Kickstarter: A Consumer's Gamble

Kandy Ho, Roger Miao

California State University, Sacramento

kandyho@csus.edu
rogermiao@csus.edu

***Abstract*— The problem when backing a Kickstarter project is that if the project doesn't get fully funded, you have essentially wasted your time and effort in backing the project. We are trying to use classification to help predict whether a project will succeed or fail based on a dataset populated with Kickstarter projects from Kaggle [3]. This can help consumers decide whether they want to back the project or not. SVM, Logistic, and MLP algorithm will be used to help us with classification. Logistic and MLP scored a similar result as SVM, however given a larger dataset Logistic and MLP will score higher than SVM.**

I. Introduction

For many entrepreneurs, Kickstarter is a useful tool to gain sufficient capital to fund the development of their product. The incentive of receiving capital without giving up any equity is very appealing. Because of this, Kickstarter is saturated with projects scrambling to receive funding. Many projects are simply created for the sole reason of receiving funding.

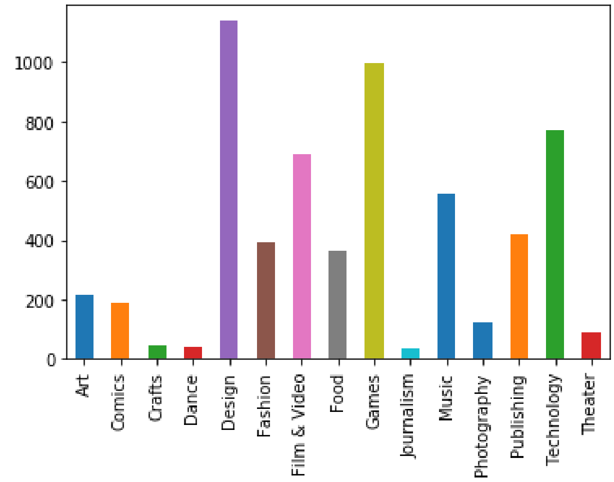


Fig. 1 Bar Graph of Categories

According to the the figure above: Design, Games, and Technology are the most common projects on Kickstarter.

The goal of this project is train a model using past Kickstarter projects and the outcome of said project. We then used that model to naively predict the outcome of current live projects. To achieve this, we started off by pruning our data. We removed projects with the state, 'cancelled' and 'undefined'. These were removed because we believed these two states would not be helpful for our predictions since these states are usually outliers. We also removed older projects and kept more recent projects from 2017-2018. We then separated the dataset into two categories: Finished Projects and Live Projects. We then used One Hot Encoding, Label Encoders, and Standard Scalers to prepare our data for training. We then performed parameter tuning to find the best parameters for training. After the parameter tuning, we trained our models with the Support Vector Machine (SVM), Logistic Regression (LR), and Multi-layer Perceptron (MLP) models with the Finished Projects. With these trained models, we naively predicted the outcome of the Live Projects. The contributions of the paper can be summarized as follows:

- Our experiments demonstrated that LR outperforms the SVM and MLP models in predicting the outcomes of the projects.

II. Problem Formulation

The main problem with Kickstarter projects is that a lot projects end up not being fully funded and will never reach the development phase. After months of waiting there is a chance that you will not see any progress. If you were able to predict whether a project could succeed or fail before the funding phase ends, it can help you decide whether you want to back the project or not. Given data such as Goal, Pledged Amount, Active Funding Days, Category and Main Category, you are able to run classification algorithms on these projects. The output you are looking for is whether a project will succeed or fail.

III. System/Algorithm Design

3.1 System Architecture

The design of our system was rather straightforward, our focus was pruning the data properly, since we were working with data that ranges from 2010 to 2018. We decided to focus only on the recent years, which included 2017 and 2018. We also removed the arbitrary features. After that, we transposed the category dataset, so that each category would become a feature with a binary representation. Since we were given the launch date and deadline, we decided to get the difference of the two dates. This gave us the total active days of each project. Since the values for goal, pledged, and active days greatly differ from the rest of the data set, we had to normalize those features. After getting the data pruned and ready, the next step was to use train test split to divide the dataset into testing and training data. We were experimenting with the size of the testing and training data. It ranged anywhere from 20% to 40% testing data. The next step was to run it through our chosen algorithms and perform parameter tuning. The metrics used to measure our algorithm are F1 score, Precision score, and Recall score. We are also going to plot the ROC curve of each individual test.

3.2 *Classification of Kickstarter Projects*

In this section, we will talk about the algorithms used for our classification of Kickstarter projects. Our algorithms will focus on predicting whether a project succeeds.

3.2.1 SVM Algorithm

SVM algorithm is used for linearly separable binary sets. The goal of SVM is design a hyperplane that classifies all training vectors into two classes. SVM takes in optional parameters, C and kernel. The C parameter affects the margin width between the two different classes. Different kernel parameters compute the similarity between each data item differently. We used grid search to perform parameter tuning on our algorithm to figure out which C and kernel value best fit our data.

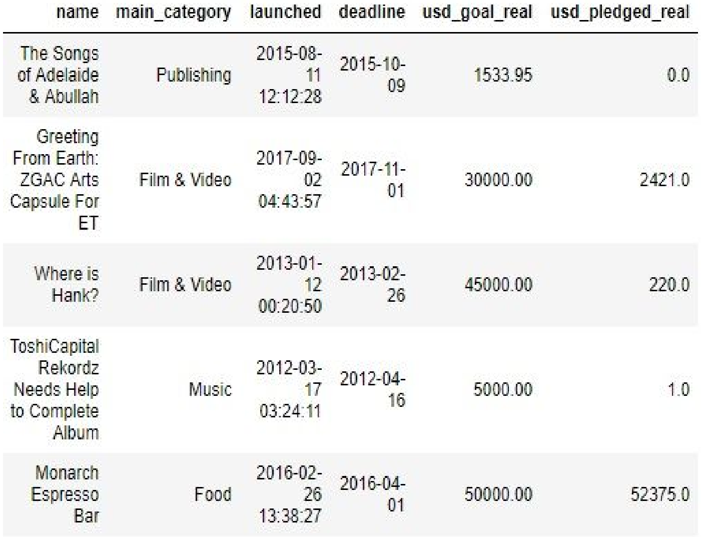
3.2.2 Logistic Regression Algorithm

Logistic Regression will model the probability of an event occurring and then estimate the probability of that event occuring for a randomly selected observation versus the probability of the event not occuring. It will then estimate whether our Kickstarter project fail or succeed. The parameter tuning on this algorithm is to find the right C value for Logistic Regression. The C value will help minimize the error between what our model predicts and what the ground truths are. The parameter tuning will help us find the C value that will help us minimize the error value.

3.2.3 MLP Classifier Algorithm

The MLP Classifier Algorithm uses hidden layers and neural network to predict if Kickstarter projects succeeds. The parameter tuning on MLP Classifier is rather inefficient making it undesirable for anyone with hardware limitations. However, we believe that the MLP Classifier may be more consistent when the data is noisier. In our grid search, we created a loop to check all combination of hidden layers with values 10 or 30. The learning_rate was also a parameter that we used, however the search only checked 3 different types of learning rate.

IV. Experimental Evaluation



name	main_category	launched	deadline	usd_goal_real	usd_pledged_real
The Songs of Adelaide & Abullah	Publishing	2015-08-11 12:12:28	2015-10-09	1533.95	0.0
Greeting From Earth: ZGAC Arts Capsule For ET	Film & Video	2017-09-02 04:43:57	2017-11-01	30000.00	2421.0
Where is Hank?	Film & Video	2013-01-12 00:20:50	2013-02-26	45000.00	220.0
ToshiCapital Rekordz Needs Help to Complete Album	Music	2012-03-17 03:24:11	2012-04-16	5000.00	1.0
Monarch Espresso Bar	Food	2016-02-26 13:38:27	2016-04-01	50000.00	52375.0

Fig. 2 Pruned Dataset with Related Features

4.1 Methodology

Main Category, Launched, Deadline, USD Goal, and USD Pledged were the top features that were used for our algorithms. After all the pruning and one hot encoding, our dataset was ready to be split into training and testing. Our pruned data set included 18871 projects. Seventy percent of the data set was used to train our model while the other thirty percent was used for our testing data. After the initial training and testing, we tested the models on our Live Projects data set.

We tried several different training and testing data sizes, to see if any algorithm would plummet in predicting the ground truths. The first set of features tested together was main category, goal, and pledged, however this only allowed us to reach up to an average of 0.78 f1 score. We then decided to get the active days of the funding stage by getting the differences of launched and deadline date and adding that to our data set.

To measure the algorithms against each other, we used the f1 score, precision, and recall to see how accurately each algorithm performed. Since the score ended up being so close to each other, we plotted an ROC curve to see the results on a graph.

We implemented other methods that ended up not being as accurate as SVM, LR, and MLP. The Naive Bayes model and the k-NN model was attempted, however the score was either incomparable or inefficient. Either LR or MLP would be our method of choice to use.

4.2 *Results*

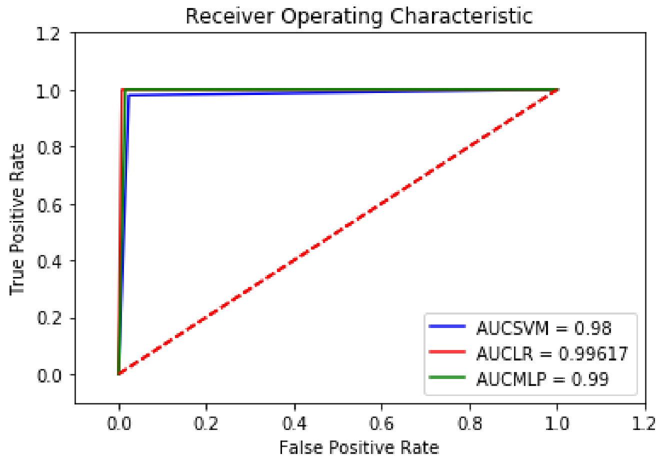


Fig. 3 ROC Curve

As you can see from above, Figure 5, both LR and MLP overlap each other while SVM is slightly lower than both. This makes SVM our baseline approach because both LR and MLP have a greater area under the curve than SVM. You can conclude that either LR or MLP is the better choice over SVM.

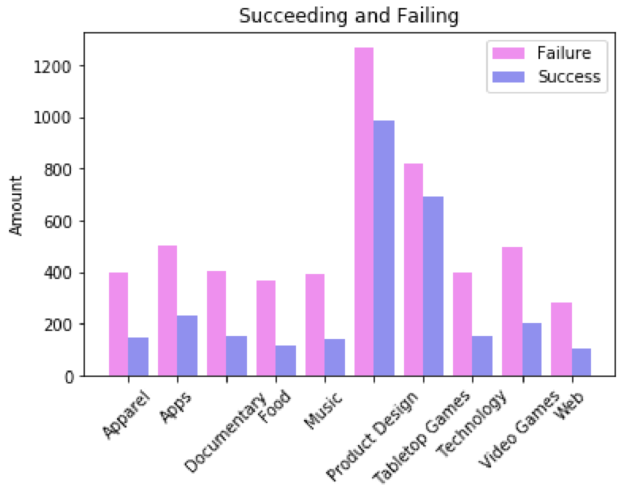


Fig. 4 Failure and Success of top 10 categories.

IV. Related works

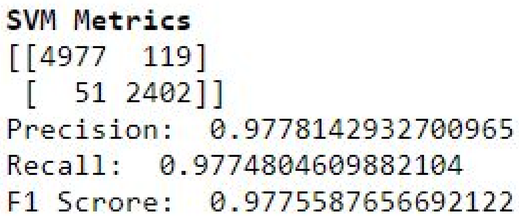
We have compared our work to Kamath and Kamat's work on supervised learning models on Kickstarter. The purpose of their project was to develop a system to classify projects. Their study shows that project properties played a vital role in predicting success as mentioned in [1]. Their method was to use R to preprocess the data and use classifier models on a randomly selected dataset. After all the models were finished running, they checked the performance of each evaluated classifier and selected the best one.

Our approach to the preprocessing and classifiers are different. In their preprocessing, they created a percentage of funding variable while we did not. They also assigned the Kickstarter projects' categories a unique key. Their goal is to predict which category the project belonged to. Our goal was to use the given data to predict which projects would succeed. We also used different types of classifiers.

We believe that our approach is better because it was more relevant than predicting which project belongs to which category. In the related project paper, they did not mention anything about the pruning of the dataset. In our dataset, we vigorously pruned away the irrelevant data so that it did not negatively affect our classifiers.

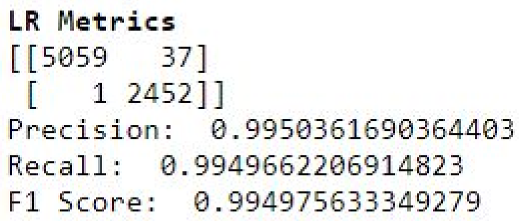
While working on this project we also attempted implementing patterns that usually lead to success to some projects. There are many factors that lead to a project succeeding or failing. Mollick was able to find some correlations to successes and failures [2]. He found a correlation where projects with higher funding goals and longer project durations usually lead to higher rates of failure. We attempted to implement this pattern but adding features to support this caused our models to overfit our predictions. In the end, we decided to omit these extra features from our final attempt.

V. Conclusion



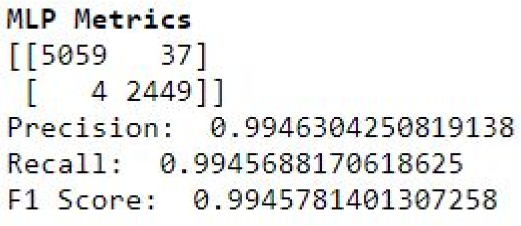
```
SVM Metrics
[[4977  119]
 [  51 2402]]
Precision:  0.9778142932700965
Recall:  0.9774804609882104
F1 Scrore:  0.9775587656692122
```

Fig. 5 SVM Metrics



```
LR Metrics
[[5059   37]
 [   1 2452]]
Precision:  0.9950361690364403
Recall:  0.9949662206914823
F1 Score:  0.994975633349279
```

Fig. 6 LR Metrics

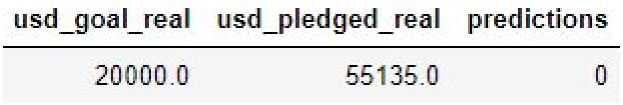


```
MLP Metrics
[[5059   37]
 [   4 2449]]
Precision:  0.9946304250819138
Recall:  0.9945688170618625
F1 Score:  0.9945781401307258
```

Fig. 7 MLP Metrics

In this project, we developed three naive prediction models for live Kickstarter projects.

Our results show that our LR model, Figure 8, was able to more accurately generalize a project's outcome when compared to SVM, Figure 7, and MLP, Figure 9. Based on the confusion matrices of LR and MLP, LR and MLP were able to predict the same amount of successes, but LR was able to predict a few more failures than MLP. This was reflected in the F1 scores, where LR is slightly larger than MLP.



usd_goal_real	usd_pledged_real	predictions
20000.0	55135.0	0

Fig. 8 Failed Prediction

After the training, we had our models predict the outcomes of live projects. Although our models were able to predict with a 95+% accuracy, the predictions on the live dataset were very inaccurate. Looking over the predictions, we noticed that there are some projects that were fully funded and surpassed their goal, but were predicted to fail. We believe that since we explicitly trained our models with certain features and not patterns for success, it would generalize incorrectly. For example, we were unsure on how to implement a pattern such as "If goal < pledged then prediction = 1 (Success)". We also believe that these bad generalizations are also due to the fact that we did not train our models with more data. When the determining the success of a project, it most certainly cannot be done with just a category and how much money has to be raised until the deadline. There were many factors that we either could not account for or publicly gather.

VI. Work Division

Roger and Kandy worked on the pruning of data together. Roger handled the one hot encoding of categories, getting the differences of the dates, and also converting the ground truths from text to numeric representation. Kandy handled normalizing the data, parameter tuning the algorithms and graphing the ROC curve. Roger created a portion at the end that was used to naively predict whether the live projects would fail or succeed.

VII. Learning Experience

What we learned through this project is that there are a lot Kickstarter projects that ended up failing rather than succeeding. This shows that having a model that could help you predict a successful project could potentially be helpful. We also learned that certain categories are more popular than others and that those categories account for a majority of the projects. The top three categories were: design, games, and technology which also happen to be the most frequent project categories that are funded on kickstarter.

References

[1] R. S. Kamath and R. K. Kamat, "SUPERVISED LEARNING MODEL FOR KICKSTARTER CAMPAIGNS WITH R MINING," *IEEE Electron Device Lett.*, vol. 4 No. 1, pp. 17–28, Feb. 2016.

[2] Mollick, E. (2014). The Dynamics of Crowdfunding: An Exploratory Study. Journal of Business Venturing, 29, 1-16.

[3] (2018) The Kaggle Website: Kickstarter Dataset. [Online]. Available: https://www.kaggle.com/kemical/kickstarter-projects