**Step1 – Data Pre-processing**

**a) Let us take a look at the input dataset :**

##

Read 68.7% of 378661 rows

Read 378661 rows and 14 (of 14) columns from 0.051 GB file in 00:00:03

## 'data.frame': 378661 obs. of 14 variables:

## $ ID : int 1000002330 1000003930 1000004038 1000007540 1000011046 1000014025 1000023410 1000030581 1000034518 100004195 ...

## $ name : chr "The Songs of Adelaide & Abullah" "Greeting From Earth: ZGAC Arts Capsule For ET" "Where is Hank?" "ToshiCapital Rekordz Needs Help to Complete Album" ...

## $ category : chr "Poetry" "Narrative Film" "Narrative Film" "Music" ...

## $ main\_category : chr "Publishing" "Film & Video" "Film & Video" "Music" ...

## $ currency : chr "GBP" "USD" "USD" "USD" ...

## $ deadline : chr "2015-10-09" "2017-11-01" "2013-02-26" "2012-04-16" ...

## $ goal : num 1000 30000 45000 5000 19500 50000 1000 25000 125000 65000 ...

## $ launched : chr "2015-08-11 12:12:28" "2017-09-02 04:43:57" "2013-01-12 00:20:50" "2012-03-17 03:24:11" ...

## $ pledged : num 0 2421 220 1 1283 ...

## $ state : chr "failed" "failed" "failed" "failed" ...

## $ backers : int 0 15 3 1 14 224 16 40 58 43 ...

## $ country : chr "GB" "US" "US" "US" ...

## $ usd.pledged : num 0 100 220 1 1283 ...

## $ usd\_pledged\_real: num 0 2421 220 1 1283 ...

The projects are divided into main and sub-categories. The pledged amount “usd\_pledged” has an equivalent value converted to USD, called “usd\_pledged\_real”. However, the goal amount does not have this conversion. So for now, we will use the amounts as is.

We can see how many people are backing each individual project using the column, “backers”.

**b) Now let us look at the first 5 records:**

The name doesn’t really indicate any specific pattern although it might be interesting to see if longer names have better success rates. Not pursuing that angle at this time, though.

## ID name

## 1 1000002330 The Songs of Adelaide & Abullah

## 2 1000003930 Greeting From Earth: ZGAC Arts Capsule For ET

## 3 1000004038 Where is Hank?

## 4 1000007540 ToshiCapital Rekordz Needs Help to Complete Album

## 5 1000011046 Community Film Project: The Art of Neighborhood Filmmaking

## 6 1000014025 Monarch Espresso Bar

## category main\_category currency deadline goal

## 1 Poetry Publishing GBP 2015-10-09 1000

## 2 Narrative Film Film & Video USD 2017-11-01 30000

## 3 Narrative Film Film & Video USD 2013-02-26 45000

## 4 Music Music USD 2012-04-16 5000

## 5 Film & Video Film & Video USD 2015-08-29 19500

## 6 Restaurants Food USD 2016-04-01 50000

## launched pledged state backers country usd.pledged

## 1 2015-08-11 12:12:28 0 failed 0 GB 0

## 2 2017-09-02 04:43:57 2421 failed 15 US 100

## 3 2013-01-12 00:20:50 220 failed [3 US 220](https://www.google.com/maps/search/3+US+220?entry=gmail&source=g)

## 4 2012-03-17 03:24:11 1 failed [1 US 1](https://www.google.com/maps/search/1+US+1?entry=gmail&source=g)

## 5 2015-07-04 08:35:03 1283 canceled 14 US 1283

## 6 2016-02-26 13:38:27 52375 successful 224 US 52375

## usd\_pledged\_real

## 1 0

## 2 2421

## 3 220

## 4 1

## 5 1283

## 6 52375

**c) Looking for missing values:**

Hurrah, a really clean dataset, even after searching for “empty” strings. 

# Check for NAs:

sapply(ksdf, function(x) sum(is.na(x)))

## ID name category main\_category

## 0 0 0 0

## currency deadline goal launched

## 0 0 0 0

## pledged state backers country

## 0 0 0 0

## usd.pledged usd\_pledged\_real

## 3797 0

# Check for empty strings:

nrow(subset(ksdf, is.na(ksdf$name)))

## [1] 0

**d) Date Formatting and splitting:**

We have two dates in our dataset – “launch date” and “deadline date”.We convert them from strings to date format.  
We also split these dates into the respective year and month columns, so that we can plot variations over time.  
So we will now have 4 new columns: launch\_year, launch\_month, deadline\_year and deadline\_month.

**Exploratory analysis:**

**a) How many projects are successful?**

prop.table(table(ksdf$state))\*100

##

## canceled failed live successful suspended undefined

## 10.2410864 52.2153060 0.7391836 35.3762336 0.4875073 0.9406831

We see that “failed” and “successful” are the two main categories, comprising ~88% of our dataset.  
Sadly we do not know why some projects are marked “undefined” or “canceled”.  
“live”” projects are those where the deadlines have not yet passed, although a few among them are already achieved their goal.  
Surprisingly, some ‘canceled’ projects had also met their goals (pledged\_amount >= goal).  
Since these other categories are a very small portion of the dataset, we will subset and only consider records with satus “failed” or “successful” for the rest of the analysis.

**b) How many countries have projects on kickstarter?**

##

## AT AU BE CA CH DE DK ES FR GB

## 485 6616 523 12370 652 3436 926 1873 2520 29454

## HK IE IT JP LU MX N,0"" NL NO NZ

## 477 683 2369 23 57 1411 210 2411 582 1274

## SE SG US

## 1509 454 261360

We see projects are overwhelmingly US. Some country names have the tag N,0“”, so marking them as unknown.

**c) Number of projects launched per year:**

##

## 2009 2010 2011 2012 2013 2014 2015 2016 2017

## 1179 9577 24049 38480 41101 59306 65272 49292 43419

Looks like some records say dates like 1970, which does not look right. So we discard any records with a launch / deadline year before 2009.  
Plotting the counts per year on a graphs: < br />From the graph below, it looks like the count of projects peaked in 2015, then went down. However, this should NOT be taken as an indicator of success rates.

Drilling down a bit more to see count of projects by main\_category.

Over the years, maximum number of projects have been in the categories:

* 1. Film & Video
  2. Music
  3. Publishing

**d) Number of projects by sub-category: (Top 20 only)**

The Top 5 sub-categories are:

* 1. Product Design
  2. Documentary
  3. Music
  4. Tabletop Games (interesting!!!)
  5. Shorts (really?! )

Let us now see “Status” of projects for these Top 5 sub\_categories:  
From the graph below, we see that for category “shorts” and “tabletop games” there are more successfull projects than failed ones.

**e) Backers by category and sub-category:**

Since there are a lot of sub-categories, let us explore the sub-categories under the main theme “Design”

Product design is not just the sub-category with the highest count of projects, but also the category with the highest success ratio.

**f) add flag to see how many got funded more than the goal.**

ksdf$goal\_flag <- ifelse(ksdf$pledged >= ksdf$goal, 1, 0)

prop.table(table(ksdf$goal\_flag))\*100

##

## 0 1

## 59.61197 40.38803

So ~40% of projects reached or surpassed their goal, which matches the number of successful projects .

**g) Calculate average contribution per backer:**

From the mean, median and max values we quickly see that the median amount contributed by each backer is only ~$40 whereas the mean is higher due to the extreme positive values. The max amount by a single backer is ~$5000.

summary(ksdf$contrib)

## Min. 1st Qu. Median Mean 3rd Qu. Max.

## 0.00 16.00 41.78 73.35 78.00 50000.00

hist(n$contrib, main = "Histogram for number of contributors")

**h) Calculate reach\_ratio**

The amount per backer is a good start, but what if the goal amount itself is only $1000? Then an average contribution per backer of $50 impies we only need 20 backers.  
So to better understand the probability of a project’s success, we create a derived metric called “reach\_ratio”.  
This takes the average user contribution and compares it against the goal fund amount.

## Min. 1st Qu. Median Mean 3rd Qu. Max.

## 0.00 0.16 0.75 6.17 2.16 166666.67

We see the median reach\_ratio is <1%. Only in the third quartile do we even touch 2%!  
Clearly most projects have a very low reach ratio. We could subset for “successful” projects only and check if the reach\_ratio is higher.

**i) Number of days to achieve goal:**

Most projects are run for a month, as seen from graph below.

**Predictive Analystics:**

We will apply a very simple decision tree algorithm to our dataset.  
Since we do not have a separate “test” set, we will split the input dataframe into 2 parts (70/30 split).  
We will use the smaller set to test the accuracy of out algorithm.

ksdf$status = ifelse(ksdf$state == 'failed', 0, 1)

## 70% of the sample size

smp\_size <- floor(0.7 \* nrow(ksdf))

## set the seed to make your partition reproductible

set.seed(486)

train\_ind <- sample(seq\_len(nrow(ksdf)), size = smp\_size)

train <- ksdf[train\_ind, ]

test <- ksdf[-train\_ind, ]

library(tree)

tree1 <- tree(status ~ goal + reach\_ratio + category + backers + country + launch\_year , data = train)

## Warning in tree(status ~ goal + reach\_ratio + category + backers + country

## + : NAs introduced by coercion

summary(tree1)

##

## Regression tree:

## tree(formula = status ~ goal + reach\_ratio + category + backers +

## country + launch\_year, data = train)

## Variables actually used in tree construction:

## [1] "backers" "reach\_ratio"

## Number of terminal nodes: 9

## Residual mean deviance: 0.02429 = 5640 / 232200

## Distribution of residuals:

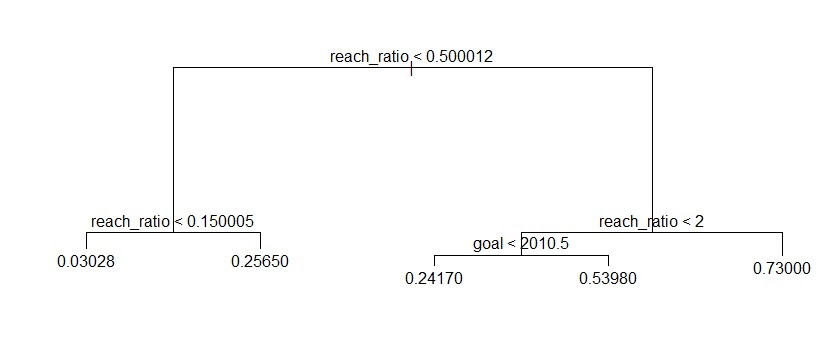
## Min. 1st Qu. Median Mean 3rd Qu. Max.

## -0.9816000 -0.0006945 -0.0006945 0.0000000 0.0410400 0.9993000

Taking a peek at the decision tree rules:

plot(tree1)

text(tree1 ,pretty =0)



kickstarter success decision tree

tree1

## node), split, n, deviance, yval

## \* denotes terminal node

##

## 1) root 232172 55900.00 0.4039000

## 2) backers < 17.5 121911 9326.00 0.0834600

## 4) reach\_ratio < 5.88118 107986 74.95 0.0006945 \*

## 5) reach\_ratio > 5.88118 13925 2774.00 0.7253000

## 10) backers < 5.5 4723 1031.00 0.3218000

## 20) reach\_ratio < 19.9667 2792 0.00 0.0000000 \*

## 21) reach\_ratio > 19.9667 1931 323.50 0.7872000 \*

## 11) backers > 5.5 9202 580.00 0.9324000 \*

## 3) backers > 17.5 110261 20210.00 0.7583000

## 6) reach\_ratio < 0.79672 40852 10190.00 0.5217000

## 12) backers < 128.5 16858 39.91 0.0023730 \*

## 13) backers > 128.5 23994 2413.00 0.8866000 \*

## 7) reach\_ratio > 0.79672 69409 6383.00 0.8975000

## 14) backers < 35.5 20535 3836.00 0.7514000

## 28) reach\_ratio < 2.85458 4816 0.00 0.0000000 \*

## 29) reach\_ratio > 2.85458 15719 284.60 0.9816000 \*

## 15) backers > 35.5 48874 1924.00 0.9590000 \*

Thus we see that “backers” and “reach-ratio” are the main significant variables.

Re-applying the tree rules to the training set itself, we can validate our model:

Predt <- predict(tree1, train)

validf <- data.frame( kickstarter\_id = train$ID, orig\_status = train$status, new\_status = Predt)

validf$new = ifelse(validf$new\_status < 0.5, 0, 1)

# contingency Tables:

table(validf$orig\_status, validf$new)

##

## 0 1

## 0 132337 6051

## 1 115 93669

# Area under the curve

library(pROC)

auc(validf$orig\_status, validf$new)

## Area under the curve: 0.9775

From the above tables, we see that the error rate = ~3% and area under curve >= 97%

Finally applying the tree rules to the test set, we get the following stats:

Pred1 <- predict(tree1, test)

From the above tables, we see that still the error rate = ~3% and area under curve >= 97%

**Conclusion:**

Thus in this tutorial, we explored the factors that contribtue to a project’s success. Main theme and sub-category were important, but the number of backers and “reach\_ratio” were found to be most critical.  
If a founder wanted to gauge their probability of success, they could measure their “reach-ratio” halfway to the deadline, or perhaps when 25% of the timeline is complete. If the numbers are lower, it means they need to double down and use promotions/social media marketing to get more backers and funding.