Ivy Fong
October 28, 2018

Breast Cancer Regression Analysis

Task 1 – Data Preparation

To setup, I loaded the glmnet package to be used. I read the data from the csv file into R using the read.csv function specifying that there is no header in the data file and to return the character "?" for missing values in the file. I also used the na.omit function to create a dataset bc with only complete information (without observations with missing values).

Task 2 – Data Subset and Descriptive Statistics

I created a subset bcn with only those without recurrence and a subset bcr with only those with recurrence. Using the mean and sd functions respectively, I calculated the mean and standard deviation of the time variable (V3) for both groups. Lastly, I created a boxplot for the time variable for the two groups from the original bc dataset.

For the group without recurrence, the mean of the disease-free time is 53.58108 and the standard deviation is 34.91935. For the group with recurrence, the mean of the recurrence time is 25.56522 and the standard deviation is 22.72703. A boxplot was created to display the distribution of the time variable (V3) for the two groups (Figure 1).

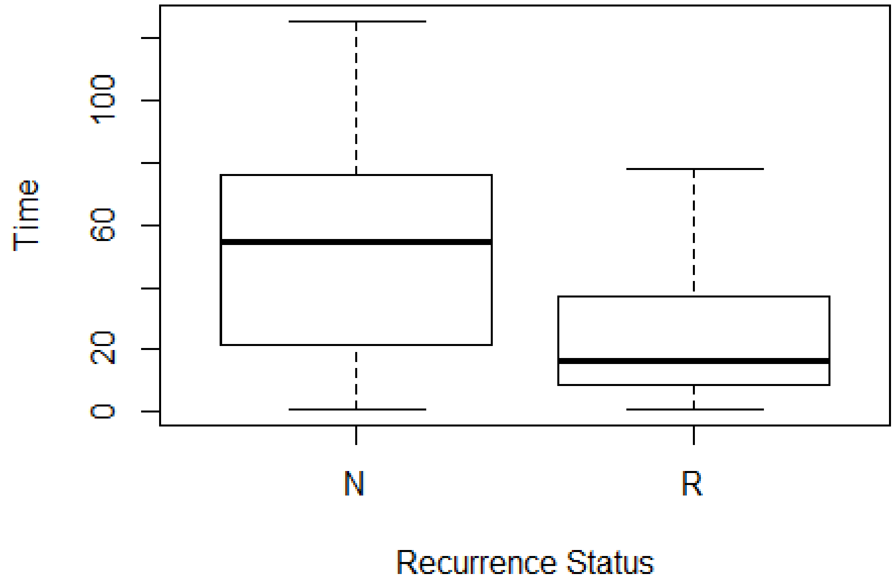

Figure 1. Boxplot for the time variable (V3) for the groups with (R) and without (N) recurrence.

Task 3 – Ridge Regression Model with Default Lambda Values

I specified the predictor (V4-13) and outcome (V3) variables in the objects x and y, respectively, then trained a ridge regression model using the default grid of values for the lambda parameter in the glmnet function and the alpha=0 argument. I then plotted the coefficients of the predictors for different levels of log lambda using the plot function and the xvar=”lambda” argument. For the ridge regression model, as the value of log lambda increases, the coefficients either increase or decrease to approach 0 (Figure 2).

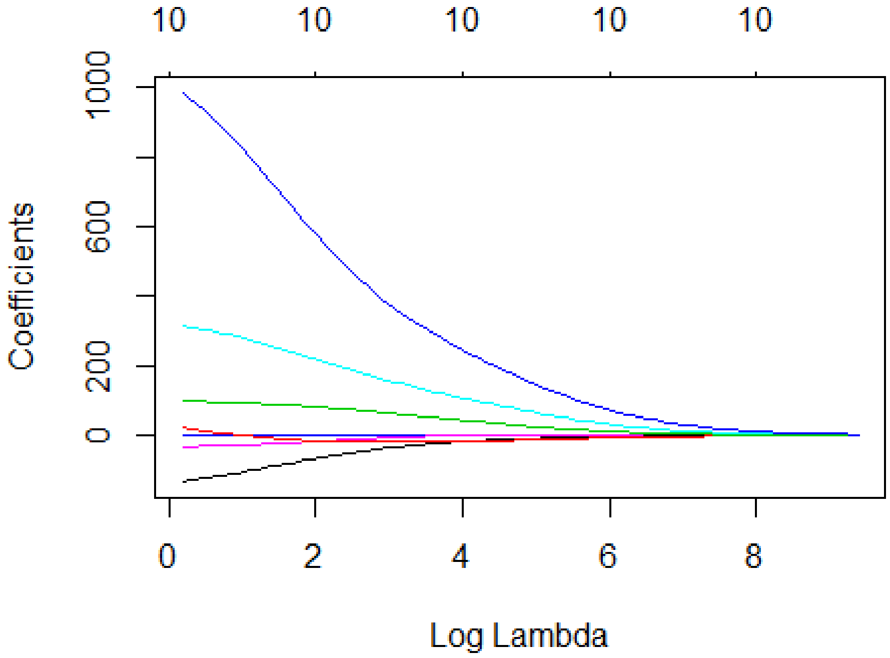

Figure 2. Coefficients of the predictors vs. log lambda for the ridge regression model.

Task 4 – Cross Validation and Optimal Lambda Value for the Ridge Regression Model

Using a 5-fold cross validation estimate calculated using the cv.glmnet function, and the alpha=0 and nfolds=5 arguments, I created a plot showing the MSE against the values of log(lambda) for the ridge regression model. Then, I printed the optimal value for lambda that minimizes the MSE, as well as the coefficients of the predictors for the optimal lambda value. The optimal lambda value that minimizes the MSE is 13.37345 (Figure 3). The predictor coefficients for the optimal lambda value are listed in Table 1.

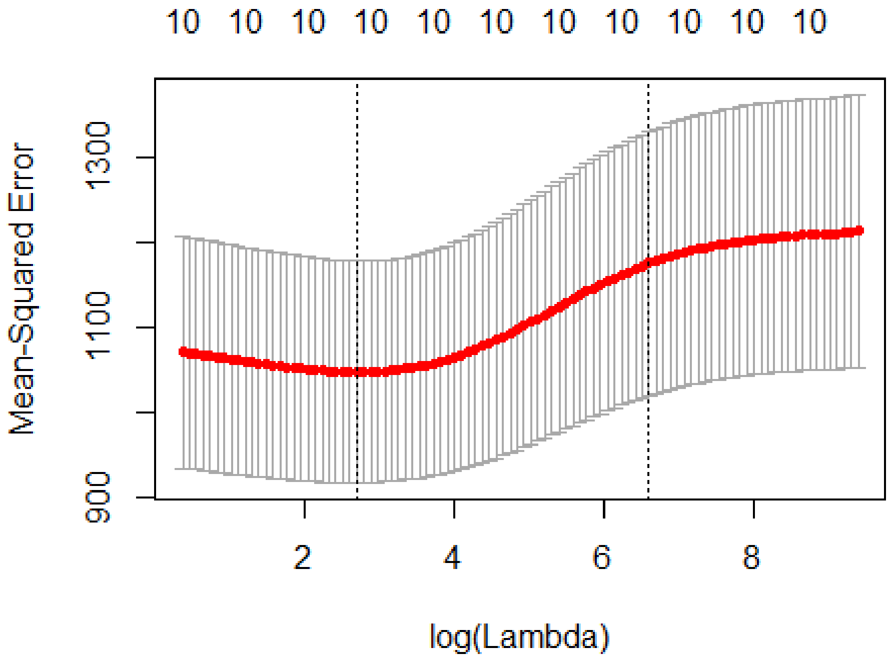

Figure 3. MSE vs. log(lambda) for the ridge regression model.

Table 1. Predictor coefficients for the optimal lambda value for the ridge regression model.

Variable Number	**Variable Name**	**Coefficient**
V4	mean radius	-0.401431533
V5	mean texture	-1.489660254
V6	mean perimeter	-0.086699448
V7	mean area	-0.003732896
V8	mean smoothness	177.468091758
V9	mean compactness	-13.158970300
V10	mean concavity	-48.074388890
V11	mean concave points	-11.901168233
V12	mean symmetry	69.336268882
V13	mean fractal dimension	485.974704557

Task 5 – MSE for the Ridge Regression Model

I used the mean function to calculate the MSE on the whole set of the non-recurrent group for the ridge regression model using the optimal lambda value. The MSE using the optimal lambda value is 985.8034.

Task 6 – Lasso Model

Lasso Model with Default Lambda Values

Using the same x and y objects with the predictor (V4-13) and outcome (V3) variables specified above in task 3, I trained a lasso model using the default grid of values for the lambda parameter in the glmnet function and the alpha=1 argument. I then plotted the coefficients of the predictors for different levels of log lambda using the plot function and the xvar=”lambda” argument. For the lasso model, as the value of log lambda increases, the coefficients either increase or decrease to 0 (Figure 4).

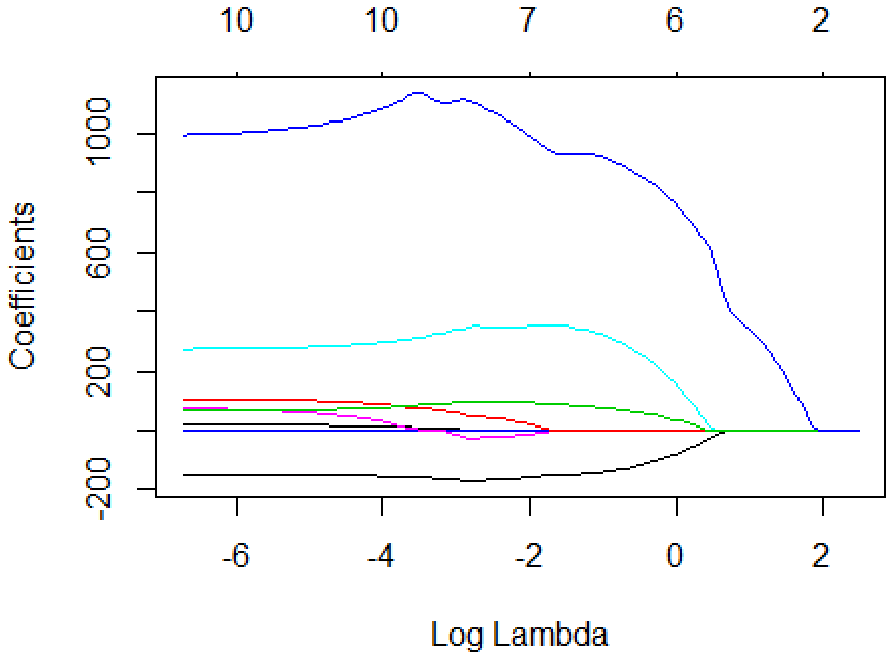

Figure 4. Coefficients of the predictors vs. log lambda for the lasso model.

Cross Validation and Optimal Lambda Value for the Lasso Model

Using a 5-fold cross validation estimate calculated using the cv.glmnet function, and the alpha=1 and nfolds=5 arguments, I created a plot showing the MSE against the values of log(lambda) for the lasso model. Then, I printed the optimal value for lambda that minimizes the MSE, as well as the coefficients of the predictors and the non-zero (selected) features for the optimal lambda value.

The optimal lambda value for the lasso model is 0.7373547 (Figure 5). The predictor coefficients for the optimal lambda value are listed in Table 2. The selected features for the optimal lambda value are mean texture (V5), mean perimeter (V6), mean smoothness (V8), mean concavity (V10), mean symmetry (V12), and mean fractal dimension (V13).

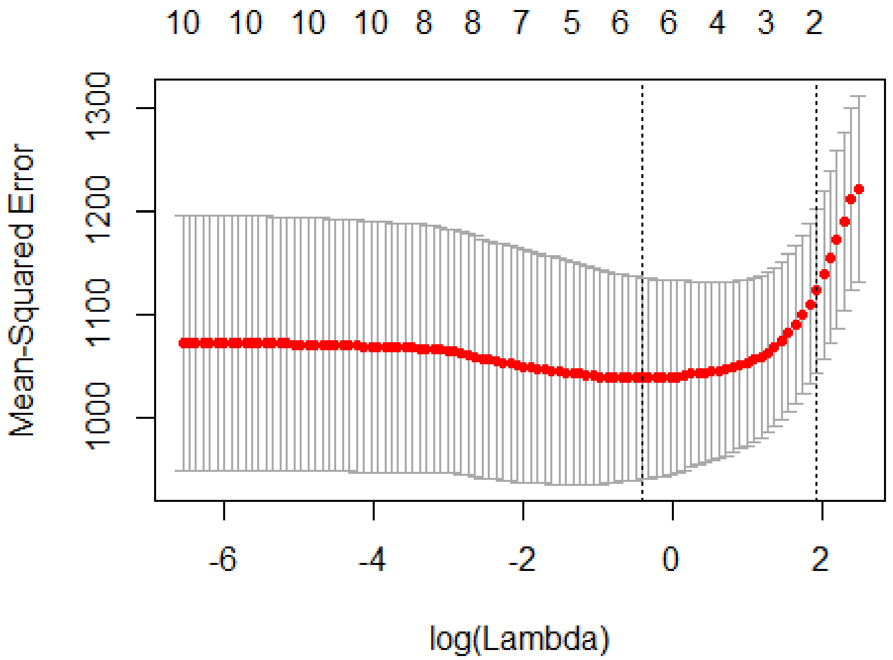

Figure 5. MSE vs. log(lambda) for the lasso model.

Table 2. Predictor coefficients for the optimal lambda value for the lasso model.

Variable Number	Variable Name	Coefficient
V4	mean radius	.
V5	mean texture	-1.85388150
V6	mean perimeter	-0.06579758
V7	mean area	.
V8	mean smoothness	224.85581587

V9	mean compactness	.
V10	mean concavity	-104.62778262
V11	mean concave points	.
V12	mean symmetry	52.30459849
V13	mean fractal dimension	826.01279857

MSE for the Lasso Model

I used the mean function to calculate the MSE on the whole set of the non-recurrent group for the lasso model using the optimal lambda value. The MSE using the optimal lambda value is 972.7765.

Task 7 – Model Comparison and Performance

Based on the results generated above, the lasso method seems to have performed better because it has a lower MSE using the optimal lambda value. The MSE for the ridge regression model using the optimal lambda is 985.8034, while the MSE for the lasso model using the optimal lambda is lower at 972.7765. Lasso tends to perform better than ridge regression when only some features are strong predictors, as is the case here – only 6 out of 10 features were selected for the optimal lambda value.

The model comparison process must be separate from the parameter tuning process. Different training, validation and test sets must be used. Thus, to compare the performance of the two prediction methods on these data, we could use the cross validation, leave one out, validation set, and or nested cross-validation methods.

Appendix: R Code

```
#load packages to be used
library(glmnet)
```

Task 1 - Read the data into R, making sure that you code the missing values properly. The character “?” is used for denoting missing values in the .csv file. Notice that there is no header in the data file. (1 point)

```
bc <- read.csv("bc_data.csv", header=F, na.strings="?") #read csv data
into R, specify no variable names, return ? for missing values
bc <- na.omit(bc) #only keep observations with complete information
```

Task 2 - Report descriptive statistics and make a box plot for the time variable for the two groups (with and without recurrence). Make a subset of the original dataset with only those without recurrence. (2 points)

```
bcn <- subset(bc, bc$V2=="N") #create subset with only those without r
ecurrence
bcr <- subset(bc, bc$V2=="R") #create subset with only those with recu
rrence

mean(bcn$V3) #calculate mean of time variable for those without recurr
ence = 53.58108
sd(bcn$V3) #calculate sd of time variable for those without recurrence
= 34.91935

mean(bcr$V3) #calculate mean of time variable for those with recurrenc
e = 25.56522
sd(bcr$V3) #calculate sd of time variable for those with recurrence =
22.72703
```

```
boxplot(bc$V3~bc$V2, xlab="Recurrence Status", ylab="Time") #create
boxplot for the time variable for the two groups
```

Task 3 - Using as predictors the mean values of the above described (a) - (j) features (which as found in columns 4-13), train a ridge regression model using the default grid of values for the lambda parameter in the glmnet R function. Make a plot showing the coefficients of these predictors for different levels of regularization. Comment on the results. (3 points)

```
x <- model.matrix(V3~V4+V5+V6+V7+V8+V9+V10+V11+V12+V13,bcn) #specify t
he predictor variables V4-13
y <- bcn$V3 #specify the outcome variable V3

rr.mod <- glmnet(x,y,family="gaussian",alpha=0) #train a ridge regress
ion model using the default values
```

```
plot(rr.mod, xvar="lambda") #plot the coefficients for different level
s of regularization
```

Task 4 - Using a 5-fold cross-validation estimate and report the optimal value for lambda (i.e. that minimizes the MSE). Make a plot showing the MSE against the values of log(lambda). Report the coefficients of the predictors for the optimal lambda value. (3 points)

```
cv.rr <- cv.glmnet(x,y,alpha=0,nfolds=5) #perform 5-fold cross-validat
ion
plot(cv.rr) #plot MSE versus log(lambda)

cv.rr$lambda.min #print the lambda associated with the min MSE = 13.37
345
coef.min <- coef(cv.rr, s = "lambda.min") #print the unstandardized co
efficients associated with the optimal lambda
coef.min
```

Task 5 - Calculate the MSE on the whole set of the non-recurrent group for the model using the optimal lambda value. (2 points)

```
mean((y-predict(rr.mod, newx=x, s= cv.rr$lambda.min))^2) #calculate th
e MSE of the whole data set using the optimal lamda = 985.8034
```

Task 6 - Repeat tasks 3-5 above but this time using the lasso method. This time report also what the selected features are for the optimal lambda value. (7 points)

```
l.mod <- glmnet(x,y,family="gaussian",alpha=1) #train a lasso model us
ing the default values
plot(l.mod, xvar="lambda") #plot the coefficients for different levels
of regularization

cv.l <- cv.glmnet(x,y,alpha=1,nfolds=5) #perform 5-fold cross-validati
on
plot(cv.l) #plot MSE versus log(lambda)

cv.l$lambda.min #print the lambda associated with the min MSE = 0.7373
547
coef.min2 <- coef(cv.l, s = "lambda.min") #print the unstandardized co
efficients associated with the optimal lambda
coef.min2
row.names(coef.min2)[as.vector(coef.min2)!=0] #print only non-zero coe
fficients = V5, V6, V8, V10, V12, V13

mean((y-predict(l.mod, newx=x, s= cv.l$lambda.min))^2) #calculate the
MSE of the whole data set using the optimal lamda = 972.7765
```