The Successes and Failures of Kickstarter Campaigns

Authors: AUTHOR1 and AUTHOR2

Summary of Research Questions:

1. **What are the factors that contribute to a Kickstarter campaign's success rate?**

Result: number of backers is the most important predictor for the success rate for a Kickstarter campaign, other factors that could have minor effect includes launch time and country. The main category of projects could also affect backer's preference or willingness to support a project, and different countries have different trends in terms of the popularity of crowdsourcing with the US having the highest backer support rate of successful projects.

2. **What are the underlying trends of types of projects that backers are more willing to support?**

Result: backers are more likely to support game projects.

3. **How accurate can a machine learning model predict the outcome of any project given the current data? What are the most important features that contributes to the accuracy of the predictions?**

Result: a DecisionTreeClassifier is able to predict whether a project will be successful or failed based on the goal amount, number of backers, main category of the project, launched month of the project, and the duration of the campaign at approximately 92% accuracy, with the goal amount and number of backers being the strongest predictor. If remove number of backers, the prediction accuracy lowers to approximately 62%.

4. **Do project success rate differ across different countries?**

Result: US has the highest project success rate among all countries.

5. **When is the most common time to launch a project on Kickstarter? Is this a major contributor to the success or failure of a project, or are other factors more important?**

Result: the most common time to launch a project on Kickstarter is in July, the second most common time is in March, and the least common time is in December. If only look at the percentage of successful projects for each month, December has the lowest success rate, July has the second lowest, while March has the highest success rate.

Motivation and Background:

Kickstarter is a platform for entrepreneurs and creators to ask their followers and supporters to fund their projects. It is also one of the most popular self-funding platforms that exist. Not surprisingly, this also means most projects that hit the ground on Kickstarter won't succeed. This could be for any number of reasons, and this is what we intend on discovering. With this dataset, we want to learn what makes a project successful, and hopefully inform creators and self-driven individuals who want to contribute to the world through Kickstarter. If we can take projects from the past and analyze the data, the answer could help us inform the future.

Dataset:

The data is collected from the Kickstarter Platform where people can raise money for their own projects through the online platform. People who pledge to the campaign are known as backers, and in general they are able to enjoy certain benefits based on the amount they pledge. The data includes projects from 23 countries, with the vast majority from the US. Other major contributors include the UK, Canada, and Australia. The launch dates of the campaigns range from April 2009, when the platform is launched, to January 2018, when the dataset is last updated.

Methodology:

1. Preprocessing data for analysis by filtering data to include only
    a. Filter to only include data from 2010-01-01 to 2017-12-31 (those years all have full-year data, more consistent for TimeSeries analysis)
    b. We will also add three columns that computing information that could be helpful for analysis. One calculates the ratio of the pledged amount with respect to the goal, which is calculated using usd_pledged_real amount divide by the usd_goal_real, and put it in a column name pledged_ratio. The other extracts the month from the launched date as a string and put it in a column name launched_month. The last one computes the duration of the campaign in days using the launched and deadline column.

2. Subtasks for answering each question:

(1) This question will serve as our driving research question, which will be broken down into different aspects and explored in the following questions. We will use this question in our conclusion after our data analysis and visualization

(2) In exploring the trends in which projects backers will support, we'll arrange the data in descending order by the number of backers. With the ordered data, we'll cut the data into percentiles and find all the unique values of a certain feature (e.g. category, main category). From this, we can calculate the percentages of each unique value within the section of data to see which features are most popular among each percentile, which we can then use to visualize in some sort of graph.

(3) To generate different machine learning models and test their accuracies, we will first filter the data to only include projects with “successful” and “failed” states.
    (a) We will test the model on two sets of features:
        (1) Usd_goal_real, backers, main_category, launched_month, duration
        (2) Usd_goal_real, main_category, launched_month, duration
    (b) We will first tune the max_depth for each decision tree model by training a model on multiple different max depths from 3 to 20 and graph out the accuracy score gained from cross validation with respect to the max depth. Then we train the model again using the max depth with the highest cross validation score and use it to predict the result with the test set, and use the resulting accuracy score to evaluate our model.
    (c) For each set of features, we will train a separate model using a max goal amount from $10000 to $40000 in $10000 increments to test if the prediction accuracy and the hyperparameter is generalizable.
    (d) We will also visualize a decision tree with a smaller goal range and see if it is informative.
(4) To see how trends differ in each country, we will first aggregate the data by the country a project falls in. From each country’s data, we can do various forms of basic statistical analysis by features, such as the percentage of successful vs. failed projects, or the most popular project categories per country.
(5) To determine the most common launch time for projects, we will first resample the data based on months. Then we will make a TimeSeries plot comparing the number of total projects and successful/failed projects, then also plot the success rate of the projects over the years. We will also group the data by launched months and generate a bar chart showing the count of projects with respect to the launch month, as well as another bar chart showing the percentage of successful projects with respect to the launch month.

Work Plan:

1. Set up the starter files and Github repository, share access
    a. Starter files include main.py, kickstarter.csv (finish by May.22nd)
    b. Create separate branches and work on different files including functions answering the problems that we each responsible for
    c. Merge progress when half way through the work process to avoid big merge conflicts
2. Responsibilities
    a. AUTHOR1: problems 3 and 5
        i. Finish problem 3 related methods by May 28th
        ii. Finish problem 5 related methods by June 1st
    b. AUTHOR2: problems 2 and 4
        i. Finish problem 2 related methods by May 29th
        ii. Finish problem 4 related methods by June 1st

3. Time estimates:
    a. Finish functions that answer the problems and code documentation by June 2
    b. Finish written report by June 7 (probably/preferably earlier)
    c. Finish the in-class presentation (materials and practice) by June 10

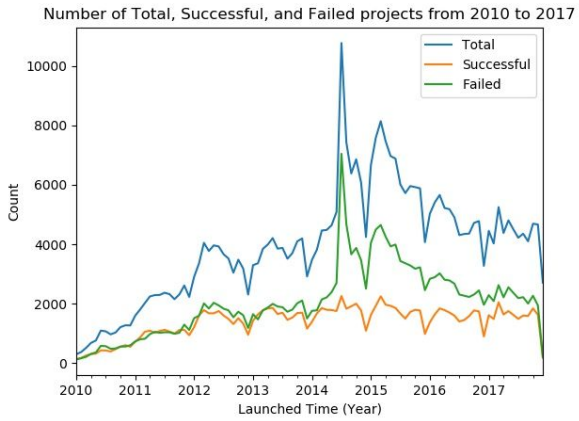
Results:

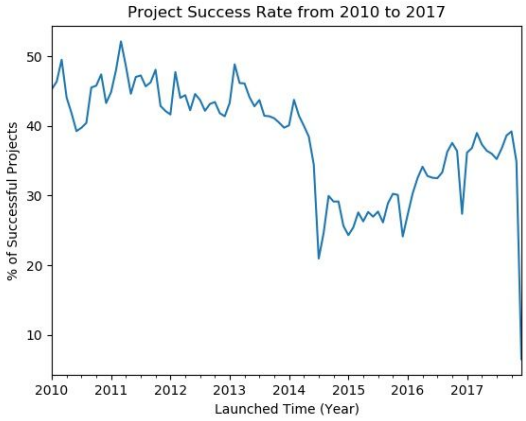
Predicting the Outcome with Machine Learning

Through multiple test trials, the output of predicting the successful/failed state with a DecisionTreeClassifier using feature set (1), which includes usd_goal_real, backers, main_category, launched_month and duration, has an accuracy score of around 0.92 across all tested max goal amounts (from $10000 to $40000 in $10000 increments), and two of the most important features that contribute to the high accuracy are backers and usd_goal_real. For DecisionTreeClassifiers, the importance of each feature is measured in “gini importance” or “mean decrease impurity”, which is defined as the total decrease in node impurity averaged over all trees of the ensemble. Basically, it is a number ranges from 0 to 1 and the closer the value is to 1, the more important the feature is for the model to make decisions. We can see from the output in ml_output1.txt that as the max goal amount increases, the feature importance of the number of backers decreases (from around 0.85 to 0.79), and the feature importance of the goal amount increases (from around 0.13 to 0.19). This is reasonable since the goal amount is expected to play a more important role in the success and failure of the campaigns as the maximum goal amount varies in a greater range. For feature set (2) where backers is removed, the accuracy score drops drastically to around 0.62. So other features are probably not a good indicator for the successful/failed state for Kickstarter projects.

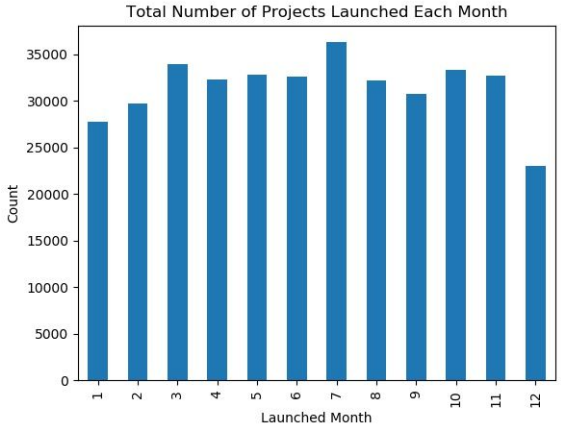
We also narrow the goal range to $3000 to $5000 and try to visualize the decision tree using graphviz. Since the resulting visualization is very big due to the complexity of the model, we do not put it in the report for poor readability. The visualization is stored in model.svg and can be examined in more detail in a Chrome browser. The accuracy score is also above 0.92 according to the output in ml_output3.txt, and the resulting decision tree has a very clear first split on the number of backers at 23.5. Most of the projects with less than or equal to 23.5 backers result in failure and the ones with more than 23.5 backers result are more likely to succeed, so it might be a good hint that if people have a project with a goal amount between the range of $3000 to $5000, they should try to attract at least 23 backers to support their campaign. Our algorithm is flexible enough for exploring different goal ranges so it can give users a good idea of how many backers they need for the project to succeed given a budget, or what goal they should set for themselves given a projected number of backers.

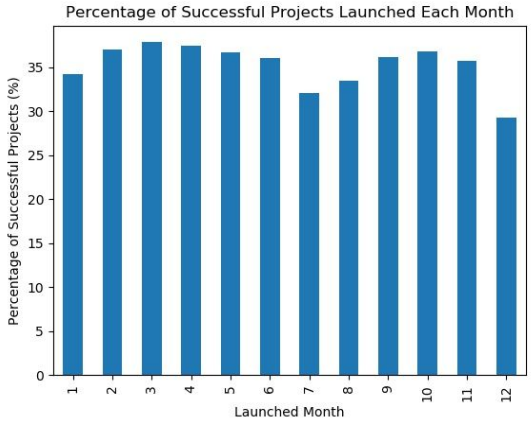
Analyzing the Most Common Launch Time





According to the graphs above, in mid-2014 there are a great number of projects launched on Kickstarter, while the number of successful projects is consistent over the years at less than 2000 per month. Correspondingly, we see a sharp drop in success rate during the influx of projects.



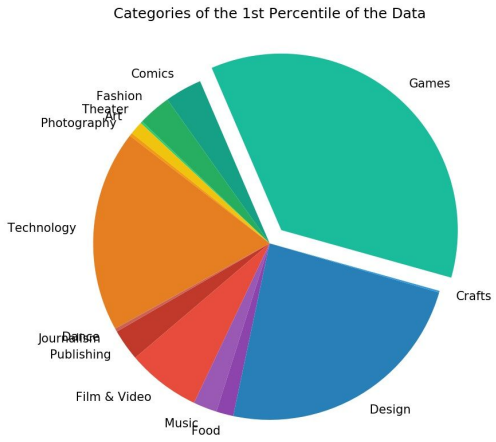


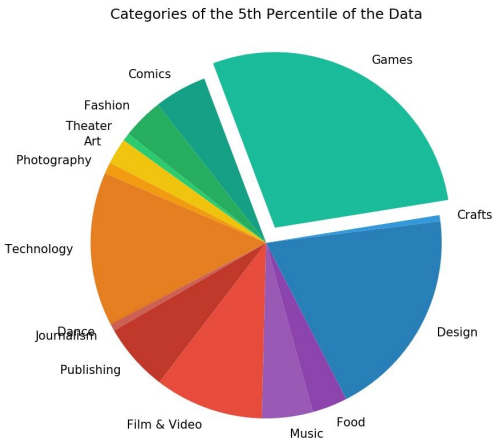
From these two graphs, we can see that the least popular month to launch a project is December, and the most popular month is July. It is reasonable that people avoid December because of a lower success rate compared to other months. However, we can see in the graph on the right that when it comes to success rate, March and October might be a better option than July since they have higher success rates as well as decent number of project counts.

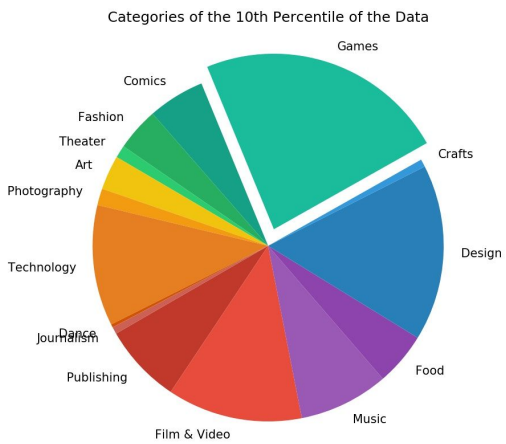
Calculating Summary Statistics

In conducting summary statistics of the data, we wanted to find out which of the projects on Kickstarter were performing the best. Since we've learned that the amount of backers is the most important aspect in determining a project's success, we decided to sort the data from top to bottom by the number of backers. From there, we identified different percentiles of the data

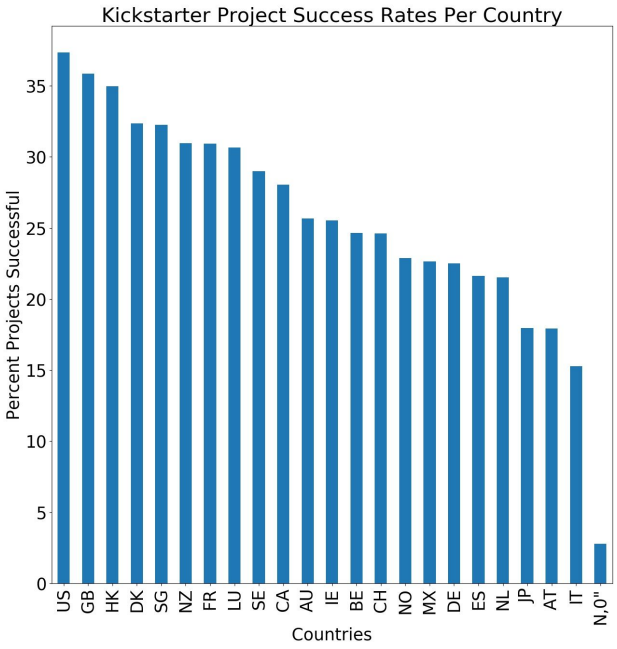
(first, fifth, and tenth). These percentiles proved to have unique statistics, which revealed some interesting facts about the data. Below are pie charts that visualize the categories that make up each percentile.







These graphs show how much of each main category take up of only **successful** projects. It is clearly visible that Games make up the majority of these successful projects consistently. The next most successful types of projects are Design and Technology. These categories also happen to be the most popular types of projects in our dataset.



We also wanted to see if there were differing trends in each country in terms of successful projects. Our thought process was that patterns of crowdsourcing in different countries would differ since the populations are different. This was found to be the case. On the left is a graph of the success rates of Kickstarter projects of each respective country.

It is clearly seen that the United States has the highest success rate of projects than any other country. This could be for many reasons, but some factors that could affect this result are the fact that Kickstarter is largely an American platform, and therefore is used most commonly in America. However, there are some regions that fall close behind the United States, such as the United Kingdom and Hong Kong; both regions are top proprietors of entrepreneurial ventures and products.

Reproducing the Results:

1. Download the dataset from this link: https://www.kaggle.com/kemical/kickstarter-projects
2. Extract the zip file and put **ks-projects-2018.csv** in the same folder as the source code; this is the *only* file you will need from the download.
3. All code should run properly in cse163 environment with an extra package graphviz installed. To install graphviz, type the following in the terminal or anaconda prompt:
   conda install -c anaconda graphviz -n cse163
   conda install python-graphviz -n cse163
4. Run main.py and get all the result in the results folder.
   - For ML-related results, all output using the same output index will be written to ml_output<output index>.txt. The output includes the accuracy score and feature importance ranking. Graphs that illustrates max depth vs. accuracy score used for testing can also be generated in the test folder if passed in test=True parameter for classifier_trial function. The file name will be in the format of max_depth_vs_accuracy_<min goal>_to_<max goal>(<output index>).jpg. The graphviz source file model.gv will also be generated in the results folder if passed in graph=True parameter for classifier_trial function, and to visualize the model, copy and paste its content in http://graphviz.it or other online graphviz visualizer.

- For launch time related results, the line graphs using TimeSeries analysis are named project_count_over_time.jpg and success_rate_over_time.jpg, and the bar graphs are named launched_count_over_month.jpg and success_rate_over_month.jpg.
- For statistical analysis related results, there are four graphs that resulted from this aspect of the data processing. These graphs will fall in the results folder. These files are listed below:
    - first_perc_categories.jpg
    - fifth_perc_categories.jpg
    - tenth_perc_categories.jpg
    - Success_rates_per_country.jpg

  These graphs illustrate the percentage of project categories that make up the first, fifth, and tenth percentiles of the entire dataset respectively. The last graph is a visualization of the success rates of Kickstarter projects in each country that has a project in the dataset.

5. Run test.py to check for function usability and get test results in the test folder.

Work Plan Evaluation:

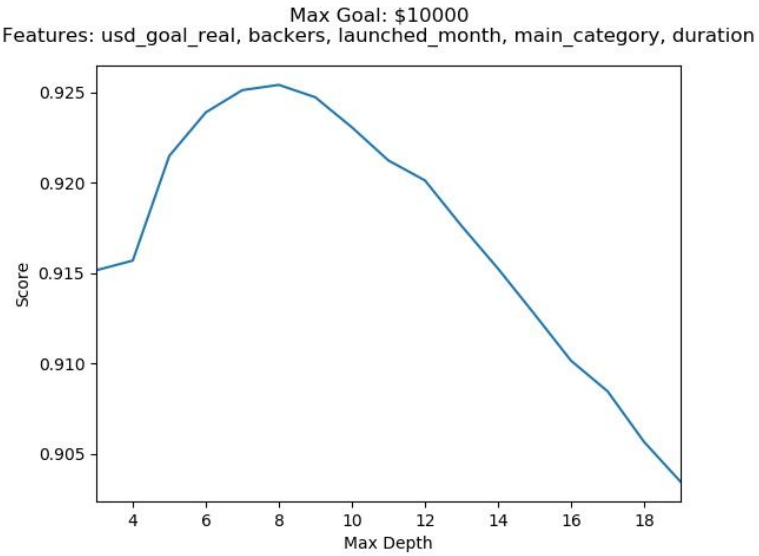
The coding portion of the project takes longer than expected and in reality, we are writing the report and optimizing the code at the same time. One part that required more thought is not only generating the output, but also making them easy to reproduce. At first we mostly printed results to the console or saved figures in the same directory as our code files, but we changed it in later stages to saving the results and test output in separate folders. Not having testing in mind while writing the functions also slows down our progress because we need to go back and figure out how to restructure our code and add flexibility for testing.

Collaboration on GitHub turns out to be a good decision and with enough communication and coordination, we did not run into big merge conflicts and can work on our own parts more efficiently.

Deadlines ended up becoming more flexible for the main development stage. While some aspects of the project were easily completed, the main development and testing stages took longer than expected. Some of the graphs proved to be more difficult to develop, as formatting was an issue with this particular dataset.

Testing:

For the machine learning portion of the project, we first find the best max depth of the model by iterating through different max depths and decide its accuracy score using cross-validation. Then we find the max depth associated with the highest accuracy score, train a model using that max depth, and use its accuracy score predicting the test set as the final measurement of accuracy. We also generate a plot to visualize the result of iterations for each model and print out what max depth the final model chose to use to make sure they match.



For example, this is a graph generated from one of the trials after setting test=True in the classifier function, and we can see that at max depth of 8, the model yields the highest cross-validation accuracy score. In our console output that prints out the max depth used for the decision tree of this trial, it says that the model is “Predicting test set using the depth of: 8”. So it proves that the algorithm actually picks the optimal depth for doing the prediction on test set instead of an arbitrary depth that could subject to the issue of overfitting. The result is also very consistent across different trials so the result should be reliable.

For the launched time analysis, we used a smaller test dataset that have only 26 projects from 2015 and generated the four figures based on that dataset. It is clear through inspection that the resulting figures are what is expected from the dataset. To retrieve those test figures, simply run test.py and the figures will show up in the test folder.

For testing the statistical analysis, many of the functions produced output similar to functions defined in previous homework assignments, so the best testing tool for these functions was the assert_equals method. For these functions, a very small dataset of two observations was used to ensure a correct output. Since these functions were designed for our specific analysis, the outputs turned out to be as expected in both tests and applications. The graphs from the statistical analysis were also as expected.

Collaboration: None