Kickstarter Projects Data Report

The Problem

Kickstarter is a crowdfunding website with a stated purpose of helping to “bring creative projects to life.” According to Kickstarter’s own website, funding tends to be “all-or-nothing.” Projects that receive substantial funding (greater than 20%) are more likely than not to hit their goals, except most projects never hit that 20% threshold. Exposure tends to snowball support, so how can projects get that initial exposure and build momentum?

Kickstarter is a good method of obtaining funding for some projects but is absolutely dreadful for others. Can we help those who are considering launching a project on Kickstarter decide whether to make the leap? Is there a way to know if a Kickstarter project will be successful before it is launched? What can be done to maximize the chance of success?

Data Wrangling

The data was obtained from Kaggle after being scraped from Kickstarter.com.

Serious Projects Only

I looked at the distribution of funding goals and determined that there were a large number of Kickstarters with targets under 1,000 USD. I decided to remove these data as they seem to be less serious campaigns. These data had a much higher probability of being successful (even when compared to the 1,000 to 2,000 USD range).

Duplicate Data

I discovered that there were duplicate project IDs. The duplicates had almost identical information except for the amount of funding received (which could impact the binary state column), so information was taken from different points in time. I was not expecting duplicates, and if they were systematic, they could have a large impact on modeling. I decided to remove the duplicates keeping the one with the larger funding received, indicating that it was newer information.

Projects Need a Chance to Succeed

I decided to remove any projects where the deadline came after the last project submitted (approximately when the data was captured). These projects would not have had sufficient time to reach their goal and should not be included in the model.

Features and Feature Engineering

I created a new column calculating the number of days between the goal date and the launch date. Thirty days was by far the most popular choice for both successes and failures.

I then created a new numerical column to count the length of the name. I counted the number of non-blank characters for both the blurb and the name. I then took these new columns and

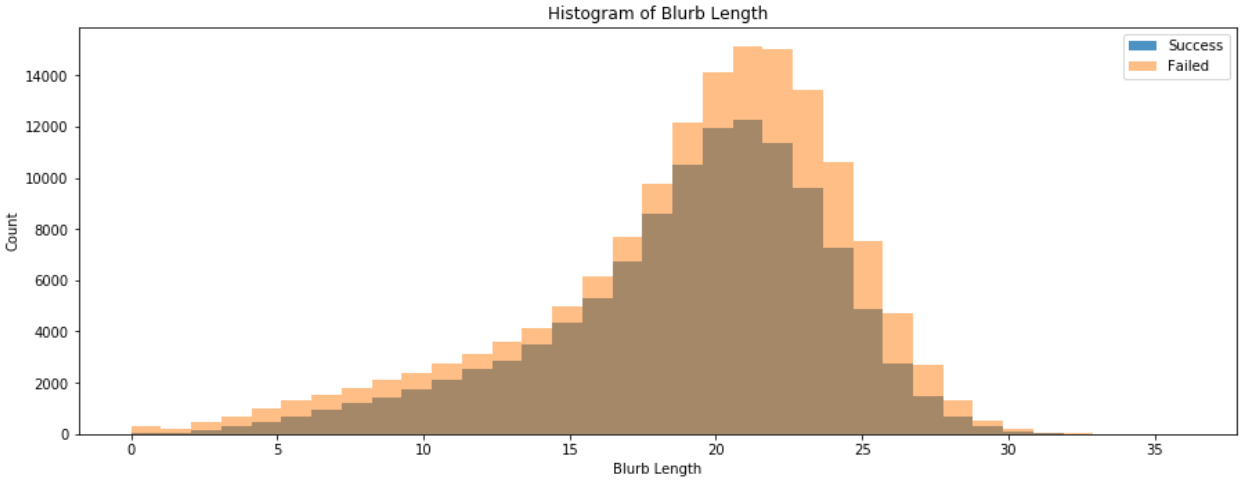
divided them by the number of words in the blurb and name to create blurb characters per word and name characters per word features.

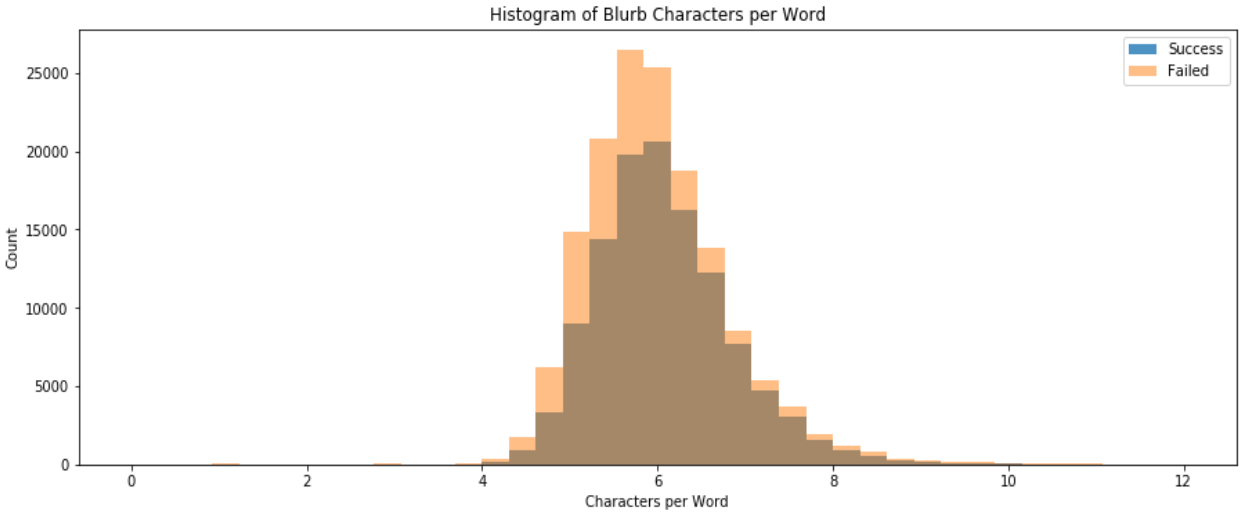
I examined both the name and the blurb text for various keywords to find out if certain words were associated with more or less failure.

The Story

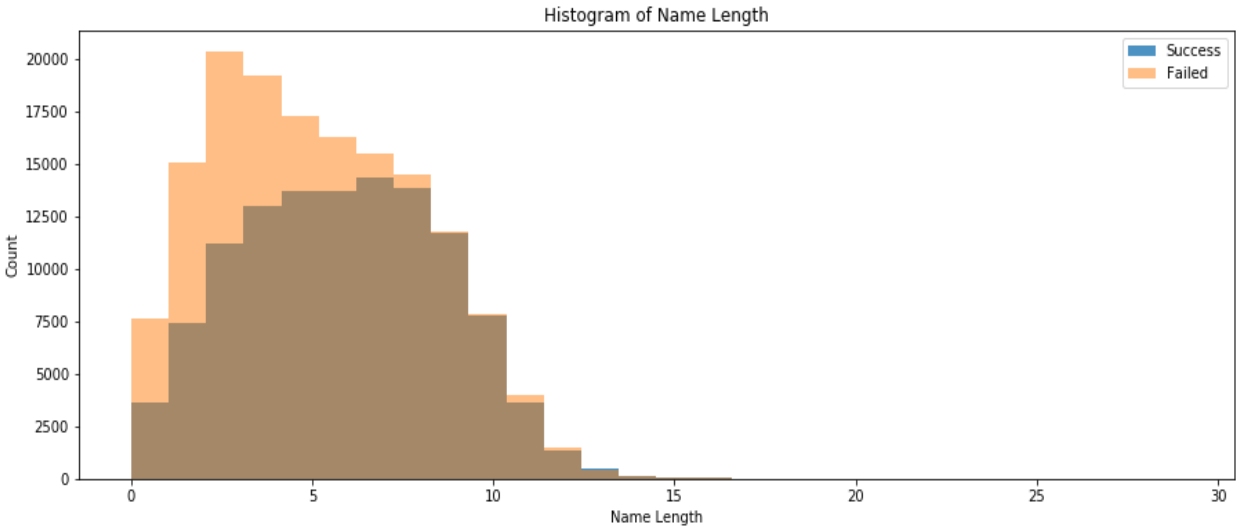
Length of Text Columns and the Number of Characters per Word

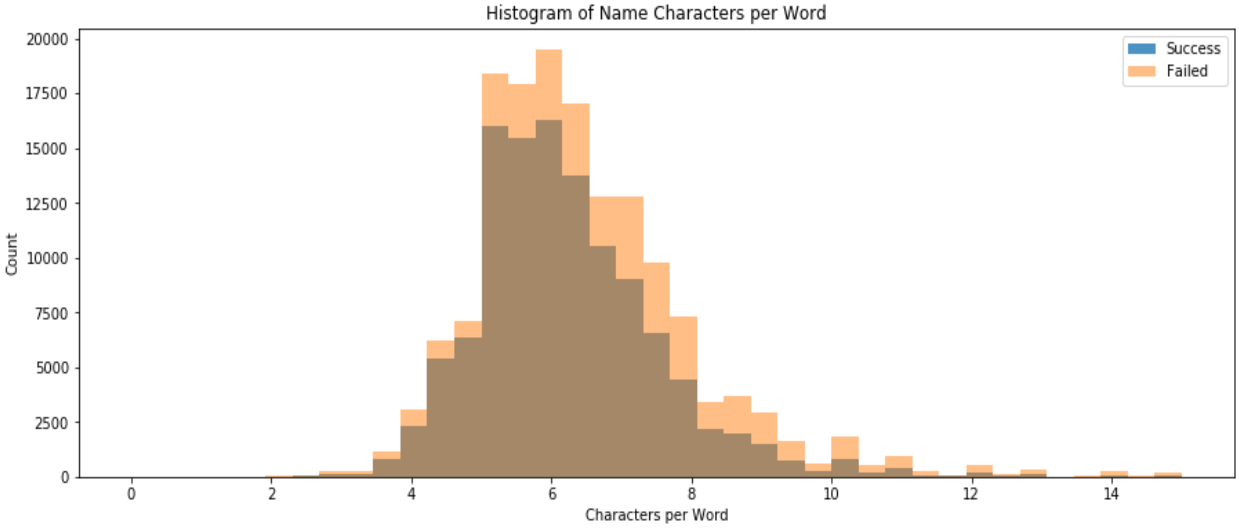
The blurb length histograms look very similar; however, the blurb length for failures appears to be slightly more skewed to the right. Blurb characters per word histograms have some obvious small differences with failures being a little more skewed to the left, so successful projects use slightly longer words.



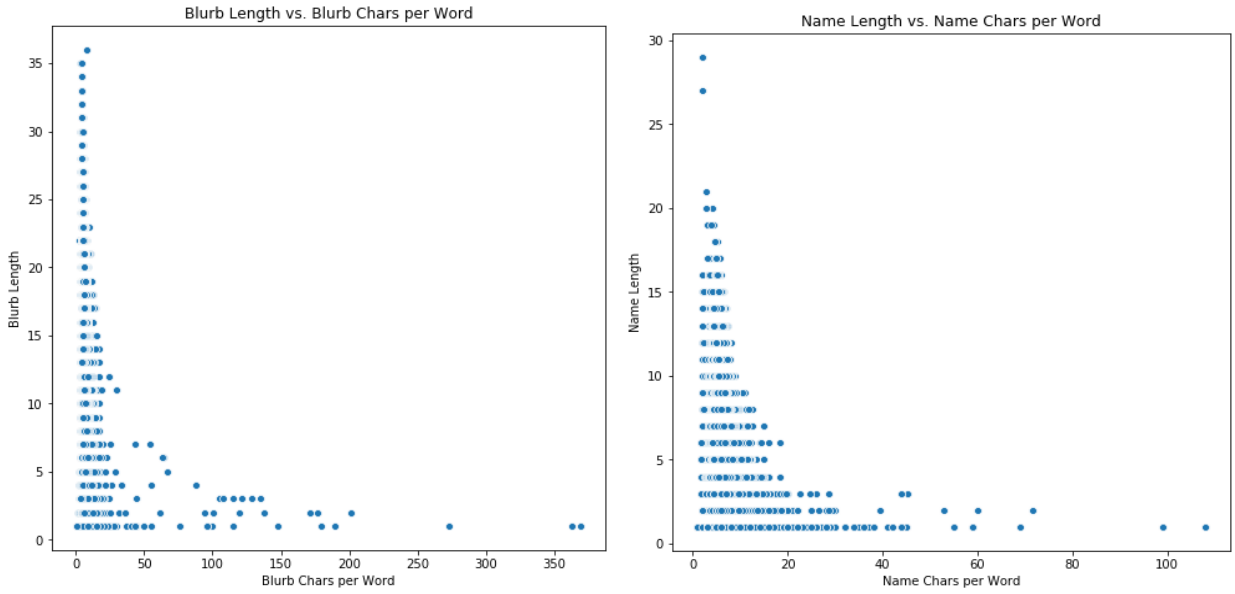


The first distribution I noted to be significantly different was the name length with failures tending towards shorter names. Opposite of blurb characters per word, longer words in the name tended to be associated with slightly more failure. A guide for new Kickstarters could be to keep the name a bit longer, yet simple and memorable, while spending some effort writing a little more complex blurb.



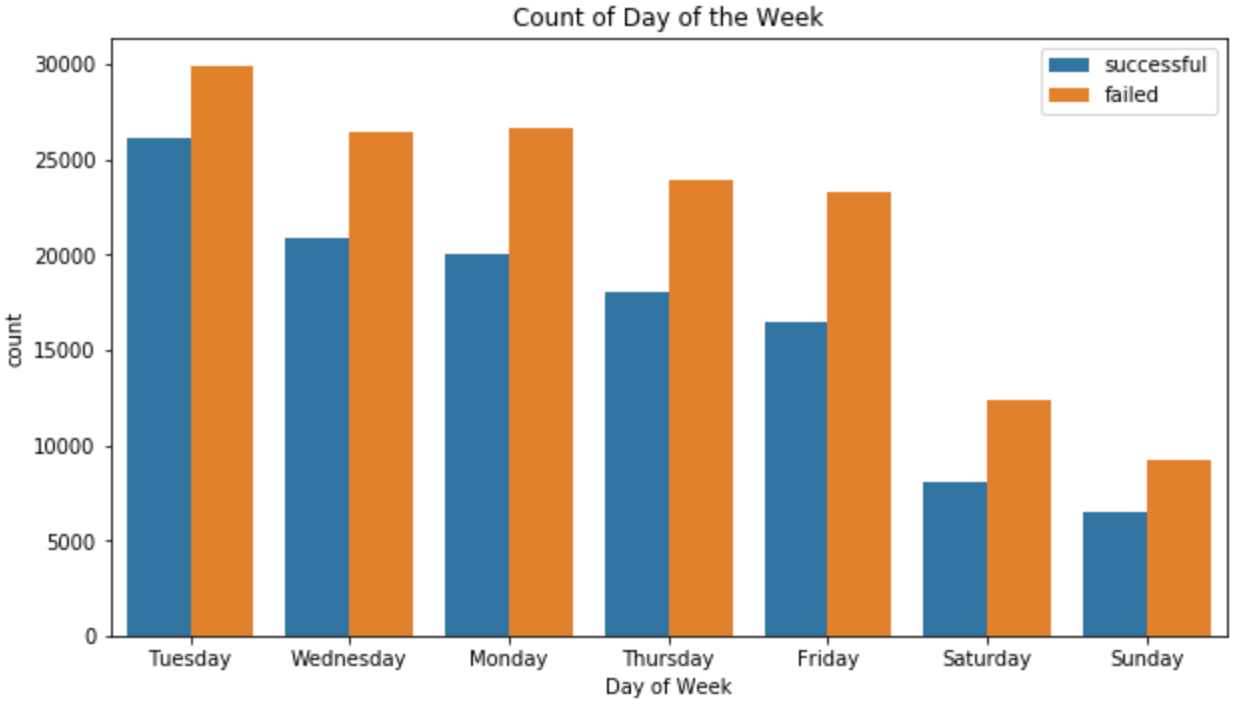


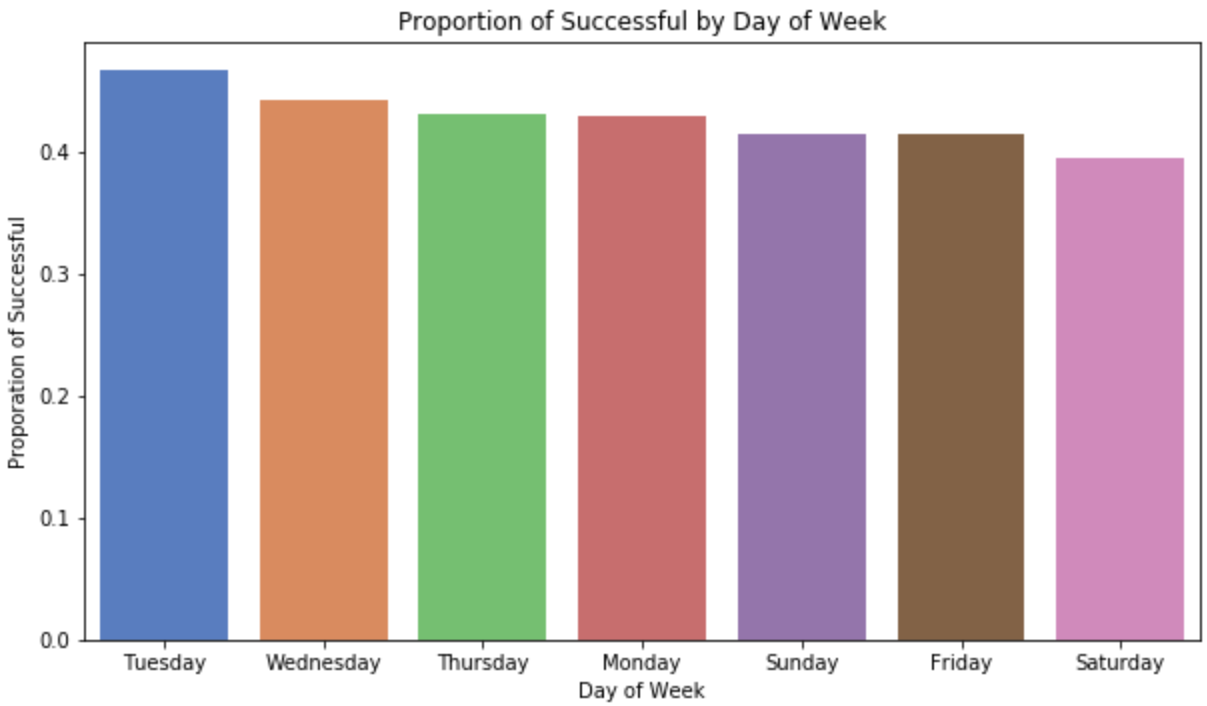
Plotting the length variables against the characters per word variables reveals a definite negative relationship that is not best represented linearly (Pearson's Rs of less than -0.31). These negative relationships are likely related to the fact there are limits on the number of characters for the name and blurb. More words mean your words must be shorter to fit within the limits.



Day of the Week

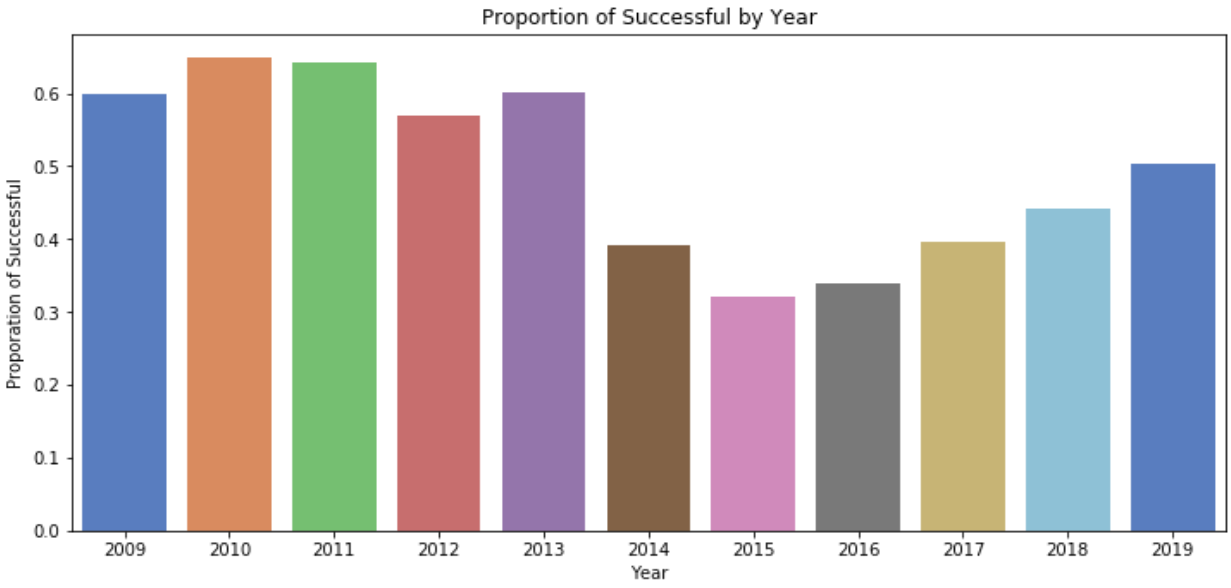
I then examine categorical variables looking for major differences between success and failure in each category. For the launch date day of the week category, there was not major variability in the success versus failure ratio, but Friday, Saturday, and Sunday had lower activity and a slightly lower chance of success. Perhaps, quick support is important for getting the ball rolling and people are less likely to visit Kickstarter on the weekend when they aren't at work.





What Happened in 2014?

The most interesting discovery for exploratory analysis came while examining a variable that is not useful for modeling, the year of launch. There was a huge drop off in success from 2013 versus 2014.

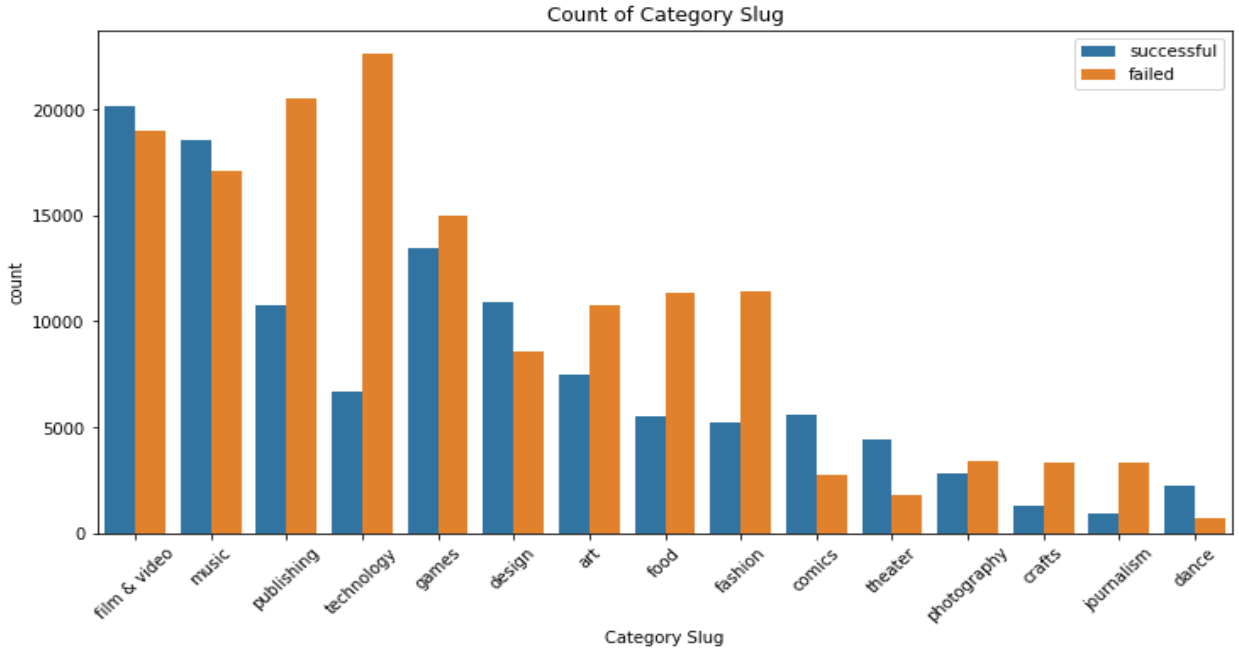


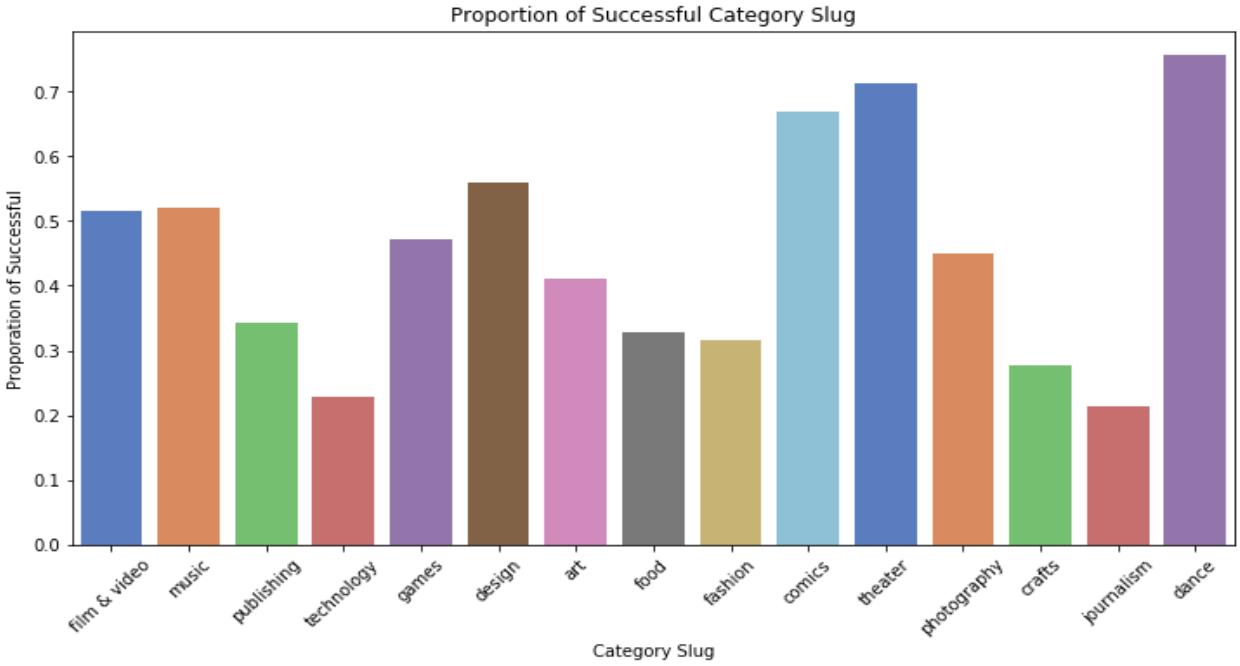
The success rate was 60% in 2013. In 2014, there were almost twice as many projects with almost a doubled average goal. With more ambition came more failure, and projects succeeded

at under 40% in 2014. I have not yet been able to decipher what changed in 2014 through research of Kickstarter.

Project Category Matters

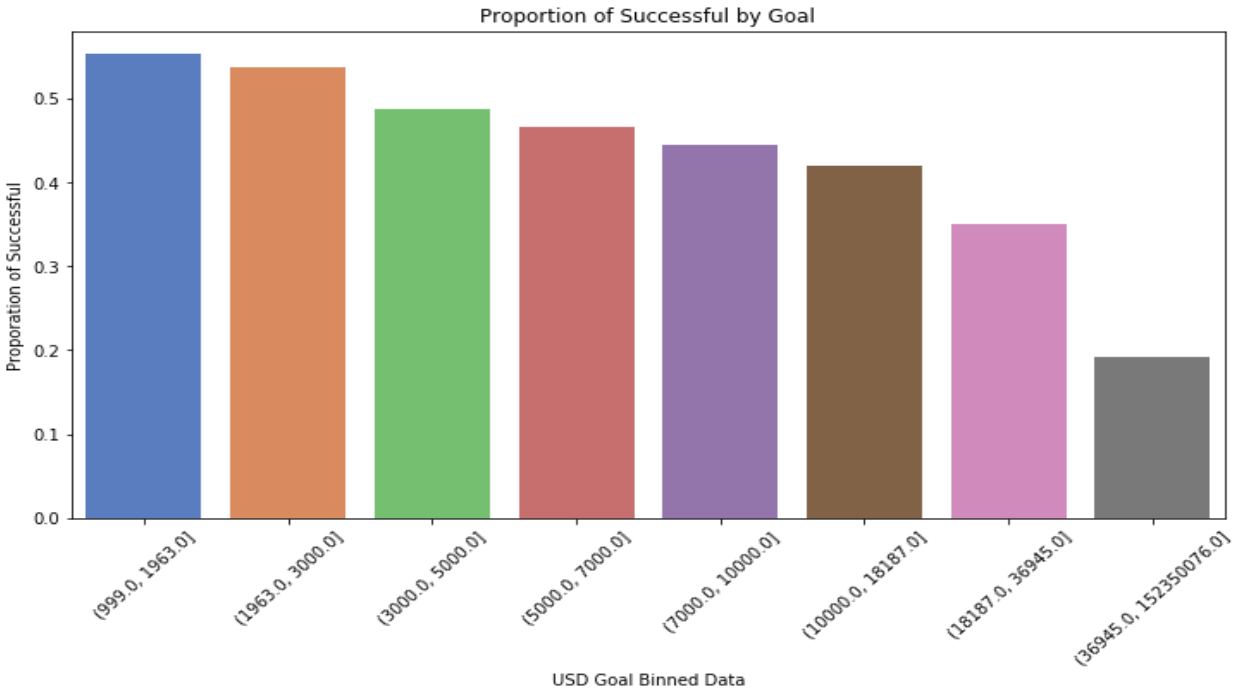
The category feature has high variability between success and failures for the numerous categories which is useful for model prediction. The most interesting categories were publishing and technology as they have a high number of instances and very imbalanced rates of success. Dance, theater, and comics all stood out as having a high probability of success.





Keep Your Eye on the Target (Goal USD)

I visualized the probability of success against the goal USD, the most obviously important feature. The more ambitious the project, the more likely it is to fail. I used the qcut function in pandas to display the categorical result vs. the numerical goal, and the probability of success across the 8 cuts steadily drops from 55% to a 19% chance of success. However, success is far from guaranteed for less ambitious projects.



Inferential Statistics:

I identified two numerical columns with very small variances between the means: the name length and the blurb length of projects. I performed bootstrapping to compare the means of the name length of successful projects and the name length of unsuccessful projects and found there was a statistically significant difference. I had similar results looking at the blurb length.

I used a chi-squared test on two categorical variables that I was unsure whether to keep for modeling purposes. I performed a chi-squared test on the day of the week column (and the result column) and the month column (and the result column). Both results were statistically significant, so I decided to keep the variables for modeling.

Machine Learning:

Data Preparation for Modeling

I encoded the relevant categorical variables (month, day of the week, project category) as numeric. I chose to encode as numeric rather than using one-hot encoding. The light gradient boosting machine's (GBM) method of dealing with categorical variables is one of the reasons I chose it as my primary modeling method.

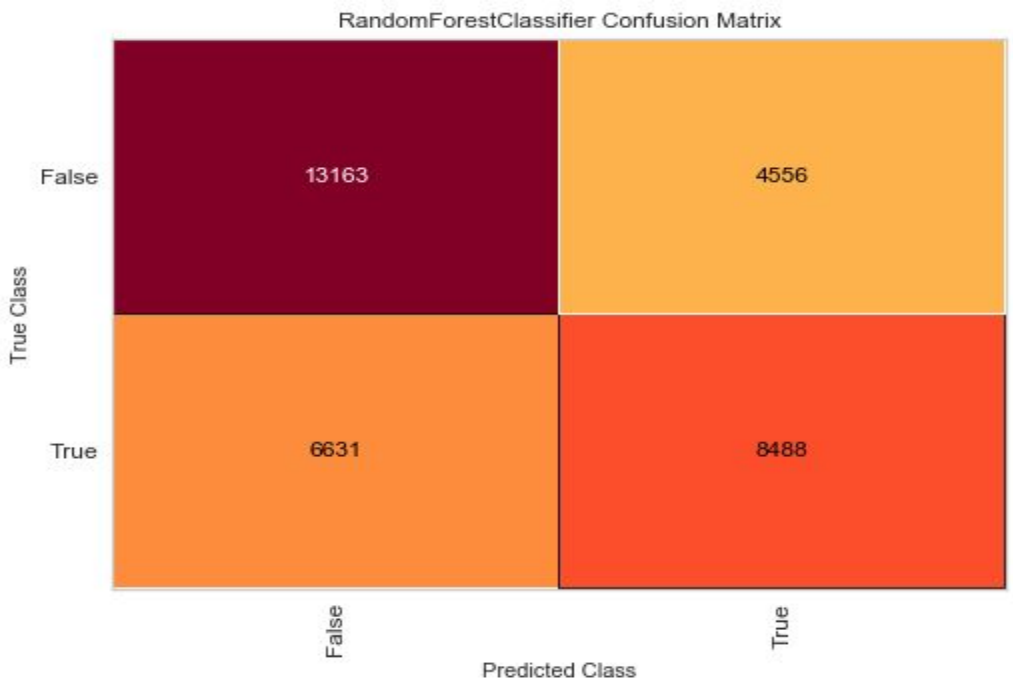
I sliced the data based on the launch date. The data extended through late April 2019, but projects launched in that month would not have hit their deadline, so I sliced data with a launch in March 2018 through April 2019 to capture a whole year for the test data. My assumption is that the most recent data is the closest to new Kickstart Projects, so it serves as the most useful test data. The rest of the data going back to 2009 was used as training data.

Random Forest Modeling

I initially chose to model my data using random forest (RF) as it is a flexible model; I have a lot of features with questionable importance and binary features (keyword features) that do not have many instances of occurrence.

I chose ROC AUC as the primary metric for measuring the performance of my models as it will indicate which features are most important for boosting success. Simply, ROC AUC measures the model's ability to distinguish between classes. The most important features will be those that boost success even when failure ends up being much more likely for the majority of projects.

My baseline RF AUC score on the test data was 0.7111 which I was able to boost to 0.7226 after numerous iterations of random search to tune the hyperparameters. The most important features by far were the goal in USD and the category (numerically encoded).



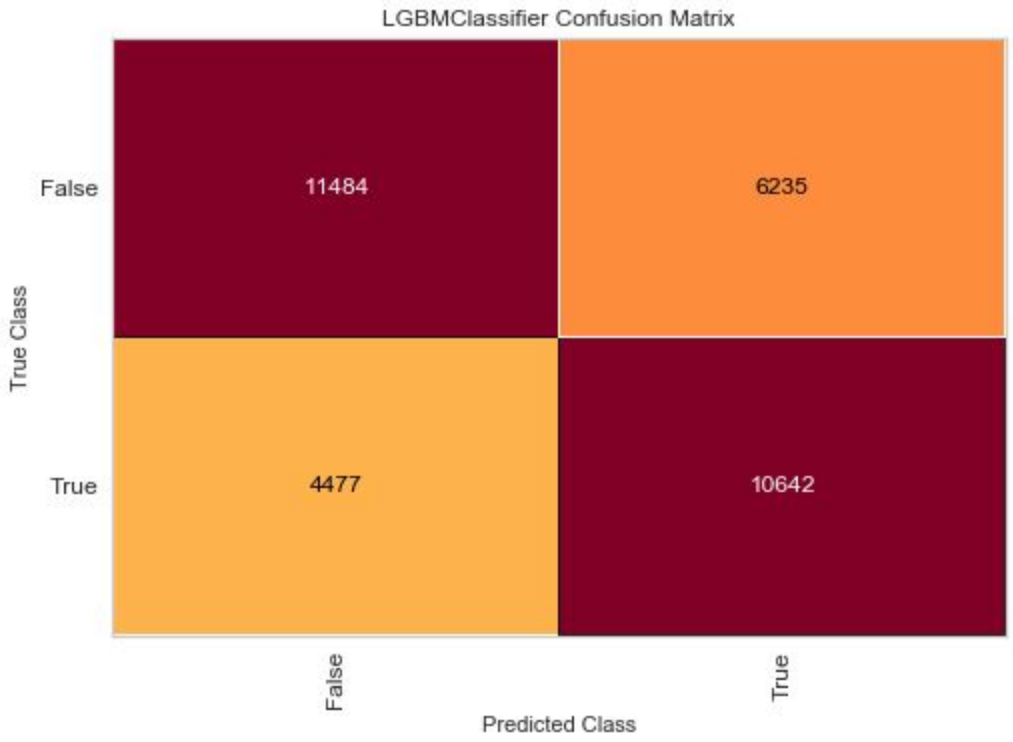
Light GBM

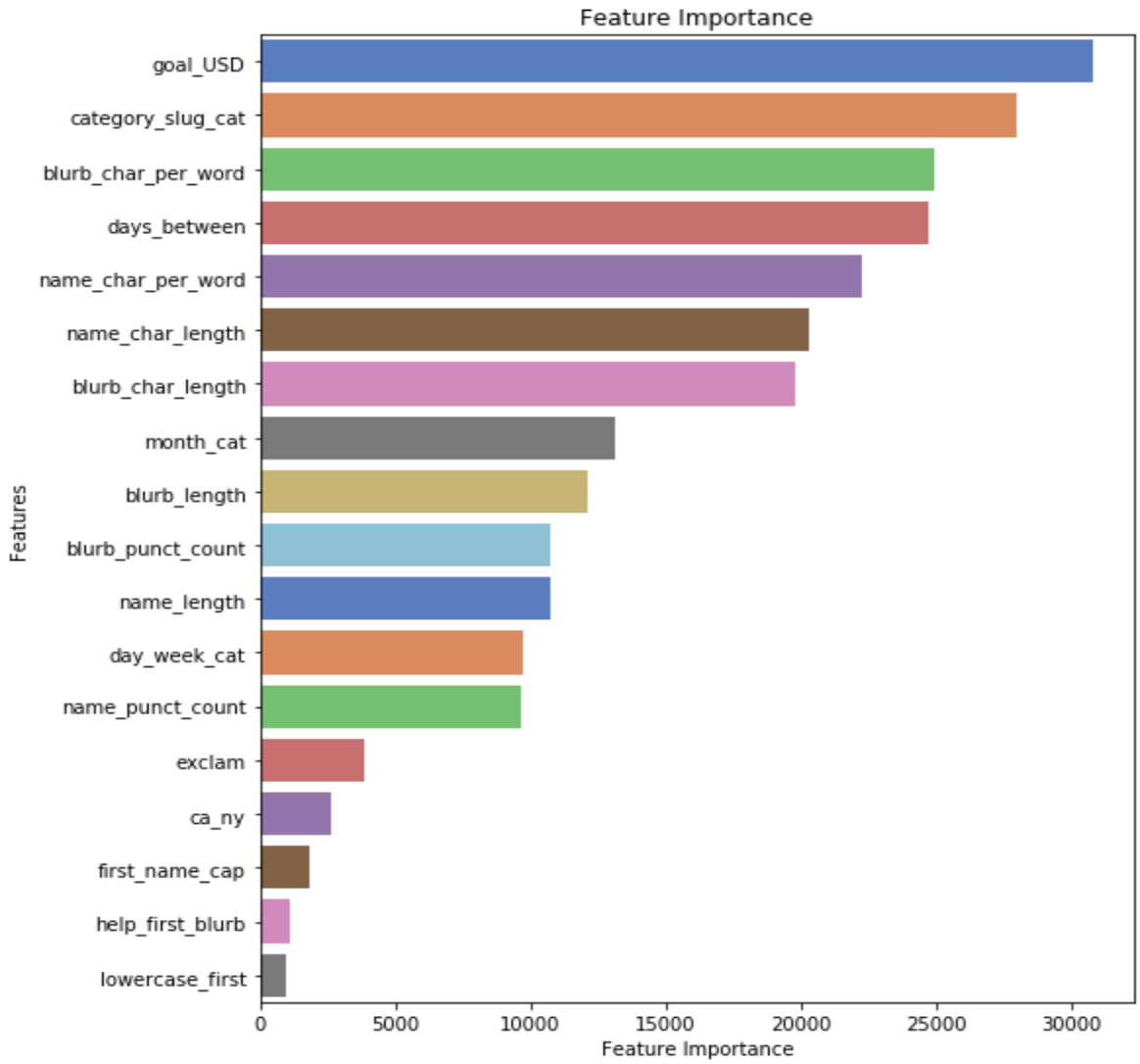
After reading up on models and methods to improve on my random forest model, I stumbled upon LGBM and decided to try it out. I liked its method of dealing with categorical data that could be an improvement from simpler models.

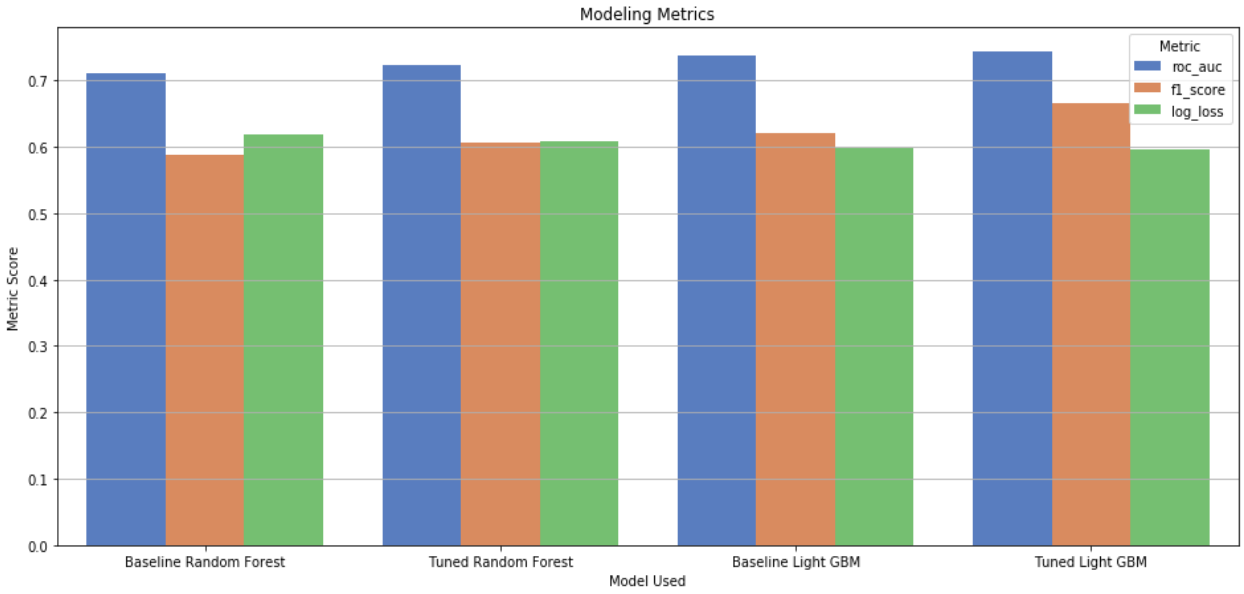
The baseline model produced a validation AUC score of 0.7600 which translated well to the test data scoring 0.7372. The baseline model outperformed the tuned RF model on all metrics. After hyperparameter tuning, I was able to achieve an AUC score of 0.7426. In addition, other metrics improved significantly, especially when compared to RF. Most notably, the hyperparameter “is_unbalanced” (defaults to False) being switched to True improved the F-score over the out of the box model by over 0.04.

The most important features in both models were the goal in USD and the project categories. Interestingly, the LGBM mode put less emphasis on the goal USD and categories when compared to the tuned random forest model.

An interesting comparison between the LGBM and the RF confusion matrices, is the LGBM model is much more likely to predict successes (16,877 total successes predicted against 13,044), while the RF model is much more likely to predict failures (19,794 failures predicted against 15,961).




Feature Importance
goal_USD
category_slug_cat
blurb_char_per_word
days_between
name_char_per_word
name_char_length
blurb_char_length
month_cat
blurb_length
blurb_punct_count
name_length
day_week_cat
name_punct_count
exclam
ca_ny
first_name_cap
help_first_blurb
lowercase_first
Features
0
5000
10000
15000
20000
25000
30000
Feature Importance


Modeling Metrics
Metric
roc_auc
f1_score
log_loss
Metric Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Baseline Random Forest
Tuned Random Forest
Baseline Light GBM
Tuned Light GBM
Model Used

Conclusion

After spending time exploring and modeling the data, I believe I can suitably answer: "What advice would you give to those considering launching a Kickstarter project?"

First of all, the goal amount is always going to matter more than anything. If you want a million dollars, Kickstarter probably isn't the platform for raising money. If you can find a way to start small, you have a better chance of snowballing and scaling up later rather than starting out extremely ambitious.

Second, Kickstarter is conducive to some categories and punishing to others. Notably, it is especially punishing to publishing/journalism and technology projects. If your ambition is in those areas, I would choose a more suitable platform to set yourself up for success.

Finally, attention to detail in the name and the blurb matters more than I would have expected. A longer name with shorter, simpler words is helpful, and a longer, descriptive blurb is helpful for pushing your project over the top and informing people of your purpose. I would also advise following proper capitalization and grammar standards when writing your blurb.

For future research and exploration, I am interested to see what features are most important in fixed goal dollar ranges and to see how that changes as the goal increases.

For another interpretation of the data from the person who did the initial scraping and cleaning, refer to his Github project.