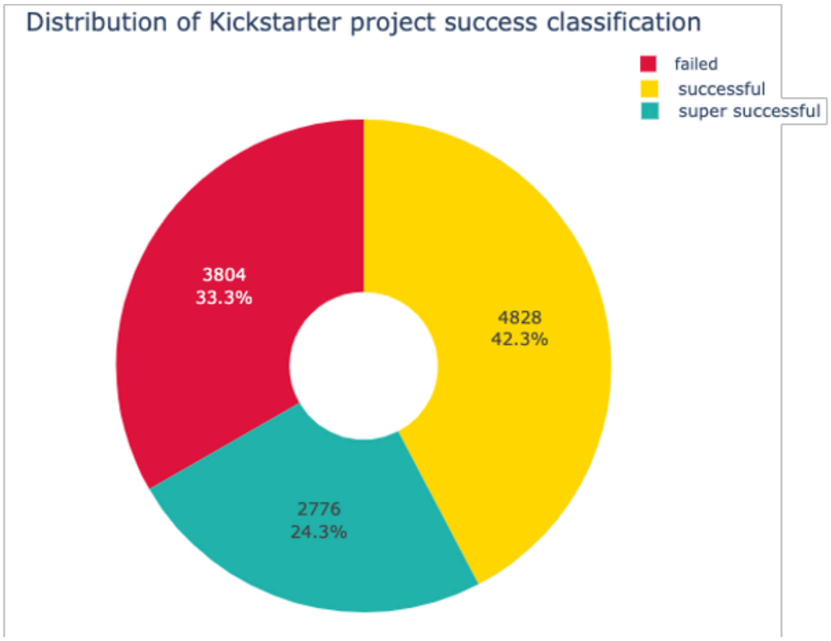


Figure 2.9: The distribution of Kickstarter project success classification in 2019.

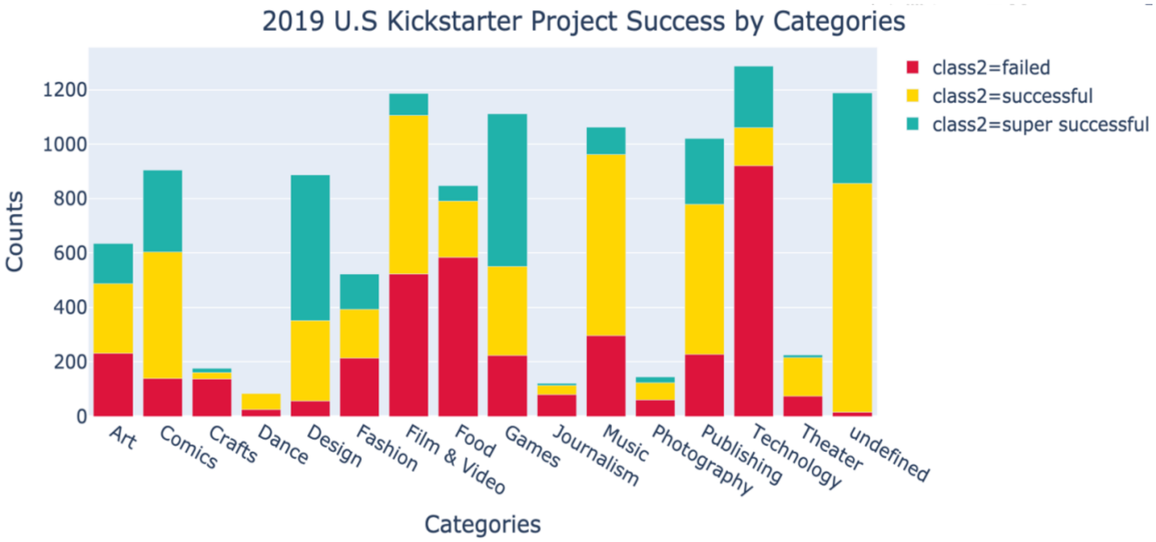


Figure 2.10: The distribution of project success by categories for the 2019 U.S Kickstarter campaigns.

2.5.2 Results from the Hierarchical Multiple Regression

First order (restrictive) and second order (full) regression models for log transformed success rate were built in a two-stage process. Table 2.5 presents the summary of results. For the control variables (funding goal, funding duration, number of available pledge options to investors, and staff pick featured), all are significant in the first order model as well as in the second order, except the project category. One noteworthy observation is that both goal and duration have a potential curve-linear relationship where larger goal amount or the longer funding period will not necessary be associated with success rate positively, especially once they pass a certain threshold (local maxima) in the distribution.

For disclosing project plan or product timeline in campaign, it is highly significant for success rate with a positive association. Additionally, having an informative description for the project pitch or blurb is highly important towards success rate because a clear project objective is critical to attract investors. However, the result shows that there is a potential curve-linear relationship of increasing word counts used in blurbs; meaning lengthy description does not always equate a higher likelihood for project success.

In terms of features associated with creators, only four variables (number of listed websites, history of created projects, earning a badge status of backer's favorite, and mentioning of relevant experience) with p-values <0.05 to show significance on the success rate positively. On the other hand, neither having history of backing other projects nor mentioning of educational background have a significant effect on the success rate. Same trend is observed for a lengthy description in biography section that is also not significant towards the success rate. All team's formation features are highly significant with p-value=0.000, except the number of friends is only significant with p-value =0.012.

For features focusing on the media used, the multiple regression shows that both images and videos that are related to the creators or the products are highly significant to success rate with p-value=0.000. Among images and videos used that are associated with creators, using videos will have a stronger positive association with the success rate than images. Similarly, for products, posting videos will also have a slightly stronger positive association with the success rate as well, while it is significant, the image of products has a higher significance towards success rate in comparison.

The results suggest that the top three important indicators for launching successful project are: 1) being featured and selected by Kickstarter's staff, 2) creators earned the badge status of being backer's favorite, and 3) having a substantial number of collaborators on projects. These characteristics will yield a higher success rate.

Table 2.5: Summary of multiple regression

	Variables (N=11,408)	Coefficient (Restrictive)	Standard Error (Restrictive)	P>|t| (Restrictive)	Coefficient (Full)	Standard Error (Full)	P>|t| (Full)
Control	Constant	-0.7965	0.118	0.000	-1.30E+00	1.31E-01	0.000
	Goal	-2.43E-05	6.36E-07	0.000	-2.50E-05	5.73E-07	0.000
	Duration	-0.0542	0.002	0.000	-0.041	0.002	0.000
	Number Of Pledge Options	0.1417	0.004	0.000	0.0782	0.004	0.000
	Staff_Pick	1.3923	0.071	0.000	0.9935	0.064	0.000
	Cat_Type	4.8525	0.963	0.000	1.6227	0.873	0.063*
Features of Products	Plan_Timeline				0.7142	0.046	0.000
	blurb_wdct				-0.0277	0.004	0.000
Features of Creators	Number Of Creator Web				0.113	0.017	0.000
	Number Of Created Proj				0.0331	0.005	0.000
	Number Of Project Backed				0.0002	0.001	0.707*
	Is_Backerfav				0.9994	0.078	0.000
	Is_Exp				0.1903	0.052	0.000
	Is_Degree				-0.0603	0.063	0.336*
	Bios_Wdct				4.06E-05	0	0.880*
Features of team formation	Num_Collabs				0.3008	0.019	0.000
	Friends				5.70E-05	2.27E-05	0.012
	Is_Member				-0.33	0.052	0.000
Media Used	Video_Creators				0.2345	0.024	0.000
	Img_Creators				0.0605	0.01	0.000
	Video_Prod				0.0397	0.019	0.033
	Img_Prod				0.0357	0.002	0.000

*cannot conclude statistically significant at 5%. All others are statistically significant.

Overall, both first order ($F_{11402,5}$ =1013, p-value <α=0.05 with adjusted R-squared=0.307) and second order ($F_{11386,16}$ =439.2, p-value <α=0.05 with adjusted R-squared=0.447) models have significant F statistic values. By ANOVA test of comparison, the difference in the sum of squares of residual errors between the full

($SSR_{2nd\ order}$ = 62257.74) and the restrictive regression ($SSR_{1st\ order}$ = 78023.76) is significant (F statistic =180.21, p-value $<\alpha=0.05$). Therefore, null hypothesis is rejected that the full model with additional term parameters is significant, and it has a higher power than the restrictive model to explain the variance in the dependent variable.

Last, the four assumptions were checked to validate the results from the multiple regression as illustrated in Figure 2.11. First, to meet the linearity and additive assumption, log transformation was applied as detailed in Section 2.4.2 for linearizing the relationship between success rate and other independent variables. With log transformation shows in Figure 2.6, the linearity is significantly improved with reduced skewness to -1.4615 from the original distribution with 12.1934 of skewness. Second, to meet the independence of the errors assumption, Durbin-Watson statistic (DW=2.027) and the plot of residual autocorrelations indicate no violations. Third, for homoscedasticity of the errors assumption, the scale plot of residuals against predicted values indicates violation. In fact, it almost shows sign of two groups of errors where it was mentioned previously that there could be a potential bimodality nature in the success rate. Forth, to meet the normality of the errors assumption, the central limit theorem allows to assume the sampling mean will approach normal since the sample size is 11,408. A Q-Q plot indicates a reasonable normality of errors. Multicollinearity was also checked to ensure no highly correlated pairs to affect the dependent variable. Variance inflation factors (VIFs) for all parameter terms is between 1.013 and 1.715, Appendix A.2 details a full list of variance inflation factors for each parameter terms.

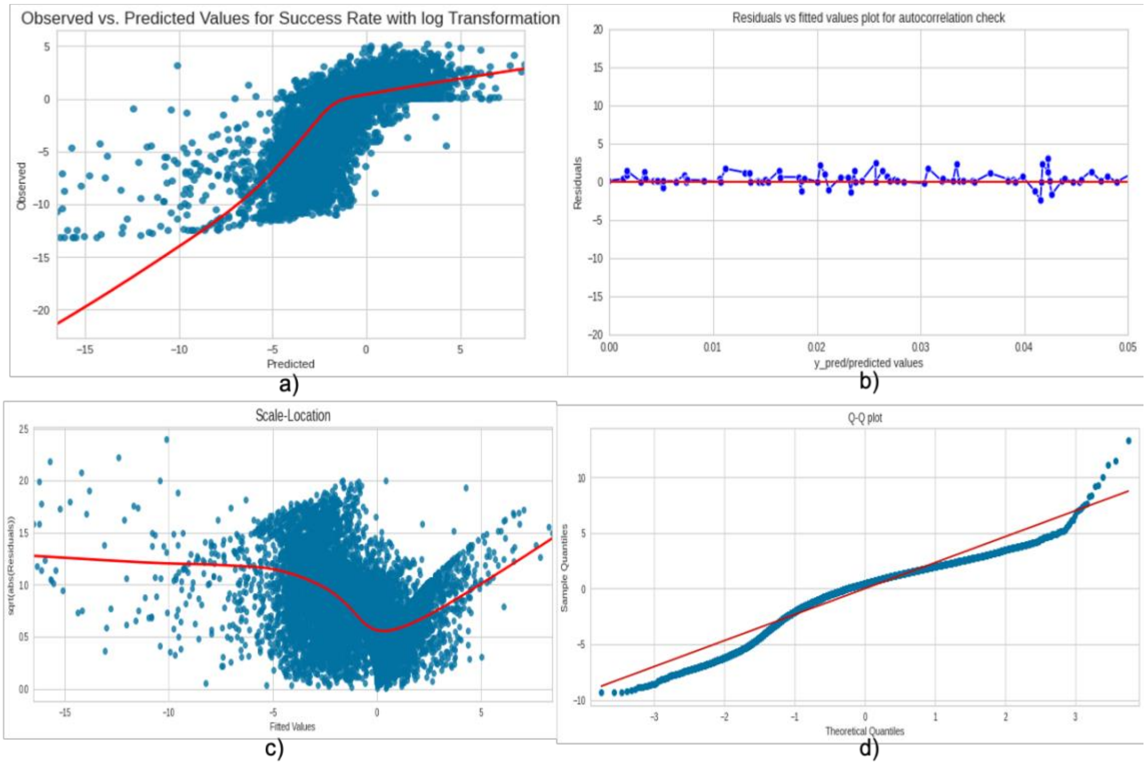


Figure 2.11: The diagnostic plots for checking the four assumptions of multiple regression on success rate after log transformation.

2.5.3 Results from the Hierarchical Ordinal Logistic Regression

A similar two-stage process of the multiple regression study took place for the first order (restrictive) and second order (full) logistic regression models. In here, the dependent variable is the probability of an event, where the event is a categorical ordered project success classification ranges from 1 to 3, the encoding represents project success level from failed to super successful respectively. Table 2.6 presents the summary of results. As shown, all the control variables (funding goal, funding duration, number of available pledge options to investors, staff pick featured, and project category) are significant in the first order and the second order model. This results also agree with the observation from the previous multiple regression that both goal and duration have a potential curve-linear relationship where larger goal amount or the longer funding period will not necessary be associated with success rate positively.

All features of products are significant in the second order model. Unlike the multiple regression, there is no indication of a potential curve-linear relationship of increasing word counts used in blurbs. Instead, it is positively associated with the project success level.

In terms of features associated with creators, only the history of backing other projects cannot reject the null hypothesis ($p=0.24>\alpha=0.05$) that it has no sign of significance on the project success level. All team's formation features are significant, same as the result in the multiple regression.

By comparing the focalization in media used that will correlate a higher likelihood of success level, the logistic regression results only partially agree with the ones in multiple regression. Both images and videos of related to creators are significant to success level, but not in projects. As learned and noted in the Section 2.5.2, between the images and videos used that associated with creators, using videos will have a stronger positive association with the success level than images. However, for media used in products, videos used is not significant to success level, only images used is significant.

In general, both first order (Log-Likelihood=-10774, p-value $<\alpha=0.05$ with pseudo adjusted R-squared=0.121) and second order (Log-Likelihood =-8876.2, p-value $<\alpha=0.05$ with pseudo adjusted R-squared=0.276) models have significant chi-squared statistics. By the likelihood ratio chi-square test of comparison, the likelihood ratio test statistic is significant (LR=3796.524, p>distributed chi-squared=0.0) with 16 degrees of freedom. Therefore, null hypothesis is rejected that the full model with additional term parameters is significant, and it has a higher power than the restrictive model to explain the variance in the dependent variable.

Table 2.6: Summary of ordinal logistic regression.

	Variables (N=11,408)	Coefficient (Restrictive)	Standard Error (Restrictive)	P>|t| (Restrictive)	Coefficient (Full)	Standard Error (Full)	P>|t| (Full)
Level	Failed/Successful	-0.9086	0.09	0.000	-0.379	0.119	0.001
	Successful/Super Successful	0.7905	0.013	0.000	1.0555	0.013	0.000
Control	Goal	-2.38E-05	1.02E-06	0.000	-3.70E-05	1.42E-06	0.000
	Duration	-0.0355	0.002	0.000	-0.0265	0.002	0.000
	Number Of Pledge Options	0.0927	0.003	0.000	0.04	0.004	0.000
	Staff_Pick	1.0007	0.052	0.000	0.9682	0.058	0.000
	Cat_Type	5.7672	0.725	0.000	5.0559	0.787	0.000
Features of Products	Plan_Timeline				0.3958	0.041	0.000
	blurb_wdct				1.6658	0.078	0.000
Features of Creators	Number Of Creator Web				0.0747	0.015	0.000
	Number Of Created Proj				0.0552	0.006	0.000
	Number Of Project Backed				0.0007	0.001	0.240*
	Is_Backerfav				1.6658	0.078	0.000
	Is_Exp				0.1922	0.047	0.000
	Is_Degree				-0.1883	0.056	0.001
	Bios_Wdct				6.00E-04	0	0.011
Features of team formation	Num_Collabs				0.3986	0.022	0.000
	Friends				9.76E-05	2.08E-05	0.000
	Is_Member				-0.2546	0.048	0.000
Media Used	Video_Creators				0.1576	0.023	0.000
	Img_Creators				0.0315	0.009	0.000
	Video_Prod				-0.0248	0.019	0.187*
	Img_Prod				0.0493	0.002	0.000

*cannot conclude statistically significant at 5%. All others are statistically significant.

Finally, the concluding results from logistic regression suggest that the top three important indicators for launching successful project is: 1) in the popular or in demand project categories, 2) creators earned the badge status of being backer's favorite, and 3) having a substantial number of words to explain clearly of project objectives in blurbs. Although some of these characteristics only partially agree between the logistic and multiple regression models, but majority of these characteristics show sign of significance towards the log of odds in project success level.

2.6 Discussion and Conclusion

In this research, the provided results offered evidence to summarize findings and achieved contributions made on critical indicators and their association strength in crowdfunding campaign success. Both multiple and logistic regression models shed light on features that project creators can focus on for a better chance to meet or even exceed the funding goal. For the most part in feature significance, the agreement between the two regression models confirmed that fundamental project features play an important role in funding success. These features include realistic funding goals (not too low or too high), appropriate funding periods (not too long or too short), being featured by staff for publicity, and a diverse range of pledge options to attract investors as supported by Mollick [10] and Zhou et al. [18]. Furthermore, the analysis of other features leads to the following conclusions and contributions made:

- Features of Products

  Pitching a descriptive blurb for project mission correlates with funding success is verified with many other studies, such as Koch et al. [19] and Zhou et al. [18]. The first novel finding is that providing budget plan or product timeline to investors for transparency, this feature is also significant and correlates with funding success.

- Features of Creators

Contrary to Koch et al. [19], this research shows that history of backing other projects does not provide any significant advantage in funding success. Instead, history of previous projects and listing of relevant websites in creators' biography are beneficial. This could be due to the sample size used in this study that covered a much wider scope of projects. Moreover, my investigation extends to examine creators' education and their relevant experience, on top of their earned badge status as backers favorite, which mitigated some of the gaps in other literatures. The second novel finding suggests that badge status and mentioning relevant experience are important to project success; while the significance in mentioning educational degree or lengthy biography is inconclusive since two models did not agree.

- Features of Team Formation

Like many other well-established studies, this results also suggest that having other project collaborators and a wide network of Facebook friends yield a likelihood of project success. On the other hand, my finding does not suggest that mentioning members will correlate a high success rate but it imposes a significance to project success, unlike the conclusion from Frydrych et al. [12] where their studies addressed the importance of organizational legitimacy and demonstrated that projects created from teams or pairs had a higher chance to succeed than projects from individuals. I speculate that this might be due to the narrower scope of coverage for having only 421 projects (unlike this study which has 11,408 projects) and Frydrych et al. [12] stated ‘ the numbers are too small to draw firm conclusions’.

- Features of Media Used

  One last important novel discovery is that this research can distinguish creators specific and product specific significance among videos and images used. This research is among the first to look beyond the aggregate level of significance in images or videos used for project success. Most other research only focused on the existence of media used as a whole and its relational strength to project success only. An interesting finding here suggests that the introduction or the storytelling from creators will be more advantageous and influential using videos as a portal, than through images when it comes to project success. By contrast, the effect for showcasing products is the opposite.

In conclusion, this research casts a new light on hidden features that were not previously explored, in addition to providing a further validation on features that have previously studied.

2.7 Limitations and Future Research

There are several limitations in this study. One major limitation is the lack of resources to extract more contextual information from media used. I could explore further the actual context of each video or each image for creator specific or project specific that correlates project success. The second limitation is the lack of detailed information in educational background. My study can only verify whether this piece of information is being mentioned or not, either in the project description or creator's biography, due to the challenge to separate out the earned degrees for a large group of team members. Another future improvement is to explore other techniques to attain a more 'linearized' multiple regression for a better result to be in line with all linear assumptions, since the homoscedasticity of the errors assumption was violated. Despite these limitations, the findings are valuable and promising considering serving as a starting point to expand on these features using other data transformation method or modeling techniques to identify additional success indicators, and contributing insights to better refine the data for similar fashion to verify these findings.

3. CLASSIFYING DIFFERENT LEVELS OF PROJECT SUCCESS WITH MACHINE LEARNING MODELS

3.1 Introduction

Given key indicators were identified in the previous chapter and provided insights for creating a project, the natural question that follows is the likelihood of success if given a set of input features from a campaign. To make classification possible and to capture the relationship among features for behavioral patterns, machine learning algorithms and optimization techniques are adopted for modeling in this chapter due to the inadequacy of using simple deterministic regression models for solving complex problems, such as crowdfunding classification. One of the objectives in this chapter is to create an optimal model for project success classification, and to detect distinguishable traits among different levels of success.

To develop an optimal model for Kickstarter's project success, different algorithms were investigated and outcomes from optimization techniques were analyzed for their impacts. As early as in 2013, research communities had gradually broadened their understanding in crowdfunding by using different machine learning models for a range of topic investigations, including project success classification. However, as mentioned before, most of the literature is limited to classifying binary success outcomes. In this research, one of the aims is to explore beyond the binary outcomes in funding success, but to also examine the characteristics of the extremely successful projects at the top end of the spectrum. Afterall, the intent of the optimized machine learning model is to classify which one of the three success categories (failed, successful, and super successful) a project would fall into, in order to serve as a tool for creators to improve campaign strategies. Therefore, to properly address this gap in multiclass problems, the first step in the research framework is to compare a class of machine learning algorithms for sifting out the best performing classifier. From naïve methods (e.g., kNN, Naïve Bayes) to more complex methods (e.g., SVM, MLP, ensemble methods), my findings suggested that Extreme Gradient Boosting (XGBoost) had the best performance. Next, I proposed a novel metaheuristic population-based algorithm, Binary Particle Swarm Optimization (BPSO), to integrate with Extreme Gradient Boosting classifier in the context of feature selection for crowdfunding classification. Then, the outcomes from five runs of BPSO were explored to understand the selected subset of features in order to solve the combinatorial feature selection problem using optimization. Most importantly, I rectified some of the previous findings on classification traits of the other machine learning models. Last, the concluded findings in this chapter would serve as a new piece of information to future entrepreneurs for better designing a project campaign.

3.2 Related Work

3.2.1 Machine Learning Models Literature Reviews

Immediately after crowdfunding started to gain attraction, there has been a considerable body of literature on using machine learning theories to research and model project features for prediction in project success. One of the earliest proposed computational methods to predict crowdfunding success was to use multiple classifiers for each set of indicators. Etter et al. [20] conducted a study on Kickstarter data using dynamic attributes of the campaign and its related Twitter data to predict success instead of relying on static attributes like other studies. Indicators were separated into money-based predictors (such as amounts of money pledged) and social predictors (such as number of tweets, number of twitter users, and backers), then applied k-Nearest Neighbors (kNN) and Markov model for money-based and Support Vector Machine (SVM) with radial basis function (RBF) for social predictors. A trained Support Vector Machine (SVM) of the two combined predictors was also integrated, the results suggested that "on average 4 hours after the launch of a campaign, the combined predictor can assess the campaign's probability of success with an accuracy higher than 76%" [20]. To expand on some previous research in Kickstarter dataset, Chen et al. [21] proposed random forest algorithm on five sets of features (intrinsic characteristics, financial mechanism, content quality and sentiment, social interaction, and progression effect) to predict project success against benchmark, and also made predictions at different time points of the 7-day campaign as well. Their study concluded that not only using random forest algorithm made improvement in prediction against benchmark, but taking different stage in campaigns for prediction could improve accuracy from 72.89% initially to 89.62% after day 7 [21].

In 2018, further studies were developed to assess machine learning performance and scalability for a more robust classification model. Yu et al. [22] suggested multilayer perceptron (MLP) neural networks, one of the representation learning methods in deep learning, to apply in the crowdfunding binary outcome classification. Their suggested model was constructed using two hidden layers of 100 and 60 neurons respectively, activated by rectified linear unit (ReLU) function and using Adam optimizer to minimize binary cross entropy loss function. With 32 input features, model performance was tested and it showed promising consistency of around 93% accuracy when using one-fourth, two-fourth, three-fourth and full data set [22]. In addition, among all the compared classifiers such as Adaptive Boosting (AdaBoost), Random Forest, Decision Tree, SVM, Logistic Regression, and Naïve Bayes, where MLP demonstrated the highest performance. The effectiveness of using MLP with better performance in success classification was also supported by the study from Wang et al. [23]

Advanced ensemble models also attracted much attention in the context of crowdfunding. Using only eight features with data collected from 2014 till February 2019, Jhaveri et al. [24] applied Random Forest (RF), Extreme Gradient Boosting (XGBoost), Category Boosting (CatBoost), and Adaptive Boosting (AdaBoost) for comparison. Their results suggested that CatBoost yielded the best performance with 83% in accuracy without subsampling [24]. In fact, the intention behind the chosen features used in this study is aligned with the purpose of my study, where only the initial funding features before launch were considered, and excluding dynamic features were collected during or at the end of the campaign period. Similar to the research that was done by Greenberg et al. [25], 13 features were extracted for 13,000 projects where the purpose of their model was to focus on pre-launching using various tree algorithms. However, their study did not provide evidence where Adaboost yielded the best results;

in fact, a less complex model (such as Random Forest) was just as well performed as boosting tree models [25].

In addition of classifying binary success outcomes in Kickstarter data, some other studies like Rakesh et al. at Wayne State University [26], focusing on developing a project recommendation system for backers through a gradient boosting tree (GBtree) where their results showed that 89% in accuracy and precision up to 80% by using only the first three days of project features. Some authors were interested in using Launchpad data from Amazon to further develop prediction in users' rating among successful and unsuccessful Kickstarter projects [27], or using Twitter data to predict project success with survival analysis using censored regression approach with logistic distribution and log-logistic distribution [28].

3.2.2 Features Selection Literature Reviews

One of the crucial steps in creating machine learning models is feature selection. This pertinent process is to subset meaningful input features to avoid "irrelevant and redundant" [29] features, in order to prevent overfitting and to ensure input data quality when training the model.   A large number of existing studies in crowdfunding reported key indicators and examined correlation strength of project features, where researchers solely extracted features and applied them directly to develop models based on their examinations. However, systematic feature selection steps were merely discussed in the context of crowdfunding exploration. Literatures on using feature selection to arrive final model are limited and rarely documented. In one of the recent crowdfunding research, Chen et al. [30] presented a method using lexicon-based feature selection to address the issue of a high dimensional extracted features by defining “content features” through text mining. The authors implemented decision trees, LASSO, and SVM-RFE methods to compare and select a subset of features where LASSO produced the best feature set [30].

Several papers also suggested a series of novel approach to perform feature selection in crowdfunding classification. Ryoba et al. [31] proposed a metaheuristic whale optimization algorithm (WOA) to perform feature selection in crowdfunding success prediction, to search the optimal set of features for the k-Nearest Neighbor (kNN) model to yield high performance with only 9 features. The Whale optimization algorithm was developed by Mirajlili and Lewis [32] in 2016 for solving optimization problems. This algorithm was being integrated and applied later in feature selection that demonstrated comparable results to other optimization algorithms, such as Genetic Algorithm (GA), the Ant Lion Optimizer (ALO), and Particle Swarm Optimization (PSO) when trained on the UC Irvine datasets [33].

Particle Swarm Optimization (PSO) has also been explored in other studies by Ryoba et al. [34] to perform feature selection with k-Nearest Neighbor (kNN) classifier from updates and comments at different phases for crowdfunding prediction. Their research successfully identified the most essential feature across five different stages during the campaign. Their findings showed that the number of updates, the polarity of comments, the readability of updates posted are important [34]. Further development in crowdfunding classification by using Particle Swarm Optimization was also used with other advanced classifiers, such as Light Gradient Boosting machine. Geng et al. [35] proposed a Swarm enhanced Light Gradient Boosting machine (S-LightGBM) to compare with logistic regression and support vector machine for project outcomes prediction performance. Their results suggested that the proposed method to tune hyperparameters was outperformed the other two classifiers and was able to attained a higher accuracy from 83.01% to 85.04% in classification performance [35]. They also compared using traditional stepwise method to PSO in parameter tuning, where their results showed that PSO was able to yield comparable performances as well [35]. This evidence was also supported by the research from Korovkinas et al. [36] when the authors used PSO to perform parameter tuning for support vector machine on different datasets, such as Amazon customer reviews dataset and the Stanford Twitter sentiment corpus dataset.

3.3 Data

To create the optimal model for project success with multiclass classification, I extended my research using the finalized dataset from last chapter. The scope of the data remains focused on all the U.S campaigns launched in 2019, and excluded campaigns with less than a goal of $1000 raised or campaigns with more than $500,000. Duplicated campaigns (with same IDs) or suspended campaign were also eliminated in the sample. Last, any campaigns with invalid nor inactive URL site were also removed. To incorporating the knowledge I learned using the identified key indicators to project success from previous chapter as input features, I also included the month when creators joined Kickstarter and the U.S state location of project with one hot encoding and frequency encoding respectively. There were then 36 features engineered for the 11,410 campaigns in the final sample. Definition of features used are detailed in Appendix A.1.

The final class for each project is labeled as one of the three classes: “failed”, “successful”, and “super successful”. Each class is assigned as a result of the computed ratio of total amount raised to the funding goal, Success Rate variable. Projects are assigned to “failed” if the Success Rate value is less than 1. Similarly, projects are assigned to “successful” if the Success Rate ratio value is greater than or equal to 1 but less than 1.5. The rest of the projects are assigned to “super successful” if the Success Rate ratio value is greater than or equal to 1.5, since these projects represents a superior fund-raising performance by exceedingly more than the funding goal.

3.4 Methods

3.4.1 Overview of Research Design

Following the data preprocessing steps described in Section 2.3 and Section 3.3, multiple steps were involved in the classification modeling process flow. The flow diagram Figure 3.1 illustrates the outline of the modeling framework creating the optimal model for classification. Before applying any computational methods, the pre-processed dataset with 11,410 observations were split by 70:30 ratio into training and testing data sets. Since the data was skewed with imbalance class as shown in Figure 3.2, a balance training dataset was created by performing the Synthetic Minority Oversampling Technique (SMOTE). SMOTE uses data augmentation approach to synthesize minority observations, where “the minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors” [37]. The number of “failed” class was up sampled from 2663 to 3379 observations. The “successful” class remained unchanged with 3379 observations, and the number of “super successful” class was induced from 1945 to 3379 with synthetic over-sampling technique in SMOTE. Instead of using a 10-fold due to expensive computation cost, a 5-fold cross validation was used to evaluate for the best classification performance with the training dataset among a series of classifiers. Next, classifications were made on testing dataset and multiple metrics were computed for each of the classifiers for comparison.

Once the best performing classifier was selected, a swarm-based feature selection method was implemented for feature selection to improve the model performance and to reduce irrelevant features. Finally, hyperparameters tuning was applied to select the optimal set of parameters to further improve classification performance.

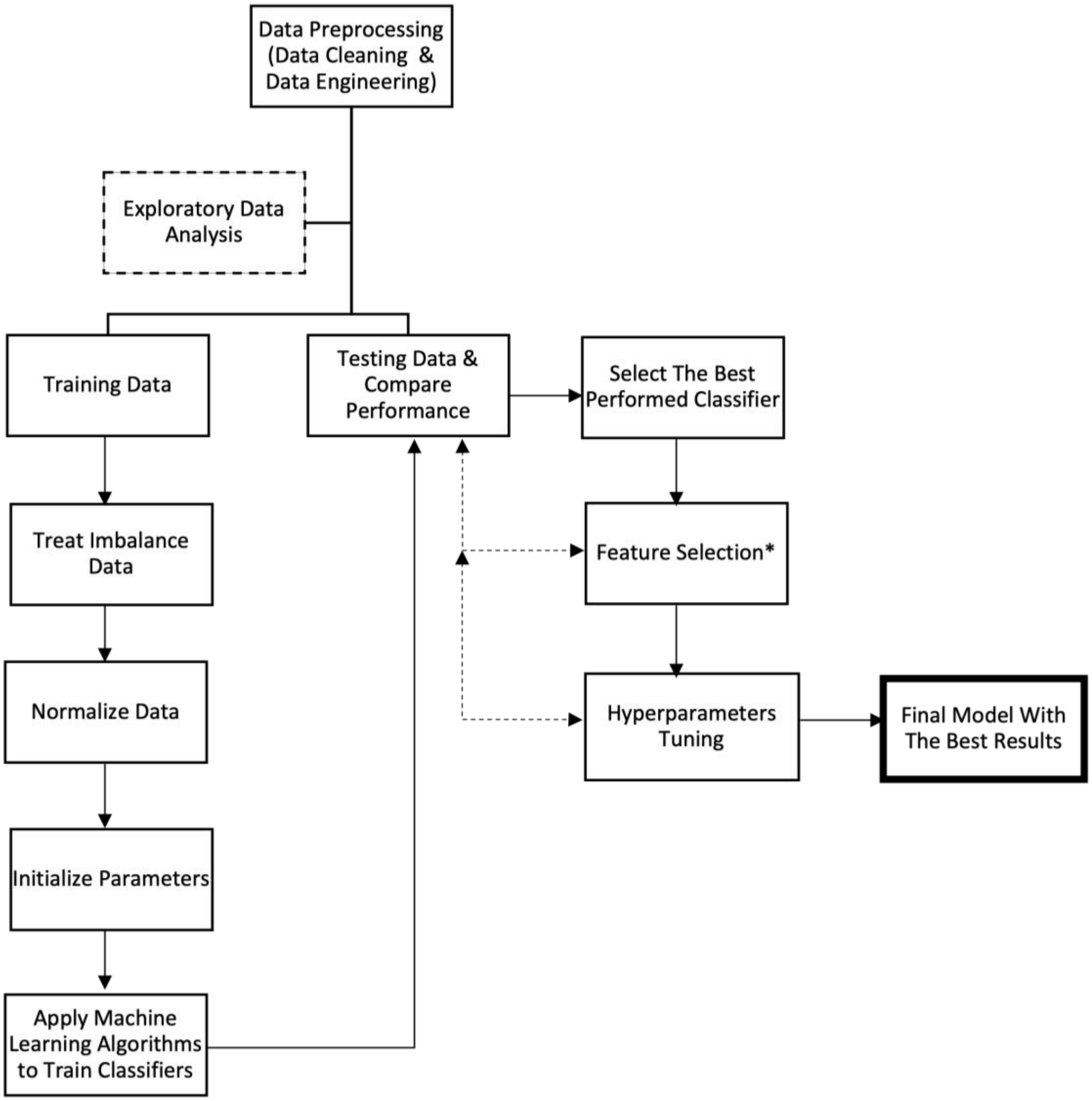

Data Preprocessing (Data Cleaning & Data Engineering)
Exploratory Data Analysis
Training Data
Testing Data & Compare Performance
Select The Best Performed Classifier
Treat Imbalance Data
Feature Selection*
Normalize Data
Hyperparameters Tuning
Final Model With The Best Results
Initialize Parameters
Apply Machine Learning Algorithms to Train Classifiers

Figure 3.1: Process flow diagram for selecting the best performing machine learning model to classify project success.

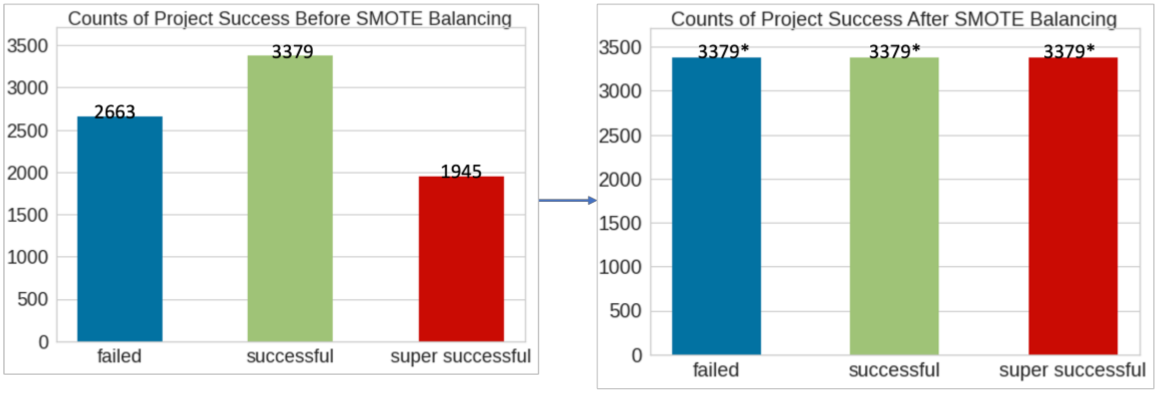


Figure 3.2: The number of observations in training data set for project success class before (top) and after (bottom) using SMOTE balancing.

3.4.2 Supervised Machine Learning Models Comparison

For multiclass classification of project success, a series of twelve supervised machine learning algorithms were proposed and compared for the best optimal model to classify project success. The twelve compared algorithms were:

1. Logistic Regression (LR): Logistic regression transforms the dependent variable Y with logit [38], where logit represents the natural logarithm of odds. Odds represents the ratio of the probability of event Y takes place to probability of event Y will not take place. The general form of logistic regression model for three project success classes is shown in Equation (3.1):

$$p(C_k|X) = Y_k(X) = \frac{\exp(a_k)}{\sum_{j=1}^{3}\exp(a_j)} \tag{3.1}$$

where C is the class event, $a = W^T X + b$, $W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,3} \\ \vdots & \ddots & \vdots \\ w_{3,1} & \cdots & w_{3,3} \end{bmatrix}$ is the matrix of weight vectors, X is a set of input independent variables, b is the matrix of biases, and a={a_1, a_2, a_3}.

2. Decision Tree (DT): Decision Tree searches feature space and "recursively partitioning" [39] it into nodes and leaves for a set of decision rules to construct tree-like structures in classification model. Gini index is used as one of the scoring criteria to measure class impurity for partition as shown in Equation (3.2) and Equation (3.3) [39]:

$$Impurity\text{=}1\text{-}\sum_{\mathrm{j}} ||\mathrm{p}_{(\mathrm{j})}\mathrm{N}_{\mathrm{j}}\,(\mathrm{t})/\mathrm{N}_{\mathrm{j}}||^2 \tag{3.2}$$

$$\mathrm{Gini}\text{=}impurity(Parent)\text{-}\sum\nolimits_k (p_k)impurity(Child_k) \tag{3.3}$$

Where p(j) is the prior probability of class j, $N_j(t)$ is the number of observations in class j for node t, and N_j is the number of observations in class j.

3. k-Nearest Neighbors (kNN): k-nearest neighbors calculates distance between the query point to the rest of the data points, then a k-number of closest data points are chosen and the most frequent class in the neighborhood will be the class assignment for the query point. Euclidean distance for any given two data points is implemented in kNN is shown in Equation (3.4):

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^2\right)^{\frac{1}{(p=2)}} \tag{3.4}$$

where power(p) =2 is equivalent as Euclidean distance in Minkowski distance metric.

4. Naïve Bayes (NB): Naïve Bayes builds on the principle of Bayes' Theorem that consists of calculating the prior and the posterior probability of an event under a set of condition as described in Equation (3.5):

$$P(y_j|x) = \frac{\prod_{k=1}^{p} P(x_k|y_j)P(y_j)}{P(\boldsymbol{x})} \tag{3.5}$$

where $P(y_j|x)$ is the calculated posterior probability for jth class, $P(x_k|y_j)$ is the likelihood of input features given jth class, $P(y_j)$ is the prior probability of jth class, and $P(\boldsymbol{x})$ is the prior probability of input features. The term "Naïve" comes from the assumption of all input variables have conditional independence property from each other; hence, Naïve Bayes is a generative model with fewer parameters [40] as a more intuitive learning approach compares to other algorithms. For classification problem, the maximum a posteriori (MAP) of Equation (3.5) is used to determine the final class assignment ($\hat{y}$) and it is written as Equation (3.6):

$$\hat{y} = \arg\max_{k} \prod_{k=1}^{p} P(x_k|y_j)P(y_j) \tag{3.6}$$

5. Quadratic Discriminant Analysis (QDA): Quadratic Discriminant Analysis is the extension of the Linear Discriminant Analysis (LDA) that can be used as a nonlinear classifier, where it does not assume covariance is identical for each class. QDA calculates the discriminant score to estimate each class for the query point by using posterior distribution [41]. The discriminant score

follows the quadratic function as depicted in Equation (3.7) for decision boundaries:

$$\delta_k(x) = \left(-\frac{1}{2}\right)\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_\mathrm{k} \qquad (3.7)$$

where μ_k is the mean of all training observations in class k, Σ_k is the covariance matrix of the class k, and the π_k is the ratio of training observations for class k to all training observations. For classification problem, QDA uses the maximum a posteriori (MAP) of the computed quadratic discriminant scores among classes in order to determine the final class ($\hat{G}$) in Equation (3.8):

$$\hat{G}(x) = \arg\max_k \delta_k(x) \qquad (3.8)$$

6. Support Vector Machines (SVM): Support Vector Machine consists of identifying support vectors to build the optimal hyperplane for hyperplane surface and minimizing quadratic function to establish the maximal marginal space for best separability as given in Equation (3.9):

$$Maximize: Margin\ (M) = \frac{2}{||w||} \rightarrow Minimize\ f(x) = \frac{1}{2}w^T w \qquad (3.9)$$

where minimizing weight vector, $\left||w|\right| = \sqrt{w^T w} = \sqrt{w_1^2 + w_2^2 + w_3^2 + \cdots + w_n^2}$, is equivalent to maximizing margin M since minimizing $\sqrt(f)$ is equivalent to minimizing f [42]. Using Lagrangian multiplier method in Equation (3.10) to solve a constrained optimization problem, the SVM problem can be rewritten in Equation (3.12) by substituting

Equation (3.9) for margin and Equation (3.11) for the general SVM formulation into Equation (3.10):

$$L(x, a) = f(x) + \sum_i a_i g_i(x) \tag{3.10}$$

$$g(x): y_i(w \bullet x_i) + b - 1 = 0 \tag{3.11}$$

$$\min L_d = \frac{1}{2}||w||^2 - \sum_{i=1}^{l} a_i y_i (x_i \ \bullet w + b) \ + \sum_{i=1}^{l} a_i \tag{3.12}$$

where $l = number\ of\ training\ observations,$ w=$\sum_{i=1}^{l} a_i y_i x_i$ and $\sum_{i=1}^{l} a_i y_i = 0.$

By rewriting the above Lagrangian primal into dual problem formulation, the final function to optimize for the SVM is written in Equation (3.13):

$$L_d = \sum a_i - 1/2 \sum a_i a_j y_i y_j K(x_i \ \bullet x_j) \tag{3.13}$$

Such that $\sum_{i=1}^{l} a_i y_i = 0 \ and\ a_i \geq 0$, and where $K(x_i \ \bullet x_j)$ is the applied kernel function if any. In this research, kernel function is applied to transform data into nonlinear by mapping data to higher dimensional feature space for hyperplane construction, and Radial Basis Function (RBF) was the chosen kernel in Equation (3.14):

$$K(x_i, x_j) = \exp\,(\frac{-||x_i - x_j||^2}{2\sigma^2}) \tag{3.14}$$

7. Multi-Layer Perceptron (MLP): A Multi-Layer Perceptron is a type of neural network that is built with artificial neurons. Its architecture is comprised of input layers with input units, hidden layers with hidden units, a transfer function to compute the net input for the activation function, where the it activates for the output layer. In this chapter, I applied a feedforward neural

network where a set of weights initialized at the input layer for a forward pass. The compared difference between the computed output in Equation (3.15) and the target output is used to update the vectors' weights through backpropagation using the update rules in stochastic gradient descent (sgd) [43], while using rectified linear unit in Equation (3.16) as activation function for learning.

$$y = f(a) = f\left(\sum_i w_i x_i + b\right) \tag{3.15}$$

where a is the activation function, w is the weight of a vector and is b the bias.

$$f(a) = \max(0, a) \tag{3.16}$$

8. Bootstrap Aggregating (Bagging): Bagging is a type of ensemble methods where multiple base models are built for majority voting or averaging results to compute final prediction. Each subset of samples is drawn with replacement in a base model. With this structure of replacement sampling, bagging can neutralize the instability among models in order to be more robust against some of the noisy data and avoid overfitting as well [44]. In this research, the base model used is a decision tree.
9. Random Forest (RF): Random Forest is a special form of bagging ensemble method where only a subset of features is being selected at random while the base models are built on a subset of samples selects at random with

replacement as well. A number of decision tree are split by using Gini impurity in Equation (3.2) and (3.3).

10. Adaptive Boosting (AdaBoost): Adaptive boosting is a type of boosting ensemble methods where it uses a weighted method to put focus on training instances that are wrongly classified through iterative reweighting method. Each sample of a base model is drawn with replacement. All training instances start with equal weight for the first round, and then compute error on the trained set, where instances are "trained" wrongly will have a set of adjusted weights [45]. This re-weighted training set will feed into the next trained model iteratively (if the error does not exceed more than 50% or else abort processes) where hundreds of rounds are performed sequentially. Finally, all rounds of results will be combined to make final decision through weighted voting or weighted averaging. In this research, the base model used is decision tree.
11. Extreme Gradient Boosting (XGBoost): Extreme gradient boosting is a sophisticated tree-based version of gradient boosting that includes implementation of regularization. Generally, Gradient Boosting is a type of boosting ensemble methods that adopts a forward stage-wise additive modeling approach, where it builds and adds classifiers to offset the weakness of existing models while optimizing the loss functions using gradient descent [46]. Unlike Gradient Boosting, Extreme Gradient Boosting calculates the second order gradients to approximate the objective function [47] using

Taylor expansion to be written in Equation (3.17). It also includes a regularization term to avoid overfitting as given in Equation (3.18).

$$\hat{O}(f) = \sum\nolimits_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \qquad (3.17)$$

where f represents tree structure, $g_i = \partial_{\hat{y}} L(\hat{y}_i, y_i)$ is the first order gradient statistics in the loss function, and $h_i = \partial_{\hat{y}}^2 L(\hat{y}_i, y_i)$ is the second order gradient statistics in the loss function.

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \left|\left| \omega \right|\right|^2 \qquad (3.18)$$

Denoted f represents tree structure with leaf weights (ω) and number of leaves (T). To manage how conservative the model should split, Lasso regularization of coefficient (γ) and ridge regularization of coefficient (λ) can be tuned. Since $\mathrm{I_j}$ defines as a set of instances at leaf j, the objective function can be rewritten in Equation (3.19) by expanding with the regularization term from Equation (3.18):

$$\hat{O}(f) = \sum\nolimits_{i=1}^{T} \left[(\sum\nolimits_{i \in I_j} g_i)\, \omega_j + \frac{1}{2} (\sum\nolimits_{i \in I_j} h_i + \beta)\, \omega_j^2 \right] + \Omega(f_t) \qquad (3.19)$$

To split trees, Extreme Gradient Boosting does not use Gini index nor entropy for decisions, it uses the following gain computation instead:

$$\mathrm{G_j} = \sum_{\mathrm{i} \in I_j} g_i \qquad (3.20)$$

$$\mathrm{H_j} = \sum_{\mathrm{i} \in I_j} h_i \qquad (3.21)$$

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L + \beta} + \frac{G_R^2}{H_R + \beta} + \frac{(G_R + G_L)^2}{H_R + H_L + \beta}\right] - \alpha \qquad (3.22)$$

where the first expression represents the score of the left child, the second expression represents the score of the right child, and the third expression represents the score if no splits occur. The α is the complexity cost if new split is added. In the research from Chen and Guestrin [48], Extreme Gradient Boosting has proven to attain higher accuracy with higher computation speed by parallelization that is suitable for high scalability.

12. Light Gradient Boosting (LightGBM): Light gradient boosting is a modified version of gradient boosting method that was created by Microsoft in April 2017 [49]. Ke et al. [50] have demonstrated that using gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) techniques in Light Gradient Boosting can significantly improve the computational efficiency and memory consumption with comparable accuracy to Extreme Gradient Boosting, especially for handling large volume of data in solving classification or machine learning problems. With the implementation of GOSS in Light Gradient Boosting, data points with small gradients or small residuals in training error will be down sampled at random, while the data points with large gradients will be retained and weighted with more focus [50]. The main advantage of using GOSS can retain accuracy performance while significantly reduce data size to improve the training efficiency. Another important component in Light Gradient Boosting is exclusive feature

bundling which it uses feature scanning algorithm to create single feature from features that are mutually exclusive [50]. This method can significantly reduce the complexity of histogram building from O(#data * #feature) to O(#data * #bundle) and increase the speed of training, by taking advantage of the sparse property to bundle features in the histogram-based partitioning process.

3.4.3 Performance Metrics

After a series of supervised machine learning algorithms were implemented, their performance metrics were assessed and compared. Python scikit-learn v_0.24.2 library is used in this research for algorithms implementation and classification performance. Several evaluation metrics were chosen to compare as described in Equation (3.23) to Equation (3.26):

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{3.23}$$

where TP and TN represent the number of true positive and the number of true negative classification respectively, while FP and FN represent the number of false positive and false negative classification respectively.

$$Recall\ (detection\ rate) = \frac{TP}{TP + FN} \tag{3.24}$$

$$Precision\ (positive\ predicted\ value) = \frac{TP}{TP + FP} \tag{3.25}$$

$$F - measure = 2 * \frac{recall * precision}{recall - precision} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{3.26}$$

Where recall can measure Type I error and precision can measure Type II. The F-measure is the harmonic mean of recall and precision that is commonly used to measure test performance when imbalance class exists in trained model, instead of accuracy.

3.4.4 Binary Particle Swarm Optimization (BPSO) for Feature Selection

After the best performing classifier was chosen, feature selection was implemented to improve classification performance. The motivation behind feature selection is to search for relevant features to best represent the targets and minimize noise. The outcome from feature selection comes with "the advantages of improving learning performance, increasing computational efficiency, decreasing memory storage, and building better generalization models" [51]. Therefore, feature selection is pertinent to create optimal model as part of the optimization process. Inspired by the metaheuristic computation technique from Eberhart et al. [52] using swarm intelligence for optimization and the feature subset framework in crowdfunding prediction by Ryoba et al. [34], I used Binary Particle Swarm Optimization to search for the best feature subset.

To introduce, Particle Swarm Optimization (PSO) is a metaheuristic, evolutionary algorithm that simulates the social behaviors of how animals communicate information among themselves for survival. For example, birds flock synchronically and each searches for an optimal spot in order to land as a group, where the chosen spot to land provides the most advantage of accessing food resources and yet avoiding predators. Similarly, the main objectives of feature selection are to find a set of features that can yield the best classification power but also with the minimal number of features in the subset to reduce redundancy. Therefore, PSO can be implemented to conduct the subset search to solve this combinatorial optimization problem. The fitness function is adopted and demonstrated in Mafarja and Mirjalili's research for feature selection using Whale Optimization [33] in Equation (3. 27):

$$Fitness = \alpha\gamma_s(R) + \beta\frac{|S|}{|N|} \tag{3.27}$$

where $\alpha \in [0,1]\ and\ \beta = (1-\alpha)$ that balance between the classification quality and subset length. The $\gamma_s(R)$ is the classification error rate of a chosen classifier, given a subset S selected features among all features N.

One of the key components in PSO is the number of particles which represent a group of organisms. Each particle is defined with a velocity and a position, such that the i^th^ particle velocity can be represented as $v_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{iD})$ and its position can be represented as $x_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iD})$ with D-dimensional search space. The lower and upper bounds of the d^th^ dimension is denoted as l_d and u_d, where $x_{id\in[l_d,u_d]}$ and $d \in [1, D]$. The i^th^ particle can update its position and velocity by learning from previous experience and other particles, either using its personal best recorded position (pBest) or the best global position searched by the swarm (gBest) as shown in Equation (3.28) and (3.29):

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_1 * (pBest_{id} - x_{id}^{t}) + c_2 * r_2 * \left(gBest_{gd} - x_{id}^{t}\right) \tag{3.28}$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \tag{3.29}$$

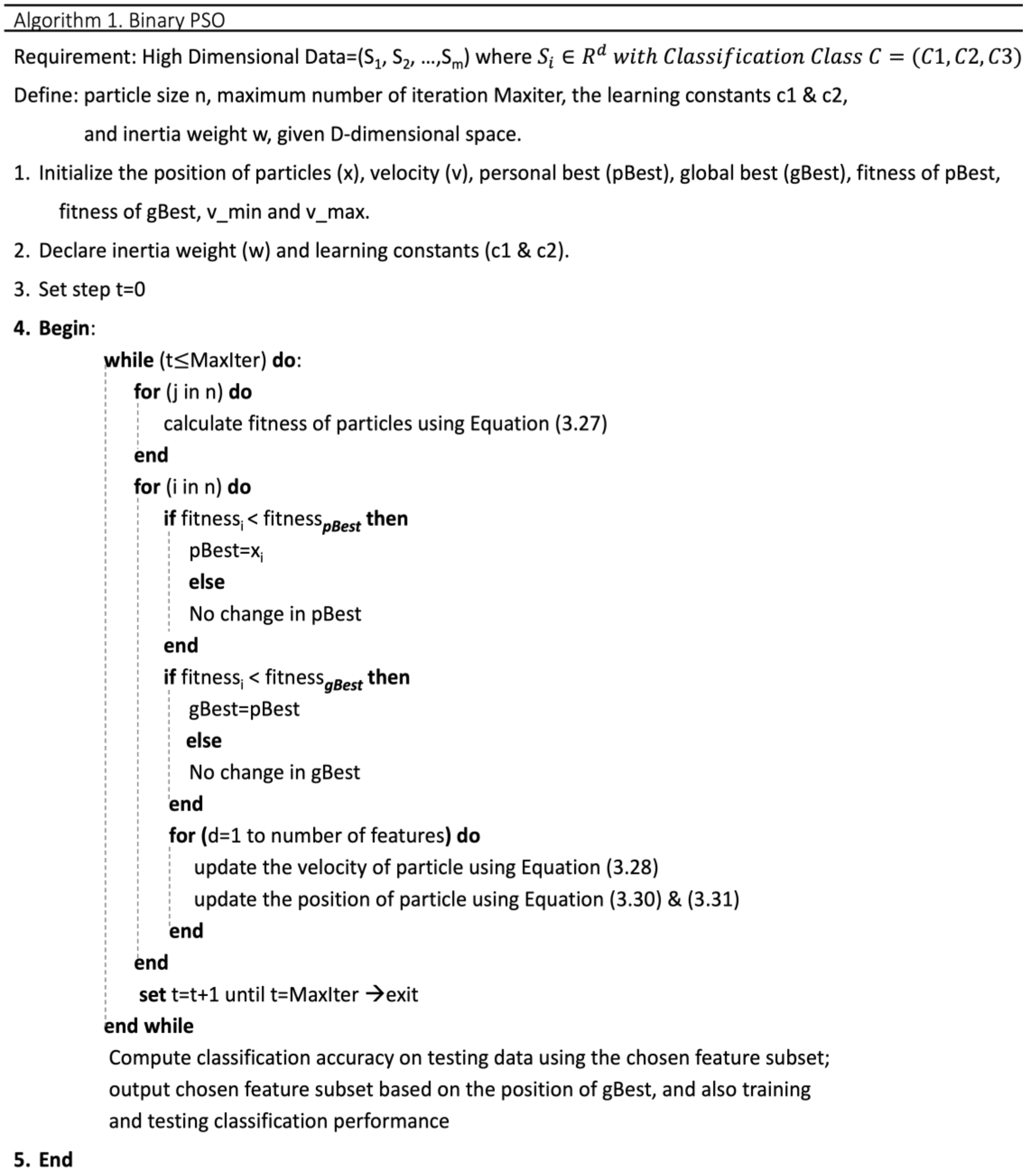
where t is the t^th^ iteration, w is the inertia weight to stabilize the trade-off between the exploration and exploitation stage in order to update the velocity of i^th^ particle. The learning constant is represented by c_1 and c_2, where random number is drawn from uniform distribution (U[0,1]) for r_1 and r_2. The best position of ith particle and the global best particle by swarm are denoted as $pBest_{id}$ and $gBest_{gd}$. The limit of v_{id}^{t} is bounded by the maximum velocity (v_{id}^{t+1}), and the value of v_{id}^{t+1} is within the predefined range, $[-v_{max}, v_{max}]$.

To apply PSO for feature selection, a binary PSO (BPSO) is integrated where each particle's position represents in binary values of 0 or 1 instead; therefore, x_{id}, pBest, gBest are encoded in binary representation [53]. To update velocity with Equation (3.28), Equation (3.30) and (3.31) can be used:

$$x_{id} = \begin{cases} 1, if\ rand < f(v_{id}) \\ 0, \quad otherwise \end{cases} \tag{3.30}$$

$$f(v_{id}) = \frac{1}{1 + e^{-v_{id}}} \tag{3.31}$$

where $rand$ is a random number from U[0,1], and $f(v_{id})$ is the sigmoid function to convert the velocity of ith particle between 0 and 1. The BPSO algorithm is presented in Algorithm 1 (Figure 3.3). In BPSO algorithm, after the completion of parameters initialization in the first step, a randomized group of swarm particles is generated. Using the fitness equation in Equation (3.27) to evaluate each particle, a global best position is determined. After finishing through a number of iterations, the overall global best particle position should be identified with the computed fitness as the candidate solution for feature selection. The parameters of BPSO are chosen as follows: inertia weight w = 0.8, acceleration constants c1 = c2 = 2, population size n= 20, and maximum iteration t= 100.



```
Algorithm 1. Binary PSO
Requirement: High Dimensional Data=($S_1$, $S_2$, …,$S_m$) where $S_i \in R^d\ with\ Classification\ Class\ C = (C1, C2, C3)$
Define: particle size n, maximum number of iteration Maxiter, the learning constants c1 & c2,
        and inertia weight w, given D-dimensional space.
1. Initialize the position of particles (x), velocity (v), personal best (pBest), global best (gBest), fitness of pBest,
   fitness of gBest, v_min and v_max.
2. Declare inertia weight (w) and learning constants (c1 & c2).
3. Set step t=0
4. Begin:
        while (t≤MaxIter) do:
            for (j in n) do
                calculate fitness of particles using Equation (3.27)
            end
            for (i in n) do
                if fitness$_i$ < fitness$_{pBest}$ then
                    pBest=x$_i$
                    else
                    No change in pBest
                end
                if fitness$_i$ < fitness$_{gBest}$ then
                    gBest=pBest
                    else
                    No change in gBest
                end
                for (d=1 to number of features) do
                    update the velocity of particle using Equation (3.28)
                    update the position of particle using Equation (3.30) & (3.31)
                end
            end
            set t=t+1 until t=MaxIter →exit
        end while
        Compute classification accuracy on testing data using the chosen feature subset;
        output chosen feature subset based on the position of gBest, and also training
        and testing classification performance
5. End
```

Figure 3.3: Binary PSO pseudocode for feature selection with three classes of project outcomes.

3.5 Results

3.5.1 Exploratory Data Analysis

In prior of applying supervised machine learning algorithms for classifying project success, different tools and data visualizations were employed to examine the general characteristics among three classes. As discussed in Figure 2.10 from Section 2.5.1, projects identified with the Technology category have the most number of failure campaigns; projects identified with the Music category have the most number of successful campaigns, excluding projects which did not identified with any categories; and both Games and Design identified projects have the most number of super successful campaigns in fundraising.

Table 3.1 provides summary statistics for each class where multiple trends are observed. Failed projects tend to have a longer mean duration of 40 days, and a wider spread of distribution in goal amount with the highest overall average of $29,939, as shown in Figure 3.4 and Figure 3.5. On the contrary, both successful and super successful projects tend to have a shorter mean duration and their mean goal amount is 1/3 less than failed projects. Extremely successful projects tend to have the highest values in these variables than successful and failed projects, as shown in Figure 3.6 and Figure 3.7, including pledge options made available to backers, creator's websites to share, Facebook friends, previously created projects in Kickstarter, backing of other Kickstarter projects, previous project collaborators, and current project collaborators. Moreover, extremely successful projects have the highest ratio with 23% of its projects being staff picked and featured. Figure 3.8 shows that 61% of successful projects and 53% of super successful projects tend to include or disclose delivery timeline or upcoming business plan on average.

In terms of media used on campaigns, Figure 3.9 shows that both successful and super successful projects include more videos and pictures of creators on average than failed projects. On average, there is at least one video of the products used in all levels of success in projects to campaign. Also, there are 5, 10, and 24 pictures of products used in failed, successful, super successful projects to campaign respectively. Figure 3.10 shows that super successful projects have a higher ratio of creators who earned backer's favorite status on average. Trends from the rest of the variables are less distinctive.

Last, correlations among variables were examined to ensure no heavy collinearity is exhibited in the set of input attributes used for machine learning models; correlation values are entailed in Appendix A.3. A multivariate RadViz plot distributes features around the circumference of a circle is also adopted to check for distinguishable separability among classes, where data points are being normalized and mapped on the axes from the center to each arc. Figure 3.11 shows all three classes are overlapped without high separability.

Table 3.1: Summary statistics of the main variables used by class.

	failed		**successful**		**super successful**	
Variables	**Average**	**SD**	**Average**	**SD**	**Average**	**SD**
goal	29939.21	60132.54	10330.90	15353.20	11731.72	23845.56
Duration	40.22	14.58	32.40	10.25	31.49	10.18
staff_pick	0.02	0.15	0.20	0.40	0.23	0.42
Number of Pledge Options	5.87	4.74	10.11	6.40	10.76	7.02
Number of Creator Web	0.97	1.27	1.48	1.43	1.82	1.50
Number of Created Proj	1.41	1.36	2.43	4.12	5.42	8.40
Number of Project Backed	2.08	10.42	13.49	43.29	31.44	67.82
Friends	268.05	799.14	506.24	1050.36	577.21	1180.15
Num_Collabs	0.17	0.55	0.47	0.98	1.29	2.07
Number of Precollabs	0.02	0.12	0.04	0.20	0.11	0.32
Video_Creators	0.45	0.81	0.85	0.94	0.74	1.20
Img_Creators	0.63	2.04	1.76	2.70	1.20	2.38
Video_Prod	0.50	1.05	0.48	1.06	1.07	1.81
Img_Prod	5.15	8.29	10.14	12.41	24.29	20.68
name_wdct	5.20	2.60	5.83	2.60	6.39	2.55
blurb_wdct	16.26	6.45	15.96	6.06	16.52	5.60
is_backerfav	0.01	0.08	0.07	0.26	0.33	0.47
Plan_Timeline	0.35	0.48	0.61	0.49	0.53	0.50
bios_wdct	63.84	82.89	73.92	87.52	70.58	84.34
is_exp	0.20	0.40	0.25	0.43	0.30	0.46
is_degree	0.16	0.37	0.16	0.37	0.10	0.31
is_member	0.79	0.41	0.78	0.41	0.69	0.46

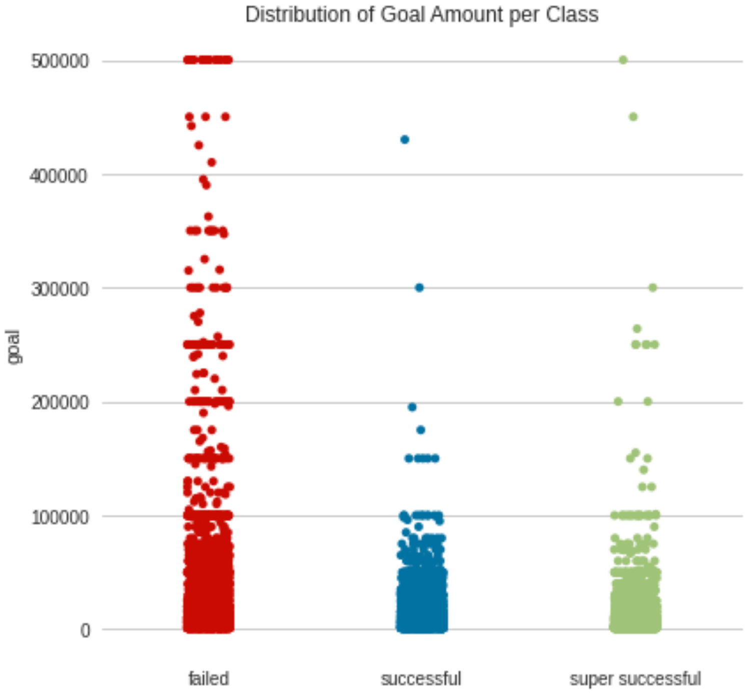


Figure 3.4: Underlying distribution of goal amount per class label with strip plot.

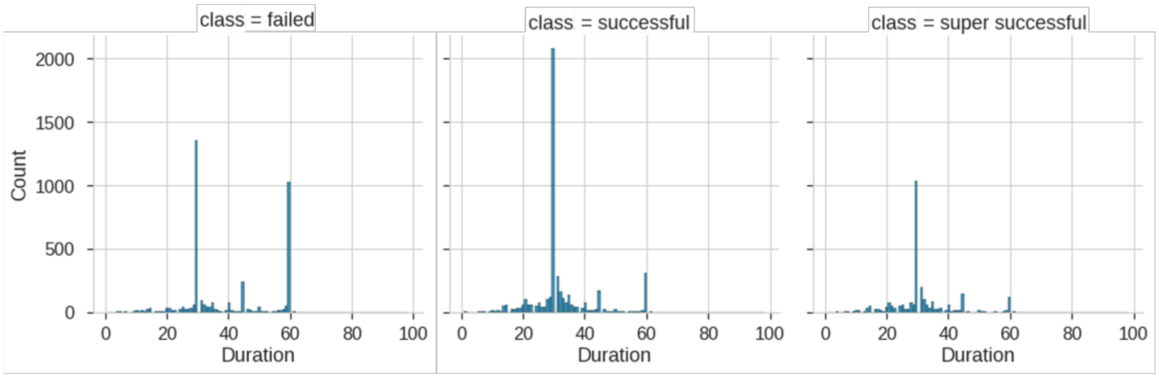


Figure 3.5: Distribution of project duration per class label.

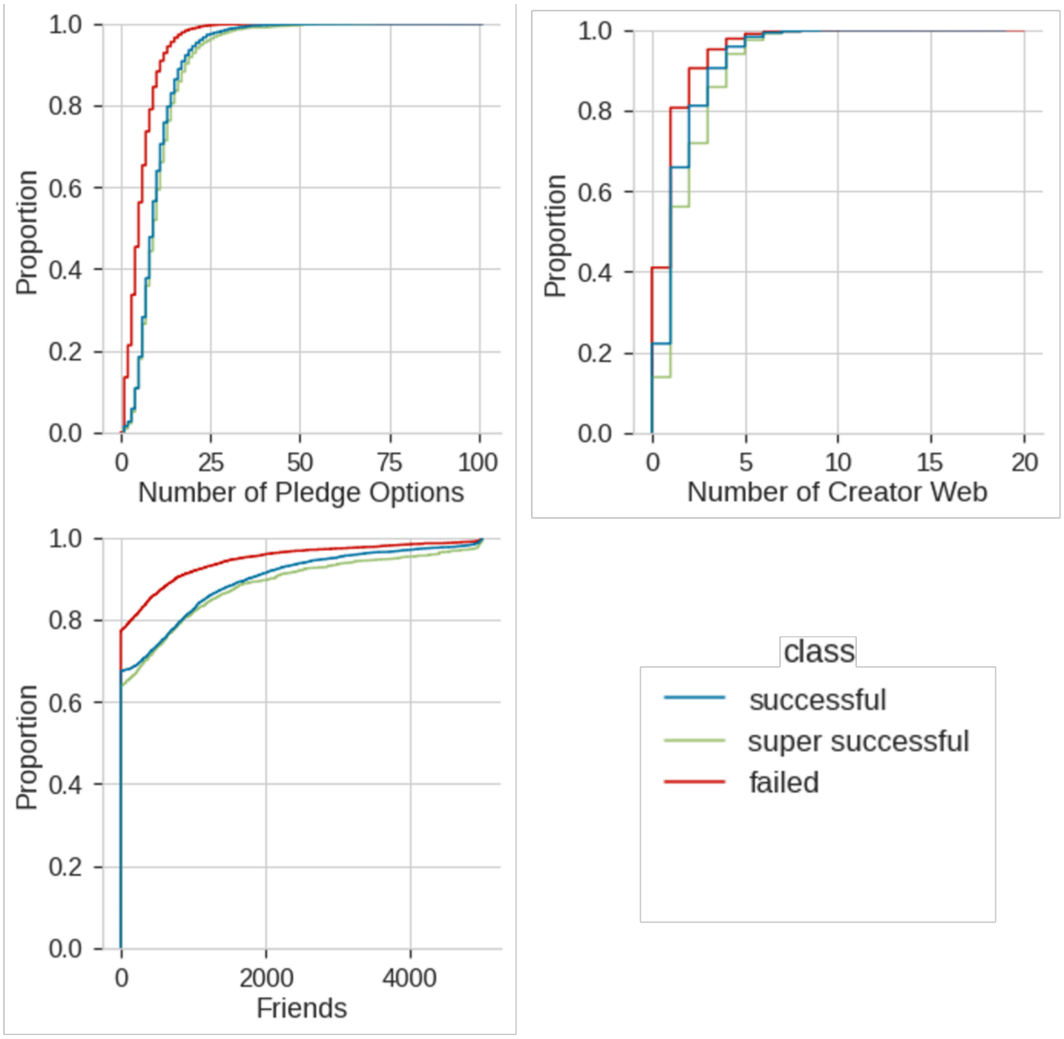


Figure 3.6: Empirical cumulative distribution of pledge options, creator's websites, and friends per class label.

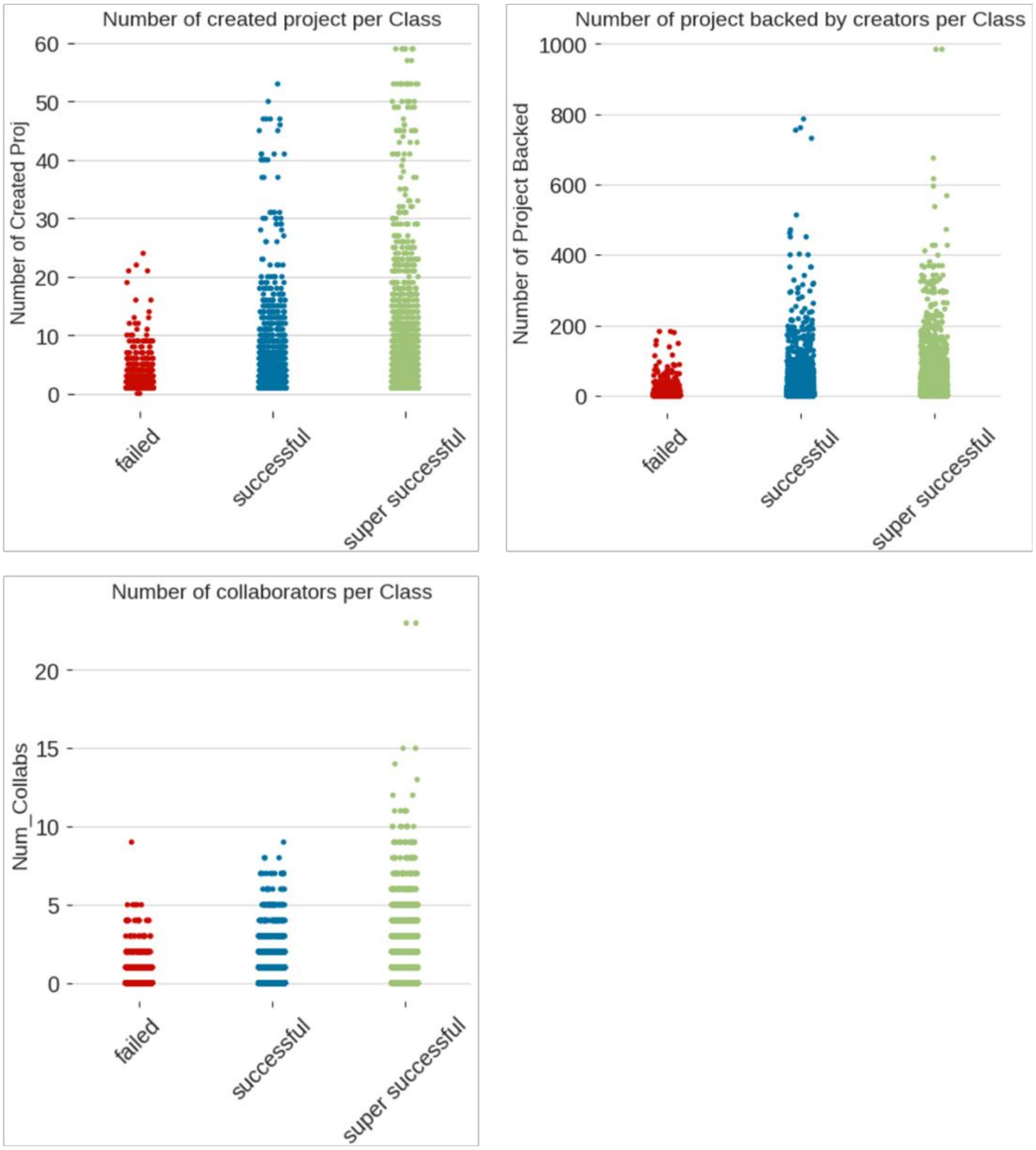


Figure 3.7: Strip plot distribution for previously created projects, backed projects, and project collaborators per class label.

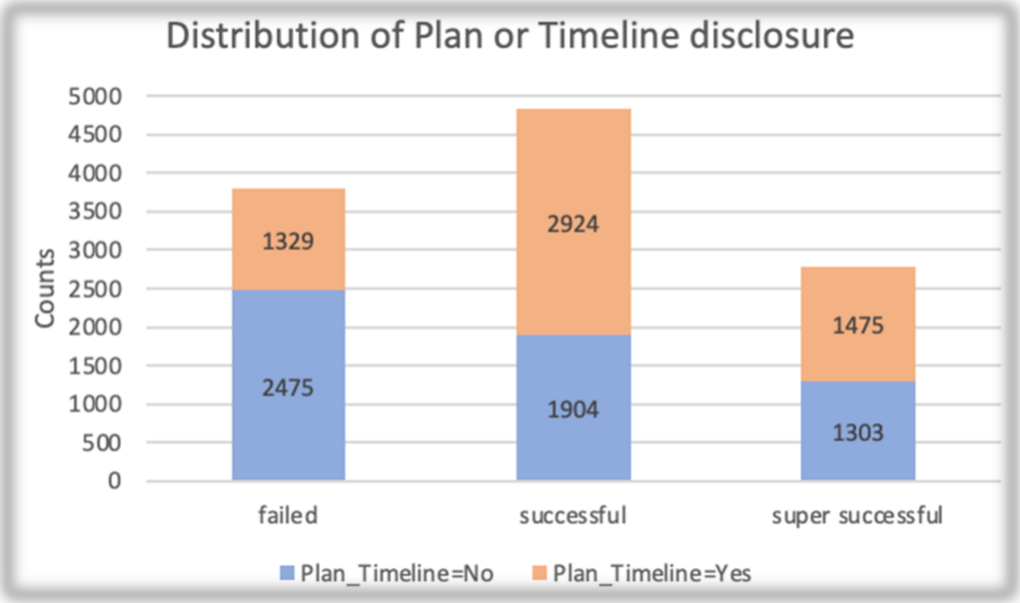


Figure 3.8: Distribution of plan or timeline disclosure in campaigns per class label.

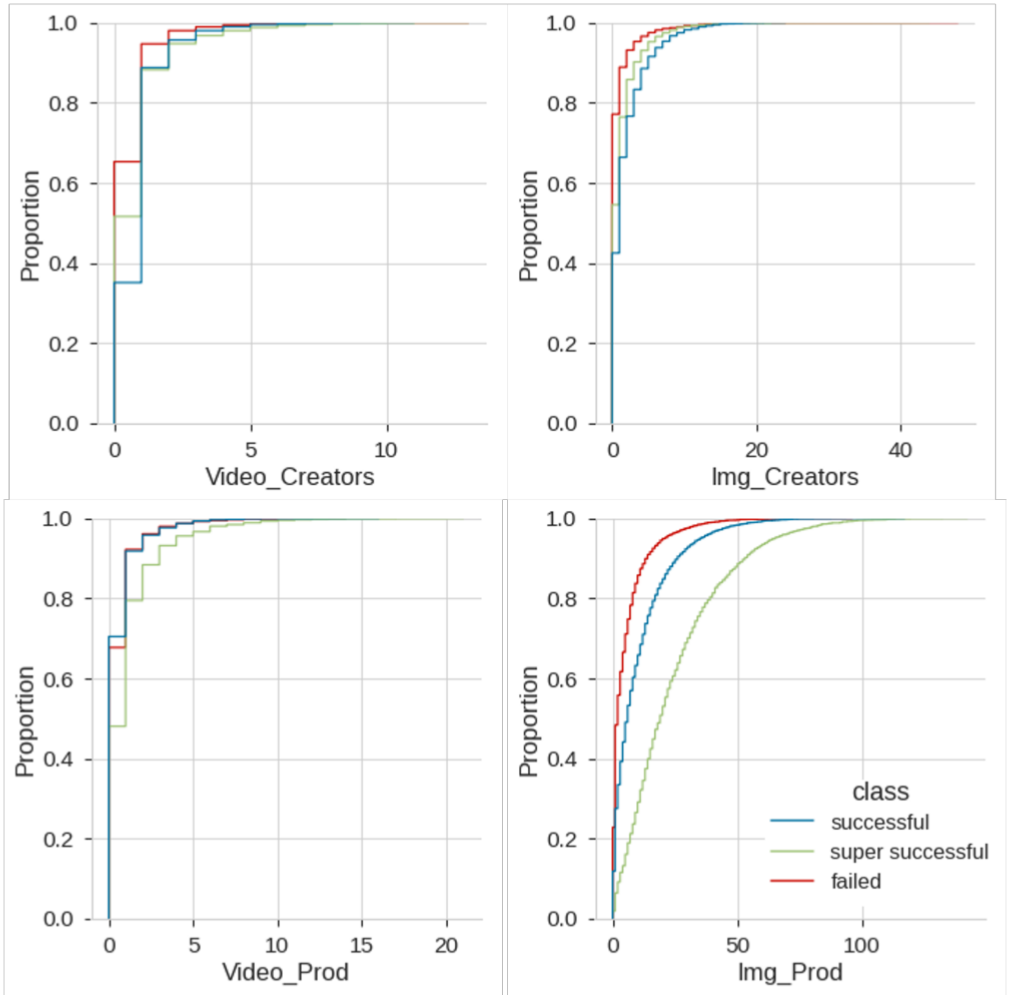


Figure 3.9: Empirical cumulative distribution of videos and images for creators and products per each class label.

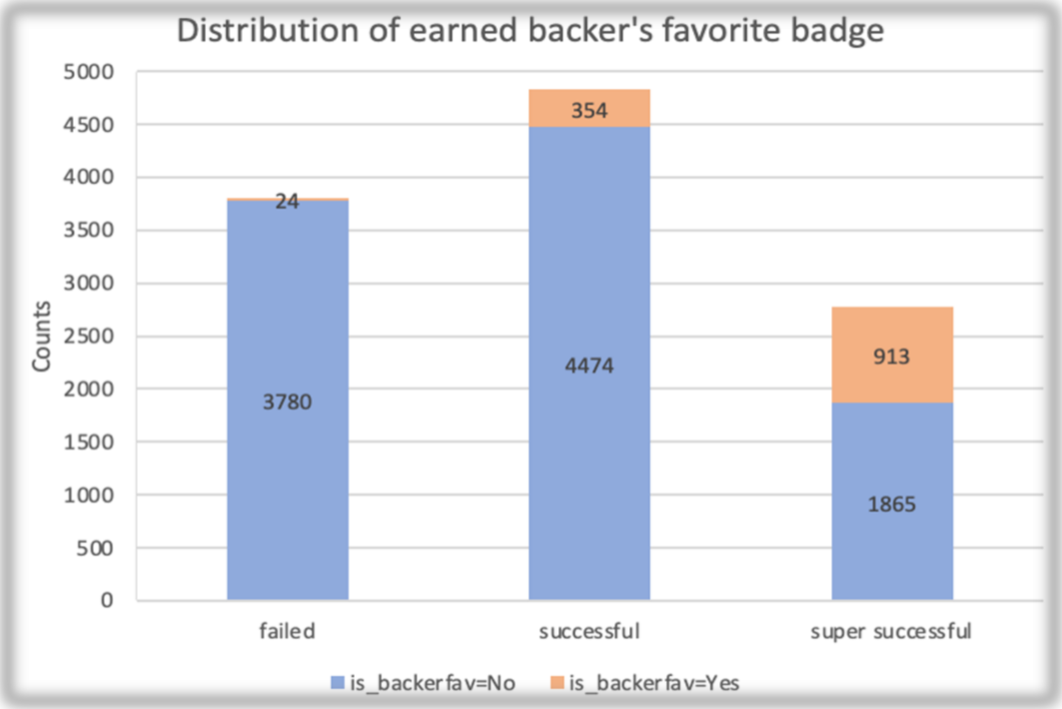


Figure 3.10: Distribution of project creators who earned the backer's favorite badge per class label.

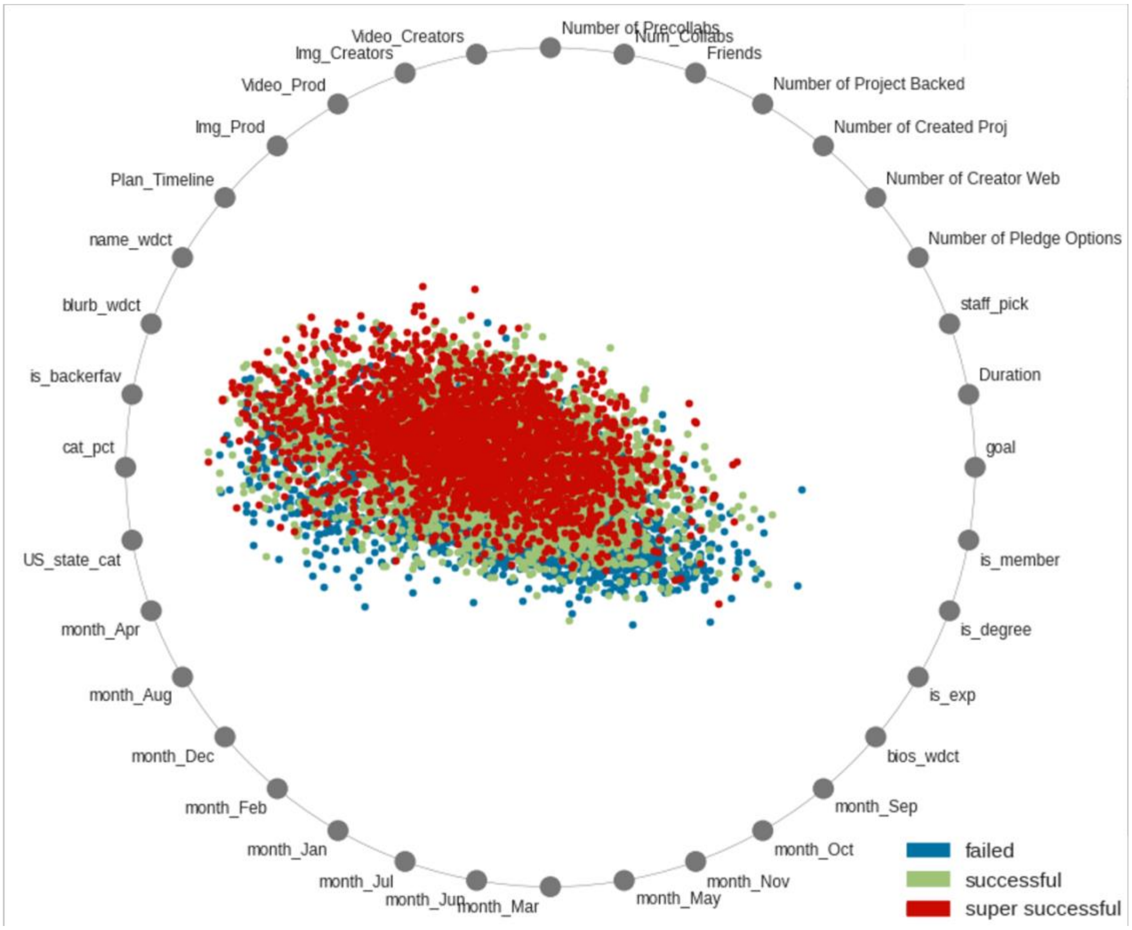


Figure 3.11: RadViz plot of attributes where no distinguishable separability showed.

3.5.2 Results from Supervised Machine Learning Models Evaluation

Performance on testing data set was compared after 12 different algorithms were trained on SMOTE balanced dataset with 5-fold cross validation. To compare across algorithms, parameters setting was standardized broad-based by using the default settings of Python scikit-learn library; in addition, settings for tree-based ensemble methods are standardized for n_estimators=300 and max_depth=3. With a GridSearch for the best estimators for SVM, C=5 and Gamma=0.01 with radial basis function is chosen. The F-score comparison on classification performance is presented in Figure 3.12 with bar plot. Both extreme Gradient Boosting and Light Gradient Boosting have comparable high F-score by a difference of 0.26%. Table 3.2 shows the performance summary on test set and the trained model performance from the 5-fold cross validation. Overall, Extreme Gradient Boosting yields the best classification performance among all algorithms, with accuracy of 72.83% and F-score of 72.41%, while k-Nearest Neighbors has the lowest accuracy of 55.97% and F-score of 55.91%.

Table 3.2: Performance summary of 12 algorithms and their trained 5-fold cross validation results, sorted by accuracy.

Algorithms	Accuracy	F-Score	Precision	Recall	5-fold CV F-Score	5-fold CV F-Score SD
XGBoosting	0.7283	0.7241	0.7277	0.7212	0.7606	0.0419
LightGB	0.7263	0.7215	0.7253	0.7184	0.7600	0.0430
SVM	0.6804	0.6800	0.6814	0.6804	0.7201	0.0409
Logistic Regression	0.6675	0.6675	0.6651	0.6750	0.6894	0.0099
Bagging	0.6670	0.6670	0.6646	0.6745	0.6900	0.0092
Ada Boost	0.6667	0.6611	0.6669	0.6568	0.7225	0.0697
Random Forest	0.6512	0.6515	0.6474	0.6686	0.6839	0.0121
MLP	0.6398	0.6359	0.6373	0.6345	0.7063	0.0497
Decision Tree	0.6243	0.6190	0.6526	0.6098	0.6445	0.0254
QDA	0.5682	0.5611	0.5898	0.5876	0.5668	0.0072
Naïve Bayes	0.5668	0.5597	0.5883	0.5910	0.5751	0.0094
k-Nearest Neighbors	0.5597	0.5591	0.5604	0.5637	0.6373	0.0368

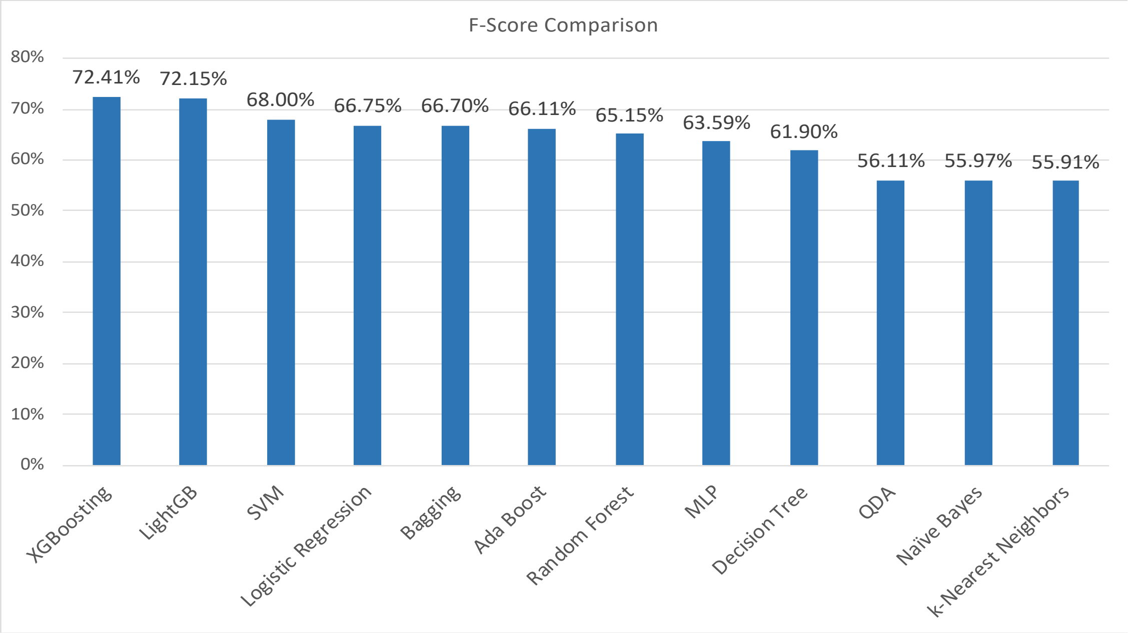


Figure 3.12: Bar plot of 12 algorithms by sorted F-Score performance.

Confusion matrix and classification report for each class label by Extreme Gradient Boosting algorithm is detailed in Table 3.3. The failed class has the highest recall and precision while the super successful class has the least recall and precision. Figure 3.13 identifies the most important features for classification from using Extreme Gradient Boosting, by ranking the calculated gain where it focuses on the average loss reduction in impurity during the feature splitting process. The number of projects backed by creators is the most important feature for Extreme Gradient Boosting tree model while the encoded variables of month when project created are the least important feature.

Table 3.3: Confusion matrix and classification report summary of Extreme Gradient Boosting algorithm.

		Predicted* Class			Classification Report			
	Total (n=3424)	failed	successful	super successful	precision	recall	f1-score	Total Observations
True Class	failed	904	202	35	0.82	0.79	0.80	1141
	successful	173	1051	225	0.69	0.73	0.71	1449
	super successful	30	265	538	0.67	0.65	0.66	833

*Term "Predicted" is used as a generalization of all projects from machine learning classification model, not for a specific project.

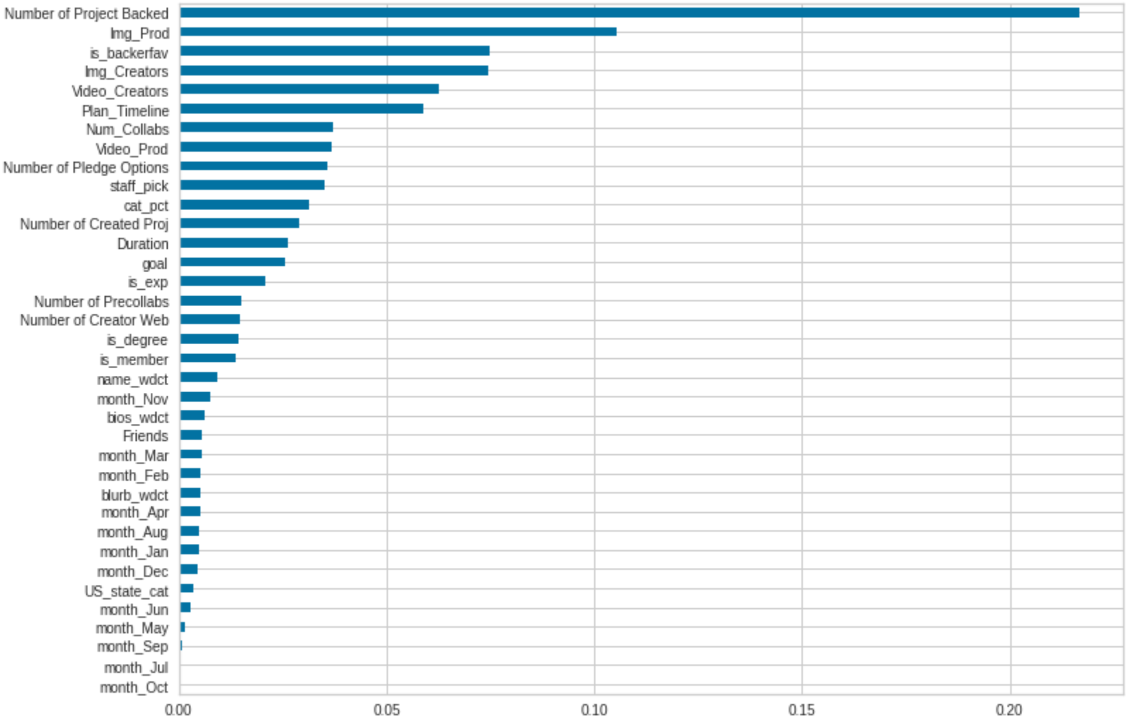


Figure 3.13: Bar plot of feature importance with XGBoost algorithm.

3.5.3 Results from Swarm Based Model Evaluation

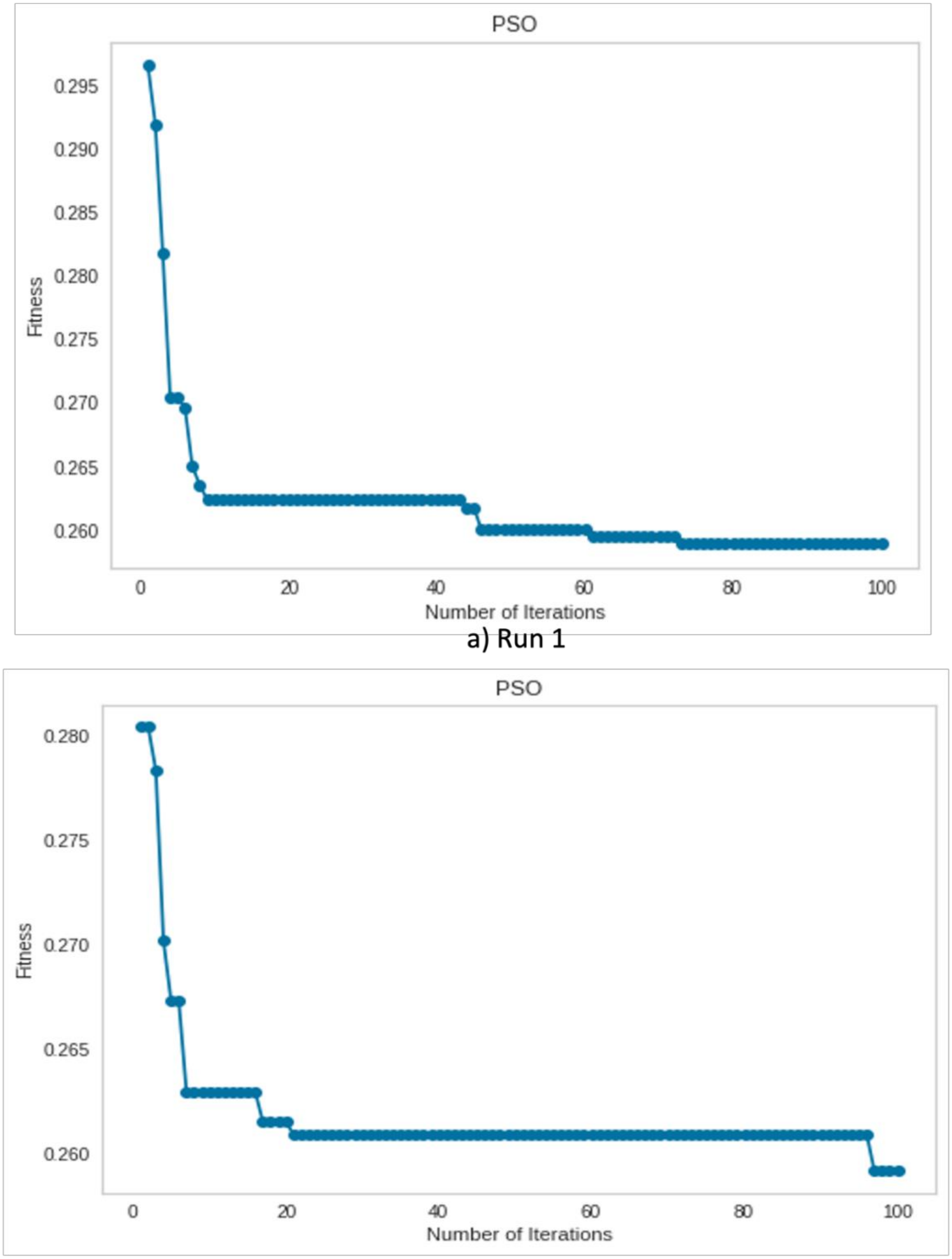
Five runs of Binary Particle Swarm Optimization for feature selection were conducted. The selected features from each run were evaluated. From previous results, Extreme Gradient Boosting is the chosen classifier to evaluate classification performance and to calculate the fitness value. Each run contains 100 iterations, where each iteration computes the fitness value as described in Equation (3.27) using the classification accuracy of training data set with the selected number of features based on the updated particle velocity and its position as described in Equation (3.30). At the end of each run, the best subset of features with the lowest fitness value that is being minimized should be the final chosen features that yields the highest classification power among 100 records of selected features from all 100 iterations.

Table 3.4 shows the compiled results from all five runs using BPSO for feature selection. The last run has the highest accuracy of 74.61% and F-score of 74.64% among all five runs, while the second run has the lowest accuracy of 74.15% and the third run has the lowest F-score of 73.64%. Overall, the mean accuracy from all five runs is 74.47% with 74.28% in F-Score. The mean number of selected features is 23; the average computation time is 200.048 minutes (3 hours, 20 minutes, and 3 seconds). Figure 3.14 through Figure 3.16 display the convergence curve from 100 iterations of each run. The last run appears to have the earliest display of convergence in comparison. Table 3.5 provides a decomposition on selected features from each run. Excluding the ratio encoded project category variable, 8 variables were selected for all 5 runs. They were: project goal amount, project duration, the number of images of creators and products, earned status of backer’s favorite, the number of project collaborators, the number of other projects backed by creators, and the number of video of creators. The word counts of project blurb was not picked from any run. One noteworthy observation is the number of videos of products was only picked once at the first run.

Last, I established a search space for a handful hyperparameters in Extreme Gradient Boosting classifier, such as learning rate, max_depth, min_child_weight, subsample and colsample_bytree, for tuning. However, no significant improvement in accuracy nor F-score was observed (less than 0.1% variance). Therefore, the current parameters used in Extreme Gradient Boosting classifier were sufficient to be considered and used in BPSO optimization. Based on the final model, the results were able to address my assumption that feature selection improved the classification performance by a smidgen of uptick between 1.32% to 1.78% in accuracy from the accuracy of 72.83% without feature selection.

Table 3.4: Summary of classification performance using BPSO selected features for five runs of optimization.

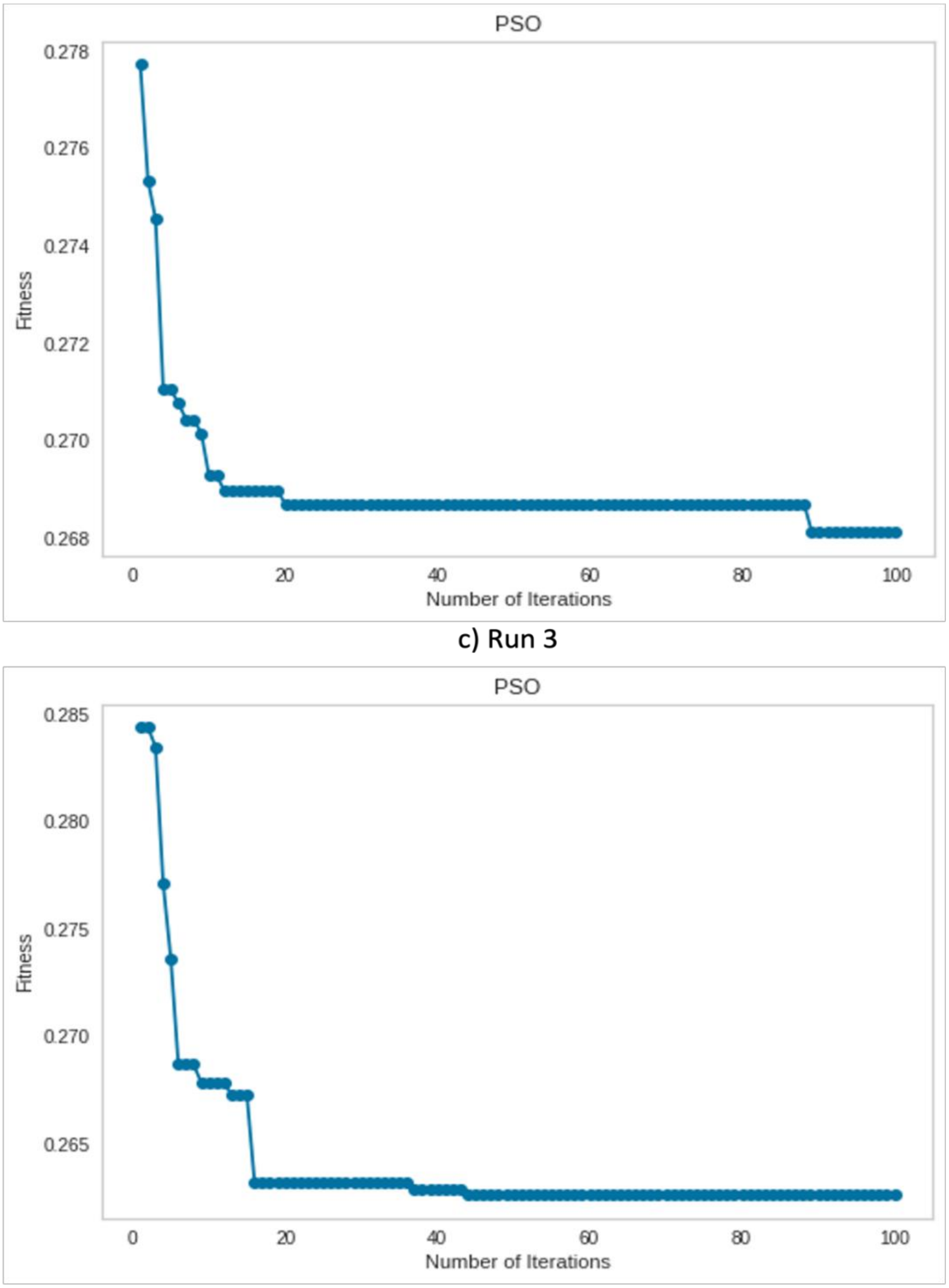
Run	Accuracy	F-Score	Precision	Recall	Number of selected features	Total Run Time (minutes)
1	0.7447	0.7447	0.7464	0.7447	22	195.07
2	0.7415	0.7413	0.7438	0.7415	22	208
3	0.7459	0.7364	0.7386	0.7359	24	199.49
4	0.7453	0.7454	0.7477	0.7453	24	197.37
5	0.7461	0.7464	0.7484	0.7461	24	200.31
Mean	0.7447	0.7428	0.7450	0.7427	23	200.05
SD	0.0019	0.0041	0.0040	0.0042	1.0954	4.89



a) Run 1

b) Run 2

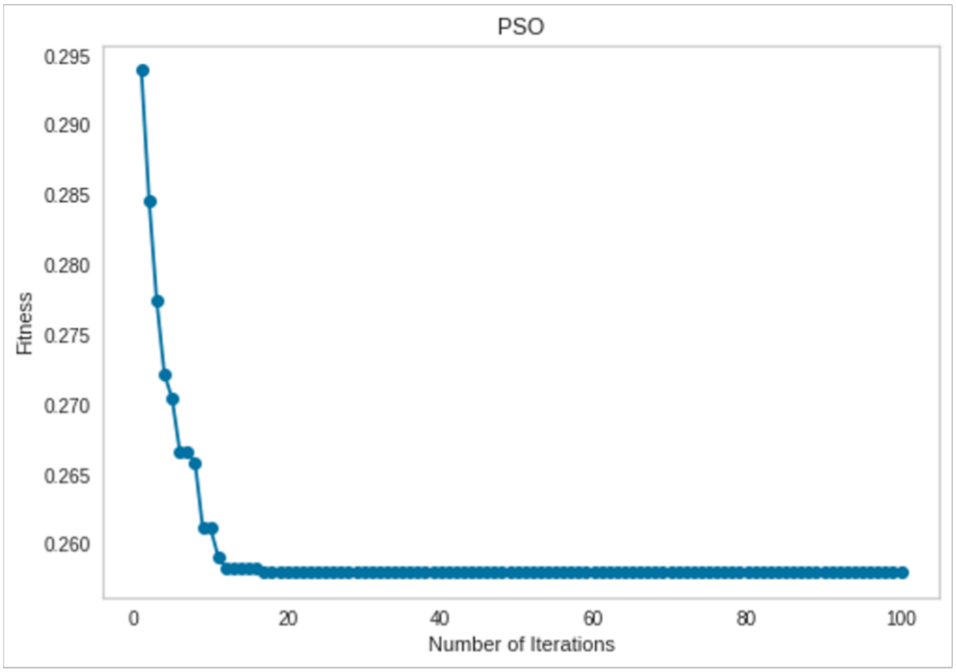
Figure 3.14: Fitness optimization with 100 iterations on first and second runs using XGB-BPSO.



c) Run 3

d) Run 4

Figure 3.15: Fitness optimization with 100 iterations on third and fourth runs using XGB-BPSO.



e) Run 5

Figure 3.16: Fitness optimization with 100 iterations on fifth run using XGB-BPSO.

Table 3.5: Frequency summary of selected features from five runs of BPSO.

Variable	Run 1	Run 2	Run 3	Run 4	Run 5	Selected Frequency
Cat_pct	x	x	x	x	x	5
Duration	x	x	x	x	x	5
Img_creators	x	x	x	x	x	5
Img_prod	x	x	x	x	x	5
Is_backerfav	x	x	x	x	x	5
Num_collabs	x	x	x	x	x	5
Number of Project Backed	x	x	x	x	x	5
Video_creators	x	x	x	x	x	5
Goal	x	x	x	x	x	5
Bios_wdct		x	x	x	x	4
Is_exp	x	x	x		x	4
Is_member	x	x		x	x	4
Month_Mar		x	x	x	x	4
Month_May	x	x	x	x		4
Number of Created Proj		x	x	x	x	4
Number of Pledge Options	x		x	x	x	4
Plan_timeline		x	x	x	x	4
Staff_pick	x		x	x	x	4
Month_Nov	x		x		x	3
Month_Sep	x	x	x			3
Number of Precollabs	x	x			x	3
US_state_cat	x		x		x	3
Friends		x	x			2
Is_degree	x			x		2
Month_Apr				x	x	2
Month_Aug	x	x				2
Month_Dec				x	x	2
Month_Feb		x		x		2
Month_Jul		x	x			2
Number of Creator Web				x	x	2
Month_Jan				x		1
Month_Jun				x		1
Month_Oct					x	1
Name_wdct	x					1
Video_prod	x					1
Blurb_wdct						0
Total selected features	**22**	**22**	**22**	**24**	**24**	

3.6 Discussion and Conclusion

Hierarchical multiple regression and hierarchical ordinal logistic regression were implemented in last chapter to identify key indicators towards funding success. However, there was no telling how those indicators correlated to different levels of success. By categorizing three levels of funding success (“failed”, “successful”, and “super successful”) in projects, the findings from the exploratory analysis provide evidence to characterize success levels in projects as second level indicators. In this chapter, I report my exploratory analysis followed by the classification performance comparison using 12 algorithms. Last, I used a Binary Particle Swarm Optimization to perform feature selection for modeling optimization.

Super successful and successful projects have a more realistic goal amounts and reasonable duration for campaign, more pledge options available to attract backers, and projects have other websites to redirect backers for further information. However, there is no suggestive evidence that the higher the magnitude of these features, the higher likelihood in fundraising success. As for super successful projects, they have high tendency to be proposed by creators who have earned the badge of backer’s favorite, and they are more willing to disclose upcoming plans and project timelines in their campaigns. Moreover, creators behind super successful projects are actively involved in the Kickstarter community by having prior experience in creating other projects, backing other projects, and have number of collaborators. Last, successful projects tend to include more videos and pictures of creators on average than failed projects, but an increasing number of media postings of creators does not necessary bring projects to raise more funds, not twice the pledge amount on average. The findings suggested that an increasing number of videos and pictures of products used in campaigns tend to show a positive association to fundraising outcomes which result as more successful. These observations are consistent with the findings in last chapter as well.

For classification modeling on three different project success levels, various features with feature engineering and the adoption of different algorithms were applied. According to the comparison of twelve adopted algorithms, XGBoost has the highest classification power with accuracy of 72.83% and F-score of 72.41% in funding success level, followed by LightGBM that almost has similar performance. This finding seems to agree with the results from Hu and Yan [54] when classifying binary class of funding success at launch time, where their results also demonstrate that XGBoost has the best performing classification power with 69.80% of accuracy among others. The top three feature importance from XGBoost are the number of projects backed by creators, the number of images for products, and the earned badge of backer's favorite statue. These features are also align with the observation from exploratory analysis, where they exhibit positive association towards extremely successful projects.

Subsequently, five runs of experiments by using Binary Particle Swarm Optimization in feature selection to optimize XGBoost model was performed. The concluded findings demonstrate that using the selected feature subsets from XGBoost-BPSO, the classification performance was able to increase a smidgen of uptick by 1.32% to 1.78% of accuracy in comparison of no feature selection was performed. This result is somewhat in line with the study of using BPSO for feature selection from Geng et al. [35]

to classify crowdfunding project outcome, where their findings also demonstrated that there is an increase in accuracy of 1.52% to 2.61% through logistic regression. Although the BPSO yields minimal classification improvement, the number of features is significantly reduced from 37 to 22 or 24 features with comparable classification performance. Having dimension reduction as an outcome is certainly an advantage in computational modeling. Nonetheless, the average computation time for XGBoost-BPSO is quite costly using over 3 hours of GPU (Tesla V100-SXM2-16GB) run time. The high computation cost is also a major source of limitation.

Overall, the findings from this chapter serve as a novel discovery on the characteristics of highly successful projects. Also, reasonable performance results were achieved by using XGBoost for multiclass classification in project success, instead of limiting to binary class as most existing studies. Moreover, to the best of my knowledge, this research is the first to apply the XGBoost integrated Binary Particle Swarm Optimization for feature selection in the context of Kickstarter project success classification.

3.7 Limitations and Future Research

Several limitations were presented in this study. Since my multiclass classification model is the first to explore and characterize extremely successful projects together with failed and successful projects, there are no known report available to validate the results nor to compare with my model's performance. However, the findings in this study can provide a basis for future research to rectify and expand on using multiclass classification with other algorithms to characterize other dependent features, in order to broaden the subject of Kickstarter campaign strategies.

High computational costs are another major limitation, especially when it comes to generating number of simulations in using BPSO for feature selection. For future research, other classifiers should be integrated with BPSO to further investigate the model improvement in the context of crowdfunding. Adopting different metaheuristic optimization techniques in feature selection should also be explored for comparison with the present findings, such as the Grey Wolf Optimizer and the Whale Optimization Algorithm.

4. COMBINING TEXTUAL INFORMATION IN PROJECT SUCCESS CLASSIFICATION WITH DEEP LEARNING

4.1 Introduction

Textual information in campaigns provides pertinent cues to investors for project objectives and project deliverables. As learned in Chapter 3, with numerical and categorical features, there are distinguishable characteristics among multiclass of project success. However, textual information should also be explored and implemented with the computational model since the campaign is judged by human interpretation and not machine. Therefore, by leveraging semantic strategies in textual information, creators can attract or influence investors to contribute to the project in addition to the other key indicators as identified in previous chapters. In this research, text mining analytics are used on project title, project's pitch (or blurb), and creator's biography as the initial exploratory analysis. This semantic approach is chosen to detect and evaluate the polarity and subjectivity of project title, project's pitch, and creator's biography for each project. I performed the linguistic assessment to understand the effectiveness of the undercurrent sentiment and its impacts on the fundraising outcomes.

In addition, sentiment features from semantic analysis can be used for text classification problem, and this implementation of using textual features was commonly adopted with traditional machine learning until the rise of Natural Language Processing (NLP) with deep learning emerged in recent years. With the growing appeal of using deep neural networks (DNNs) with linguistic information, many works have studied using textual features for project success classification, such as project's pitch. However, only few studies have integrated both textual information and meta information of the campaign to fully exploit different types of features. Therefore, I adopted Natural Language Processing (NLP) as part of the computational approach to ingest and analyze the linguistic information from the crowdfunding campaigns to imitate a more humanlike assessment on project information. Subsequently, I combined textual representation features with categorical and numeric features in the fully connected dense layers as part of the deep neural networks to perform multiclass classification for project success. Text of project's pitch and creator's biography are extracted and preprocessed to fuse with meta data[1] for the multimodal deep learning.

The benefit of using the joined framework is that it can capture the low-level interaction among multiple types of features, and most importantly, it can characterize different aspects of a campaign by taking advantages of using more than a single modality of features for performance improvement. With this aim in mind, the proposed framework is to demonstrate the feasibility of joining textual features and meta features that yield favorable classification performance, without compromising other aspect of information in campaigns. Furthermore, this dissertation serves as a novel finding to adopt creator's biography as part of the textual features in multiclass classification.

[1] All numeric and categorical features that were used in Chapter 3 for Extreme Gradient Boosting

4.2 Related Work

4.2.1 Text Analytics with Traditional Machine Learning Models Literature Reviews

In order to obtain more valuable information instead of relying solely on the meta data for project outcome prediction, researchers have turned to text mining and text analytics to discover new patterns in linguistic information that were not detected by using meta data alone. As early as 2014, only a handful of researchers had investigated the phrases used in Kickstarter projects. The work of Mitra and Gilbert [55] is one of the very first to look at the persuasion principles of phrases used in project's pitch to confirm the interconnection between language used and social behavior in crowdfunding. By employing both meta attributes and project's pitch in the penalized logistic regression for classification, their results showed positive binary classification performance where error rate dropped from 17.03% to 2.4% [55] from the baseline model. Du et al. [56] conducted similar study on project description with Elaboration Likelihood Model (ELM) to assess investor's contribution decisions and its influence on project success, by adopting the Gunning fog index [56] and the number of words in project description. The empirical logistic results showed significant prediction performance around 71% of accuracy as "evidence on the influence of project descriptions on funding success".

Sawhney et al. [57] reported using primary and secondary linguistic features of project title and project's pitch, such as Flesch-Kincaid Readability score and topics from Latent Dirichlet Allocation, to make prediction using a support vector machine (SVM) that yields 71% accuracy. Other studies like Wang et al. [58] implemented both machine learning (Conditional Random Field and Support Vector Machine) and lexical-based methods (association rule mining) to investigate title, blurb, the first 100 words, and detailed description of a campaign from Kickstarter data, where SVM with Part Of Speech (POS) tagging yield the highest precision and recall in terms of sentiment polarity. In their model, by adding sentiment variables to meta data, they concluded that not only positive sentiment in blurb and description was able to attract investors, but also the classification accuracy raised by 7.3% in improvement, while there was no significant impact in title.

Later on, Silva et al. [59] suggested a two-phase modeling with SVM and regression models where sentiment variables were used to predict funded amount before launching and during campaigning; the test results achieved around 71% and 85% respectively. Other studies, such as Faralli et al. [60], have reported leveraging emotional aspect of text sentiment for prediction in the domain of mobile games projects, with the advantage of fewer variables required by this framework.

4.2.2 Text Analytics with Deep Learning Models Literature Reviews

The rapid growth and appeals for deep learning has been dominating the domain of linguistic computation in recent years. Given the tremendous popularity of using deep learning with natural language processing, researchers have driven the further development of linguistic signaling in crowdfunding prediction with neural networks. In late 2018, Lee et al. [61] modeled text mining on Kickstarter data through sequence to sequence (seq2seq) deep neural network (DNN) and Hierarchical Attention-based Network (HAN) to predict crowdfunding success at sentence level attention for technology category projects. By using Gated Recurrent Units with GloVe word embedding on text data from campaign in the Attention Network, text model of combining updates, comments and speech sections of a campaign could yield up to 91% in predictive accuracy [61]. Katsamakas and Sun [62] also examined project description data with sentiment analysis, and the textual representation was trained on recurrent neural networks (RNNs) and Long Short -Term Memory (LSTM) which achieved 75.57% binary classification accuracy. Recently, Zhou [63] proposed a sequence-enhanced capsule network (CapsNet) model to classify project success where both sequence semantic information and spatial location information were factored in the model. Using project's pitch, the proposed Bidirectional Gated Recurrent Units (BiGRU) and CapsNet connected model was able to achieve 87% accuracy [63].

Despite a considerable body of literature on crowdfunding prediction, previous studies have almost exclusively focused on using either meta information or textual information of the project. Only few studies have investigated employing more than one type of information, or adopting a multimodal approach, for prediction. In 2016, study by Kim et al. [64] applied Google Speech API to extract text from speech of project video at the of project page, then utilized IBM Watson Tone Analyzer to extract emotion, writing styles and social propensities on text and speech narratives with Kickstarter project's meta information. Their results suggested that both speech and text are strong predictors in project prediction [64]. In addition, Raab et al. [65] also extracted emotion factor using facial expression through project pictures to assess influence on project success. Thereafter, more studies have also employed a multimodal approach to include signaling features or cues in prediction.

In late 2019, Cheng et al. [66] suggested a multimodal deep learning (MDL) framework where the extracted textual description, visual image, and meta information of the Kickstarter campaigns were used for training through three different branches and "the feature maps from three streams are concatenated into one feature map" through convolution neural network (CNN) or connected hidden layer in neural network. Their trained model attained a 83% accuracy in prediction when all modalities were joined [66]. Similarly, in the recent study of Kaminski and Hopp [67], proposed a broader approach to join all features of description, speech, and video data of technology and product design categories. Two paragraph vectors, Distributed Bag-of-Words (PV-DBOW) and Distributed Memory (PV-DM), were used as the representation models to first train through six different classifiers for description, speech, and video data individually, before combining all in one matrix to train with Logistic Regression at the end. Their results found that the combined Logistic Regression generated the highest model performance with all three signals to mimic human perception of information in decision making [67].

To note, only few works have examined using multimodal information in crowdfunding prediction and many aspects of textual information have not been investigated; therefore, my dissertation utilizes both project's pitch and creator's biography with project meta information in a deep learn architecture for project classification.

4.3 Data

Two forms of information were used for the multiclass project success classification with deep learning in this chapter, meta information and textual information. For meta information, the processed data was described and extended from Section 3.3 where 36 engineered features remained for the 11,410 campaigns in the final sample. Data definition can be found in in Appendix A.1. In addition, the selected set of 24 features from Binary Particle Swarm Optimization was also used for performance fine-tuning.

For textual information, project's title, project's pitch and creator's biography were processed for exploratory analysis and each underwent a series of text preprocessing steps as follow:

1. Converted all characters to lowercase form.
2. Removed URLs and HTML tags.
3. Expanded contractions. For example, "I'll" expanded as "I will".
4. Expanded chat words. For example, "LOL" expanded as "Laugh out loud".
5. Removed Emoji and Emoticons.
6. Removed numbers, punctuations, accented characters.
7. Removed stop words with NLTK stop words corpus (http://www.nltk.org/nltk_data/).
8. Removed the overall top 10 most frequent words of project's pitch, creator's biography, and project's title. Table 4.1 through Table 4.3

provides a list of top 10 most frequent words in each text attribute by class.

9. Applied stemming to reduce words in their root form for text normalization. For example, “researches”, “researching”, “researched”, and “researcher” are all reduced to “research”.

After text processing, projects with null project’s pitch or null creator’s biography were removed, and 11,349 projects remained in the final dataset.

Table 4.1: Top 10 most frequent words in project’s pitch.

Overall		**Failed**		**Successful**		**Super Successful**	
word	**counts**	**word**	**counts**	**word**	**counts**	**word**	**counts**
new	1020	app	251	new	514	new	270
book	743	help	241	album	459	game	205
help	695	new	236	book	422	book	198
album	632	film	181	help	355	first	106
game	491	food	157	music	268	art	104
film	473	music	153	film	253	edition	103
music	457	creating	140	first	221	deck	100
world	416	series	137	us	201	world	99
first	410	create	135	world	192	help	99
series	407	people	134	series	175	card	98

Table 4.2: Top 10 most frequent words in creator's biography.

Overall		Failed		Successful		Super Successful	
word	**counts**	**word**	**counts**	**word**	**counts**	**word**	**counts**
years	2702	years	1097	music	1486	games	695
new	2338	love	629	new	1252	years	616
music	2252	life	596	years	989	new	512
love	1767	new	574	work	801	design	506
work	1680	business	573	film	796	art	495
art	1617	music	523	love	758	products	479
world	1601	world	513	art	730	work	469
also	1548	time	504	also	696	game	459
life	1468	make	453	world	672	kickstarter	427
artist	1396	also	452	artist	641	world	416

Table 4.3: Top 10 most frequent words in project's title.

Overall		Failed		Successful		Super Successful	
word	**counts**	**word**	**counts**	**word**	**counts**	**word**	**counts**
album	497	app	142	album	382	book	110
new	447	film	78	new	292	edition	80
book	360	new	77	book	195	worlds	78
film	283	album	71	film	181	cards	78
game	212	project	69	debut	137	new	78
cards	190	music	68	short	118	game	75
art	180	food	65	ep	102	first	65
music	179	game	60	music	99	playing	65
project	175	cards	60	project	88	enamel	57
debut	171	art	58	art	77	pins	50

4.4 Methods

4.4.1 Overview of Research Design

To combine textual information for classification, the experimental design contains three parts. First, I started off with text and sentiment analysis for exploratory study to gain insight into the textual information; second, I constructed different deep neural networks (DNNs) with multiple combination of features (project's blurb with meta data, creator's biography with meta data, or both text attributes with meta data) to classify project success for comparison. Once the best performing deep neural network was identified, hyperparameter tuning was done to optimize classification performance.

- Text Mining and Sentiment Analysis:

  To begin understanding all three sets of textual information (project's blurb, creator's biography, and project's title), lexicon features (polarity and subjectivity), length features (difficult word counts, syllable counts, and word counts), and vocabulary features (Flesch Reading Ease Formula and Gunning Fog Index) were assessed in the text mining and sentiment analysis. Sentiment analysis, also known as opinion mining, "is the computational study of people's opinions, attitudes and emotions toward an entity" [68]. Polarity and subjectivity are sentiment lexicon features which can be computed using Textblob, a python library that provides an API to perform textual processing tasks. Polarity scores lie within the range of [-1.0, 1.0] where 1.0 represents the sentiment of a sentence is very positive and -1.0 represents the sentiment of a sentence is very negative. Subjectivity scores lie within the range of [0.0, 1.0] where 0.0 represents the opinion of a sentence is very objective and 1.0 represents the opinion of a sentence is very subjective.

  For length features and vocabulary features, Textstat, another python library was used to compute readability, complexity and grade level statistics.

Flesch-Kincaid Reading Ease formula in Equation (4.1) analyzes the U.S grade level as a general readability measurement "from the average sentence length and the average syllable number of words in the text" [69].

$$206.835 - (1.015 \times \ ASL) - \left(84.6 \times \frac{n_{sy}}{n_w}\right) \tag{4.1}$$

where ASL is the average sentence length, n_{sy} is the number of syllables, and n_w is the number of words. Gunning's Fog Index "calculates the proportion of difficult words in the text" [69] to assess the readability and grade level, but more focus on clarity and simplicity for business writing. The formula in Equation (4.2) shows Gunning's Fog index concentrates on words with 3 syllables or more.

$$0.4 \times (ASL + 100 \times \frac{n_{wsy\geq 3}}{n_w}) \tag{4.2}$$

where ASL is the average sentence length, $n_{wsy\geq 3}$ is the number of words with 3 syllables or more, and n_w is the number of words.

- Deep Learning Implementation:
  1) Splitting Training & Testing Data:

     To prepare for deep learning, the pre-processed dataset with 11,349 observations were split by 70:30 ratio into training and testing data set, while 10% of the training data set was used for validation. Since the data was skewed with imbalance class as shown in Figure 3.2 in previous chapter, a balance training dataset was created by performing the random oversampling on the minority classes instead of Synthetic Minority

Oversampling Technique (SMOTE) that was used in Chapter 3. Unlike meta data with only numeric and categorical encoding values, text features would not be applicable with SMOTE algorithm where it synthesizes data for data augmentation to treat imbalance data set. While there are other data augmentation techniques for text, such as Generative Adversarial Networks (GANs); however, the current research framework would not include this technique as part of the scope. Data set was also scaled between [0,1] for normalization. The training set contains total of 10,137 observations where each class has 3,379 observations after the random oversampling on the minority classes.

2) Baseline Model:

Naïve Bayes classifier was adopted as a baseline model with Term Frequency or Term-Document Matrix to compare the classification performance using deep neural networks. General information of Naïve Bayes algorithm is detailed in Section 3.4.2. CountVectorizer function of scikit-learn was used to count word occurrence and vectorized text representation following a Bag of Words model. Laplace smoothing is applied (alpha=1) to avoid zero probabilities as depicted in Equation (4.3):

$$\hat{P}\left(x_i \middle| w_j\right) = \frac{\Sigma\, tf\left(x_i, d \in \omega_j\right) + \alpha}{\Sigma N_{d \in \omega_j} + \alpha \cdot \mathcal{V}} \tag{4.3}$$

where x_i is the ith word in the feature vector, $\Sigma\, tf\big(x_i, d \in w_j\big)$ is the sum of term frequencies for each word in the feature vector from all documents that belongs to class ω_j for training the Naïve Bayes model. $\Sigma N_{d\in\omega_j}$ is the sum of all term frequencies for all documents in class ω_j where Laplace smoothing parameter and the size of vocabulary are denoted as α and $\mathcal{V}$.

3) Word Embedding Used:

   Word embedding can reduce sparsity in text representation and compact semantic relationships while boosting computation efficiency. The 300-dimensional Global Vectors for Word Representation (GloVe) was chosen as the pre-trained word embeddings for both text inputs. GloVe is developed by Pennington et al. [70] in 2014 at Stanford University and introduced as a "log bilinear regression model for the unsupervised learning of word representation", where it is used to capture linear substructures among words by aggregating global word co-occurrence statistics from a corpus.

4) Architecture of DNNs using TensorFlow Keras:

   Tesla P100-PCIE-16GB GPU was used to compute the baseline model as well as the four chosen neural network models: long short-term memory (LSTM), gate recurrent units (GRU), convolutional neural network (CNN), and the hybrid of LSTM and CNN. Methodology details are discussed in the subsequent sections. Each of the neural network models was trained on text information individually before joining meta information with fully dense connected layers for final outputs. For multiclass classification, SoftMax in Equation (4.4) was used for the activation function:

$$f(s)_i = \frac{e^{S_i}}{\Sigma_j^C e^{S_j}} \tag{4.4}$$

where S_j are the scores calculated by the net for each class in C, and S_i is the given class for $i = 1, \ldots, k$. For cost function, categorical cross entropy in Equation (4.5) was used:

$$CE \; = -\log\left(\frac{e^{S_p}}{\Sigma_j^C e^{S_j}}\right) \tag{4.5}$$

where S_p is the classifier score for the positive class. Adam optimization was adopted for gradients. I also applied L2 regularization and dropout regularization between TensorFlow layers to drop neurons with probability p>0 to avoid overfitting. Last, metrics used to compare performance is the same as described in Section 3.4.3.

4.4.2 Recurrent Neural Networks (RNNs)

RNNs is a type of neural networks for sequential processing with information that is being transmitted to the next iteration recursively as in Figure 4.1 and Figure 4.2. RNNs have a very similar architecture as the Multi-Layer Perceptron that built with artificial neurons. The basic architecture of neural networks is described in Section 3.4.2. RNNs also comprise of input layers with input units, hidden layers with hidden units, and an activation function for the output layer.

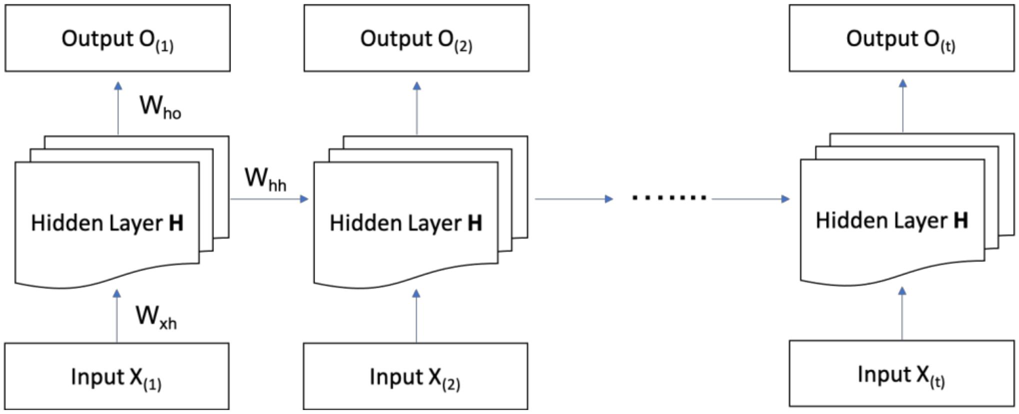


Figure 4.1: Recursive iteration in RNN architecture.

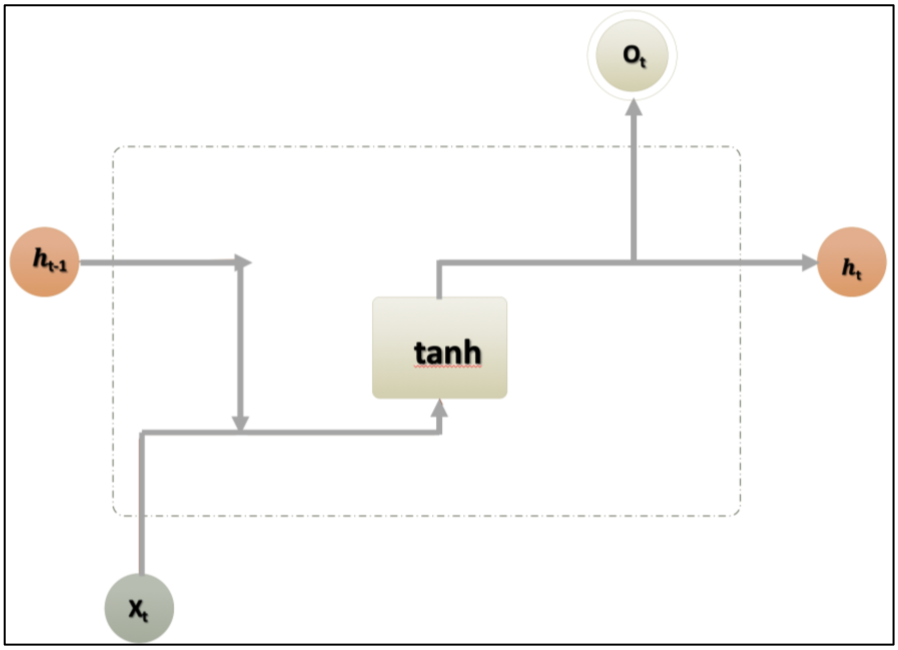


Figure 4.2: An example of basic RNN composition flow diagram with tanh activation function.

In addition, RNNs extend “the functionality of Feedforward Networks to also take into account previous inputs X_{t-1} and not only the current input X_t” [71], where multiple hidden layers are recursively created as one hidden layer block. The hidden variable can be expressed as:

$$H_t = \phi_h(\boldsymbol{X}_t\boldsymbol{W}_{xh} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hh} + b_h) \tag{4.6}$$

where H_t is the hidden state at time t, ϕ_h is the activation function for h as the number of hidden units. $\boldsymbol{W}_{xh}$ is the weight matrix of input to hidden state where $\boldsymbol{W}_{xh} \in \mathbb{R}^{d \times h}$ as d is the number of inputs of each sample. $\boldsymbol{W}_{hh}$ is the weight matrix of hidden state to hidden state where $\boldsymbol{W}_{hh} \in \mathbb{R}^{h \times h}$, and b_h is the bias term for hidden units. The output for variable is described in Equation (4.7):

$$\boldsymbol{O}_t = \phi_o(\boldsymbol{H}_t\boldsymbol{W}_{ho} + b_o) \tag{4.7}$$

To backpropagate through time at timestep t , the partial derivative w.r.t weight matrix $\boldsymbol{W}$ is formulated as follows with loss function:

$$L(\boldsymbol{O},\boldsymbol{Y}) = \sum_{t=1}^{T} \ell_{\mathrm{t}}(\boldsymbol{O}_t,\boldsymbol{Y}_t) \tag{4.8}$$

where $\boldsymbol{O}$ is the output value and $\boldsymbol{Y}$ is the target value; ℓ_{t} is the loss term at time step t for each update and T is the total time steps.

$$\frac{\partial \boldsymbol{L}^{(T)}}{\partial \boldsymbol{W}} = \sum_{t=1}^{T} \frac{\partial \boldsymbol{L}^{(T)}}{\partial \boldsymbol{W}} |_{(t)} \tag{4.9}$$

To further break down the partial derivative in Equation (4.9) for gradients, there are three weight matrices $\boldsymbol{W}_{ho}, \boldsymbol{W}_{hh}, \boldsymbol{W}_{xh}$ through backpropagation as below:

$$\frac{\partial L}{\partial \boldsymbol{W}_{xo}} = \sum_{t=1}^{T} \frac{\partial \ell_{\mathrm{t}}}{\partial \boldsymbol{O}_t} \cdot \frac{\partial \boldsymbol{O}_t}{\partial \phi_0} \cdot \boldsymbol{H}_t \tag{4.10}$$

$$\frac{\partial L}{\partial \boldsymbol{W}_{hh}} = \sum_{t=1}^{T} \frac{\partial \ell_{\mathrm{t}}}{\partial \boldsymbol{O}_t} \cdot \frac{\partial \boldsymbol{O}_t}{\partial \phi_0} \cdot \boldsymbol{W}_{ho} \sum_{t=1}^{T} (\boldsymbol{W}_{hh}^{T})^{t-k} \cdot \boldsymbol{H}_k \tag{4.11}$$

$$\frac{\partial L}{\partial \boldsymbol{W}_{xh}} = \sum_{t=1}^{T} \frac{\partial \ell_{\mathrm{t}}}{\partial \boldsymbol{O}_t} \cdot \frac{\partial \boldsymbol{O}_t}{\partial \phi_0} \cdot \boldsymbol{W}_{ho} \sum_{t=1}^{T} (\boldsymbol{W}_{hh}^T)^{t-k} \cdot \boldsymbol{X}_k \tag{4.12}$$

$(\boldsymbol{W}_{hh}^T)^{t-k}$can be denoted as $\frac{\partial \mathbf{H}_{\mathrm{t}}}{\partial \boldsymbol{H}_k}$. One of the common problem in RNNs is vanishing (or exploding) gradient problem where “if there are small values (< 1) in the matrix multiplication this causes the gradient to decrease with each layer (or time step) and finally vanish” [71] or exploding gradient if large values are in the matrix multiplication from the long sequential information backpropagation through time. Meaning the “memory” of the early inputs in the information will be disappearing as the training of the model goes on, and the information is not being captured properly. Therefore, to mitigate this problem, Horchreiter and Schmidhuber introduced “Long Short-Term Memory” [72] and Cho et al.[73] proposed “Gated Recurrent Units”.

4.4.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory is a form of recurrent neural networks, it is especially beneficial when handling long sequential information to avoid the vanishing or exploding gradient problem as described in RNNs. Moreover, one of the advantageous of LSTM is its update complexity per weight and time step is O(1) for a relatively fast computation speed [72]. There are four main components in LSTM as depicted in Figure 4.3:

1) Forget Gate(F_t) —it controls how much information to be discarded

$$\boldsymbol{F}_t = \sigma(\boldsymbol{X}_t \boldsymbol{W}_{xf} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hf} + b_f) \tag{4.13}$$

2) Input Gate (I_t)—it takes input and to store information

$$\boldsymbol{I_t} = \sigma(\boldsymbol{X}_t \boldsymbol{W}_{xi} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hi} + b_i) \tag{4.14}$$

where $\boldsymbol{I}_t \in [0,1]$ from σ sigmoid activation, and transform input with tanh activation where estimated memory cell state $\widetilde{\boldsymbol{C}_t} \in [-1,1]$ to be carried along in the record as described in Equation (4.14).

$$\widetilde{\boldsymbol{C}_t} = tanh(\boldsymbol{X}_t\boldsymbol{W}_{xc} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hc} + b_c) \tag{4.15}$$

3) Cell State ($\boldsymbol{C}_t$)—it records the overall memory to carry along the sequence processing where memory is being updated in here as well. To update, adding forget gate values with previous cell state to input gate with the estimated cell state to yield the new state for new memory.

$$\boldsymbol{C}_t = \boldsymbol{F}_t * \boldsymbol{C}_{t-1} + \boldsymbol{I}_t * \widetilde{\boldsymbol{C}}_t \tag{4.16}$$

4) Output Gate ($\boldsymbol{O}_t$)—it computes by using input at time t with its weight matrix and the previous hidden state with its weight matrix, where the hidden state at time t is computed with the current output with the updated memory.

$$\boldsymbol{O}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xo} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{ho} + b_o) \tag{4.17}$$

$$\boldsymbol{H}_t = \boldsymbol{O}_t * \tanh(\boldsymbol{C}_t) \tag{4.18}$$

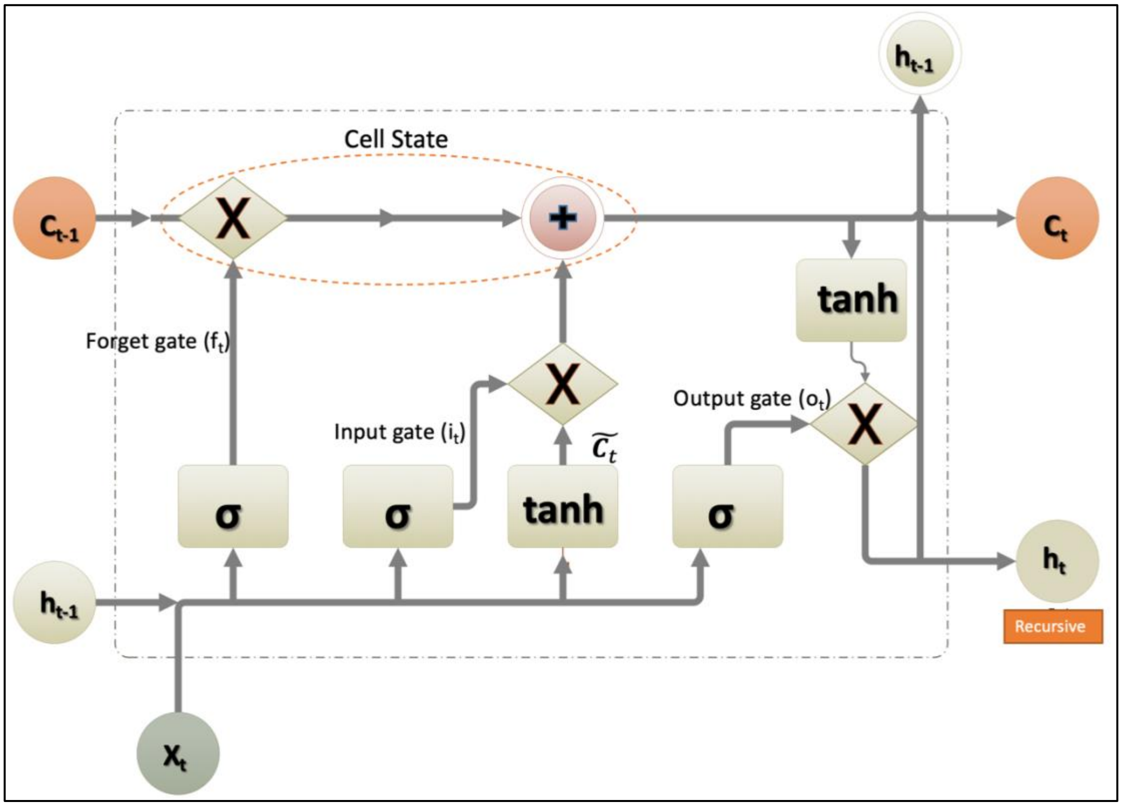


Figure 4.3: Basic LSTM composition flow diagram.

4.4.4 *Gated Recurrent Unit (GRU)*

Gated Recurrent Unit is similar to LSTM but with less gate control, yet another form of recurrent neural networks that can also mitigate the vanishing or exploding gradient problem when training for long sequential process. GRU also has different gates to control memory flow to carry information along the cell processing as shown in Figure 4.4:

1) Reset Gate (R_t) —it controls how much information to be discarded from previous time steps in order to carry along for the future.

$$\boldsymbol{R}_t = \sigma(\boldsymbol{X}_t \boldsymbol{W}_{xr} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hr} + b_r) \tag{4.19}$$

2) Update Gate (Z_t) —Instead of having separate input gate and forget gate, they are combined into one update gate to simplify the architecture.

$$\boldsymbol{Z}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xz} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hz} + b_z) \tag{4.20}$$

To update the memory in hidden state for output:

$$\widetilde{\boldsymbol{H}}_t = tanh(\boldsymbol{X}_t\boldsymbol{W}_{xh} + r_t \odot \boldsymbol{H}_{t-1}\boldsymbol{W}_{hh} + b_h) \tag{4.21}$$

$$\boldsymbol{H}_t = (1 - \boldsymbol{Z}_t) \odot \boldsymbol{H}_t + \boldsymbol{Z}_t \odot \widetilde{\boldsymbol{H}}_t \tag{4.22}$$

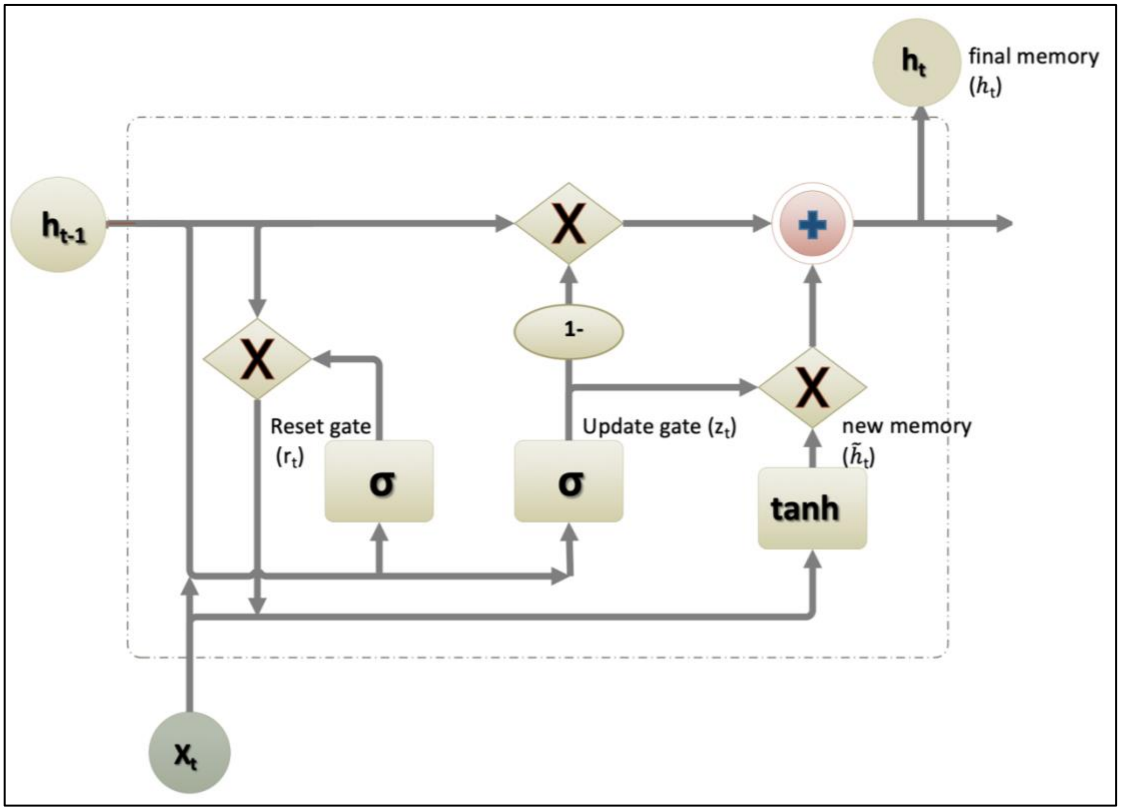


Figure 4.4: Basic GRU composition flow diagram.

4.4.5 Convolutional Neural Networks (CNNs)

Convolutional neural network is a type of neural networks that was first introduced by Yann LeCun and others between 1980s and 1990s. Research on measuring prediction performance for the MNIST handwritten digit recognition from LeCun et al. [74] in 1995 reported that, a Boosted LeNet 4 using boosting ensemble methods which yields the best score after an architecture improvement from LeNet 1, the first convolutional network, among other neural networks. Around the same time, LeCun and Bengio [75] also provided new development in pattern recognition research on images, speech, and time series data using convolutional networks. Since then, attraction from research community in adopting convolutional networks for different applications has expanded outside of image and object detection, but also text classification tasks as well. In my research, a 1-D convolutional network is used for text sequencing and classification as text information is in 1-dimensional space for signals, unlike object detection with multiple arrays.

The basic architecture of CNNs comprised of convolution layer with filters, pooling layer, and fully connected layer [76] to train through input information for text:

1) Convolution layer—This layer contains kernels and filters. Kernel is a window with defined length to slide along the input information where it scans and computes a weighted sum by multiply input elements with a matrix of weights to pass through for the max pooling layer. Unlike a 2-D convolutional network with a 2-D kernel size, a 1-D kernel will only be a 1-D matrix. Each filter takes the vector representation from the view of the kernel

window and transforms into a single feature. The number of filters equals to the desired number of output features.

2) Pooling layer—This layer will conduct a down-sampling from the convolution layer once all features are mapped in the feature map to extract important features and reduce dimensional space. Common types of pooling operations such as extracting maximum value or averaging values of the convolution layer. I selected max pooling for all CNN trainings, where only the maximum value of the vectorized kernel outputs will be retained.

3) Fully connected layer—This layer will flatten the inputs from the joint vectorized outputs in the final pooling layer and pass through to the output layer in CNN using activation function for final outputs.

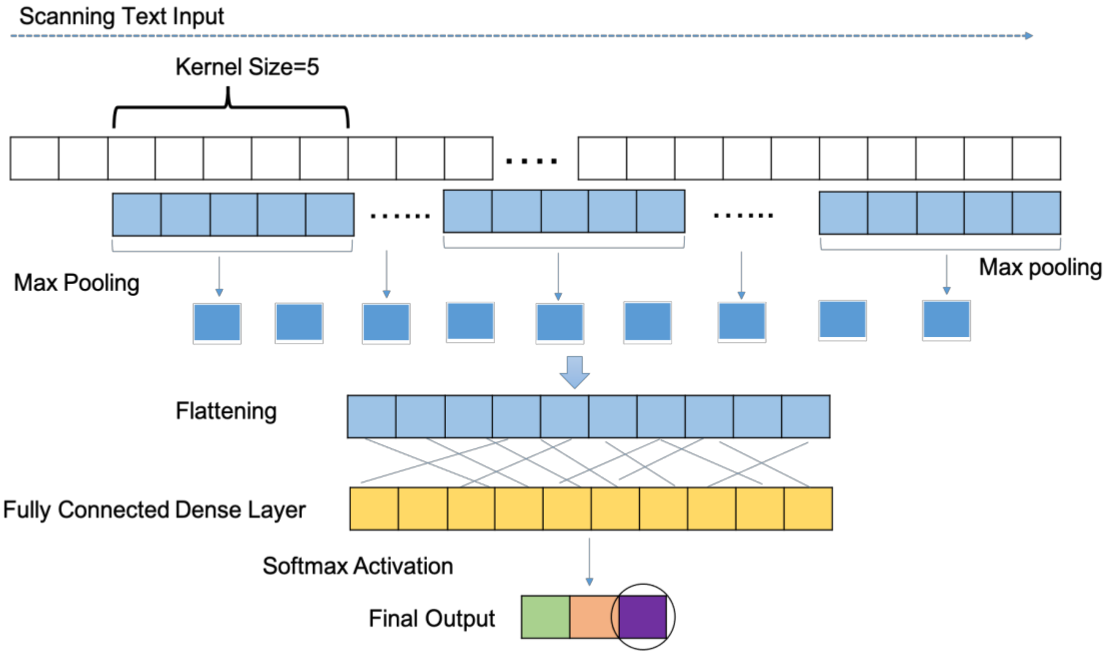


Figure 4.5: 1-D Convolutional neural network flow diagram for text sequential processing.

4.5 Results

4.5.1 Exploratory Sentiment Analysis

To gain insights into the characteristics of three text variables (project's pitch, project's title, and creator's biography), a descriptive summary statistics of different lexicon features (polarity and subjectivity), length features (difficult word counts, syllable counts, and word counts), and vocabulary features (Flesch Reading Ease Formula and Gunning Fog Index) were computed for sentiment analysis and text mining in Table 4.4.

Table 4.4: Summary statistics of text variables by class.

		failed		successful		super successful	
	Text Metrics	Mean	SD	Mean	SD	Mean	SD
Creator's Biography	Flesch Reading Ease	50.22	35.91	47.37	34.73	48.23	33.18
	Gunning Fog	14.14	10.34	14.81	10.06	14.66	10.18
	Polarity	0.16	0.19	0.16	0.18	0.18	0.21
	Subjectivity	0.40	0.24	0.40	0.23	0.44	0.23
	Difficult Words	13.81	16.04	17.27	19.47	16.29	16.80
	Syllable Counts	95.92	121.20	110.68	130.16	105.19	123.66
	Word Counts	64.89	83.16	73.92	87.52	70.58	84.34
Project's Pitch	Flesch Reading Ease	57.86	26.48	59.58	24.90	57.94	24.27
	Gunning Fog	10.48	4.88	10.22	4.67	10.17	4.43
	Polarity	0.12	0.24	0.13	0.25	0.13	0.25
	Subjectivity	0.34	0.30	0.36	0.30	0.40	0.30
	Difficult Words	4.37	2.35	4.22	2.24	4.59	2.19
	Syllable Counts	24.53	9.36	23.79	8.89	24.89	8.31
	Word Counts	16.25	6.44	15.96	6.06	16.52	5.60
Project's Title	Flesch Reading Ease	59.31	45.46	63.76	38.82	59.67	34.27
	Gunning Fog	7.26	8.10	6.42	6.79	7.10	6.21
	Polarity	0.05	0.19	0.05	0.20	0.05	0.21
	Subjectivity	0.15	0.27	0.18	0.28	0.20	0.30
	Difficult Words	1.62	1.23	1.70	1.23	2.07	1.31
	Syllable Counts	8.15	4.18	8.81	4.02	9.98	4.10
	Word Counts	5.20	2.60	5.83	2.60	6.39	2.55

I plotted Figure 4.6 through Figure 4.8 for the overall distribution of polarity and subjectivity scores for each text input variables with boxplots of each class. Both polarity and subjectivity of creator’s biography have a higher median in super successful class than the rest; same trend exists for project’s pitch as well. As shown in Figure 4.8, project’s title contains short text that have a high sparsity in metrics used. Therefore, going forward, this variable would not be considered for further analysis and would not be implemented with the deep learn models as well.

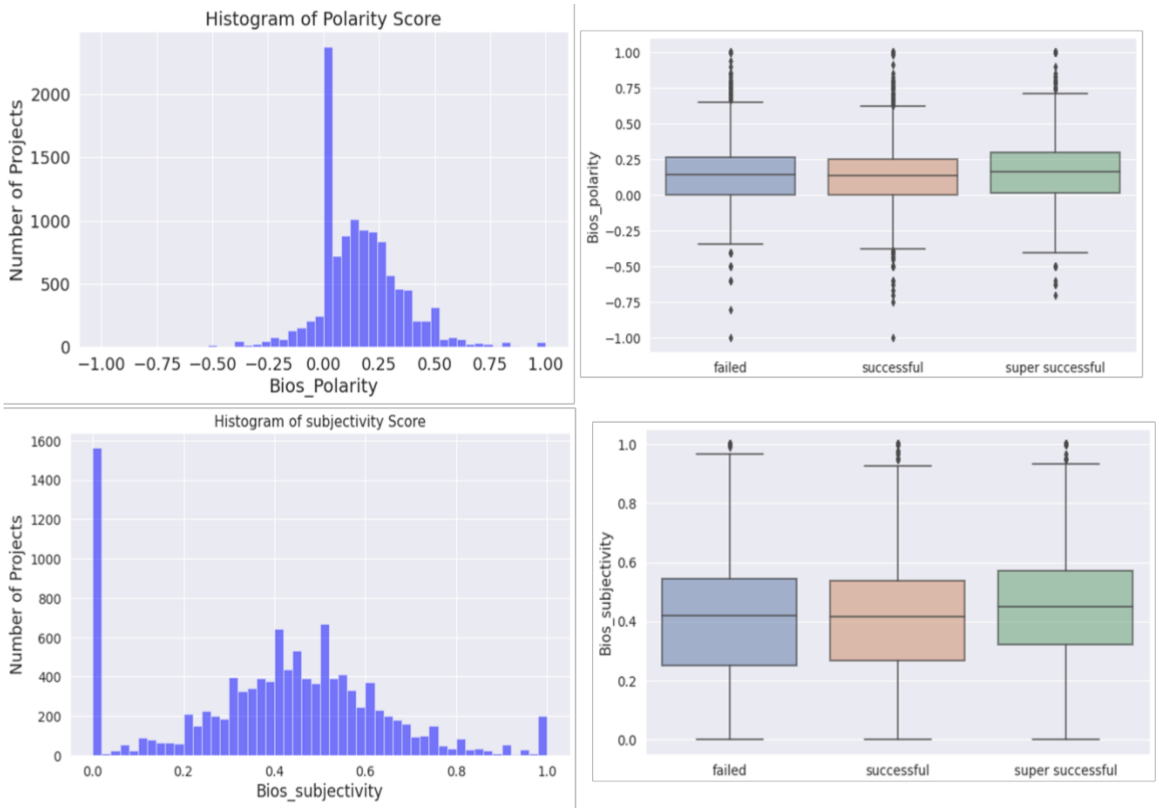


Figure 4.6: The overall subjectivity and polarity score distribution for creator's biography with boxplot for each class.

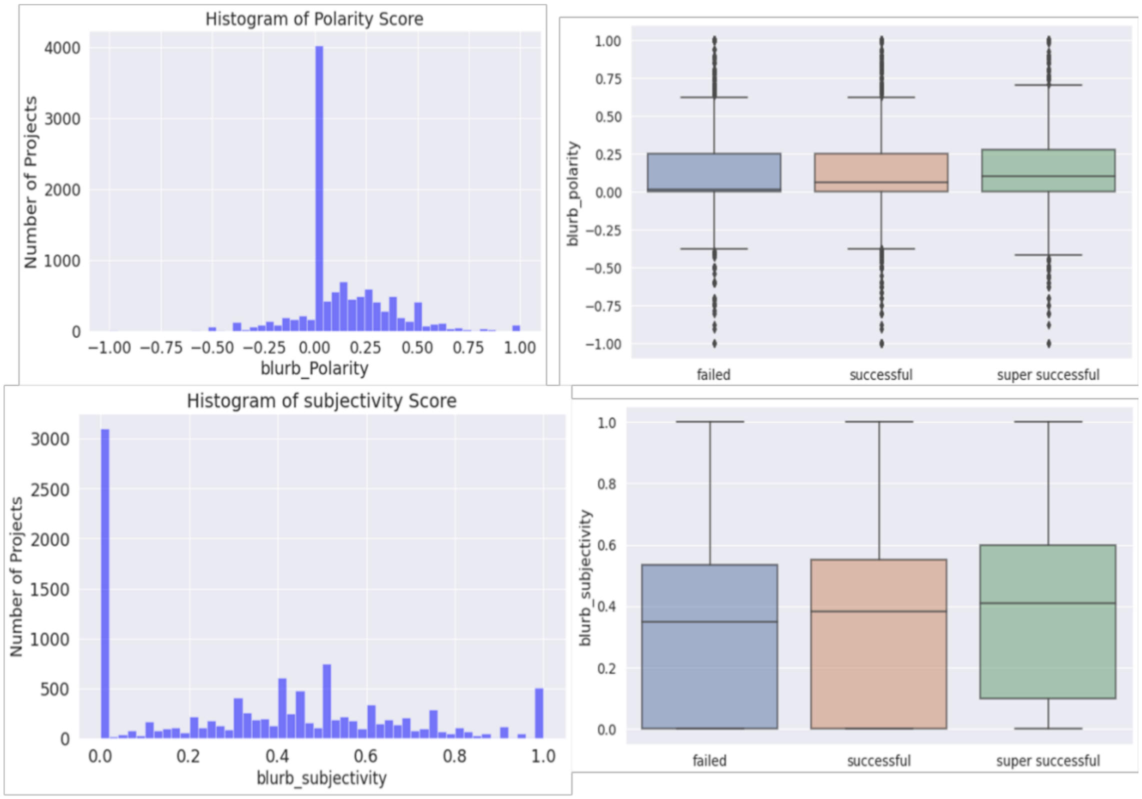


Figure 4.7: The overall subjectivity and polarity score distribution for project's pitch with boxplot for each class.

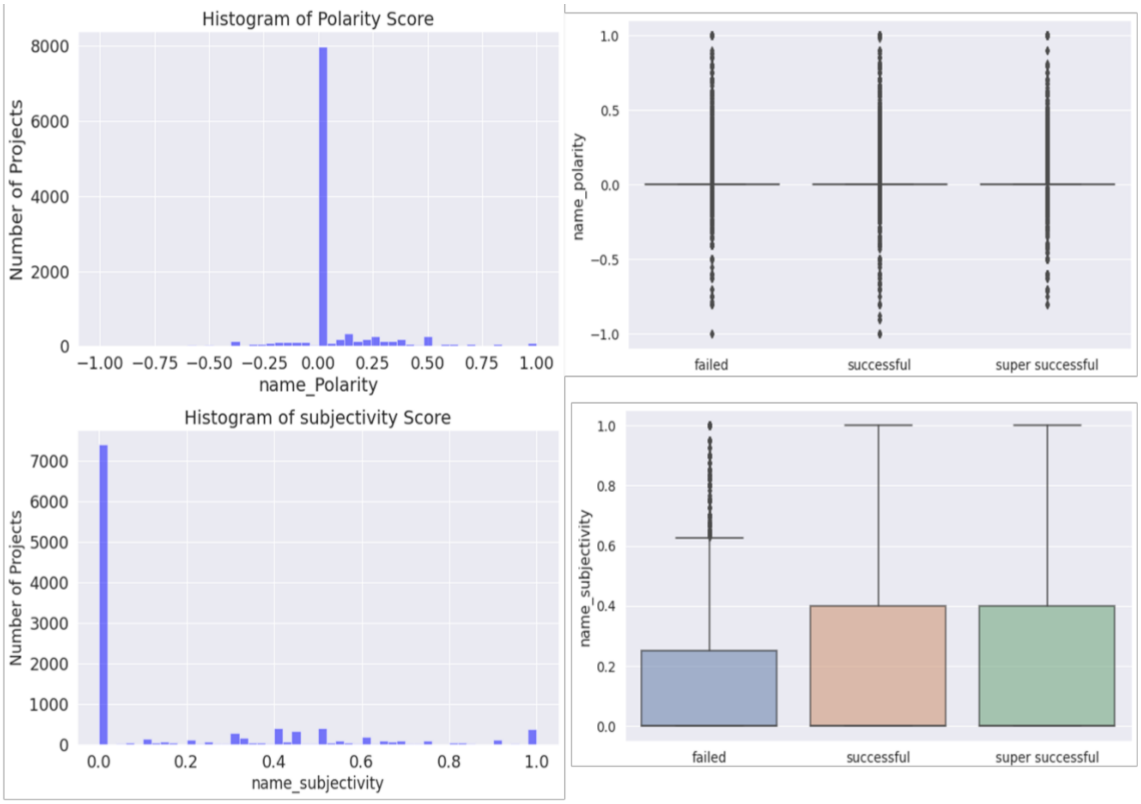


Figure 4.8: The overall subjectivity and polarity score distribution for project's title with boxplot for each class.

To investigate further in the polarity and subjectivity characteristics for creator's biography and project's pitch, I also plotted Figure 4.9 and Figure 4.10 by the project category level, Figure 4.11 and Figure 4.12 at the geographical level as well. For project's pitch in Figure 4.9, both design and fashion categories have the highest value of the 75th percentile in polarity and subjectivity, also these two categories have larger interquartile range than other categories. For creator's biography in Figure 4.10, both design and craft categories have the highest value of the 75th percentile in polarity and subjectivity. Fashion and crafts have a larger interquartile range in polarity while technology has the largest interquartile range in subjectivity. In Figure 4.11, North

Dakota has the highest polarity score (0.3132) while Montana has the lowest polarity score (0.1225) for creator’s biography; Arkansas has the highest subjectivity score (0.5236), and Rhode Island has the lowest subjectivity (0.2844) score for creator’s biography. In Figure 4.12, Delaware has the highest polarity score (0.2025) while Nebraska has the lowest polarity score (-0.0042) for project’s pitch; West Virginia has the highest subjectivity score (0.5252), and Wyoming has the lowest subjectivity score (0.2087) for project’s pitch.

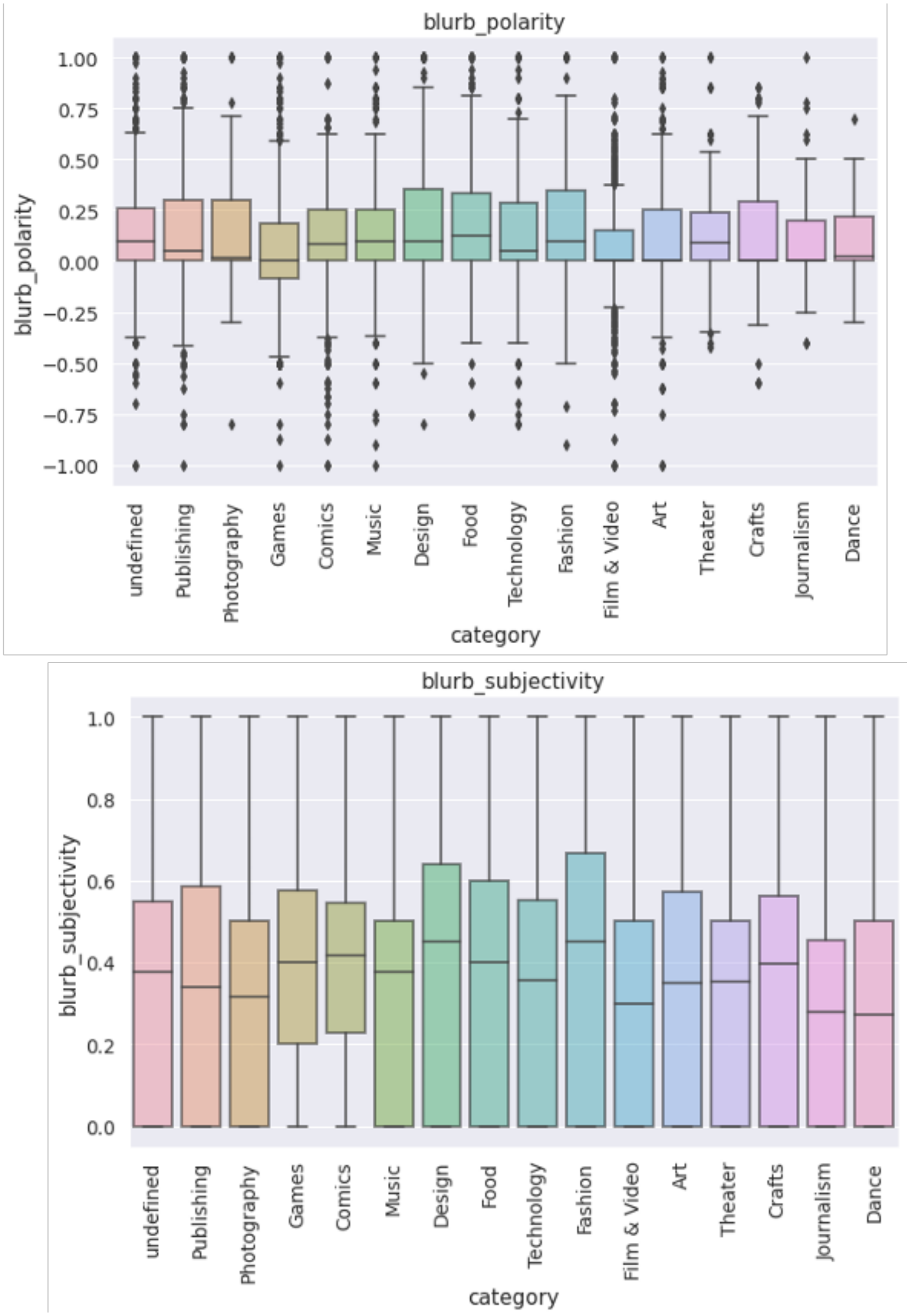


Figure 4.9: The overall subjectivity and polarity score distribution for project's pitch by project categories.

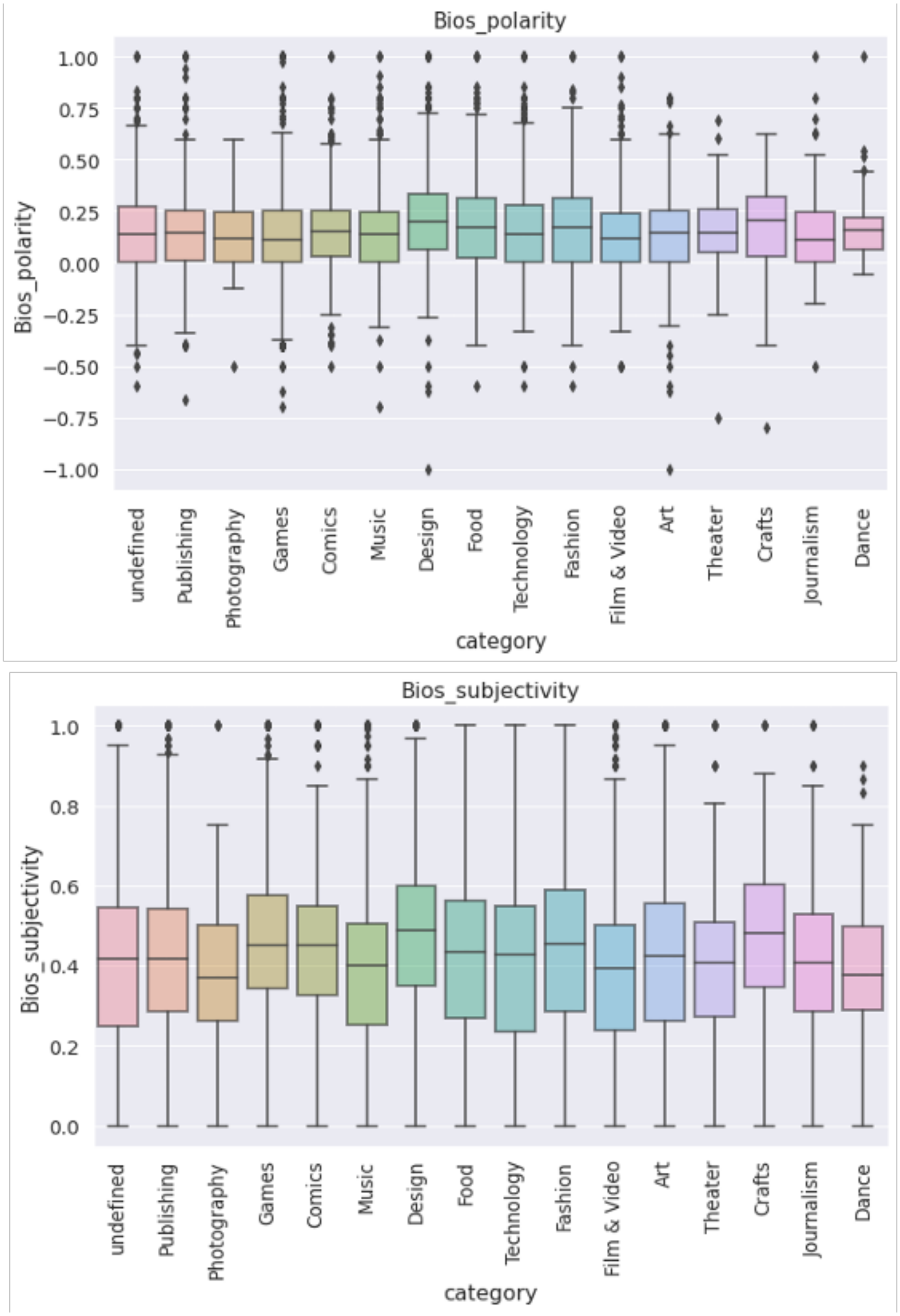


Figure 4.10: The overall subjectivity and polarity score distribution for creator's biography by project categories.

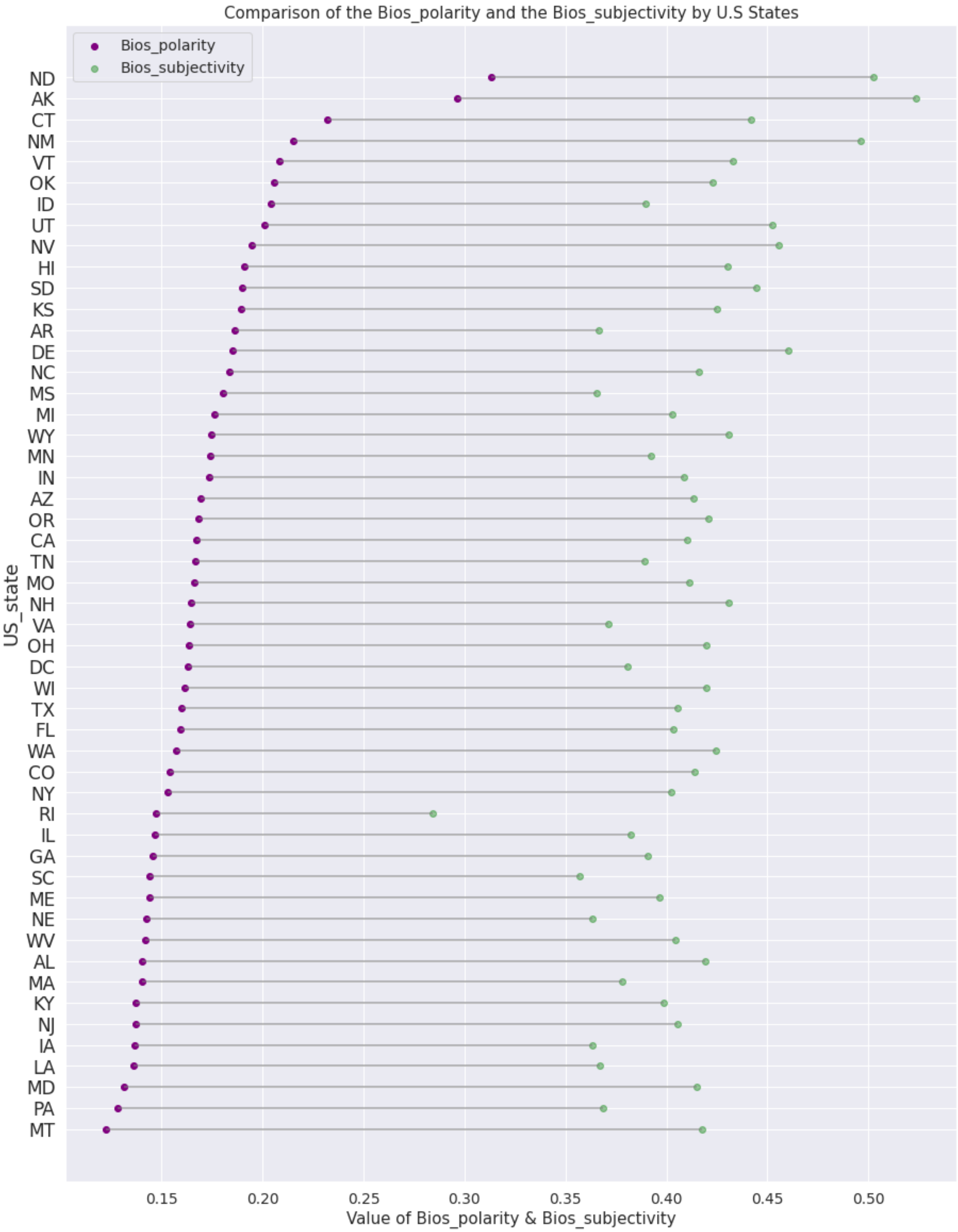


Figure 4.11: The comparison of subjectivity and polarity score distribution for creator's biography by U.S states.

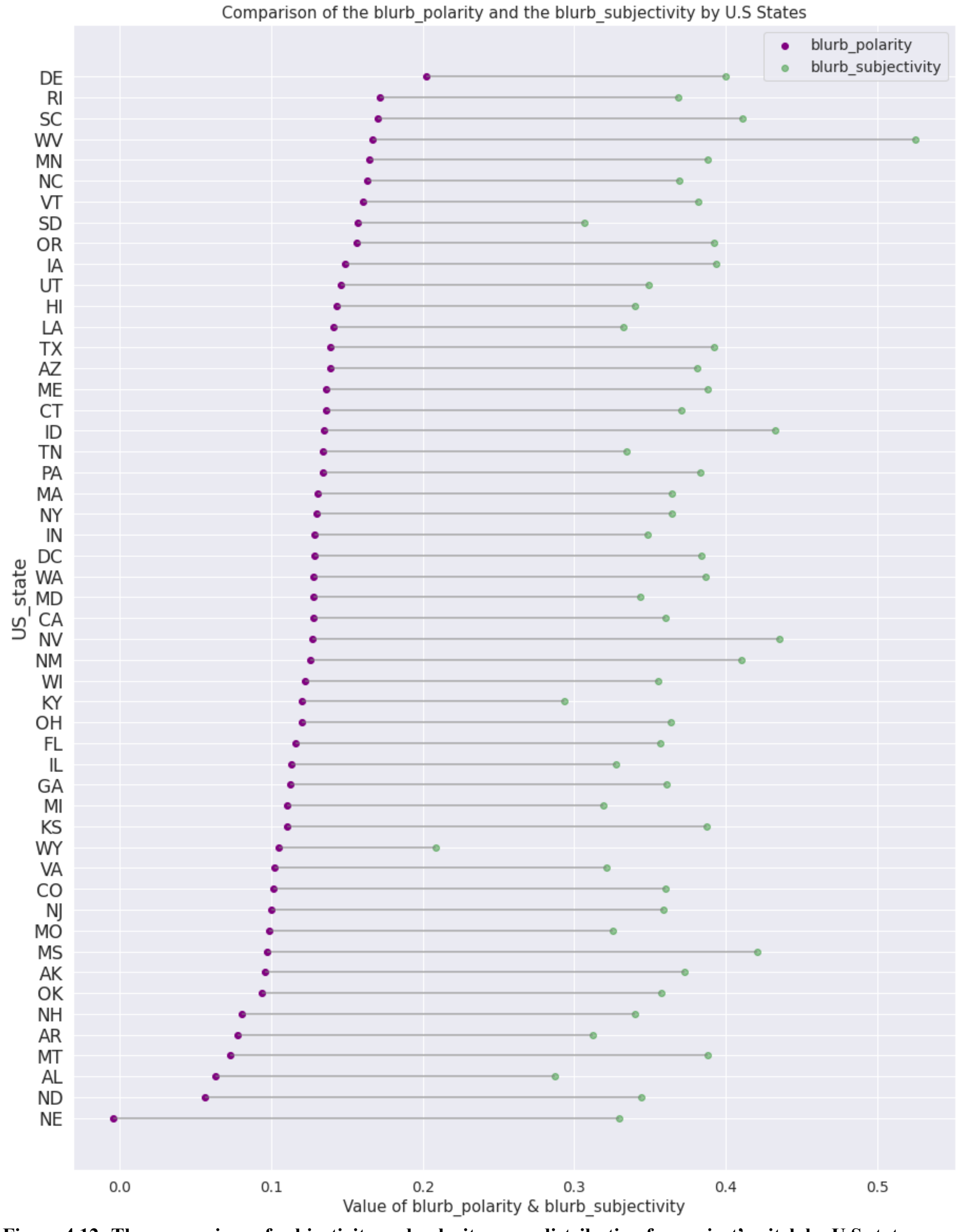


Figure 4.12: The comparison of subjectivity and polarity score distribution for project's pitch by U.S states.

For statistical significance exhibits among all three classes, a one-way ANOVA was implemented, followed by Tukey test to further investigate within class for significance difference. Table 4.5 provides the results of the one-way ANOVA statistics and Tukey test for significance testing among class. Figure 4.13 through Figure 4.17 provide the overall distribution and individual class distribution on length features and vocabulary features.

For creator’s biography, all means comparison of the computed metrics are significant with p-value $<\alpha=0.05$ by one-way ANOVA test. Within class, the compared means between failed class and successful class shows significance with Tukey test for the length and vocabulary features, but not the lexicon features. On the contrary, there is a statistical evidence shows a significant distinction when super successful level compared to the other levels with Tukey’s p-value $<\alpha=0.05$.

For project’s pitch, all means comparison of the computed metrics are significant with p-value $<\alpha=0.05$ by one-way ANOVA, except polarity. Within class, the compared means between failed class and successful class shows significance with Tukey test for the length and vocabulary features, except word counts. For significant trait in super successful class, only subjectivity and counts of difficult words show significant distinction when compared to the other classes across with Tukey’s p-value $<\alpha=0.05$.

For project’s title, all means comparison of the computed metrics are significant with p-value $<\alpha=0.05$ by one-way ANOVA, except polarity. Within class, there is a significant difference in means between failed class and successful class with Tukey test for length features, vocabulary features, and subjectivity score. For significant trait in super successful class, only counts of difficult words, syllable, and all words show significant distinction when compared to the other levels across with Tukey’s p-value <α=0.05.

Table 4.5: Tukey’s test statistics by within class comparison and the overall one way anova statistics.

		failed & successful		failed & super successful		successful & super successful		All Class
	Text Metrics	Mean Diff	Tukey P Value	Mean Diff	Tukey P Value	Mean Diff	Tukey P Value	ANOVA P Value*
Creator's Biography	Flesch Reading Ease	2.85	0.00	1.99	0.06	-0.86	0.55	0.00
	Gunning Fog	-0.66	0.01	-0.52	0.11	0.15	0.80	0.01
	Polarity	0.01	0.23	-0.01	0.02	-0.02	0.00	0.00
	Subjectivity	0.00	0.90	-0.04	0.00	-0.04	0.00	0.00
	Difficult Words	-3.46	0.00	-2.49	0.00	0.98	0.05	0.00
	Syllable Counts	-14.76	0.00	-9.27	0.01	5.49	0.16	0.00
	Word Counts	-9.03	0.00	-5.70	0.02	3.34	0.23	0.00
Project's Pitch	Flesch Reading Ease	-1.72	0.01	-0.08	0.90	1.64	0.02	0.00
	Gunning Fog	0.26	0.03	0.31	0.02	0.05	0.87	0.01
	Polarity	-0.01	0.45	-0.01	0.20	0.00	0.75	0.21*
	Subjectivity	-0.02	0.00	-0.06	0.00	-0.04	0.00	0.00
	Difficult Words	0.15	0.01	-0.22	0.00	-0.37	0.00	0.00
	Syllable Counts	0.74	0.00	-0.36	0.24	-1.10	0.00	0.00
	Word Counts	0.29	0.07	-0.27	0.19	-0.56	0.00	0.00
Project's Title	Flesch Reading Ease	-4.45	0.00	-0.36	0.90	4.09	0.00	0.00
	Gunning Fog	0.83	0.00	0.16	0.63	-0.68	0.00	0.00
	Polarity	-0.01	0.28	0.00	0.69	0.00	0.82	0.31*
	Subjectivity	-0.03	0.00	-0.04	0.00	-0.01	0.10	0.00
	Difficult Words	-0.08	0.01	-0.45	0.00	-0.37	0.00	0.00
	Syllable Counts	-0.66	0.00	-1.83	0.00	-1.17	0.00	0.00
	Word Counts	-0.63	0.00	-1.19	0.00	-0.56	0.00	0.00

*cannot conclude statistically significant at 5% by one-way ANOVA. Italic bold values are statistically significant at 5%.

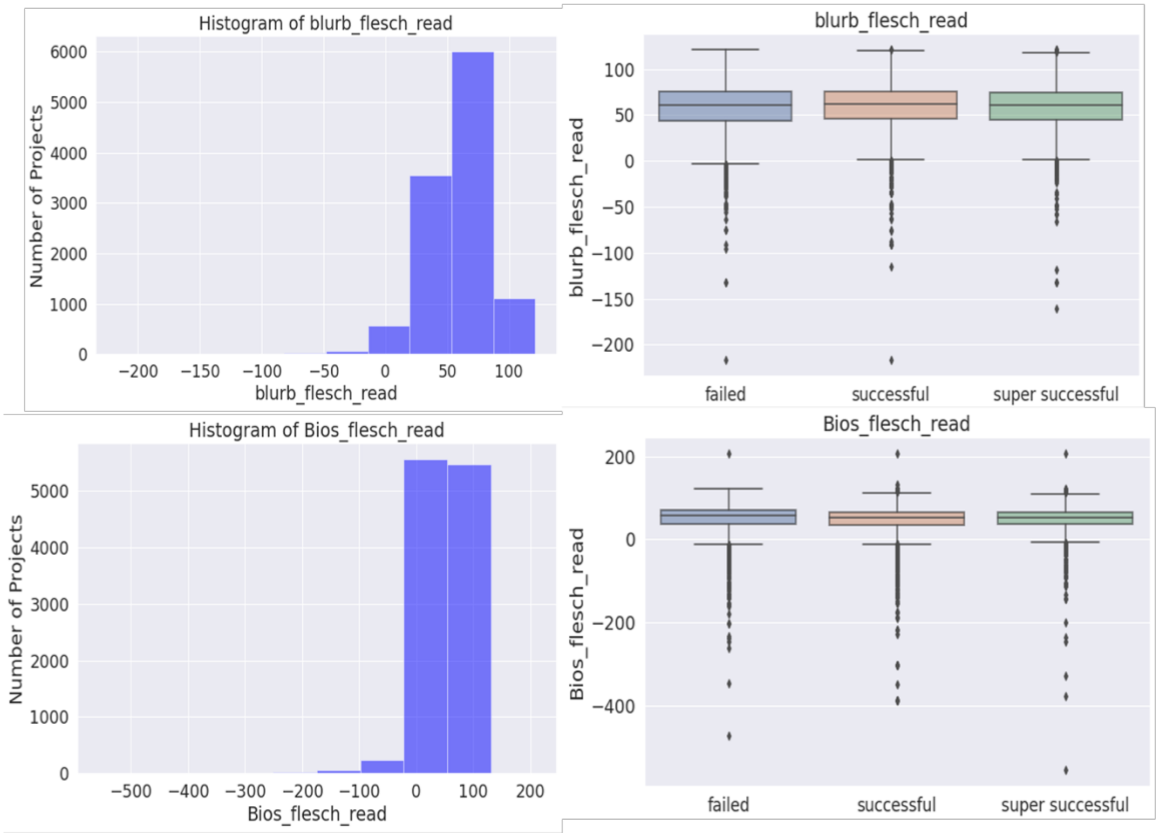


Figure 4.13: The overall Flesch Reading Ease Score distribution for creator's biography and project's title with boxplot for each class.

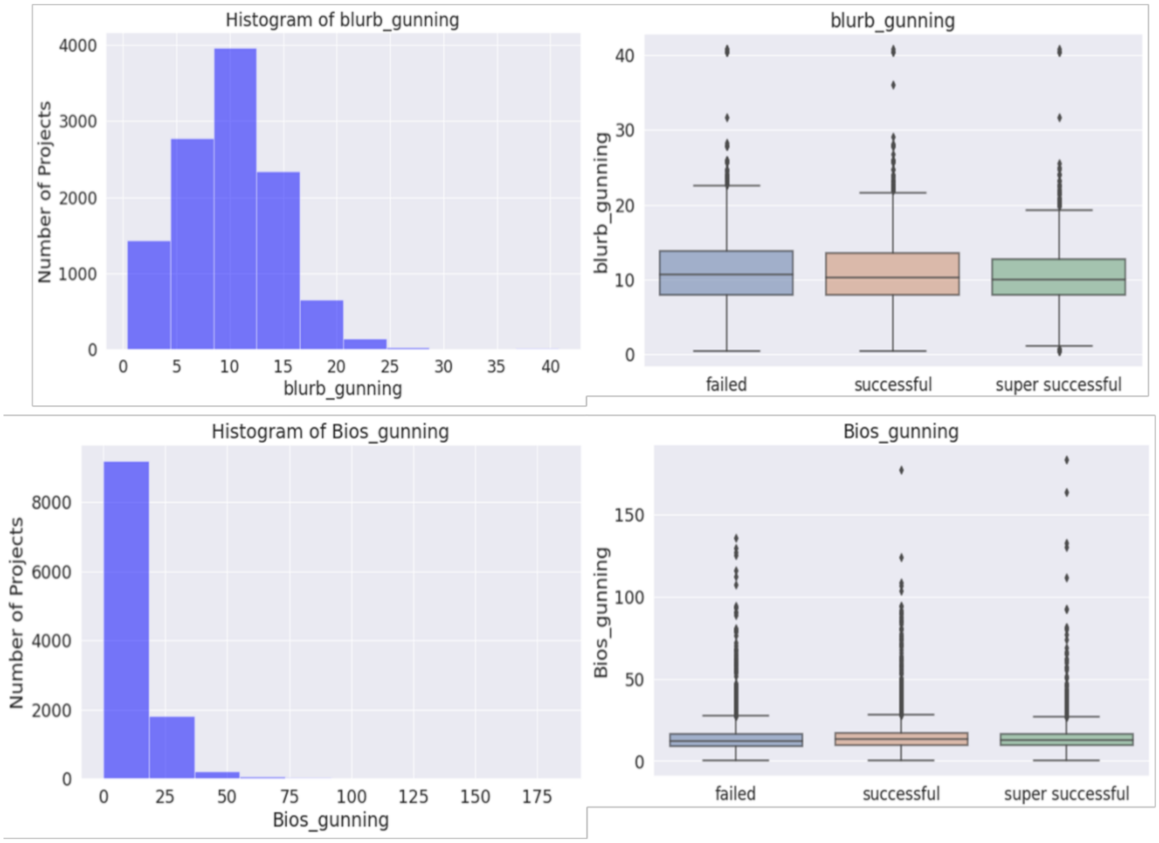


Figure 4.14: The overall Gunning's Fog Index score distribution for creator's biography and project's title with boxplot for each class.

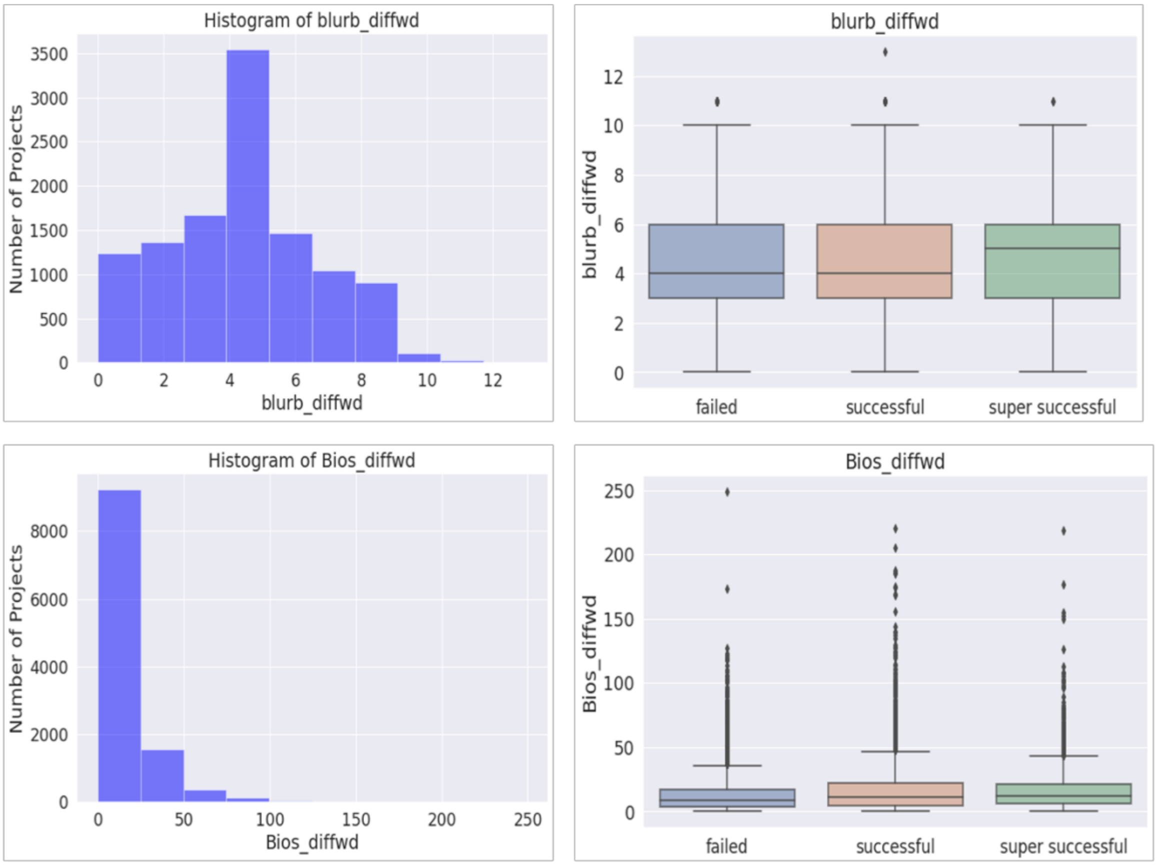


Figure 4.15: The overall difficult word counts distribution for creator's biography and project's title with boxplot for each class.

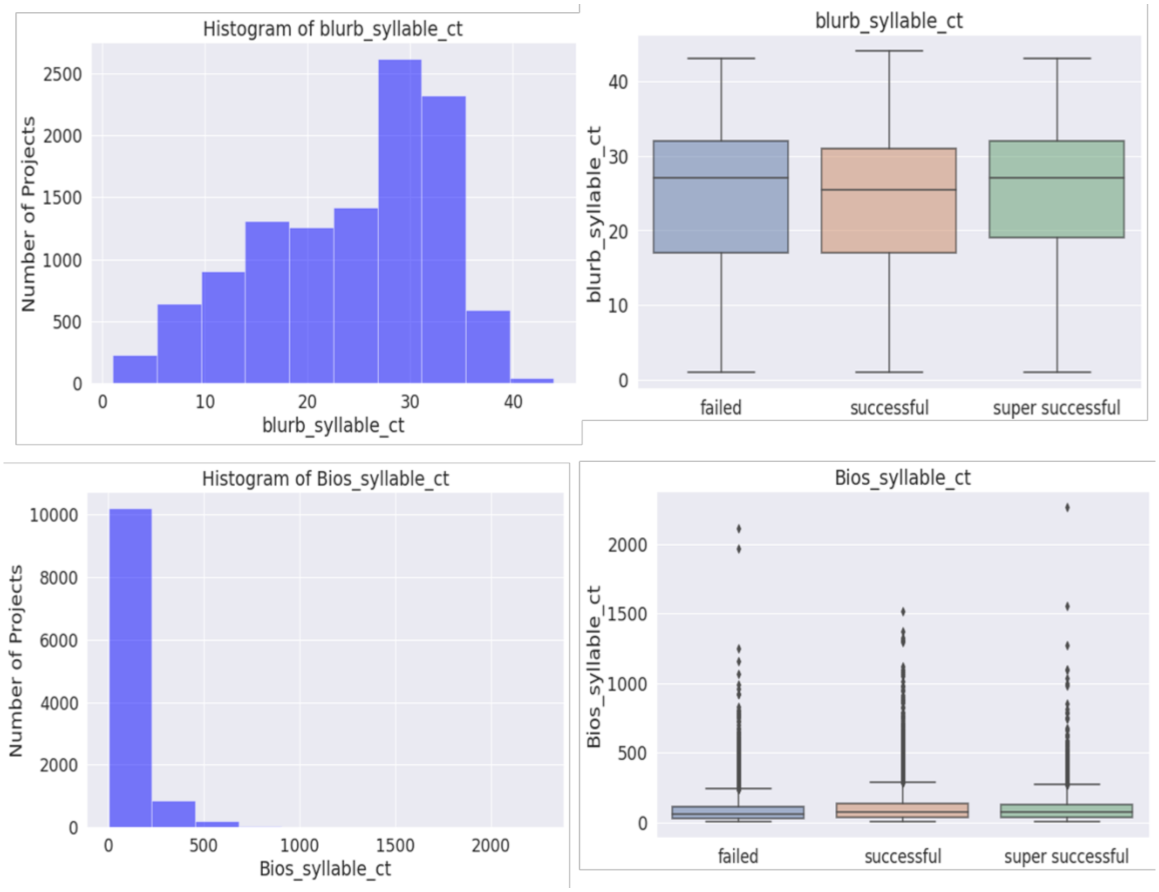


Figure 4.16: The overall syllable counts distribution for creator's biography and project's title with boxplot for each class.

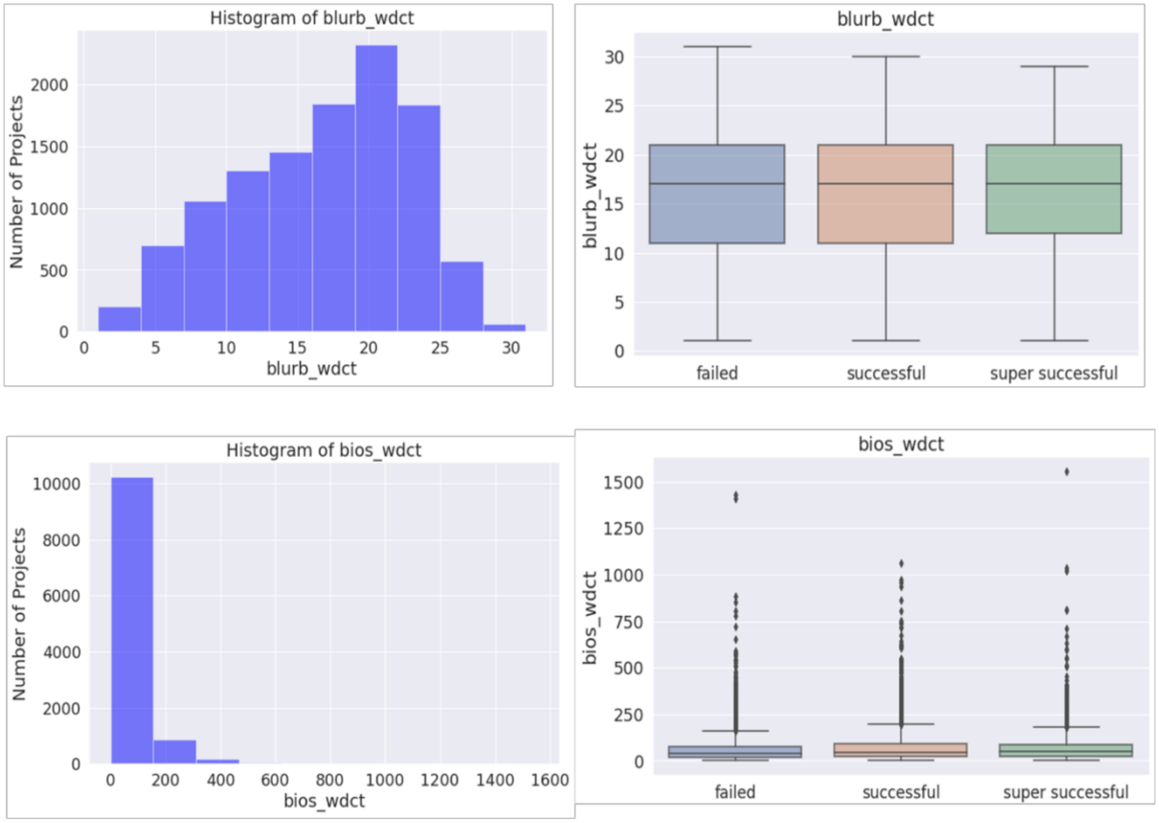


Figure 4.17: The overall word counts distribution for creator's biography and project's title with boxplot for each class.

4.5.2 Results from Baseline Model

Given the observed distributions from the findings in exploratory analysis, I chose to only include creator's biography and project's pitch as the primary text inputs for deep neural networks. From the results in Table 3.2 of Section 3.4.2, Naïve Bayes trained with only meta information yields 56.68% accuracy with precision of 58.83% and recall of 59.10%. Based on the results in Table 4.6, as different type of information are being introduced into the model, the higher is the accuracy using Naïve Bayes with alpha value=1. In addition, the trained Naïve Bayes using the joint input of both creator's biography and project's pitch with meta information performed best, it yields 63.91%

accuracy with precision of 63.13% and recall of 63.96%. Table 4.7 shows the testing confusion matrix for classification performance when trained on the model of multimodalities. The failed class has the highest precision, recall and F1 score; the super successful class has the lowest precision and F1 score but not recall.

Table 4.6: Performance summary of baseline model with Naïve Bayes using different combinations of text and meta input features.

Input Features	Accuracy	Precision	Recall	F1-Score	Run Time (seconds)
Blurb	0.5633	0.5545	0.5495	0.5517	5.66
Bios	0.5877	0.5787	0.5840	0.5803	9.99
Meta+Blurb	0.6200	0.6214	0.6048	0.6109	0.548
Meta+Bios	0.6226	0.6146	0.6222	0.6177	1.11
Meta+Blurb+Bios	0.6391	0.6313	0.6396	0.6346	1.61

Table 4.7: Confusion matrix and classification report of Naïve Bayes algorithm using both project's pitch and creator's biography with meta information.

Meta+Blurb+Bios		**Predicted* Class**			**Classification Report**			
	Total (n=3405)	**failed**	**successful**	**super successful**	**precision**	**recall**	**f1-score**	**Total Observation**
True Class	failed	799	248	89	0.6620	0.7033	0.6820	1136
	successful	309	894	275	0.6617	0.6049	0.6320	1478
	super successful	99	209	483	0.5702	0.6106	0.5897	791

*Term "Predicted" is used as a generalization of all projects from machine learning classification model, not for a specific project.

The baseline results from Naïve Bayes are as expected considering the underlying assumption that the probability of each observed word is conditionally independent from each other. To increase the classification performance, deep learning algorithms were trained using TensorFlow Keras in python. The goal was to obtain improvement by reducing training errors for using deep learn algorithms. Deep learn algorithms can generate a denser vectorization which might have better weight estimation on each feature through weight calibration by using gradient descents in each iteration, instead of relying on probability of each word per each document.

4.5.3 Results from Multimodal Deep Learning

A standardization set of hyperparameters was used to train and test on four deep learning algorithms with Adam optimizer as listed in Table 4.8. By using the GloVe word embedding, the maximum number of words to be used in each text input variable was set to 10,000 where both embedding dimension and the input vector length were set to size of 100. To compare, only one hidden layer of either LSTM, GRU, CNN or the stacked LSTM-CNN were implemented for model training, and SoftMax activation function was used along with categorical cross entropy for multiclass classification using 50 epochs.

Table 4.8: Standardized hyperparameter settings for deep learning algorithms comparison.

Hyperparameters	Value
Input Vector Units	100
Algorithm Internal Units	64
Learning Rate	0.0002
Decay	0.001
Dropout Rate	0.5
Fully Connected Dense Layer Units	256
Dense Layer Output Units	3
Epochs	50
Batch Size	128

I trained all four deep learning algorithms on different combinations of variable types: text only, text and meta information, and all text with meta information. The TensorFlow architectures of three possible combinations in text and meta data inputs are depicted in Figure 4.18 through Figure 4.20. For multimodalities training, a dense layer was used on meta information that contains numeric and categorical features. Subsequently, the output of the deep neural networks of text input and the output of the meta data dense layer were concatenated as an input into the fully connected layer and computed by SoftMax activation function for a final output for classification.

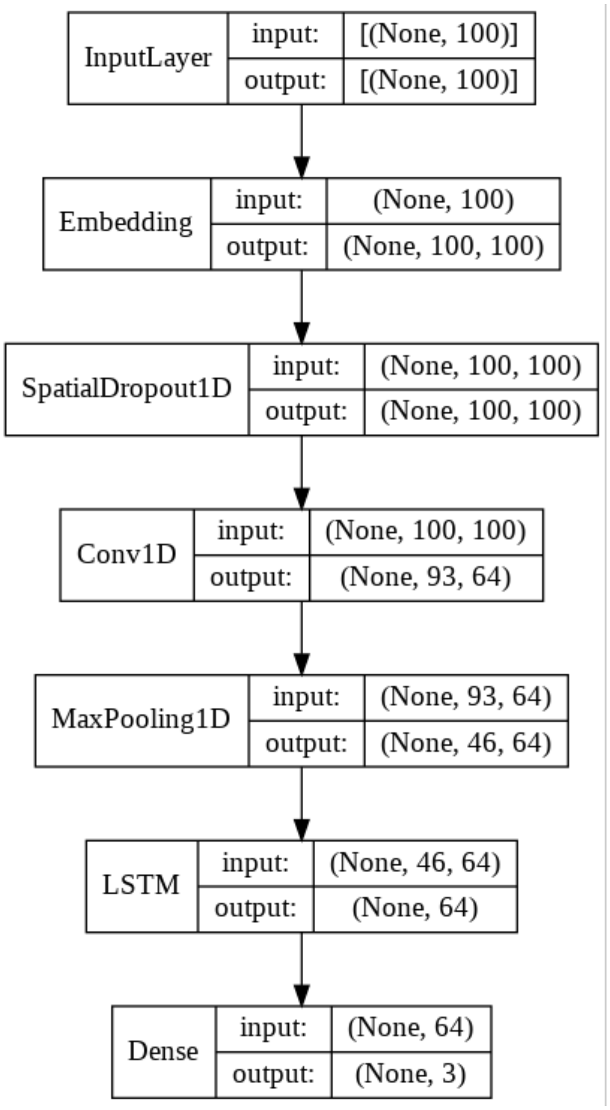


Figure 4.18: An example of architecture workflow for training a single text input using stacked CNN-LSTM model.

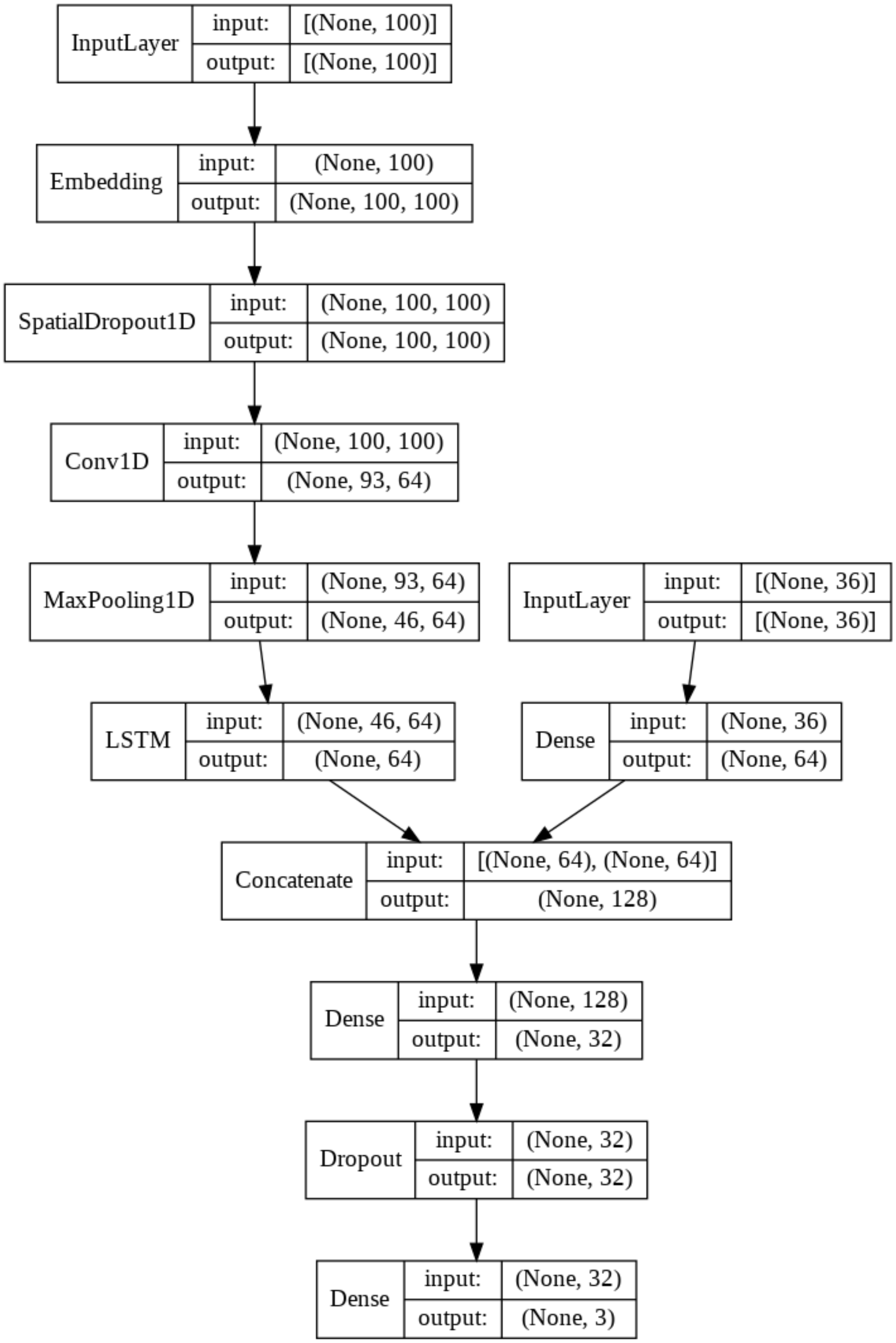


Figure 4.19: An example of architecture workflow for training a single text input with meta information using stacked CNN-LSTM model.

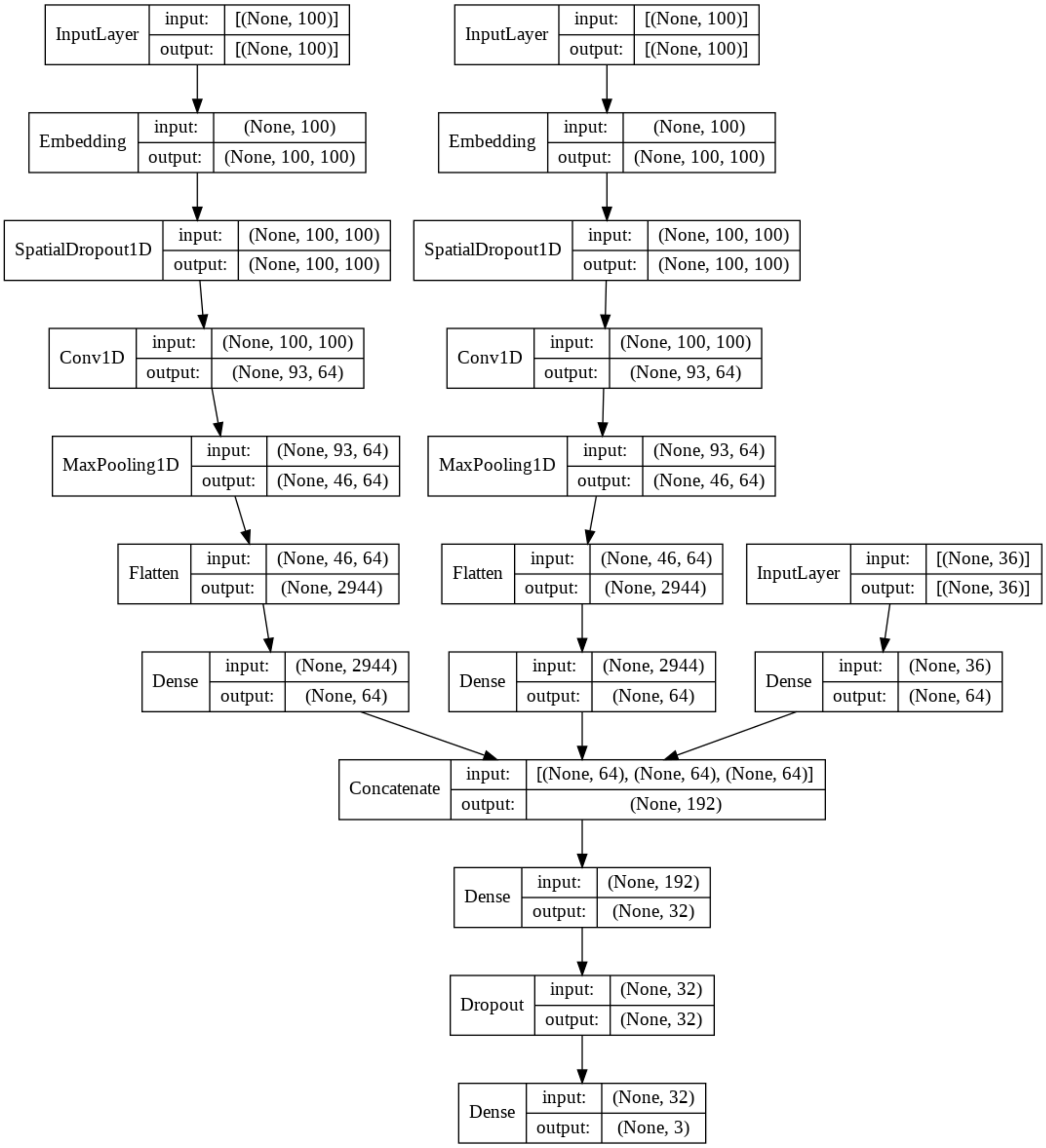


Figure 4.20: An example of architecture workflow for training both text inputs with meta information using stacked CNN-LSTM model.

The classification results on test dataset from the trained deep neural networks are presented in Table 4.9 with the loss error, where accuracy were used to evaluate performances. Same trend observed in the Naïve Bayes baseline model, when both text inputs and meta features were integrated in the training, it yields the best classification performance in comparison to other data input combinations, except for GRU model. Instead, the GRU model trained with the combination of creator's biography and meta information appears to have a slightly better performance by 0.47%. Based on the performance of 50 epochs, LSTM model that trained on both text inputs and meta features yields the best classification accuracy of 70.54% and had the longest training time among all models. On the contrary, CNN provides the least performance with 69.10% in accuracy, but it required the least training time as a trade-off.

As shown in Table 4.10 for the confusion matrix of the best performing LSTM, its failed class has the highest precision, recall and F1 score while the super successful class has the lowest precision, recall, and F1 score. Figure 4.21 includes the training accuracy and validation loss.

Table 4.9: Performance summary of deep neural networks using different combinations of text and meta input features.

Input Data Combination (Naïve Bayes' Accuracy)	Deep Learn Algorithms (50 Epochs)	Accuracy	Loss	Precision	Recall	F1-Score	Run Time (m=minutes, s=seconds)
Project's Pitch (NB: 0.5633)	LSTM	0.5439	1.1470	0.5383	0.5450	0.5405	25m 48s
	GRU	0.5078	0.9912	0.4995	0.5001	0.4994	10m 56s
	CNN	0.4620	1.1298	0.4503	0.4513	0.4506	25s
	CNN+LSTM	0.4799	1.0759	0.4728	0.4702	0.4672	6m 27s
Creator's Biography (NB: 0.5877)	LSTM	0.5703	1.0630	0.5629	0.5695	0.5652	26m 40s
	GRU	0.5313	0.9623	0.5382	0.5402	0.5263	10m 59s
	CNN	0.4799	1.1519	0.4749	0.4788	0.4736	14s
	CNN+LSTM	0.5263	1.0019	0.5234	0.5309	0.5216	6m 12s
Project's Pitch + Meta Information (NB: 0.6200)	LSTM	0.6740	0.8261	0.6742	0.6685	0.6711	15m 25s
	GRU	0.6907	0.6925	0.6879	0.6968	0.6917	13m 1s
	CNN	0.6849	0.7246	0.6806	0.6916	0.6849	41s
	CNN+LSTM	0.6866	0.7395	0.6872	0.6872	0.6868	7m 32s
Creator's Biography + Meta Information (NB: 0.6226)	LSTM	0.6925	0.7936	0.6913	0.6885	0.6898	15m 19s
	GRU	0.6969	0.6896	0.6946	0.7038	0.6982	13m 3s
	CNN	0.6884	0.9483	0.6651	0.6725	0.6681	22s
	CNN+LSTM	0.6790	0.7406	0.6759	0.6856	0.6798	7m 39s
Project's Pitch + Creator's Biography + Meta Information (NB: 0.6391)	LSTM	0.7054	0.7021	0.7054	0.7012	0.7030	23m 42s
	GRU	0.6922	0.7014	0.6884	0.7004	0.6931	18m 57s
	CNN	0.6910	0.7122	0.6874	0.6944	0.6902	39s
	CNN+LSTM	0.6940	0.6941	0.6904	0.7004	0.6945	11m 12s

Table 4.10: Confusion matrix and classification report of LSTM using both project's pitch and creator's biography with meta information.

Meta+Blurb+Bios		Predicted* Class			Classification Report			
	Total (n=3405)	failed	successful	super successful	precision	recall	f1-score	Total Observation
True Class	failed	840	235	48	0.7902	0.7480	0.7685	1123
	successful	185	1018	246	0.6769	0.7026	0.6895	1449
	super successful	38	251	544	0.6492	0.6531	0.6511	833

*Term "Predicted" is used as a generalization of all projects from machine learning classification model, not for a specific project.

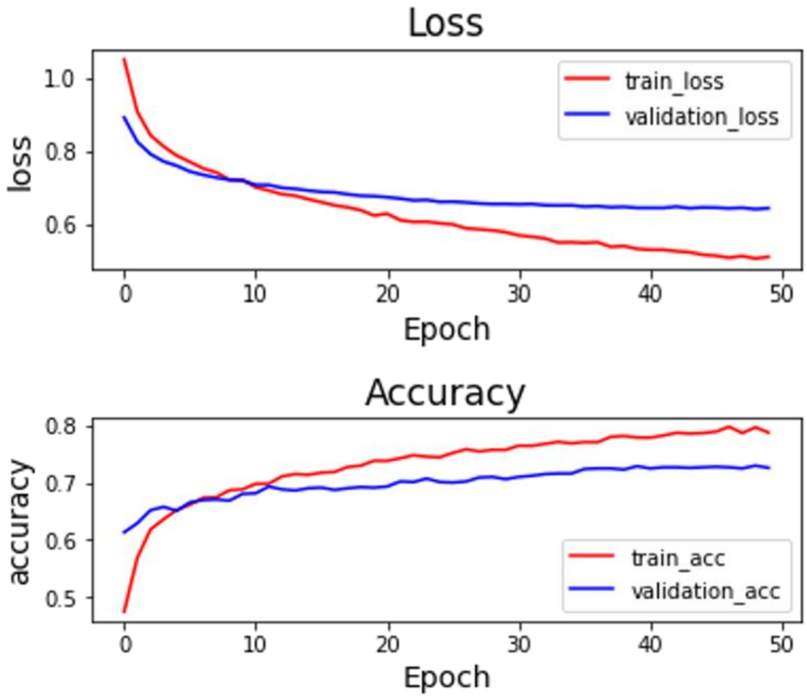


Figure 4.21: Accuracy and validation loss for the best performing LSTM model.

4.5.4 Results from Tunning Hyperparameters

To further improve the best performing LSTM model from Table 4.9, I tried to increase the hidden layers to a two-layered LSTM and adjust the hidden units use (32, 64, 128, or 256), as well as the units in the dense layer (64, 128, or 256). I also trained with different a different learning rate (0.00008 to 0.0004) and a different batch size (128 or 256), along with different number of epochs. Bidirectional LSTM layers were also tested to avoid overfitting.

After many trials of tuning, a single layer of LSTM with 64 hidden units at learning rate of 0.00008 (decay=0.001) and training batch size of 128 achieves the highest accuracy of 71.04%, where it yields 71.16% in recall and 70.74% in precision along with 70.89% of F1-score when trained with 200 epochs. Dense layer units of meta information remained the same, no adjustment made, except I used the BPSO feature selected set of meta information with 24 features as shown in Figure 4.22. Total training time was 94 minutes and 4 seconds. Performance summary is reported in Table 4.11 with Figure 4.23 for training accuracy and validation loss. Similar to the observed trend, again, failed class has the highest precision, recall and F1 score while the super successful class has the lowest precision, recall, and F1 score.

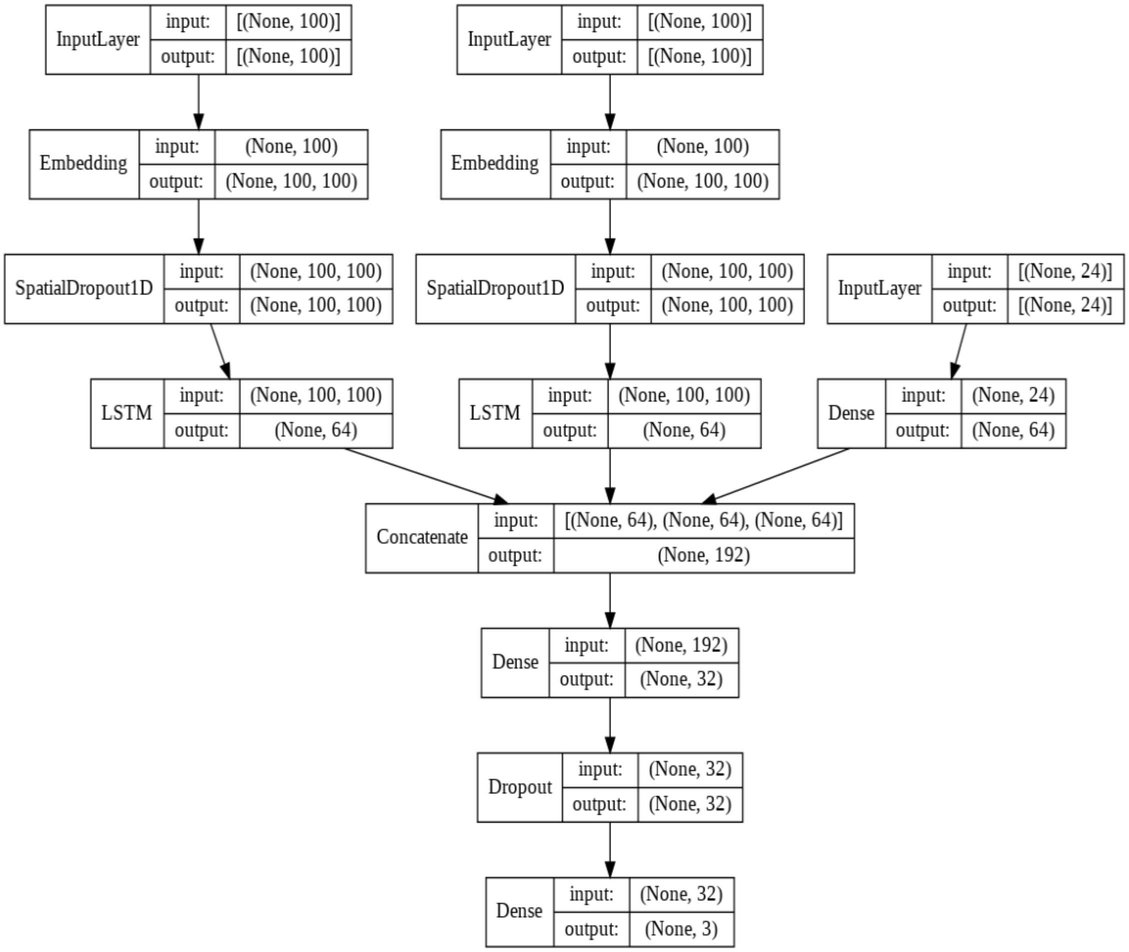


Figure 4.22: The architecture workflow of the best tuned LSTM model.

Table 4.11: Confusion matrix and classification report of the best tuned LSTM using both project's pitch and creator's biography with meta information.

Meta+Blurb+Bios		Predicted* Class			Classification Report			
	Total (n=3405)	failed	successful	super successful	precision	recall	f1-score	Total Observation
True Class	failed	833	250	48	0.8033	0.7418	0.7713	1123
	successful	163	1026	260	0.6804	0.7081	0.6939	1449
	super successful	41	232	560	0.6512	0.6723	0.6615	833

*Term "Predicted" is used as a generalization of all projects from machine learning classification model, not for a specific project.

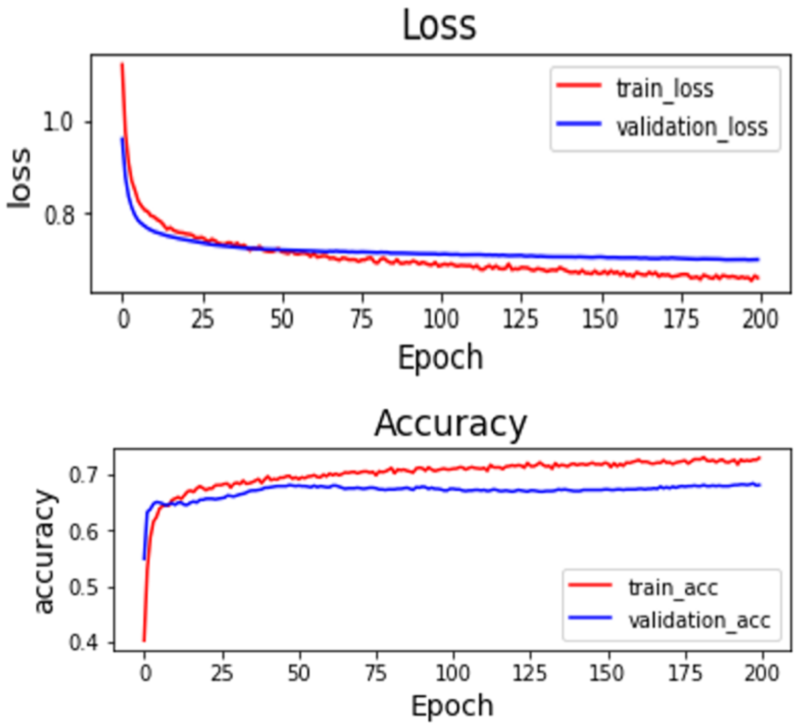


Figure 4.23: Accuracy and validation loss for the best tuned LSTM model.

4.6 Discussion and Conclusion

The findings from the exploratory analysis of texts using lexicon, vocabulary, and length features suggest that there are significant differences among the three success classes. This indicates textual information could potentially improve classification performance in addition of using meta information from the project alone. With the implementation of NLP and DNNs using TensorFlow Keras architecture, four different DNNs were compared against the Naïve Bayes baseline model. Moreover, different combinations of information to feed in the DNNs were also examined.

The overall results show that the more information was introduced to the baseline model or any DNNs, the better the classification performance shows. When trained on using either project's pitch or creator's biography text input only without any meta information, the Naïve Bayes baseline model outperformed any DNNs. When combined text input(s) with meta information, all DNNs outperformed the Naïve Bayes baseline model. Among all the performance of 50 epochs, LSTM model that trained on both text inputs and meta features yields the best classification accuracy of 70.54%, and had the longest training time among all models. On the contrary, CNN provides the least performance with 69.10% in accuracy, but it required the least training time as a trade-off. This compared result is in line with the studies from Zhou [63] on using the project's pitch from Kickstarter for text classification based on text summarization, where the performance of CNN with accuracy of 78% is slightly behind LSTM with accuracy of 82% and GRU with accuracy of 80% when compared.

With further tuning on hyperparameters and extending 50 epochs to 200 epochs, using both text inputs and a BPSO selected set of meta information, a single layer of LSTM with 64 hidden units at learning rate of 0.00008 (decay=0.001) achieves the highest accuracy of 71.04%, where it yields 71.16% in recall and 70.74% in precision along with 70.89% of F1-score when trained with a batch size of 128. However, the tuned result from the LSTM model did not show a substantial improvement nor outperformed the presented XGBoost-BPSO model from Section 3.5.3 in Chapter 3, where it achieves the highest accuracy of 74.61% and F-score of 74.64% as the most optimal model in comparison. I did not extend further tuning with more than 200 epochs considering only a small significant change would observe at best and might cause overfitting as well. Although Katsamakas and Sun [62] conducted their studies using the project's pitch of Indiegogo data to predict funding success, a similar conclusion was reached by showing SVM achieved the highest accuracy of 89.43% where LSTM only provided 75.57% in accuracy.

My findings in this research are in accordance with the reported findings when applied NLP with DNNs for multimodalities. In summary, this study is the first to utilize creator's biography text information in deep neural networks for multiclass classification. Despite a classical machine learning model provides the optimal classification performance instead, the presented deep learn architectures can provide a general mechanism for other text information with other modalities such as speech from videos.

4.7 Limitations and Future Research

Although multimodalities are certainly considered a promising alternative approach on utilizing additional information other than numerical or categorical features of the project, limitations arise. Text processing in expanding chat words and contractions are limited since there are broad sets of new urban terms continuously that might not be captured in result of obstructing a better text representation. Furthermore, instead of expanding emoticons and emojis, I chose to remove them since not all emoticons and emojis have a mapping. Therefore, some projects resulted with null text inputs after processing that were excluded from my sample.

Additionally, by using only the wording embedding of 100 in length, a different vectorized length can be explored since only partial information are used in the model for classification; even a different word embedding other than GloVE can be adopted, such as FastText and Word2Vec.

As reported, the highest tuned accuracy of 71.04% for the best performing LSTM model is comparable to the observed and the reported, which it outperformed the baseline model. Though, there is an outstanding of 38.96% classification gap could be mitigated with further hyperparameters tuning, and different DNNs scheme to compare with, such Capsule Network (CapsNet) based model to address problems in spatial relationships to minimize information loss from pooling with more focus on global features [77].

5. CONCLUSION

Crowdfunding is an effective way to introduce more investors to crowdfund ideas without much financing hurdles. This novel practice gives many entrepreneurs an easy, open access platform to launch their products and reach wider group of investors. The research motivation for this dissertation is to inform and guide project creators to construct and optimize their campaigns before launching their ideas through crowdfunding. With the different levels of emergent success, three tiers of success in reward-based campaigns of the 2019 Kickstarter were evaluated. For this purpose, my aims were to examine and analyze features that were influential to crowdfunding success when campaigns first launched, and to develop classification models using relevant features at the beginning of the funding period. In this final chapter, I summarize my results and contributions, then discuss limitations and possible next steps for future research.

5.1 Results Summary and Contributions

Three primary research questions were proposed:

1) What are the indicators and their association strength in crowdfunding campaign success?

Before I addressed this first objective, I developed scripts to web scrape meta data of each of 11,924 projects automatically for data collection.

Additionally, I manually classified videos or images of each campaign into product relevant or creator relevant categories to extract my features, same process for information on description of timeline or future planning, and team formation behind project. Afterwards, I addressed this objective by applying a hierarchical multiple regression and a hierarchical ordinal logistic regression to assess the significance of each feature, and the strength of association on the crowdfunding success.

In addition to some of the known indicators for project success, such as realistic funding goals and appropriate funding periods, several novel findings were discovered. The first novel finding is that providing budget plan or product timeline to investors for transparency was important. This feature is also significant and correlates with funding success. The second novel finding suggests that badge status and mentioning relevant experience are important to project success. While the significance in mentioning educational degree or lengthy biography proved inconclusive since two models did not agree. Another contribution is that this research is the first to distinguish creators-specific and product-specific significance among videos and images used. Results suggested that the introduction or the storytelling by creators will be more advantageous and influential through videos instead of using images, when it comes to project success. By contrast, using images for showcasing products is a better way to attract investors with higher likelihood for project funding success.

2) How much competitive advantage in the classification power for campaign success comes by using features that are creator-specific and product-specific?

By incorporating discovered knowledge from the first objective, I assimilated a set of features, especially ones with high significance towards project success, to train on twelve different machine learning classifiers. XGBoost model was selected as the best performing classifier among others, which it achieves with the highest accuracy of 72.83% and F1-score of 72.41% in funding classification.

Subsequently, I adopted a novel method with Binary Particle Swarm Optimization (BPSO) for feature selection to further improve classification performance. With five runs of experiments on XGBoost-BPSO model, the best optimal model is improved with accuracy to 74.61% for all three tiers of success classes. The number of features was significantly reduced from 37 to 24 features as well.

My proposed XGBoost integrated Binary Particle Swarm Optimization for feature selection is the first to be used in the context of Kickstarter project success classification. The novel findings shows promising aspect of using BPSO for feature selection with crowdfunding data and serves as a basis for future study on other types of crowdfunding data other than reward-based, such as equity-based. Most important, my research is the first to conduct multiclass classification with Kickstarter data along with the crafted new features being collected in the first objective.

3) Will the addition of textual information with other variables improve classification performance?

To address this objective, NLP and deep learning were implemented with both creator's biography and project's pitch to join the meta data of a campaign, where LSTM, GRU, CNN, and CNN-LSTM were compared against the Naïve Bayes baseline model. The overall results showed that the more information was introduced to the baseline model or any DNNs, the better the classification performance was. When trained on using either project's pitch or creator's biography text input only without any meta information, the Naïve Bayes baseline model outperformed any DNNs. With 50 epochs, LSTM model that trained on both text inputs and meta features yielded the best classification test accuracy of 70.54% for all three classes, and CNN has the least classification performance of 69.10% in accuracy.

After many trials of tuning the hyperparameters, both creator's biography and project's pitch that trained on 200 epochs with a single layer of LSTM (64 hidden units and learning rate=0.00008 with batch size=128) achieves the highest accuracy of 71.04%, when joined with a BPSO selected meta data by a fully connected layer. My findings in this research are in accordance with the reported findings when applied NLP with DNNs for multimodalities. Additionally, this research is the first to utilize creator's biography text information in deep neural networks for multiclass classification.

5.2 Limitations

Aside some of the already discussed limitations at the end of each chapter, several major limitations are encountered in the overall research. The first major limitation is the lack of the verification on the manually collected data. Due to my research being the first to categorize videos and pictures into product relevance or creator relevance, or collecting experience and educational information of creator, no publicly collected data source is available for comparison. Data can only be extracted by human screening and mainly conducted by a single person to ingest information. There were no other verifiers to collect the same information for cross validation to correct or account for human error, or to distinguish systematic or human error for explaining gaps in classification performance.

The second major limitation is the lack of resources to extract more contextual information from the media used. My novel discovery of the significance using videos and images for product aspect and creator aspect only provide a good starting point for the discussion of multimedia impact on project success. However, factors in multimedia can provide information on their association with project success, such as speech from videos or quality of videos.

Besides presentation, technology has improved since the launch of Kickstarter, other than videos and pictures, animations such as GIFs are also now popularly used in many campaigns. In my research framework, I treated them as pictures; however, it might be informative to separate them as an independent indicators to shed light on their impact on project success. Last, my research only focused on factors when projects are first launched and the sample only included completed projects, no dynamic elements such as number of comments or updates were tracked during the campaign period for association on project success. Therefore, some underlying latent variables might not be captured in a live setting after project is launched.

5.3 Future Research

Despite all the discussed limitations, the findings are valuable and promising as a starting point to expand on methods in crowdfunding classification for future studies. My results using the 2019 Kickstarter for multiclass classification are broadly consistent with other studies of binary predictions for Kickstarter. However, further investigations are necessary to validate the conclusions drawn from this study for similar findings with multiclass classification.

Second, future research should assess and test computational methods used on both Kickstarter data and other crowdfunding data such as Indiegogo, to rectify and compare results on key indicators and classification performance for multiclass project success. Kickstarter adopts “All or Nothing” approach, and Indiegogo allows creator to choose between “All or Nothing” or receiving the funds as pledge dollars invested during funding period. Considering both Kickstarter and Indiegogo are both reward-based crowdfunding with a different platform setup, further studies should explore any potential effects on campaign strategies through meta information or text sentiments if applying similar computational modeling framework.

Furthermore, one of the interesting topics for future research will be expanding data scope to 2020 and 2021 data when COVID-19 pandemic hit. Many interesting research questions can be derived from the unexpected pandemic to explore how the campaign strategies of crowdfunding has shifted since 2019 and the association of social behavioral response from project backers or project creators on project success. More research can be done and gained better understanding on this “black swan” events [78].

Last, although a slight improvement has shown when adopted BPSO for feature selection with XGBoost, a different metaheuristic optimization technique in feature selection should also be explored for comparison with the Particle Swarm Optimization, such as Grey Wolf Optimizer [79], Whale Optimization Algorithm [32]. If more computing resources are available, a higher number of simulations epochs should implement to further test and confirm the benefits of BPSO. Another application is to use metaheuristic optimization to select features of multimodalities and integrate them with deep learning algorithms for further expansion on this initial finding.

In conclusion, I have applied novel methods to extract meaningful features along with statistical assessment. Subsequently, I conducted studies on using both classical machine learning and deep learning for project success classification with multimodalities. Findings from my research provide valuable information for a thorough understanding of the new emergent levels of success for project creators to fully optimize the crowdfunding marketing space.

APPENDIX

Appendix A.1: Data Dictionary for Kickstarter Dataset

Variable Name	Data Type	Definition	Used in model for Ch.2	Used in model for Ch.3	Used in model for Ch.4
Success Rate	Float	Ratio of 'Usd_Pledged' to 'Goal'.	x		
Badge	String	Badge status reflects how actively involved creator is in the Kickstarter's community. Types of badges include: n-time Creator, Backer Favorite, Backer Favorite & Superbacker. Badge status is found at creator's profile. According to Kickstarter, to earn the status of Superbacker, one must support more than 25 Kickstarter projects with pledges of at least $10 in the past year.			
Bios	String	Profile description of creator where it shares background, skillsets, or project mission to investors.			x
Bios_wdct	Integer	Total number of words in variable 'Bios'.	x	x	x
Blurb	String	Brief description of the project.			x
Blurb_wdct	Integer	Total number of words in variable 'Blurb'.	x	x	
Button_Flag	Boolean	A button feature used in the project spotlight page for experienced creators to either show case other projects, allow			

Variable Name	Data Type	Definition	Used in model for Ch.2	Used in model for Ch.3	Used in model for Ch.4
		investors to pre-order, or redirect investors to other pages to attract interests. It is encoded to 1 if true where the button exists on campaign page, or 0 if false.			
Cat_Type	String	Encoded categorical variable to represent one of the 16 project categories for the project using frequency encoding.	x	x	x
City	String	City where project is located.			
Degree	String	The highest educational degree of creator mentioned in the project: B=Bachelor, M=Master, P=PhD or MD or JD, NA=None of the above			
Duration	String	Number of funding days from the project launched till the project ended.	x	x	x
Final_Class	String	Classification of the project outcome using ordinal encoding based on the ratio of 'Usd_Pledged' to 'Goal': 'failed'=1, 'successful'=2 and 'super successful'=3. No 'suspended' projects are included.	x	x	x
Friends	Integer	Total number of friends that are within the creator's Facebook network.	x	x	x
Goal	Integer	Total target amount for the project sets by creator.	x	x	x
Id	String	Unique identifier for project. It is used to identify duplicates.			
Img_Creators	Integer	Total number of images (including GIFs) that associate with either creator or team. This is manually collected and	x	x	x

Variable Name	Data Type	Definition	Used in model for Ch.2	Used in model for Ch.3	Used in model for Ch.4
		categorized.			
Img_Prod	Integer	Total number of images (including GIFs) that associate with either creator or team. This is manually collected and categorized.	x	x	x
Is_backerfav	Boolean	A Boolean Y/N flag where creator's badge contains 'backer favorite' status or not. It is encoded to 1 if flag=Y, or 0 if flag=N.	x	x	x
Is_degree	Boolean	A Boolean Y/N flag where creator mentioned educational degree in the project or not. It is encoded to 1 if flag=Y, or 0 if flag=N.	x	x	x
Is_exp	Boolean	A Boolean Y/N flag where creator mentioned relevant experience for the project or not. It is derived from variable 'Year_Exp'. It is encoded to 1 if flag=Y, or 0 if flag=N.	x	x	x
Is_members	Boolean	A Boolean Y/N flag where creator mentioned or listed any team members or not. It is encoded to 1 if flag=Y, or 0 if flag=N.	x	x	x
Joined_Month	Integer	Month when the creator joined Kickstarter.		x	x
Joined_Year	Integer	Year when the creator joined Kickstarter.			
Name	String	A project title.			x
Name_wdct	Integer	Total number of words in variable 'Name'.		x	x
Num_Collabs	Integer	Total number of collaborators on the project.	x	x	x

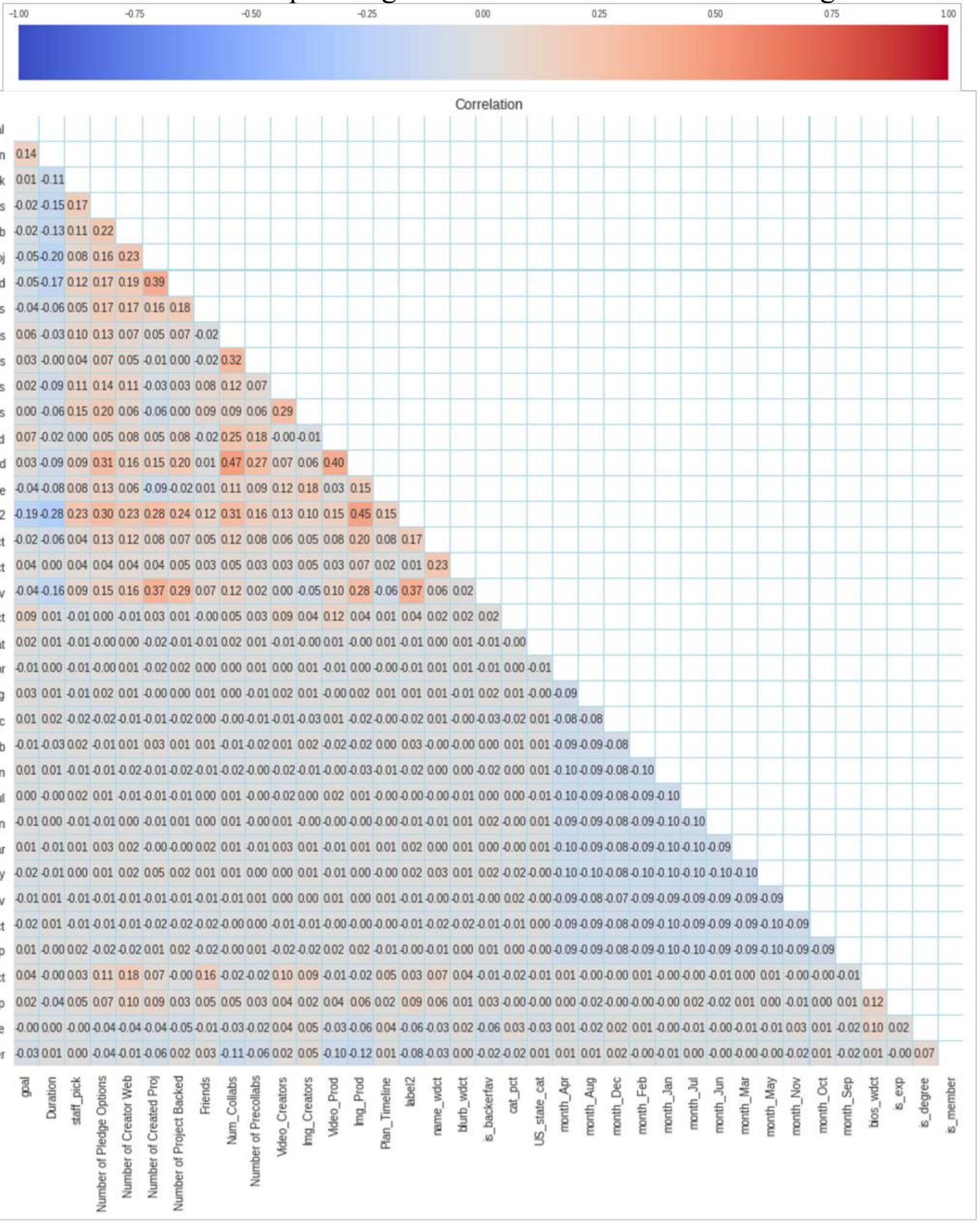
Variable Name	Data Type	Definition	Used in model for Ch.2	Used in model for Ch.3	Used in model for Ch.4
Num_Members	Integer	Total number of team members are listed or mentioned in the project.			
Number Of Created Proj	Integer	Total number of projects created by the creator.	x	x	x
Number Of Creator Web	Integer	Total number of websites listed in creator's profile.	x	x	x
Number Of Pledge Options	Integer	Total number of rewards for investors to support the project.	x	x	x
Number Of Precollabs	Integer	Total number of previous collaborators on the project.		x	x
Number Of Project Backed	Integer	Total number of projects that are supported by the creator.	x	x	x
Orig_Class	String	Binary classification of the project outcome. Only 'failed' and 'successful' projects are included, all 'suspended' projects are excluded in the sample.			
Plan_Timeline	Boolean	A Boolean Y/N flag where creator mentioned an upcoming product plan or deliverable timeline for the project. It is encoded to 1 if flag=Y, or 0 if flag=N.	x	x	x
Staff_Pick	Boolean	A Boolean Y/N flag of the selected project that is labelled as 'project we love' by the Kickstarter's staff. To earn this status, a general guideline is to have a clear project image with thorough description and clear transparency to demonstrate creativity for investors' interest. It is encoded to 1 if flag=Y, or 0 if flag=N.	x	x	x

Variable Name	Data Type	Definition	Used in model for Ch.2	Used in model for Ch.3	Used in model for Ch.4
US_State	String	States where project is located.		x	x
Usd_Pledged	Integer	Total raised amount for the project by investors.			
Video_Creators	Integer	Total number of videos that associate wither either creator or team. This is manually collected and categorized.	x	x	x
Video_Prod	Integer	Total number of videos that associate with either creator or team. This is manually collected and categorized.	x	x	x
Year_Exp	Float	Number of year experience that is relevant to the project as mentioned by creator either in bios or the main content of the project. This is a manually collected variable.			

Appendix A.2: Variance Inflation Factors (VIF) for 21 parameters used in hierarchical multiple regression. Sorted in descending order of VIF value.

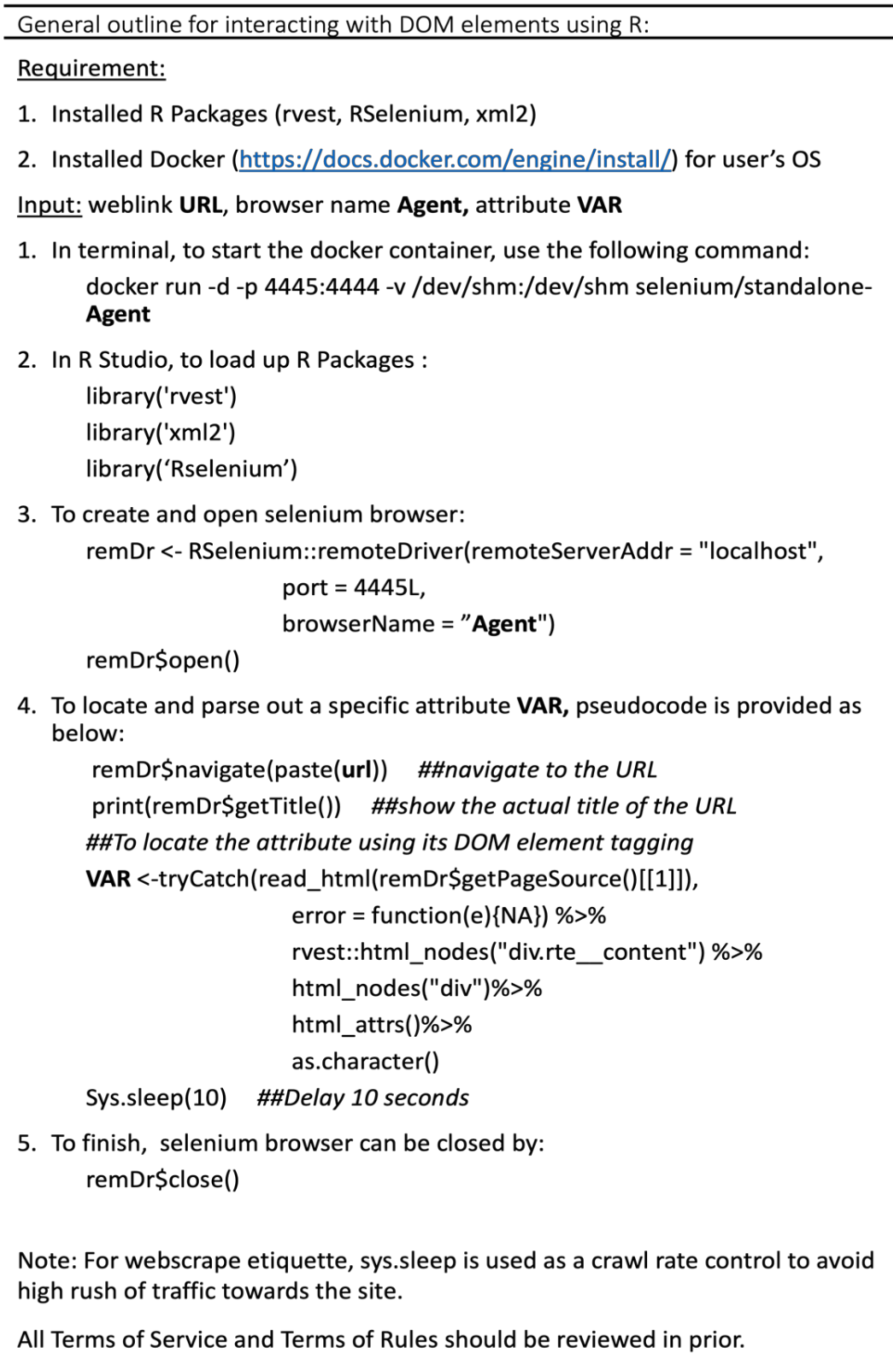
Variables	**VIF**
Img_Prod	1.7146
Number Of Created Proj	1.3849
Num_Collabs	1.3236
Number Of Project Backed	1.2857
Is_Backerfav	1.2753
Number Of Pledge Options	1.2578
Video_Prod	1.2395
Img_Creators	1.1774
Number Of Creator Web	1.1746
Video_Creators	1.1427
Duration	1.1090
Friends	1.1079
Bios_Wdct	1.1051
Plan_Timeline	1.0975
Staff_Pick	1.0784
Goal	1.0443
Is_Member	1.0374
Cat_Type	1.0357
Is_Exp	1.0341
Is_Degree	1.0302
Blurb_Wdct	1.0131

Appendix A.3: Correlation heatmap among variables used in machine learning models.



	goal	Duration	staff_pick	Number of Pledge Options	Number of Creator Web	Number of Created Proj	Number of Project Backed	Friends	Num_Collabs	Number of Precollabs	Video_Creators	Img_Creators	Video_Prod	Img_Prod	Plan_Timeline	label2	name_wdct	blurb_wdct	is_backerfav	cat_pct	US_state_cat	month_Apr	month_Aug	month_Dec	month_Feb	month_Jan	month_Jul	month_Jun	month_Mar	month_May	month_Nov	month_Oct	month_Sep	bios_wdct	is_exp	is_degree	is_member
goal																																					
Duration	0.14																																				
staff_pick	0.01	-0.11																																			
Number of Pledge Options	-0.02	-0.15	0.17																																		
Number of Creator Web	-0.02	-0.13	0.11	0.22																																	
Number of Created Proj	-0.05	-0.20	0.08	0.16	0.23																																
Number of Project Backed	-0.05	-0.17	0.12	0.17	0.19	0.39																															
Friends	-0.04	-0.06	0.05	0.17	0.17	0.16	0.18																														
Num_Collabs	0.06	-0.03	0.10	0.13	0.07	0.05	0.07	-0.02																													
Number of Precollabs	0.03	-0.00	0.04	0.07	0.05	-0.01	0.00	-0.02	0.32																												
Video_Creators	0.02	-0.09	0.11	0.14	0.11	-0.03	0.03	0.08	0.12	0.07																											
Img_Creators	0.00	-0.06	0.15	0.20	0.06	-0.06	0.00	0.09	0.09	0.06	0.29																										
Video_Prod	0.07	-0.02	0.00	0.05	0.08	0.05	0.08	-0.02	0.25	0.18	-0.00	-0.01																									
Img_Prod	0.03	-0.09	0.09	0.31	0.16	0.15	0.20	0.01	0.47	0.27	0.07	0.06	0.40																								
Plan_Timeline	-0.04	-0.08	0.08	0.13	0.06	-0.09	-0.02	0.01	0.11	0.09	0.12	0.18	0.03	0.15																							
label2	-0.19	-0.28	0.23	0.30	0.23	0.28	0.24	0.12	0.31	0.16	0.13	0.10	0.15	0.45	0.15																						
name_wdct	-0.02	-0.06	0.04	0.13	0.12	0.08	0.07	0.05	0.12	0.08	0.06	0.05	0.08	0.20	0.08	0.17																					
blurb_wdct	0.04	0.00	0.04	0.04	0.04	0.04	0.05	0.03	0.05	0.03	0.03	0.05	0.03	0.07	0.02	0.01	0.23																				
is_backerfav	-0.04	-0.16	0.09	0.15	0.16	0.37	0.29	0.07	0.12	0.02	0.00	-0.05	0.10	0.28	-0.06	0.37	0.06	0.02																			
cat_pct	0.09	0.01	-0.01	0.00	-0.01	0.03	0.01	-0.00	0.05	0.03	0.09	0.04	0.12	0.04	0.01	0.04	0.02	0.02	0.02																		
US_state_cat	0.02	0.01	-0.01	-0.00	0.00	-0.02	-0.01	-0.01	0.02	0.01	-0.01	-0.00	0.01	-0.00	0.01	-0.01	0.00	0.01	-0.01	-0.00																	
month_Apr	-0.01	0.00	-0.01	-0.00	0.01	-0.02	0.02	0.00	0.00	0.01	0.00	0.01	-0.01	0.00	-0.00	-0.01	0.01	0.01	-0.01	0.00	-0.01																
month_Aug	0.03	0.01	-0.01	0.02	0.01	-0.00	0.00	0.01	0.00	-0.01	0.02	0.01	-0.00	0.02	0.01	0.01	0.01	-0.01	0.02	0.01	-0.00	-0.09															
month_Dec	0.01	0.02	-0.02	-0.02	-0.01	-0.01	-0.02	0.00	-0.00	-0.01	-0.01	-0.03	0.01	-0.02	-0.00	-0.02	0.01	-0.00	-0.03	-0.02	0.01	-0.08	-0.08														
month_Feb	-0.01	-0.03	0.02	-0.01	0.01	0.03	0.01	0.01	-0.01	-0.02	0.01	0.02	-0.02	-0.02	0.00	0.03	-0.00	-0.00	0.00	0.01	0.01	-0.09	-0.09	-0.08													
month_Jan	0.01	0.01	-0.01	-0.01	-0.02	-0.01	-0.02	-0.01	-0.02	-0.00	-0.02	-0.01	-0.00	-0.03	-0.01	-0.02	0.00	0.00	-0.02	0.00	0.01	-0.10	-0.09	-0.08	-0.10												
month_Jul	0.00	-0.00	0.02	0.01	-0.01	-0.01	-0.01	0.00	0.01	-0.00	-0.02	0.00	0.02	0.01	-0.00	-0.00	-0.00	-0.01	0.00	0.00	-0.01	-0.10	-0.09	-0.08	-0.09	-0.10											
month_Jun	-0.01	0.00	-0.01	-0.01	0.00	-0.01	0.01	0.00	0.01	-0.00	0.01	-0.00	-0.00	-0.00	-0.00	-0.01	-0.01	0.01	0.02	-0.00	0.01	-0.09	-0.09	-0.08	-0.09	-0.10	-0.10										
month_Mar	0.01	-0.01	0.01	0.03	0.02	-0.00	-0.00	0.02	0.01	-0.01	0.03	0.01	-0.01	0.01	0.01	0.02	0.00	0.01	0.00	-0.00	0.01	-0.10	-0.09	-0.08	-0.09	-0.10	-0.10	-0.09									
month_May	-0.02	-0.01	0.00	0.01	0.02	0.05	0.02	0.01	0.01	0.00	0.00	0.01	-0.01	0.00	-0.00	0.02	0.03	0.01	0.02	-0.02	-0.00	-0.10	-0.10	-0.08	-0.10	-0.10	-0.10	-0.10	-0.10								
month_Nov	-0.01	0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	0.01	0.00	0.00	0.01	0.00	0.01	-0.01	-0.00	-0.01	-0.00	0.02	-0.00	-0.09	-0.08	-0.07	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09							
month_Oct	-0.02	0.01	-0.01	-0.01	-0.01	-0.02	-0.02	-0.02	-0.00	0.00	-0.01	-0.01	-0.00	-0.00	-0.00	-0.01	-0.02	-0.02	-0.01	-0.01	0.00	-0.09	-0.09	-0.08	-0.09	-0.10	-0.09	-0.09	-0.09	-0.10	-0.09						
month_Sep	0.01	-0.00	0.02	-0.02	-0.02	0.01	0.02	-0.02	-0.00	0.01	-0.02	-0.02	0.02	0.02	-0.01	-0.00	-0.01	0.00	0.01	0.00	-0.00	-0.09	-0.09	-0.08	-0.09	-0.10	-0.10	-0.09	-0.09	-0.10	-0.09	-0.09					
bios_wdct	0.04	-0.00	0.03	0.11	0.18	0.07	-0.00	0.16	-0.02	-0.02	0.10	0.09	-0.01	-0.02	0.05	0.03	0.07	0.04	-0.01	-0.02	-0.01	0.01	-0.00	-0.00	0.01	-0.00	-0.00	-0.01	0.00	0.01	-0.00	-0.00	-0.01				
is_exp	0.02	-0.04	0.05	0.07	0.10	0.09	0.03	0.05	0.05	0.03	0.04	0.02	0.04	0.06	0.02	0.09	0.06	0.01	0.03	-0.00	-0.00	0.00	-0.02	-0.00	-0.00	-0.00	0.02	-0.02	0.01	0.00	-0.01	0.00	0.01	0.12			
is_degree	-0.00	0.00	-0.00	-0.04	-0.04	-0.04	-0.05	-0.01	-0.03	-0.02	0.04	0.05	-0.03	-0.06	0.04	-0.06	-0.03	0.02	-0.06	0.03	-0.03	0.01	-0.02	0.02	0.01	-0.00	-0.01	-0.00	-0.01	-0.01	0.03	0.01	-0.02	0.10	0.02		
is_member	-0.03	0.01	0.00	-0.04	-0.01	-0.06	0.02	0.03	-0.11	-0.06	0.02	0.05	-0.10	-0.12	0.01	-0.08	-0.03	0.00	-0.02	-0.02	0.01	0.01	0.01	0.02	-0.00	-0.01	0.00	-0.00	-0.00	-0.00	-0.02	0.01	-0.02	0.01	-0.00	0.07	

Appendix A.4: General outline of using R to extract DOM elements



General outline for interacting with DOM elements using R:

Requirement:

1. Installed R Packages (rvest, RSelenium, xml2)
2. Installed Docker (https://docs.docker.com/engine/install/) for user’s OS

Input: weblink **URL**, browser name **Agent,** attribute **VAR**

1. In terminal, to start the docker container, use the following command:

```
docker run -d -p 4445:4444 -v /dev/shm:/dev/shm selenium/standalone-
Agent
```

2. In R Studio, to load up R Packages :

```
library('rvest')
library('xml2')
library(‘Rselenium’)
```

3. To create and open selenium browser:

```
remDr <- RSelenium::remoteDriver(remoteServerAddr = "localhost",
                                 port = 4445L,
                                 browserName = ”Agent")
remDr$open()
```

4. To locate and parse out a specific attribute **VAR,** pseudocode is provided as below:

```
 remDr$navigate(paste(url))    ##navigate to the URL
 print(remDr$getTitle())    ##show the actual title of the URL
##To locate the attribute using its DOM element tagging
VAR <-tryCatch(read_html(remDr$getPageSource()[[1]]),
                    error = function(e){NA}) %>%
                    rvest::html_nodes("div.rte__content") %>%
                    html_nodes("div")%>%
                    html_attrs()%>%
                    as.character()
Sys.sleep(10)    ##Delay 10 seconds
```

5. To finish, selenium browser can be closed by:

```
remDr$close()
```

Note: For webscrape etiquette, sys.sleep is used as a crawl rate control to avoid high rush of traffic towards the site.

All Terms of Service and Terms of Rules should be reviewed in prior.

REFERENCES

[1] J. Pedersen *et al.*, “Conceptual Foundations of Crowdsourcing: A Review of IS Research,” in *2013 46th Hawaii International Conference on System Sciences*, Wailea, HI, USA, Jan. 2013, pp. 579–588. doi: 10.1109/HICSS.2013.143.

[2] C. S. H. Hoi, D. Khowaja, and C. K. Leung, “Constrained Frequent Pattern Mining from Big Data Via Crowdsourcing,” in *Big Data Applications and Services 2017*, vol. 770, W. Lee and C. K. Leung, Eds. Singapore: Springer Singapore, 2019, pp. 69–79. doi: 10.1007/978-981-13-0695-2_9.

[3] R. Abhyanker, “Startup Funding: 8 Best Ways To Raise Capital,” *Business Class: Trends and Insights | American Express*. https://www.americanexpress.com/en-us/business/trends-and-insights/articles/startup-funding-8-best-ways-to-raise-capital/ (accessed Mar. 01, 2021).

[4] T. Tetreault, “10 Eye-opening Crowdfunding Statistics 2020,” *Fit Small Business*, Jan. 09, 2020. https://fitsmallbusiness.com/crowdfunding-statistics/ (accessed Mar. 02, 2021).

[5] Securities and Exchange Commission, “Small Entity Compliance Guide - Regulation Crowdfunding: A Small Entity Compliance Guide for Crowdfunding Intermediaries.” https://www.sec.gov/divisions/marketreg/tmcompliance/cfintermediaryguide.htm#_ftnref3 (accessed Mar. 01, 2021).

[6] “SEC.gov | Investor Bulletin: Crowdfunding Investment Limits Increase.” https://www.sec.gov/oiea/investor-alerts-and-bulletins/ib_crowdfundingincrease (accessed Mar. 01, 2021).

[7] J. Hemer, “A snapshot on crowdfunding,” Fraunhofer Institute for Systems and Innovation Research (ISI), Working Papers “Firms and Region” R2/2011, 2011. Accessed: Mar. 11, 2021. [Online]. Available: https://econpapers.repec.org/paper/zbwfisifr/r22011.htm

[8] “Press — Kickstarter.” https://www.kickstarter.com/press?ref=about_subnav (accessed Mar. 06, 2021).

[9] C. Baldwin and E. von Hippel, “Modeling a Paradigm Shift: From Producer Innovation to User and Open Collaborative Innovation,” *Organ. Sci.*, vol. 22, no. 6, pp. 1399–1417, 2011.

[10] E. Mollick, “The dynamics of crowdfunding: An exploratory study,” *J. Bus. Ventur.*, vol. 29, no. 1, pp. 1–16, Jan. 2014, doi: 10.1016/j.jbusvent.2013.06.005.

[11] P. Crosetto and T. Regner, “Crowdfunding: Determinants of success and funding dynamics,” Jena Economic Research Papers, Working Paper 2014–035, 2014. Accessed: Mar. 12, 2021. [Online]. Available: https://www.econstor.eu/handle/10419/108542

[12] D. Frydrych, A. J. Bock, T. Kinder, and B. Koeck, “Exploring entrepreneurial legitimacy in reward-based crowdfunding,” *Venture Cap.*, vol. 16, no. 3, pp. 247–269, Jul. 2014, doi: 10.1080/13691066.2014.916512.

[13] A. Fernandez-Blanco, J. Balsera, V. Montequín, and H. Morán Palacios, “Key Factors for Project Crowdfunding Success: An Empirical Study,” *Sustainability*, vol. 12, p. 599, Jan. 2020, doi: 10.3390/su12020599.

[14] H. Zhao *et al.*, “Tracking the Dynamics in Crowdfunding,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, Aug. 2017, pp. 625–634. doi: 10.1145/3097983.3098030.

[15] A. Ralcheva and P. Roosenboom, “Forecasting success in equity crowdfunding,” *Small Bus. Econ.*, vol. 55, no. 1, pp. 39–56, Jun. 2020, doi: 10.1007/s11187-019-00144-x.

[16] “Kickstarter Datasets,” *Web Scraping Service*. https://webrobots.io/kickstarter-datasets/ (accessed Mar. 15, 2021).

[17] J. W. Osborne and E. Waters, “Four assumptions of multiple regression that researchers should always test,” *Pract. Assess. Res. Eval.*, vol. 8, no. 2, Jan. 2002, doi: 10.7275/R222-HV23.

[18] M. Zhou, B. Lu, W. Fan, and G. A. Wang, “Project description and crowdfunding success: an exploratory study,” *Inf. Syst. Front.*, vol. 20, no. 2, pp. 259–274, Apr. 2018, doi: 10.1007/s10796-016-9723-1.

[19] J.-A. Koch and M. Siering, “Crowdfunding Success Factors: The Characteristics of Successfully Funded Projects on Crowdfunding Platforms,” Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 2808424, Apr. 2015. Accessed: Jan. 24, 2021. [Online]. Available: https://papers.ssrn.com/abstract=2808424

[20] V. Etter, M. Grossglauser, and P. Thiran, “Launch hard or go home!: predicting the success of kickstarter campaigns,” in *Proceedings of the first ACM conference on Online social networks - COSN ’13*, Boston, Massachusetts, USA, 2013, pp. 177–182. doi: 10.1145/2512938.2512957.

[21] S.-Y. Chen, C.-N. Chen, Y.-R. Chen, C.-W. Yang, W.-C. Lin, and C.-P. Wei, “Will Your Project Get the Green Light? Predicting the Success of Crowdfunding Campaigns,” 2015, Accessed: May 22, 2021. [Online]. Available: https://aisel.aisnet.org/pacis2015/79

[22] P.-F. Yu, F.-M. Huang, C. Yang, Y.-H. Liu, Z.-Y. Li, and C.-H. Tsai, “Prediction of Crowdfunding Project Success with Deep Learning,” in *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*, Oct. 2018, pp. 1–8. doi: 10.1109/ICEBE.2018.00012.

[23] W. Wang, H. Zheng, and Y. J. Wu, “Prediction of fundraising outcomes for crowdfunding projects based on deep learning: a multimodel comparative study,” *Soft Comput.*, vol. 24, no. 11, pp. 8323–8341, Jun. 2020, doi: 10.1007/s00500-020-04822-x.

[24] S. Jhaveri, I. Khedkar, Y. Kantharia, and S. Jaswal, “Success Prediction using Random Forest, CatBoost, XGBoost and AdaBoost for Kickstarter Campaigns,” in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Mar. 2019, pp. 1170–1173. doi: 10.1109/ICCMC.2019.8819828.

[25] M. D. Greenberg, B. Pardo, K. Hariharan, and E. Gerber, “Crowdfunding support tools: predicting success & failure,” in *CHI ’13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA ’13*, Paris, France, 2013, pp. 1815–1820. doi: 10.1145/2468356.2468682.

[26] V. Rakesh, J. Choo, and C. K. Reddy, “Project Recommendation Using Heterogeneous Traits in Crowdfunding,” *Proc. Int. AAAI Conf. Web Soc. Media*, vol. 9, no. 1, Art. no. 1, Apr. 2015, Accessed: May 28, 2021. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14624

[27] V. Sharma and K. Lee, “Predicting Highly Rated Crowdfunded Products,” in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Barcelona, Aug. 2018, pp. 357–362. doi: 10.1109/ASONAM.2018.8508797.

[28] Y. Li, V. Rakesh, and C. K. Reddy, “Project Success Prediction in Crowdfunding Environments,” in *Proceedings of the Ninth ACM International Conference on Web*

Search and Data Mining, San Francisco California USA, Feb. 2016, pp. 247–256. doi: 10.1145/2835776.2835791.

[29] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, no. null, pp. 1157–1182, Mar. 2003.

[30] M.-Y. Chen, J.-R. Chang, L.-S. Chen, and E.-L. Shen, “The key successful factors of video and mobile game crowdfunding projects using a lexicon-based feature selection approach,” *J. Ambient Intell. Humaniz. Comput.*, Mar. 2021, doi: 10.1007/s12652-021-03146-4.

[31] M. J. Ryoba, S. Qu, and Y. Zhou, “Feature subset selection for predicting the success of crowdfunding project campaigns,” *Electron. Mark.*, Jan. 2020, doi: 10.1007/s12525-020-00398-4.

[32] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[33] M. Mafarja and S. Mirjalili, “Whale optimization approaches for wrapper feature selection,” *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018, doi: 10.1016/j.asoc.2017.11.006.

[34] M. J. Ryoba, S. Qu, Y. Ji, and D. Qu, “The Right Time for Crowd Communication during Campaigns for Sustainable Success of Crowdfunding: Evidence from Kickstarter Platform,” *Sustainability*, vol. 12, no. 18, p. 7642, Sep. 2020, doi: 10.3390/su12187642.

[35] S. Geng, M. Huang, and Z. Wang, “A Swarm Enhanced Light Gradient Boosting Machine for Crowdfunding Project Outcome Prediction,” in *Machine Learning for Cyber Security*, Cham, 2020, pp. 372–382.

[36] K. Korovkinas, P. Danėnas, and G. Garšva, “Support vector machine parameter tuning based on particle swarm optimization metaheuristic,” *Nonlinear Anal. Model. Control*, vol. 25, no. 2, Mar. 2020, doi: 10.15388/namc.2020.25.16517.

[37] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[38] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, “An Introduction to Logistic Regression Analysis and Reporting,” *J. Educ. Res.*, vol. 96, no. 1, pp. 3–14, Sep. 2002, doi: 10.1080/00220670209598786.

[39] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, “An introduction to decision tree modeling,” *J. Chemom.*, vol. 18, no. 6, pp. 275–285, Jun. 2004, doi: 10.1002/cem.873.

[40] D. D. Lewis, “Naive (Bayes) at forty: The independence assumption in information retrieval,” in *Machine Learning: ECML-98*, vol. 1398, C. Nédellec and C. Rouveirol, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 4–15. doi: 10.1007/BFb0026666.

[41] F. AlOmari and G. Liu, “Analysis of Extracted Forearm sEMG Signal Using LDA, QDA, K-NN Classification Algorithms,” *Open Autom. Control Syst. J.*, vol. 6, no. 1, pp. 108–116, Jul. 2014, doi: 10.2174/1874444301406010108.

[42] V. Kecman, “Support Vector Machines – An Introduction,” in *Support Vector Machines: Theory and Applications*, vol. 177, L. Wang, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1–47. doi: 10.1007/10984697_1.

[43] F. Murtagh, “Multilayer perceptrons for classification and regression,” *Neurocomputing*, vol. 2, no. 5–6, pp. 183–197, Jul. 1991, doi: 10.1016/0925-2312(91)90023-5.

[44] D. Opitz and R. Maclin, “Popular Ensemble Methods: An Empirical Study,” *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, Aug. 1999, doi: 10.1613/jair.614.

[45] T. Hastie, S. Rosset, J. Zhu, and H. Zou, “Multi-class AdaBoost,” *Stat. Interface*, vol. 2, no. 3, pp. 349–360, 2009, doi: 10.4310/SII.2009.v2.n3.a8.

[46] E. A. Daoud, “Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset,” Jan. 2019, doi: 10.5281/ZENODO.3607805.

[47] X. Ren, H. Guo, S. Li, S. Wang, and J. Li, “A Novel Image Classification Method with CNN-XGBoost Model,” in *Digital Forensics and Watermarking*, vol. 10431, C. Kraetzer, Y.-Q. Shi, J. Dittmann, and H. J. Kim, Eds. Cham: Springer International Publishing, 2017, pp. 378–390. doi: 10.1007/978-3-319-64185-0_28.

[48] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[49] A. Omar and K. Belkhayat, “XGBoost and LGBM for Porto Seguro’s Kaggle challenge: A comparison.” Preprint Semester Project, Jan. 20, 2018.

[50] G. Ke *et al.*, “LightGBM: a highly efficient gradient boosting decision tree,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, Dec. 2017, pp. 3149–3157.

[51] J. Li *et al.*, “Feature Selection: A Data Perspective,” *ACM Comput. Surv.*, vol. 50, no. 6, p. 94:1-94:45, Dec. 2017, doi: 10.1145/3136625.

[52] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. Morgan Kaufmann, 2001.

[53] B. Tran, B. Xue, and M. Zhang, “Overview of Particle Swarm Optimisation for Feature Selection in Classification,” in *Simulated Evolution and Learning*, Cham, 2014, pp. 605–617. doi: 10.1007/978-3-319-13563-2_51.

[54] W. Hu and R. Yang, “Predicting the Success of Kickstarter Projects in the US at Launch Time,” in *Intelligent Systems and Applications*, Cham, 2020, pp. 497–506. doi: 10.1007/978-3-030-29516-5_39.

[55] T. Mitra and E. Gilbert, “The language that gets people to give: phrases that predict success on kickstarter,” in *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, New York, NY, USA, Feb. 2014, pp. 49–61. doi: 10.1145/2531602.2531656.

[56] Q. Du, W. Fan, Z. Qiao, G. Wang, X. Zhang, and M. Zhou, “Money Talks: A Predictive Model on Crowdfunding Success Using Project Description,” *AMCIS 2015 Proc.*, Jun. 2015, [Online]. Available: https://aisel.aisnet.org/amcis2015/BizAnalytics/GeneralPresentations/37

[57] K. Sawhney, C. Tran, and R. Tuason, “Using Language to Predict Kickstarter Success,” p. 9.

[58] W. Wang, K. Zhu, H. Wang, and Y.-C. J. Wu, “The Impact of Sentiment Orientations on Successful Crowdfunding Campaigns through Text Analytics,” *IET Softw.*, vol. 11, no. 5, pp. 229–238, Oct. 2017, doi: 10.1049/iet-sen.2016.0295.

[59] L. Silva, N. F. Silva, and T. Rosa, “Success prediction of crowdfunding campaigns: a two-phase modeling,” *Int. J. Web Inf. Syst.*, vol. 16, no. 4, pp. 387–412, Jul. 2020, doi: 10.1108/IJWIS-05-2020-0026.

[60] S. Faralli, S. Rittinghaus, N. Samsami, D. Distante, and E. Rocha, “Emotional Intensity-based Success Prediction Model for Crowdfunded Campaigns,” *Inf. Process. Manag.*, vol. 58, no. 1, p. 102394, Jan. 2021, doi: 10.1016/j.ipm.2020.102394.

[61] S. Lee, K. Lee, and H. Kim, “Content-based Success Prediction of Crowdfunding Campaigns: A Deep Learning Approach,” in *Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing*, Jersey City NJ USA, Oct. 2018, pp. 193–196. doi: 10.1145/3272973.3274053.

[62] E. Katsamakas and H. Sun, “Machine Learning Crowdfunding:,” *Int. J. Knowl.-Based Organ.*, vol. 10, no. 2, pp. 1–11, Apr. 2020, doi: 10.4018/IJKBO.2020040101.

[63] G. Zhou, “Research on text summarization classification based on crowdfunding projects,” *MATEC Web Conf.*, vol. 336, p. 06020, 2021, doi: 10.1051/matecconf/202133606020.

[64] J. Kim, D. Cho, and B. Lee, “The Mind Behind Crowdfunding: An Empirical Study of Speech Emotion in Fundraising Success,” *ICIS 2016 Proc.*, Dec. 2016, [Online]. Available: https://aisel.aisnet.org/icis2016/Crowdsourcing/Presentations/23

[65] M. Raab, S. Schlauderer, S. Overhage, and T. Friedrich, “More than a feeling: Investigating the contagious effect of facial emotional expressions on investment decisions in reward-based crowdfunding,” *Decis. Support Syst.*, vol. 135, p. 113326, Aug. 2020, doi: 10.1016/j.dss.2020.113326.

[66] C. Cheng, F. Tan, X. Hou, and Z. Wei, “Success Prediction on Crowdfunding with Multimodal Deep Learning,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, China, Aug. 2019, pp. 2158–2164. doi: 10.24963/ijcai.2019/299.

[67] J. C. Kaminski and C. Hopp, “Predicting outcomes in crowdfunding campaigns with textual, visual, and linguistic signals,” *Small Bus. Econ.*, vol. 55, no. 3, pp. 627–649, Oct. 2020, doi: 10.1007/s11187-019-00218-w.

[68] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, Dec. 2014, doi: 10.1016/j.asej.2014.04.011.

[69] Y. Zhang, N. Lin, and S. Jiang, “A Study on Syntactic Complexity and Text Readability of ASEAN English News,” in *2019 International Conference on Asian Language Processing (IALP)*, Shanghai, Singapore, Nov. 2019, pp. 313–318. doi: 10.1109/IALP48816.2019.9037695.

[70] J. Pennington, R. Socher, and C. Manning, “Glove: Global Vectors for Word Representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162.

[71] R. M. Schmidt, "Recurrent Neural Networks (RNNs): A gentle Introduction and Overview," *ArXiv191205911 Cs Stat*, Nov. 2019, Accessed: Aug. 19, 2021. [Online]. Available: http://arxiv.org/abs/1912.05911

[72] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[73] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *ArXiv14061078 Cs Stat*, Sep. 2014, Accessed: Aug. 20, 2021. [Online]. Available: http://arxiv.org/abs/1406.1078

[74] Y. Lecun *et al.*, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural Netw. Stat. Mech. Perspect.*, pp. 261–276, 1995.

[75] Y. Lecun and Y. Bengio, "Convolutional networks for images, speech, and time-series," *Handb. Brain Theory Neural Netw.*, 1995, Accessed: Aug. 21, 2021. [Online]. Available: https://nyuscholars.nyu.edu/en/publications/convolutional-networks-for-images-speech-and-time-series

[76] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning--based Text Classification: A Comprehensive Review," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–40, Jun. 2021, doi: 10.1145/3439726.

[77] Y. Li, M. Ye, and Q. Hu, "HCapsNet: A Text Classification Model Based on Hierarchical Capsule Network," in *Knowledge Science, Engineering and Management*, vol. 12816, H. Qiu, C. Zhang, Z. Fei, M. Qiu, and S.-Y. Kung, Eds. Cham: Springer International Publishing, 2021, pp. 538–549. doi: 10.1007/978-3-030-82147-0_44.

[78] N. N. Taleb, *The Black Swan: Second Edition: The Impact of the Highly Improbable: With a new section: "On Robustness and Fragility,"* 2nd ed. edition. New York: Random House Publishing Group, 2010.

[79] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

BIOGRAPHY

Sze Wing Wong currently works at the Securities and Exchange Commission as a Senior Data Scientist. In this role she supports the Office of the Chief Data Officer for data research in enforcement, examinations, and policymaking. Previously she worked for Noblis in support of IRS insider threat analytics. Before Noblis, she also worked as a Senior Technology Analyst and a Mathematical Statistician at the Federal Reserve Board of Governors and the U.S Census Bureau respectively. She received her Master's degree in Statistics from University of New Mexico, and her Bachelor's degree in Biology from the New Mexico Institute of Mining and Technology. She is expecting to earn her Doctor of Philosophy in Computational Science and Informatics from George Mason University in 2021.