Predict the probability of success when a project land on Kickstarter platform

Zhitian Liu

December 10, 2020

Abstract

Kickstarter is a world-famous online crowdfunding platform. Building a predictive model for Kickstarter can benefits both project owners and the platform itself, by using the data collected from kickstarter, we use multilevel logistic regression model to forcast a project will succeed or failed in the end. Our results show that our model can achieve a prediction accuracy of 0.7. Adding more information into the model and take inflation into account may be next step to improve the predictive model.

Introduction

Background

Kickstarter is a world-famous online crowdfunding platform. The platform mainly focuses on creativity and merchandising. Project owners propose there projects on the platform, and backers donate the money if they like this idea and wish it to come true. The question I want to solve here is: **if I'm the manager of the Kickstarter company, can I forecast whether each project will succeed in the end When they just land on the website.** The reason I'm interested in this question is that by having this Success rate forecasting system, we can inform the project owners when they upload their proposals, we can remind them of the probability that their proposal will be successfully funded, and **automatically suggest** to them that they can increase the probability of success, such as reducing their target amount and extending the deadline. If we predict that the success probability of a crowdfunding project is too low, we can even add some **paid services**, such as advertising for them or giving them a high position in search engines. Generally speaking, if we can predict the success probability of a crowdfunding project, it will benefit both the project owners and the platform.

About the data and model

The data I found from Kaggle is collected from Kickstarter Platform. The data set is large, it has **over 300k observations**. Each observation describe a project's name, ID, country category, time they launched, time they closed,the amount of money which the project owner hope to raise, and the amount of money they actually raised, number of backers the project owned, also it has a indicator variable indicates the current condition (failed or successful) the project is in (2018). the data set can divided into 159 groups by category of the project. The outcome of the model is binary variable (failed or successful). So, it can be used to fit a logistic multilevel model. I would be use **glmer** function in **lmerTest** package to generate the regression model.

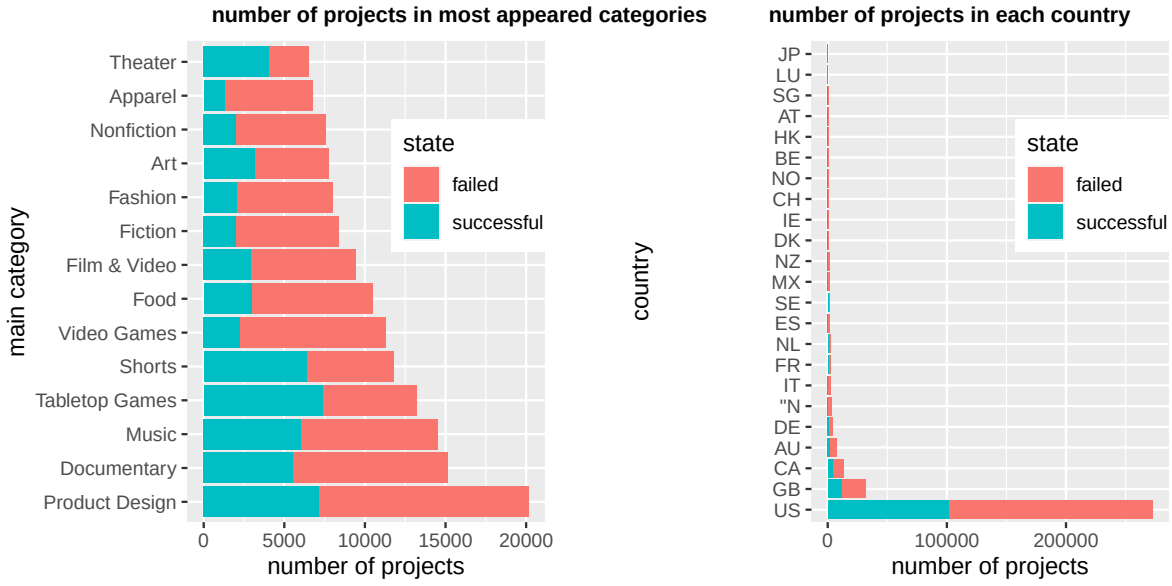
Methods

data processing

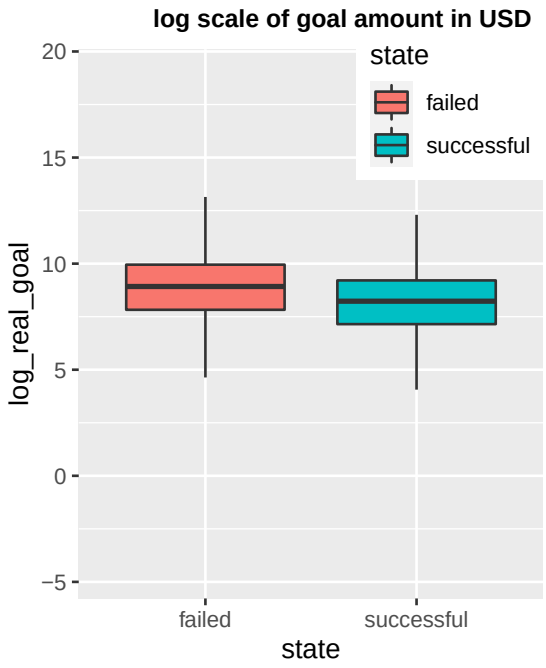
To start the analysis, we firstly need to select the predictor we might use in the regression model. Knowing that our purpose is to predict whether or not a crowdfunding project will succeed when they just land on to the website, so we can't use predictors like number of backers, the money they raised, because these information is collected at the end of crowdfunding process. So the variables that may contribute to the model are launched time, closed time, category, country, goal amount in USD (the USD conversion made by fixer.io api) Through the summary data set, we found that in addition to successful and failed, the outcome has a very small amount of other states, such as canceled and suspended. In order to build a more intuitive model, I defined all observations that the state variable is not successful as failed. Then convert the outcome into 0-1 variable, **1** represent the project successfully raised money they want, **0** represent failed. I also used package **lubridate** to deal with the time variable, convert the launched time into launched year, calculate the duration of the project in days. I found that the numeric variable "goal amount in USD" has a very big range, so I took the log scale of it for a better model fitting later.

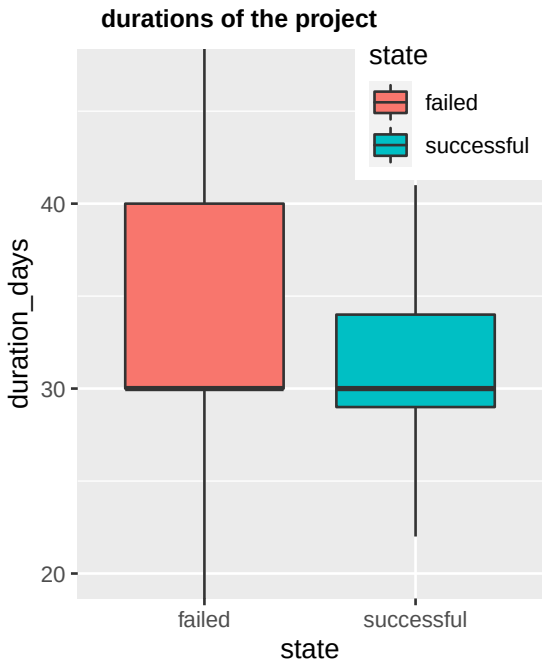
Exploratory data analysis

In order to better observe these variables, I generate some EDA. Firstly I want to see the distribution of the categorical variables: *category* and *country* of the project. Due to there are 159 categories in total, it's pretty hard to plot them all, so I only visualized the 15 most popular categories, we can see from the bar chart on the left, The most popular category of the crowdfunding project is Product Design, there were over 20000 cumulative projects focus on product design proposed from 2009-2018. Also we can see that Games, Music and Documentary were also very popular categories. Except for Tabletop games,Shorts and Theater, most of the project in these categories are more likely to failed to raised the money at last. The chart on the right indicates the country distribution of projects on Kickstarter platform, it is very obvious the distribution is skewed, most of the projects are proposed from the US (over 250000).

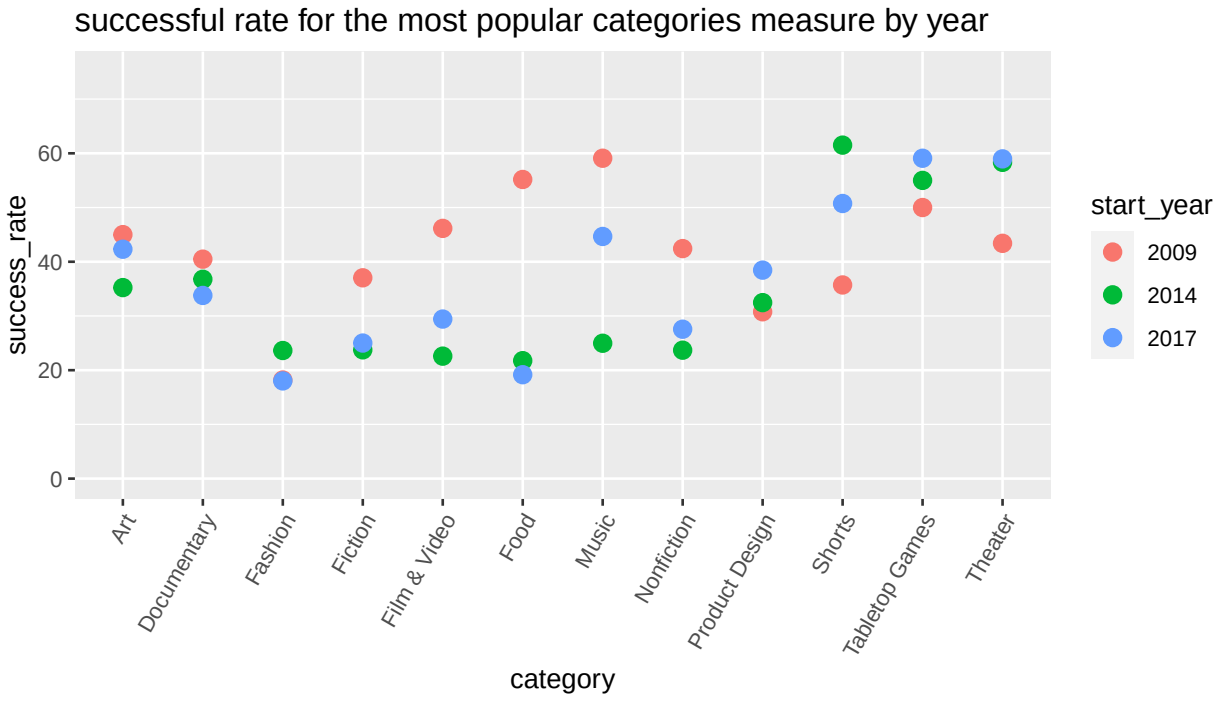


Then, I create 2 boxplots to explore the 2 continuous variable:**log scale of goal amount in USD** and **durations of the project**. The boxplot on the left shows that the projects which were failed in the end had relatively higher funding needs than the projects which were successful raised enough money in the end. The plot on the right shows that both failed and successful projects has the same average duration-30 days. But the opening duration of failed projects is much more uncertain.





The last EDA graph is a dot plot indicates the successful rate for the projects of most popular categories in 2009,2014 and 2017, I want to see whether if there are an obvious trend of the successful rate changing through year. The answer is yes, in most categories, the successful rate of projects in 2009 is higher than 2017, we can assume there's a decreaing trend of the successful rate through time. Although the trend is not very significant since sometimes the successful rate of project in 2014 is lower than 2017.



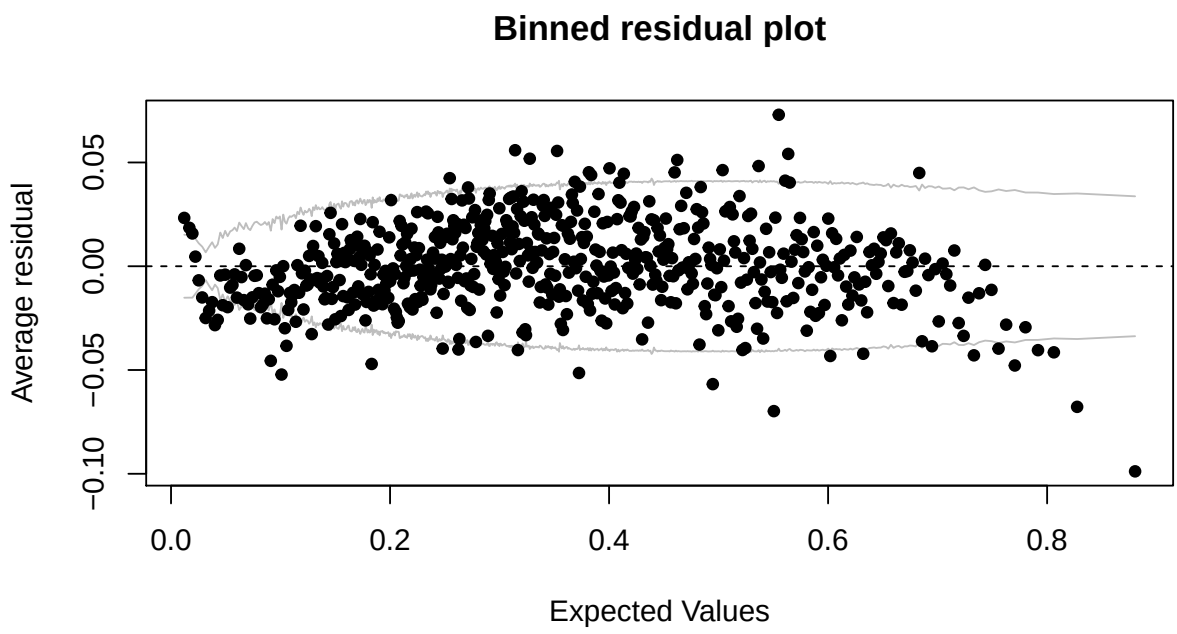
Result

model selection

After the EDA part, we learned that all the predictors we selected have a strong relationship with the outcome. So now we can start the model selection part. By using **glmer** function in **lmerTest** package, we first add the random effect predictor, which is **country** and **category** of the project, and then we add fix effect predictor (**start year**,**duration**, **log scale goal amount in usd**)one by one. By checking the AIC value and the binned residual plot, checking the coefficient value, I have my best fit logistic multilevel regression model. All the predictor I just mentioned before is in this model.

validation

I generate a binned residual plot firstly, It indicates a good quite good fit for this model. The negative and positive residuals are almost evenly distributed-that is good. Most residual points are in the acceptable range which is theoretical 95% error bounds. Although there are several points in the two tails of the the binned residual plot are out of the bounds, but is acceptable.



I also did a predict test for this model, I put the model into my original data set and compare the predicted result with the real value, and got the accuracy of 0.695-nearly**0.7**, It is also the best we can get among all the model we tried.

Inference

From the output of the **glmer**, We know all the fix effect are significant, I also calculate the 95% confidence interval for fix effect.. None of these predictor has a CI across 0, which is good.

Discussion

Overall speaking, The effect of the model has met our expectations, however, there are also many drawbacks that can be improved in the future. ## limitation The model is not perfect, The accuracy to predict a new project is only 0.7, (we don't even know the performance to predict a new data set). There are a lot of

drawbacks for this model. First of all, the data is very asymmetrical, and the number of projects in each country, category, and year is very different. Secondly, We didn't take into account the inflation, Among those years, the value of money is changing, I think it would be better calculate the real value of the goal amount. Thirdly, The predictors are not enough, we only have 5 predictors in the model, In my opinion, The model would be better if we can put more useful variables in it. At last, I didn't try varying slope model because I don't know how to interpret. ## How to improve We can improve the model by adding more useful information into the model, The Kickstarter platform can send a questionnaire when project owners submits the application and ask them more. We can also take into account the inflation.

Bibliography

The function we used to generate the multilevel logistic regression is *glmer* from package **lmerTest** The data source is download from https://www.kaggle.com/kemical/kickstarter-projects?select=ks-projects-201801.csv ordained from https://www.kickstarter.com/

Appendix

95% CI for the final model and the predicting accuracy

```
summary(fit3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 0) [glmerMod]
##  Family: binomial  ( logit )
## Formula: state_01 ~ log_real_goal + start_year + duration_days + (1 |
##     category) + (1 | country)
##    Data: data1
## Control: glmerControl("bobyqa")
##
##       AIC       BIC    logLik  deviance  df.resid
##  405231.5  405296.1 -202609.7  405219.5    352708
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5514 -0.7060 -0.4481  0.9012 19.9566
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  category (Intercept) 0.4740   0.6885
##  country  (Intercept) 0.5981   0.7734
## Number of obs: 352714, groups:  category, 159; country, 23
##
## Fixed effects:
##                   Estimate  Std. Error z value            Pr(>|z|)
## (Intercept)   107.3534538   4.7259593   22.72 <0.0000000000000002 ***
## log_real_goal  -0.2696581   0.0025713 -104.87 <0.0000000000000002 ***
## start_year     -0.0524291   0.0023420  -22.39 <0.0000000000000002 ***
## duration_days  -0.0177826   0.0003211  -55.38 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) lg_rl_ strt_y
```

```
## log_real_gl  0.029
## start_year  -0.999 -0.033
## duratin_dys -0.177 -0.142  0.176
```

```
fm1W <- confint(fit3, method="Wald")
fm1W
```

```
##                      2.5 %       97.5 %
## .sig01                  NA           NA
## .sig02                  NA           NA
## (Intercept)   98.09074387 116.61616372
## log_real_goal -0.27469780  -0.26461844
## start_year    -0.05701929  -0.04783896
## duration_days -0.01841201  -0.01715324
```

```
#accuracy
predict_value=ifelse(fitted(fit3)>=0.5,1,0)
predict_value=as.vector(predict_value)
accuracy=cbind.data.frame(predict_value,data1$state_01)
accuracy$correct=ifelse(accuracy$predict_value==accuracy$`data1$state_01`,1,0)
acc=sum(accuracy$correct)/nrow(accuracy)
acc
```

```
## [1] 0.6949313
```

```
#model selection
#fit0=glmer(state_01 ~1+(1|country),data=data1,family=binomial(link="logit"),control=glmerControl("boby
#summary(fit0)
#accuracy
#predict_value=ifelse(fitted(fit0)>=0.5,1,0)
#predict_value=as.vector(predict_value)
#accuracy=cbind.data.frame(predict_value,data1$state_01)
#accuracy$correct=ifelse(accuracy$predict_value==accuracy$`data1$state_01`,1,0)
#acc=sum(accuracy$correct)/nrow(accuracy)
#acc


#fit0.5=glmer(state_01 ~1+(1|main_category)+(1|country),data=data1,family=binomial(link="logit"),contro
#summary(fit0.5)

#predict_value=ifelse(fitted(fit0.5)>=0.5,1,0)
#predict_value=as.vector(predict_value)
#accuracy=cbind.data.frame(predict_value,data1$state_01)
#accuracy$correct=ifelse(accuracy$predict_value==accuracy$`data1$state_01`,1,0)
#acc=sum(accuracy$correct)/nrow(accuracy)
#acc



#fit1=glmer(state_01 ~duration_days+log_real_goal+(1|main_category)+(1|country),data=data1,family=binom
#summary(fit1)


#fit2=glmer(state_01 ~log_real_goal+start_year+duration_days+(1|main_category)+(1|country),data=data1,f
#summary(fit2)
```