Department of Computer Science,
The University of Hong Kong

CrowdInsight: An AI Platform for Predicting and Explaining Kickstarter Success

Fung Angus, Li Ka Ho, Qian Yongkun Jonathan

Interim Report, @FYP24073

Supervisor: Dr Chow Ka Ho

January 26, 2025

Abstract

Crowdfunding enables small businesses and startups with creative ideas to secure the capital needed to bring them to life. While crowdfunding platforms have become popular and widespread, the success rate of crowdfunding campaigns remains relatively low. Therefore, there is a need for a platform to predict the likelihood of campaign success. This project develops a platform named CrowdInsight, designed to predict and explain the success rates of campaigns on Kickstarter. It leverages a multimodal neural network to enhance prediction accuracy while utilizing explainable artificial intelligence techniques, specifically Deep Shapley Additive exPlanations and few-shot prompting, to provide interpretable results. A user-friendly web application will offer campaign creators clear insights into both the predictions and their explanations. The project is progressing on schedule. So far, the outcomes include feature engineering of data collected from Kickstarter and the completion of the website insights module. The next step involves training the multimodal neural network to predict campaign success rates.

Acknowledgments

First, we would like to express our heartfelt gratitude to the Department of Computer Science and, in particular, our supervisor, Dr. Chow Ka Ho, for providing us with the opportunity to undertake this research. Dr. Chow's expert guidance, constructive feedback, and unwavering support have been instrumental in shaping the direction and quality of our work. His insightful advice and encouragement throughout the research process have been invaluable, and we are sincerely thankful for his mentorship.

We would also like to acknowledge the dedication and efforts of our teammates, whose collaboration and hard work have been essential in achieving our project goals.

Lastly, we extend our thanks to Miss Mable Choi for her support in managing progress reports. Her timely advice, responsiveness, and attention to detail were crucial in ensuring smooth project execution.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
ASCII	American Standard Code for Information Interchange
API	Application Programming Interface
FAISS	Facebook AI Similarity Search
GUI	Graphic User Interface
ICPSR	Inter-University Consortium for Political and Social Research
IP	Internet Protocol
JSON	JavaScript Object Notation
KNN	K-Nearest Neighbours
LLM	Large Language Model
MOE	Margin of Error
MSE	Mean Square Error
OCR	Optical Character Recognition
RAG	Retrieval-Augmented Generation
RF	Random Forest
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
UI	User Interface
URL	Uniform Resource Locator
XAI	Explainable Artificial Intelligence

Chapter 1

Introduction

1.1 Overview of Crowdfunding

Nowadays, small businesses and startups might struggle to obtain capital through traditional institutions such as banks due to their lack of collateral. As such, they fell back on crowdfunding platforms to finance their ideas (Camilleri & Bresciani, 2022). Crowdfunding refers to the alternative model for project financing, where a broad and diverse audience participates through relatively small financial contributions, in return for tangible, financial, or social rewards (Alegre & Moleskis, 2016). Crowdfunding eliminates certain stringent requirements such as the need for collateral or credit scoring associated with traditional funding sources, positioning it as an ideal option for small startups or individuals. As such, the emergence of crowdfunding platforms, Kickstarter for instance, has radically altered the mechanisms by which innovators raise money. Crowdfunding can be classified into four categories: reward-based, social/donation-based, equity-based, and debt-based (Wangchuk, 2021). Among them, reward-based crowdfunding stands as the most popular form available (Fundable, 2014). In reward-based systems, backers receive a reward, which could be the product promised, a chance to involve in the development of the product, or a token of appreciation (Cumming & Johan, 2020).

However, the number of successfully funded projects leaves room for improvement, the success rate of campaigns on Kickstarter is 41.98% (Kickstarter, 2025), for instance. This underscores the insecurity and challenges those creators aiming to draw backers face. In addition, the dearth of knowledge & consciousness to package and market the campaigns could be the reasons leading to the low success rate. Many factors could be contributing to an accomplished campaign. For instance, Vesterlund (2003) illustrates that the unveiling of an entrepreneur's previously received contributions to potential backers could benefit the entrepreneur's reputation, as this could be an indication of project quality. Thus, it is pivotal for creators to know what is missing when preparing for a campaign. Nonetheless, they lack access to a platform that can predict their chances of success and provide suggestions on areas of improvement.

Numerous crowdfunding platforms, such as Kickstarter, which this project focuses on, state that creators will not receive the amount raised unless the funding goal is reached. In other words, they use an "all or nothing" strategy (Robertson & Wooster, 2015), which highlights the need for educating more entrepreneurs on the factors that influence the effectiveness of their campaigns. Research has shown that crowdfunding contributes to digital financial inclusion (Halim, 2024) and the transformation

of our society into a sustainable society (Testa et al., 2018), while boosting entrepreneurship and fueling economic and societal growth. Given the above problem, there exists a demand for a platform that guides creators to launch a successful campaign.

1.2 Gaps in Existing Work

There have been similar research efforts concentrated on predicting the success of crowdfunding campaigns. Methods leveraged include survival analysis, Ordinary Least Square, Discriminant Analysis, Hierarchical Multiple Regression, Negative Binomial Regression and Machine Learning (Cavalcanti et al., 2024). Among the machine learning techniques, for instance, Kaminski and Hopp (2019) utilized multimodal AI and data samples from text, speech, and video content. Feng et al. (2024) discovered that feature selection could significantly improve prediction performance, and tested out algorithms such as k-nearest neighbours (KNN), support vector machine (SVM), random forest (RF), etc.

Many studies have employed machine learning; that being said, there are still limitations in them. For instance, Kaminski and Hopp (2019) restricted their data sample to campaigns only in the categories of "Product Design" and "Technology". Also, it is noticed that few studies have emphasized on techniques such as Large Language Models (LLM) and Explainable Artificial Intelligence (XAI), including methods like Deep Shapley Additive exPlanations (DeepSHAP).

1.3 Major Pain Points

There exist several shortcomings to the original Kickstarter website and the available research on predicting the success of crowdfunding.

1.3.1 Lack of Insights

Campaign creators often lack information about ideal funding goals and emerging trends, and therefore, no actionable insights are provided to them. For instance, most reports, such as those from Statista and SearchLogistics, only present the total number of projects and funds raised over the years, but fail to offer guidance on how to set up a successful campaign.

1.3.2 Shortcomings of Filtering Tools

The existing filtering tools available on the Kickstarter website have significant limitations. For instance, users can only filter by broad base categories, and further filtering by subcategories is not allowed when they want to specifically locate particular niche areas. Similarly, when users want to filter campaigns by funding goal and pledged amount, they can only choose from predefined ranges (such as <$1000, $1000–$10,000, etc.), meaning they cannot set their own preferred range, which hinders the process of finding related campaigns. Furthermore, there is a lack of a customizable timeframe, preventing users from screening out campaigns within their designated time range. Instead, they are shown projects from

all time periods. Finally, users are only allowed to select live, successful, or upcoming projects, but not failed ones, which means they cannot learn from the mistakes of failed campaigns.

1.3.3 Uncertainty in Outcomes

Existing models only predict outcomes after the campaign went live, such that they focus on post-launch dynamics. For instance, Zhao et al. (2017) emphasized post-launch factors such as backers, comments, and pledged fund dynamics, Etter et al. (2013) focused on social network activity and early funding trends for prediction. They offer little pre-launch guidance.

The only similar website in terms of functions, Jumpstart3r (2025), however, only uses basic machine learning models such as RF, and is currently non-functional, reflecting the need for accessible pre-launch solutions.

1.3.4 Lack of Feedback

It is noticed that few studies have emphasized the explainability issue of predictions and failed to provide practical advice. Their output is often limited to numeric success predictions without offering guidance for improvement, thus hindering campaign creators from finding the correct way to enhance their campaigns.

1.4 Proposed Extension

To build on existing work and address the pain points mentioned above, this project will further explore the use of machine learning to predict the success rate of crowdfunding campaigns, with more stress on the use of broader categories of data and multimodal neural networks. Also, in addition to only outputting a probability, this project will explore the explanation side of the results applying DeepSHAP, addressing the dearth of research on related topics. In addition to the prediction and explanation, another key emphasis of this project is to develop an advanced filtering tool upon the original one provided by Kickstarter, offering a more detailed and customized view of project data and allowing users to better analyze trends and success factors across different campaigns.

1.5 Objectives, Scope and Deliverables

This project aims to develop **CrowdInsight**, an end-to-end platform serving as a one-stop solution with two key parts: the first being an **Insights** section, which allows users to leverage advanced filtering options on all Kickstarter campaign data. This can help them better understand the dynamics of the market and extract actionable insights on trends and best practices. The second part is the **Analysis** section, which features a *multimodal neural network* with explainable artificial intelligence (XAI) to predict the success rate of crowdfunding campaigns and provide data-driven recommendations to improve the likelihood of reaching the funding goal.

1.6 Report Outline

The report is organized into four chapters. The first chapter offers an overview of the background, areas of improvement in the existing work, and sets the project objectives.

The second chapter examines the project's methodology. It covers the data scraping and pre-processing approach for Kickstarter, the design of the multimodal neural network, the employment of XAI with DeepSHAP, and finally, the website design.

The third chapter presents the current progress of the project. It will also show the project schedule, the future steps we are about to take, and the challenges we face at present.

The final chapter concludes the report. It sums up the major progress made until now.

Chapter 2

Methodology

2.1 Project Overview

This project adopts a structured methodology to develop a web-based system powered by a multi-modal neural network for predicting Kickstarter campaign success. Key stages include **data collection**, **pre-processing**, **model development**, and integrating **explainability** mechanisms to enhance user understanding of predictions. The final step involves connecting the model to a **user-friendly interface**. Each phase is elaborated in detail throughout this chapter.

2.2 Data Collection

The data for the items listed in Kickstarter is necessary to train and test our machine learning model. To be specific, to accomplish the goal of predicting the success rate of a Kickstarter project, ground truth status of the past projects, and a set of features are required. The data can be obtained by scraping content from the Kickstarter platform.

A point to note is that Kickstarter currently does not provide official documentation for accessing the data of the items listed on its website, or the database storing the relevant information. Therefore, one challenge is choosing the most suitable scraping method from the multiple possible options, each with its own pros and cons. A comparison and justification of these methods are to be discussed in Section 3.7.1. In a nutshell, the Kickstarter dataset from Web Robots, a Lithuanian web crawler service, is primarily utilized, with additional data obtained via Selenium web scraping.

2.3 Feature Engineering

Before training the model to predict Kickstarter project status, pre-processing and cleansing of the data are essential. The following three types of data will be utilized: textual, categorical, and numerical. To be prepared to pass through a dense layer in the model for capturing patterns (see Section 2.4), the different types of data will be pre-processed as follows:

2.3.1 Textual Data

Textual data, such as project names and long descriptions, are processed using a text-embedding model from Hugging Face. Text embeddings are generated for media content descriptions related to the projects, which include blurb and project descriptions.

2.3.2 Categorical Data

Categorical features, like project categories and statuses, were originally planned to be converted using one-hot encoding or embedded variables. For example, a true-or-false variable can be converted to a one and zero variable. This quantification of the textual and categorical data makes it easier for the model to process them.

However, as we proceeded, we found that one-hot encoding is not feasible for certain cases due to the high dimensionality introduced. For instance, with nearly 200 countries and over 100 subcategories, one-hot encoding would result in an excessively large number of dimensions, leading to increased computational overhead and sparsity issues. Therefore, we have decided to adopt embedding layers for subcategories and countries, which can efficiently represent these features in a dense and compact form. More details on this approach will be discussed in Subsection 3.3.3.

2.3.3 Numerical Data

Numerical features, such as pledged amounts, backer counts, and the number of projects initiated by a creator, are essential for assessing a project's viability and popularity. They usually require no complicated pre-processing, except for the monetary values, which have to be unified to US Dollars based on the corresponding currency data.

2.4 Model Architecture

To predict the success rate of Kickstarter projects, a **multimodal neural network** is employed to process both structured data, comprising numerical and categorical inputs, and unstructured data, specifically text-based information. As illustrated in Figure 2.1, the core design of the model consists of three distinct unimodal models that handle different types of input data. After pre-processing the input data through the numerical data processing, text, and categorical embedding layers described in section 2.3, the outputs are concatenated in the Fusion Module. This fusion layer combines the representations into a unified vector and passes them through dense layers, where neurons learn joint representations from the different data sources.

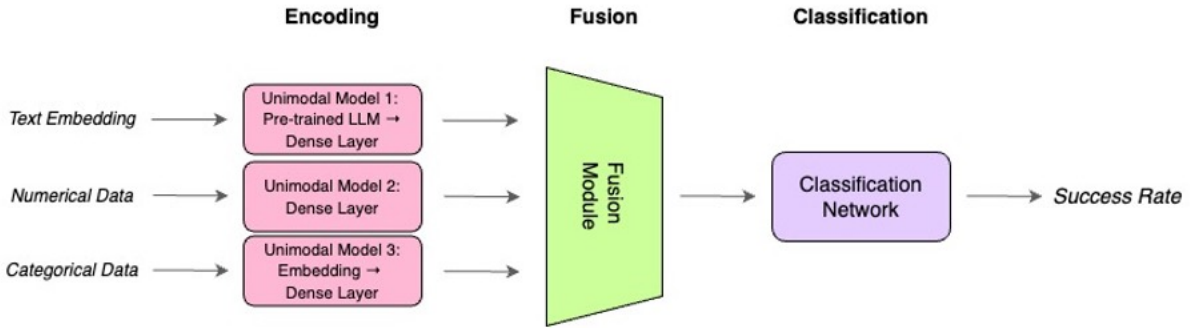
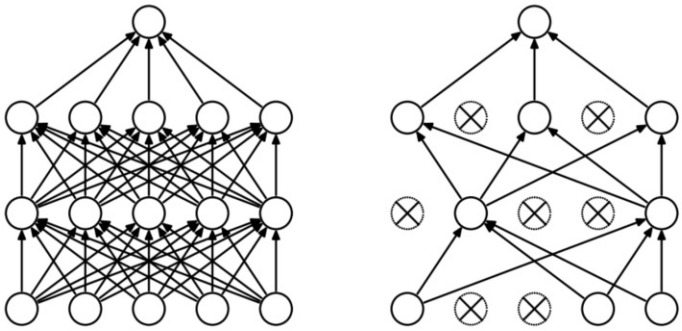


Figure 2.1: Multimodal Neural Network Architecture

The concatenated feature vector is then fed into a classification network tasked with predicting the success rate of the campaign. As shown in Figure 2.2a, this network comprises several fully connected layers, which are responsible for refining combined input features and learning complex, non-linear relationships between the multiple data types. To limit generalization error, as illustrated in Figure 2.2b, the dropout technique is broadly implemented to alleviate overfitting, guaranteeing that neural units do not co-adapt too excessively or precisely to a specific set of training dataset (Srivastava et al., 2014; Liu et al., 2023; Salehin & Kang, 2023). By deactivating random neurons during training, the dropout layers effectively avoid the model from over-dependence on specific neurons, hence strengthening its generalization capabilities.



(a) Standard Neural Net (b) After applying dropout.

Figure 2.2: Dropout Neural Net Model. **a**: A standard neural network with fully connected layers processing multimodal data. **b**: An example of the same network after applying dropout, where crossed-out units represent neurons that have been randomly dropped during training. (Srivastava et al., 2014, Figure 1)

Ultimately, the classification network concludes with an activation function, which returns a probability score between zero and one, signifying the predicted likelihood of campaign success. In essence, the multimodal neural network architecture promises flexibility in processing different input features while also improving robustness through the application of dropout to maintain generalization.

2.5 Explainability and Interpretation

To enhance the explainability of our model and provide actionable insights for creators' improvement, we have exploited the **Deep SHapley Additive exPlanations (DeepSHAP)** methodologies approach. Although deep learning models often outperform simpler models, they are often referred to as "black boxes" due to their inherent complexity and lack of transparency. Consequently, XAI has emerged as a significant focus in parallel with the advancement of these "black boxes" models. Shapley values, while effective in addressing the opacity, are computationally expensive, particularly for deep learning models (Norbye, 2023). **DeepSHAP**, an extension of DeepLIFT was developed to mitigate the computational burden inherent in Shapley value calculations. The method is formalized as follows:

$$\phi_i(v) = \frac{1}{|R|} \sum_{r \in R} \left[v(P_i^r \cup \{i\}) - v(P_i^r) \right] \tag{2.1}$$

Where:

ϕ_i: Shapley value approximation for feature i
R: A set of representative reference values
P_i^r: Set of features with ordering r
$v(P_i^r)$: Contribution of the set P_i^r without feature i
$v(P_i^r \cup \{i\})$: Contribution of the set P_i^r with feature i.

The key difference in the original Shapley value formula is the use of representative reference values (R) instead of all feature subsets (N). The original Shapley approach averages contributions across all subsets, making it computationally expensive. **DeepSHAP** reduces this by averaging over a limited set of reference values (R) at each layer (Koenen, 2024), acting as proxies for feature subsets and minimizing overhead.

Since its introduction, DeepSHAP has been applied in various domains, including medicine (Keleko et al., 2023; Bhattarai et al., 2024), agriculture (Khamankar et al., 2024), and weather forecasting (Konstantinou & Hatziargyriou, 2024). In our case, DeepSHAP is used to explain the importance and contribution of each feature to the model's success prediction, allowing us to visually present the features' respective roles in the decision-making process. To enhance user understanding, we will integrate a pre-trained LLM to generate a natural language explanation for these values, and display the results in an interactive table, providing users with interpretable insights into how various aspects of their project affect its predicted success. The incorporation of the LLM will be further discussed in Section 3.6.6.

2.6 Website Design and User Interaction

2.6.1 Website Overview

The website is primarily designed for entrepreneurs, enabling them to gain insights into trending industries, determine optimal funding goals for campaigns, and identify the key information necessary to attract backers on platforms like Kickstarter. It features two main sections to support this goal. The Insights

section offers customizable filters to explore projects, while the Analysis section allows users to input project Uniform Resource Locators (URL) or details, generating success rate predictions and offering feature explanations to provide a clearer understanding of the factors influencing project outcomes.

2.6.2 Insights Section: Advanced Project Filtering

This section enhances the navigation capabilities of Kickstarter by offering more advanced filtering and customization options. Users can narrow results by specific time periods and project features. This functionality is made possible through simple database retrieval process, which allows for a more detailed and personalized view of project data. As a result, users can analyze successful campaigns and derive valuable insights from failed ones, ultimately helping them make more informed decisions for their own projects.

2.6.3 Analysis Section: Project Success Prediction and Explanation

This section allows users to assess the potential success of a Kickstarter project by either providing a project URL or manually entering all key details. If a URL is provided, the data required would be retrieved through website scraping. The processed data will then be fed into the multimodal neural network in Section 2.4, calculating the project's likelihood of success, and presenting the user with a percentage score. Additionally, as described in section 2.5, a natural language explanation and an interactive table are used to interpret the DeepSHAP values, helping users grasp why certain aspects of their project might positively or negatively influence the likelihood of success.

2.7 Summary

This chapter proposed the rationale behind our project. DeepSHAP was described, along with detailed explanations of the multimodal neural network used for predicting project success. The chapter also covered essential aspects like data collection, feature engineering, and the integration of the model with the website's front-end for entrepreneurs, offering a seamless flow from data input to actionable insights. Additionally, the website's filtering and analysis capabilities were discussed. The next section will detail current progress and next steps.

Chapter 3

Results and Discussions

3.1 Overview

This chapter outlines the current progress of the project, including data retrieval, scraping, and storage processes (Section 3.2), feature engineering techniques for categorical, numerical, and textual data (Section 3.3), and website implementation details (Section 3.4). Additionally, the project schedule and immediate future plans are presented in Sections 3.5 and 3.6, while challenges and limitations encountered during development are discussed in Section 3.7. The overall progress aims to ensure that the project remains on track to deliver actionable insights and user-centric functionalities.

3.2 Data Collection

This section details the processes for retrieving, scraping, and storing data to support model development and analysis. It covers the primary dataset, additional dynamic content scraping, and the implementation of a cloud-based storage solution to ensure scalability and accessibility for subsequent steps.

3.2.1 Dataset

The primary data source for this project is the **Web Robots platform**, which has developed a Kickstarter dataset with current and historical data (Web Robots, 2025). The data is presented in JavaScript Object Notation (JSON) format and updated monthly. As of the dataset created on January 13, 2025, there are 214,186 unique projects, with their dates ranging from 2009 to 2025.

The datasets from Web Robots store the data with a wide range of over 100 variables, structured similarly to the Kickstarter Application Programming Interface (API). These variables include, but are not limited to, name, category, goal amount, project status, country, and creator information, all of which can be used in the feature set for model development. Notably, the project status indicates the project's state at the time of data scraping, with possible statuses including successful, failed, submitted, cancelled, live, started, or suspended. This status data can serve as the ground truth for the model. Figure 3.1 illustrates the distribution of the item statuses as of the dataset created on January 13, 2025.

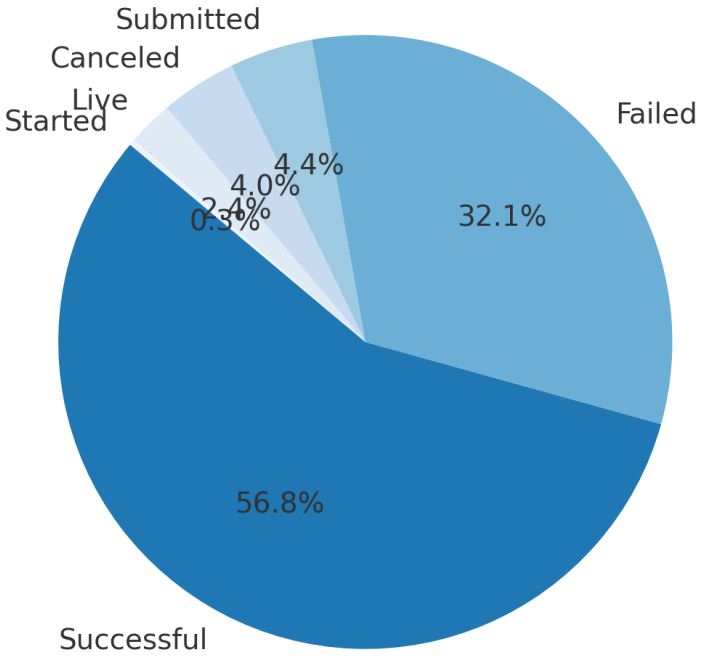


Figure 3.1: Status of the Kickstarter items in the Web Robots dataset

While the third-party dataset from Web Robots offers significant advantages due to its similarity in structure and variables with Kickstarter's official but undocumented API (Soderberg et al., 2019), there are limitations in balancing data scraping efficiency and data source reliability. The limitations and the proposed solution are discussed in Section 3.7.1.

3.2.2 Dynamic Content Scraping

For the multimodal model training (see Section 3.6), in addition to the basic information in the dataset, textual content about the project is needed, such as text descriptions, the number of links, and visual media in the project description. These data exist as JavaScript-rendered dynamic elements on each project website and are not recorded in the database. Therefore, an additional tool is required to scrape this dynamic text content.

Selenium, a web scraping library based on Python, is chosen as the scraping tool due to its ability to extract dynamic content from webpages on a large scale. It can also disguise its mass data retrieval to avoid Internet Protocol (IP) address blocking by mimicking normal human browser behaviour. However, extensive scraping that involves rendering complete browser behaviours could result in time- and computationally-intensive processes, which may significantly limit project progress. The details and solutions are discussed in Section 3.7.2.

3.2.3 Data Storage

The final deliverable of this project is a web application accessible via the Internet. Therefore, the application architecture should prioritize scalability, reliability, and accessibility. To ensure round-the-clock availability, a cloud database can be integrated to maximize uptime. In this project, MongoDB

Atlas has been chosen as the data management tool.

MongoDB Atlas, the cloud-based version of MongoDB, supports the native JSON data format, which is also used by the Web Robots dataset. Moreover, it offers built-in security features, automatic backups, and convenient connection through API, enabling efficient data management and queries. After pre-processing and feature engineering, the dataset will be migrated to MongoDB Atlas for instant access and cloud-based operations.

3.3 Feature Engineering

As described in Section 3.2, the team have successfully retrieved raw data from the Web Robots dataset, which provides comprehensive information on Kickstarter projects. The data retrieved will primarily be utilized in two aspects of the project: data searching with filter and sort functionality in the web application, and as the dataset for training the multimodal model. For the former, the dataset requires minimal modifications. However, for the latter, substantial processing is necessary to ensure data quality and that the trained model can accurately capture the data patterns. This subsection outlines the steps we have taken to pre-process raw data for the AI training dataset.

3.3.1 Project State Engineering

The model's ground truth and predictions are categorized as either successful or failed, based exclusively on completed project states. Projects with ongoing statuses, such as "submitted", "live", or "started", are excluded from the training set, as their outcomes remain indeterminate and cannot contribute meaningful labels for classification. Similarly, projects with a "suspended" status are removed, as they are terminated due to rule violations rather than campaign performance, making them irrelevant for predictive analysis.

For cancelled projects, the reasons for termination can vary. Some cancelled campaigns exceed 100% of their funding goal, suggesting factors unrelated to backer support. To address this variability, cancelled projects are evaluated based on their campaign progress. Projects that did not fulfil their funding goal and were terminated after 40% of their duration are considered failed. However, cancelled projects terminated within the first 40% of their duration are excluded from the dataset, as they are likely abandoned for reasons other than campaign settings. Of the 8,937 cancelled projects, 3,199 were excluded for being terminated early, while 5,738 were converted to "failed" for analysis.

After applying these filters, the refined dataset includes 189,796 projects out of an initial 201,910 entries. The dataset consists of 112,707 successful projects and 71,351 failed ones, ensuring that only meaningful and relevant data is retained for training the predictive model.

3.3.2 Numerical Feature Engineering

The initial idea was to simply normalize the numerical values to a range between 0 and 1. However, further investigation unveiled the presence of certain outlier campaigns, where the funding goal is set to an unrealistically high number (e.g., 1 billion), making it practically impossible to reach. Such instances, often created for amusement, can significantly skew the data.

If normalization is applied in these cases, it would disproportionately compress the scale of normal data values, which could lead to potential inaccuracies. To tackle this issue, we considered two approaches. The first is to screen out these anomalous data values. The second is to apply techniques such as log transformation, which can reduce the skewness of measurement variables (West, 2021) and compress the influence of outliers, preserving the overall data distribution while enhancing prediction performance.

3.3.3 Categorical Feature Encoding

The dataset contains multiple categorical variables, which cannot be directly parsed for AI training due to their textual nature. Common methods of categorical feature engineering include one-hot encoding and embedding layers. While one-hot encoding transforms categorical variables into multiple Boolean columns, indicating the presence of each category, it becomes impractical for features with many categories.

Embeddings offer a more efficient alternative by converting categorical variables into fixed-size vector spaces, reducing dimensionality and computational complexity. Additionally, embeddings can better capture the patterns and relationships between categories, allowing the model to learn meaningful representations during training.

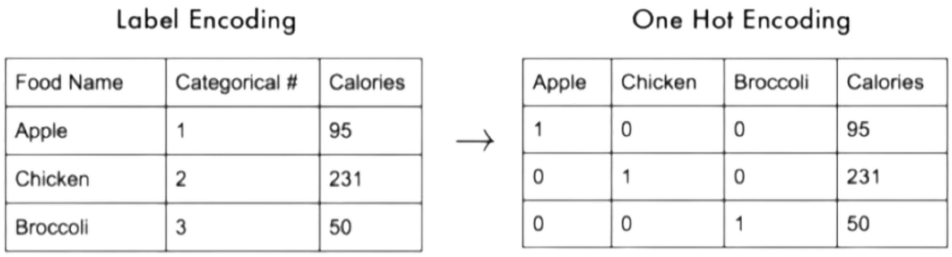


Figure 3.2: One-hot encoding demonstration. **Left**: label encoding with categorical numbers. **Right**: one-hot encoding with binary vectors for each category.

Therefore, while one-hot encoding as illustrated in Figure 3.2 is used for categorical features with less than 20 distinct categories, for example, project category, embedding conversion is performed for countries and subcategories in the dataset. There are 159 subcategories and approximately 40 countries in the Kickstarter dataset, we utilized embedding layers to represent these categorical variables. Each subcategory was mapped to a fixed vector size of 32 dimensions, while the country variable was mapped to a vector size of 24 dimensions. These sizes were determined based on the heuristic:

$$\text{Embedding Size} = \lceil 6 \times \log_2(\text{Number of Categories}) \rceil \tag{3.1}$$

For example, with 159 subcategories, the vector size of 32 provides an efficient trade-off between representational capacity and computational cost. Similarly, the 40 unique country values were mapped to a 24-dimensional space. This allows the model to learn latent relationships between categories during training and significantly reduce input dimensionality compared to one-hot encoding. By using embeddings, the model can refine these representations to optimize prediction performance, capturing patterns and relationships across categories that are relevant to the success of Kickstarter campaigns.

3.3.4 Text Feature Engineering

One example of textual data is the blurb of the campaign, such as "Discover 15+ easy and unique recipes, each one showcasing the rich culinary traditions of Greece". When applying embeddings to such data, it is pivotal to balance efficiency and comprehensiveness.

During our exploration of embedding models on Hugging Face, we identified two potential candidates: all-MiniLM-L6-v2 and jina-embeddings-v3.

After our exploration on Hugging Face, we have chosen all-MiniLM-L6-v2, which has a size of only 22 MB, making it highly efficient and lightweight. Its small size allows extremely fast performance and minimal hardware requirements, which may even suffice for real-time running on our platform. The model creates embeddings with 384 dimensions and handles up to 256 word pieces which is well enough for processing the description sentences. Compared to some other embedding models with more complex architecture and significantly larger size (measured in gigabytes) that require higher-end GPUs, all-MiniLM-L6-v2 seem to best suit this project as we might not have sufficient computational resources.

One potential concern is that some Kickstarter campaigns are listed in languages other than English, such as Spanish and French. However, considering that some other good multilingual embedding models tend to have much higher sizes and require subscriptions, we might need to filter out the non-English campaigns. This approach, however, might risk introducing biases. As such, we plan to conduct more in-depth research to solve this issue.

3.3.5 Summary

In this section, we successfully retrieved and processed data from the Web Robots dataset and those scraped with Selenium. The data underwent comprehensive feature engineering, ensuring that the data is fully prepared and optimized for model training. The next section will elaborate on the infrastructure design, detailing how the components are structured to support the end-to-end system, and the implementation of the insights and analysis module.

3.4 Website Implementation

The web application integrates frontend design, data management, and predictive model functionalities into a unified platform hosted on the Internet. To ensure continuous availability and ease of deployment, Streamlit, a Python-based, open-source, and cloud-compatible platform, has been selected as the hosting solution (Treuille, 2019). Streamlit facilitates the integration of machine learning features into web applications, making it a suitable choice for this project. The application will be directly linked to the official project website, hosted by the University of Hong Kong on WordPress, to ensure seamless navigation between the platform's components. This section will outline the web application's design and functionalities. This includes a detailed description of the website's user interface (UI) design, the insights module offering advanced filtering and dynamic data analysis, and the analysis module that enables project success predictions along with actionable explanations.

As shown in Appendix A, the UI designs provide a visual representation of the website's layout. This

section will cover the design and functionalities of the website.

To start with, the **Home Page** (Figure A.1) serves as the default landing page and provides a simple and intuitive navigation interface for users to access the platform's core functionalities. Key highlights include:

- **Module Previews:** Sections introduce primary features, such as insights on crowdfunding trends, project success prediction, and campaign improvement tools.
- **Visual Appeal:** A clean, image-centric design with clear headings ensures user engagement and seamless navigation.
- **Navigation Bar:** Offers quick access to other pages, such as the insights module, prediction tools, and user guides.

And the **Insights Section** (Figure A.2) is divided into two components. The **Data Insights Component** allows users to explore key crowdfunding metrics. Users can select either all categories or a specific category, and choose a timeframe of 30, 60, 90, 180 days, and even 1 or 2 years. Upon initiating a search, the following metrics are displayed:

- Number of campaigns, total funds raised, successful campaigns, and success rate for the selected timeframe and category, with relative changes compared to the previous timeframe.
- **Trending Categories:** Displays the growth rates of categories based on the number of new campaigns.
- **Funding Goal Distribution:** Visualizes the breakdown of campaigns by funding goals.
- **Top Project Locations:** Highlights the geographical regions with the most active campaigns.
- **Average Funding Per Backer by Category:** Illustrates the average amount pledged per backer for different categories.
- **Top 5 Funded Campaigns:** Lists the highest-funded projects in the given timeframe and category, with clickable links to the campaign pages.

While the **Advanced Filtering Component** (Figure A.3) allows users to refine their searches, it expands on Kickstarter's limitations by offering:

- **Advanced Subcategory Filtering:** Users can filter projects at the subcategory level, enabling detailed exploration of niche areas within broader categories like "Technology" or "Design."
- **Combined Filtering and Sorting:** Allows applying multiple filters of the same type while simultaneously sorting by metrics like popularity and success rate.
- **Dynamic Range Filtering:** Customizable sliders for received amount, target amount, and percentage funded replace Kickstarter's rigid predefined ranges, offering greater flexibility.

- **Expanded States:** Includes additional project states like "failed" and "suspended," providing a complete view of projects beyond successful campaigns.
- **Timeframe Selection:** Users can choose specific timeframes, such as 30, 60, or 90 days, unlike Kickstarter's fixed options, enabling dynamic analysis of recent or historical trends.

Finally, the **Analysis Section** (demo, Figure A.4) enables users to input a project URL or manually enter project details, as described in Section 2.6.3. While the user interface has been designed, the backend implementation is scheduled for completion in the coming months. Once submitted, the input data will be processed by the predictive model to calculate the success rate and generate DeepSHAP values. Additionally, a natural language explanation and tailored suggestions for improvement will be provided. Further details on these functionalities can be found in Section 3.6.6.

3.5 Project Schedule

Table 3.1: Project Timeline and Deliverables

Time Periods	Tasks
Sep., 2024	Literature Review: - Kickstarter campaigns and crowdfunding platforms - Exploration of available datasets and APIs
	Phase 1 Deliverables: ***- Detailed project plan*** ***- Project website***
Oct. - Nov., 2024	Data Collection: - Retrieve project data using Web Robots dataset - Additional data retrieved wih Kickstarter's unofficial API
	Data Preprocessing: - Initial feature engineering for features
Dec., 2024	Website Design and Integration: - Draft design of website infrastructure - Implement the Insights Section with interactive filters
	Phase 2 Deliverables: ***- Scraping tool and preprocessed dataset ready for analysis*** ***- Functional Insights Section integrated into the website***
Jan. - Mar., 2025	Model Implementation: - Design and implement neural network architecture - Conduct initial model training and parameter tuning - Evaluate model's performance on preprocessed dataset
	Explainability Integration: - Integrate DeepSHAP for XAI capabilities - Test explainability component with trained model
	Phase 3 Deliverables: ***- Functional model with integrated XAI features***
Apr., 2025	***Phase 4 Deliverables:*** ***- Fully launched platform*** ***- Exhibition video*** ***- Final presentation*** ***- Final report***

Table 3.1 provides an overview of the project schedule. We have completed data retrieval and preprocessing, preparing a clean dataset for analysis and feature engineering, as described in Section 3.3.

Additionally, the design of the website infrastructure and initial testing phases are achieved, ensuring steady alignment with the project timeline.

3.6 Preliminary Results and Future Directions

As we approach the next stages, our focus will center on completing the multimodal neural network development, refining its performance through advanced tuning, and integrating XAI techniques, including a pre-trained LLM with DeepSHAP to enhance model transparency and user understanding. Simultaneously, we will finalize the website's interactive features, ensuring seamless integration with the predictive model and an intuitive user experience. These efforts aim to deliver a robust platform that aligns with the project's goals and provides valuable, actionable insights to users.

3.6.1 Concerns over Imbalance of Features in Concatenated Vector

After performing feature engineering separately on textual, categorical, and numerical data, the resulting vectors are concatenated into a single vector before being fed into the neural network. In this process, we raise a concern regarding the potential imbalance in the contributions of features from different vectors. For instance, textual data could occupy hundreds of dimensions, whereas numerical data may only account for a few. This raises the question of whether the training process might disproportionately emphasize one single input modality, such as textual data, causing the network to become biased towards it.

As a result, ensuring generalization is crucial so that non-dominant input modalities can contribute more effectively during training. However, this concern may or may not hold true, and further research and experiments are needed to investigate this issue and determine whether additional measures are required to address it. For example, Das et al. (2023) introduced a novel training setup with a regularizer to mitigate issues caused by disparities between modalities.

3.6.2 Loss Function

During training, a loss function evaluates the model's performance by computing the deviation of the predictions from the "ground truth" (Bergmann & Stryker, 2024). In this project, we predict the success rate as a continuous value rather than solving a binary classification problem. Instead of binary cross-entropy loss, suited for classification tasks, we will employ mean squared error (MSE) as the loss function. MSE measures the average squared difference between predicted success rates and actual values, ensuring the model learns effectively by minimizing large prediction errors.

3.6.3 Optimizer

During backpropagation, the gradient of the loss function is calculated, and an optimization algorithm will utilize it to update the weights and biases of the neural network in order to minimize the loss function. It is therefore essential to select a suitable optimizer for this project.

Several optimization algorithms are available, with Stochastic Gradient Descent (SGD) being the basic one. However, we have decided not to use SGD in this project as it applies the same global learning rate to all parameters, regardless of their scale. In addition, it is not particularly effective at escaping saddle points.

Many other options are available, such as Mini-Batch Gradient Descent, RMSProp, Adam, etc. For this project, we are inclined to use Adam as the optimizer. **Adam** has been developed for large datasets and high-dimensional parameter spaces, and it combines the advantages of other optimization methods such as AdaGrad to cope with sparse gradients and RMSProp to deal with non-stationary objectives (Kingma & Lei Ba, 2015). While Adam is our initial choice, further exploration of alternative optimization methods will be conducted to ensure the most effective approach is utilized.

3.6.4 Validation

During the validation stage, across all epochs, our objective is to identify the model that delivers the best performance. Some performance metrics will be employed, including accuracy, precision, recall, F1-score and ROC AUC score, among others. A key question, however, is determining the optimal performance without going running through all epochs, as this would take up excessive time and computational resources.

To address this, we will apply a **plateauing technique**, such that training is halted early if there is a fall in validation accuracy or if performance improvements stop. Also, we will perform a learning curve analysis by plotting the training and validation losses. A converging curve with low validation loss might indicate that the model is already well-trained, else a diverging curve might suggest overfitting or the need for better tuned hyperparameters.

3.6.5 Hyperparamter Tuning

Our objective is to locate the "global best" combinations of hyperparameters, rather than settling for the "local best", so hyperparameter tuning will be carried out. Hyperparameter Tuning refers to the practice of identifying and selecting the optimal hyperparameters for use in training a machine learning model (Belcic & Stryker, 2024). When executed correctly, it should help minimize the loss function of the model. Key hyperparameters to be considered include the number of hidden layers, the number of neurons per hidden layer, learning rate, epochs, batch size, regularization strength, and choice of optimizer, among others.

To perform the hyperparameter optimization, a variety of methods are available. The most popular approaches include grid search, random search, and Bayesian optimization. We are unlikely to employ grid search in this project because it is exhaustive and involves training the model for all possible configurations of those discrete hyperparameter values. This approach would be very time-consuming, inefficient, and computationally expensive, especially given the limitations of our GPU resources. As such, given the large distributions in the search space, we would instead start working with **random search**, which samples a fixed number of parameter combinations at random from a described distribution. This approach allows us to save a lot of time to identify an effective combination, although it carries the risk of finding a "close-enough" solution rather than the optimal one.

3.6.6 Explainability Implementation Approach

To enhance the interpretability and actionable feedback provided by our system, we plan to integrate **Retrieval-Augmented Generation (RAG)** with a pre-trained LLM, specifically FLAN-T5. The proposed pipeline consists of the three components below:

1. **Data pre-processing**
   The predicted success rate generated by the multimodal neural network will be pre-processed alongside user-provided project data. The system will compare the user input against successful project practices, generating:

   - A list of items that align with successful project practices.
   - A list of items that deviate from successful project practices, highlighting potential areas for improvement.

2. **Retrieval with FAISS**
   A *vector database*, indexed using FAISS (Facebook AI Similarity Search), will store descriptions of top-performing Kickstarter campaigns. The system retrieves the 5 most relevant and successful project descriptions based on user-provided inputs. This ensures that recommendations are informed by best practices observed in successful projects.

3. **LLM Generation**
   The FLAN-T5 model is utilized for generating specific parts of the feedback:

   - A generic explanation interpreting DeepSHAP results, providing insights into the contribution of features to the predicted success rate.
   - Contextual explanations for deviations in project inputs from typical ranges observed in successful campaigns.
   - Improvement suggestions, generated using a five-shot prompting technique, informed by retrieved descriptions of the top 5 most relevant and successful projects. This ensures actionable recommendations tailored to refining campaign descriptions and parameters.

   FLAN-T5 is chosen for its instruction-tuned optimization and ability to generate detailed, context-aware responses, making it suitable for providing insights and guidance for campaign optimization.

The final feedback is concatenated by four main parts:

- **Announcement of Results:** Reports the predicted success rate and provides a hardcoded greeting message for high success rates or a failure message for lower rates. Additionally, the list of items done well is listed out.
- **DeepSHAP Explanation:** Provides an interpretable breakdown of how input features contributed to the predicted success rate. This includes a table displaying the DeepSHAP values, a fixed explanation of how DeepSHAP works, and a generic description interpreting the significance of the results.

- **Interpretation:** Identifies inputs that deviate from the ranges typically observed in successful projects. For example, differences in project goals, duration, or the number of links in the project description are highlighted. This is achieved by directly utilizing the list of deviations generated during data pre-processing and presenting explanations for these deviations to the user.
- **Improvement Suggestions:** Based on a five-shot prompting technique, the system generates recommendations informed by the retrieved descriptions of the top 5 most relevant and successful projects. This approach ensures actionable and data-driven feedback for refining campaign descriptions and other parameters.

This structured pipeline ensures that creators benefit from a data-driven and user-centric explainability framework, designed to optimize campaign outcomes effectively.

3.6.7 Summary

This section outlines progress on refining the multimodal neural network, integrating XAI techniques and addressing challenges such as feature imbalance and computational constraints. A RAG pipeline is proposed to provide actionable feedback for campaign creators. All these advancements will be incorporated into the website's analysis module, as described in Section 3.4 and illustrated in Figure A.4.

3.7 Limitations and Difficulties

3.7.1 Balancing Data Scraping Efficiency and Data Reliability

The Kickstarter platform does not offer official guidance on retrieving project data. However, after thorough research and investigation, the following data scraping options have been identified:

1. **Requests**
2. **Selenium**
3. **API Scraping**
4. **Kickstarter Data from the Inter-University Consortium for Political and Social Research (ICPSR)**
5. **Kickstarter Dataset from Web Robots**

The first three methods are categorized as website content scraping, which involves using scripts to automate the process of extracting content from the website, while the remaining two approaches gather data from external databases. When choosing the most suitable method, it is crucial to maintain a balance between scraping efficiency and data reliability. The following details the key options:

1. Requests and Selenium

Requests and Selenium are both Python libraries that enable the mass scraping of websites. However, there are critical distinctions between the two approaches. Requests simply retrieves the static website content, while Selenium mimics normal browser behaviors, allowing it to extract not only static content but also dynamic and rendered content in JavaScript. However, in terms of execution efficiency, Requests is more competitive since it does not rely on browser operations.

2. API Scraping

Meanwhile, API scraping is an alternative approach that typically utilizes APIs established by platforms for internal development, enabling instance calling and information retrieval. Although Kickstarter does not have a documented or official API, the Kickstats project developed by Soderberg et al. (2019) identified an undocumented API used by Kickstarter, which can be accessed on the project website. Below are sample calls and the data retrieved, presented in JSON format.

Sample API Data
: The Kickstarter platform features an undocumented API that can be accessed to retrieve JSON-formatted data by constructing specific URL calls. For instance, the following API call retrieves data for the search term *Firedance Jewelry Small Dichroic Earrings*:

```
https://www.kickstarter.com/discover/advanced?google_chrome_workaround&sort=magic&seed=2580500&page=1&format=json&term=Firedance%20Jewelry%20Small%20Dichroic%20Earrings
```

When the keywords are encoded in American Standard Code for Information Interchange (ASCII) text strings in the %term= parameter, the JSON response provides detailed project data as shown below:

```
{
  "projects": [
    {
      "id": 645289718,
      "photo": {
        "key": "assets/011/959/314/781beee179bfd90bf438f8b38cc0f3cb_original.JPG",
        ...
      },
      "name": "Firedance Jewelry Small Dichroic Earrings",
      "blurb": "Firedance Jewelry pieces are uniquely handcrafted using fused
          dichroic glass. Firedance Jewelry - the fusion of art & science.",
      "goal": 500.0,
      "pledged": 411.0,
      "state": "failed",
      "slug": "firedance-jewelry-small-dichroic-earrings",
      "disable_communication": false,
      "country": "US",
      "country_displayable_name": "the United States",
      "currency": "USD",
```

```
      "currency_symbol": "$",
      "currency_trailing_code": true,
      "deadline": 1418943600,
      ...
    }
  ],
  ...
}
```

The data retrieved contains over one hundred variables, encompassing media assets, financial details, location, creator attributes, and category classifications. While this API provides comprehensive information, it is not officially documented by Kickstarter. This lack of documentation indicates that the API is likely intended for internal use and could be restricted at any time. Furthermore, scraping a large number of projects using this method is time-intensive due to its reliance on individual API calls for data retrieval.

3. ICPSR Database

The remaining two approaches gather data from external databases. The first of them is the Kickstarter data created by Leland (2023), available in the ICPSR data archive. ICPSR is an academic database for various statistics, and the data within it is considered more reliable. However, there are several drawbacks to use it. First, it is not updated on a regular schedule. The most recent version was published in April 2024, while the previous version was released in September 2022. The unspecified update schedule makes it challenging to modify the training and testing dataset for future model updates. Secondly, the data only consists of basic information about Kickstarter items, which does not match the comprehensive JSON data from API. For reference, the dataset has 20 features, while the JSON data contains over 100 variables. Finally, some of its features, for instance, the project names, are restricted to the public users. It requires the research approval from the Institutional Review Board in the jurisdiction, in this case, the United States, and also the degree requirements, potentially including a doctoral degree. Given the prerequisites and the lack of complete information, this approach is considered less feasible.

4. Kickstarter Data from Web Robots

The final approach is to use the dataset provided by Web Robots (2025). This dataset features the same data structure and variables as the unlisted API, offering better data quality compared to the dataset from ICPSR. It could also save time spent on scraping. However, its reliability and data integrity are not assured since it is from a third-party source.

The solution adopted is utilizing the dataset from Web Robots as the primary data source, with additional data, for example, textual project descriptions, scraped using Selenium. In terms of data completeness and accuracy, the official but undocumented API is given the highest priority. However, since it takes a significant amount of time to scrape every project's data using the API, the immediately accessible external dataset with a similar data structure, Web Robots, is used instead. Given the reliability concerns associated with a third-party non-official data source, measures are taken to validate the dataset.

To ensure the reliability of Web Robots, random entries in the dataset are cross-checked with the API data. The number of random entries chosen for cross-checking is determined by the statistical formula designed by Krejcie and Morgan (1970):

$$s = \frac{X^2 NP(1-P)}{d^2(N-1) + X^2 P(1-P)} \tag{3.2}$$

Where:

s: required sample size.

X^2: chi-square for one degree of freedom at the desired confidence level.

N: population size.

P: population proportion, defaulted to be 0.5 for maximum sample size.

d: desired Margin of Error (MOE).

By applying the formula, the total number of data entries is regarded as the population size, and the sample size is the required minimum number of entries to cross-check, given the confidence level and MOE. Using a common 95% confidence level and a MOE of 0.05, the corresponding chi-square distribution score of 1.96 is applied. After substituting the population size of 214,186, i.e., the total number of unique projects in the dataset, the rounded-up sample size is 384. Hence, 384 random entries are cross-checked with the API.

After determining the required sample size, a Selenium script was created to automatically compare key and static variables in both the dataset and the API to check for matches. As a result, all 384 random entries matched perfectly, confirming the data reliability of Web Robots.

3.7.2 Technical Constraint from Insufficient Computational Power

The dataset comprises over two hundred thousand data entries, causing inefficiencies in scraping additional textual data for each project using Selenium, as well as in processing and training the AI model without substantial computational power. To address these technical constraints and facilitate the normal development process, a compromise is necessary.

To scrape the textual project descriptions using Selenium, the average time exceeds twenty seconds because Selenium mimics normal human browser behaviour to extract dynamic content. Given the dataset's volume, it is neither feasible nor efficient to scrape additional data for every project. Therefore, only the most recent ten thousand completed projects, filtered by successful and failed states, will be included in the training set. This ensures a more efficient AI training process and simultaneously captures the most recent data patterns.

3.7.3 Selenium Scraping Risk

Another risk of using Selenium as a scraping tool is that, since Selenium mimics browser behaviour, requests are continuously sent to the Kickstarter platform for data retrieval. Extensive scraping may potentially lead to an IP ban from the platform, stifling the entire project. To better disguise the mass data retrieval as normal browsing, several measures have been implemented.

Firstly, in addition to the native Selenium library, extra Python libraries are imported to enhance the stealthiness of data retrieval. These include undetected_chromedriver (2024), which reduces the likelihood of triggering anti-bot services, and SeleniumBase (2025), which allows the use of multiple browsers in the same test. Moreover, randomized behaviours are incorporated to add further ambiguity, such as varying browser configurations, user agents, and delays between requests.

Chapter 4

Conclusion and future plans

Due to the low success rates of crowdfunding campaigns, CrowdInsight has been developed as a tool to predict the likelihood of success for campaign creators. This project aims to build an AI-driven platform that integrates a multimodal neural network and explainable AI techniques, specifically DeepSHAP, few-shot prompting and pre-trained LLM to provide accurate success rate predictions for Kickstarter campaigns, accompanied by detailed explanations. If successfully implemented, this project will offer valuable insights to campaign creators, enabling them to improve their campaigns and, consequently, increase their chances of securing funding.

The data for this project has undergone extensive feature engineering, including logarithmic transformation, one-hot encoding, and embedding conversion. Additionally, the website's cloud infrastructure has been designed to support the platform. However, the primary roadblock currently hindering progress is the lack of sufficient computational resources. Training AI models requires considerable computing power and is inherently time-intensive. Future research will focus on optimizing the dataset size to better suit the model's requirements, thereby making the computational tasks more efficient and manageable.

Looking ahead, there is a vision to scale this platform beyond Kickstarter to include other crowdfunding platforms such as Indiegogo, Fundable, and more. This expansion would enable the tool to support a broader range of creators, fostering greater innovation and helping bring more creative projects to life.

References

Alegre, I., & Moleskis, M. (2016, October 7). *Crowdfunding: A Review and Research Agenda*. Papers.ssrn.com. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2900921

Belcic, I., & Stryker, C. (2024, July 23). *What Is Hyperparameter Tuning? | IBM*. Ibm.com. https://www.ibm.com/think/topics/hyperparameter-tuning

Bergmann, D., & Stryker, C. (2024, July 12). *What is Loss Function? | IBM*. Www.ibm.com. https://www.ibm.com/think/topics/loss-function

Bhattarai, P., Thakuri, D. S., Nie, Y., & Chand, G. B. (2024). Explainable AI-based Deep-SHAP for mapping the multivariate relationships between regional neuroimaging biomarkers and cognition. *European Journal of Radiology, 174*, 111403. https://doi.org/10.1016/j.ejrad.2024.111403

Camilleri, M. A., & Bresciani, S. (2022). Crowdfunding small businesses and startups: A systematic review, an appraisal of theoretical insights and future research directions. *European Journal of Innovation Management*. https://doi.org/10.1108/EJIM-02-2022-0060

Cavalcanti, G. D. C., Mendes-Da-Silva, W., dos Santos Felipe, I. J., & Santos, L. A. (2024). Recent advances in applications of machine learning in reward crowdfunding success forecasting. *Neural Computing and Applications*, 36(26), 16485-16501. https://doi.org/10.1007/s00521-024-09886-6

Cumming, D. J., & Johan, S. A. (Eds.). (2020). *Crowdfunding: Fundamental cases, facts, and insights*. Academic Press. https://www.sciencedirect.com/book/9780128146378/crowdfunding

Das, A., Das, S., Sistu, G., Horgan, J., Bhattacharya, U., Jones, E., Glavin, M., & Eising, C. (2023). *Revisiting Modality Imbalance In Multimodal Pedestrian Detection*. ArXiv.org. https://arxiv.org/abs/2302.12589

Etter, V., Grossglauser, M., & Thiran, P. (2013). Launch hard or go home! predicting the success of kickstarter campaigns. *Proceedings of the First ACM Conference on Online Social Networks*, https://doi.org/10.1145/2512938.2512957

Feng, Y., Luo, Y., Peng, N., & Niu, B. (2024). Crowdfunding performance prediction using feature-selection-based machine learning models. *Expert Systems*. https://doi.org/10.1111/exsy.13646

Fundable. (2014). Types of Crowdfunding. https://www.fundable.com/crowdfunding101/types-of-crowdfunding

Halim, M. A. (2024). Does crowdfunding contribute to digital financial inclusion? *Research in Globalization*, 9, 100238. https://doi.org/https://doi.org/10.1016/j.resglo.2024.100238

Huang, X., & Marques-Silva, J. (2023). The inadequacy of Shapley values for explainability. *arXiv preprint arXiv:2302.08160*.

Jumpstart3r. (2025). *Jumpstart3r-old.streamlit.app*. Retrieved from https://jumpstart3r-old.streamlit.app

Kaminski, J. C., & Hopp, C. (2019). Predicting outcomes in crowdfunding campaigns with textual, visual, and linguistic signals. *Small Business Economics, 55*(3), 627–649. https://doi.org/10.1007/s11187-019-00218-w

Keleko, A. T., Kamsu-Foguem, B., Ngouna, R. H., & Tongne, A. (2023). Health condition monitoring of a complex hydraulic system using Deep Neural Network and DeepSHAP explainable XAI. *Advances in Engineering Software, 175*, 103339. https://doi.org/10.1016/j.advengsoft.2023.103339

Khamankar, S., Sahu, S., & Tripathi, A. (2024). Design of an efficient DeepSHAP model for smart farming-based recommendations using residual deep networks. In *Recent Advances in Science, Engineering & Technology* (pp. 338-345). CRC Press.

Kickstarter. (2024a). *What happens when a project is suspended?*. Retrieved from https://help.kickstarter.com/hc/en-us/articles/115005136354-What-happens-when-a-project-is-suspended

Kickstarter. (2024b). *What is the maximum project duration?*. Retrieved from https://help.kickstarter.com/hc/en-us/articles/115005128434-What-is-the-maximum-project-duration

Kickstarter. (2025). *Stats*. Retrieved January 23, 2025, from https://www.kickstarter.com/help/stats

Kingma, D., & Lei Ba, J. (2015). *ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION*. https://arxiv.org/pdf/1412.6980

Koenen, N. (2024). Deep Shapley additive explanations. *Leibniz Institute for Prevention Research and Epidemiology*. https://bips-hb.github.io/innsight/reference/DeepSHAP.html

Konstantinou, T., & Hatziargyriou, N. (2024). Complex terrains and wind power: Enhancing forecasting accuracy through CNNs and DeepSHAP analysis. *Frontiers in Energy Research, 11*, 1328899. https://doi.org/10.3389/fenrg.2024.1328899

Krejcie, R. V., Morgan, D. W. (1970). Determining sample size for research activities. *Educational and Psychological Measurement*, 30(3), 607–610. https://doi.org/10.1177/001316447003000308

Leland, J. (2024). *Kickstarter data, global, 2009-2023*. Inter-university Consortium for Political and Social Research [Distributor]. https://doi.org/10.3886/ICPSR38050.v3

Liu, Z., Xu, Z., Jin, J., Shen, Z., & Darrell, T. (2023, July). Dropout reduces underfitting. In *International Conference on Machine Learning* (pp. 22233-22248). PMLR.

Louhichi, M., Nesmaoui, R., Mbarek, M., & Lazaar, M. (2023). Shapley values for explaining the black box nature of machine learning model clustering. *Procedia Computer Science, 220*, 806-811. https://doi.org/10.1016/j.procs.2023.09.103

McGregor, M. (2014). A new way to explore an incredible creative universe. *Kickstarter News Archive*. Retrieved from https://www.kickstarter.com/blog/a-new-way-to-explore-an-incredible-creative-universe

Norbye, T. K. (2023). Shapley values and explainability in AI models. Retrieved from https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3092847

Robertson, E., & Wooster, R. B. (2015). Crowdfunding as a social movement: The determinants of success in Kickstarter campaigns. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2631320

Salehin, I., & Kang, D. K. (2023). A review on dropout regularization approaches for deep neural networks within the scholarly domain. *Electronics, 12*(14), 3106. https://doi.org/10.3390/electronics12143106

seleniumbase. (2025). *SeleniumBase*. Retrieved from https://github.com/seleniumbase/SeleniumBase

Soderberg, B., Litt, A., Tortorello, J., & Sonnemaker, C. (2019). *Kickstats: What makes a successful Kickstarter project?*. Cornell College. Retrieved from https://kickstats.org/

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*, 1929-1958.

Testa, S., Nielsen, K. R., Bogers, M., & Cincotti, S. (2019). The role of crowdfunding in moving towards a sustainable society. *Technological Forecasting and Social Change*, 141, 66-73. https://doi.org/https://doi.org/10.1016/j.techfore.2018.12.011

Treuille, A. (2019). Turn Python scripts into beautiful ML tools. *Towards Data Science*, Medium. Retrieved from https://towardsdatascience.com/turn-python-scripts-into-beautiful-ml-tools

ultrafunkamsterdam. (2024). *undetected-chromedriver*. Retrieved from https://github.com/ultrafunkamsterdam/undetected-chromedriver

Vesterlund, L. (2003). The informational value of sequential fundraising. *Journal of Public Economics, 87*(3), 627–657. https://doi.org/10.1016/S0047-2727(01)00187-7

Wangchuk, P. (2021). Common types of Crowdfunding Models, Related Concepts and Its Impact on Business: A Brief Literature Review. Asian Journal of Economics, Business and Accounting, 56-63. https://doi.org/10.9734/ajeba/2021/v21i1430471

Web Robots. (2025). *Kickstarter datasets*. Retrieved January 26, 2025, from https://webrobots.io/kickstarter-datasets/

West, R. M. (2021). Best practice in statistics: Use the Welch t-test when testing the difference between two groups. *Annals of clinical biochemistry,* 58(4), 267-269.

Zhao, Q., Chen, C.-D., Wang, J.-L., & Chen, P.-C. (2017). Determinants of backers' funding intention in crowdfunding: Social exchange theory and regulatory focus. *Telematics and Informatics*, 34, 370-384. https://doi.org/10.1016/j.tele.2016.06.006

Appendix A

CrowdInsight UI Design

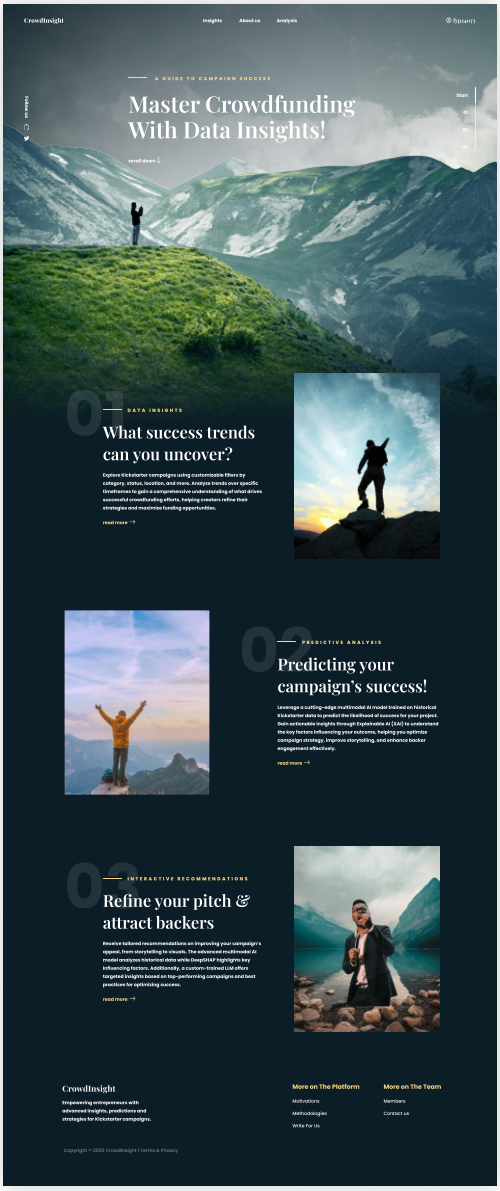


Figure A.1: CrowdInsight Home Page Design

Figure A.2: Insights Section: Statistics

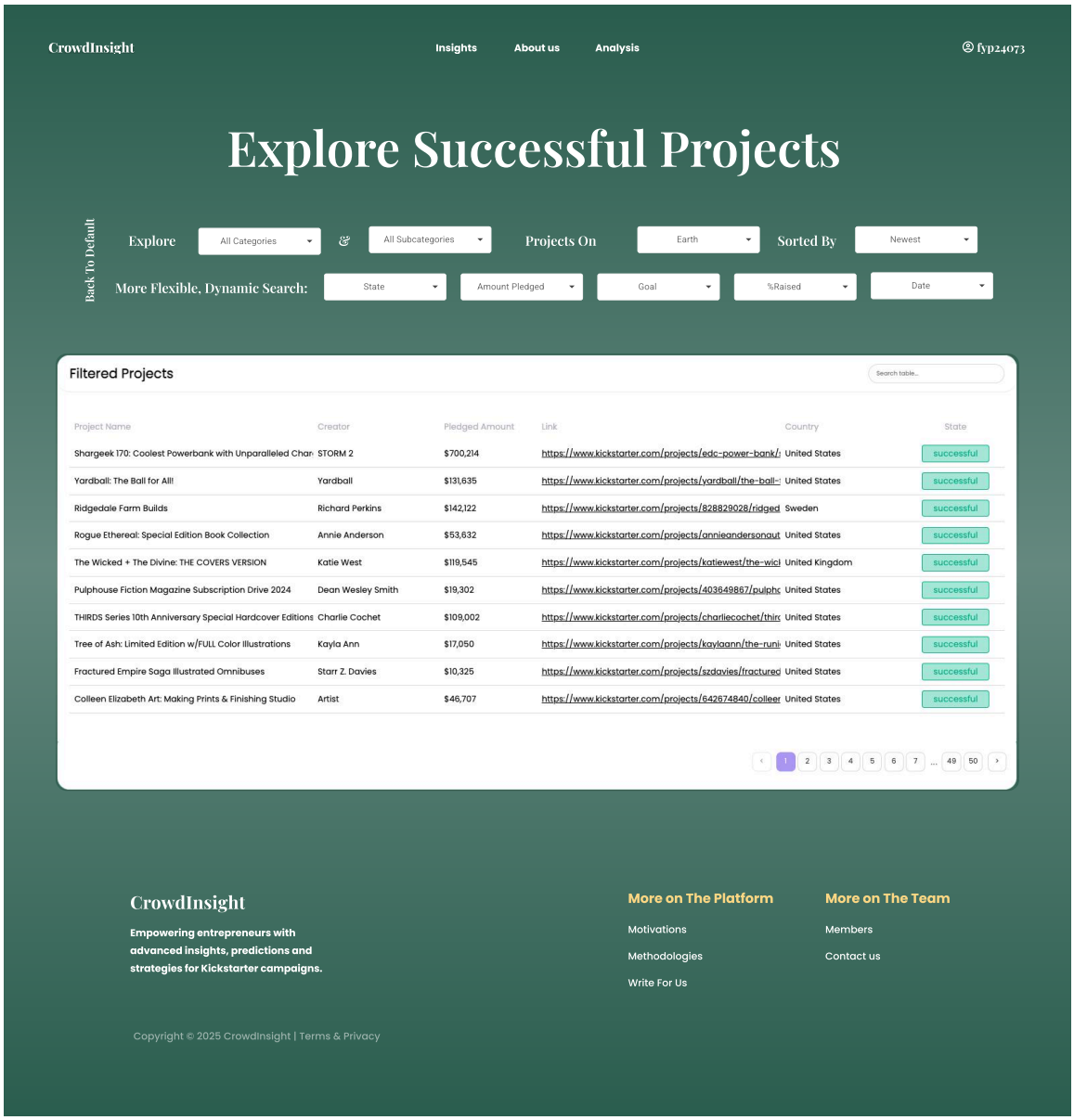


Figure A.3: Insights Section: Filters

Figure A.4: Analysis Section