Mini Project 4: Report

Raheel Ahmed

Rs170130

Section 1

Question 1

a. **Given that we are interested in comparing means, perform an exploratory analysis of the data. Which graphic is more appropriate for the task at hand: side-by-side box plots of observations from the two methods or a box plot of difference in observations from the two methods?**
   There are 72 patients in the oxygen saturation data. Each column corresponds to one of the two different methods for recording oxygen saturation data. Each row provides oxygen saturation data using the two methods for ONE individual patient. Additionally, a boxplot of difference in observations from the two methods (different columns for the same set of rows) will be a more useful metric of evaluation in examining the difference in population means for the two methods.
b. **Provide a point estimate ˆθ of θ, appropriate estimates of bias and standard error of the estimate, and a 95% confidence interval for θ. Interpret the results. (Before doing this exercise, you may want to refresh your memory about, e.g., the difference between a two-sample t-test and a paired t-test.)**
   a. Point estimate of ˆθ: -0.4125
   b. Bias of estimate: 9.65894e-15 (Essentially 0)
   c. Standard error of the estimate: 0.1427011
   d. 95% confidence interval for θ: (-0.5552011, -0.2697989)

   It appears that, in general, the oxygen saturation measured by method 1 (pulse oximetry) falls 0.4125 behind method 2 (osm). The estimates for an observation neither tend to overshoot nor tend to undershoot in terms of the difference in values between the two methods. The standard error allows us to confirm with 95% confidence that the difference in value between the two methods lies somewhere between -0.5552011 and -0.2697989.

c. **Write your own code to compute (nonparametric) bootstrap estimates of bias, standard error of ˆθ, and a 95% confidence interval for θ computed using the percentile method. How do these results compare with those in (b)?**
   a. Mean estimate of ˆθ: -0.415675
   b. Bias of estimate: -0.003175
   c. Standard error of the estimate: 0.0157955
   d. 95% confidence interval for θ: (-0.6750347, -0.1513542)

   Compared to b, we can see that the mean estimate is very similar. The bias found via bootstrapping does indicate that estimates tend to undershoot the difference in values found between the two methods but barely; it appears very close to 0 just like the value found for b. Similarly, the standard error is very close to what it was in part b. The confidence interval covers the same interval found in b but has a wider range of values that are required for 95% confidence.

d. **Repeat the computation in (c) using boot package and make sure the results match.**
   a. Mean estimate of ˆθ: -0.4125

b. Bias of estimate: 0.002316667
c. Standard error of the estimate: 0.01684293
d. 95% confidence interval for θ: (-0.6944, -0.1278)

The computation using the boot package is extremely similar to our own bootstrap version. The mean estimate is the same as in part b. The bias for b, c, and d are all close to 0. The standard error is similar in value ~ 0.015. The 95% confidence interval for θ covers a larger interval like in part c but is still very close to the original confidence interval for the difference in values for the two methods.

e. **What would you conclude about the population means of the two methods?**
I would conclude that they inform us that the pulse oximeter reads tend to report lower oxygen saturation than oxygen saturation monitors. Additionally, I would report that the difference in the values of the methods estimations do not usually overshoot or undershoot in value. I feel like, in comparison to the large range (70-100) that oxygen saturation values can be reported within, that the standard error is minimal and that there is no very significant difference in the measurements provided by either method.

Question 2

a. **Fit a linear regression model using all predictors and compute its test MSE.**
Shown in code, with test MSE = 0.8189307 without bias adjustment and 0.8167796 with bias adjustment.
b. **Use best-subset selection based on adjusted R2 to find the best linear regression model. Compute the test MSE of the best model.**
The best adjusted R2 value was found for the best model with 4 predictors. The test MSE of the best model was: 0.6145249 without bias adjustment and 0.6141675 with bias adjustment.
c. **Use forward stepwise selection based on adjusted R2 to find the best linear regression model. Compute the test MSE of the best model.**
The best adjusted R2 value was found for the best model with 4 predictors. The test MSE of the best model was: 0.6145249 without bias adjustment and 0.6141675 with bias adjustment.
d. **Use backward stepwise selection based on adjusted R2 to find the best linear regression model. Compute the test MSE of the best model.**
The best adjusted R2 value was found for the best model with 4 predictors. The test MSE of the best model was: 0.6145249 without bias adjustment and 0.6141675 with bias adjustment.
e. **Use ridge regression with penalty parameter chosen optimally via LOOCV to fit a linear regression model. Compute the test MSE of the model.**
Shown in code, with estimated test MSE = 0.5996864
f. **Use lasso with penalty parameter chosen optimally via LOOCV to fit a linear regression model. Compute the test MSE of the model.**
Shown in code, with estimated test MSE = 0.5522568
g. Make a tabular summary of the parameter estimates and test MSEs from (a) - (f). Compare the results. Which model(s) would you recommend?

Models	Parameter estimates	Test MSEs

Full	(Intercept) -0.685796, cancervol 0.069454, weight 0.001380, age -0.002799, benpros 0.087470, vesinv1 0.782623, capspen -0.026521, gleason 0.358153	0.8189307
Best SS	(Intercept) -0.65013, cancervol 0.06488, benpros 0.09136, vesinv1 0.68421, gleason 0.33376	0.6145249
Forward SS	(Intercept) -0.65013037, cancervol 0.06487865, benpros 0.09136387, vesinv1 0.68420905, gleason 0.33375914	0.6145249
Backward SS	(Intercept) -0.65013037, cancervol 0.06487865, benpros 0.09136387, vesinv1 0.68420905, gleason 0.33375914	0.6145249
Ridge Reg	(Intercept) -0.029640992 Cancervol 0.032075764 weight 0.001150147 age 0.004362445 benpros 0.035502480 vesinv1 0.442132423 capspen 0.029310835 gleason 0.247425068	0.5996864
Lasso	(Intercept) -1.899770e-01 cancervol 5.871548e-02 weight 7.861665e-05 age . benpros 6.037468e-02 vesinv1 5.641867e-01 capspen . gleason 2.877950e-01	0.5522568

I would recommend using a model based off of ridge regression or Lasso, although it is good to try all the methods out, in this case, we can see that stepwise selection and best subset selection come up with the same best model with the best adjusted R^2 value and end up having the same LOOCV due to formulating the same ideal models. In this case, it appears that lasso and ridge regression both shrink the unnecessary features towards 0. The continuous nature of this shrinking allows for both optimization and for development of reasonable models with estimated test MSEs that were noticeably lower as well.

Question 3

a. **Fit a logistic regression model using all predictors and compute its test error rate.**
   Shown in code, with an estimated test error rate of 0.249000 without bias adjustment and 0.248446 with bias adjustment.
b. **Use best-subset selection based on AIC to find the best logistic regression model. Compute the test error rate of the best model.**
   Note: I used fewer than all predictors for my best-subset selection. Bestglm requires 15 or fewer predictors and 21 predictors took a really long time to even attempt to run, so I selected commonly useful features from my other stepwise selection models. The test error rate I found was 0.240000 without bias adjustment and 0.238906 with bias adjustment.
c. **Use forward stepwise selection based on AIC to find the best logistic regression model. Compute the test error rate of the best model.**

Using forward stepwise selection based on AIC, I found that the test error rate of the logistic regression model was 0.240000 without bias adjustment and 0.238906 with bias adjustment.

d. **Use backward stepwise selection based on AIC to find the best logistic regression model. Compute the test error rate of the best model.**

   Using backward stepwise selection based on AIC, I found that the test error rate of the logistic regression model was 0.240000 without bias adjustment and 0.238906 with bias adjustment.

e. **Use ridge regression with penalty parameter chosen optimally via LOOCV to fit a logistic regression model. Compute the test error rate of the model.**

   Test error rate was 0.1532243.

f. **Use lasso with penalty parameter chosen optimally via LOOCV to fit a logistic regression model. Compute the test error rate of the model.**

   Test error rate was 0.1521238.

g. Make a tabular summary of the parameter estimates and test error rates from (a) - (f). Compare the results. Which model(s) would you recommend? How does this recommendation compare with what you recommended in Mini Project 3?

Models	Parameter estimates	Test MSEs
Full	(Intercept) checkingstatus1A12 checkingstatus1A13 checkingstatus1A14 duration historyA31 historyA32 0.4005027032 -0.3748533845 -0.9656768269 -1.7118879549 0.0278633245 0.1433777014 -0.5861135632 historyA33 historyA34 purposeA41 purposeA410 purposeA42 purposeA43 purposeA44 -0.8531614098 -1.4357715801 -1.6664669545 -1.4887859369 -0.7916103762 -0.8915834370 -0.5227827424 purposeA45 purposeA46 purposeA48 purposeA49 amount savingsA62 savingsA63 -0.2163959040 0.0362838335 -2.0594327737 -0.7400868495 0.0001282747 -0.3577405779 -0.3760728784 savingsA64 savingsA65 employA72 employA73 employA74 employA75 installment -1.3391988399 -0.9466891929 -0.0669104342 -0.1828309822 -0.8310018182 -0.2766245208 0.3300898152 statusA92 statusA93 statusA94 othersA102 othersA103 residence propertyA122 -0.2754548085 -0.8160779448 -0.3670718835 0.4360476126 -0.9786160157 0.0047760501 0.2814382403 propertyA123 propertyA124 age otherplansA142 otherplansA143 housingA152 housingA153 0.1945346780 0.7304477374 -0.0145354910 -0.1232005664 -0.6463286585 -0.4436209848 -0.6838601772 cards jobA172 jobA173 jobA174 liable teleA192 foreignA202 0.2720759275 0.5361303832 0.5547174978 0.4794752439 0.2646713870 -0.3000079729 -1.3922159416	0.249000
Best SS	(Intercept) checkingstatus1A12 checkingstatus1A13 checkingstatus1A14 historyA31 historyA32 historyA33 1.7495411395 -0.3900151943 -1.0240813337 -1.7177165353 -0.1187723563 -0.8303100692 -0.9097303516 historyA34 purposeA41 purposeA410 purposeA42 purposeA43 purposeA44 purposeA45 -1.4917085142 -1.6072584850 -1.4349202508 -0.7404978116 -0.9194787129 -0.5250944762 -0.1424475387 purposeA46 purposeA48 purposeA49 savingsA62 savingsA63 savingsA64 savingsA65 0.1435654512 -2.1643060184 -0.7826591225 -0.3282181546 -0.4303583691 -1.2894344545 -0.9628458451 othersA102 othersA103 statusA92 statusA93 statusA94 otherplansA142 otherplansA143 0.4874390675 -1.0404263012 -0.2872095837 -0.8227884511 -0.4169133414 -0.0786395281 -0.6994941104 foreignA202 housingA152 housingA153 teleA192 duration installment amount -1.3824571920 -0.4415028980 -0.1496754072 -0.2794110592 0.0256787050 0.3299308314 0.0001294275 age -0.0130932622	0.240000
Forward SS	(Intercept) checkingstatus1A12 checkingstatus1A13 checkingstatus1A14 duration historyA31 historyA32 1.7495411395 -0.3900151943 -1.0240813337 -1.7177165353 0.0256787050 -0.1187723563 -0.8303100692 historyA33 historyA34 purposeA41 purposeA410 purposeA42 purposeA43 purposeA44 -0.9097303516 -1.4917085142 -1.6072584850 -1.4349202508 -0.7404978116 -0.9194787129 -0.5250944762 purposeA45 purposeA46 purposeA48 purposeA49 savingsA62 savingsA63 savingsA64 -0.1424475387 0.1435654512 -2.1643060184 -0.7826591225 -0.3282181546 -0.4303583691 -1.2894344545 savingsA65 othersA102 othersA103 installment statusA92 statusA93 statusA94 -0.9628458451 0.4874390675 -1.0404263012 0.3299308314 -0.2872095837 -0.8227884511 -0.4169133414 amount otherplansA142 otherplansA143 foreignA202 age housingA152 housingA153 0.0001294275 -0.0786395281 -0.6994941104 -1.3824571920 -0.0130932622 -0.4415028980 -0.1496754072 teleA192 -0.2794110592	0.240000
Backward SS	(Intercept) checkingstatus1A12 checkingstatus1A13 checkingstatus1A14 duration historyA31 historyA32 1.7495411395 -0.3900151943 -1.0240813337 -1.7177165353 0.0256787050 -0.1187723563 -0.8303100692 historyA33 historyA34 purposeA41 purposeA410 purposeA42 purposeA43 purposeA44 -0.9097303516 -1.4917085142 -1.6072584850 -1.4349202508 -0.7404978116 -0.9194787129 -0.5250944762 purposeA45 purposeA46 purposeA48 purposeA49 amount savingsA62 savingsA63 -0.1424475387 0.1435654512 -2.1643060184 -0.7826591225 0.0001294275 -0.3282181546 -0.4303583691 savingsA64 savingsA65 installment statusA92 statusA93 statusA94 othersA102 -1.2894344545 -0.9628458451 0.3299308314 -0.2872095837 -0.8227884511 -0.4169133414 0.4874390675 othersA103 age otherplansA142 otherplansA143 housingA152 housingA153 teleA192 -1.0404263012 -0.0130932622 -0.0786395281 -0.6994941104 -0.4415028980 -0.1496754072 -0.2794110592 foreignA202 -1.3824571920	0.240000

Ridge Reg	(Intercept) 4.208060e-01 checkingstatus1A12 -3.872375e-02 checkingstatus1A13 -1.405668e-01 checkingstatus1A14 -2.234135e-01 duration 4.435963e-03 historyA31 1.159767e-01 historyA32 -2.373473e-02 historyA33 -6.257237e-02 historyA34 -1.239285e-01 purposeA41 -1.729866e-01 purposeA410 -1.621511e-01 purposeA42 -6.944319e-02 purposeA43 -9.354871e-02 purposeA44 -3.189319e-02 purposeA45 4.806682e-03 purposeA46 4.618861e-02 purposeA48 -1.761064e-01 purposeA49 -5.296754e-02 amount 1.538143e-05 savingsA62 -3.690909e-02 savingsA63 -7.792623e-02 savingsA64 -1.360088e-01 savingsA65 -1.135197e-01 employA72 3.767911e-02 employA73 8.212216e-03 employA74 -7.339887e-02 employA75 -1.574570e-02 installment 3.644497e-02 statusA92 4.372850e-03 statusA93 -6.338909e-02 statusA94 -2.776965e-02 othersA102 7.092241e-02 othersA103 -1.397183e-01 residence 1.205833e-03 propertyA122 3.384623e-02 propertyA123 2.535824e-02 propertyA124 7.908186e-02 age -1.565441e-03 otherplansA142 6.621589e-03 otherplansA143 -7.261012e-02 housingA152 -5.903265e-02 housingA153 -5.237314e-02 cards 2.832277e-02 jobA172 1.200414e-03 jobA173 1.369389e-02 jobA174 1.072915e-02 liable 2.717486e-02 teleA192 -3.999467e-02 foreignA202 -1.189836e-01	Using MSE: 0.1532243
Lasso	(Intercept) 5.045467e-01 checkingstatus1A12 -6.776554e-02 checkingstatus1A13 -1.734811e-01 checkingstatus1A14 -2.663229e-01 duration 4.652504e-03 historyA31 7.819270e-02 historyA32 -5.558568e-02 historyA33 -9.102933e-02 historyA34 -1.631554e-01 purposeA41 -1.973603e-01 purposeA410 -1.842948e-01 purposeA42 -8.441645e-02 purposeA43 -1.074131e-01 purposeA44 -2.204106e-02 purposeA45 . purposeA46 3.195660e-02 purposeA48 -1.807585e-01 purposeA49 -6.877760e-02 amount 1.724927e-05 savingsA62 -2.807689e-02 savingsA63 -6.757026e-02 savingsA64 -1.330953e-01 savingsA65 -1.153288e-01 employA72 2.541419e-02 employA73 . employA74 -8.055320e-02 employA75 -1.295629e-02 installment 4.012529e-02 statusA92 . statusA93 -7.026409e-02 statusA94 -2.522027e-02 othersA102 5.935305e-02 othersA103 -1.525607e-01 residence . propertyA122 2.410370e-02 propertyA123 1.718060e-02 propertyA124 7.198598e-02 age -1.577220e-03 otherplansA142 . otherplansA143 -7.589441e-02 housingA152 -5.718509e-02 housingA153 -5.636087e-02 cards 2.902000e-02 jobA172 . jobA173 6.327735e-03 jobA174 . liable 2.042530e-02 teleA192 -3.661234e-02 foreignA202 -1.224698e-01	Using MSE: 0.1521238

Similarly to our last attempt at feature subset selection, we notice that stepwise models often yield the same model when using the same metric of evaluation, even exhaustively. I would again recommend that a person try all of these approaches, but in this case, it appears that ridge regression and lasso are ideal for model selection. They both have low MSE and the continuous nature of their shrinking help optimize better than the step-wise nature of the other approaches in this case.

This recommendation is different from what I formulated in Mini Project 3 in terms of the number of features. Both the ridge regression and lasso models had many more features used compared to what I selected in the other project. However, both of these models and my custom model did share many features that were significant, such as checkingstatus1, history, duration, savings, installment, and otherplans.

Code Portion

Question 1

```
library(boot)

# Obtain oxygen monitoring data from the oxygen saturation txt file
o2_data <- read.table("oxygen_saturation.txt", header = T, sep="\t")
# Obtain data from pulse oximeter and oxygen saturation monitor separately
pos_data <- o2_data[, 1]
osm_data <- o2_data[, 2]
# Find the vector of differences between the two methods per individual
patient
diff_data <- pos_data - osm_data
# See the number of patients and the
# distribution of values found using the two methods
nrow(o2_data)
hist(pos_data)
hist(osm_data)
# Print out the mean values for pulse oximeter readings
# and oxygen saturation monitors separately
mean(pos_data)
mean(osm_data)

# Point estimate of the difference in means between the two methods
pt_estim_mean <- mean(pos_data) - mean(osm_data)

# Bias of the vector produced from looking at the difference in methods per
patient (diff_data)
mean(diff_data) - pt_estim_mean
# Standard error of diff_data
st_err <- sd(diff_data) / sqrt(72)
# Confidence interval via percentile approach: quantile(diff_data, c(0.025,
0.975))
# 95% confidence interval of diff_data
c(mean(diff_data) - st_err, mean(diff_data) + st_err)


# Number of entries
n <- nrow(o2_data)
# number of resamples
b <- 1000

# Track estimates of difference of means
estimates <- c()
for(i in 1:b){
  # For each resample, sample from 1 to n with replacement
  indices_sampled <- sample(1:n, n, replace = T)
  # Use these indices to select rows to calculate the difference of means
from
  estimates <- c(estimates,
                 mean(pos_data[indices_sampled]) -
mean(osm_data[indices_sampled])
                )
}
# Metrics you can evaluate
```

```
# Mean estimate of the difference of values between the two methods
mean(estimates)
# Variance found via bootstrapping
var(estimates)
# Bias found via bootstrapping
mean(estimates) - pt_estim_mean
# Standard Error found via bootstrapping
sd(estimates) / sqrt(72)
# Confidence Interval (via percentile approach)
quantile(estimates, c(.025, .975))

# Establish difference of means function
mean.fn <- function(x, indices) {
  # Return difference of means over two methods for certain indices
  result <- mean(x[indices,1]) - mean(x[indices,2])
  return(result)
}
# Test the function for the original data
mean.fn(o2_data, 1:nrow(o2_data))

# Conduct a boostrapping using the boot package.
mean.boot <- boot(o2_data, mean.fn, R = 1000)
mean.boot
# Estimate for difference of means
mean.boot$t0
# Bias for estimate of difference of means
mean(mean.boot$t) - mean.boot$t0
# Standard errorof the estimate of difference of means
sd(mean.boot$t) / sqrt(72)
# Confidence intervals produced with 95% levels
boot.ci(mean.boot, type = "perc")
```

Question 2

```
library(boot)
library(leaps)
library(glmnet)

# Read in prostate cancer data
pc_data <- read.csv("prostate_cancer.csv")
# Eliminate subject number feature
pc_data <- pc_data[,-1]
# Treat vesinv as a qualitative variable
pc_data$vesinv <- factor(pc_data$vesinv, order=F, levels = c(0, 1))

# Conduct a natural log transformation on the response
# to adjust it's distribution to something more appropriate.
pc_data[, 1] <- log(pc_data[, 1])
hist(pc_data[, 1])

# Make a full model of a linear regression with psa as response and all
features as predictors
# Calculate test MSE via LOOCV
full_model <- glm(psa ~ ., data = pc_data)
cv.err <- cv.glm(pc_data, full_model)
```

```
cv.err$delta

# Find a reasonable subset of features to implement a linear regression model
with
# via the best-subset selection accounting for the best adjusted R^2.
regfit_full = regsubsets(psa ~ ., data = pc_data, nvmax = 7)
regfit_summ <- summary(regfit_full)
which.max(regfit_summ$adjr2)
coef(regfit_full, 4)
# Find the test MSE via LOOCV
cv.glm(pc_data, glm(psa ~ cancervol + benpros + vesinv + gleason,
data=pc_data))$delta

# Find a reasonable subset of features to implement a model with using
forward
# subset selection with best adjusted R^2 value.
fit.fwd = regsubsets(psa ~ ., data = pc_data, nvmax = 7, method = "forward")
summary(fit.fwd)
which.max(summary(fit.fwd)$adjr2)
coef(fit.fwd, 4)
cv.glm(pc_data, glm(psa ~ cancervol + benpros + vesinv + gleason,
data=pc_data))$delta

# Find a reasonable subset of features to implement a model with using
backward
# subset selection with best adjusted R^2 value.
fit.bwd = regsubsets(psa ~ ., data = pc_data, nvmax = 7, method = "backward")
summary(fit.bwd)
which.max(summary(fit.bwd)$adjr2)
coef(fit.bwd, 4)
cv.glm(pc_data, glm(psa ~ cancervol + benpros + vesinv + gleason,
data=pc_data))$delta

# Select response and feature data as y and X respectively
y <- pc_data$psa
X <- model.matrix(psa ~ ., pc_data)[, -1]
# Set up a grid of potential lambda values
grid <- 10^seq(10, -2, length = 100)
# Using alpha = 0, conduct a ridge regression
ridge.mod <- glmnet(X, y, alpha = 0)
# Use LOOCV to determine the best penalty parameter
cv_ridge <- cv.glmnet(X, y, alpha = 0)
best_ridge_lambda <- cv_ridge$lambda.min
best_ridge_lambda

# Use the best lambda to find the best ridge regression model
best_ridgemod <- glmnet(X, y, alpha = 0, lambda = best_ridge_lambda)
coef(best_ridgemod)
# Predict the values using the best ridge regression model and find the MSE
y_pred <- predict(ridge.mod, s = best_ridge_lambda, newx = X)
mean((y_pred - y)^2)

# Using alpha=1, conduct lasso
lasso.mod <- glmnet(X, y, alpha=1)
# Use LOOCV to find the best penalty parameter
cv_lasso <- cv.glmnet(X, y, alpha=1)
best_lasso_lambda <- cv_lasso$lambda.min
```

```
best_lasso_lambda
# Make the best lasso model with the best lambda value
best_lassomod <- glmnet(X, y, alpha=1, lambda=best_lasso_lambda)
coef(best_lassomod)
# Predict the appropriate values using the best lasso model and calculate MSE
y_pred <- predict(lasso.mod, s = best_lasso_lambda, newx = X)
mean((y_pred-y)^2)
```

Question 3

```
library(boot)
library(MASS)
library(glmulti)
library(glmnet)

# Read in german credit dataset and store as a dataframe
german_data <- read.csv("germancredit.csv", stringsAsFactors = T)
# Produce a logistic regression model using all the predictors
all_lr <- glm(Default ~ ., data = german_data, family = "binomial")
summary(all_lr)
coef(all_lr)

# Produce a logistic regression model using no predictors (yields intercept
only model)
empty_lr <- glm(Default ~ 1, data=german_data, family = "binomial")
summary(empty_lr)


# Use boot package to estimate LOOCV for log-reg models
# Make cost function
cost <- function(r, pi = 0){mean(abs(r - pi) > 0.5)}
# Calculate LOOCV for both the full model and the null model
all_lr.err <- cv.glm(german_data, all_lr, cost, K=nrow(german_data))
all_lr.err$delta
empty_lr.err <- cv.glm(german_data, empty_lr, cost, K=nrow(german_data))
empty_lr.err$delta

# Find best subset selection model with fewer predictors to improve run time.
# AIC is our method of evaluation, run exhaustive checks for the best model.
best_sub <- glmulti(Default ~ checkingstatus1 + duration + history + purpose
+ savings +
                      others + installment + status + amount + otherplans +
foreign +
                      age + housing + tele, data=german_data, level=1,
        method="h", crit="aic", confsetsize = 2,
        plotty=F, report=F,
        fitfunction = "glm", family=binomial)
# Plug in the formula to the glm function and run a logistic regression
best_lr <- glm(best_sub@formulas[[1]], data=german_data, family="binomial")
summary(best_lr)
coef(best_lr)
# Print LOOCV estimate of test error.
best_lr.err <- cv.glm(german_data, best_lr, cost, K=nrow(german_data))
best_lr.err$delta
```

```
# Find forward stepwise selection model with AIC as our method of evaluation
# I started with the null model and progressed forward with a specified scope
forward_step <- stepAIC(empty_lr,
                        scope = list(lower=empty_lr, upper=all_lr),
                        direction = "forward")
# Feed the formula found into the glm function and conduct a logistic
regression
forward_lr <- glm(Default ~ checkingstatus1 + duration + history + purpose +
savings +
                  others + installment + status + amount + otherplans +
foreign +
                  age + housing + tele, data = german_data,
family="binomial")
summary(forward_lr)
coef(forward_lr)
# Calculate LOOCV estimate of test error
forward_lr.err <- cv.glm(german_data, forward_lr, cost, K=nrow(german_data))
forward_lr.err$delta

# Find backward stepwise selection model with AIC as our method of evaluation
# I started with the full model and moved backward with the specified scope
backward_step <- stepAIC(all_lr,
                        scope = list(lower=empty_lr, upper=all_lr),
                        direction = "backward")
# Feed the formula found into the glm function and conduct a logistic
regression
backward_lr <- glm(Default ~ checkingstatus1 + duration + history + purpose +
amount +
                   savings + installment + status + others + age +
otherplans +
                   housing + tele + foreign, data = german_data, family =
"binomial")
summary(backward_lr)
coef(backward_lr)
# Calculate LOOCV estimate of test error
backward_lr.err <- cv.glm(german_data, backward_lr, cost,
K=nrow(german_data))
backward_lr.err$delta

# Assign response and feature matrix to y and X respectively
y <- german_data$Default
X <- model.matrix(Default ~ ., german_data)[, -1]
# Set up a grid of potential lambda values
grid <- 10^seq(10, -2, length = 100)
# Conduct a ridge regression where alpha=0
ridge.mod <- glmnet(X, y, alpha = 0)
# Use LOOCV to determine the best penalty parameter
cv_ridge <- cv.glmnet(X, y, alpha = 0)
best_ridge_lambda <- cv_ridge$lambda.min
best_ridge_lambda
# Make the best ridge regression model using the best lambda
best_ridgemod <- glmnet(X, y, alpha = 0, lambda = best_ridge_lambda)
coef(best_ridgemod)

# Predict the values of the training data using the best ridge regression
model
y_pred <- predict(ridge.mod, s = best_ridge_lambda, newx = X)
```

```r
# Compare the values to the true values to find test error
mean((y_pred - y)^2)

# Conduct a lasso model with alpha=1
lasso.mod <- glmnet(X, y, alpha=1)
# Use LOOCV to find the best lambda
cv_lasso <- cv.glmnet(X, y, alpha=1)
best_lasso_lambda <- cv_lasso$lambda.min
best_lasso_lambda
# Formulate the best lasso model with the best lasso lambda
best_lassomod <- glmnet(X, y, alpha=1, lambda=best_lasso_lambda)
coef(best_lassomod)
# Predict the values of our training data and compare to the true values to
find the test error
y_pred <- predict(lasso.mod, s = best_lasso_lambda, newx = X)
mean((y_pred-y)^2)
```