Logistic Regression with Kickstarter Data - Seasons

Marcelo Sanches

August 23, 2018

Project Overview and Limitations

This project is a continuation of Project No.1, so the scope and limitations of that project apply. See full description in the Project No. 1 files.

This project focuses on seasonal aspects of kickstarter projects.

1. Data Cleaning and Preparation

The data cleaning process is the same up until the creation of dummy variables for the main categories. We do not use kickstarter main categories in this project so we dump that variable.

```
##   state   launched   deadline duration logloggoal
## 2     0 2017-09-02 2017-11-01       60   2.333013
## 3     0 2013-01-12 2013-02-26       45   2.371590
## 4     0 2012-03-17 2012-04-16       30   2.142087
## 5     0 2015-07-04 2015-08-29       56   2.290327
## 6     1 2016-02-26 2016-04-01       35   2.381376
## 7     1 2014-12-01 2014-12-21       20   1.932645
```

To create seasons out of the launched and deadline dates, I've re-formatted the `getSeason` function created by Josh O'Brien in **this 2012 StackOverflow post**. (See code appendix for details.)

```
##   state launched deadline duration logloggoal
## 2     0   Summer     Fall       60   2.333013
## 3     0   Winter   Winter       45   2.371590
## 4     0   Winter   Spring       30   2.142087
## 5     0   Summer   Summer       56   2.290327
## 6     1   Winter   Spring       35   2.381376
## 7     1     Fall   Winter       20   1.932645
```

```
# percentage of different seasons for launched and deadline
round(nrow(us18[us18$launched != us18$deadline, ])/nrow(us18),4)*100
```

```
## [1] 37.13
```

We see that 37% of the time the launched and deadline dates fall into different seasons. Several avenues of thought could be pursued:

- Use launched dates: campaign efforts tend to skew toward the beginning of a project
- Use deadline dates: compare with launched dates
- Create a season variable using a more fine-toothed comb: if a project mostly happened during one season, re-code it for that season
- Dump all the ambiguous projects: speedy results

We could also reorient the analysis toward a particular date and compute project success as a function of how far from that date a project was launched or met the deadline.

As a first quick attempt, we dump all the ambiguous projects, since even after 37% of the data is gone we still have 181,464 rows. Then we create dummy variables for K-1 seasons (base case = Fall).

```
##    state duration logloggoal winter spring summer
## 3      0       45   2.371590      1      0      0
## 5      0       56   2.290327      0      0      1
## 8      0       45   2.315169      1      0      0
## 9      0       35   2.462667      0      1      0
## 10     0       30   2.405335      0      0      1
## 16     0       30   2.057202      1      0      0
```

---

2. Data Analysis

We use a logistic regression model with an interaction between duration and goal.

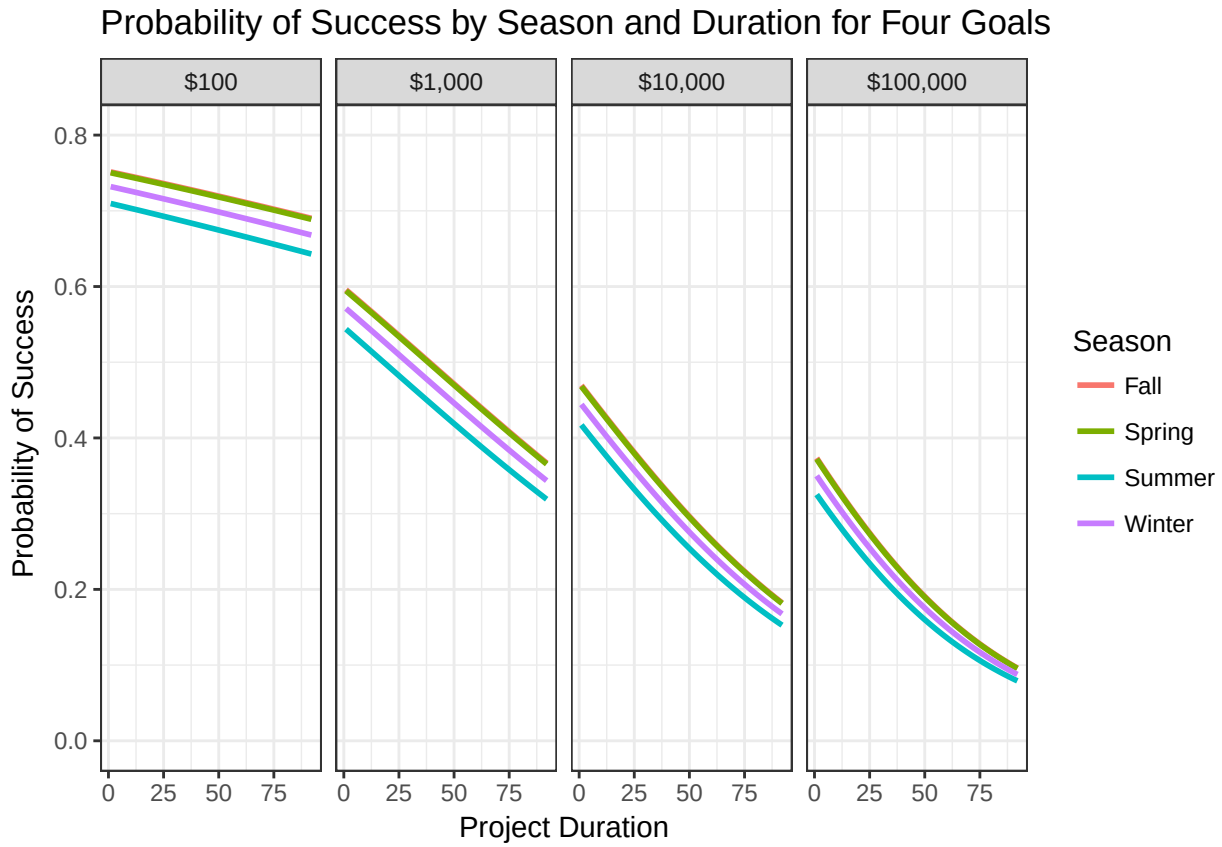
```
##
## Call:
## glm(formula = state ~ duration * logloggoal + winter + spring +
##     summer, family = binomial, data = seasonmatch)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6658  -0.9793  -0.7964   1.2733   2.1428
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           3.789052   0.170179  22.265  < 2e-16 ***
## duration              0.022631   0.005388   4.201 2.66e-05 ***
## logloggoal           -1.754713   0.081341 -21.572  < 2e-16 ***
## winter               -0.101460   0.014736  -6.885 5.77e-12 ***
## spring               -0.005816   0.013647  -0.426     0.67
## summer               -0.212262   0.013651 -15.549  < 2e-16 ***
## duration:logloggoal -0.017017   0.002553  -6.666 2.63e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 241576  on 181055  degrees of freedom
## Residual deviance: 232277  on 181049  degrees of freedom
## AIC: 232291
##
## Number of Fisher Scoring iterations: 4
```

We then calculate prediction probabilities for plotting, which required many lines of code, found in the appendix.

A tidy data frame holds the probabilities ('values' column) and associated categorical variables for a season and a goal amount for plotting. This data frame has 1472 rows = 92 durations * 4 seasons * 4 goal amounts chosen for display. Here we see the first change in seasons:

```
# tidy data frame with factor variables
dfm.melt <- melt(dfm, id = "Duration")
dfm.melt$Season <- rep(c("Fall", "Winter", "Spring", "Summer"), each = 92*4)
dfm.melt$Goal <- factor(rep(rep(c("$100", "$1,000", "$10,000", "$100,000"), each = 92), 4))
dfm.melt$Goal <- factor(dfm.melt$Goal, levels(dfm.melt$Goal)[c(3,1,2,4)])
dfm.melt <- dfm.melt[,-2]; dfm.melt[366:371, ]
```

```
##     Duration      value Season     Goal
## 366       90 0.09939455   Fall $100,000
## 367       91 0.09771114   Fall $100,000
## 368       92 0.09605319   Fall $100,000
## 369        1 0.73193706 Winter     $100
## 370        2 0.73127792 Winter     $100
## 371        3 0.73061777 Winter     $100
```



Fall and spring have very similar probabilities so spring over-rides fall in the coloring scheme; both have the highest probabilities. Projects take a dip in the summer, and winter probabilities of success are halfway between fall/spring and summer.

As previously seen, the highest the goal and duration, the lowest the probability of success, but the rate of decay can be quantified and this plot provides a visualization. The probabilities are somewhat stable for cheaper projects, but take a quicker dip through the more expensive projects in the $10,000s$, and flatten out by the time projects are really expensive.

It is doubful that shorter projects with a $\$100,000$ goal have a near 40% probability of success, however, so this model needs revision and fine-tuning.

Code Appendix

Data Cleaning and Preparation

```
# load pertinent variables
ks18 <- read.csv(file = "./kickstarter data/ks-projects-201801.csv",
                 header = TRUE)[,-c(1:3, 5, 9, 11, 13:15)]

# subset US projects, remove country variable
us18 <- ks18[ks18$country == "US", -6]

# remove "live" projects (no outcome to base prediction on)
us18 <- us18[us18$state != "live", ]

# convert factors to date variables
us18$deadline <- as.Date(us18$deadline)
us18$launched <- as.Date(us18$launched)

# compute duration (project length) variable
us18$duration <- as.numeric(us18$deadline - us18$launched)

# re-order data
us18 <- us18[,c(5,4,2,6,3,1)]

# dummy-code outcome variable 'state' to predict success
us18$state <- ifelse(us18$state == "successful", 1, 0)

# discard senseless date outliers (projects from 1970)
us18 <- us18[us18$duration < 100, ]

# discard projects under $100 --> gaming the system
us18 <- us18[us18$goal > 99, ]

# transform goal to log(log(goal)) to normalize it for regression
us18$logloggoal <- log(log(us18$goal))

# dump goal and main_category
us18 <- us18[, -c(5,6)]
head(us18)
```

```
# recode dates as seasons
# getSeason function (from 2012 StackOverflow post by Josh O'Brien)
getSeason <- function(DATES) {
    WS <- as.Date("2012-12-21", format = "%Y-%m-%d") # Winter Solstice
    SE <- as.Date("2012-3-21",  format = "%Y-%m-%d") # Spring Equinox
    SS <- as.Date("2012-6-21",  format = "%Y-%m-%d") # Summer Solstice
    FE <- as.Date("2012-9-21",  format = "%Y-%m-%d") # Fall Equinox

    # Convert dates from any year to 2012 dates
    d <- as.Date(strftime(DATES, format="2012-%m-%d"))
```

```
    ifelse (d >= WS | d < SE, "Winter",
      ifelse (d >= SE & d < SS, "Spring",
        ifelse (d >= SS & d < FE, "Summer", "Fall")))
}

# pass us18 date variables to getSeason
us18$launched <- getSeason(us18$launched)
us18$deadline <- getSeason(us18$deadline)
head(us18)
```

```
# percentage of different seasons for launched and deadline
round(nrow(us18[us18$launched != us18$deadline, ])/nrow(us18),4)*100
```

```
# dump 37% of the data with ambiguous seasons
seasonmatch <- us18[us18$launched == us18$deadline, -2]

# dummies for seasons, base case == Fall
seasonmatch$winter <- ifelse(seasonmatch$deadline == "Winter", 1, 0)
seasonmatch$spring <- ifelse(seasonmatch$deadline == "Spring", 1, 0)
seasonmatch$summer <- ifelse(seasonmatch$deadline == "Summer", 1, 0)

# dump deadline
seasonmatch <- seasonmatch[, -2]
head(seasonmatch)
```

Data Analysis

```
# logistic regression call
mod1 <- glm(state ~ duration * logloggoal + winter + spring + summer,
            family=binomial,
            data=seasonmatch)
summary(mod1)
```

```
# a0 = Film & Video project with 0-day duration and $0 logloggoal
a0 <- coef(mod1)[1]
b_duration <- coef(mod1)[2]
b_logloggoal <- coef(mod1)[3]
b_winter <- coef(mod1)[4]
b_spring <- coef(mod1)[5]
b_summer <- coef(mod1)[6]
b_interaction <- coef(mod1)[7]


# PREDICTIONS
# initial values for loops
durations <- 92
loglogGOAL1 <- log(log(100))
loglogGOAL2 <- log(log(1000))
loglogGOAL3 <- log(log(10000))
loglogGOAL4 <- log(log(100000))

# FALL probs
Fprobs1 <- rep(NA, durations)
```

```
Fprobs2 <- rep(NA, durations)
Fprobs3 <- rep(NA, durations)
Fprobs4 <- rep(NA, durations)
for (i in 1:durations) {
    regr1 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL1 +
            b_interaction * i * loglogGOAL1
    regr2 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL2 +
            b_interaction * i * loglogGOAL2
    regr3 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL3 +
            b_interaction * i * loglogGOAL3
    regr4 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL4 +
            b_interaction * i * loglogGOAL4
    Fprobs1[i] <- unname(exp(regr1)/(1+exp(regr1)))
    Fprobs2[i] <- unname(exp(regr2)/(1+exp(regr2)))
    Fprobs3[i] <- unname(exp(regr3)/(1+exp(regr3)))
    Fprobs4[i] <- unname(exp(regr4)/(1+exp(regr4)))
}

# WINTER probs
Wprobs1 <- rep(NA, durations)
Wprobs2 <- rep(NA, durations)
Wprobs3 <- rep(NA, durations)
Wprobs4 <- rep(NA, durations)
for (i in 1:durations) {
    regr1 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL1 +
            b_winter +
            b_interaction * i * loglogGOAL1
    regr2 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL2 +
            b_winter +
            b_interaction * i * loglogGOAL2
    regr3 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL3 +
            b_winter +
            b_interaction * i * loglogGOAL3
    regr4 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL4 +
            b_winter +
            b_interaction * i * loglogGOAL4
    Wprobs1[i] <- unname(exp(regr1)/(1+exp(regr1)))
```

```
    Wprobs2[i] <- unname(exp(regr2)/(1+exp(regr2)))
    Wprobs3[i] <- unname(exp(regr3)/(1+exp(regr3)))
    Wprobs4[i] <- unname(exp(regr4)/(1+exp(regr4)))
}

# SPRING probs
Sprobs1 <- rep(NA, durations)
Sprobs2 <- rep(NA, durations)
Sprobs3 <- rep(NA, durations)
Sprobs4 <- rep(NA, durations)
for (i in 1:durations) {
    regr1 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL1 +
            b_spring +
            b_interaction * i * loglogGOAL1
    regr2 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL2 +
            b_spring +
            b_interaction * i * loglogGOAL2
    regr3 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL3 +
            b_spring +
            b_interaction * i * loglogGOAL3
    regr4 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL4 +
            b_spring +
            b_interaction * i * loglogGOAL4
    Sprobs1[i] <- unname(exp(regr1)/(1+exp(regr1)))
    Sprobs2[i] <- unname(exp(regr2)/(1+exp(regr2)))
    Sprobs3[i] <- unname(exp(regr3)/(1+exp(regr3)))
    Sprobs4[i] <- unname(exp(regr4)/(1+exp(regr4)))
}

# SUMMER probs
Eprobs1 <- rep(NA, durations)
Eprobs2 <- rep(NA, durations)
Eprobs3 <- rep(NA, durations)
Eprobs4 <- rep(NA, durations)
for (i in 1:durations) {
    regr1 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL1 +
            b_summer +
            b_interaction * i * loglogGOAL1
    regr2 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL2 +
            b_summer +
            b_interaction * i * loglogGOAL2
```

```
    regr3 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL3 +
            b_summer +
            b_interaction * i * loglogGOAL3
    regr4 <- a0 +
            b_duration * i +
            b_logloggoal * loglogGOAL4 +
            b_summer +
            b_interaction * i * loglogGOAL4
    Eprobs1[i] <- unname(exp(regr1)/(1+exp(regr1)))
    Eprobs2[i] <- unname(exp(regr2)/(1+exp(regr2)))
    Eprobs3[i] <- unname(exp(regr3)/(1+exp(regr3)))
    Eprobs4[i] <- unname(exp(regr4)/(1+exp(regr4)))
}
```

```
# VISUALIZATION
# install-load required packages
if("ggplot2" %in% rownames(installed.packages()) == FALSE) {
    suppressWarnings(install.packages("ggplot2"))
}
suppressMessages(require(ggplot2))
if("reshape2" %in% rownames(installed.packages()) == FALSE) {
    suppressWarnings(install.packages("reshape2"))
}
suppressMessages(require(reshape2))

# data frame to hold probabilities
dfm <- data.frame(
        "Duration" = 1:92,
        "Fall 100" = Fprobs1,
        "Fall 1,000" = Fprobs2,
        "Fall 10,000" = Fprobs3,
        "Fall 100,000" = Fprobs4,
        "Winter 100" = Wprobs1,
        "Winter 1,000" = Wprobs2,
        "Winter 10,000" = Wprobs3,
        "Winter 100,000" = Wprobs4,
        "Spring 100" = Sprobs1,
        "Spring 1,000" = Sprobs2,
        "Spring 10,000" = Sprobs3,
        "Spring 100,000" = Sprobs4,
        "Summer 100" = Eprobs1,
        "Summer 1,000" = Eprobs2,
        "Summer 10,000" = Eprobs3,
        "Summer 100,000" = Eprobs4
        )
```

```
# tidy data frame with factor variables
dfm.melt <- melt(dfm, id = "Duration")
dfm.melt$Season <- rep(c("Fall", "Winter", "Spring", "Summer"), each = 92*4)
dfm.melt$Goal <- factor(rep(rep(c("$100", "$1,000", "$10,000", "$100,000"),
                                                        each = 92), 4))
dfm.melt$Goal <- factor(dfm.melt$Goal, levels(dfm.melt$Goal)[c(3,1,2,4)])
```

```
dfm.melt <- dfm.melt[,-2]
dfm.melt[366:371, ]
```

```
ggplot(data=dfm.melt, aes(x=Duration,y=value, color=Season)) +
    geom_line(size=1) +
    ylim(0,.8) +
    facet_grid(. ~ Goal) +
    theme_bw() +
    ggtitle("Probability of Success by Season and Duration for Four Goals") +
    xlab("Project Duration") +
    ylab(" Probability of Success") +
    labs(color = "Season")
```
