Week 2, HW4: Logit and LASSO

Instructor: Jacob LaRiviere

Email: jlariv@microsoft.com

1. Go to github and read in the SAT dataset. You’ll see there is a .txt file too which is a readme. It has descriptions of all the variables.
   1. Lets try to predict who is going to take the SAT like the online Rmd output.
   2. First use dplyr to summarize the data. What percent of the sample took the SATs?
2. What is the average score rank of high school students who took the SAT versus those that didn’t?

# Rank for Took SAT

Median :75.00

#Rank for Didn’t Take SAT

Median :52.00

# Score for Took SAT

Median : 960.0

# Score for Didn’t Take SAT

Median: 0

* 1. Use everything as a RHS variable except pict, lgsc, and mosaic [since we don’t know what they are and sat since it isn’t a direct outcome

Note: I think normalized scores are normalized SAT scores rather than normalized grades, which is what I thought initially so you might leave those out too.

ln\_reg = lm(formula = satobs ~ rank + mlhs + mcol + flhs + fcol + black + hisp + asian +

female + rdsc + vocab + matsc + nsib, data = sat)

summary(ln\_reg)

* 1. Consider the following logit object commands. What would you use to compare OLS coefficients to logit coefficients (hint: it’s the third one!)? Does this comparison make sense? Estimate the OLS model and compare visually.

summary(fit) # display results confint(fit)

> confint(ln\_reg)

2.5 % 97.5 %

(Intercept) -0.8747160998 -0.705245483

rank 0.0007699596 0.001748841

mlhs -0.0444263166 0.011744021

mcol 0.0386017120 0.090751395

flhs -0.0458685446 0.010455416

fcol 0.0758117218 0.129267788

black 0.1373141968 0.215605806

hisp -0.0242035619 0.080688130

asian 0.0365901569 0.236982794

female -0.0347131147 0.009491982

rdsc 0.0013451053 0.004606261

vocab 0.0035216515 0.006452685

matsc 0.0106412979 0.013905177

nsib -0.0187269083 -0.009019931

# 95% CI for the coefficients exp(coef(fit))

exp(coef(ln\_reg))

(Intercept) rank mlhs mcol flhs fcol black hisp asian female

0.4538535 1.0012602 0.9837916 1.0668139 0.9824493 1.1079813 1.1929867 1.0286449 1.1465833 0.9874686

rdsc vocab matsc nsib

1.0029801 1.0049996 1.0123489 0.9862224

# exponentiated coefficients

exp(confint(fit))

exp(confint(ln\_reg))

2.5 % 97.5 %

(Intercept) 0.4169804 0.4939873

rank 1.0007703 1.0017504

mlhs 0.9565461 1.0118133

mcol 1.0393564 1.0949968

flhs 0.9551675 1.0105103

fcol 1.0787594 1.1379948

black 1.1471885 1.2406132

hisp 0.9760870 1.0840328

asian 1.0372678 1.2674193

female 0.9658825 1.0095372

rdsc 1.0013460 1.0046169

vocab 1.0035279 1.0064735

matsc 1.0106981 1.0140023

nsib 0.9814474 0.9910206

# 95% CI for exponentiated coefficients

predict(fit, type="response") # predicted values

residuals(fit, type="deviance") # residuals

* 1. If you believed this model, what would you do from a policy perspective to try to encourage more students to take the SAT?

1. Let’s return to the orange juice dataset and investigate how store demographics are related to demand.
   1. Run a LASSO model for the same model cross validated OLS that gave you the lowest MSE.
   2. What are the coefficients that are selected from the LASSO technique? Is it all of them?

i. What are the point estimates of those features?

* 1. Now set *alpha* in the glmnet function to .5. Does the predictive power of the model increase? Why or why not do you think?
  2. Now use all of this same code but start by withholding 10% of the OJ data for the training of LASSO.

i. Use this holdout data to perform out of sample testing of the LASSO model with the lambda with the min out of sample MSE.

Here is the code for using an existing model on new data:

predict(cvfit, newx = x[1:5,], s = "lambda.min")

See: <https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html> NOTE: you can get around the feature engineering by doing the following:

X <- model.matrix(formula, dataframe) # where formula is a general formula like you would run for OLS.

<https://www.rdocumentation.org/packages/MatrixModels/versions/0.4-1/topics/model.Matrix>