Crystal Mosley

Predict 422

Project #1

**Introduction**

This report examines the diabetes data in Efran et al (2003) to review the effects of ten predictor variables on a quantitate measure of disease progression on year after the baseline. The predictor variables are: age, sex, bmi (body mass index), map (avg blood pressure), and six blood serum measurements – tc, ldl, hdl, tch, ltg, and glu. There are 442 diabetes patients within this dataset. This set has been broken down into training data (75%) and test data (25%). The machine learning techniques that were used are: least squares regression, best subset selection using BIC (bayesian information criterion), ridge regression using 10-fold cross-validation, and lasso using 10-fold cross validation.

**Analysis**

Model 1 – Least Squares Regression using all 10 predictors

Coefficients

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 149.920 2.976 50.382 < 2e-16 \*\*\*

age -66.758 68.946 -0.968 0.33364

sex -304.651 69.847 -4.362 1.74e-05 \*\*\*

bmi 518.663 76.573 6.773 6.01e-11 \*\*\*

map 388.111 72.755 5.335 1.81e-07 \*\*\*

tc -815.268 537.549 -1.517 0.13034

ldl 387.604 439.162 0.883 0.37811

hdl 162.903 269.117 0.605 0.54539

tch 323.832 186.803 1.734 0.08396 .

ltg 673.620 206.888 3.256 0.00125 \*\*

glu 94.219 79.590 1.184 0.23737

---

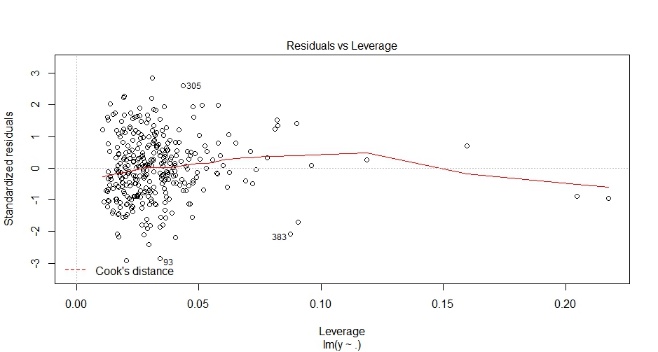
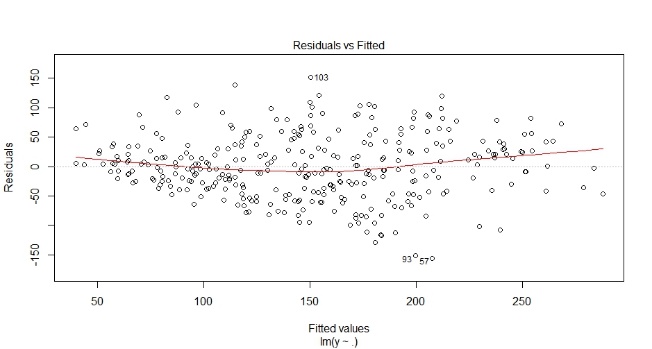
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 54.05 on 321 degrees of freedom

Multiple R-squared: 0.5213, Adjusted R-squared: 0.5064

F-statistic: 34.96 on 10 and 321 DF, p-value: < 2.2e-16

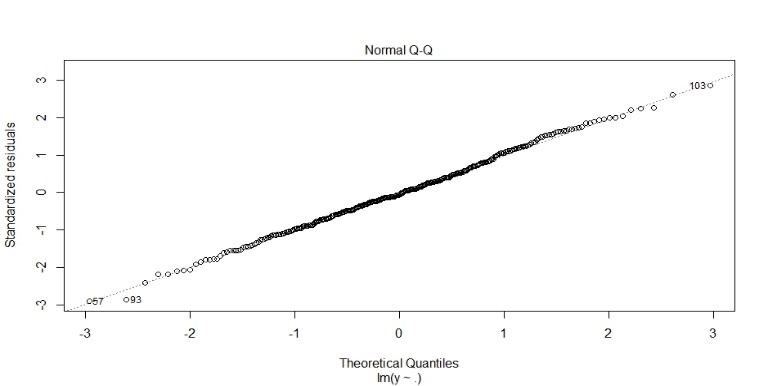
The highlighted predictors are the only significant using 0.05 as alpha.



The first plot shows the residuals vs fitted values. There looks to be no pattern, and all the values flowing are showing homoscedasticity.

The second plot shows the residuals vs leverage. There are 3 values to the far right that are outliers and may need to be examined more to see if they’re relevant.

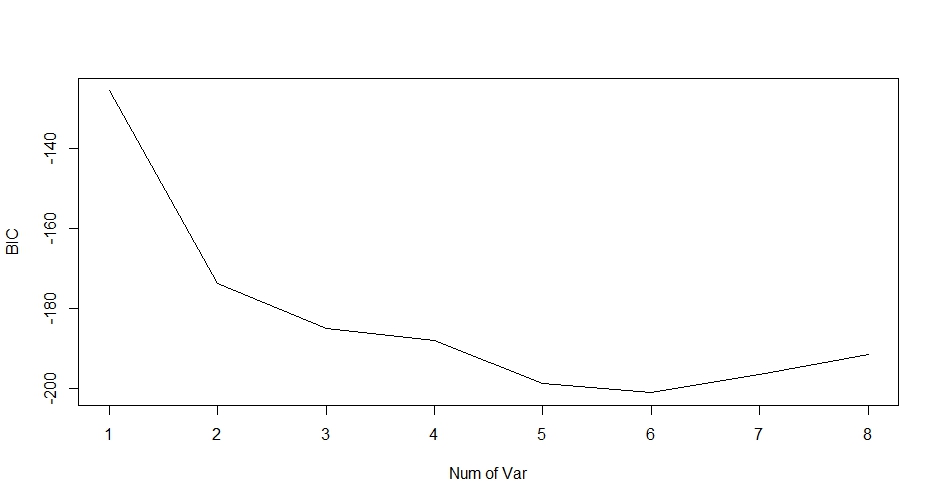
The next plot is a QQ normal plot of the residuals. All the points show a straight line, and nothing else.



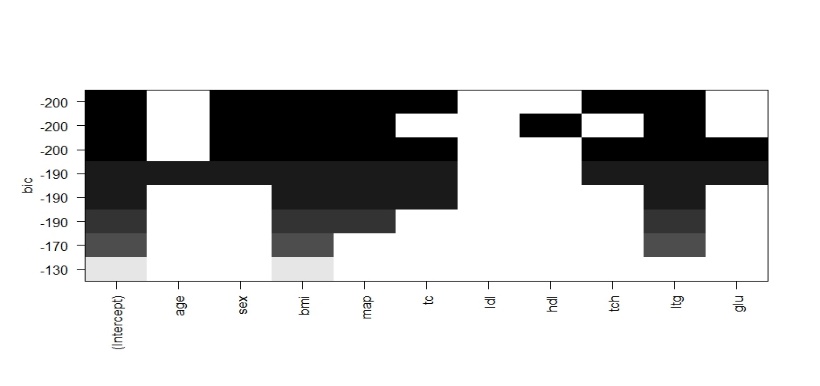
|  |  |
| --- | --- |
| Mean Prediction Error for Test | Standard Error of the Prediction Error |
| 31851.8 | 2581.989 |

Model 2 - Best subset selection using BIC to select the number of predictors

|  |  |
| --- | --- |
| Subset selection object  Call: regsubsets.formula(y ~ ., data.train)  10 Variables (and intercept)  Forced in Forced out  age FALSE FALSE  sex FALSE FALSE  bmi FALSE FALSE  map FALSE FALSE  tc FALSE FALSE  ldl FALSE FALSE  hdl FALSE FALSE  tch FALSE FALSE  ltg FALSE FALSE  glu FALSE FALSE | 1 subsets of each size up to 8  Selection Algorithm: exhaustive  age sex bmi map tc ldl hdl tch ltg glu  1 ( 1 ) " " " " "\*" " " " " " " " " " " " " " "  2 ( 1 ) " " " " "\*" " " " " " " " " " " "\*" " "  3 ( 1 ) " " " " "\*" "\*" " " " " " " " " "\*" " "  4 ( 1 ) " " " " "\*" "\*" "\*" " " " " " " "\*" " "  5 ( 1 ) " " "\*" "\*" "\*" " " " " "\*" " " "\*" " "  6 ( 1 ) " " "\*" "\*" "\*" "\*" " " " " "\*" "\*" " "  7 ( 1 ) " " "\*" "\*" "\*" "\*" " " " " "\*" "\*" "\*"  8 ( 1 ) "\*" "\*" "\*" "\*" "\*" " " " " "\*" "\*" "\*" |



The summary of the regsubsets fit shows the best 2-variable model contains bmi/ltg; the best 3-variable model contains bmi/map/ltg; the best 4-variable model contains bmi/map/tc/ltg ...and so on.



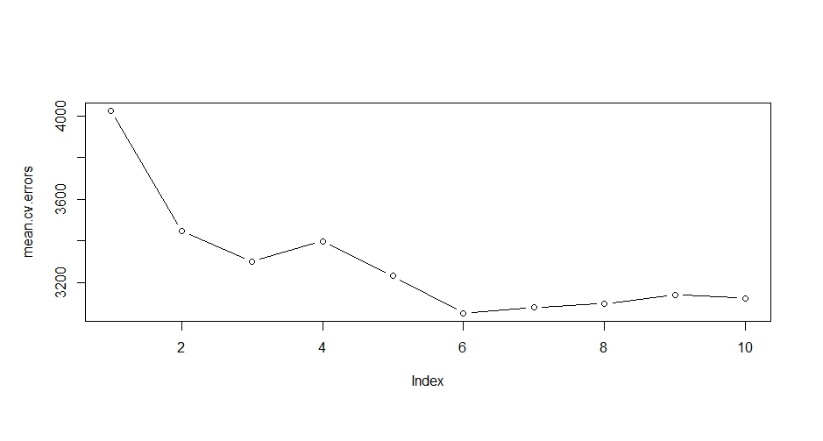
The model with the lowest BIC is the 6-variable model, -200; so, we use the coefficient associated with this 6-variable model.

Coefficients

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| (Intercept) | Sex | Bmi | Map | Tc | Tch | Ltg |
| 150.1166 | -306.0420 | 538.8274 | 389.0673 | -379.0379 | 332.6735 | 527.5658 |

|  |  |
| --- | --- |
| Mean Prediction Error for Test | Standard Error of the Prediction Error |
| 31851.8 | 2581.989 |

Model 3 - Best subset selection using 10-fold cross-validation to select the number of predictors



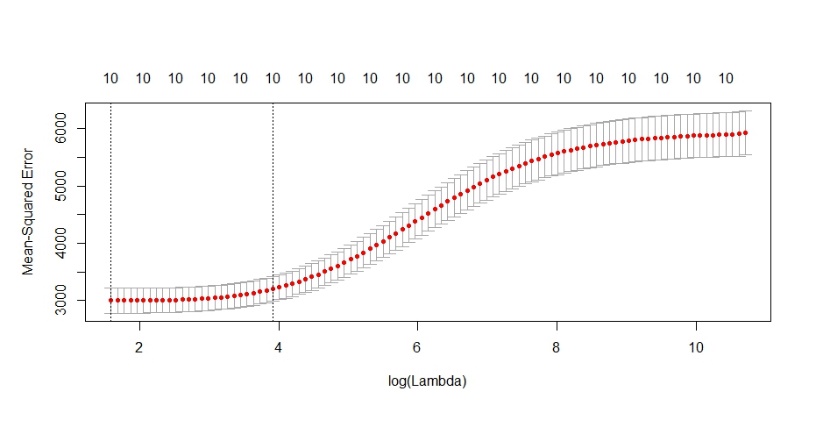
The 6 variables utilized from the best subset shows the lowest mean cross-validation error.

Coefficients

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| (Intercept) | Sex | Bmi | Map | Tc | Tch | Ltg |
| 150.1166 | -306.0420 | 538.8274 | 389.0673 | -379.0379 | 332.6735 | 527.5658 |

|  |  |
| --- | --- |
| Mean Prediction Error for Test | Standard Error of the Prediction Error |
| 31851.8 | 2581.989 |

Model 4 - Ridge regression modeling using 10-fold cross-validation to select the largest value of lambda



We used *lambda.1se* to find the largest value of lambda with the cv error being within 1 std. error of the minimum. This value is **50.19418**.

|  |  |
| --- | --- |
| Mean Prediction Error for Test | Standard Error of the Prediction Error |
| 31851.8 | 2581.989 |

Coefficients

(Intercept) 149.977557

age -8.557134

sex -149.136379

bmi 364.867848

map 257.973377

tc -28.362552

ldl -62.483558

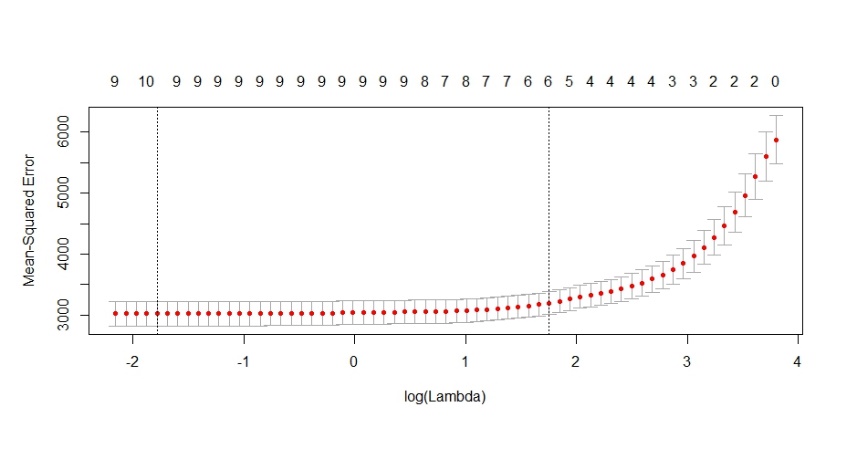
hdl -171.383585

tch 122.170756

ltg 299.972608

glu 134.577219

Model 5 - Lasso model using 10-fold cross-validation to select the largest value of lambda



We used *lambda.1se* to find the largest value of lambda with the cv error being within 1 std. error of the minimum. This value is **5.771111**.

|  |  |
| --- | --- |
| Mean Prediction Error for Test | Standard Error of the Prediction Error |
| 31851.8 | 2581.989 |

Coefficients

(Intercept) 149.926939

age .

sex -82.459870

bmi 500.664831

map 251.759713

tc .

ldl .

hdl -153.214079

tch .

ltg 388.281077

glu 5.506856

**Results**

|  |  |  |
| --- | --- | --- |
| Model | Mean Prediction Error for Test | Standard Error of the Prediction Error |
| Model 1 | 31851.8 | 2581.989 |
| Model 2 | 31851.8 | 2581.989 |
| Model 3 | 31851.8 | 2581.989 |
| Model 4 | 31851.8 | 2581.989 |
| Model 5 | 31851.8 | 2581.989 |

**Conclusion**

All the model results came out the same, which could mean I did something wrong within the code, but in the case I did not, this means that any model using the 6 predictor variables (sex, bmi, map, hdl, ltg, and glu) can be used for further investigation in predicting the quantitative measure of disease progression one year after the baseline.

**Reference**

Hastie, T., James, G., Tibshirani, R., Witten, D. *An Introduction to Statistical Learning, with Applications*

*in R* (2013). Springer New York Heidelberg Dordrecht London.

<http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Seventh%20Printing.pdf>

Accessed 29 July 2018.

**Appendix**

**R Code**

# Load the diabetes data

library(lars)

data(diabetes)

data.all <- data.frame(cbind(diabetes$x, y = diabetes$y))

# Partition the patients into two groups: training (75%) and test (25%)

n <- dim(data.all)[1] # sample size = 442

set.seed(1306) # set random number generator seed to enable

# repeatability of results

test <- sample(n, round(n/4)) # randomly sample 25% test

data.train <- data.all[-test,]

data.test <- data.all[test,]

x <- model.matrix(y ~ ., data = data.all)[,-1] # define predictor matrix

# excl intercept col of 1s

x.train <- x[-test,] # define training predictor matrix

x.test <- x[test,] # define test predictor matrix

y <- data.all$y # define response variable

y.train <- y[-test] # define training response variable

y.test <- y[test] # define test response variable

n.train <- dim(data.train)[1] # training sample size = 332

n.test <- dim(data.test)[1] # test sample size = 110

# load library's for lm and glmnet procedures

library(leaps)

library(glmnet)

# perform least squares regression using all variables

lm.fit=lm(y~.,data.train)

lm.fit

# model coeff est's

summary(lm.fit)

confint(lm.fit) #conf interval for coeff

# predict responses for test set

lm.fitpredict=predict(lm.fit, data.test)

# plot diag's

plot(lm.fit)

# mean predict error for Test

mean((lm.fitpredict=y.test)^2)

# std error of predict error

sd((lm.fitpredict=y.test)^2)/sqrt(n.test)

# use predict function to product confidence and prediction intervals for response, given predictor

predict(lm.fit, data.frame(data.train), interval = "confidence")

predict(lm.fit, data.frame(data.train), interval = "prediction")

plot(data.train, y)

abline(lm.fit)

# plot residuals

plot(predict(lm.fit), residuals(lm.fit))

# perform best subset selection using BIC to select # of predictors

lm.subset.bic=regsubsets(y~., data.train)

regfit.full=regsubsets(y~., data.train)

summary(regfit.full)

# best 2-variable model contains bmi/ltg

# best 3-variable model contains bmi/map/ltg

# best 4-variable model contains bmi/map/tc/ltg ...and so on.

# summaries of BIC

reg.summary=summary(lm.subset.bic)

reg.summary$bic

# plot BIC

plot(reg.summary$bic, xlab = "Num of Var", ylab = "BIC", type = "l")

# plot regsubsets

plot(regfit.full, scale = "bic")

# model with the lowest BIC is the 6-variable model, -200; so

# we use coef associated with this 6-variable model.

# plot coeff of BIC

coef(lm.subset.bic,6)

# use predict function for regsubsets, per book page 249

predict.regsubsets=function(object, newdata, id,...)

{

form=as.formula(object$call[[2]])

mat=model.matrix(form, newdata)

coefi=coef(object, id=id)

xvars=names(coefi)

mat[,xvars]%\*%coefi

}

# predict test set with BIC subset

lm.subset.bic.pred = predict(lm.subset.bic, data.test, id=6)

# mean predict error for Test

mean((lm.subset.bic.pred=y.test)^2)

# std error of predict error

sd((lm.subset.bic.pred=y.test)^2)/sqrt(n.test)

# perform best subset using 10-fold cross-validation

k=10

folds <- sample(1:k, nrow(data.train), replace = TRUE)

cv.errors=matrix(NA, k, 10, dimnames = list(NULL, paste(1:10)))

for(j in 1:k){

lm.subset.cv=regsubsets(y~., data.train[folds!=j,],

nvmax = 10)

for(i in 1:10){

pred=predict(lm.subset.cv, data.train[folds==j,], id=i)

cv.errors[j,i]=mean((data.train$y[folds==j]-pred)^2)

}

}

mean.cv.errors=apply(cv.errors, 2, mean)

mean.cv.errors

par(mfrow=c(1,1))

plot(mean.cv.errors, type = "b")

# model cv for best subset, 6

lm.subset.cv.best=regsubsets(y~., data.train, nvmax = 6)

# model coeff est's for cv best subset, 6

coef(lm.subset.cv.best, 6)

# predict test set for cv

lm.subset.cv.best.pred=predict(lm.subset.cv.best, data.test, id=6)

# mean predict error for Test

mean((lm.subset.cv.best.pred=y.test)^2)

# std error of predict error

sd((lm.subset.cv.best.pred=y.test)^2/sqrt(n.test))

# perform ridge regression using 10-fold cv

cv.out <- cv.glmnet(x.train, y.train, alpha = 0)

plot(cv.out)

biglam.ridge <- cv.out$lambda.1se

#lambda.1se - the most regularized model such that error is within one standard error of the minimum

biglam.ridge

bestlam=cv.out$lambda.min

#lambda.min - the value of \(\lambda\) that gives minimum mean cross-validated error

bestlam

# model coeff est's for cv ridge regression

ridge.model=glmnet(x.train, y.train, alpha = 0, lambda = 45.73507)

coef(ridge.model)

# predict test set for ridge regression

ridge.model.pred=predict(ridge.model, newx = x.test)

# mean predict error for Test

mean((ridge.model.pred=y.test)^2)

# std error of predict error

sd((ridge.model.pred=y.test)^2)/sqrt(n.test)

# perform lasso using 10-fold cv

cv.out <- cv.glmnet(x.train, y.train, alpha = 1)

plot(cv.out)

bestlam2=cv.out$lambda.min

bestlam2

biglam.lasso <- cv.out$lambda.1se

biglam.lasso

# model coeff est's for lasso

lasso.model=glmnet(x.train, y.train, alpha = 1, lambda = 5.771111)

coef(lasso.model)

# predict test set for lasso

lasso.model.pred=predict(lasso.model, newx = x.test)

# mean predict error for Test

mean((lasso.model.pred=y.test)^2)

# std error of predict error

sd((lasso.model.pred=y.test)^2)/sqrt(n.test)