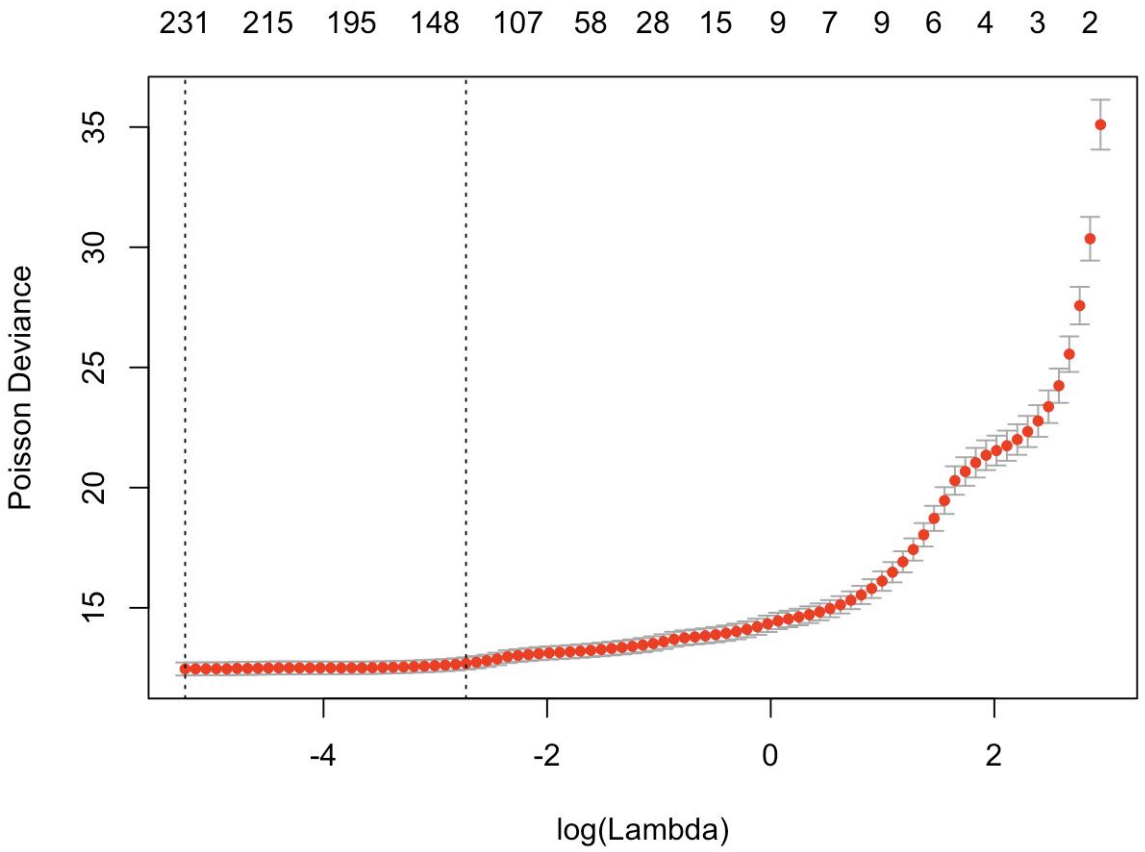
CS498 AML - Home Work 7

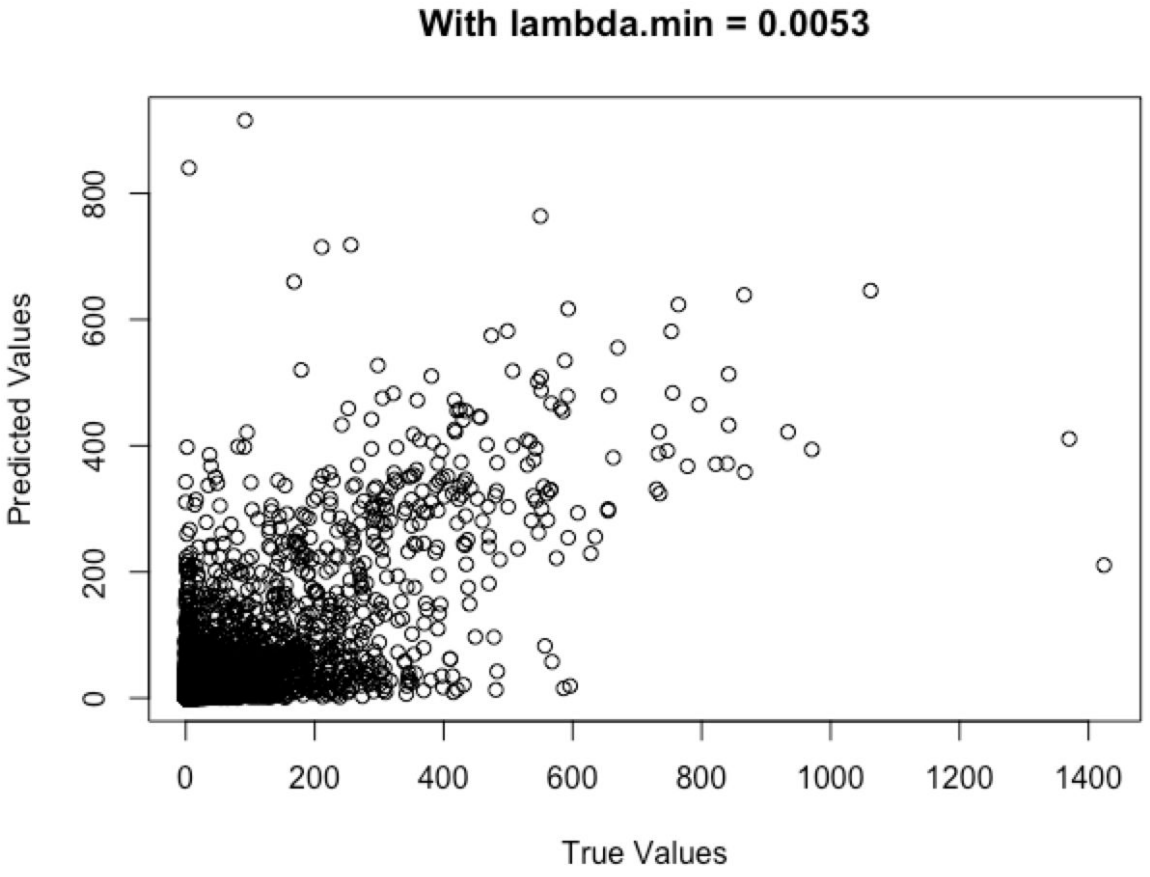
Net IDs (ts8, tanvi3)

1. Show your plot of the cross-validated deviance of the model against the regularization variable. This plot should come from cv.glmnet



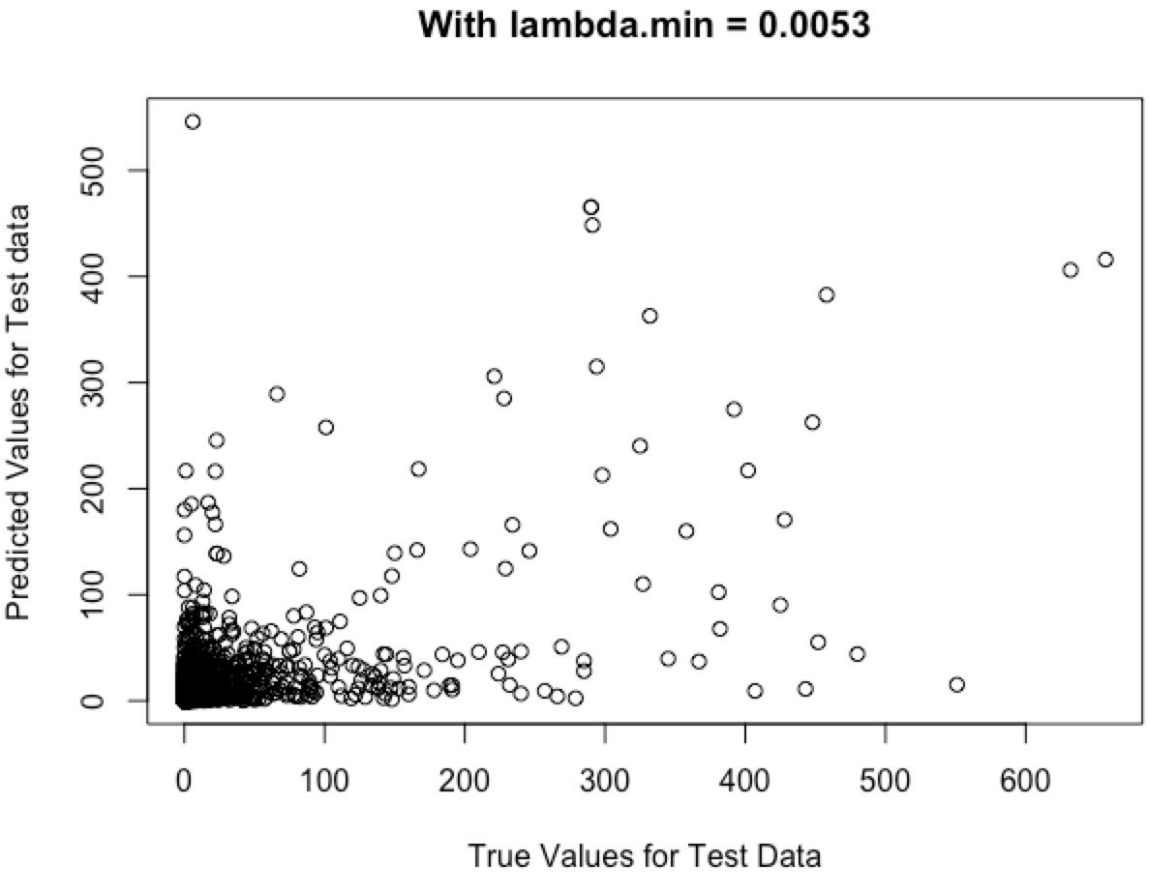
2. Indicate the value of regularization constant that you choose, and show the scatter plot of true values vs predicted values for your training data

Regularisation Constant: 0.0053 which also lambda.min



3. Indicate the value of regularization constant that you choose, and show the scatter plot of true values vs predicted values for your testing data.

Regularisation Constant: 0.0053 which also lambda.min



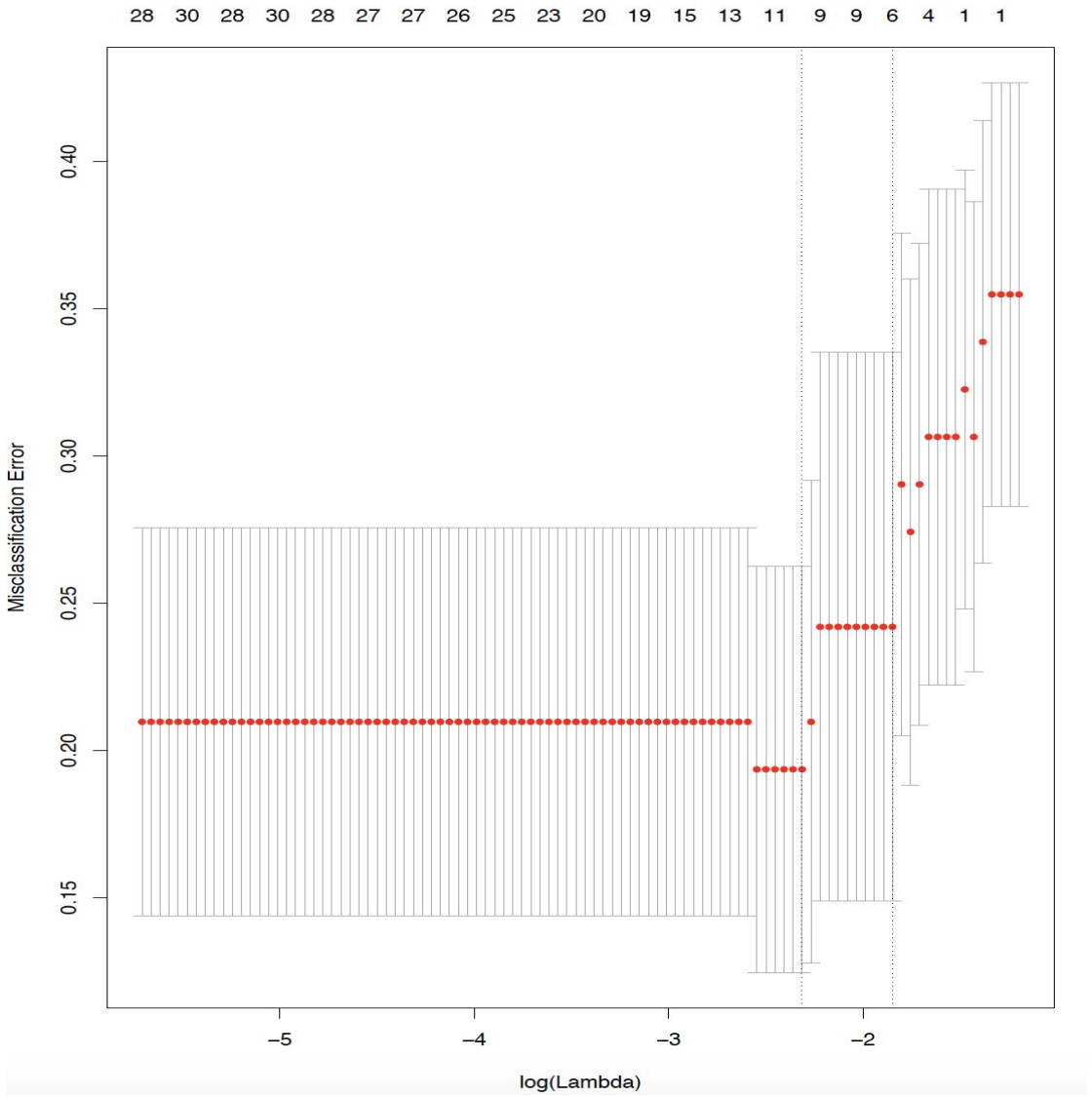
4. Compare the two plots and comment on the performance of the model. Provide comment on why this regression is difficult.

Compared with plot for training data on the previous page, with the test data plot, shows that our model performs worse with the test data than with training data. There are way more points lying far from diagonal of the plot.

Even after cross validation, it is clear to see that the model does not fit even the train data properly making the regression difficult.

- Independent variables clearly do not have a linear relationship with the dependent variable. A non linear model should be considered instead.
- Lasso forces the coefficients of some variables to zero and hence drops those features but these might contain some useful information.

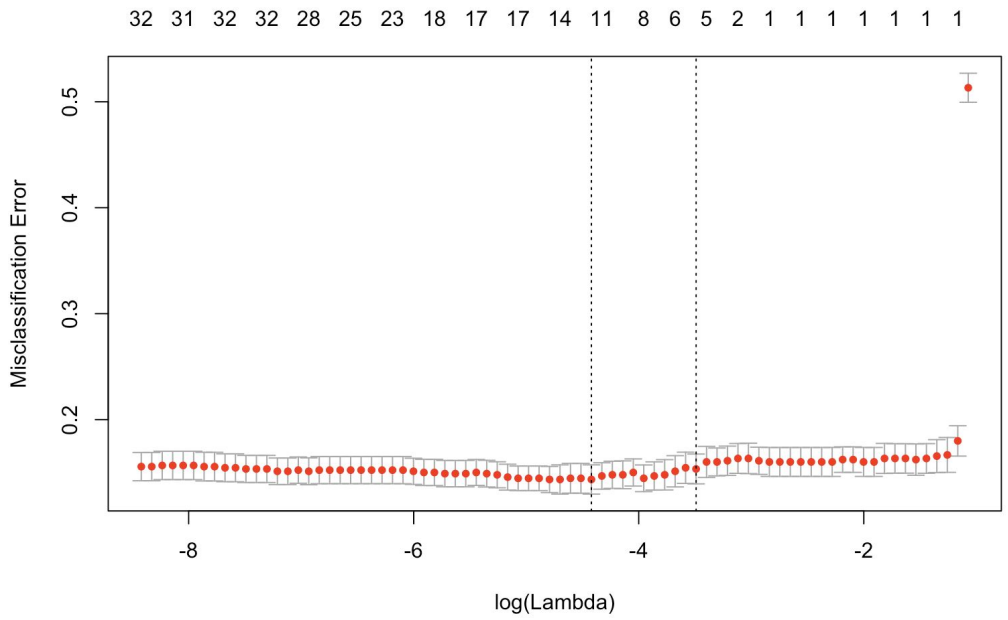
5. Show the plot of the classification error of the model against the regularization variable. Indicate the value of regularization constant of your choice and provide comment on the model performance compared with the baseline. Remember to include the classification accuracy in the comparison.



Regularisation Constant = lambda.min = 0.09

Model Performance Compared with Baseline:
The lasso model performs way better (with lambda.min) than the baseline/random guess model and the classification accuracies are:
Classification Accuracy: For baseline = 64.51%
For lambda.min = 87.09%

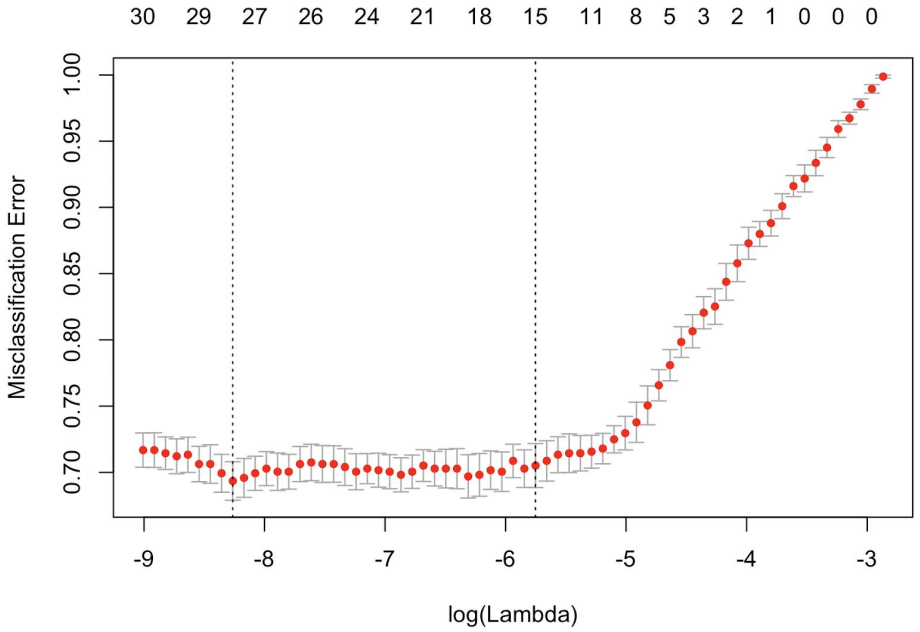
6. Predict gender with the features. Show the plot of the classification error of the model against the regularization variable. Indicate the value of regularization constant of your choice and provide comment on the model performance compared with the baseline. Remember to include the classification accuracy in the comparison.



Regularisation Constant = lambda.min = 0.01589668

Lasso Model Performance is better Compared with the Baseline model with accuracies:
Classification Accuracy: For baseline = 50.88%
For lambda.min = 86.53%

7. Predict the strain of a mouse with the features. Show the plot of the classification error of the model against the regularization variable. Indicate the value of regularization constant of your choice and provide comment on the model performance compared with the baseline. Remember to include the classification accuracy in the comparison.



Regularisation Constant = lambda.min = 0.0002577546

Lasso Model Performance is better Compared with the Baseline model with accuracies:
Classification Accuracy: For baseline = 2.33%
For lambda.min = 70.39%

8. 1 page Code screenshot. It should include code for using glmnet, making the plot and data preprocess.

```r
#HW7
#12.3
setwd("/Users/tanvimalhotra/Desktop/AML-7/")

#https://archive.ics.uci.edu/ml/datasets/BlogFeedback
#280 independent features about blog post
library(glmnet)

#a.Predict the dependent variable using all features, a generalized linear model
#Use only the data in blogData train.csv.

blog_data <- read.csv('./BlogFeedback/blogData_train.csv', header=FALSE)

X_mat <- as.matrix(blog_data[, -c(281)])
y_mat <- as.matrix(blog_data[, 281])

fit3 <- cv.glmnet(X_mat, y_mat, family="poisson", alpha = 1, nfolds = 10)
plot(fit3) #11

bestLassoLambda = fit3$lambda.min #0.005314608
LassoLambda_1se = fit3$lambda.1se #0.06552097

#lassoLambdaCoefficients = fit3$glmnet.fit$beta[, fit3$glmnet.fit$lambda == bestLassoLambda]
#print(lassoLambdaCoefficients)
#minLassoMse = min(fit3$cvm)
#print(minLassoMse)

#b.Choose a value of the regularization constant that yields a strong model, at least by the deviance criterion. Now produc
#of true values vs predicted values for data in blogData train.csv.

#lambda.min
min_pred<-predict(fit3, as.matrix(blog_data[, -c(281)]),s='lambda.min', type = "response")
plot(blog_data[, 281], min_pred, xlab="True Values", ylab="Predicted Values", main="With lambda.min = 0.0053") #12

#lambda.1se
#se1_pred<-predict(fit3, as.matrix(blog_data[, -c(281)]),s='lambda.1se', type = "response")
#plot(blog_data[, 281], se1_pred,  xlab="True Values", ylab="Predicted Values", main="With lambda.1se = 0.065") #14

#c.Choose a value of the regularization constant that yields a strong model, at least by the deviance criterion. Now produc

temp = list.files(pattern="*.csv")
```

```r
#c.Choose a value of the regularization constant that yields a strong model, at least by the deviance criterion. Now pr

temp = list.files(pattern="*.csv")
test_files = temp[-61]

for (i in 1:length(test_files)) {
  temp2 = read.csv(test_files[i], header = FALSE)
  if (i==1) {
    test<-temp2
  }
  else {
    test <- rbind(test, temp2)
  }
}

#lambda.min
test_pred<-predict(fit3, as.matrix(test[, -c(281)]),s='lambda.min', type = "response")
plot(test[, 281], test_pred, xlab="True Values for Test Data", ylab="Predicted Values for Test data", main="With lambda
```

```r
#HW7
#12.4
setwd("/Users/tanvimalhotra/Desktop/AML-7/12.4")
library("data.table")

#Read the HTML file
new <- as.data.frame(read.table("new", quote="\""))
new_T<-((transpose(new)))

new_1 <- read.table("new_1", quote="\"", comment.char="")

#For normal Tissue=1
#For tumor Tissue=0
for (i in 1:dim(new_1)[1])  {
  if (new_1[i,]<0){
  new_1[i,]=0
  }

  else {
    new_1[i,]=1
  }
}

glm<-cv.glmnet(as.matrix(new_T),as.factor(unlist(new_1)),type.measure = "class",alpha=1,family='binomial')
#The value of lambda which gives the low classification error=0.09016021
plot(glm)

#Baseleine Accuracy:64.51%
Baseline_Accuracy=max(table(new_1))/dim(new_1)[1]

#Prediction based on lambda.min
lmpredic<-predict(glm,as.matrix(new_T),type='class',s='lambda.min')
numright<-sum(new_1==lmpredic)
# The accuracy on lambda.min =87.09%
Accuracy_lambda_min<-numright/dim(lmpredic)[1]

#Prediction based on lambda.1sde
#l1predn<-predict(glm,as.matrix(new_T),type='class',s='lambda.1se')
#n1umright<-sum(new_1==l1predn)
# The accuracy on lambda.1se =85.48%
#Accuracy_lambda_1se<-n1umright/dim(l1predn)[1]
```

2 (Top Level)

```r
#HW7
#12.5
setwd("/Users/tanvimalhotra/Desktop/AML-7/12.5")

library('caTools')
library('glmnet')

#a.We will predict the gender of a mouse from the body properties and thebehavior.
Crusio1 <- read.csv("Crusio1.csv")
View(Crusio1)

#The variables you want are columns 4 through 41 of the dataset
X<-Crusio1[,4:41]
Y<-as.data.frame(Crusio1[,2])

X_Y<-cbind(X,Y)
X_Y<-na.omit(X_Y)

glm<-cv.glmnet(as.matrix(X_Y[,1:38]),as.factor(as.numeric(X_Y[,39])),type.measure = "class",alpha=1,family='multinomial')
plot(glm) #31
#The minimum value of lambda which gives the low classification error=0.01589668
#glm$lambda.1se      0.02306336

#Baseleine Accuracy:50.88%
Baseline_Accuracy=max(table(X_Y[,39]))/dim(X_Y)[1]

#Prediction based on lambda.min
lmpredic<-predict(glm,as.matrix(X_Y[,1:38]),type='class',s='lambda.min')
numright<-sum(as.numeric(X_Y[,39])==lmpredic)
# The accuracy on lambda.min =86.53%
Accuracy_lambda_min<-numright/dim(lmpredic)[1]

#Prediction based on lambda.1sde
#l1predn<-predict(glm,as.matrix(X_Y[,1:38]),type='class',s='lambda.1se')
#n1umright<-sum(as.numeric(X_Y[,39])==l1predn)
# The accuracy on lambda.1se =85.65
#Accuracy_1sde<-n1umright/dim(l1predn)[1]

#b.We will predict the strain of a mouse from the body properties and the behavior. The variables you want are columns 4 th
X<-Crusio1[,4:41]
Y<-as.data.frame(as.numeric(Crusio1[,1]))
```

(Top Level) R Script

```r
#b.We will predict the strain of a mouse from the body properties and the behavior. The variables you want are columns 4 th
X<-Crusio1[,4:41]
Y<-as.data.frame(as.numeric(Crusio1[,1]))

X_Y<-cbind(X,Y)
X_Y<-na.omit(X_Y)
set.seed(123)

# Shuffling the data set
X_Y<-X_Y[sample(nrow(X_Y)),]

# Removing the classes(categories)  which have less than 10 rows
table(X_Y[,39])<10
#REMOVE (1,10,11,18,19,33,37,40,45,50)

tf=X_Y[,39]==1 | X_Y[,39]==10 | X_Y[,39]==11 | X_Y[,39]==18 | X_Y[,39]==19 | X_Y[,39]==33 | X_Y[,39]==37 | X_Y[,39]==40 | X
X_Y<-subset(X_Y,tf==FALSE)

glm<-cv.glmnet(as.matrix(X_Y[,1:38]),as.factor(X_Y[,39]),type.measure = "class",alpha=1,family='multinomial')
#The minimum value of lambda which gives the low classification error=0.0002577546
#glm$lambda.1se 0.003177719
plot(glm) #32

#Baseleine Accuracy:2.33%
Baseline_Accuracy=max(table(X_Y[,39]))/dim(X_Y)[1]

#Prediction based on lambda.min
lmpredic<-predict(glm,as.matrix(X_Y[,1:38]),type='class',s='lambda.min')
numright<-sum(X_Y[,39]==lmpredic)
# The accuracy on lambda.min =70.39%
Accuracy_lambda_min<-numright/dim(lmpredic)[1]

#Prediction based on lambda.1sde
```