AIDI 1002 - PROJECT SOW V1

Kickstarter Campaign Success Predictor

—

Terence Yu

Shail Patel

Sherap Gyaltsen

Jaspreet Singh Marwah

Github link: https://github.com/Zantorym/AIDI-1002-Project

I. Introduction

Kickstarters are a great way for people with creative project ideas to generate funds to materialize these ideas through community support. It allows innovators to work on projects that a community deems worthy of financial support, while not having to rely on companies or investors that may limit their creative freedom. It also allows the creators to have a much more intimate relationship with their target demographic throughout the process of materializing their ideas.

That said, Kickstarter projects only have a success rate of 39.20%. With the amount of time, effort, and energy required to launch a successful Kickstarter campaign, if would-be entrepreneurs could get an early prediction on whether their campaign would succeed, it would greatly help them manage their expectations and perhaps make changes to increase their chances of success. With that in mind, we plan to develop an AI system that leverages the extensive history of Kickstarter campaigns available publicly in order to predict whether a campaign will succeed.

II. The Dataset

We will primarily be using the following dataset from Kaggle:
https://www.kaggle.com/kemical/kickstarter-projects

It has the details of 321,616 kickstarter projects from 2010 to 2017. It has the following features:

- Internal Kickstarter ID
- Name of the project
- Category of the project
- Category of the campaign
- Currency used to support
- Deadline for crowdfunding
- Fundraising goal
- Date launched
- Amount pledged by “crowd”
- Current condition of the project

Additionally, we plan on (potentially) merging the kaggle dataset with the the kickstarter data from the following link: https://webrobots.io/kickstarter-datasets

The kickstarter data from the webrobots dataset contains updated entries from 2017 onwards but is in a different format, and some fields consist of json strings that would need to be parsed through to derive relevant information, so we will decide whether to use this dataset in conjunction with our primary dataset after our Exploratory Data Analysis phase. If it is possible for us to extract the relevant fields from the secondary dataset, such that we can reasonably train our model on the data we derive from it, we will merge the data with our primary dataset.

III. Schedule

Time Frame		Tasks
Start Date	End Date	
25-Oct	1-Nov	● Exploratory Data Analysis ○ Analyzing and summarizing the main characteristics of the dataset ○ Noting any abnormalities, such as missing data, and developing a plan on how the deal with them ○ Figuring out which features would be relevant to our objective and which would serve as noise ● Merging our secondary dataset with our primary dataset, if possible.
1-Nov	8-Nov	● Data pre-processing, cleaning and preparation.
8-Nov	8-Nov	PROJECT SOW V2 SUBMISSION
8-Nov	10-Nov	● Shortlisting classifier models to build based on insights from EDA phased ● Allocating team members to groups, each group handles one model
10-Nov	22-Nov	● Building, training and testing the models ● Comparing model performances using evaluation metrics and recording the values in a report
22-Nov	22-Nov	PROJECT MODELLING SUBMISSION
22-Nov	6-Dec	● Picking the best model based on performance report generated using evaluation metrics ● Fine tuning the best model for the prototype ● Reviewing documentation and code quality, amending where necessary

6-Dec	6-Dec	PROJECT PROTOTYPE SUBMISSION
6-Dec	13-Dec	• Final quality check • Ensuring everything is polished and ready for deployment
13-Dec	13-Dec	PROJECT DEPLOYMENT

IV. Preliminary Task Distribution

Task	Members			
	Jaspreet Marwah	Terence Yu	Sherap Gyaltsen	Shail Patel
Exploratory Data Analysis	✔	✔	✔	✔
Data pre-processing & cleaning	✔	✔		
Shortlisting Classifier model			✔	✔
Building, training, testing models (each person/group will work on a separate model)	✔	✔	✔	✔
Creating performance report			✔	✔
Fine tuning final model	✔	✔		
Final quality checks (final code needs to be unanimously approved by the group to pass)	✔	✔	✔	✔

V. Evaluation Metrics

Accuracy

If our EDA reveal the dataset to be balanced (or if we are able to balance the dataset during our data preparation phase), we will opt for accuracy as a performance metric, where

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

(TP = True Positive)
(TN = True Negative)
(FP = False Positive)
(FN = False Negative)

Precision & Recall

In addition to accuracy, we will also use precision and recall, where

Precision = (TP)/(TP+FP)

and Recall = (TP)/(TP+FN)

F1-Score

To ensure that there's a balance between our precision and recall, we will also use F1-scores, where

F_1 = 2 * (precision * recall) / (precision + recall)

Log loss

If our model's output consists of multiclass prediction probabilities (particularly if we opt for implementing a neural net), we will check for categorical cross-entropy using the following formula:

$$LogarithmicLoss = \frac{-1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} * \log(p_{ij})$$

Where

N is the number of samples,

M is the number of classes for the output,

Y_{ij} is 1 if the sample i belongs to class j (and 0 otherwise), and

P_{ij} is the probability our classifier predicts of sample i belonging to class j.