**Project Report – Group 3**

**Topic: Factors impacting the success rate of a crowdfunding campaign on Kickstarter**

**Introduction**

1. **Problem statement**

Crowdfunding has become one of the most popular funding techniques that start-ups utilize. However the probability of successfully raising funds through crowdfunding is still quite uncertain. If the start-ups can know the factors that impact the probability of successfully collecting the goal amount, they could tweak their campaign or business model and in turn increase their chances of success. That is what we aim to do through this project. We want to understand what factors impact the success and if we can come up with a model that can be used to predict the probability of success.

1. **What persuaded us to work on the problem**

A lot of people have innovative ideas but very few people have enough capital to implement their ideas. Most of the start-ups are unable to receive loans from the banks and venture capitalist because these financial institutions are trying to minimize risk and they avoid start-ups that are not well established. Thus the only hope that remains for these start-ups is crowdfunding. Currently, Kickstarter is the largest and most popular crowdfunding platform. A project that does not succeed on Kickstarter is detrimental to both the creator and the backer, especially so for the creators who will not be able to further develop and launch their products. Thus, creating a model which predicts whether a project will succeed or not will be helpful for both creators and backers. Besides, an accurate prediction can motivate and inspire creators to mold their projects in order to have a higher chances of success.

1. **What we do in this project**

We have used a dataset of about 379,000 rows and 15 columns. This data contains details of 379,000 projects uploaded on Kickstarter from 2009 to 2018. (*Data source:* [*https://www.kaggle.com*](https://www.kaggle.com)*)*. We perform exploratory data analysis on this dataset to find relations between various columns, specifically impact of various variables on “state” of the project. The variable state depicts whether the campaign was successful, failed, cancelled, live etc. After an extensive analysis, we understood what factors impact the success rate of projects. We used K-means clustering to group main categories into three groups. For future updates, we can think of using these groups in the model. From exploratory analysis we have an idea of what factors are playing a role, we go on to creating a regression model that will consider probability of success as the independent variable. However, before building the model, we have to consider the significance test for each variable. We limit the number of variables based on results of recursive feature elimination (RFE). After performing these tests, we picked up the important variables and used these to come up with the logistic regression model. To understand how variation in goal amount affects the success rate, we simulate the goal amount and calculate the success rate. In the final step we compute the efficiency of our code by using time functions.

**Computational Setup – Outline of the steps on Python**

1. **Automated dataset load**

We have imported the libraries so that we can use their features throughout the project. We have set up an automated process that would go to the website and download the zip file in your local directory. It would then unzip the file and import the data file into Python using read\_csv function from Pandas library. We have generated the descriptive statistics that summarize the central tendency, dispersion and shape of a dataset’s distribution using .describe() function from the pandas library. Lastly, we check to see if any of the columns have null data. Since the two columns that have null data are not important to the analysis, we ignore it.

1. **Exploratory data analysis**

Some of the exploratory analysis we performed is as follows:

* We found the number of projects launched on Kickstarter per year, success or failure rate, total number of projects in every main category, percentage of successful projects every year.
* We found % of successful and % of failure projects for each month to see if there is any seasonality.
* Next analysis calculates the average pledged amount for a successful and failed project.
* To analyze influence of category more, we calculate success rate per category.
* The last graphical analysis we do is with number of backers for each category.
* We depicted the success rates of projects in each country on an interactive world map.

1. **Clustering**

Since our data set has 15 main categories, we will try and group these categories into clusters by considering the parameters "Goal amount" and "Pledged sum". We first create a dataframe called "df\_cluster" which contains total amounts of goal and total amounts of pledged for all projects in each original categories. We then used Sklearn to implement Elbow method to determine the number of clusters. As we notice the elbow of SSE appears between 3 and 4 clusters, we decided to implement 3 clusters. Next, we used Sklearn to utilize K-means method to cluster the data. And then we grouped the main categories into 3 major categories.

1. **Linear regression**

To perform linear regression, we make use of the sklearn and statsmodel.formula.api libraries. We performed linear regression to get a relationship among three variables - backers, usd pledged real and usd goal. We used a heat map to get correlation between the three variables and we now restrict the data to only successful and failed state. After plotting a pair plot using seaborn, we observed a negative correlation for backers and usd goal and pledged real and usd goal. We then created a model. From the model we saw that backers and pledged amount have a positive correlation but the relationship is not linear. Thus we used a log scale to see the correlation and created a linear model using the log of backers and log of pledged amount.

1. **Logistic regression**

Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In our dataset, the state of the launch, as successful or failed, was chosen as the binary dependent variable. We started out with first filtering the data and checking for any variations present in the data. It was observed that the goal amount was widely distributed. Since the log transformation can be used to make highly skewed distributions less skewed therefore we carried out log transformation on this column. Next we created dummy variables for all of the categorical variables present in the dataset.

We then used the Recursive Feature Elimination (RFE) to work out the combination of attributes that contribute to the prediction on the target variable. Recursive feature elimination is based on the idea to repeatedly construct a model and choose either the best or worst performing feature (for example based on coefficients), setting the feature aside and then repeating the process with the rest of the features. This process is applied until all features in the dataset are exhausted. Features are then ranked according to when they were eliminated. This process was included because it’s an efficient optimization technique for finding the best performing subset of features. The influential columns we got from this as output was used as the input variables for the logistic regression model.

We implemented the model and found out that accuracy of logistic regression classifier was 64%. Then we carried out 10-fold Cross-Validation to train our model in order to avoid overfitting. 10-fold cross validation average accuracy for the model came out to be 0.64.  As the average accuracy remained same as the Logistic Regression model accuracy; hence, we concluded that our model generalizes well. We then plotted the confusion matrix, the results obtained conveyed that we have 48,342+15,549 correct predictions and 24,452+11,160 incorrect predictions. We then calculated the precision, recall and F-measure. As we know, the area under the ROC curve of a test is used as a criterion to measure the test's discriminative ability, i.e. how good is the test in a given situation. Therefore, we plotted the ROC curve to get an understanding of sensitivity and specificity. The area under the ROC curve came out to be 0.6, which implied that our model was fairly good however there were scope for improvements.

1. **Simulations**

We carried out the simulation on the logistic regression model. Here we used the coefficients obtained from the logistic regression model to run a simulation. We varied the goal for a start-up & observed the change in probability of having a successful launch.

**Computation time and efficiency**

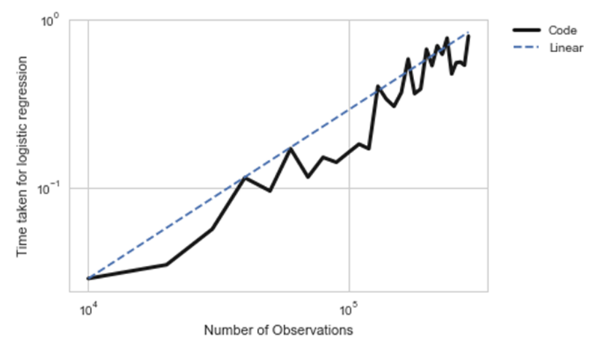
1. **Computational challenges and solutions**

Our logistic regression included main category as one of the independent variable. The main category has 15 unique values. Thus we had to create dummy variables to account for the variable. Using the normal programming concepts, we used loops to create these dummy variables and tried to use each dummy variable in the regression calculation. However since there were so many variables, there was a lot of processes running on the backend. Our code threw memory full error. To overcome this situation, we researched about libraries that perform the same task as above but in an optimized manner.  This is when we came across the recursive feature elimination. We used the code and we were finally able to create our model without errors.

1. **Slowest part of the code**

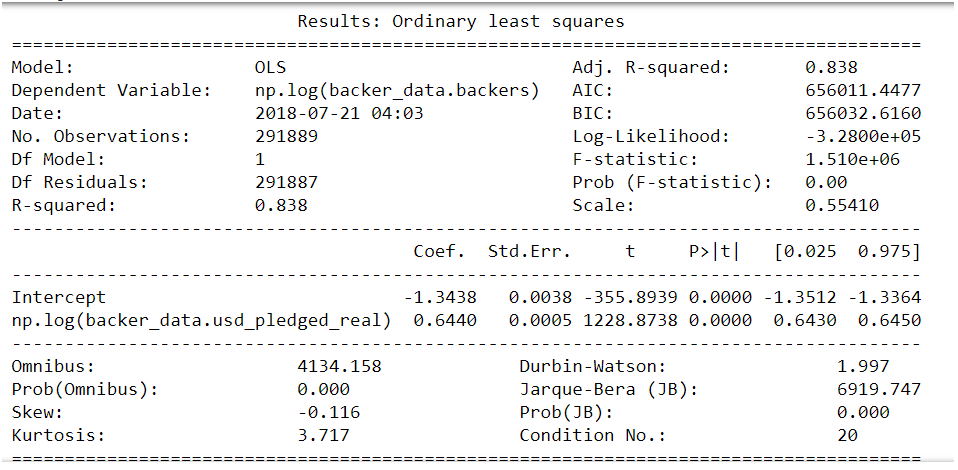
The slowest part of our code is present within the logistic regression. Before actually calculating the model summary, we use Recursive Feature Elimination (RFE) which is based on the idea to repeatedly construct a model and choose either the best or worst performing feature, setting the feature aside and then repeating the process with the rest of the features. This process is applied until all features in the dataset are exhausted. The goal of RFE is to select features by recursively considering smaller and smaller sets of features. Since this process recursively iterates and calculates the model, it calls a lot of functions and the time taken is significantly greater than run time of creating a simple linear model.

1. **Variation in run time based on number of observations**

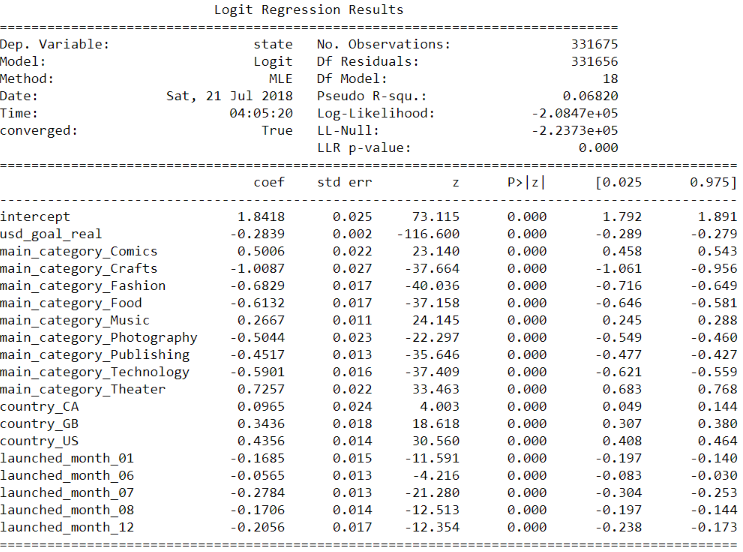
To understand how an increase in number of variation can affect our run time, we conducted a simulation that increased the number of observations and calculated the run time for our slowest piece of code – logistic regression. We notice a non-linear upward trend. Thus if we increase our data size by two times or three times, the run time for our code, specifically the logistic regression will increase substantially.

**Results**

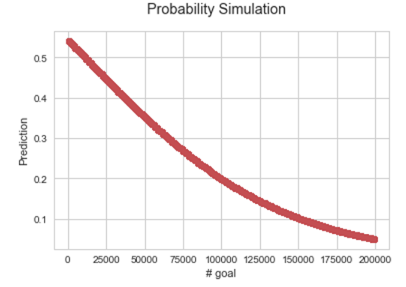
* Our logistic regression model has an accuracy of 64%. We have validated this conclusion via confusion matrix and ROC plot
* Linear regression model:



* Logistic regression model:



* Simulation graph: This shows that with other variables kept constant, the goal amount is negatively correlated with the probability of success.



**Summary and Conclusion**

* After performing the exploratory data analysis, we realized the following factors to have an impact on probability of a project successfully receiving funds via Kickstarter:
  1. The month of launch of the campaign; the campaign has a higher success rate in February, March and April and have a lesser success rate in July and December.
  2. The main category of the product; some of the categories have a higher success rate.
  3. Country of the campaign; US seems to have a higher success rate than other countries
* We were able to further group the categories into three groups using K-means clustering
* We used linear regression and correlation plots to establish a relationship between number of backers, goal amount and pledged amount. We were able to come with a strong relation between log of number of backers and log of pledged amount.
* We also used simulations to predict the probability of success by changing the goal amount. Through the graph we could see that goal is negatively correlated with probability of success. Thus to increase the chances of success, the startups should set a conservative goal without a buffer.
* Finally, we came up with a prediction model using logistic regression that could be used in the future to predict success rate given a particular category, country, goal and month of launch.

**Interesting extensions**

After going through the entire analysis and understanding the driving factors, we thought of some interesting theories that could be explored further. During the exploratory analysis, we found that there is a positive correlation between the number of backers and the success rate. The campaign itself decides whether a project looks attractive to backers or not. Thus there is still room for utilizing promoting tools to increase project's exposure and attract potential backers. Therefore, investigating in how the different promoting channels would make impact the success rate would be useful for project creators. Specifically, we could further investigate how the promotion activities on the social network platforms would alter the rate of success. In addition, developing an algorithm to track in-progress funding campaigns and making predictions based on live data of the in-progress projects could also be valuable. With this algorithm, we would not only be able to provide prediction on the rate of success but also be able to provide customized, timely and cost-efficient promoting solutions to project creators.