Final Report – Capstone Project 1

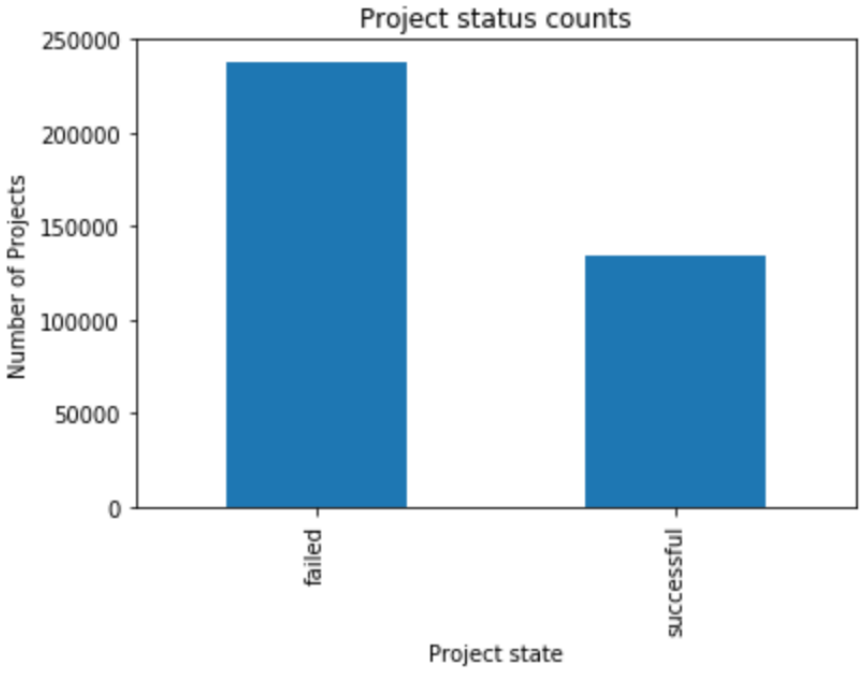
Hacking Kickstarter: Predicting Crowdfunding Project Success

1. The Project

Our communities are full of people with great ideas. However, it often takes money to help bring those ideas to life. That's where Kickstarter comes in! Through the platform, artists, builders, and other creatives can turn to online audiences for support to materialize their ideas.

Since Kickstarter launched in 2009, 473,134 teams and individuals have sought funding for their creative projects. 176,314 of these projects have been successful in meeting their funding goals, a success rate of only 37.49%.[1] When a project does not meet its funding goal, the backers are not charged for the money they pledged. In other words, there is no such thing as partial funding on Kickstarter – a project that fails gets nothing. It's a lose-lose situation: the creator loses an opportunity to make and sell their product, and the backers lose a chance to partake in their investment.[2]

Yet teams seeking funding devote time and effort into promoting and advertising to potential backers to raise the funds they need. How can a team maximize their chances of meeting funding goals? Can we predict what factors lead projects to success or failure, or is this merely a matter of luck?



[1] Statistics are taken from the Kickstarter official stats page.

[2] A rule of Kickstarter is that all projects must culminate in something that can be shared with others.

The above chart shows the number of failed projects compared to the number of successful projects from 2009 to early 2018.

Crowdfunding platforms such as Kickstarter, GoFundMe, IndieGoGo, Causes, Patreon, and LendingClub may benefit from a prediction model to offer advice to funding-seekers using their platforms. This advice can include insights into what kinds of projects to choose, realistic funding goals, and appropriate timelines for fundraising. This is also directly useful to the users of these platforms since it can serve as guidelines to give their projects the best chance at success. Moreover, these insights can be translated into parallel advice for start-ups in the investment phases of their growth, as well as charity and non-profit fundraising campaigns.

2. Data Wrangling

This project utilizes the Kickstarter dataset obtained from Kaggle. This dataset has information for 375,764 Kickstarter projects between 2009 and early 2018, including their titles, descriptions, categories, geographical location, launch dates and deadlines, target funding goal, final total funding, outcomes, and number of contributors. This information can allow us to look for investment patterns across time, region, and project category, as well as success or failure of various kinds of projects, defined as whether or not the funding goal was met.

For this particular dataset, I explored each column individually as univariate data, dividing them into categorical and numeric columns. I also generated boxplots, histograms, and bar plots to create visualizations of basic statistical features, such as the mean, quartiles, and outliers for the numeric columns, as well as the value counts for the categorical columns. I then calculated and plotted success rates of projects based on whether the project was reported to have succeeded or failed. Through this process, anomalous data points such as impossible dates and country names, or unusable values such as projects labeled as “live” revealed themselves. After eliminating these problematic areas from the dataset, I finally saved the data to a new, clean CSV file. The details of the cleaning and exploration steps are as follows:

1. After loading my dataset to a Jupyter Notebook, I used the pandas library to convert it to a data frame. A cursory glance at the first few rows of the data frame did not reveal anything too out of the ordinary.
2. The first wrangling step I took at this point was related to the columns with date data. I ensured that the two columns with this data, the “launched” and “deadline” columns, were in datetime format, and then I created a new column called “duration” by subtracting the launch date from the deadline in order to create an integer value for the number of days each project lasted.
3. I divided my columns into numerical and categorical data. For each numerical column, I generated a boxplot in order to get an idea of the distribution of the data. These plots tended to have many outliers above the upper quartile. Because there were so many outliers, I will leave them as is for the time being.
4. For the categorical columns, I generated a bar plot for each one to show the number of projects falling into each category. In examining the launch date column, the first very

strange data points were identified: seven projects were reported as having launched in 1970. Because Kickstarter has only existed from 2009 onwards, these dates were impossible. Therefore, I removed the rows with 1970 as a launch date from the data frame.

5. Next, with the countries column, I observed that there was a country strangely labeled as N,0". Although this could have been a typo for the country abbreviation for Norway (NO), it was also strange that the N,0" category had more projects than all the Scandinavian countries combined, including the others labeled as NO. Furthermore, projects with this label had their currency set to USD, AUD, GBP, and EUR. Because I could not confirm this to be NO or another country, I removed rows with this label from the data frame.
6. The next wrangling step arrived upon examining the "state" column, which provides an overview of the status of each column as either successful, failed, live, suspended, cancelled, or undefined. Since the larger goal of this project is to generate a model that predicts the outcome of a project, a binary classification of either success of failure is preferable to a six-way classification. To reduce the options here, I removed all rows that were classed as live and undefined, since these projects technically do not have outcomes. Furthermore, I re-classed the suspended and cancelled projects as failed, since these are technically not successful. Successful projects, of course, remained as successful.
7. Next I generated a second set of visualizations for the numerical and categorical columns. For the numerical ones, I again used box plots, now grouped by the "state" column, to show the success rates of the projects according to that category.
8. To show the success rates for the columns with categorical data, I changed "successful" and "failed" in the "state" column to 1 and 0, respectively. I generated a plot to show the success rates of different kinds of projects in each column, followed by a value counts plot with the types of projects listed in the same order. In order to do this, I had to create a list of the values according to the order of their success rates, and apply the list to the value counts plot using .loc so that they displayed in the same order and were thereby easier to compare.
9. Finally, I saved the cleaned data to a new CSV file so that it would be ready to use for more in depth exploratory analysis.

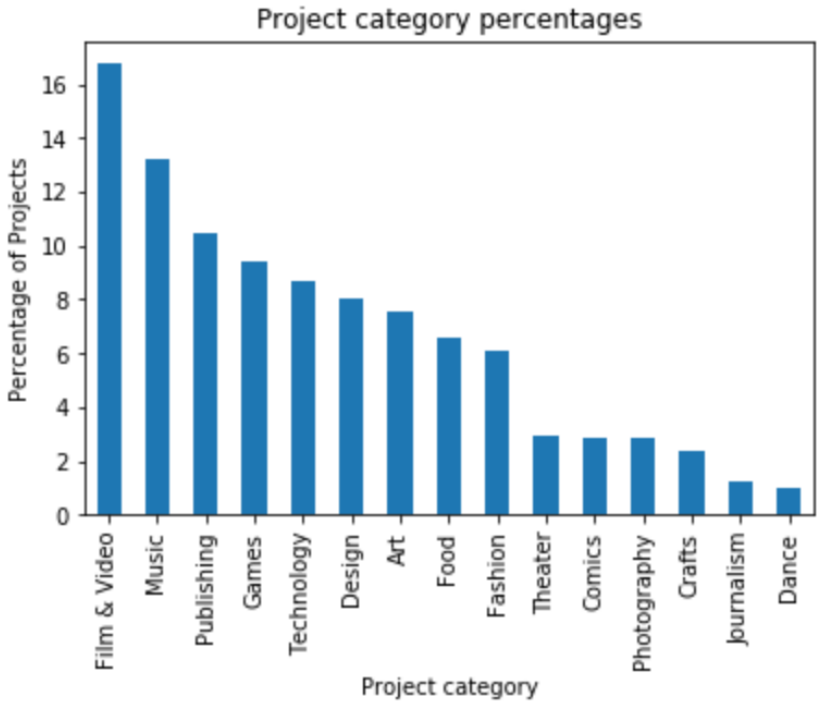
3. Exploratory and Statistical Analysis

To begin exploring the data, I began with some follow-up questions to my overall question of what makes a Kickstarter project successful. Specifically, I asked:

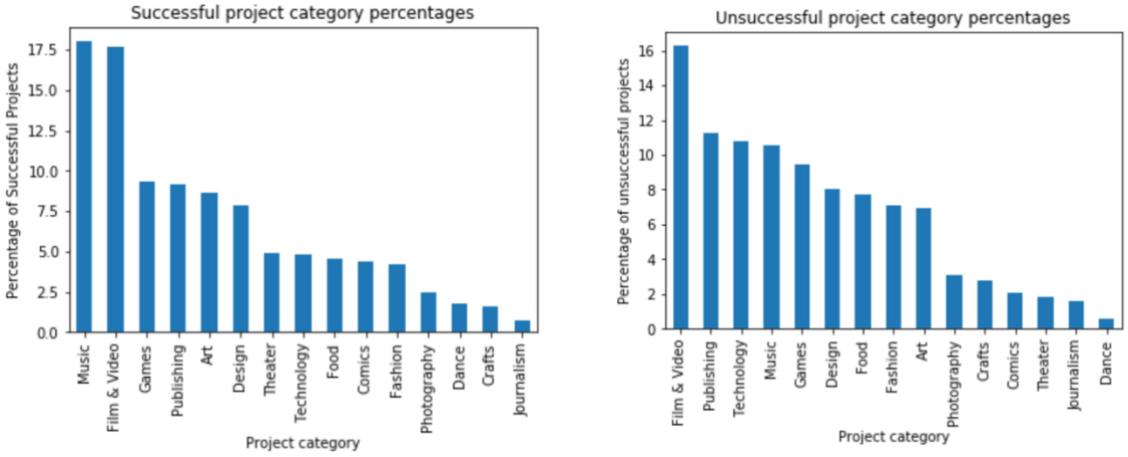
- Were different project types more successful than others?
- Is it harder for more expensive projects to meet their goals?
- Do longer projects give more people more time to contribute?

These questions revealed some interesting findings.

In examining project categories in relation to success, I looked at the distribution of project types overall and the distribution of project categories among successful and failed projects.



Overall distribution of projects by category

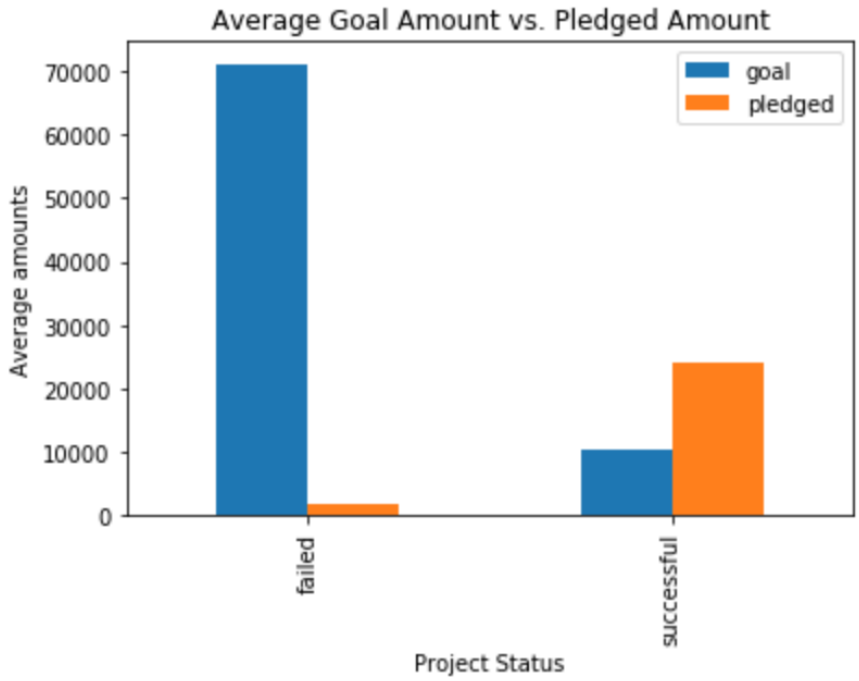


Side-by-side view of distributions of successful and unsuccessful projects

Almost 17% of all Kickstarter projects were in the Film & Video category, making it the most popular of all categories. This distribution was is rather similar among both failed and successful populations, with a few exceptions such as music, which made up a slightly higher

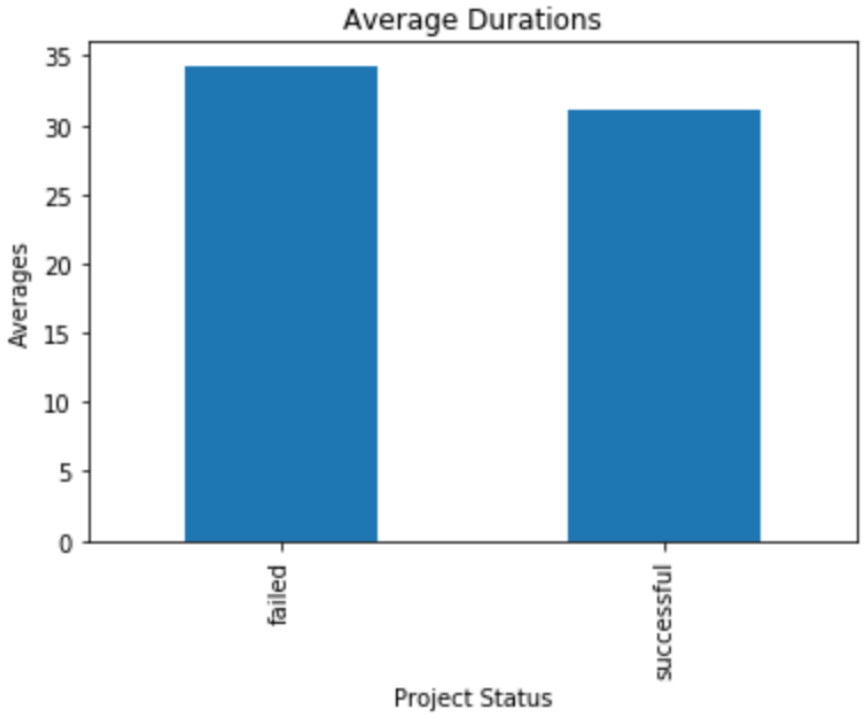
percentage of successful projects, and fashion, which made up a slightly higher percentage of unsuccessful projects.

Next I looked at success as it related to funding and time. First, I looked at the success rates of projects with high funding targets and compared them to the amount of funding the projects actually achieved.



It seems that projects that ask for less money in general were more likely to meet the goal amount, and more people contributed to projects that asked less on average. However, the amount individuals tended to contribute, on average, was still very similar (roughly $91 per backer), with less than a dollar difference between the two means.

As for project durations, I found that successful projects lasted on average three days less than unsuccessful ones, and that successful projects saw roughly eight contributors per day, whereas unsuccessful ones saw on average one backer every two days:



4. Statistical Analysis

After completing a general overview of the data through exploratory analysis, I applied statistical test to each column in the data set systematically. This was to uncover which variables in the data were statistically significant.

I began by dividing my dataset into numerical and categorical data. For the numerical data, I first performed a two-sided t-test for two independent samples using ttest_ind() from the scipy.stats library, which tests the two samples against the null hypothesis that they have identical expected values. I took successful and failed projects as the two samples, and I ran the test for the data in each column (goal and pledged amounts, goal and pledged amounts in USD, backers, and project duration). In all cases, we were able to reject the null hypothesis, indicating that the two samples could not have been achieved randomly.

Next I performed difference of means and difference of standard deviation tests through simulating 1000 additional samples by drawing bootstrap replicates. I also permuted the success and unsuccessful data 1000 times to simulate randomly achieved data under the null hypothesis that any project could be successful or unsuccessful, and so reassigning success and failure should not make a difference in the results. I then performed a difference of means test again and compared it with the observed difference. In each case, the simulated differences from the permutation test were all normally distributed with a mean close to zero, and the observed difference of means, and the differences of means from the bootstrapped replicas were all far outside of the permuted distribution. Again, for each column, the observed data could not have been achieved randomly and so we rejected the null hypothesis.

For the categorical data, I consulted with my mentor and chose to perform Chi-squared

tests on each column (category, subcategory, currency, country, launch year, month and day and deadline year, month, and day). To do this, I turned each column into a contingency table according to the value counts of the successful and unsuccessful projects, using pd.crosstab(), then setting the status (successful or failed) as the index. I then used chi2_contingency() from the scipy.stats library to carry out the test, which compares the observed values in the table with expected, or randomly achieved values. Like the other tests performed, this also assumes a null hypothesis that the observed data will be identical to the expected. For each column of categorical data in my dataset, we were able to reject the null hypothesis, again finding that it was unlikely these results were achieved by random chance.

Based on these results, it seems to be the case that the differences between successful and failed projects are all statistically significant. In other words, there are strong correlations between the independent variables designated by the columns and the dependent variable of project success and project failure. Because of this, we must assume for the time being that all variables will be meaningful and taken into account as I move into the modeling stage of the project.

5. Machine Learning

After completing statistical analysis, we can move on to machine learning. Because this data had both input and output variables, I was able to exclude unsupervised learning techniques, and because the output variable values were just zero and one, I could further narrow down to a binary classification problem.

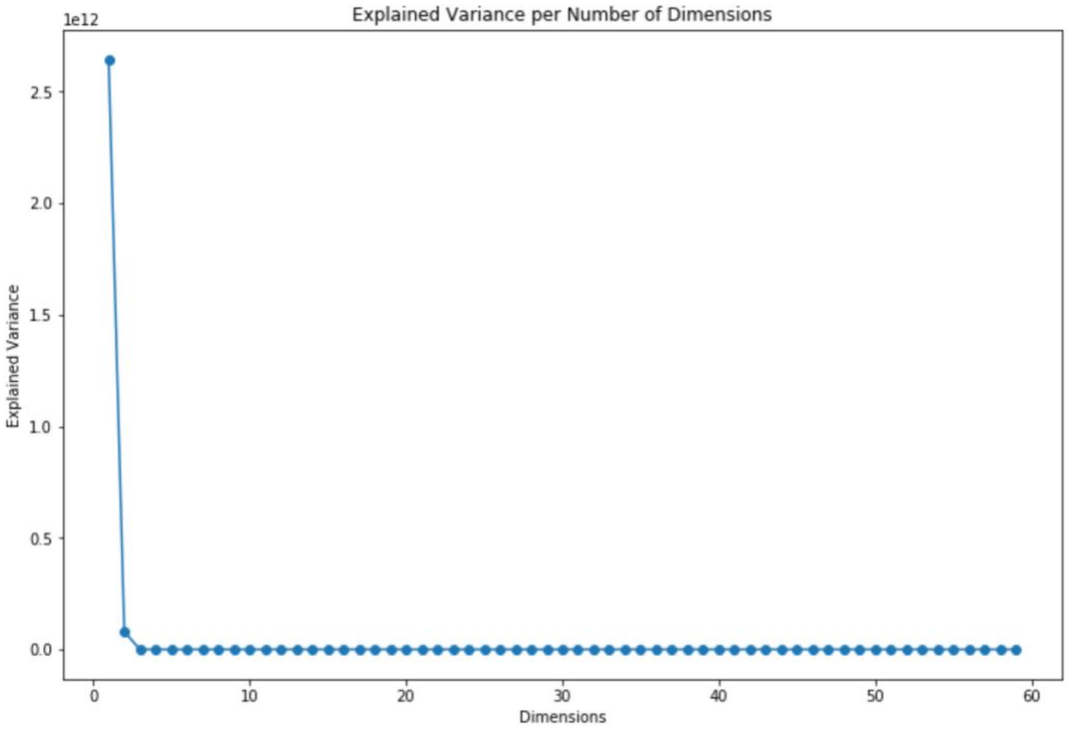
Before testing algorithms, though, the first step I took was to further prepare the data for machine learning, the main issue in this case being the conversion of categorical data to numeric data. I began by dropping some columns that were unnecessary from the data:

- Columns with all unique values (Index, ID, Name)
- Subcategory column, because it correlates with the main category
- USD Pledged column and State column, because data is repeated elsewhere

Next I separated the date columns (launchdate and deadline) into year, month, and day columns, and then dropped the original date columns. I then took the remaining categorical columns (main category, country, currency) and converted them into numeric data with dummy variables and appended this to the rest of the numeric data. Finally I separated the completely numeric dataframe into input and output variables for machine learning. For the input variable, I dropped the pledged amount columns, since knowing this amount would result in a 100% accuracy rate, and I dropped the rate column, since this is the output variable.

The first classification model I tested was logistic regression. Although this turned out to not be the best model for this particular data, the various adjustments to the data needed to make it work were revealing. Running the model with the default parameters gave an accuracy score of 88%, but upon further examining the statsmodels table, it seemed like there was something wrong with either the data itself or with the application of this model to the data. For example, the value for the Pseudo R^2 was “-inf” and many columns had p-values of NaN.

I took several approaches to try to alleviate the problems shown in the statsmodels table. First, I tried cross-validation and tuning of the hyperparameter C (inverse of regularization strength), although the result was unchanged. Then I used principal component analysis (PCA) to reduce the dimensionality of the data. Using the elbow method, I determined two or three to be the optimal number of dimensions, based on the almost-perpendicular graph below:

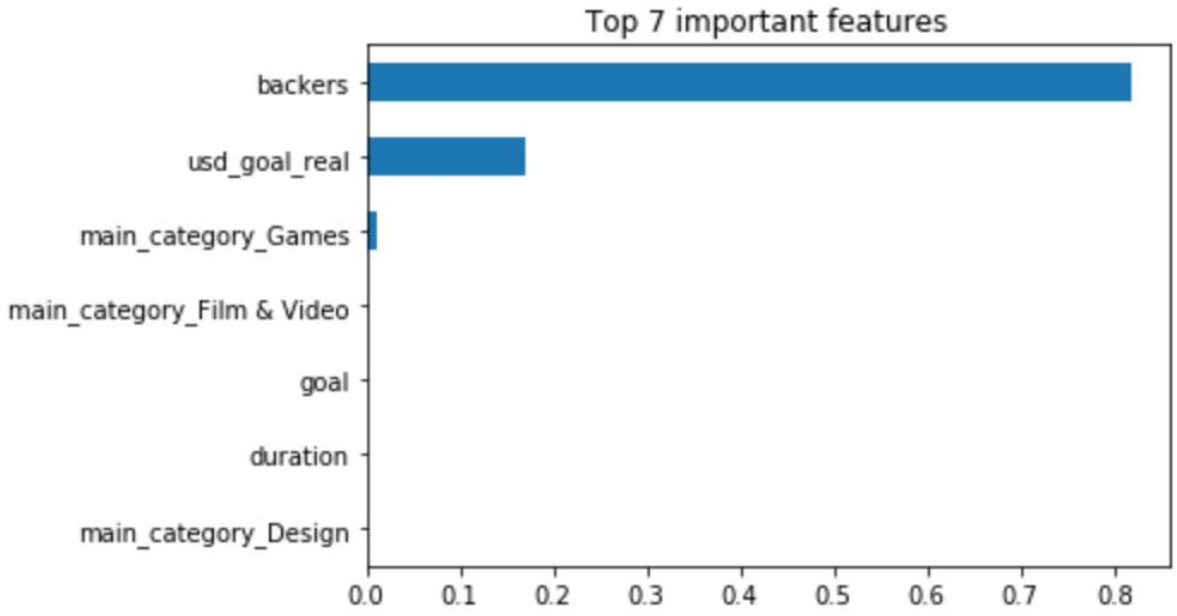


When this also did very little to change the results, I tried adding in a column of all ones to adjust the intercept point, and I tried both scaling the data, and applying a log transform to convert the data to normal distribution. Because these methods were also not fruitful, I moved on to try other algorithms.

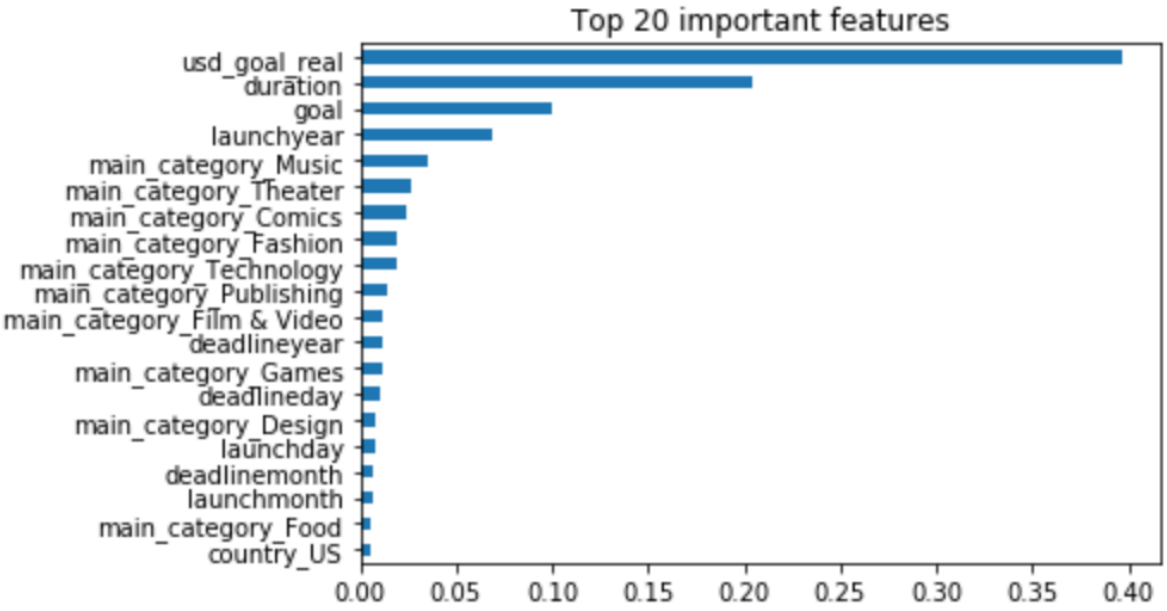
The next algorithm I tried was random forests. When tested with default parameters the model overfit with a near 100% accuracy rate on the training data. Hyperparameter tuning significantly reduced the overfitting, but the accuracy rate on the test data was 87%, lower than the logistic regression accuracy. Further tuning to balance the weight ratio between data classified as successful and failed improved this score to 88%.

The initial test on a decision tree model with default parameters was also overfit. However, with hyperparameter tuning, the issue of overfitting was resolved and the accuracy score for the training data was 92.88% and for the testing data 92.47%. The parameters tuned in this case were max_depth, min_samples_leaf, and min_samples_split. I also tried balancing the data, as I also did for the random forest model, but this did not improve the score. The scores from the tuned decision tree model were in fact the best scores I was able to achieve out of all the models I tested.

I further examined which columns were most important for making splitting decisions in the tree, and I found that the backers column was by far the most significant:



Testing without the backers column resulted in a wider range of columns being used to make the determination, although the accuracy was significantly reduced for the testing data: from 92.47% to 64.29%:



I also tested a K-Nearest Neighbors model which resulted in a maximum accuracy score of 90.57% for the test data. This was also a very good score, but the decision tree model was still better.

The decision tree model has a number of advantages that made it a good candidate for a dataset like this one:

- It works well with categorical values and can discretize continuous values.
- It is not susceptible to outliers.
- It is a non-parametric algorithm, and therefore works for many functional forms.
- It does not make assumptions about the distribution of data.

However, there are also some drawbacks:

- It is easy to overfit.
- The training can result in tree bias if certain features are more significant.
- It requires careful attention to parameter tuning.

When testing the decision tree model with default parameters, we did see that it overfit. And with the backers column removed, the performance was much less accurate. This may be due to some of the drawbacks of the decision tree model noted above. The backers column was clearly significant for splitting decisions, so it could be the case that the tree was biased and performed less well without it. This issue could also stem from the fact that both the backers and usd_goal_real columns were continuous. But even though it is continuous, the range of values in the backers column is much less than the range of values in usd_goal_real, which may make it easier to find repeated values and easier to discretize.

In hindsight it might have been a good idea to leave the backers column out from the beginning in the machine learning section, as this is not a variable that would be known when first starting a project on Kickstarter. In other words, if we wanted to help individuals optimize their projects before going live, we should only look at variables that the project owners can control. However, in this case we looked at the number of backers without considering at all the amount contributed. Perhaps this can suggest that project owners who focus their efforts on getting as many backers as possible might increase their chances of success. In any case, if the number of backers is known in advance, without any data to suggest the amount pledged, it is possible to predict the success of a project at a fairly accurate rate.