Predicting Kickstarter Project Success

Patrick Vacek

March 14th 2018

Contents

1 Introduction

The concept of crowdfunding is still in its infancy, and with this comes the challenge of separating ”meaningful” projects from ”meaningless” projects. At the forefront of this crowdfunding revolution lies *Kickstarter*. Like any other major tech company, *Kickstarter* keeps track of the essential details of each project. However they try to make access to more informative features challenging for the layperson. We seek to determine the factors – both explicit and implicit – that make an effective classifier for determining successful projects.

2 Dataset

The dataset from Kaggle has over 300,000 *Kickstarter* projects from 2009-2017.[2] Initial excursions into this dataset reveal that the patterns may be fairly noisy. The month with the largest amount of projects launched was July 2014, and more interestingly it had the lowest success rate at approximately 20%.

Initial analyses were conducted with the full category set, but it soon became obvious that the effects of predictors depended on the project type. This has to do with the fact that *Kickstarter* requires users to specify a project type, so it can be placed in one of over 150 categories. Instead, the project scope was narrowed down to the five most populous categories: *Food*, *Product Design*, *Art*, *Music*, *Documentary.*

Original	Transformed	Description	Units
Category		The category of the project.	Categorical
USD Goal Real	*Normalized Goal*	How large the project goal is compared to the group	log (US Dollars), adjusted for currency, inflation
launched,deadline	*Normalized Duration*	How many days the project has to complete its goal compared to the group	Days
Country		The country that the project was launched in.	Categorical
state	*outcome*	Coded as 0 if state was *Failed* or *Canceled* and 1 if *Successful*	Binary

Furthermore, it should be noted that the original dataset had six countries present: *United States*, *United Kingdom*, *Canada*, *Australia*, *Netherlands*, *New Zealand.* To avoid sampling issues with respect to cross validation, *Netherlands* and *New Zealand* were dropped because their counts were less than 20 in the reduced dataset.

3 Exploratory Data Analysis

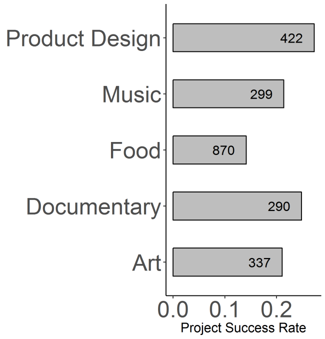
We start this section by examining a table of the summary statistics of all five variables:

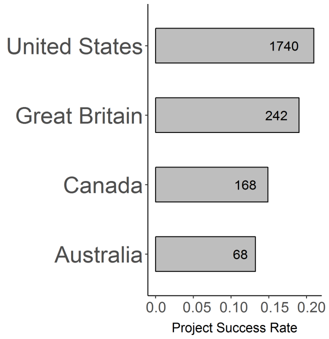
Variable	Summary
Category	*Food*: 870, *Product Design*: 422, *Art*: 337, *Music*: 299, *Documentary*: 290
Normalized Goal	*Min*: -5.083, *1Q*: -.7898, *Median*: .0300, *3Q*: .7170, *Maximum*: 4.315
Normalized Duration	*Min*: -2.941, *1Q*: -.3519, *Median*: -.1833, *3Q*: -.1804, *Maximum*: 2.6690
Country	*United States*: 1740, *United Kingdom*: 242, *Canada*: 168, *Australia*: 68
state	0: 1773, 1: 445

There are a few essential things we can see from examining these summary statistics. First, *Food* has notably has the most projects, which may be interesting to see later on. There are also major outlying values in the normalized goal variable. There was a specific observation in the dataset that caused the positive outlier, where someone launched a project called *”FUCK Potato Salad. Paleo Potato Brownies!”* and asked for $166,361,391 as their goal. This request was obviously not met. Conversely, the negative outlier was called *”Documentary About Potato Salad”*, which gives a hint about the flavor of this dataset.

There is also an inherent American presence to *Kickstarter*, since most of the projects are U.S. based. Lastly, there is somewhat of a class imbalance in the data, with approximately 80% of the projects failing. This imbalance may make accuracy measurements more challenging. For all future accuracy metrics, we will define the accuracy rate to set predictions as 1 if the value exceeds the class average, and 0 otherwise.

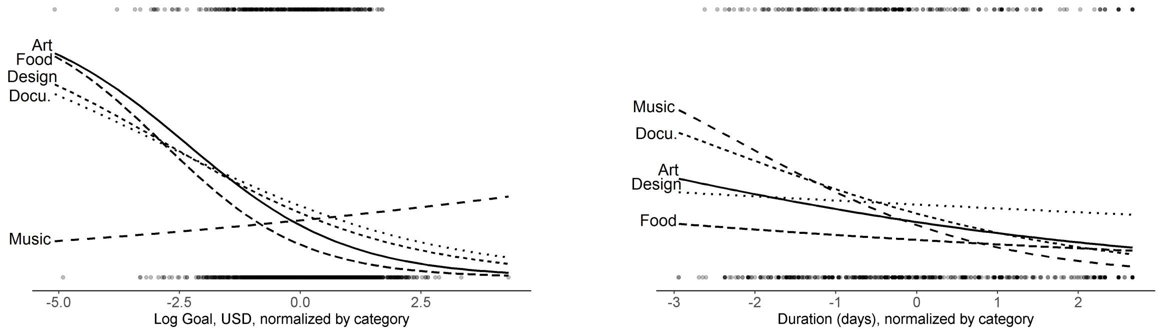
We now move towards answering some essential questions regarding the relationships between the variables. It is important to determine how the success rate changes depending on the categorical variables, so two bar plots were produced:





There are no specific categories or countries in which the success rate is much higher or lower. The most interesting thing that can be inferred is that the *Food* category has the lowest success rate, but with the highest count. This may give some credence to the idea that each project type functions as a "market", with lower quality ones losing out on funding compared to others. This may challenge the assumption of independence, although the previous statement is simply conjecture.

We now examine the effects that the continuous variables have on project success, with each effect being modeled by category. Note the estimated effects were modeled with a logistic fit. Using more conventional smoothers – such as B-Splines and G.A.M. – would have noisy curves which would make the plot harder to interpret. It is also important to remember the essential goal of this project is prediction; we want to see how the effects look inside the model, not outside.



(a) Goal

(b) Duration

Most of the results from these graphics appear fairly intuitive. There is a strong negative effect in asking for more money comparatively for most categories. There is a slight negative effect for having a longer lasting project, which may be explained by the viral nature of crowdfunding. Interestingly enough, Music has a positive effect for the Goal variable. This suggests that Music projects that ask for more money are more likely to succeed, meaning that there may only be serious projects in this category.

4 Model Performance on Primary Features

Since a prediction problem is being undertaken here, we are not concerned with the exact effects or diagnostics, but rather the predictive power. Each model was fitted using five-fold cross validation, with the *Area Under the Curve* and accuracy at the class cutoff being reported back. Four separate models were fitted:

1. **Model 1: Baseline Model** outcome˜goal. We simply consider how much the person is asking for compared to the other projects in their category. This model did not perform very well since it had an AUC of .6043[4]. This is likely because the model did not account for the difference in effects between categories. The accuracy for the model is also .5564, which is poor.

2. **Model 2: Category-Goal Model** outcome˜goal*category. We now consider what category the project is in and what the goal is comparatively. This model performs better in terms of AUC, with a score of .6527. The accuracy is lower (.5406), which means that the classifier is likely misclassifying failed projects as successes.

3. **Model 3: Country Model** outcome˜goal*(category+country). We now consider which country the project is based in when determining whether a project will be successful. This lowers the AUC to .6472 and barely increases the accuracy to .5423 so it appears that it is not meaningful to consider the country of the project.

4. **Model 4: Full Model** outcome˜(goal+duration)*(category+country) We now see if considering the project duration will have a significant effect. Adding duration increases the AUC to .6610 and increases the accuracy to .5739. This seems to be the best model when balancing out AUC and accuracy, when restricted to the initial features.

5 Auxiliary Variables

While some of the more informative features were not present in the Kaggle dataset, it was still possible to extract them through a data archive. Additional data up through August 2014 was found in a large JSON file.[3] The auxiliary features were scraped with the help of Bash and then joined to their corresponding project ID. Two essential features were extracted. While it may have been useful to get more, there was not enough time to formalize any relationships for variables beyond the two features.

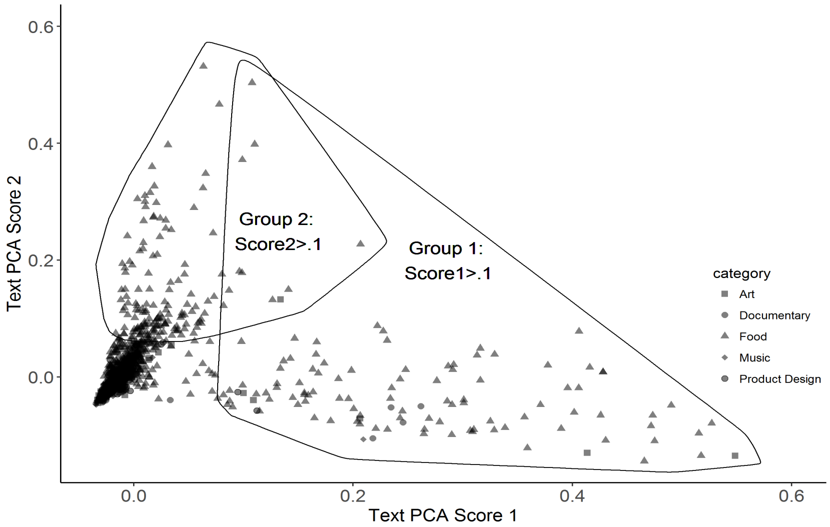
The first of these new features is called **Text**. It was created by combining a previously less informative feature from the Kaggle dataset (the project title) with a newly-extracted **blurb** feature. The intent of this variable is to perform analysis on the text. It is wise to have as much text as possible for each project to get an idea of what the creator is talking about, hence the merging. The application of this variable will be discussed in the next section.

The other feature is **Hours**. This variable was obtained by scraping the time of project creation and then taking the difference between project launch time and creation time. This variable was very right-skewed in shape – potentially due to some creators working extra hard on their projects – so the values were log transformed. Since this variable is continuous, it was normalized by each respective category. Intuitively, the non-transformed variable's measurement was in hours.

6 Text Mining

The purpose of including the **Text** variable is that there actually can be useful information extracted from it. In order to extract this information we must use *Natural Language Processing.* The essential concept that is used from this topic is the construction of a *Term-Document Matrix.* Essentially, all of the projects are stored as rows and the incidence of non-trivial words are stored as columns. A more detailed explanation of this process is covered in Appendix B.

The *Term-Document Matrix* is generally very large and has interesting patterns that cannot be detected very easily by the human eye. In order to reduce the complexity of this structure, **Principal Components Analysis** was implemented on the matrix. The rationale for using P.C.A. here was to use an ad hoc dimensionality reduction technique; the components are less of interest but rather the potential clusters. We can get an idea of how the text is structured from the visual below:



There is obviously a non-trivial pattern in the data here. While most of the documents are centered around zero, there are two specific sets of documents deviating away from this cloud. A notable subset of documents have large scores on either the first or second axis, but there are not many in both. It is also apparent that a large amount of the outlying documents are from the *Food* category.

The heuristic cutoff selected was to allocate the document into an outlying group if one of its P.C.A. scores was larger than > .1. While this may not be a proper cutoff, using a formal clustering technique would have been too involved for this project. The most common technique, *k*-means, would be a poor choice because of its tendency to produce equally-sized clusters.

Now that there is evidence distinct text groups exist, we want to see which potential words are causing the variation in the text data. By examining the most frequent words in each group, the result becomes more clear:

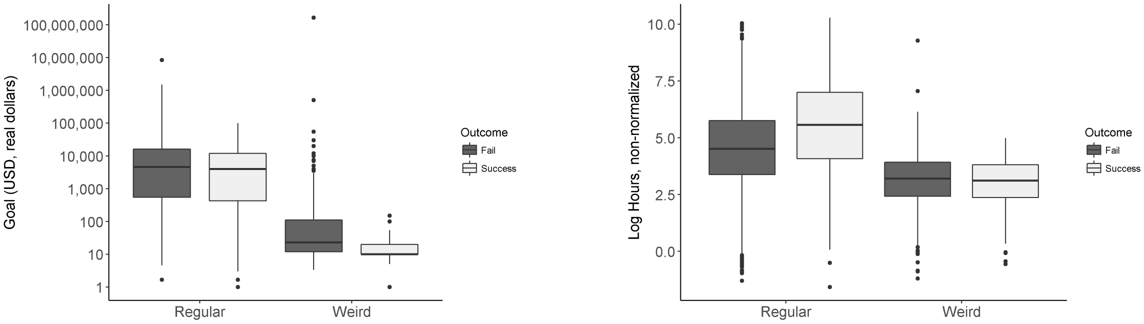
Group	first	second	third	fourth	fifth
Low Scoring	make	help	want	need	music
Score1>.1	salad	potato	make	want	better
Score2>.1	make	pizza	want	cookie	chip

While all groups share some commonality with their syntax, the most striking difference can be seen with the unusual word choices in high-scoring groups. It seems the group with a large Score1 is frequently mentioning "potato salad"; the large Score2 group is mentioning pizza and cookies. This actually has to do with a very infamous *Kickstarter* project where Zack Danger Brown asked for \$10 to make potato salad, but ended up receiving \$55000.[6] It seems likely that most of these are copycat projects. We now wish to separate these projects from the more serious ones, so we define a new variable called **Weird**:

$$\textbf{Weird} = \begin{cases} 1 & \text{Score1} > .1 \cup \text{Score2} > .1 \\ 0 & \text{otherwise} \end{cases}$$

7 Exploring Auxiliary Features

We now examine the effects of **Weird** on both the **Goal** and **Hours** variables. The visuals paint a very striking picture:



(a) Goal

(b) Hours to launch

The most obvious takeaway from this is that **Weird** projects do not get funding for very large amounts of money. We can see even amounts small as $100 are considered unusual if the project was successful. This indicates that Zack Danger Brown's feat was largely a one-off occasion. We can also infer two essential things from the plot of the hour effects. First, regular projects tend to have a higher success rate if the **Hours** variable is longer; this effect is not present for the irregular projects. Second, the **Weird** projects tend to have much lower hour totals, which implies the users are not taking their work seriously. Based on this evidence, it was decided placing the irregular projects in a separate "Alternative" category would be useful for improving prediction power.

8 Model Performance with Auxiliary Features

We now turn back to model performance, with the **Category** variable being replaced by **Category2**.

1. **Model 5: Goal-Hour Model** outcome˜(goal+hour)*category2. This can be considered as the most parsimonious model that includes the updated features. It significantly outperforms the full model from the last section, giving evidence to the idea that the new features are useful. The AUC for this model is .7028 and the accuracy is .6366.

2. **Model 6: Goal-Hour-Duration Model** outcome˜(goal+hour+duration+goal*hour)*category2. This model exists to test how much the AUC can be increased by including nuisance variables. This model results in a small increase of the AUC (.7213) but has slightly lower accuracy (.6344). It is also not as easy to interpret as the previous model.

The final model selected was Model 5. A full description of this model is included in Appendix A.

9 Conclusion

Analyzing *Kickstarter* projects can be quite an ordeal. This can be evidenced by how different certain variable effects are across categories. It is also challenging because there may be "unlabeled" project types nested inside the categories. This may not hold for all months of *Kickstarter*, as July 2014 was chosen due to its very strange patterns compared to other months. However, it is possible to get a fairly reasonable model for this data, as we have seen by the results of Model 5. With only three major predictors – plus the behind-the-scenes text mining – it is possible to get a good idea of whether a project will fail or succeed.

There were definitely a handful of setbacks when working on this project. One of the main setbacks was how using the Lasso was not effective for this data. Justification on the omission of Lasso is included in Appendix B. Another issue that appeared was the presence of "canceled" projects. It would not make sense to drop these, as it would have made the models over-perform due to obvious bias. A good suggestion that was considered was to use a zero-inflated model; however, this was not feasible due to time constraints. There is major potential to examine this data further, as the performance can still be improved. This project did not seek to perfect prediction, but rather to demonstrate the methodologies that would build an effective classifier.

10 Appendix A: Description of Final Model

The final model selected, among all six candidates to describe the solution to the problem is Model 5. The reason of this being is that it paints a parsimonious picture about how the projects can be predicted with a strong accuracy rate. The equation for the model is:

$$logit(E(Y|X)) = \beta_0 + \beta_{10}\tilde{X}_{Goal} + \beta_{20}\tilde{X}_{Hours} + \sum_{j=1}^{5} \mathbf{1}_j(\beta_{1j}\tilde{X}_{Goal} + \beta_{2j}\tilde{X}_{Hours}) \tag{1}$$

Where the $j = 1, \ldots, 5$ corresponds to {Alternative, Documentary, Food, Music, Product Design} with *Art* being the reference class due to the category having the most similar success rate to the baseline. This model is useful in the sense that someone who uses it can get a solid estimate of the probability of project success by asking a set of three questions:

- What category does their project fall under?
- How much money is the project maker asking for compared to project makers in the same category?
- How many hours of effort did they put into their project?

Once these three are answered, a valid estimate can be obtained about how likely their project will succeed. No coefficient values are included due to the fact that cross validation was utilized.

11 Appendix B: Miscellaneous Topics

11.1 Justification of Lasso Omission

Lasso regularization is usually an effective tool for dealing with logistic regression for sparse datasets. It can be useful for picking out significant features from a noisy dataset. It works especially well in the case of multicollinearity and large sets of features. The optimal choice of λ can even be determined with respect to AUC, which would sound perfect for this kind of problem.

Unfortunately this did not work as intended for the dataset. Here are a few reasons why:

- **A lack of continuous predictors** Even after getting the auxiliary data, there were only three continuous predictors. Using quadratic terms and their interactions did not improve the prediction power either.
- **The category/country interactions were not meaningful** While it may seem intuitive to think that Lasso could pick out significant categorical interactions, this was not the case. The two categorical variables, **Category** and **Country**, did not have any meaningful patterns that could be attributed to noise.
- **Lack of multicollinearity** The relationship between **Duration** and **Goal** seemed to have a mild positive correlation, but the **Goal** and **Hours** variable were close to zero correlation after standardization.
- **Using a tuning set was restrictive** Having to fit the lasso on a separate tuning set diminished the effects of the predictors. There were also outlying values in **Goal** for example, so it may have caused major differences in the optimal model across all sets. Furthermore, it would have been difficult to convey classification metrics (such as AUC) because the metric may have depended on the choice of tuning set.

11.2 Natural Language Processing

Below is a detailed step-by-step of how the project was transformed from **Text** to **Text Scores**:

- The original text data had to be cleaned. Certain special characters (e.g. slash, double comma) were removed from the text.
- We had to keep our data as *a priori*. This was an issue because some of the canceled projects added *(Canceled)* to their title. This would violate the assumption if it was left in, so all mentions of "canceled" were removed.
- Words that are common in the English language are called *stopwords*. All of the stopwords in the text were removed.
- The remaining words were *stemmed*, so that they could be joined together if the beginning of the word was the same. This meant words like *documentary* and *documentaries* became *documentari*.
- The term-document matrix is constructed with n rows and p columns, where p is the number of unique words across all project texts.[5]
- The metric used is called *term frequency–inverse document frequency* or tf-idf for short. An informal sketch of the equation is as follows:

$$tf_idf(t,D) = tf(t,D) \cdot idf(t,D) = \frac{|\text{Occurences of t in D}|}{|\text{Words in D}|} \log \frac{n}{|\text{Occurences of t in Documents}|} \quad (2)$$

- Principal Component Analysis was implemented on this matrix, hence resulting in the text scores seen earlier.

12 Appendix C: References

References

[1] Hans-Georg Müller. *Generalized Linear Models: Lecture Notes.*

[2] Kaggle. *Kickstarter projects: More than 300,000 Kickstarter projects.*
https://www.kaggle.com/kemical/kickstarter-projects

[3] Web Robots *Kickstarter Datasets.*
https://webrobots.io/kickstarter-datasets/

[4] Thomas G. Tape *The Area Under an ROC Curve*
http://gim.unmc.edu/dxtests/roc3.htm

[5] Brandon Rose *Document Clustering with Python*
http://brandonrose.org/clustering

[6] Zack Danger Brown *Potato Salad: I'm making potato salad.*
https://www.kickstarter.com/projects/zackdangerbrown/potato-salad

13 Appendix D: Code

The project code was very immense and spanned multiple programming languages. Because of this, the code is not included in the report, but instead can be viewed at:
https://github.com/pvacek/UC-Davis/tree/master/STA%20223