Predict raised for projects using log

Jaymeen Gandhi

5/25/2020

require(tidyverse)

## Loading required package: tidyverse

## -- Attaching packages ----------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0 v purrr 0.3.4  
## v tibble 3.0.1 v dplyr 0.8.5  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts -------------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

require(modelr)

## Loading required package: modelr

projects = read\_csv('C:/Users/jayme/Downloads/projects.csv')

## Parsed with column specification:  
## cols(  
## id = col\_double(),  
## backers = col\_double(),  
## parentCat = col\_character(),  
## duration = col\_double(),  
## goal = col\_double(),  
## raised = col\_double(),  
## faqs = col\_double(),  
## comments = col\_double(),  
## description.length = col\_double()  
## )

projects

## # A tibble: 1,000 x 9  
## id backers parentCat duration goal raised faqs comments description.len~  
## <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 0 Publishi~ 30.0 4000 0 0 0 2780  
## 2 2 874 Publishi~ 60 80000 85750. 0 142 8230  
## 3 3 2505 Games 30 8000 49574. 4 193 3418  
## 4 4 38 Publishi~ 60.8 8133 8142 1 0 2294  
## 5 5 90 Film & V~ 31.1 5000 5480 0 11 2067  
## 6 6 43 Music 45.4 4500 5110 0 6 3272  
## 7 7 149 Games 30.0 3500 18988 0 153 9215  
## 8 8 34 Photogra~ 30.0 1500 2008 1 1 2163  
## 9 9 26 Publishi~ 55.6 1000 1000 1 0 3065  
## 10 10 63 Film & V~ 30.0 6000 6055. 0 5 3111  
## # ... with 990 more rows

f=raised ~ backers + duration  
model = lm(f,data = projects)  
summary(model)

##   
## Call:  
## lm(formula = f, data = projects)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -551250 -2383 608 2292 2700914   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2268.601 8649.338 0.262 0.793   
## backers 119.305 5.683 20.993 <2e-16 \*\*\*  
## duration -134.010 223.156 -0.601 0.548   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 95550 on 997 degrees of freedom  
## Multiple R-squared: 0.3066, Adjusted R-squared: 0.3052   
## F-statistic: 220.4 on 2 and 997 DF, p-value: < 2.2e-16

## We will be predicting the raised variable based upon other variables.

## Since if we plot the raised variable graph it is positively scewed so we transform it to log.

projects\_l = projects %>% mutate(log\_raised=log(raised))  
projects\_l

## # A tibble: 1,000 x 10  
## id backers parentCat duration goal raised faqs comments description.len~  
## <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 0 Publishi~ 30.0 4000 0 0 0 2780  
## 2 2 874 Publishi~ 60 80000 85750. 0 142 8230  
## 3 3 2505 Games 30 8000 49574. 4 193 3418  
## 4 4 38 Publishi~ 60.8 8133 8142 1 0 2294  
## 5 5 90 Film & V~ 31.1 5000 5480 0 11 2067  
## 6 6 43 Music 45.4 4500 5110 0 6 3272  
## 7 7 149 Games 30.0 3500 18988 0 153 9215  
## 8 8 34 Photogra~ 30.0 1500 2008 1 1 2163  
## 9 9 26 Publishi~ 55.6 1000 1000 1 0 3065  
## 10 10 63 Film & V~ 30.0 6000 6055. 0 5 3111  
## # ... with 990 more rows, and 1 more variable: log\_raised <dbl>

##Since we need to get rid of infinities we filter out projects greater than 1.

projects = projects %>% filter(raised > 1)  
projects1 =projects %>% mutate(log\_raised = log(raised))  
projects1

## # A tibble: 963 x 10  
## id backers parentCat duration goal raised faqs comments description.len~  
## <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2 874 Publishi~ 60 80000 85750. 0 142 8230  
## 2 3 2505 Games 30 8000 49574. 4 193 3418  
## 3 4 38 Publishi~ 60.8 8133 8142 1 0 2294  
## 4 5 90 Film & V~ 31.1 5000 5480 0 11 2067  
## 5 6 43 Music 45.4 4500 5110 0 6 3272  
## 6 7 149 Games 30.0 3500 18988 0 153 9215  
## 7 8 34 Photogra~ 30.0 1500 2008 1 1 2163  
## 8 9 26 Publishi~ 55.6 1000 1000 1 0 3065  
## 9 10 63 Film & V~ 30.0 6000 6055. 0 5 3111  
## 10 11 2 Music 59.7 2000 35 0 0 361  
## # ... with 953 more rows, and 1 more variable: log\_raised <dbl>

f1 = log\_raised ~ backers + duration  
model\_l = lm(f1, data = projects1)  
summary(model\_l)

##   
## Call:  
## lm(formula = f1, data = projects1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.4739 -0.8448 0.2921 1.1592 4.0255   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.3131715 0.1553867 47.064 <2e-16 \*\*\*  
## backers 0.0016803 0.0001008 16.676 <2e-16 \*\*\*  
## duration 0.0027502 0.0040054 0.687 0.492   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.691 on 960 degrees of freedom  
## Multiple R-squared: 0.225, Adjusted R-squared: 0.2234   
## F-statistic: 139.4 on 2 and 960 DF, p-value: < 2.2e-16

predictions = predict(model\_l , data =projects1)  
projects %>% mutate(predictions = predictions)

## # A tibble: 963 x 10  
## id backers parentCat duration goal raised faqs comments description.len~  
## <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2 874 Publishi~ 60 80000 85750. 0 142 8230  
## 2 3 2505 Games 30 8000 49574. 4 193 3418  
## 3 4 38 Publishi~ 60.8 8133 8142 1 0 2294  
## 4 5 90 Film & V~ 31.1 5000 5480 0 11 2067  
## 5 6 43 Music 45.4 4500 5110 0 6 3272  
## 6 7 149 Games 30.0 3500 18988 0 153 9215  
## 7 8 34 Photogra~ 30.0 1500 2008 1 1 2163  
## 8 9 26 Publishi~ 55.6 1000 1000 1 0 3065  
## 9 10 63 Film & V~ 30.0 6000 6055. 0 5 3111  
## 10 11 2 Music 59.7 2000 35 0 0 361  
## # ... with 953 more rows, and 1 more variable: predictions <dbl>

#exp(predictions)  
  
projects2=projects %>% mutate(predictions = exp(predictions))  
projects2

## # A tibble: 963 x 10  
## id backers parentCat duration goal raised faqs comments description.len~  
## <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2 874 Publishi~ 60 80000 85750. 0 142 8230  
## 2 3 2505 Games 30 8000 49574. 4 193 3418  
## 3 4 38 Publishi~ 60.8 8133 8142 1 0 2294  
## 4 5 90 Film & V~ 31.1 5000 5480 0 11 2067  
## 5 6 43 Music 45.4 4500 5110 0 6 3272  
## 6 7 149 Games 30.0 3500 18988 0 153 9215  
## 7 8 34 Photogra~ 30.0 1500 2008 1 1 2163  
## 8 9 26 Publishi~ 55.6 1000 1000 1 0 3065  
## 9 10 63 Film & V~ 30.0 6000 6055. 0 5 3111  
## 10 11 2 Music 59.7 2000 35 0 0 361  
## # ... with 953 more rows, and 1 more variable: predictions <dbl>

##Now let us understand the difference why to use log. So lets consider we are nor using any log. ## creating a new model for comparision ##Creating a model without any log

f = raised ~ duration  
model\_without\_log = lm(f, data = projects)  
summary(model\_without\_log)

##   
## Call:  
## lm(formula = f, data = projects)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -18658 -15488 -13505 -8718 2929107   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 19686.09 10685.13 1.842 0.0657 .  
## duration -96.95 276.66 -0.350 0.7261   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 116800 on 961 degrees of freedom  
## Multiple R-squared: 0.0001278, Adjusted R-squared: -0.0009127   
## F-statistic: 0.1228 on 1 and 961 DF, p-value: 0.7261

predictions = predict(model\_without\_log,data = projects)  
project3=projects %>% mutate(predict = predictions,err=predict-raised)   
project3

## # A tibble: 963 x 11  
## id backers parentCat duration goal raised faqs comments description.len~  
## <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2 874 Publishi~ 60 80000 85750. 0 142 8230  
## 2 3 2505 Games 30 8000 49574. 4 193 3418  
## 3 4 38 Publishi~ 60.8 8133 8142 1 0 2294  
## 4 5 90 Film & V~ 31.1 5000 5480 0 11 2067  
## 5 6 43 Music 45.4 4500 5110 0 6 3272  
## 6 7 149 Games 30.0 3500 18988 0 153 9215  
## 7 8 34 Photogra~ 30.0 1500 2008 1 1 2163  
## 8 9 26 Publishi~ 55.6 1000 1000 1 0 3065  
## 9 10 63 Film & V~ 30.0 6000 6055. 0 5 3111  
## 10 11 2 Music 59.7 2000 35 0 0 361  
## # ... with 953 more rows, and 2 more variables: predict <dbl>, err <dbl>

project3 %>% summarise(rmse = sqrt(mean(err^2)),rrmse = sqrt(mean(err^2/raised)))

## # A tibble: 1 x 2  
## rmse rrmse  
## <dbl> <dbl>  
## 1 116705. 1027.

##We can classify that, based upon the rmse values(error values) of normal model and the another of the log model.

projects4 = projects %>% mutate(log\_raised=log(raised))  
  
f1<-log\_raised~ duration  
  
model\_with\_log=lm(f1, data=projects4)  
summary(model\_with\_log)

##   
## Call:  
## lm(formula = f1, data = projects4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.5591 -1.0344 0.1482 1.2436 7.2382   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.557954 0.175583 43.045 <2e-16 \*\*\*  
## duration 0.003325 0.004546 0.731 0.465   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.92 on 961 degrees of freedom  
## Multiple R-squared: 0.0005563, Adjusted R-squared: -0.0004837   
## F-statistic: 0.5349 on 1 and 961 DF, p-value: 0.4647

l\_predictions=predict(model\_with\_log, projects4)  
predictions=exp(l\_predictions)  
  
projects4 =projects%>%mutate(predict=predictions, err=predict-raised)  
projects4%>%summarise(rmse=sqrt(mean(err^2)) , rrmse=sqrt(mean((err/raised)^2)))

## # A tibble: 1 x 2  
## rmse rrmse  
## <dbl> <dbl>  
## 1 117552. 38.9

##Transforming the input variable.

f4= raised ~ goal  
model2 = lm(f4,data = projects)  
summary(model2)

##   
## Call:  
## lm(formula = f4, data = projects)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -673781 -8468 -5451 -3638 2873085   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5918.6729 3775.3315 1.568 0.117   
## goal 0.6688 0.0724 9.237 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 112000 on 961 degrees of freedom  
## Multiple R-squared: 0.08155, Adjusted R-squared: 0.08059   
## F-statistic: 85.32 on 1 and 961 DF, p-value: < 2.2e-16

rmse(model2,data =projects )

## [1] 111852.1

f5 = raised ~ sqrt(goal)  
model3 = lm(f5,data = projects)  
summary(model3)

##   
## Call:  
## lm(formula = f5, data = projects)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -416411 -13053 -1211 7776 2830736   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -24616.59 5362.19 -4.591 5e-06 \*\*\*  
## sqrt(goal) 441.98 43.29 10.211 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 111000 on 961 degrees of freedom  
## Multiple R-squared: 0.09787, Adjusted R-squared: 0.09693   
## F-statistic: 104.3 on 1 and 961 DF, p-value: < 2.2e-16

rmse(model3,data = projects)

## [1] 110853.8