Zhenbang Wu / Chao Xu

Prof. D.A. Forsyth

CS498 Applied Machine Learning

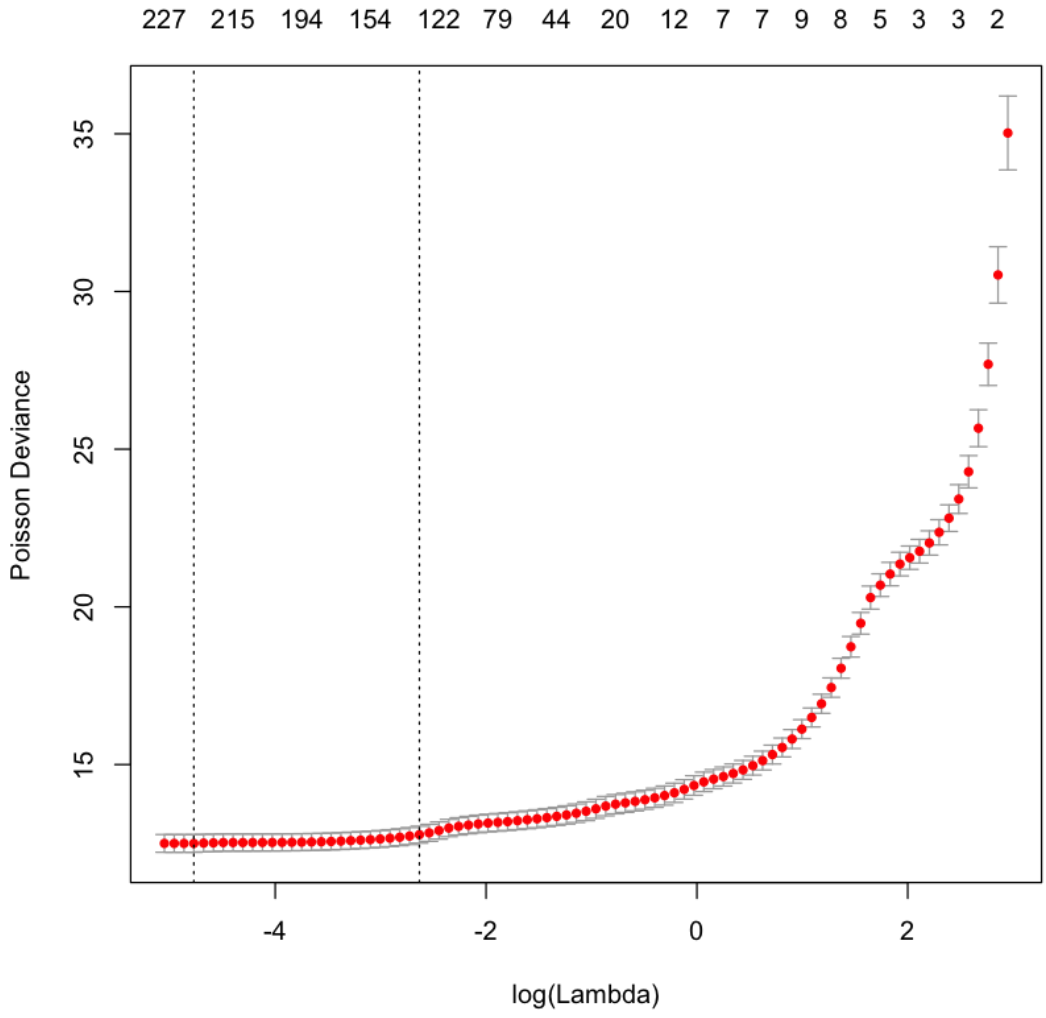
28 October 2018

Homework 7

Page 1: 12.3

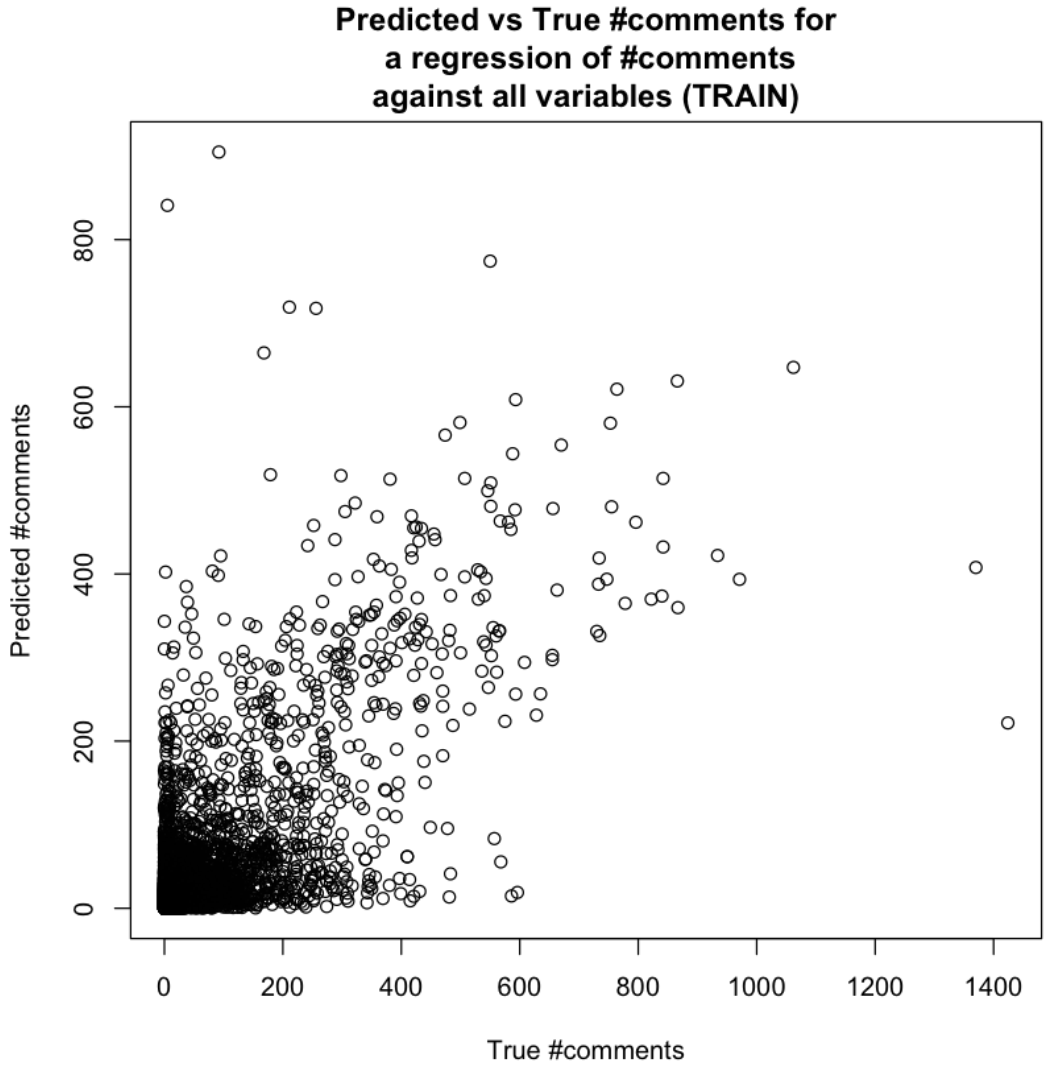
Plot of the Cross-Validated Deviance of the Model Against

the Regularization Variable



Page 2: 12.3

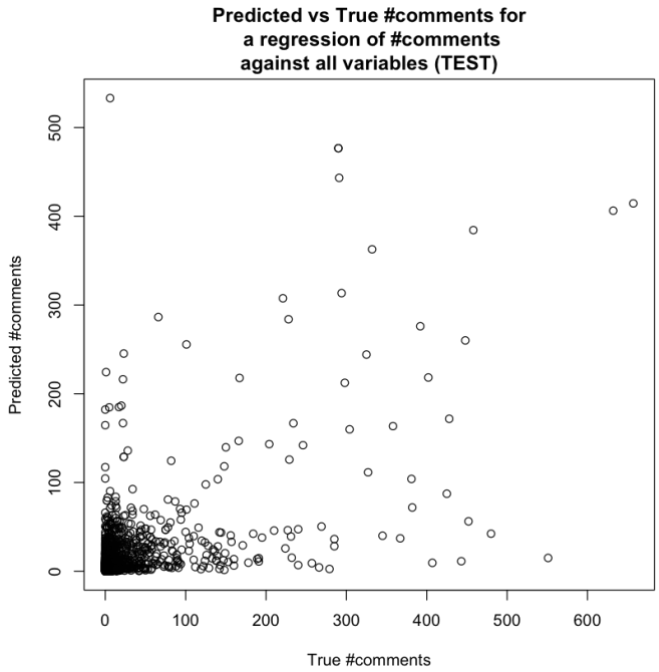
Scatter Plot of True Values vs Predicted Values for Training Data



Value of regularization constant we choose lambda. min = 0.00846243499862599

Page 3: 12.3

Scatter Plot of True Values vs Predicted Values for Test Data



Value of regularization constant we choose $\text{lambda.min} = 0.00846243499862599$

Q1: Compare the two plots and comment on the performance of the model.

Answer: Both of two predictions/plots are not satisfactory. The model's performance is very awful. The predictions don't fit the true value well, the residual is very high. But the predictions seem to work better for those blog posts which have a large number of comments. This phenomenon is more obvious in the scatter plot for the train set: the plot tends to fit $y = x$ for hot blog posts (hot: high number of comments). So for comparison, it works a little better on the train set (the first plot). After all, it is trained on the train set.

Q2: Provide comment on why this regression is difficult.

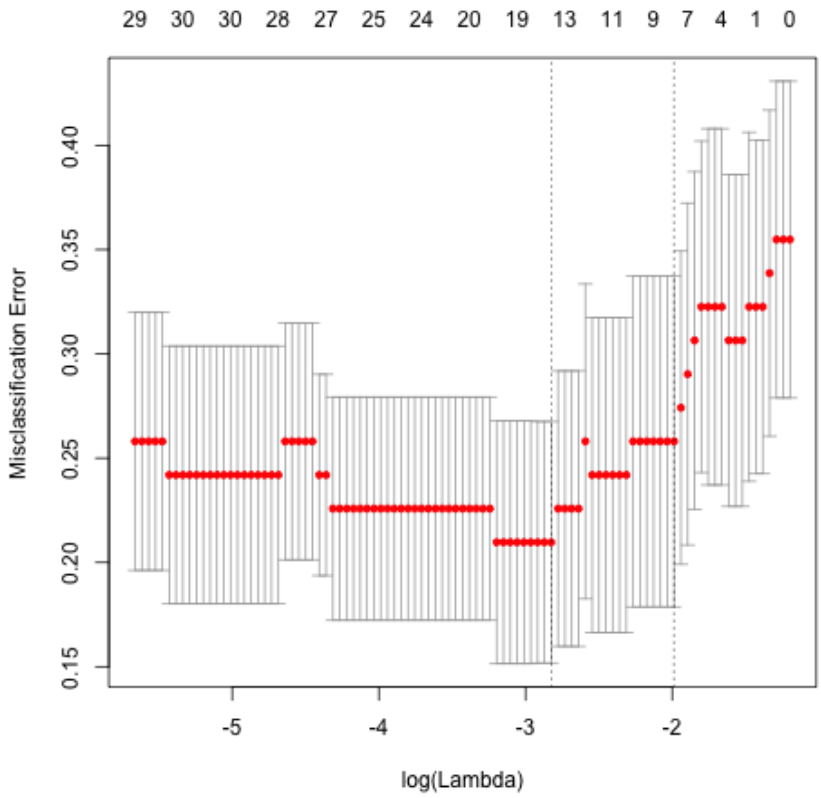
Answer: 1. This regression supposes that it is a linear model. But the relationship between features and the label might is not very linear. e.g. If the number of comments decline in an exponential way. A linear model won't work well.

2. There are a lot of repetitive features in the data. Though it seems the number of instances is very large, too much repeat in some features will actually affect the training effectiveness.

3. There're a lot of zeros in the label/target. It shows data quality might be not good. At least, as the first reason says, relationships between variables are not very linear.

Page 4: 12.4

Plot of the Classification Error of the Model against the Regularization Variable



Value of regularization constant we choose = 0.0593194

Classification accuracy = 0.9193548

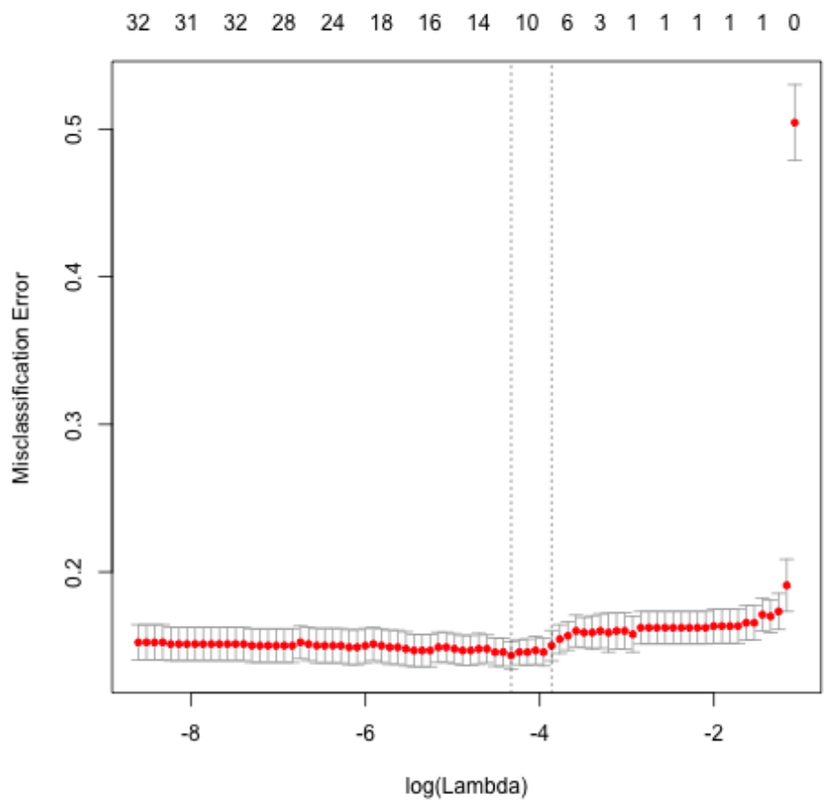
Baseline = $tumor\%$ = 0.6451613

Comment on the model performance compared with the baseline: If we use the baseline of predicting the most common class, we will get an accuracy about 0.6451613. And if we use the logistic regression and the lasso, we will get an accuracy about 0.9193548, which is much better than the baseline. As a remark, we think the reason why we can get a high accuracy by logistic regression and the lasso is that we get enough observations and the two types of tissues (normal and tumor) are very different from each other.

Page 5: 12.5

Predict Gender with the Features

Plot of the Classification Error of the Model against the Regularization Variable



Value of regularization constant we choose = 0.0131977

Classification accuracy = 0.8620309

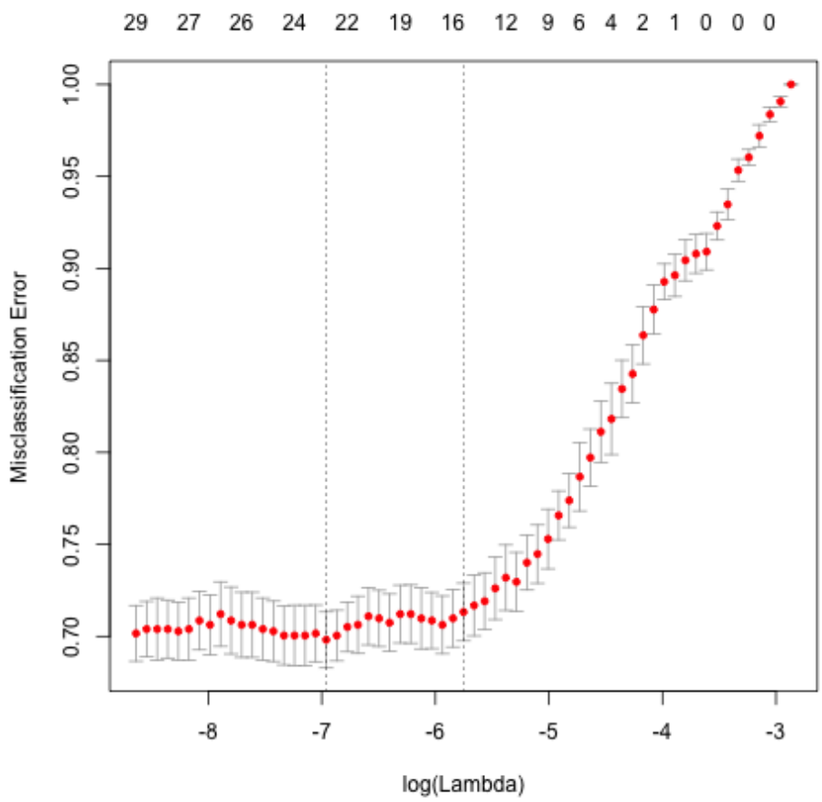
Baseline = $male\%$ = 0.50883

Comment on the model performance compared with the baseline: If we use the baseline of predicting the most common gender for all mice, we will get an accuracy about 0.50883. And if we use the logistic regression and the lasso, we will get an accuracy about 0.8620309, which is much better than the baseline.

Page 6: 12.5

Predict the Strain of a Mouse with the Features

Plot of the Classification Error of the Model against the Regularization Variable



Value of regularization constant we choose $= 0.0009481192$

Classification accuracy $= 0.6200466$

Baseline $= accuracy\ of\ predicting\ at\ random = 0.022222$ (total number of strains is 55 and 10 of them are removed because of having fewer than 10 rows)

Comment on the model performance compared with the baseline: If we use the baseline of predicting a strain at random, we will get an accuracy about 0.022222. And if we use the logistic regression and the lasso, we will get an accuracy about 0.6200466, which though it not a high value, is still much better than the baseline. As a remark, we think that the reason why the accuracy is low is that there are too many types of strains and some of the strains do not have enough observations. This significantly increases the influences of outliers, which decrease the final accuracy. What's more, intuitively, some strains may be quite similar to each other, and thus also increases the difficulty of predicting.

Page 7: Code Screenshot

Using glmnet

```
########### 12.3 ###########
# train a generalized linear poisson model
fit <- cv.glmnet(X, Y, family = "poisson")
# predict on the training set (predict() from glmnet)
trainPred = predict(fit, newx = X, s = fit$lambda.min, type = "response")
# predict on the test set (predict() from glmnet)
testPred = predict(fit, newx = testX, s = fit$lambda.min, type = "response")
########### 12.4 ###########
# Build a logistic regression of the label (normal vs tumor) against
# the expression levels for those genes.
model<-cv.glmnet(genes_mat, tissues, family = "binomial", type.measure ="class")
tissues_prediction = predict(model, genes_mat, type = "class", s = model$lambda.min)
########### 12.5 (a) ###########
gender_model<-cv.glmnet(variables, gender, family = "binomial", type.measure ="class")
gender_prediction = predict(gender_model, variables, type = "class", s = gender_model$lambda.min)
########### 12.5 (b) ###########
# Preditct strain using measurements chosen, using multinomial logistic regression
# and the lasso
strain_model<-cv.glmnet(variables, strain, family = "multinomial", type.measure ="class")
strain_prediction = predict(strain_model, variables, type = "class", s = strain_model$lambda.min)
```

Making the Plot

```
############ 12.3 ############
# a plot of the cross-validated deviance of the model
plot(fit)
# a scatter plot of true values vs predicted values for the training set
plot(Y, trainPred,
     xlab = "True #comments", ylab = "Predicted #comments",
     main = "Predicted vs True #comments for\n a regression of #comments
     against all variables (TRAIN)")
# a scatter plot of true values vs predicted values for the test set
plot(testY, testPred,
     xlab = "True #comments", ylab = "Predicted #comments",
     main = "Predicted vs True #comments for\n a regression of #comments
     against all variables (TEST)")
############ 12.4, 12.5 ############
plot(model)
```

Data Preprocess

```
########### 12.3 ###########
# INPUT: read blog data (train & test)
trainData<-read.csv('BlogFeedback/blogData_train.csv')
X<-as.matrix(trainData[, -c(281)])
Y<-as.matrix(trainData[, 281])
# combine test data
# using "cat *.csv > all_test.csv" in terminal
testData<-read.csv('BlogFeedback/all_test.csv')
testX<-as.matrix(testData[, -c(281)])
testY<-as.matrix(testData[, 281])
```

```
########### 12.4 ###########
# reading data
genes<-transpose(read.table('I2000.txt'))
genes_mat<-matrix(unlist(genes), nrow = nrow(genes))
tissues<-unlist(read.table('Tissues.txt'))
# cluster the tissues into normal and tumor
#  - a positive sign to a normal tissue => 0
#  - a negative sign to a tumor tissue => 1
tissues[tissues>0]<-0
tissues[tissues<0]<-1
```

```
########### 12.5 (a) ###########
# reading data
data_orig<-read.csv('Crusio1.csv')
data_mat<-matrix(unlist(data_orig), nrow = nrow(data_orig))
# extract useful data
#  col 2: gender, col 4 - col 41: measurements
data_mat<-data_mat[,c(2, 4:41)]
# deal with N/A
data_mat<-na.omit(data_mat)
# extract x and y
gender<-data_mat[,1]
variables<-data_mat[,-c(1)]
```

```
########### 12.5 (b) ###########
# extract useful data
#  col 1: strain, col 4 - col 41: measurements
data_mat<-data_mat[,c(1, 4:41)]
# deal with N/A
data_mat<-na.omit(data_mat)
# drop strains with fewer than 10 rows
small_strain<-c()
for (strain_idx in c(1:nlevels(data_orig$strain))) {
  if (sum(data_mat[,1] == strain_idx) < 10) {
    small_strain<-c(small_strain, c(strain_idx))
  }
}
for (strain_idx in small_strain) {
  keep<-data_mat[,1] != strain_idx
  data_mat<-data_mat[keep,]
}
# extract x and y
strain<-data_mat[,1]
variables<-data_mat[,-c(1)]
```

In [1]:
```
# import necessary libraries
library(glmnet)
library(doParallel)
# accelerate using parallel computing
registerDoParallel(makeCluster(detectCores()))
```

...

In [7]:
```
# INPUT: read blog data (train & test)
trainData <- read.csv('BlogFeedback/blogData_train.csv')
X <- as.matrix(trainData[, -c(281)])
Y <- as.matrix(trainData[, 281])
# combine test data using "cat *.csv > all_test.csv" in terminal
# Reference: Piazza post No. 603
testData <- read.csv('BlogFeedback/all_test.csv')
testX <- as.matrix(testData[, -c(281)])
testY <- as.matrix(testData[, 281])
```

In [8]:
```
# train a generalized linear poisson model
fit <- cv.glmnet(X, Y, family = "poisson")
```

In [9]:
```
# a plot of the cross-validated deviance of the model
plot(fit)
```

...

In [41]:
```
# predict on the training set (predict() from glmnet)
trainPred = predict(fit, newx = X, s = fit$lambda.min, type = "respons
```

In [42]:
```
# a scatter plot of true values vs predicted values for the training s
plot(Y, trainPred,
xlab = "True #comments", ylab = "Predicted #comments",
main = "Predicted vs True #comments for\n a regression of #comments
against all variables (TRAIN)")
```

...

In [43]:
```
# predict on the test set (predict() from glmnet)
testPred = predict(fit, newx = testX, s = fit$lambda.min, type = "resp
```

In [44]:
```
# a scatter plot of true values vs predicted values for the test set
plot(testY, testPred,
xlab = "True #comments", ylab = "Predicted #comments",
main = "Predicted vs True #comments for\n a regression of #comments
against all variables (TEST)")
```

...

```r
# CS498AML HW7 12.4
# written by Zhanbang Wu and Chao Xu on Oct 26 2018

#####################################################################
# Initialization
#####################################################################
# set the workspace here
setwd('/Users/Zachary/playground/CS498/HW7/12.4')
getwd()
# include library
library("glmnet")
library("data.table")

#####################################################################
# Data Pre-processing
#####################################################################
# reading data
genes<-transpose(read.table('I2000.txt'))
genes_mat<-matrix(unlist(genes), nrow = nrow(genes))
tissues<-unlist(read.table('Tissues.txt'))
# cluster the tissues into normal and tumor
#  - a positive sign to a normal tissue => 0
#  - a negative sign to a tumor tissue => 1
tissues[tissues>0]<-0
tissues[tissues<0]<-1
# calculate normal and tumor percents
tumor_percent<-tissues==1
tumor_percent<-sum(tumor_percent) / (sum(tumor_percent) +
sum(!tumor_percent))
normal_percent<-(1 - tumor_percent)

#####################################################################
# Regression
#####################################################################
# Build a logistic regression of the label (normal vs tumor) against
# the expression levels for those genes.
model<-cv.glmnet(genes_mat, tissues, family = "binomial", type.measure
="class")
png(filename='output/12.4.png')
plot(model)
dev.off()
tissues_prediction = predict(model, genes_mat, type = "class", s =
model$lambda.min)

#####################################################################
# Evaluation
```

```
#################################################################
gotright<-tissues == tissues_prediction
accuray<-sum(gotright) / (sum(gotright) + sum(!gotright))
```

```r
# CS498AML HW7 12.5
# written by Zhanbang Wu and Chao Xu on Oct 26 2018

#####################################################################
# Initialization
#####################################################################
# set the workspace here
setwd('/Users/Zachary/playground/CS498/HW7/12.5')
getwd()
# include library
library("glmnet")

########################## 12.5 (a) ##########################

#####################################################################
# Data Pre-processing
#####################################################################
# reading data
data_orig<-read.csv('Crusio1.csv')
data_mat<-matrix(unlist(data_orig), nrow = nrow(data_orig))
# extract useful data
#  col 2: gender
#  col 4 - col 41: measurements
data_mat<-data_mat[,c(2, 4:41)]
# deal with N/A
data_mat<-na.omit(data_mat)
# extract x and y
gender<-data_mat[,1]
variables<-data_mat[,-c(1)]
# calculate male and female percents
female_percent<-gender==1
female_percent<-sum(female_percent) / (sum(female_percent) +
sum(!female_percent))
male_percent<-(1 - female_percent)

#####################################################################
# Regression
#####################################################################
# Preditct gender using measurements chosen, using a logistic regression
# and the lasso
gender_model<-cv.glmnet(variables, gender, family = "binomial",
type.measure ="class")
png(filename ='output/12.5a.png')
plot(gender_model)
dev.off()
```

```
gender_prediction = predict(gender_model, variables, type = "class", s =
gender_model$lambda.min)

#################################################################
# Evaluation
#################################################################
gender_gotright<-gender == gender_prediction
gender_accuray<-sum(gender_gotright) / (sum(gender_gotright) +
sum(!gender_gotright))

########################## 12.5 (b) ##########################

#################################################################
# Data Pre-processing
#################################################################
# reading data
data_orig<-read.csv('Crusio1.csv')
data_mat<-matrix(unlist(data_orig), nrow = nrow(data_orig))
# extract useful data
#  col 1: strain
#  col 4 - col 41: measurements
data_mat<-data_mat[,c(1, 4:41)]
# deal with N/A
data_mat<-na.omit(data_mat)
# drop strains with fewer than 10 rows
small_strain<-c()
for (strain_idx in c(1:nlevels(data_orig$strain))) {
  if (sum(data_mat[,1] == strain_idx) < 10) {
    small_strain<-c(small_strain, c(strain_idx))
  }
}
for (strain_idx in small_strain) {
  keep<-data_mat[,1] != strain_idx
  data_mat<-data_mat[keep,]
}
# extract x and y
strain<-data_mat[,1]
variables<-data_mat[,-c(1)]

#################################################################
# Regression
#################################################################
# Preditct strain using measurements chosen, using multinomial logistic
regression
# and the lasso
strain_model<-cv.glmnet(variables, strain, family = "multinomial",
type.measure ="class")
png(filename ='output/12.5b.png')
plot(strain_model)
```

```
dev.off()
strain_prediction = predict(strain_model, variables, type = "class", s =
strain_model$lambda.min)

#################################################################
# Evaluation
#################################################################
strain_gotright<-strain == strain_prediction
strain_accuray<-sum(strain_gotright) / (sum(strain_gotright) +
sum(!strain_gotright))
```