Predicting the success of Kickstarter campaigns

Nida Hussain
A12505277
nzhussain@ucsd.edu

Kareem Kamel
A53080318
kkamel@ucsd.edu

Archana Radhakrishna
A53099070
arradhak@ucsd.edu

ABSTRACT

The internet has become a popular medium for crowdfunding campaigns, aimed at raising money from large numbers of contributors. Kickstarter is on ef the most popular a crowdfunding platforms on the internet. This paper aims to predict the success or failure of a Kickstater campaign at launch time. We evaluate a number of binary classification models. Random Forests performed the best, acheiving an accuracy of 80.37% and precision and recall values of 0.78 and 0.74 respectively. Our models outperform related work done to predict the success or failure of Kickstarter campaigns [4] [2].

Keywords

Crowdfunding; Kickstarter; Logistic Regression; Random Forests; kNearest Neighbours; Support Vector Classifiers

1. INTRODUCTION

Much like it's not too distant ancestor cooperatives, crowdfunding aims to raise monetary contributions from a large number of people. While crowdfunding predates the internet, most of today's crowdfunding happens online through platforms, the most prominent of which are Kickstarter, Indiegogo, and Microventures. The crowdfunding model consists of three types of actors: the project initiator who proposes the idea and/or project to be funded, people or groups who back the idea, and a mediator (the "platform") that brings the parties together.

For the purpose of this research paper, we will focus on Kickstarter, named "Best Inventions of 2010" by Time magazine. Project creators can start crowdfunding campaign, choose a deadline and a minimum funding goal. If the goal is not met by the deadline, no funds are collected. While the platform is only open to creators from US, UK, Canada, Australia, New Zealand, The Netherlands, Denmark, Ireland, Norway, Sweden, Spain, France, Germany, Austria, Italy, Belgium, Luxembourg, Switzerland and Mexico, backers may be from anywhere in the world. Kickstarter charges each campaign a 5% fee on the total amount of the funds raised. An additional 3 to5% is applied as a processing fee. Kickstarter lays no claims to ownership over the projects and the work produced by the campaigns.

Kickstarter projects fall into any one of 13 categories and 36 subcategories: Art, Comics, Dance, Design, Fashion, Film and Video, Food, Games, Music, Photography, Publishing, Technology and Theater. In addition to its wide range of categories, we chose Kickstarter for its standardized webpage layouts. The standard orgaization of their HTML simplified the otehrwise grueling work required to scrape project pages. The scraped data was supplemented by a number of Kickstarter datasets downloaded from webrobots.io. The latest of these datasets was dated February 15th 2017, about a month old prior to the writing of this paper.

Since only about 44% of campaigns reach their funding goal, it would be extremely useful for creators to know the probability of success before launch. Given Kickstarter campaign data, we would like to predict the success or failure of a campaign. Given the binary nature of this task, we evaluate a number of binary classifiers on the datasets. The rest of this paper is organized as follows: section 2 discusss exploratory analysis conducted on the dataset, section 3 is a literature review of previous work, section 4 introduces the prediction task, section 5 discusses feature selection, section 6 evaluates dfferent classification methods used, and finally, section 7 discusses results obtained and finally section 8 concludes wraps up and discusses limitations and future work.

2. PRELIMINARY ANALYSIS

We started with a pre-scraped dataset consisting of 175,085 projects. Each project belonged to one state from the set (failed, successful, live, suspended, canceled). After removing the live, suspended and canceled projects, we sorted the data based on the launch time of projects and chose the most recent subset of projects of size 60000 to perform our experiments on. We reduced the size of the dataset in order to perform scraping of extra information regarding the projects in a timely manner. For each project in the reduced dataset, we scraped the Kickstarter website to get information about the number of pledge levels, their corresponding values, the length of the project description and whether or not it contained a video.

The dataset contains 59,956 projects starting from April 2015 to February 2017, which have raised a total of $880 million in pledges. The goal range of projects runs from under a dollar to $152 million. About 23% of projects are never funded and while about 11% are staff picks. Staff picks being a section users can browse on Kickstarter site. Kickstarter allows projects to last up to 60 days, but recommends that the duration of a project be under 30 days. These two values are reflected in the peaks at week 4 and week 8 in Fig 1. where number of projects vs duration of

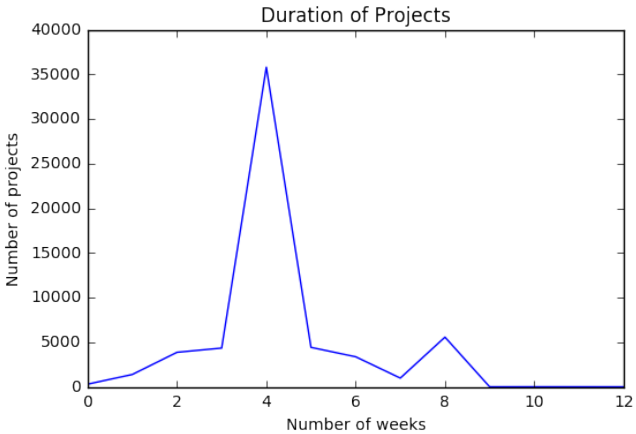

Figure 1: Number of projects per week

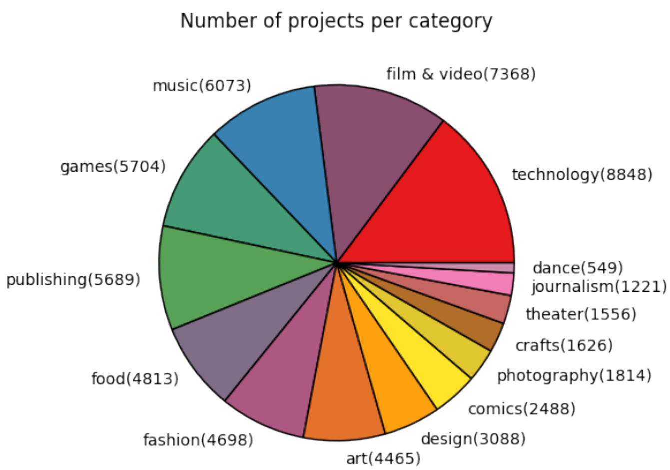

Figure 2: Number of projects per Category

project in weeks are plotted.

The projects are divided into 15 categories with Technology being the most popular and Dance being the least as shown in Fig.2. Interestingly, when we measure category popularity'' while using data spanning back to 2009, Music was the most popular category. Fig.3 is a graph of average goal of a category plotted against the success rate of the category showing that the categories with an average goal above \$200,000 have a less than 50% success rate. Design and Comics are the most successful categories, while Technology is among the least successful ones.

Projects rarely meet their goal exactly. If they succeed they are over-funded, sometimes many times over. The number of projects that are over-funded falls exponentially as we increase the number of times they are over-funded. About 25% of the projects are funded more than 70 times over their goal as shown in Fig.4. The projects are spread across 21 countries, with the United States being the most popular and Luxembourg being the least. Within the United States, California has the most projects (7463) and Wyoming has the least (60). A heat map of the number of projects per

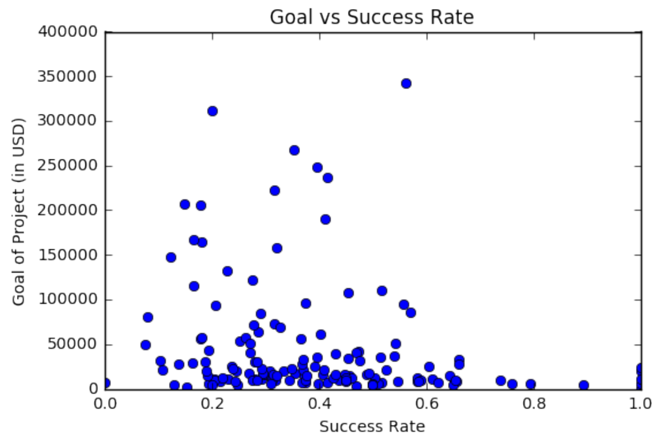

Figure 3: Average Goals vs Success Rate per Category

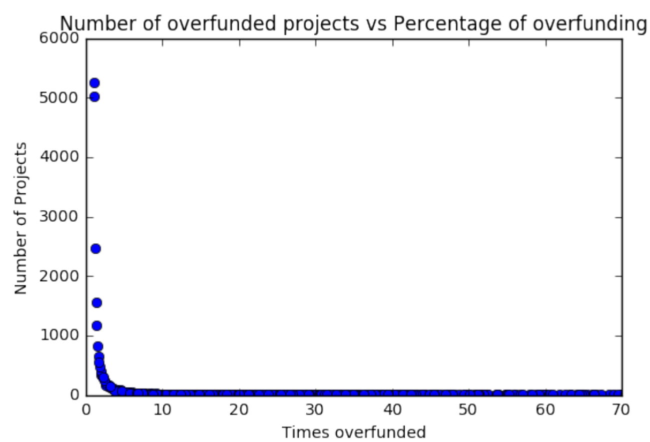

Figure 4: Number of projects and the times they are overfunded

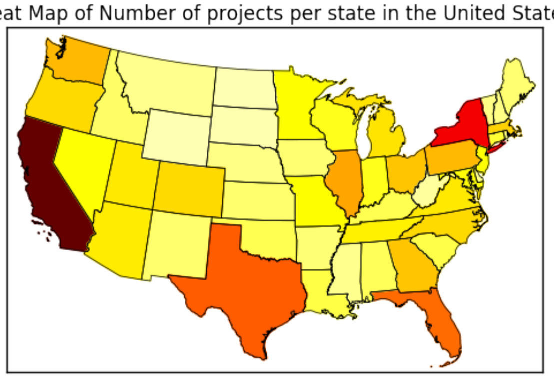

Figure 5: Heat map of number of projects per state in the US

state in the United States is shown below.

3. LITERATURE REVIEW

There have been several studies regarding the prediction of success in crowdfunding projects. Research done by Ethan Mollick of Wharton [1] focused on studying the correlation between geography, the social network of project creators, and project backers to the outcome of a project. His work shows that the higher the perceived quality of a project the higher chance of success and the factors that contribute to this perception. Etter et al. [2] use Markov Chain Modeling, using the time variant features of amount of money pledged and number of backers, combined with an SVM modeling the social features regarding the creators/backers of projects through their twitter activity. They are able to predict the success of a project with 76% accuracy.

Desai et al. [3] focus solely on text mining of the language used in project descriptions and compare the performance of models like SVMs, k-Nearest Neighbours (k-NNs) and multinomial Naive Bayes. Greenberg et al. [4] analyze the effect on project outcome of static features that are available from the launch of a project. They use SVMs and decision trees, assisted with boosting to build a classifier that achieves an accuracy of 67%.

From this landscape of prior work, we decided on a preliminary set of candidate features to experiment with while developing our models. The simplest of these are duration of project, category of project and the goal of the project in USD. The predictive task as well the as the dataset we have chosen limited us from choosing features relating to the social network and presence of associated people and features that become available as a project progresses such as comments, updates and number of backers.

The state of art in Kickstarter predictions have been real-time predictors. Work done by Etter [2] was able to achieve 75% accuracy within 4 hours of a project being launched. Their results were achieved using an SVM to combine individual k-NN and Markov chain predictors using social media in addition to data directly from Kickstarter. These results have been made available to the public in the form of a real-time scoreboard at a site called Sidekick [7].

Our findings were validated by the literature, specifically the predictive power of geography, goal amount and number of reward levels. As we will discuss later in the paper, we were able to achieve greater levels of accuracy than any of the papers discovered in our literature review by using Random Forests. Based on this literature we chose to include features, described in the next section.

4. PREDICTIVE TASK AND MEASURES

We will be predicting the success or failure of Kickstarter projects based on their initial conditions at project launch. We chose this task, rather than real-time tasks due to the features available in our data. Our dataset did not contain any transient data. Using the pre-existing dataset from web scraping startup Web Robots [8] and data we scraped ourselves directly from Kickstarter we created a dataset of 59,956 projects. This data scraping was an important task and resulted in our most predictive feature: number of reward levels. This set was split into a training set of roughly 35,000 projects, our validation set was 15,000 values and our test set contained 10,000 values. We randomized the order of our datasets before splitting and eliminated projects from our dataset with incomplete data for the features of interest.

4.1 Trivial Predictor

A trivial predictor was built that predicts the outcome of the project based only on the prior probabilities of successes and failures. The success rate of the dataset is 42.65%. The trivial predictor achieved a mean accuracy of 51.07%. We use this as a baseline to compare the performance of the rest of our models. We also calculated accuracy, precision, and recall for each of our classifiers and report those numbers against our naive classifier later on.

5. FEATURES

We used a mixture of simple features and complex features to perform our predictive task. Features which got their value directly from fields in our data were deemed to be simple features. Features that involved more computation and construction were deemed as complex features.

5.1 Simple features

5.1.1 Time features

As part of time features, we included a one-hot encoding of the year and month a project was launched at. However, they seemed not add much value to the performance of the classifiers. Projects that lasted around the 4 week mark were more successful than others, which led us to include the duration of a project as a feature. We converted the total duration of the project to weeks and added that as a feature.

5.1.2 Location features

As mentioned in the previous section, some countries were more successful than others and including the one-hot encoded representation of the country as a feature proved to improve the accuracy. We also experimented with adding a one-hot encoding of the states in the United States. However, that did not improve the performance of the models.

5.1.3 Category features

We included separate one-hot encodings for the category and the subcategory of the projects. The category feature had 15 values and the subcategory feature had 153 values. Our experimentation led us to conclude that while including both these features helps improves the performance of the classifiers, just including one of them does not help.

5.1.4 Text features

The ideal length of blurbs was found to be 22 words. The length of the blurb was included as features to model this.

5.1.5 Goal

As discussed in the above section regarding success rate of categories when compared with their average goal values, we found that the goal value provides information that can help predict success of the project.

5.1.6 Staff picked

While staff picked projects account for only about 12% of the dataset, their success rate is 83% almost double that of projects that are not staff picked.

5.1.7 Number of reward levels

Having more reward levels in a project provides incentive to backers to pledge money at various levels in an attempt to win rewards. This trend can be seen in projects being successful when they have a higher number of reward levels.

5.2 Complex features

5.2.1 Previous projects by creator

We sorted the data and for each project we calculated the number of previous projects that had been successful by the creator. As users create more projects, they seem to get an idea of what works and get more successful. We combined the staff picked feature with the number of successful projects by creator to create a measure of perceived reliability of the project.

5.2.2 Average money per reward level

The goal of the project was divided by the number of reward levels to get a an average money per reward level. This feature helped in improving the accuracy of the models.

5.2.3 Preparedness

Having a video in the project and including a URL that links to relevant information in the blurb shows a measure of preparedness. We created a feature to measure this by adding these values. This helped in projects being more successful.

5.2.4 Distance to mean blurb length

We calculated the mean blurb length of the projects that were staff picked to be about 18 words. We then calculated the square of the difference between a project's blurb length and the mean. This allowed us to penalize blurbs that were too short or too long.

6. MODELS

We explored the effectiveness of four different classification techniques in increasing complexity, k-nearest neighbors (k-NNs), followed by logistic regression, support vector classification, and finally random forests. We used a variety of methods to optimize parameters for each of these models. For k-NNs we used used systematic addition of features to determine the features with greatest predictive power and also compared performance on k sizes from 1 to 100 in order to find the best tradeoff between time and performance. For logistic regression, our ablation experiment is described below.

SVC took the longest comparatively to run. After initial search with fewer iterations, we narrowed down our C and gamma values to the set described in Table 1. We ran SVC with an rbf kernel, finding that polynomial, sigmoid, and especially linear, while much faster to converge performed very poorly. This is likely due to the lack of complexity allowed by those kernels in separating the feature space. Our gamma value is quite low and represents the extent to which projects should match each other. Given the sparsity of our feature set, it would make sense that we would not require a high level of similarity across feature vectors. Our C is indicative of the complexity of the model and the extent to which it is penalized. Initial experiments showed that c values above 100 were significantly more time consuming with lower accuracy, so we capped our exploration at C = 20.

For the Random Forest Model, we experimented with different values of estimators. Using a low number of estimators, like 10, was not as accurate. This might be due to the number of estimators not being able to capture the complexity of the features. Although Random Forests are more robust to overfitting, having a large number of estimators, like 100, proved to be detrimental to the performance of the model on the validation set. The optimal value of number of estimators was found to be 30. It captured the features and didn't overfit the data.

6.1 Ablation experiment

Using the above set of features, we performed Ablation experiments to find the most important features, using Logistic Regression. The simple features like goal, staff picked, length of blurb and number of reward levels proved to be the most useful features with the accuracy dropping from around 71% to 66-68% when they were removed. The number of reward levels was the most important feature with accuracy dropping to 60% when it was removed, a mere 10% above a tivial predictor. We removed features like length of project description and one-hot encoding of campaign's state, since they added no value to the classifier.

7. RESULTS

7.1 kNearest Neighbours

Using k=53 neighbours with k-NN proved to be optimal in that it gave the highest level of accuracy. As compared with logistic regression, we found through experimentation that only 4 of our simple features were of predictive value: the goal amount, the length of the project description, the number of reward levels, and the average dollar amount per reward level. Using only these simple features, we were able to attain an accuracy of 72.35%, with precision of 0.66 recall of 0.75 and an f-score of 0.69. Performance actually worsened when including our complex features, eg number

Table 1: Performance with different values of C, γ

	C = 1	C = 10	C = 20
γ = 1e-7	72.59	72.03	71.47
γ = 1e-8	70.58	71.26	71.1

Table 2: Performance of Models

Model	Accuracy(simple)	Accuracy (complex)
kNN	73.35	68.66
Log. Regression	69.08	72.51
SVC	71.55	72.59
Random Forest	78.95	80.37

of pledge levels, previous successful projects done by creator, etc. The mixed feature set gave an accuracy of 68.66%, with precision of 0.63, recall of 0.63 and an f-score of 0.63. These numbers are reported for the test set. Our accuracy was slightly higher in validation, above 1% higher but not 2% at any point.

7.2 Logistic Regression

We performed logistic regression on the train set with L2 regularization and the value of λ was tuned to be 0.00001. The classifier run with just simple features yielded an accuracy of 69.08%, while adding the complex features increased the accuracy to 72.51%. The values of precision, recall and f-Score for the model with both simple and complex features were 67%, 69% and 68% respectively for the test set.

7.3 Support Vector Classifier

We built Support Vector Classifiers [6] and found that our optimal parameters were $C = 1$ and $\gamma = 1e - 8$. The maximum acuracy achieved was 72.58% and was acheived on a feature set which included all of the features described including our complex features. The precision, recall and f-score were 0.63, 0.78, 0.69.

7.4 Random Forest

Random forest classifier builds decision trees and outputs the mode of the predicted classes as the class of the data point [5]. We built a random forest classifier and tuned it using the validation test to get the best number of estimators as 30. The random forest classifier is robust to overfitting and provided an accuracy of 78.95% on the test set, with only simple features. Including the complex features provided a slight increase in accuracy to 80.37%. The values of precision, recall and f-Score were 0.78, 0.74 and 0.76 respectively for the test set.

8. CONCLUSION AND FUTURE WORK

Random Forest proved to be the best classifier for the task, providing an accuracy of 80.37% and a precision and recall of 0.78 and 0.74 respectively. The parameter we fitted was the number of estimators. Using a large number of estimators would have led to overfitting, while using a small one would have not achieved the best accuracy possible. 30 estimators was optimal regarding both the aspects. All the models we implemented performed better than the baseline of a trivial predictor. The results of the models we experimented with have been summarized in table 2. The performance of this model also outperformed the results obtained by related work [4] [2] for a similar predictive task despite Etter et al.[2] using a number of additional features obtained after the launch of a campaign.

Extensions to the experiments we performed would be to combine the models we built using AdaBoost and try to improve the prediction rate. If we relax the initial constraint of launch time, we may be able to use temporal and social features as the project progresses to obtain more up to date indicators of the performance of a campaign. This would help in performing better as well as predicting success in real-time.

9. REFERENCES

[1] Mollick, Ethan. "The dynamics of crowdfunding: An exploratory study." Journal of business venturing 29.1 (2014): 1-16.

[2] Etter, Vincent, Matthias Grossglauser, and Patrick Thiran. "Launch hard or go home!: predicting the success of kickstarter campaigns." Proceedings of the first ACM conference on Online social networks. ACM, 2013.

[3] Desai, Nihit, Raghav Gupta, and Karen Truong. "Plead or Pitch? The Role of Language in Kickstarter Project Success." (2015).

[4] Greenberg, Michael D., et al. "Crowdfunding support tools: predicting success and failure." CHI'13 Extended Abstracts on Human Factors in Computing Systems. ACM, 2013.

[5] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

[6] Suykens, Johan AK, and Joos Vandewalle. "Least squares support vector machine classifiers." Neural processing letters 9.3 (1999): 293-300.

[7] http://sidekick.epfl.ch/

[8] https://webrobots.io/kickstarter-datasets/