Kickstarter Campaigns Analysis: Success or Failure

Parin Patel

Table of Contents

Introduction

Project Background and Description

Crowdfunding is a great way to raise money for a project or passion. For example, take inventors like Igor Zamlinksy and Gleb Polykov, who are trying to use crowdfunded-sources to invent a PID-controlled Espresso Machine. While the concept of using crowd-sourcing through the internet to fund ideas is much better than the alternative, like getting a bank loan, a crowdfunding campaign can make or break a business idea or career. Therefore, the goal of this project is to investigate what makes a campaign successful, using data taken from Kickstarter – one of the best-known crowdfunding organization.

Specifically, our Kickstarter crowdfunding campaigns dataset focused on campaign success prediction. According to Kickstarter's website, "Kickstarter helps artists, musicians, filmmakers, designers, and other creators find the resources and support they need to make their ideas a reality. To date, tens of thousands of creative projects — big and small — have come to life with the support of the Kickstarter community." The company provides opportunities for individuals to fund their creative projects using entirely crowdfunding resources, meaning that the public is what sends these projects into production. Every project is brought to reality while friends, fans, or complete strangers offer their funds in return for rewards or the finished product itself.

Business Questions

Based on a preliminary examination of the dataset, the following key questions should be addressed during the analysis. These focal questions aim to diagnose potential factors that affect Kickstarter campaign effectiveness:

- What demographics (if any) are most likely to succeed?
- Which categories of product or service are more successful in funding their projects?
- Which is the most critical factor affecting each campaign?

While many potential variables could be the cause of success or failure, the goal is to build a prediction model that would identify those major campaign success factors.

Data Overview

About the Data

The data was scraped from webrobots.io (https://webrobots.io/kickstarter-datasets/), combined, pre-processed, and created into a dataset that contains project data from 2014 to February 2019. The dataset is available on Kaggle.com: https://www.kaggle.com/yashkantharia/kickstarter-campaigns.

The dataset contains 20 variables and over 192K records. Majority of the data describes various campaign characteristics such as campaign length, location, start and end dates, amount of money needed, and whether campaigns succeeded in being funded. Variable named “status” identifies each campaign success or failure. The goal is to build an accurate classification model using various machine learning algorithms, predicting the binary outcome of each campaign: succeeded or failed.

A data dictionary is used for this dataset to keep track of the variables. For each variable in the dataset, the data dictionary contains the index, column name, the definition of the variable, and variable type. The data dictionary is provided in Table 1.

Table 1. Data Dictionary

Index	Column Name	Definition	Variable Type
1	id	Campaign id	numeric
2	name	Name of the Kickstarter Campaign	character, unique values
3	currency	Currency used	nominal (14 distinct values)
4	main_category	Main category of the project	nominal (15 distinct values)
5	sub_category	Sub_category of the project	nominal (159 distinct values)
6	launched_at	Launching date of the campaign	character, date
7	deadline	Deadline for the campaign. This is the date when the campaign ends	character, date
8	duration	Number of days the campaign was online	numeric
9	goal_usd	Goal set by the campaign owner	numeric
10	city	city	character
11	state	state	character

12	country	country	character, 22 distinct values
13	blurb_length	Word count of the description of campaign	numeric
14	name_length	Word count of the name of campaign	numeric
15	status	Status of the project	nominal, binary: successful, failed
16	start_month	Month when the project started	numeric, 12 distinct values
17	end_month	Month when the project ended	numeric, 12 distinct values
18	start_Q	Quarter when the project started	nominal, 4 distinct values
19	end_Q	Quarter when the project ended	nominal, 4 distinct values
20	usd_pledged	Amount collected at the end of the campaign in USD	numeric

Data structure is defined as a specific form of organizing and storing data. R supports five basic types of data structure: vector, matrix, list, data frame, and factor. Kickstarter campaigns dataset is structured as a data frame where each component is of the same length. The structure of the dataset can be generated with str() function in R.

Data Selection

Data selection is defined as the process of determining the appropriate data type and source, as well as suitable instruments to collect data. The dataset in question contains quantitative data described in numerical values, for example, campaign durations in days, campaign goal amount, and nominal data with distinct values, for example, country, city, and state, which help to measure specific demographic attributes.

The customer dataset contains both types of data: discrete and continuous. Continuous variables consist of measured data, such as campaign goal and pledged amounts. The majority of variables in the dataset contain discrete data that helps to identify demographic attributes. Analysis of demographic data helps to identify specific patterns related to campaign success.

To better answer our business questions, it was decided to analyze the entire dataset of Kickstarter campaigns. This provides more information on different campaign segments, allowing for comparison of the demographic behaviors, start and end terms, monetary goals that affect campaign success.

The analysis focused on identifying how various demographic attributes, start and end terms, monetary goals, or independent variables, affect campaign outcome, which in this analysis represents the dependent variable.

Data Preprocessing

Data cleansing is the process of preparing data for analysis by modifying or removing any incorrect, incomplete, irrelevant, or duplicated data in a given storage resource. The below-listed steps were performed in the data cleaning stage to prepare the data into a format compatible for analysis.

- Data Type conversions
- Re-ordering data columns to a meaningful order
- Check for duplicates
- Check for missing values
- Variable creation, sub-setting the date variables
- Data consolidation used to reduce the number of parameters in the predictive analysis

At initial glance, the Kickstarter data has 192548 observations across 20 variables.

First, data type conversion was performed. This conversion was first done to the ID and Name columns, which were made to be “Character” data types. Next, deadline and Launched_at were converted into the Date datatype. Then, the data frame’s columns were re-ordered to better understand the information.

Going further, check for duplicate and missing values yielded to no results. This first part of cleaning was relatively quick and straightforward, but the next few tasks were much more tedious.

Subsetting the date variables and consolidating some of the columns was beneficial due to the sheer volume of the data. There are many categorical variables with multiple levels. When running initial predictive analysis, there were many data-size restrictions. The solution was found in the consolidation of some variables, such as currency and country. Further, the deadline and launched date fields were separated into the respective year and month as variables, deadline_year, deadline_month and launched_year, launched_month.

Country is the first consolidated variable. The highest number of projects were launched in the US, Great Britain, Canada, and Australia, as shown in Figure 1. It is beneficial to focus on these countries and consolidate all projects launched in "AT", "BE", "CH", "DE", "DK", "ES", "FR", "HK", "IE", "IT", "JP", "LU", "MX", "NL", "NO", "NZ", "SE", "SG" into a group called “other”. The post-consolidation spread of countries is shown in Figure 2.

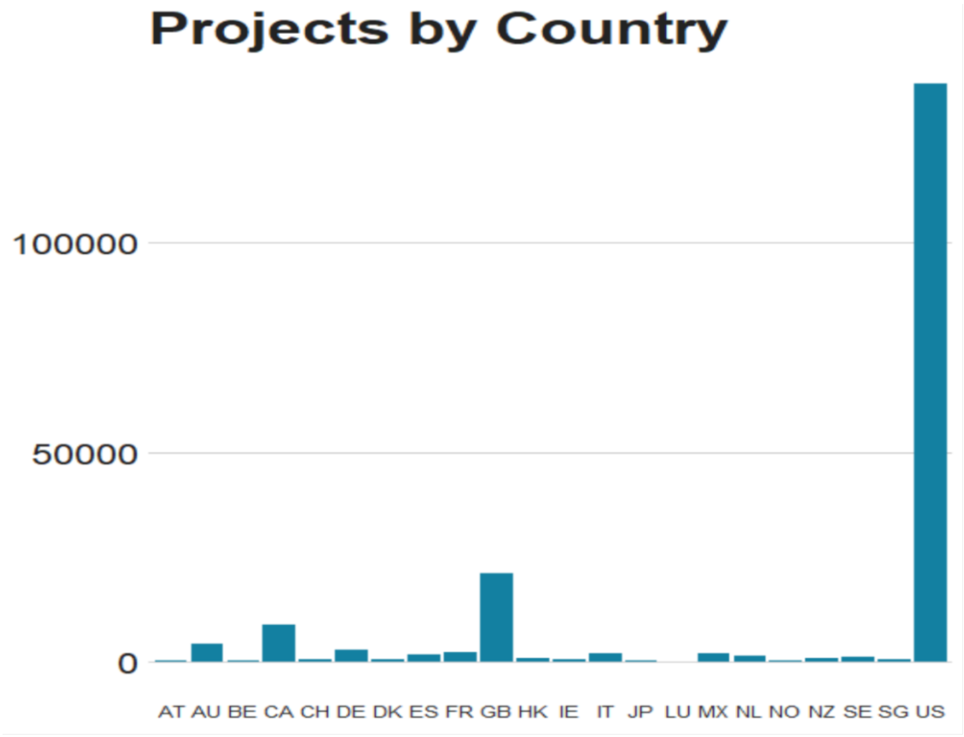


Figure 1

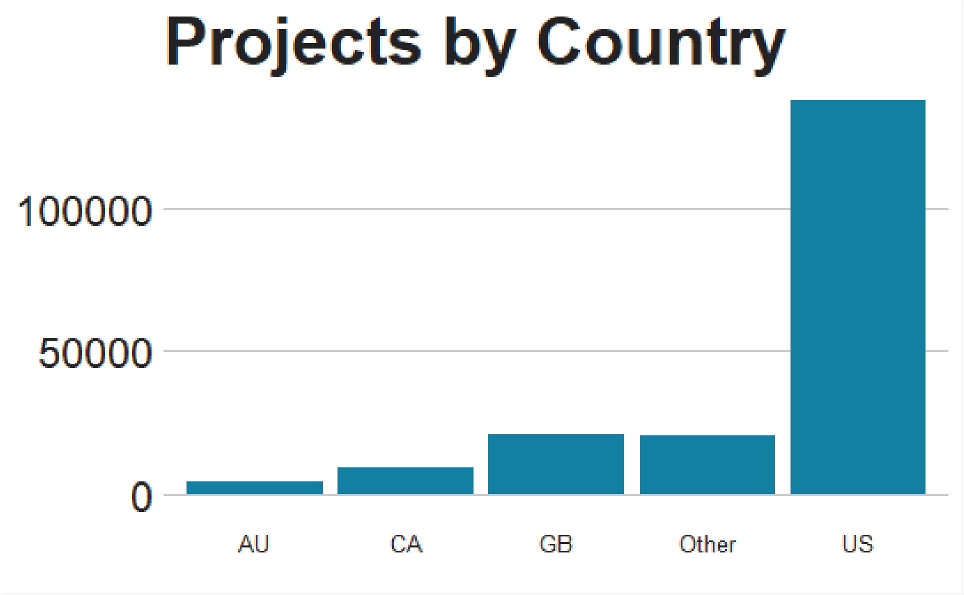


Figure 2

Next, the currency column of the dataset was cleaned. Initially, there were 14 different forms of currencies used (Figure 3). After displaying the frequencies of the currencies, it was decided to focus on the following: US Dollar, British Pound, Euro, Canadian

Dollar, and Australian Dollar. The rest of the currencies were grouped into, ‘Other’. Figure 4 shows the spread of the final currency types.

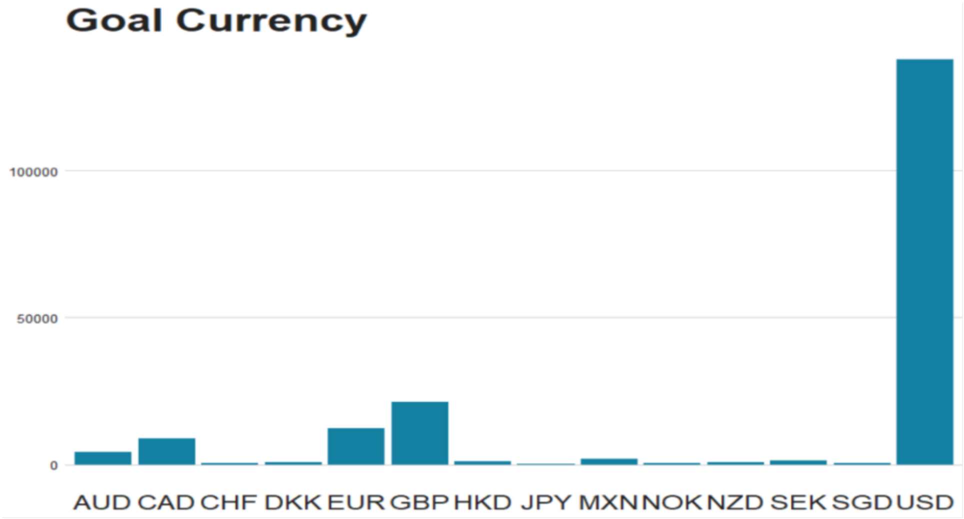
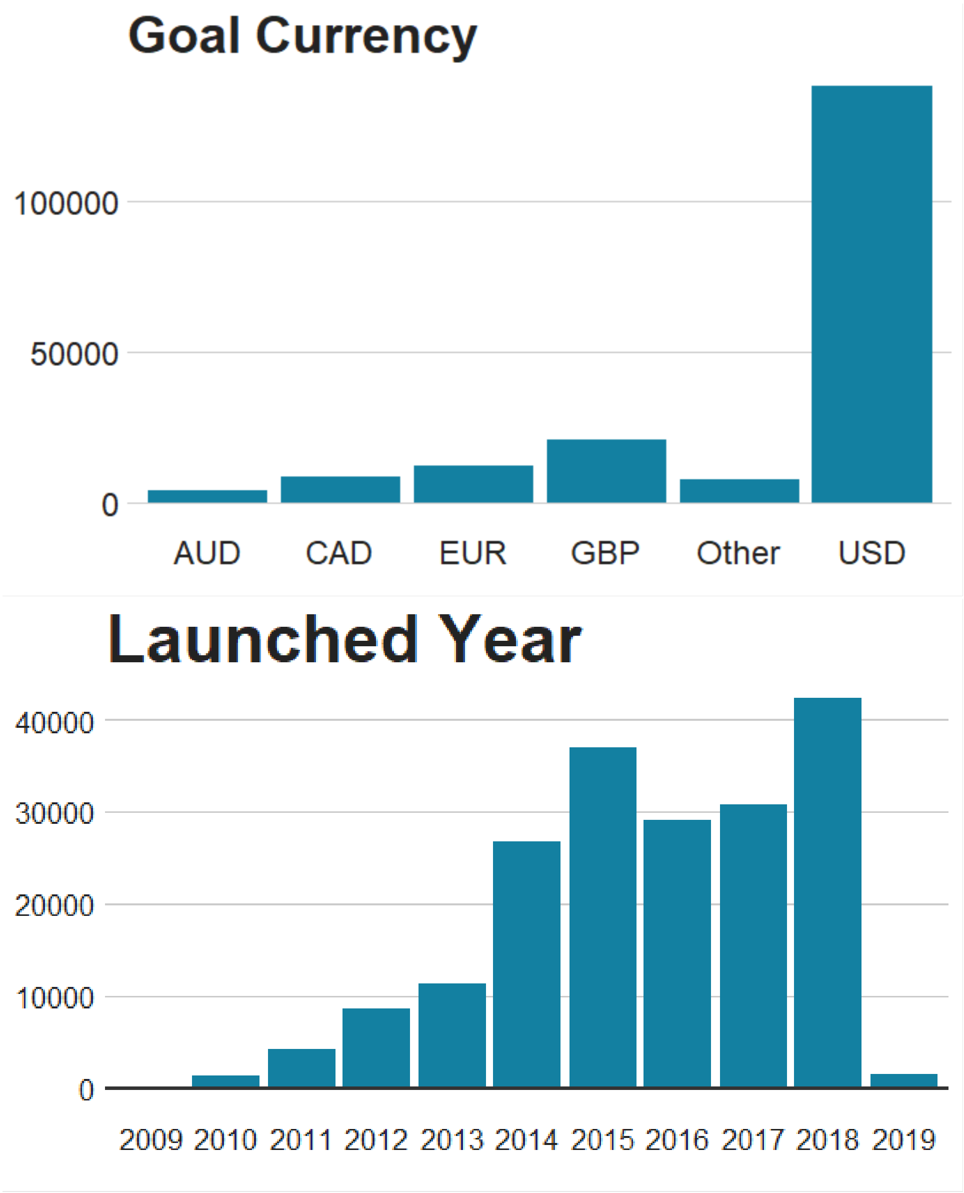


Figure 3

Figure 4

Finally, breakdown of complex date variables was performed into individual columns for month, year, and day. This decision came about mainly out of errors that were occurring in the later analysis due to the date-time format. It is beneficial to reduce the number of levels from 10 to 8. Looking at the spread of years (Figure 5), a small number of projects were launched before 2012. These projects were placed into a category called “Before 2012”. Figure 6 shows this result.



Goal Currency

Launched Year

Figure 5

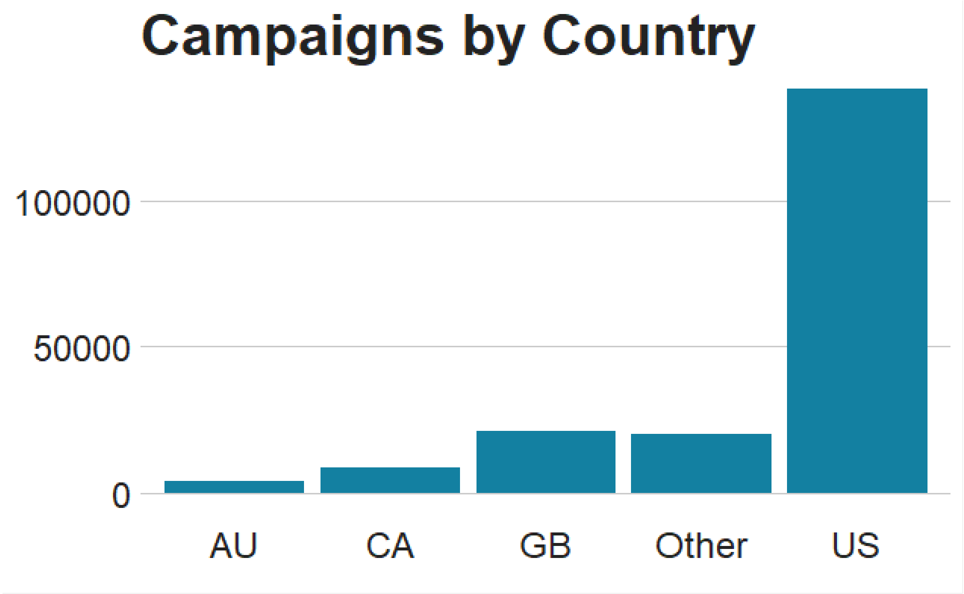


Figure 6

The cleaned dataset now has 192 during the analysis stage and converted blurb_length and name_length were converted to numeric data types.

Data Overview Conclusions

The initial data findings are based on overall quality of data, initial data cleansing, transformation and preparing subsets of data. During the first assessment it was determined that the entire dataset contained high quality data with only few missing values that are potentially absent due to a lack of data at the time of data collection. Column names were easily understood and did not require renaming. Next step was to identify the type of data for each variable, discrete or continuous, as discrete variables can be a detrimental factor in further statistical analysis. Dataset's structure was investigated using str() function, some columns were consolidated, and subsets of data were created for the analysis.

Analysis & Models – Descriptive Analysis

Descriptive Statistics

The descriptive analysis provides the first glimpse of the data. First, the response variable, status, was plotted to reflect the campaign status. This variable indicates which projects Kickstarter deemed to be successful. Out of the 192548 campaigns, 117307 were successful while 75241 were unsuccessful. Figure 7 shows the distribution below.

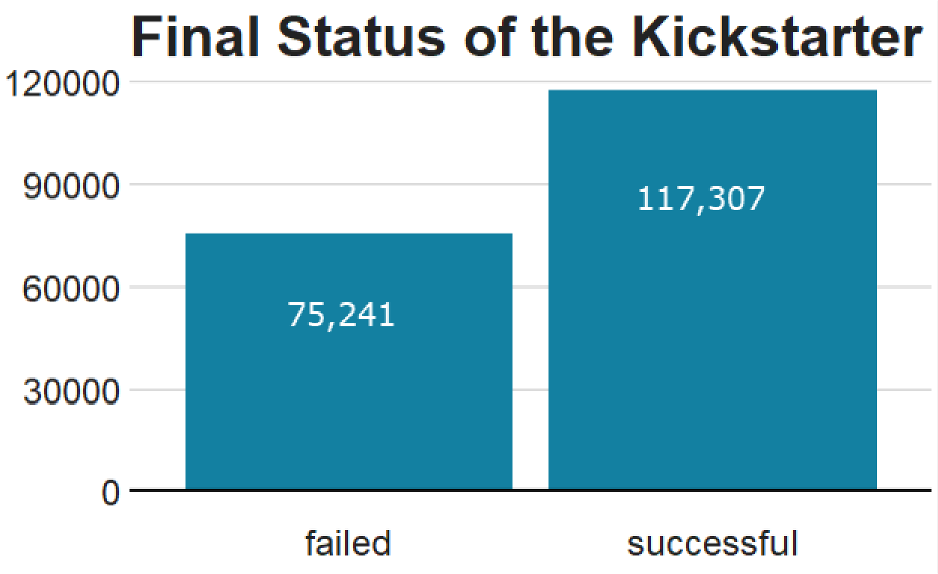


Figure 7

It is beneficial to see what the distribution of categories looks like, and what was the number of successful/failed projects within it. Figure 8 was created to show the categories with the highest percentage of projects, and then the status of the projects within these categories was also plotted.

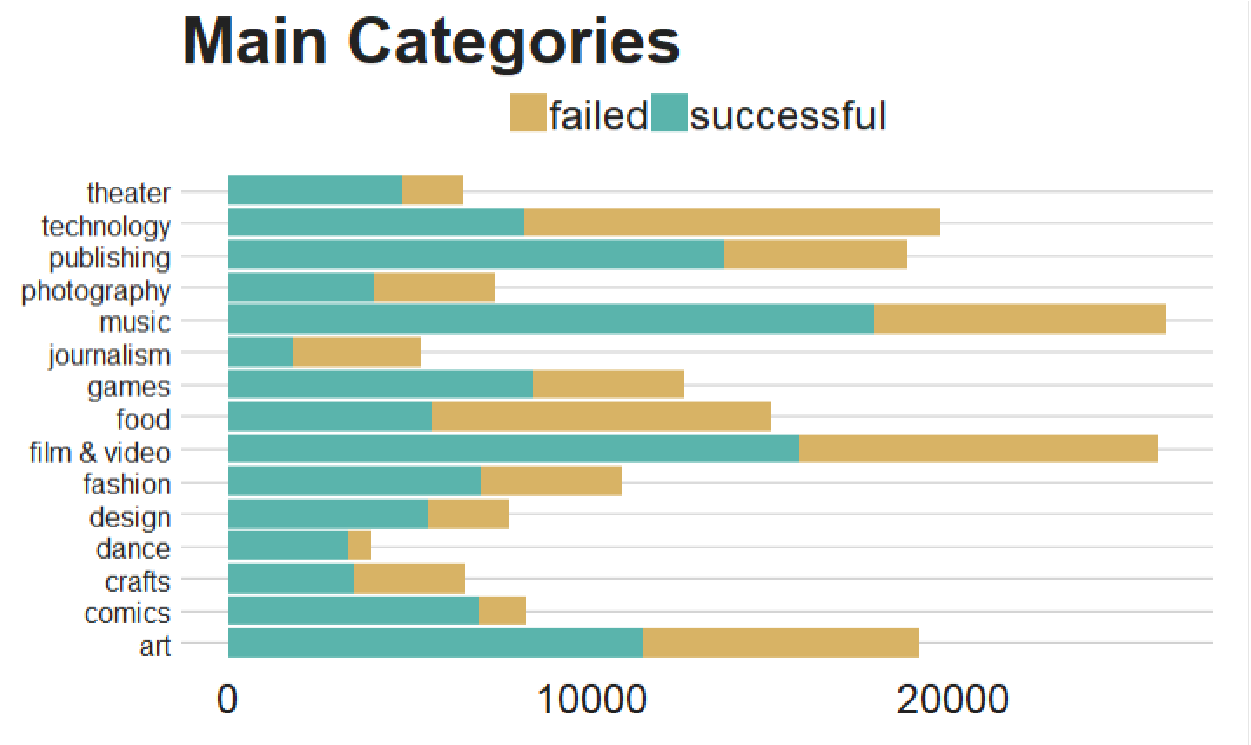


Figure 8

Next, the histogram of project status within each country was created. Since Figure 4 already showed the number of projects within each country, Figure 9 shows the status of projects by country.

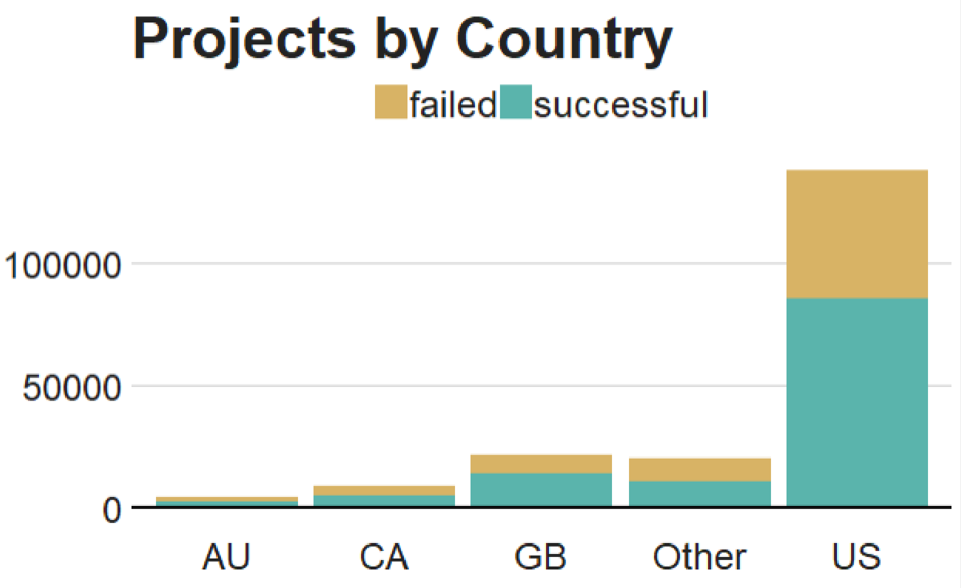


Figure 9

It is vital to investigate the duration of the campaigns more in-depth. The graph to the left in Figure 10 plots the length of campaigns the status, whether it was successful or not. Additionally, boxplot on the right shows the active duration of the campaigns based on the status, which helps to understand the mean and outliers of the duration.

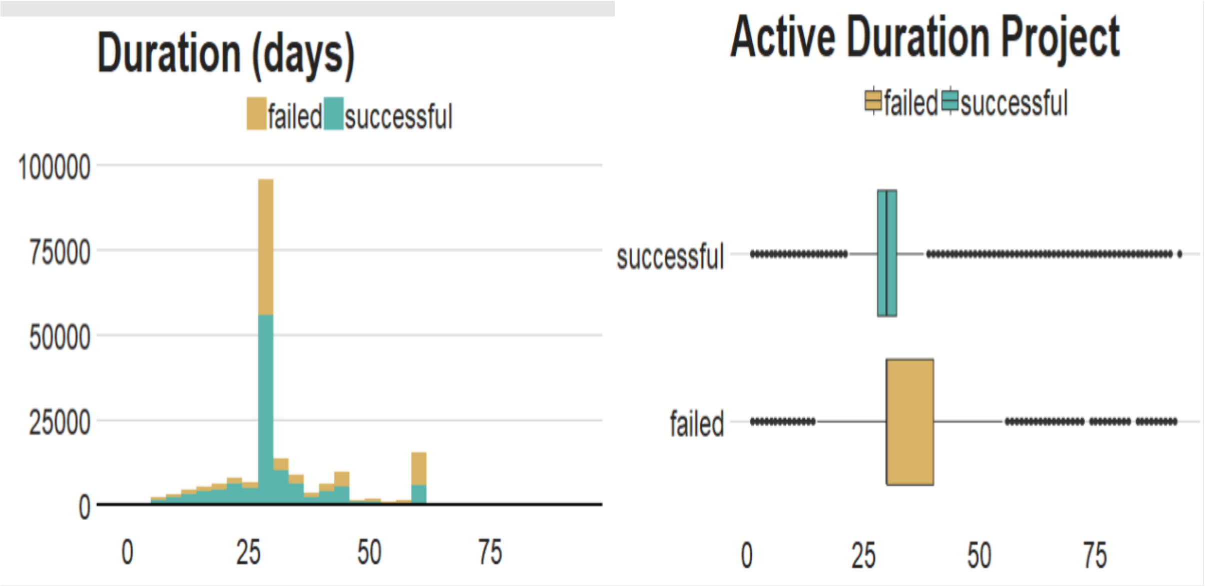


Figure 10

Going further, the goals that were set by the different campaigns were analyzed to get a better sense of the distribution and the outliers. Unlike the duration analysis, the log transformation of the goal was performed to understand the distribution better. Figure 11 shows this a density plot and boxplot.

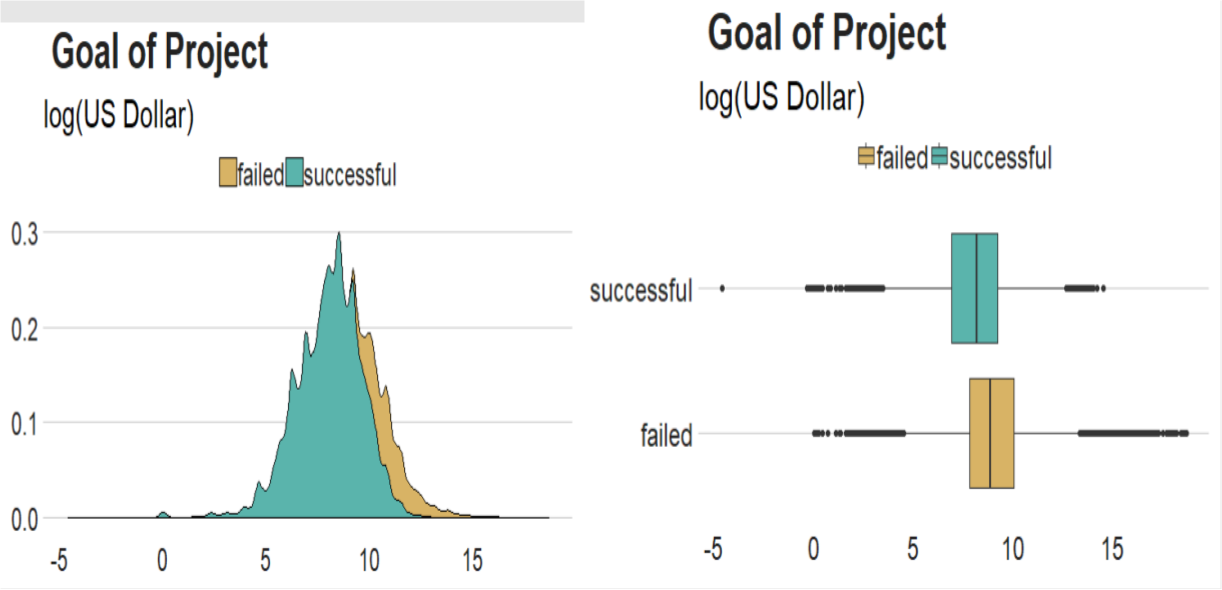


Figure 7

Descriptive Analytics Conclusion

Descriptive statistics provide the first glimpse into the Kickstarter data and simply describe what is or what the data shows. It is interesting to notice that majority of Kickstarter campaigns originate in the United States and in general there are more successful campaigns than failed. The most popular categories are in Music and Film & Video with higher percentage of succeeded campaigns. Majority of the successful campaigns have around 30 days duration period.

Analysis & Models – Predictive Analysis

The next stage of analysis is predicting the success of the campaigns. Predictive models were created through the use of logistic regression, classification trees, clustering, and association rules mining.

Logistic Regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous or binary such as “status” in the Kickstarter’s data. It is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Logistic regression major assumptions:

- The dependent variable should be binary.
- There should be no outliers in the data.
- There should be no high correlations (multicollinearity) among the predictors.

The goal of the logistic regression analysis is to estimate the log odds of an event.

The first step required the creation of training and test data. The seed was set to 460000 and split it 60% towards training , 20% towards testing, and 20% towards validation. Next, the regression model was built, where “status” was assigned as the response variable against six of the following other variables:

- main_category
- launched_year
- duration
- goals_usd
- blurb_length
- and name_length

The model’s output will be discussed further in the results section.

Next, the parameter, pcut, was tuned. This is the optimal cut-off probability for classifying projects into the successful or failure class. For this parameter, a three-fold cross-validation method was used on the training data. Using a symmetric cost function for wrongly classified KS projects, the optimal cut-off probability was first calculated.

The optimal cut-off probability is the probability at which the logistic regression model has the least misclassification rate. Therefore, the optimal cut-off probability was plotted to find the probability with the least misclassification error. 0.54 was chosen as the optimal cut-off probability with a CV cost of 0.34.

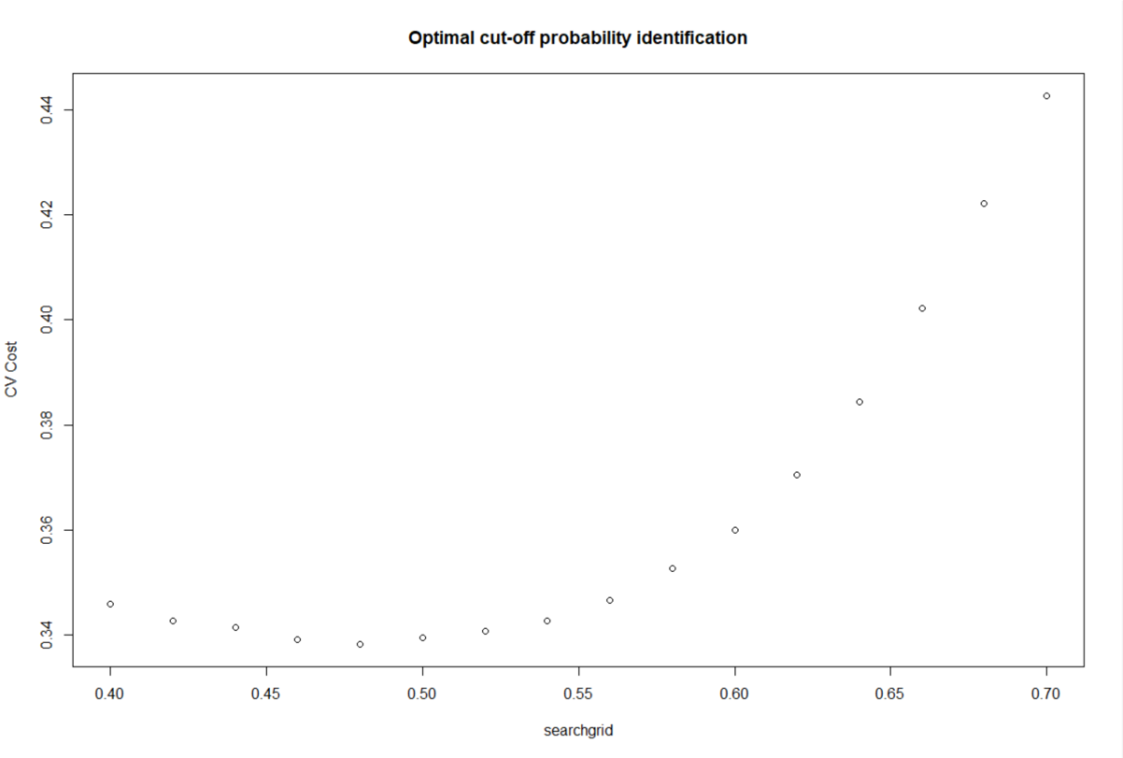
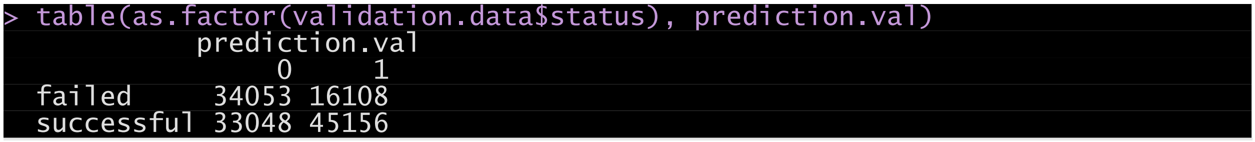
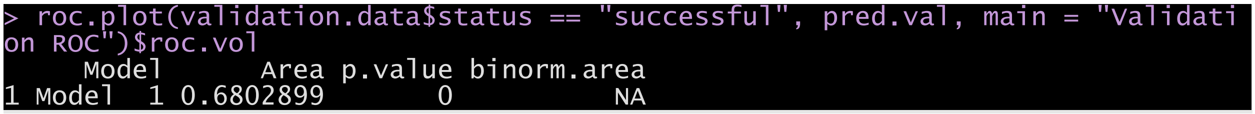


Figure 12

The prediction of successful or failed status is tabled below. In addition, a ROC curve was plotted to show the predictive threshbold of our binary classifer model for training and valdation model. Additionally, the area under the curve was tabled below to show models accuracy rate. It has a 68% accuracy, which is ok.



```
> table(as.factor(validation.data$status), prediction.val)
            prediction.val
                 0     1
  failed     34053 16108
  successful 33048 45156
```



```
> roc.plot(validation.data$status == "successful", pred.val, main = "Validati
on ROC")$roc.vol
     Model      Area p.value binorm.area
1 Model  1 0.6802899       0          NA
```

Classification Tree

Decision tree is a supervised learning predictive model that uses a set of binary rules to determine the output value. The decision rules generated by the algorithm are visualized as a binary tree. Decision trees work by finding the variable that best splits the outcome into two groups. The recursion stops when all groups are sufficiently small. Decision tree model have three components:

- Root node – the node that performs the first split.
- Terminal node – nodes that predict the outcome.
- Branches – representing flow from the question to answer.

Decision tree model was built as the next predictive analysis method. The tree was built on the binary response variable “status”. The data for the tree was split based on the predictor variables. The below image shows the separation between failed and successful campaigns.

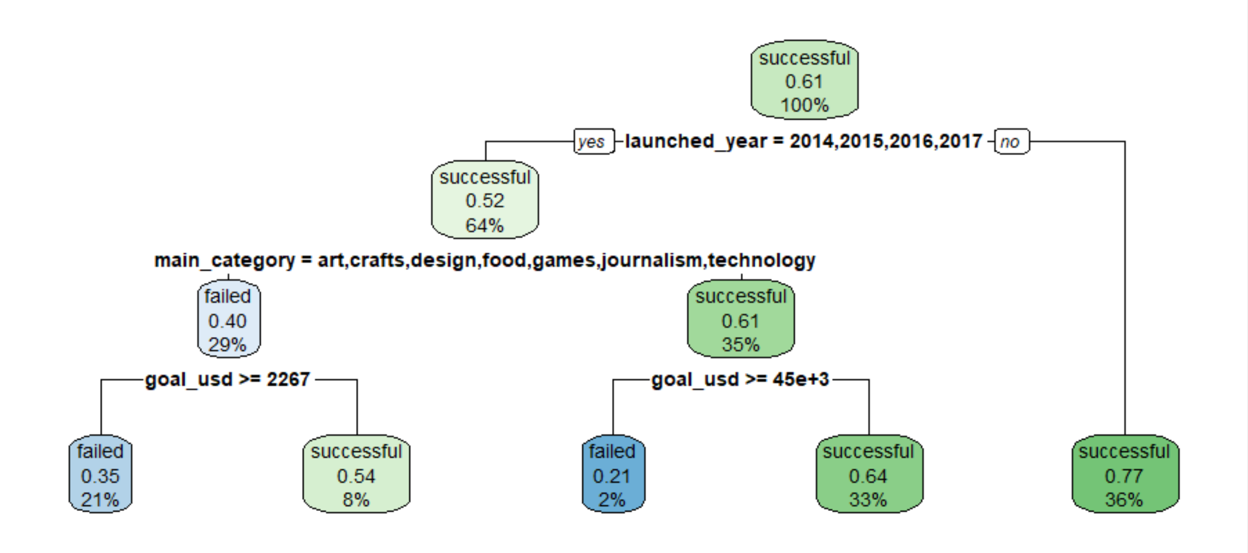


Figure 13

Additionally, the relative error was plotted to better understand the accuracy of the model. The goal during inspection was to find the optimal model by using the plotcp table as the measure by depicting the deviation until the minimum value is reached. This was done by adjusting the rpart control parameters to plot the cp value against the geometric mean.

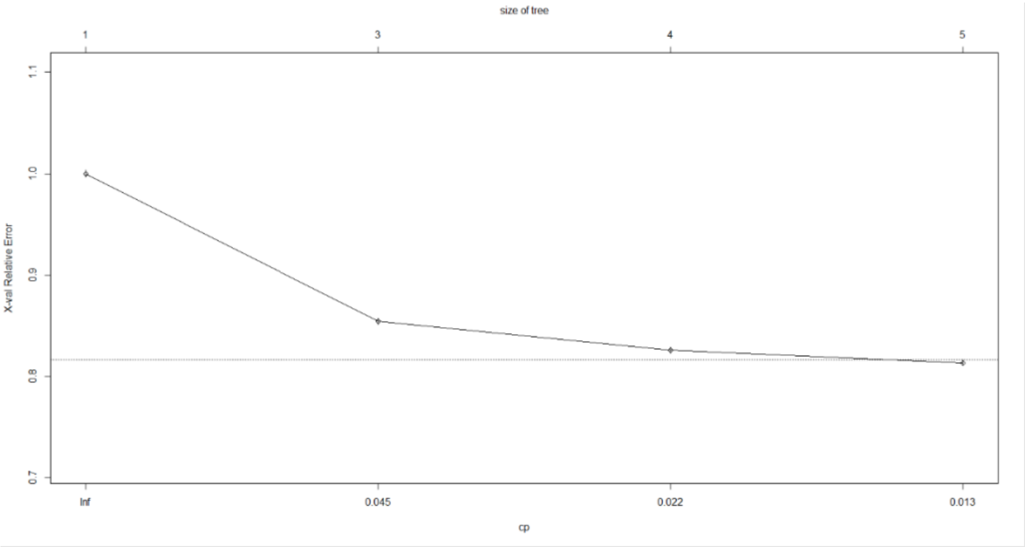


Figure 14

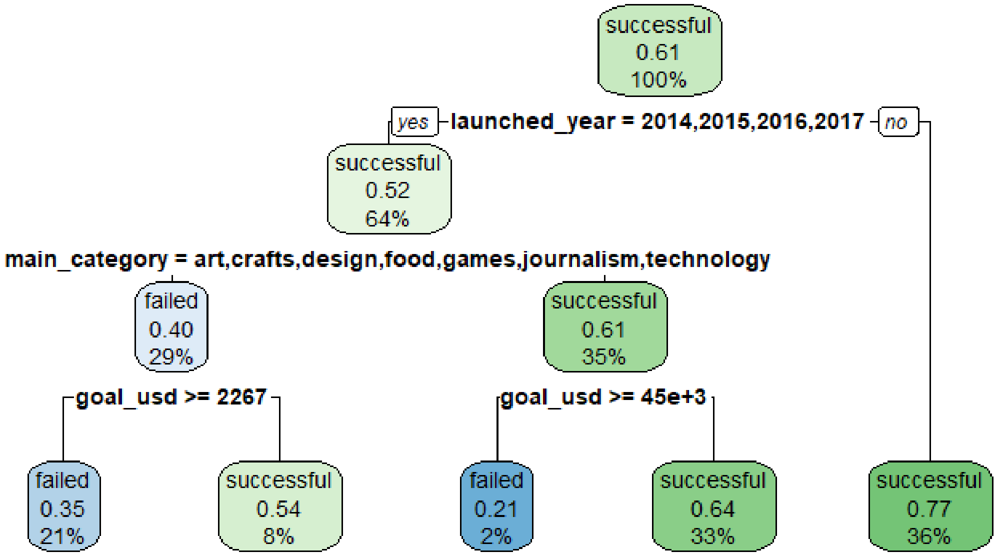


Figure 15

Notice from the cp plot that the Tuned Model 1 (above) did not change from the original decision tree model. Therefore, the parameters in the second tuned model were adjusted for primarily maxdepth. The min-split was kept the same as the previous model's value, which was 10 . The second models the maxdepth was increased to 5. The inspection of the second tuned model shows that the relative error does not change. It appears that the original model was tuned as best as it could be.

Because of this, ROC curve was plotted and used to access the model through the help of a contingency table and the tabled area under the curve. Through this, an error rate of around 36.38% (since the area under the curve is 0.6362) was determined. This accuracy rate is similar to the results of our logit regression model.

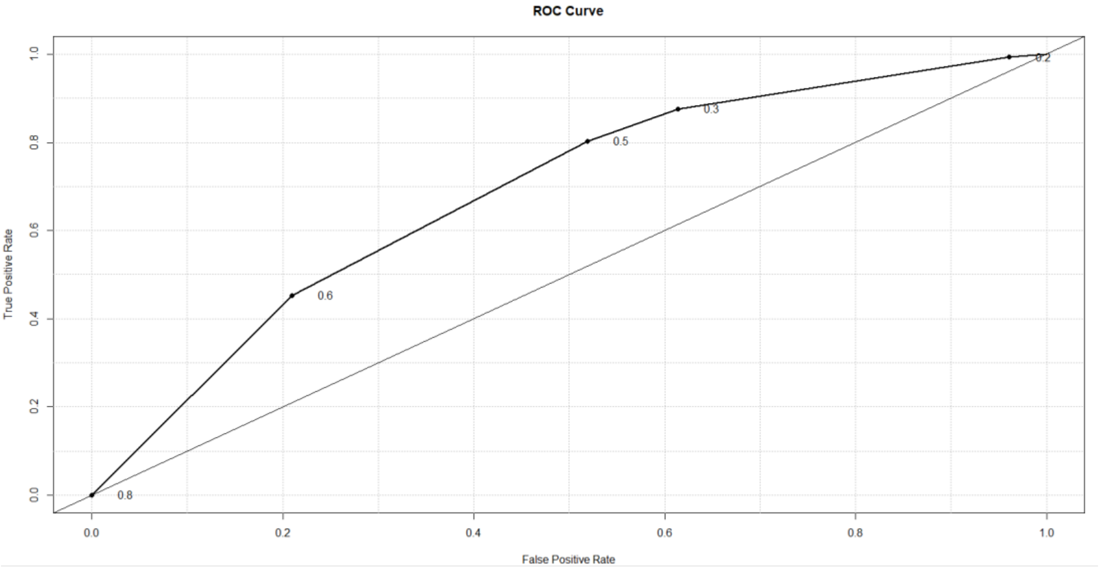
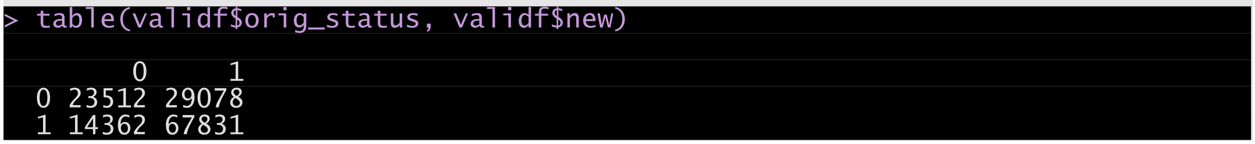
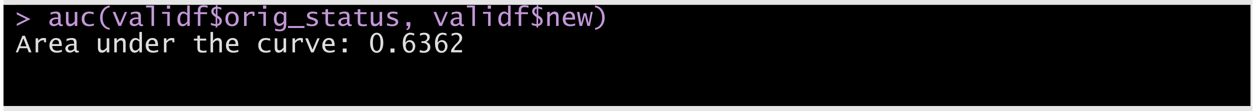


Figure 16



```
> table(validf$orig_status, validf$new)

        0     1
  0 23512 29078
  1 14362 67831
```



```
 > auc(validf$orig_status, validf$new)
 Area under the curve: 0.6362
```

Other Decision tree models

A decision tree was created using the J48 algorithm. The attributes chosen were the goal in US dollars, main category, subcategory, duration, name length, blurb length, and country of the campaign. Pruning was enabled with a minimum confidence level of 0.5. The model had a cross-validation factor of three and had a 66%/34% train/test split.

The gain ratio for the selected features is below.

```
Ranked attributes:
 0.04122   4 sub_category
 0.01622   5 goal_usd
 0.01464   1 main_category
 0.01144   3 duration
 0.00572   7 name_length
 0.00281   2 country
 0.00141   6 blurb_length
```

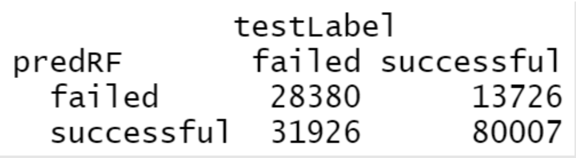
The resultant tree has 2,858 leaves, far too many to visualize. Looking at the confusion matrix, the decision tree has an overall accuracy of 76.4%

```
=== Summary ===

Correctly Classified Instances      50036              76.4305 %
Incorrectly Classified Instances     15430              23.5695 %
Total Number of Instances           65466

=== Confusion Matrix ===

     a     b   <-- classified as
 17423  8195 |     a = failed
  7235 32613 |     b = successful
```

Random Forest Model

Random forest has a constraint of 53 categories for any one variable. This caused sub-category to not be considered, resulting in the following variables: goal in US dollars, main category, duration, name length, blurb length, and country of the campaign. The dataset was shuffled by randomizing the row order using a seed of 2001.

The number of decision trees contained in the forest is 500, with a mean tree size of 711. There was a normal distribution in the tree size across the random forest. This model resulted in a 70.3% accuracy rate.



```
            testLabel
predRF        failed successful
  failed       28380      13726
  successful   31926      80007
```

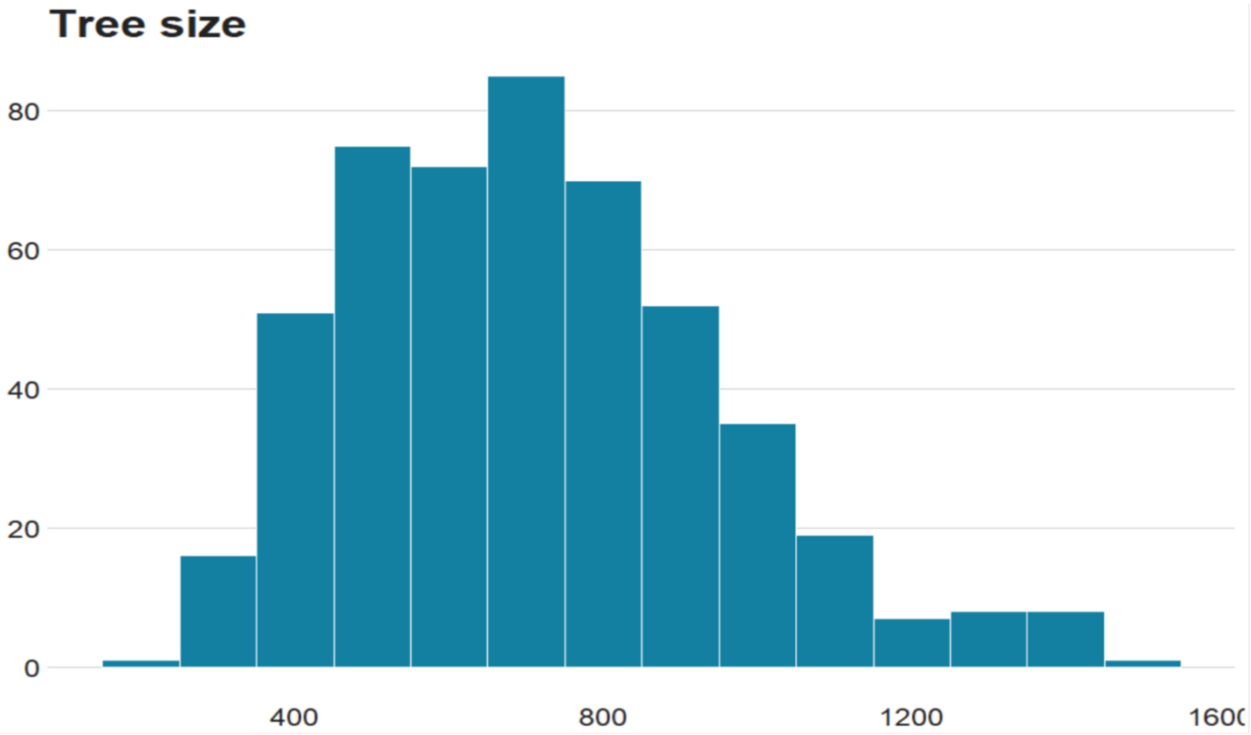


Figure 17

Results of Logistic Regression Model and Classification Tree

Through the logistic regression model, it is apparent that the main categories play a major role in project success. In addition, the results of the summary output (see appendix) of the model, showed that for every unit of change in the following categories , the log odds of success increased the most:

- Design
- Film & Video
- Games
- Music
- Publishing
- Theatre

However, in the following categories, for every one unit of change, the log odds of failure increase the most:

- Crafts
- Food
- Journalism
- Technology

- Photography

These outputs coincide with those of the classification tree and the indicators of what was suggested through the descriptive statistics analysis. The classification tree specifically showed the highest success rate for campaigns that had the following attributes:

- Launched after year 2014
- Is not in the following categories: art, craft, design, food, game, journalism, or technology
- Has a USD Goal of less than $45,000

Additionally, because the accuracy of the classification tree and the logistic regression model were conducted on the same binary response variable, “status”, a ROC comparison curve was created (see curve below). By doing so, it further demonstrated that the performance of the logistic model was barely better than that of the classification tree. Overall, both models had similar results. Besides supporting the accuracy and models conclusions, the similar results of both models’ ROC curve supports the data’s reproducibility. Therefore, we can say with confidence that these two models can be used in conjunction when determining our final results.

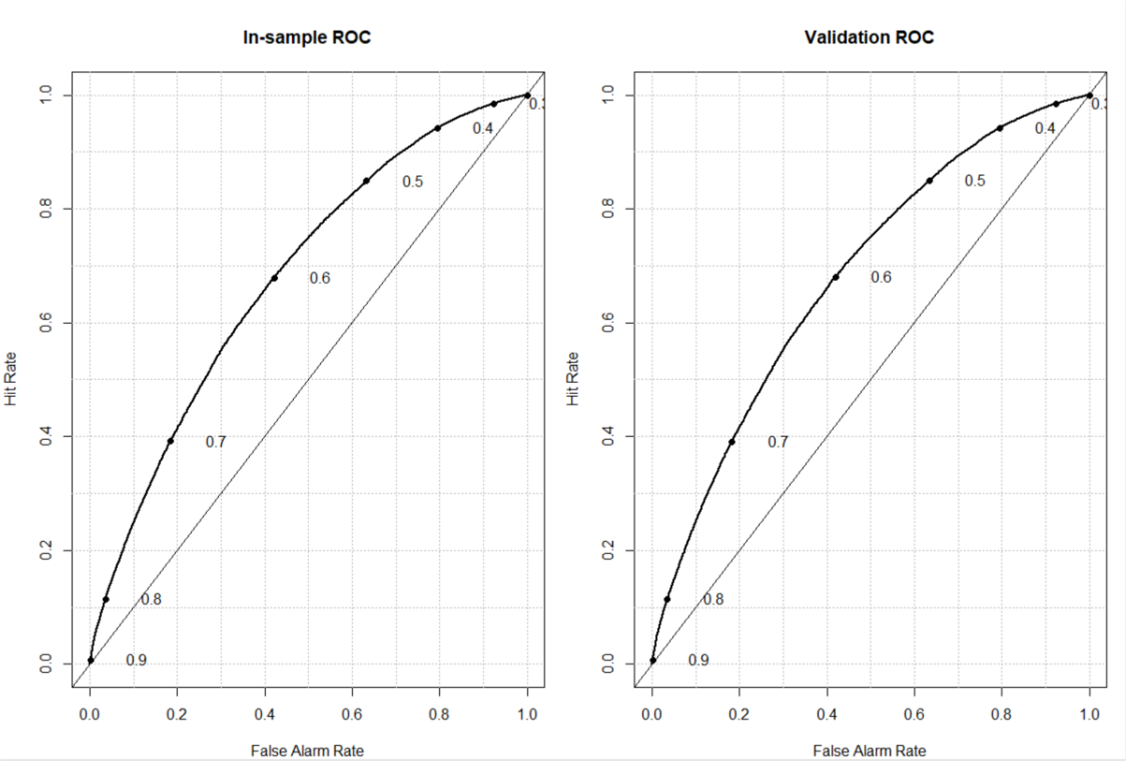


Figure 18

Association Rules Mining

Apriori Algorithm & Important Metrics

Association analysis helps to discover patterns and interesting relationships in large sets of data. These relationships can be represented as a set of frequent rules, where features frequently occur together or correlated. The goal of associated rule mining is to find associations of items that occur together more often than it is expected from a random sampling. The algorithm is used with categorical non-numeric data. There are three essential measures of the effectiveness of the rule:

Support – measures how much historical data supports the rule, and it is calculated as the joint probability of features: A and B when finding support for these two items together.

Confidence – measures a fraction of rows containing B or conditional probability of feature B given A.

Lift – measures ratio Confidence is to Support. Lift is expected to be greater than 1, meaning that features A and B are positively correlated.

Apriori algorithm is used to run association rule mining. "Apriori uses an iterative approach known as level-wise search. Basically, at the first level, we have one item sets (i.e., individual items such as bread or milk), which are frequently found (i.e., have sufficient support, as defined in the apriori command). At the next level, the two-item sets we need to consider must have the property that each of their subsets must be frequent enough to include (i.e., they have sufficient support). The algorithm keeps going up levels until there is no more analysis to do." (Saltz, 2018).

Pros of the Apriori algorithm: It is easy to implement and used on large datasets.

Cons of the Apriori algorithm: It may need to find a large number of candidate rules which might be computationally expensive.

Data transformation

Data transformation is the process of converting data from one format or structure into another, including converting data types, cleaning missing values and duplicate data, performing aggregation, and other step depending on the project. Association rule mining requires numerical values to be transformed into categorical values, which is accomplished with discretization. Transformation of continuous numeric variables to categorical attributes involves deciding on how many categories to have to map values into these categories.

Following variables were discretized: duration was discretized into 3 groups, goal_usd was discretized into 4 groups, and usd_pledged was discretized into 4 groups. Start_month and End_month variables were converted into factors. A total of 14 variables were used to run Association Rules mining analysis:

currency
main_category
sub_category
duration
goal_usd
city
state
country
status
start_month
end_month
start_Q
end_Q
usd_pledged

Association Rules Mining Results

The purpose is to find patterns and relationships between campaign success and other variables such as country, duration, and goals. To find patterns in the current data, it is vital to find associations between different variables in the dataset (left-hand-side) and PEP (right-hand-side) or how likely that a Kickstarter campaign success is associated with other attributes.

One of the most important metrics to consider is Lift. “Lift takes into account the support for a rule, but also gives more weight to rules where the LHS and/or the RHS occurs less frequently. In other words, lift favors situations where LHS and RHS are not abundant, but where the relatively few occurrences always happen together. The larger the value of lift, the more interesting the rule may be.” (Saltz, 2018)

Top 5 rules sorted by Lift presented in the table below:

Table 2. Associated Rules Mining Output

Left Hand Side	**Right Hand Side**	**Support**	**Confidence**	**Lift**
currency=USD,duration=1	status=successful	0.3279709	0.6388692	1.050937
duration=1,country=US	status=successful	0.3279709	0.6388692	1.050937

duration=1	status=successful	0.4499041	0.6287040	1.034215
currency=USD,goal_usd=1	status=successful	0.4425041	0.6186199	1.017627
currency=USD	status=successful	0.4425041	0.6185207	1.017464

Figure 11 represents 31 rules created by apriori algorithm with minimum support parameter = 0.2 and confidence = 0.6.

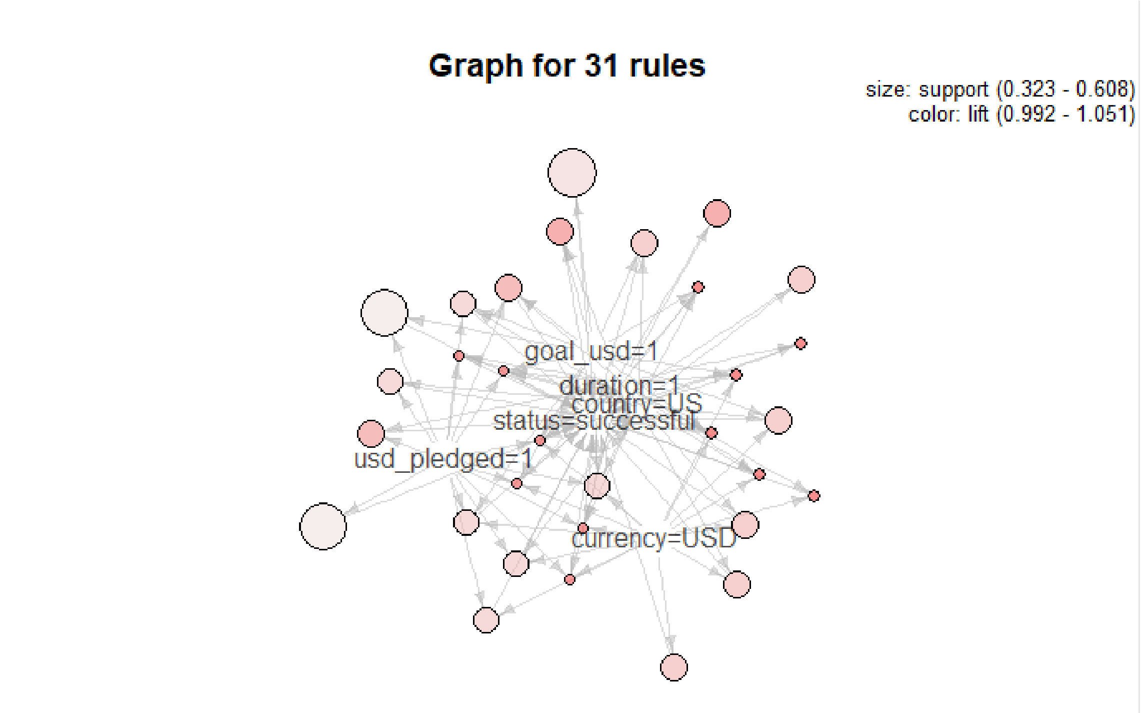


Figure 19

In conclusion, of the most critical parameters to consider in association rules mining is lift, which is expected to have value more than one meaning, is that features are positively correlated. The output of Apriori algorithm was sorted by value of Lift parameter. Success of each Kickstarter campaign is primarily associated with United States and US Dollar currency as well as shorter duration and smaller amounts that are pledged.

Clustering Analysis

Clustering analysis is a type of unsupervised learning by dividing data points into several groups with similar characteristics. In other words, clustering analysis creates a group of objects based on similarity.

There is no good criteria or rule for the clustering analysis, and it depends on the data and analysis objectives. One of the popular clustering methods is a partitioning method – partition objects into K number of clusters, where the distance is a major parameter. The key component is to be able to determine the number of clusters.

K-Means Clustering Algorithm

In K-Means clustering algorithm partitions data points into K clusters. Each cluster is represented by its centroid, which corresponds to the mean of data points assigned to each cluster. The number of clusters has to be determined prior to the analysis and has to be specified as a parameter. The basic idea behind partitioning methods, such as k-means clustering, is to define clusters such that the total intra-cluster variation or total within-cluster sum of square errors (SSE) is minimal. The Elbow method looks at the total sum of square errors as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't improve much better the total SSE. In this project, the optimal number of clusters is 4 determined by values of SSE.

The output of K means clustering is represented in Table 3:

Within cluster sum of squared errors: 559858.2754369406

Table 3. Cluster Output

Attribute	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**
Main category	film & video	music	music	technology
Subcategory	Shorts	Indie Rock	Rock	Web
Duration	30	30	31	41
Goal	31K	20K	19K	80K
Country	US	US	US	US
Start Quarter	Q2	Q4	Q3	Q3
End Quarter	Q2	Q4	Q3	Q4
Status	successful	successful	successful	failed

Figure 13 represents plotted Clusters: Status (X) versus Goal in USD (Y). The cluster placements show that successful campaigns were set with smaller goals while failed campaign's goals varied from smaller to much higher amounts.

Figure 20

K-means clustering results

K-Means clustering is a good option for finding similarities between various campaign attributes. The value of K was set to four. The results of clustering analysis confirmed the previously done results of logistic regression. Some major characteristics of successful Kickstarter campaigns are outlined below:

Successful campaigns that fall under Music and Film & Video categories.

- The average duration of 31-33 days.
- Average goals between 20K to 31K.
- Started and ended within the same Quarter.

Naïve Bayes Models

Naïve Bayes classifier is a probabilistic machine learning classifier. It assumes that the presence of a specific feature in a class is unrelated to the presence of any other feature.

Pros:

- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.

- It performs well in case of categorical input variables compared to numerical variables.

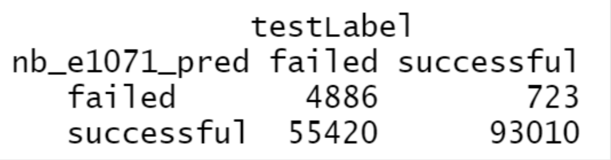
Cons:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”.
- Limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

A Naïve-Bayes model was created using with the following variables: goal in US dollars, main category, subcategory, duration, name length, blurb length, and country of the campaign. The data order of the data was shuffled by randomizing the order of the rows using a seed of 2001. The first 20% of rows were used to train the model, with the remaining 80% for testing. The campaign status was set aside from the test data to allow for future comparison.

Naïve Bayes Model Summary

The model was applied to the test data, then compared against the outcome that was previously set aside. The resultant confusion matrix had an overall accuracy of 63.6%, which is only slightly better than if it was assumed that every campaign would succeed (60.9%). This is what the model mostly did by predicting 96.6% of the campaigns would succeed.



```
               testLabel
nb_e1071_pred  failed successful
   failed        4886        723
   successful   55420      93010
```

Support Vector Machine (SVM) model

Support Vector Machines algorithm is a linear classifier that can solve linear separable and inseparable problems. The algorithm outputs an optimal hyperplane that classifies test examples. There are several kernels are available to build a prediction model. One of the important tasks to consider is C regularization parameter or “penalty” for allowing close to margin points.

The SVM model was built to have the default C regularization parameter of 1 in order to have a decent margin between the classifications, but not lose too much information in the process. Five folds for the k-fold cross-validation was chosen in training the model. This allows us to keep the model from overfitting during the training process.

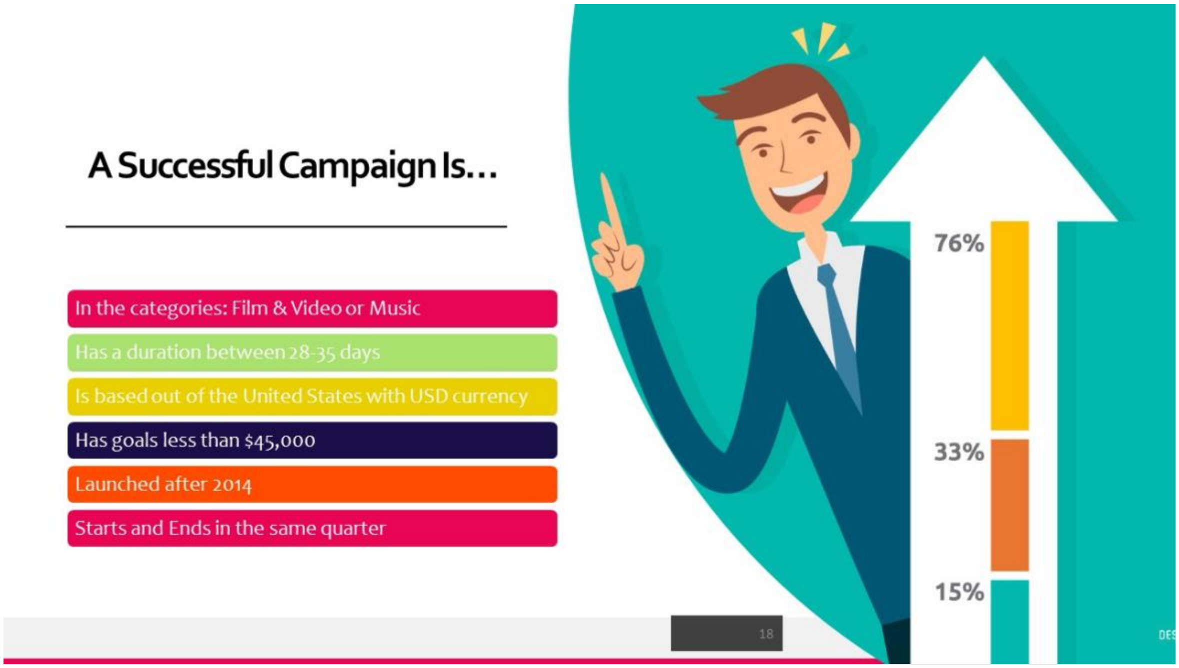
The model used the following variables: goal in US dollars, main category, subcategory, duration, name length, blurb length, and country of the campaign. The data order of the data was shuffled by randomizing the order of the rows using a seed of 2001. The first 20% of rows were used to train the model, with the remaining 80% for testing. The campaign status was set aside from the test data to allow for future comparison. The first attempt used a linear kernel to no avail, with the model predicting every campaign would fail. The polynomial kernel was attempted next, with degree of 3 and 4.

Support Vector Machine results

Using the polynomial kernel resulted in similar accuracy between degree 3 and 4, with a difference of 0.3%, however the separate models achieved this accuracy in different ways. The polynomial kernel with three degrees predicted success 98.9% of the time whereas a four-degree polynomial predicted success in 62.3% of the time. The overall accuracy using the three-degree polynomial and four-degree polynomial, respectively was 61.9% and 61.6%. These SVMs are not viable for determining the success of future Kickstart campaigns.

Conclusion

What does Kickstarter data tell its inventors? Recommendations are outlined below:



A Successful Campaign Is…

- In the categories: Film & Video or Music
- Has a duration between 28-35 days
- Is based out of the United States with USD currency
- Has goals less than $45,000
- Launched after 2014
- Starts and Ends in the same quarter

In conclusion, if someone wanted to understand their chance of having a successful Kickstarter campaign, like inventors Gleb and Igor, it would be useful for them to look at their chance of success from the Kickstarter data. Gleb and Igor are in two of the highest failing categories, technology, and food. It might be a good idea for them to look to other sources for help. Also, they may consider breaking up their Kickstarter into shorter projects with more reachable goals. Also, it is important to note that this Kickstarter dataset lacked information on the number of contributors that put money towards a campaign.

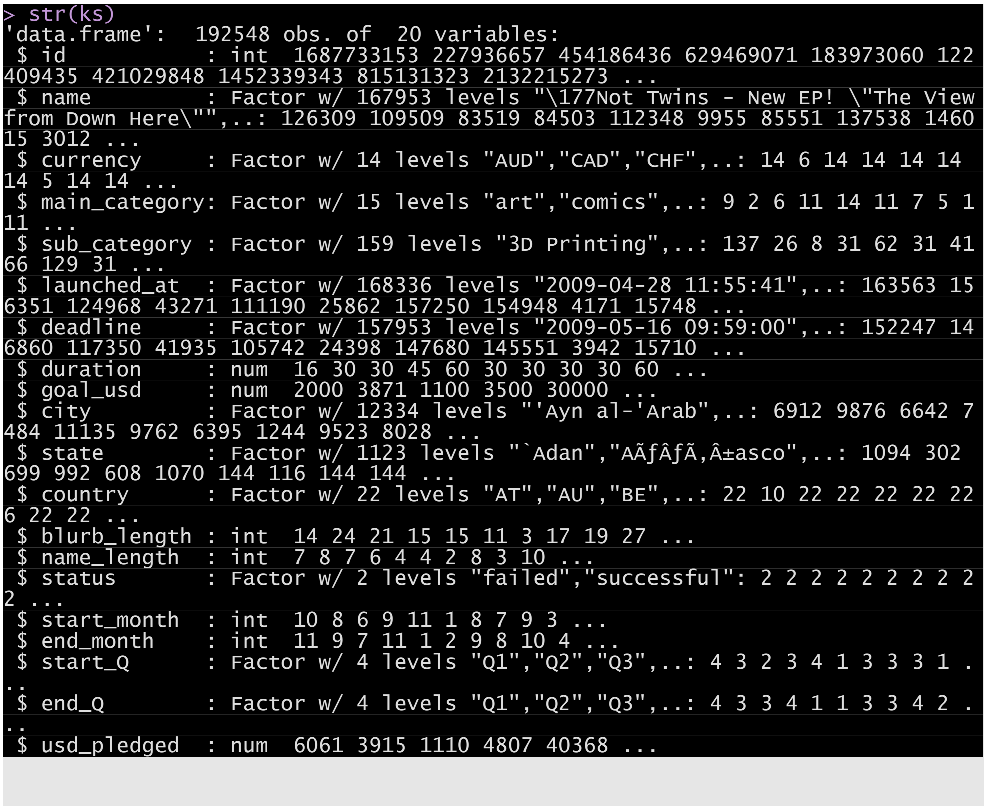
Having data at a finer grain would give us more insight into what makes a campaign successful. The sphere of influence is inherent when it comes to the success of a crowd-funded project. Ideally, we would have every instance of when this campaign was shared on social media, whether the post was public or private, and what the reach of each person is. This desired information is not new in the world of marketing, but it easier to quantify.

Having every individual donation in the dataset would also allow us to compare how much an initial launch impacted the overall success. How much does the number of backers at the very beginning of a campaign (three days or less) impact the overall success? Could this information be applied outside of the realm of Kickstarter, such as political campaigns? This could potentially lead to saving a good portion of time and money for campaigns that are doomed to fail.

Appendix I

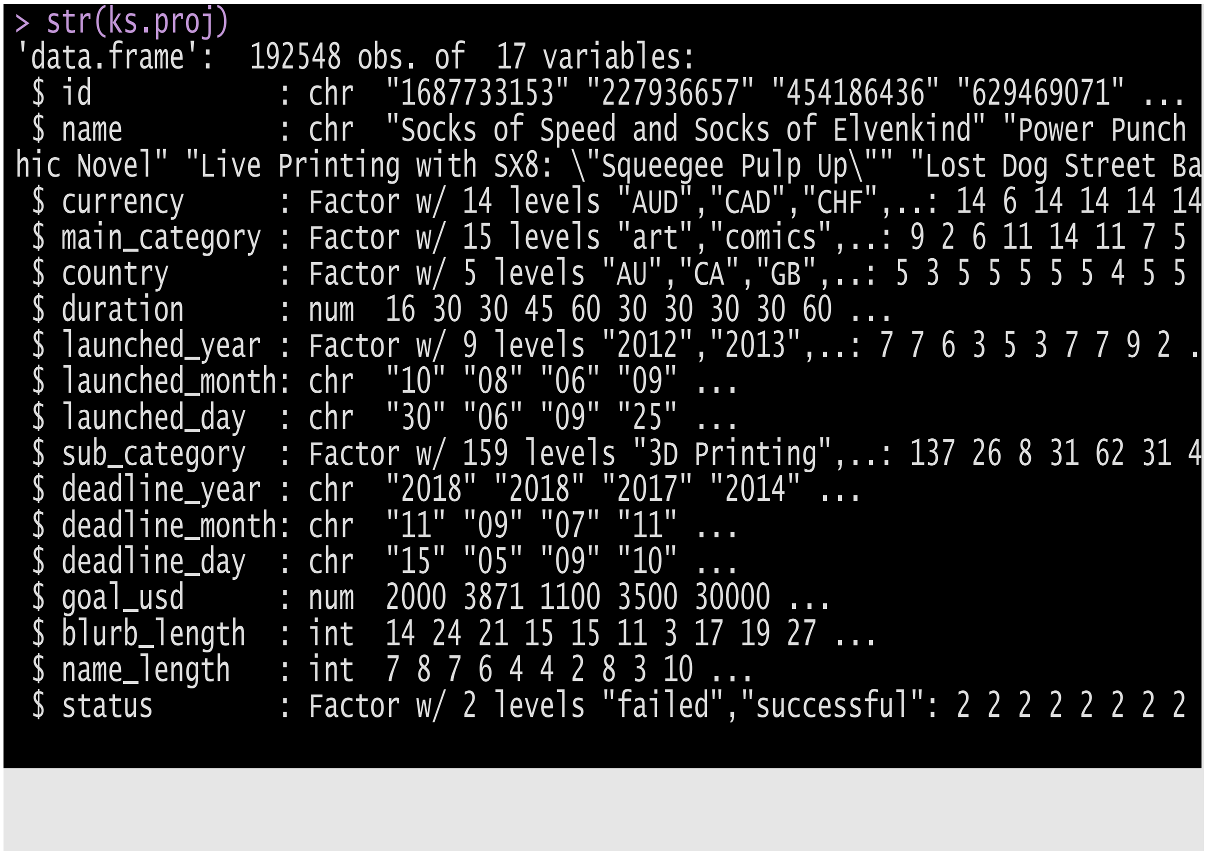
Data Cleaning Pre and Post Variables:

Our initial data prior cleaning stage looked like the following:



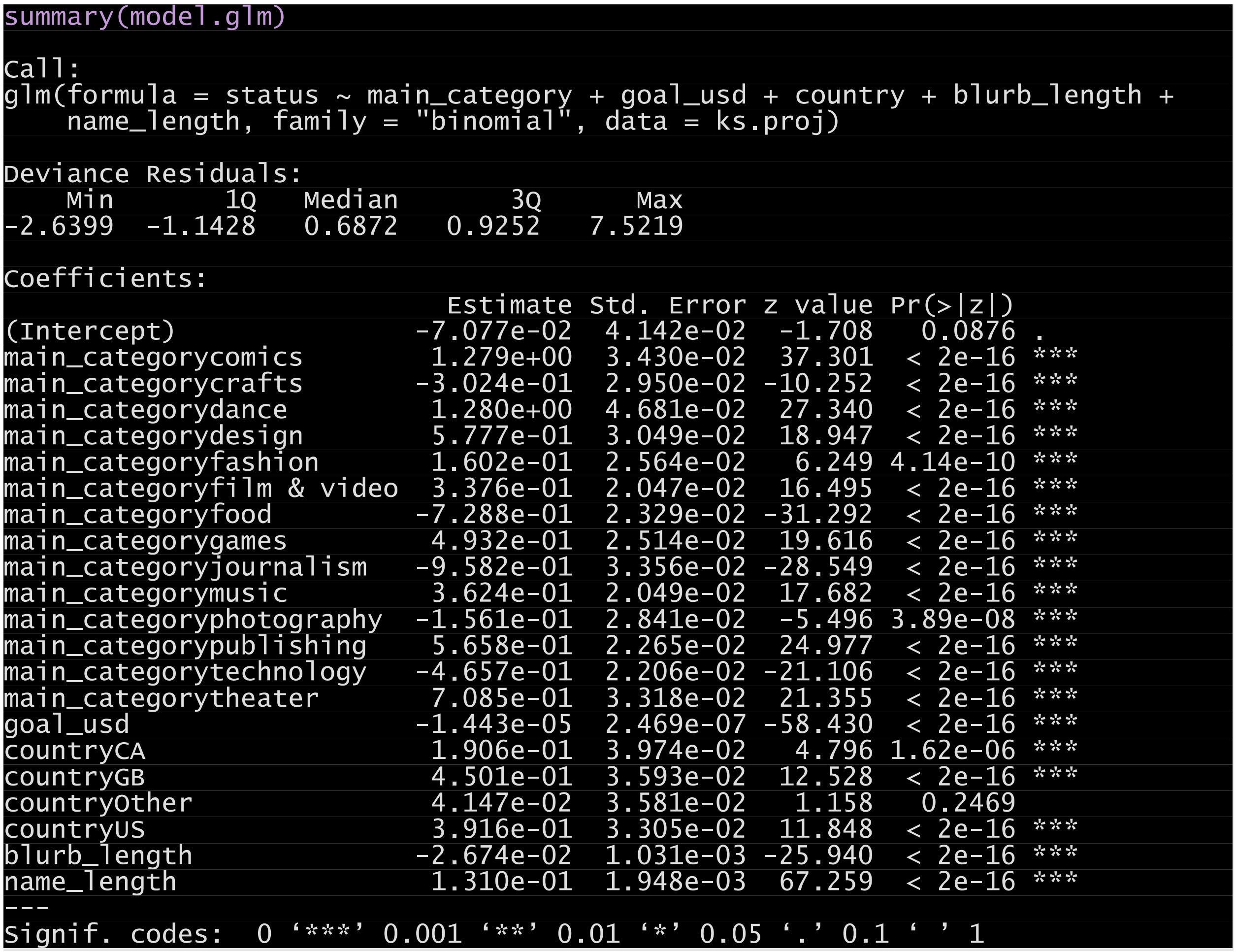
```
> str(ks)
'data.frame':	192548 obs. of  20 variables:
 $ id          : int  1687733153 227936657 454186436 629469071 183973060 122
409435 421029848 1452339343 815131323 2132215273 ...
 $ name        : Factor w/ 167953 levels "\177Not Twins - New EP! \"The View
from Down Here\"",..: 126309 109509 83519 84503 112348 9955 85551 137538 1460
15 3012 ...
 $ currency    : Factor w/ 14 levels "AUD","CAD","CHF",..: 14 6 14 14 14 14
14 5 14 14 ...
 $ main_category: Factor w/ 15 levels "art","comics",..: 9 2 6 11 14 11 7 5 1
11 ...
 $ sub_category : Factor w/ 159 levels "3D Printing",..: 137 26 8 31 62 31 41
66 129 31 ...
 $ launched_at  : Factor w/ 168336 levels "2009-04-28 11:55:41",..: 163563 15
6351 124968 43271 111190 25862 157250 154948 4171 15748 ...
 $ deadline     : Factor w/ 157953 levels "2009-05-16 09:59:00",..: 152247 14
6860 117350 41935 105742 24398 147680 145551 3942 15710 ...
 $ duration     : num  16 30 30 45 60 30 30 30 30 60 ...
 $ goal_usd     : num  2000 3871 1100 3500 30000 ...
 $ city         : Factor w/ 12334 levels "'Ayn al-'Arab",..: 6912 9876 6642 7
484 11135 9762 6395 1244 9523 8028 ...
 $ state        : Factor w/ 1123 levels "`Adan","AÃƒÂƒÃ‚Â±asco",..: 1094 302
699 992 608 1070 144 116 144 144 ...
 $ country      : Factor w/ 22 levels "AT","AU","BE",..: 22 10 22 22 22 22 22
6 22 22 ...
 $ blurb_length : int  14 24 21 15 15 11 3 17 19 27 ...
 $ name_length  : int  7 8 7 6 4 4 2 8 3 10 ...
 $ status       : Factor w/ 2 levels "failed","successful": 2 2 2 2 2 2 2 2 2
2 ...
 $ start_month  : int  10 8 6 9 11 1 8 7 9 3 ...
 $ end_month    : int  11 9 7 11 1 2 9 8 10 4 ...
 $ start_Q      : Factor w/ 4 levels "Q1","Q2","Q3",..: 4 3 2 3 4 1 3 3 3 1 .
..
 $ end_Q        : Factor w/ 4 levels "Q1","Q2","Q3",..: 4 3 3 4 1 1 3 3 4 2 .
..
 $ usd_pledged  : num  6061 3915 1110 4807 40368 ...
```

Post-cleaning, our data now looks like this:



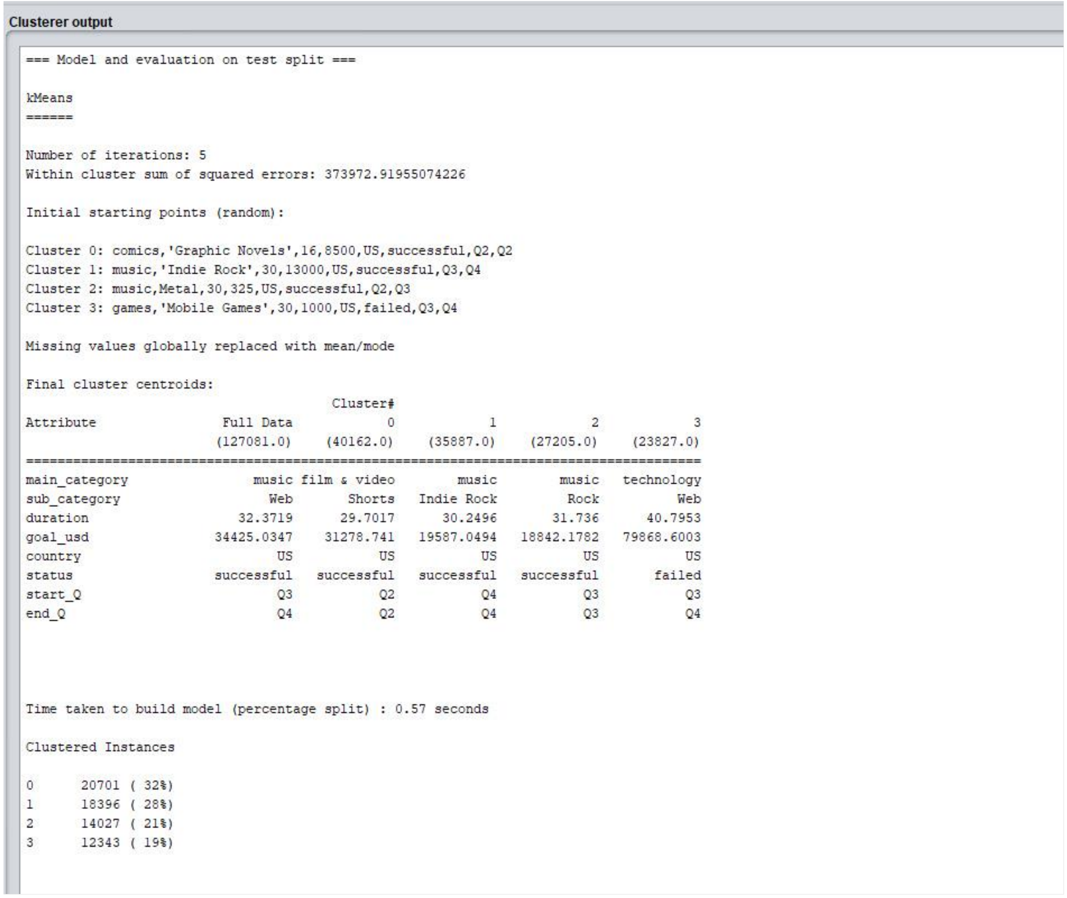
```
> str(ks.proj)
'data.frame':	192548 obs. of  17 variables:
 $ id            : chr  "1687733153" "227936657" "454186436" "629469071" ...
 $ name          : chr  "Socks of Speed and Socks of Elvenkind" "Power Punch
hic Novel" "Live Printing with SX8: \"Squeegee Pulp Up\"" "Lost Dog Street Ba
 $ currency      : Factor w/ 14 levels "AUD","CAD","CHF",..: 14 6 14 14 14 14
 $ main_category : Factor w/ 15 levels "art","comics",..: 9 2 6 11 14 11 7 5
 $ country       : Factor w/ 5 levels "AU","CA","GB",..: 5 3 5 5 5 5 5 4 5 5
 $ duration      : num  16 30 30 45 60 30 30 30 30 60 ...
 $ launched_year : Factor w/ 9 levels "2012","2013",..: 7 7 6 3 5 3 7 7 9 2 .
 $ launched_month: chr  "10" "08" "06" "09" ...
 $ launched_day  : chr  "30" "06" "09" "25" ...
 $ sub_category  : Factor w/ 159 levels "3D Printing",..: 137 26 8 31 62 31 4
 $ deadline_year : chr  "2018" "2018" "2017" "2014" ...
 $ deadline_month: chr  "11" "09" "07" "11" ...
 $ deadline_day  : chr  "15" "05" "09" "10" ...
 $ goal_usd      : num  2000 3871 1100 3500 30000 ...
 $ blurb_length  : int  14 24 21 15 15 11 3 17 19 27 ...
 $ name_length   : int  7 8 7 6 4 4 2 8 3 10 ...
 $ status        : Factor w/ 2 levels "failed","successful": 2 2 2 2 2 2 2 2
```

Summary of Logistic Regression Output.



```
summary(model.glm)

Call:
glm(formula = status ~ main_category + goal_usd + country + blurb_length + 
    name_length, family = "binomial", data = ks.proj)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.6399  -1.1428   0.6872   0.9252   7.5219  

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)    
(Intercept)               -7.077e-02  4.142e-02  -1.708   0.0876 .  
main_categorycomics        1.279e+00  3.430e-02  37.301  < 2e-16 ***
main_categorycrafts       -3.024e-01  2.950e-02 -10.252  < 2e-16 ***
main_categorydance         1.280e+00  4.681e-02  27.340  < 2e-16 ***
main_categorydesign        5.777e-01  3.049e-02  18.947  < 2e-16 ***
main_categoryfashion       1.602e-01  2.564e-02   6.249 4.14e-10 ***
main_categoryfilm & video  3.376e-01  2.047e-02  16.495  < 2e-16 ***
main_categoryfood         -7.288e-01  2.329e-02 -31.292  < 2e-16 ***
main_categorygames         4.932e-01  2.514e-02  19.616  < 2e-16 ***
main_categoryjournalism   -9.582e-01  3.356e-02 -28.549  < 2e-16 ***
main_categorymusic         3.624e-01  2.049e-02  17.682  < 2e-16 ***
main_categoryphotography  -1.561e-01  2.841e-02  -5.496 3.89e-08 ***
main_categorypublishing    5.658e-01  2.265e-02  24.977  < 2e-16 ***
main_categorytechnology   -4.657e-01  2.206e-02 -21.106  < 2e-16 ***
main_categorytheater       7.085e-01  3.318e-02  21.355  < 2e-16 ***
goal_usd                  -1.443e-05  2.469e-07 -58.430  < 2e-16 ***
countryCA                  1.906e-01  3.974e-02   4.796 1.62e-06 ***
countryGB                  4.501e-01  3.593e-02  12.528  < 2e-16 ***
countryOther               4.147e-02  3.581e-02   1.158   0.2469    
countryUS                  3.916e-01  3.305e-02  11.848  < 2e-16 ***
blurb_length              -2.674e-02  1.031e-03 -25.940  < 2e-16 ***
name_length                1.310e-01  1.948e-03  67.259  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

K-Means Clustering Output



```
=== Model and evaluation on test split ===

kMeans
======

Number of iterations: 5
Within cluster sum of squared errors: 373972.91955074226

Initial starting points (random):

Cluster 0: comics,'Graphic Novels',16,8500,US,successful,Q2,Q2
Cluster 1: music,'Indie Rock',30,13000,US,successful,Q3,Q4
Cluster 2: music,Metal,30,325,US,successful,Q2,Q3
Cluster 3: games,'Mobile Games',30,1000,US,failed,Q3,Q4

Missing values globally replaced with mean/mode

Final cluster centroids:
                                       Cluster#
Attribute                Full Data            0            1            2            3
                        (127081.0)    (40162.0)    (35887.0)    (27205.0)    (23827.0)
======================================================================================
main_category                music film & video        music        music   technology
sub_category                   Web       Shorts   Indie Rock         Rock          Web
duration                   32.3719      29.7017      30.2496       31.736      40.7953
goal_usd                34425.0347    31278.741   19587.0494   18842.1782   79868.6003
country                         US           US           US           US           US
status                  successful   successful   successful   successful       failed
start_Q                         Q3           Q2           Q4           Q3           Q3
end_Q                           Q4           Q2           Q4           Q3           Q4




Time taken to build model (percentage split) : 0.57 seconds

Clustered Instances

0      20701 ( 32%)
1      18396 ( 28%)
2      14027 ( 21%)
3      12343 ( 19%)
```