Solving Kickstarter

Predicting Campaign Success

SENG 474 Project
Shreyas Devalapurkar - V00827994
Alexander Pollard - V00821661

1. Introduction

This section will outline the background information for our project, the dataset used, as well as our problem statement/goal we wished to solve through this project.

1.1. Background

Crowdfunding, a method of financing an idea through a large number of small donations, has gained undeniable traction in recent years [1]. Kickstarter functions in a fairly typical manner for a crowdfunding platform. Campaigns start with the presentation of an idea, a goal (a monetary value that the creator deems sufficient for funding their project), and a funding deadline, with the expectation that the public will express their interest and donate to projects they would like to see realized. Kickstarter addresses these donors as backers. If a project meets its goal before the set deadline, it is deemed a success, and it is expected that the campaign creator follows through with their promises [2]. Kickstarter is among the most popular of these crowdfunding platforms, with over fifty-two million pledges having been processed by the platform since its initiation.

1.2. The Dataset

The data we had to work with was a list of over 350,000 recent Kickstarter campaigns obtained from Kaggle but collected from the Kickstarter platform, with their features available in a tabular format. The features of these campaigns included, but were not limited to, title, country, main category, success state, monetary goal, amount of money pledged, and the number of backers.

1.3. Problem Statement and Objective

The main problem that made our project interesting was that although our dataset contained many features, not all of these features would be available for analysis purposes upon a campaign's inception. Features such as the number of backers or the amount of money pledged would only be known when the success state of a campaign had been determined. Taking this into account, we decided to only examine features that would be available at campaign inception and solve the problem of understanding what makes a Kickstarter campaign a success or a failure.

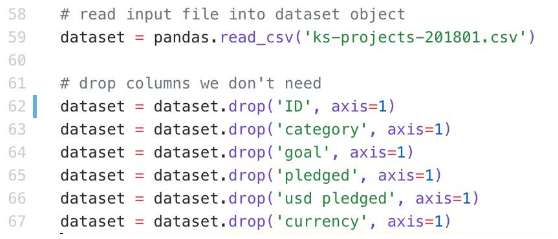
Out of all the attributes available to us upon campaign inception, we decided to focus most on the campaign title. Our hypothesis was that the campaign title would be an extremely important feature because anytime someone views an advertising campaign or petition, the first thing that they look at is its title or slogan. As once said by Paul Hoffman (CTO, Space-Time Insight), "if you want to understand people, especially your customers, then you have to be able to possess a strong capability to analyze text [3]." As a result, we believed that the way a title is phrased could have a significant impact on campaign success. Our goal was to use the campaign title along with other features to predict with relative accuracy, the success or failure of a kickstarter campaign.

2. Implementation and Approach

This section outlines our implementation and approach in terms of how we tailored our dataset to suit our needs, and the machine learning models we used build our kickstarter campaign success predictor.

2.1. Tailoring the Dataset

As mentioned above, our dataset had over 350,000 data points with many features, but we decided to only work with ones we knew would be available upon campaign inception; with more prominence on the campaign title. As a result, we had to remove all features of our dataset that we were not interested in. In order to do so, we read in our dataset into dataframe and dropped certain columns as shown in Figure 1.0.



```
# read input file into dataset object
dataset = pandas.read_csv('ks-projects-201801.csv')

# drop columns we don't need
dataset = dataset.drop('ID', axis=1)
dataset = dataset.drop('category', axis=1)
dataset = dataset.drop('goal', axis=1)
dataset = dataset.drop('pledged', axis=1)
dataset = dataset.drop('usd pledged', axis=1)
dataset = dataset.drop('currency', axis=1)
```

Figure 1.0 - Reading in dataset into dataframe and removing unwanted columns

Along with this, we also had to deal with textual components (TCs). These were columns in our dataset that contained non-numerical values such as the title, main category, and country of the kickstarter campaign. For the country and main category, we used pandas dummy variables to map them to numerical values. This was an effective approach because those columns contained data that was unordered, nominal, and categorical. We represented these columns using a vector where a 1 represented a true value for the respective country/main category and a 0 represented a false value.

As for the title of the campaign, we decided to take on a slightly different approach. We were interested in analyzing two different aspects of a campaign's title; the positivity score and the reading ease score. In order to calculate the positivity score, we used an approach popular in the field of Natural Language Processing known as Sentiment Analysis. The aim of Sentiment Analysis is to gauge the attitude, sentiments, evaluations, attitudes and emotions of a speaker/writer based on the computational treatment of subjectivity in a text [3]. The algorithm used was the VADER (Valence Aware Dictionary and Sentiment Reasoner), which returned a value between the range of -1 and +1, indicating a very negative or positive phrase respectively. For the reading ease score, a common formula was applied to calculate the Flesch-Kinaid score [4] of the phrase, as shown in Figure 2.0.

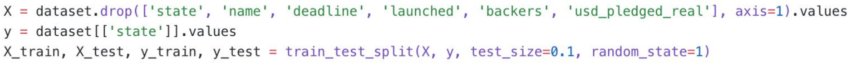
$$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right)+11.8\left(\frac{\text{total syllables}}{\text{total words}}\right)-15.59$$

Figure 2.0 - Flesch-Kinaid grade level calculation formula

This formula returned a numerical value corresponding to the approximate U.S grade point level that the phrase was estimated to be at. With all our additional columns removed and textual components converted to numerical ones, we were ready to train our models for classification.

2.2. Random Forest and Support Vector Machine Classifiers

The two classification models we used were the Random Forest classifier in order to calculate the relative feature importance between the available features of our tailored dataset, as well as a Linear Support Vector classifier in order to predict campaign success and measure prediction accuracy. In order to effectively test the results of our models, we split our dataset into a training and testing set, where the training set was 90% of the dataset size and the test set was the remaining 10%. This splitting of our dataset is shown in Figure 3.0 below.

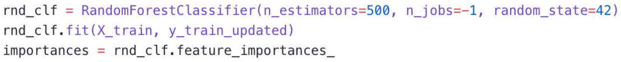


```
X = dataset.drop(['state', 'name', 'deadline', 'launched', 'backers', 'usd_pledged_real'], axis=1).values
y = dataset[['state']].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=1)
```

Figure 3.0 - Splitting dataset into training and testing sets

2.2.1. Measuring Feature Importance

Once we eliminated features from our dataset that we deemed as extremely useful but unavailable upon campaign inception, we were interested in ranking the remaining features in terms of how important they would be for predicting campaign success. This analysis would allow us to fit and train our model with the important features and further eliminate features that seemed to be unimportant or potentially negatively affecting the model. The code used to generate feature importances for the various features of our dataset is shown in Figure 4.0.



```
rnd_clf = RandomForestClassifier(n_estimators=500, n_jobs=-1, random_state=42)
rnd_clf.fit(X_train, y_train_updated)
importances = rnd_clf.feature_importances_
```

Figure 4.0 - Generating feature importance using Random Forest classifier [5]

Plotting and analyzing the relative feature importances of our features revealed that the most important feature out of the ones available upon campaign inception was the goal amount set for the campaign. This made sense because lower target goal amounts are easier to reach than larger ones. The interesting aspect was that the second and third most important features were the Flesch-Kinaid reading ease score followed by the positivity score of the campaign title as shown in Figure 5.0.

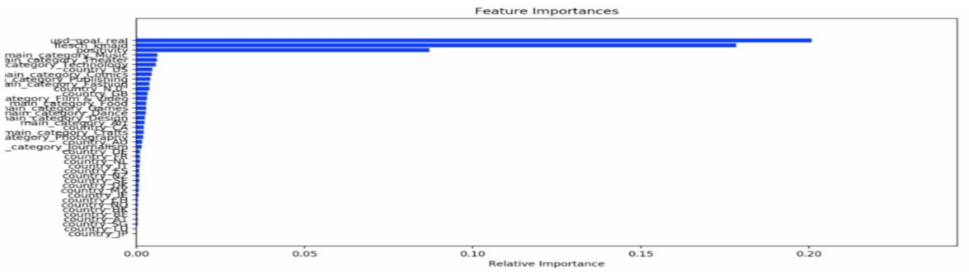
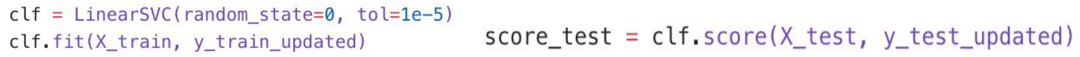


Figure 5.0 - Relative feature importance values [6]

This result made it evident that both the reading ease and the positivity score of the campaign title were essential features for a model to accurately predict campaign success. The next question we set out to answer was with what level of accuracy a model could predict campaign success.

2.2.2. Prediction Accuracy

In order to predict the success state of a campaign and classify a set of features, we decided to use a support vector machine model; the LinearSVC classifier. This was because this model provided flexibility in its choice of penalties and loss functions while supporting a large number of inputs [7]. We fit our model according to our training dataset, and then calculated the mean accuracy on the given test data and labels as shown in Figure 6.0.



```
clf = LinearSVC(random_state=0, tol=1e-5)
clf.fit(X_train, y_train_updated)
```

```
score_test = clf.score(X_test, y_test_updated)
```

Figure 6.0 - Fitting a LinearSVC model and calculating the score (mean accuracy)

We found that our model was able to predict with 66% accuracy the success or failure of a campaign when provided the goal amount, the title reading ease score, and the title positivity score. This is 16% better than a randomized model that will predict success or failure using a 50% coin toss approach. We also noticed that the prediction accuracy dropped to 36% when the reading ease and positivity scores were removed, which shows that those features are critical in helping a model learn how to effectively classify campaigns.

2.3. Creating the Ideal Kickstarter Campaign Title

Now that we knew the importance of the reading ease and positivity scores in being able to accurately predict campaign success, we wished to analyze how positive and simple-to-read titles perform as compared to negative and difficult-to-read ones in terms of their success rates. In order to analyze this, we classified the positivity and reading ease scores for all our campaigns into three sets: very negative/easy-to-read (-1), neutral (0), very positive/difficult-to-read (1) and plotted our results as shown in Figures 7.0 and 8.0.

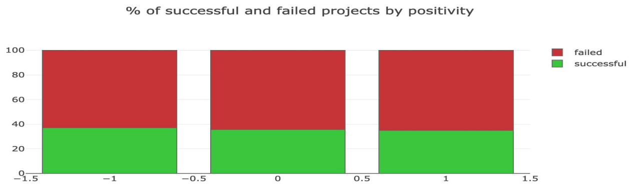


Figure 7.0 - Project success by positivity

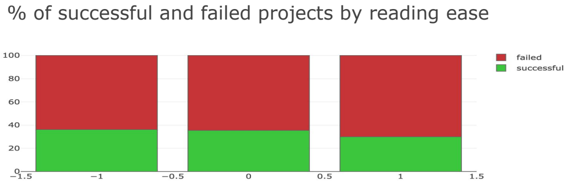


Figure 8.0 - Project success by reading ease

As seen in Figures 7.0 and 8.0, we determined that creating a very positive campaign title does not impact your rate of success as all three categories had approximately a 33% rate of success. Making sure your title is simple to read; however, does play a minor role and was shown to have a 8% increase in success rate as compared to campaigns with difficult to understand titles.

3. Conclusion

Our initial hypothesis was proven to be partially correct, while also being slightly inaccurate. It has been shown that the title of a Kickstarter campaign is a very important feature when it comes to being able to effectively predict whether or not a campaign will be successful. Using a Linear Support Vector classification approach, the success state of a campaign can be predicted with 66% accuracy when the positivity and reading ease scores are included in the feature set, and only with an accuracy of 36% when they are not. Interestingly, although these features are instrumental in allowing the model to accurately predict campaign success, neither the positivity or reading ease of a kickstarter campaign title are very influential in creating a successful campaign; although it might benefit a campaign starter to create a simple-to-read title.

4. References

[1] "Kickstarter Focuses Its Mission on Altruism Over Profit", *Nytimes.com*, 2019. [Online]. Available: https://www.nytimes.com/2015/09/21/technology/kickstarters-altruistic-vision-profits-as-the-means-not-the-mission.html. [Accessed: 01. Apr, 2019]

[2] L. Gannes, "Kickstarter: We Don't Have Anything Against Celebrity Projects (cc: Zach Braff)", *AllThingsD*, 2019. [Online]. Available: http://allthingsd.com/20130509/kickstarter-we-dont-have-anything-against-celebrity-projects-cc-zach-braff/. [Accessed: 01. Apr, 2019]

[3] P. Pandey, "Simplifying Sentiment Analysis using VADER in Python (on Social Media Text)", *Analytics Vidhya*, 2018. [Online]. Available: https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f. [Accessed: 22. Mar, 2019]

[4] J. Kincaid, R. Fishburne, R. Rogers, B. Chissom, "Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel", *Research Branch Report 8-75, Millington, TN: Naval Technical Training, U. S. Naval Air Station, Memphis, TN*, Feb. 1975. [Accessed: 23. Mar, 2019]

[5] "Random Forest Classifier", *SciKIT Learn.* [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed: 22. Mar, 2019]

[6] "Random Forest Feature Importance Chart using Python", *Stack Overflow,* Jul. 15, 2018. [Online]. Available: https://stackoverflow.com/questions/44101458/random-forest-feature-importance-chart-using-python. [Accessed: 22. Mar, 2019]

[7] "LinearSVC", *SciKIT Learn.* [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html. [Accessed: 22. Mar, 2019]