Evaluation of models

Jaymeen Gandhi

April 4, 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()

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 = projects %>% arrange(desc(comments))  
projects = projects %>% filter(comments<10000)  
projects

## # A tibble: 999 x 9  
## id backers parentCat duration goal raised faqs comments  
## <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 331 3027 Games 32.0 400000 1.70e5 8 3525  
## 2 674 1126 Games 30.4 20000 1.40e5 5 3033  
## 3 428 2621 Film & V~ 27.0 60000 3.22e5 6 1662  
## 4 193 5694 Technolo~ 30 250000 1.21e6 19 1229  
## 5 217 4965 Technolo~ 30 750000 8.99e5 12 1160  
## 6 52 2639 Games 37.5 9000 8.25e4 7 975  
## 7 82 208 Games 30.0 12800 3.53e4 6 810  
## 8 329 2087 Games 31.5 12000 8.81e4 4 736  
## 9 246 2327 Design 60.0 20000 1.06e5 1 650  
## 10 874 224 Games 30.0 7500 1.67e4 8 544  
## # ... with 989 more rows, and 1 more variable: description.length <dbl>

f = comments ~ duration + goal + description.length

projects %>% summarise(mean(comments),var(comments))

## # A tibble: 1 x 2  
## `mean(comments)` `var(comments)`  
## <dbl> <dbl>  
## 1 25.6 31429.

##Since the var is very high than mean we can say that the distribution is overspread. So this model will be quasi poisson. ##If the variance and mean are equal than use poissons.

model = glm(f,data = projects,family = quasipoisson)  
summary(model)

##   
## Call:  
## glm(formula = f, family = quasipoisson, data = projects)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -40.807 -5.072 -4.333 -2.830 128.245   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.889e+00 5.504e-01 3.433 0.000622 \*\*\*  
## duration 4.795e-03 1.325e-02 0.362 0.717419   
## goal 3.270e-06 6.503e-07 5.029 5.85e-07 \*\*\*  
## description.length 1.713e-04 1.723e-05 9.942 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for quasipoisson family taken to be 534.3791)  
##   
## Null deviance: 146303 on 998 degrees of freedom  
## Residual deviance: 92147 on 995 degrees of freedom  
## AIC: NA  
##   
## Number of Fisher Scoring iterations: 8

## Since this model does nor have any r-squared value we have to check whether this model is a good fit for prediction or no.

##We will see that as follows

require(broom)

## Loading required package: broom

glance(model) %>% summarise(pseudoR2=1-deviance/null.deviance)

## # A tibble: 1 x 1  
## pseudoR2  
## <dbl>  
## 1 0.370

##Now we will see model prediction and evaluate that prediction

set.seed(1234)  
training\_data = projects %>% sample\_frac(size = 0.7)  
testing\_data = anti\_join(projects,training\_data, by= 'id')

model\_l = glm(f,data = training\_data,family = gaussian)  
model\_p= glm(f,data = training\_data,family = poisson)  
model\_qp = glm(f,data = training\_data, family = quasipoisson)  
  
models = list(model\_l,model\_p,model\_qp)

map\_dfr(models, ~glance(.x) %>% summarise(pseudoR2=1-deviance/null.deviance))

## # A tibble: 3 x 1  
## pseudoR2  
## <dbl>  
## 1 0.228  
## 2 0.405  
## 3 0.405

GAM Models

#install.packages('mgcv')  
require(mgcv)

## Loading required package: mgcv

## Loading required package: nlme

##   
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':  
##   
## collapse

## This is mgcv 1.8-31. For overview type 'help("mgcv-package")'.

require(modelr)

## Loading required package: modelr

##   
## Attaching package: 'modelr'

## The following object is masked from 'package:broom':  
##   
## bootstrap

mtcars %>% head()

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

gam\_model = gam(mpg ~ s(disp), data = mtcars)  
lm\_model = lm(mpg ~ disp , data = mtcars)  
lm\_t\_model = lm(mpg~I(1/disp^2),data = mtcars)  
  
models=list(gam\_model,lm\_model,lm\_t\_model)  
  
(rsquared=map\_dbl(models,~rsquare(.x,mtcars)))

## [1] 0.9029159 0.7183433 0.8168744

mtcars = mtcars %>% mutate(id = row\_number())  
set.seed(1234)  
training\_data = mtcars %>% sample\_frac(size = 0.7)  
testing\_data = anti\_join(mtcars,training\_data, by= 'id')

gam\_model = gam(mpg ~ s(disp), data = training\_data)  
lm\_model = lm(mpg ~ disp , data = training\_data)  
lm\_t\_model = lm(mpg~I(1/disp^2),data = training\_data)  
  
models=list(gam\_model,lm\_model,lm\_t\_model)  
  
(rmses = map\_dbl(models,~rmse(.x,testing\_data)))

## [1] 3.464412 4.575304 2.196294

tibble(model=c("gam\_model","lm\_model","lm\_t\_model"),rsquared,rmses) %>% mutate\_if(is.numeric,~round(.x, 2))

## # A tibble: 3 x 3  
## model rsquared rmses  
## <chr> <dbl> <dbl>  
## 1 gam\_model 0.9 3.46  
## 2 lm\_model 0.72 4.58  
## 3 lm\_t\_model 0.82 2.2