Final Exam

Sean Wendlandt

## Instructions

Please complete the questions on this template and upload your solutions in a single knitted Word or pdf document. Please also upload your completed template.

In light of the exam context, the data sets for the questions have been generated clearly to satisfy or obviously to violate the requirements of the statistical procedures. If reasonable exploratory analysis is done, there should be little ambiguity as to whether the given data satisfy the requirements. This is unrealistic, but less stressful for students and graders alike. For consistency, the random number generation version is specified in the setup block above.

# Question 1, model selection for logistic regression

The data sets “dat1train”, “dat1valid”, and “dat1test”, loaded below, have explanatory variables “x1”, “x2”,…“x8”, and a binary outcome variable, “y”.

The data sets “dat1train.mm”, “dat1valid.mm”, and “dat1test.mm”, also loaded below, include the data in “dat1train”, “dat1valid”, and “dat1test” together with columns for all pairwise interactions of “x1”, “x2”,…“x8”.

The goal is to use the validation set to select a logistic regression model or penalized logistic regression model from among those fit on the training data, then to estimate the generalization error by applying the selected model to predict the outcomes on the test data.

The parts of this question fit a backward model, a forward model based on AIC, a forward model based on cross-validation, two cross-validated ridge models, and two cross-validated lasso models.

For these questions, the models, including the models fit using cross-validation, are fit on the training data because the validation data will be used to select among the fitted models.

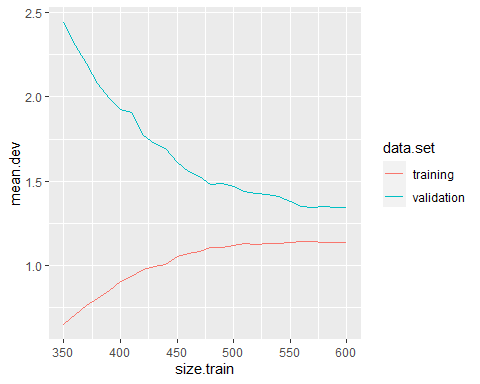
(10 points each)

load("dat1train.RData")  
load("dat1valid.RData")  
load("dat1test.RData")  
  
load("dat1train\_mm.RData")  
load("dat1valid\_mm.RData")  
load("dat1test\_mm.RData")

## Question 1, part 1

The code below generates a plot of the training error and the validation error for models of “y” based on “x1”,“x2”,…“x8” and their pairwise interactions. The models are fit on subsets of the training data of sizes 350 to 600 by 10’s. The error is measured by the mean deviance. Based on these plots, is a training set of size 400 adequate to fit a good model for these data? Please answer yes or no. You may provide an explanation. Is a training set of size 600 adequate to fit a good model for these data? Please answer yes or no. You may provide an explanation.

# function to calculate mean deviance  
dev.mean.get<-function(m.this,dat.this.new){  
 pred<-predict(m.this,dat.this.new,type="response")  
 return(-2\*mean(dat.this.new$y\*log(pred)+(1-dat.this.new$y)\*log(1-pred)))  
}  
  
# formula for "y" in terms of "x1","x2",..."x8" and their pairwise interactions   
nam<-str\_c("x",1:8)  
fmla<-as.formula(str\_c("y~(",str\_c(nam,collapse = "+"),")^2"))  
  
# sizes of training sets  
size.mat<-matrix(seq(350,nrow(dat1train),by=10),ncol=1)  
  
# function to calculate training and validation mean deviances for the sizes of training set under consideration  
  
dev.by.size.get<-function(sz){  
 m.this<-glm(fmla,data=dat1train[1:sz,],family="binomial")  
 dev.train.this<-dev.mean.get(m.this,dat1train[1:sz,])  
 dev.valid.this<-dev.mean.get(m.this,dat1valid)  
 return(c(dev.train.this,dev.valid.this))  
}  
  
# matrix of training and validation mean deviances for the sizes of training set under consideration  
devs.mat<-apply(size.mat,1,dev.by.size.get)  
  
# data frame of training and validation mean deviances for the sizes of training set under consideration  
dat.devs<-data.frame(t(devs.mat))  
names(dat.devs)<-c("training","validation")  
dat.devs$size.train<-size.mat  
  
# data frame of training and validation mean deviances formatted for plotting  
dat.devs<-pivot\_longer(dat.devs,cols=training:validation,   
 names\_to = "data.set",values\_to = "mean.dev")  
ggplot(dat.devs,aes(x=size.train,y=mean.dev,color=data.set))+geom\_line()



No, a training set of 400 would not be adequate in this case. Based on the graph above, with 400 samples the model hasn’t reached peak optimization as the training and validation data sets are still significantly reducing the deviance distance between the two sets of data.

Yes, a training set size of 600 demonstrates an healthy size of data for this model. As observed in the graph above, the mean deviance between the two data sets are becoming more parallel/horizontal indicating getting closer to a peak in optimization. While you could continue to add data, it is likely you wouldn’t see improvement past this. In a real world with limited data, I would argue data sets with a size of 550 would be fine as it also looks to reach this point. This may allow for better (larger) testing/validation data sets as the model is already trained adequately.

## Question 1, part 2

Below, a backward model for “y” based on “x1”,“x2”,…“x8” and their pairwise interactions is fit on “dat1train.mm”. This allows for interactions to be retained whether or not the interacting variables are retained. The stopping criterion used is AIC.

The deviance on the training data is verified. Please calculate and display the deviance on the validation set. Please begin the vector of model names and the vector of validation deviances to save the results of parts 2-6 for use in the overall model selection.

m2b<-glm(y~.,dat1train.mm,family="binomial")  
  
m.backward<-step(m2b,direction="backward",trace=0)  
summary(m.backward)

##   
## Call:  
## glm(formula = y ~ x1 + x4 + x8 + x1.x2 + x2.x7 + x2.x8 + x3.x5 +   
## x5.x6 + x5.x8 + x6.x7, family = "binomial", data = dat1train.mm)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.89610 -1.00492 0.02813 0.99952 2.26516   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.08818 0.10487 0.841 0.400417   
## x1 0.59187 0.10789 5.486 4.12e-08 \*\*\*  
## x4 -0.32843 0.10657 -3.082 0.002058 \*\*   
## x8 -0.35167 0.09631 -3.652 0.000261 \*\*\*  
## x1.x2 0.22714 0.09427 2.409 0.015976 \*   
## x2.x7 0.20609 0.10901 1.891 0.058674 .   
## x2.x8 -0.34267 0.11632 -2.946 0.003220 \*\*   
## x3.x5 -0.21556 0.10203 -2.113 0.034621 \*   
## x5.x6 0.81785 0.13176 6.207 5.41e-10 \*\*\*  
## x5.x8 0.23147 0.12015 1.926 0.054048 .   
## x6.x7 -0.21643 0.08830 -2.451 0.014241 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 831.78 on 599 degrees of freedom  
## Residual deviance: 693.56 on 589 degrees of freedom  
## AIC: 715.56  
##   
## Number of Fisher Scoring iterations: 4

dev.get<-function(model.this,dat.this){  
 pred<-predict(model.this,dat.this,type="response")  
 return(-2\*sum(dat.this$y\*log(pred)+(1-dat.this$y)\*log(1-pred)))  
}  
  
dev.get(m.backward,dat1train.mm)

## [1] 693.5614

# Calculating deviance on validation set  
validation.dev<-dev.get(m.backward, dat1valid.mm)  
  
# Displaying deviance on validation set  
validation.dev

## [1] 263.5624

# Please uncomment and run  
model.vec<-c()  
model.vec<-c(model.vec,"backward AIC")  
dev.vec<-c(validation.dev)

## Question 1, part 3

Below, a forward model for “y” based on “x1”,“x2”,…“x8” and their pairwise interactions is fit on “dat1train.mm” using AIC as a stopping criterion. Please calculate the deviance on the validation set. Please extend the vector of model names and the vector of validation deviances to save these results for use in the overall model selection.

m1<-glm(y~1,data=dat1train.mm,family="binomial")  
fmla.mm<-as.formula(str\_c("y~",  
 str\_c(names(dat1train.mm)[2:ncol(dat1train.mm)],collapse="+")))  
  
m.forward<-step(m1,scope=fmla.mm,direction="forward",trace=0)  
summary(m.forward)

##   
## Call:  
## glm(formula = y ~ x5.x6 + x1 + x8 + x4 + x2.x8 + x6.x7 + x5.x7 +   
## x1.x2 + x3.x5 + x1.x5, family = "binomial", data = dat1train.mm)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.92637 -0.98598 -0.01233 0.99592 2.16856   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.05431 0.10502 0.517 0.605064   
## x5.x6 0.86976 0.12893 6.746 1.52e-11 \*\*\*  
## x1 0.56318 0.10855 5.188 2.12e-07 \*\*\*  
## x8 -0.32595 0.09462 -3.445 0.000571 \*\*\*  
## x4 -0.33556 0.10664 -3.147 0.001652 \*\*   
## x2.x8 -0.20953 0.09946 -2.107 0.035156 \*   
## x6.x7 -0.21106 0.09050 -2.332 0.019691 \*   
## x5.x7 0.22942 0.11384 2.015 0.043868 \*   
## x1.x2 0.16200 0.10024 1.616 0.106057   
## x3.x5 -0.24495 0.10726 -2.284 0.022394 \*   
## x1.x5 0.18608 0.12038 1.546 0.122165   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 831.78 on 599 degrees of freedom  
## Residual deviance: 693.75 on 589 degrees of freedom  
## AIC: 715.75  
##   
## Number of Fisher Scoring iterations: 4

#Calculating validation deviance  
validation.dev<- dev.get(m.forward, dat1valid.mm)  
  
  
model.vec<-c(model.vec, "forward AIC")  
dev.vec<-c(dev.vec,validation.dev)

## Question 1, part 4

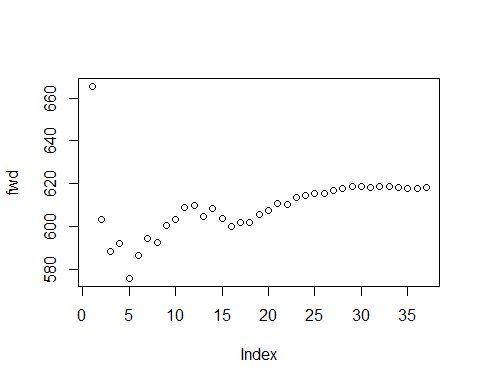
Below, a forward model for “y” based on “x1”,“x2”,…“x8” and their pairwise interactions is fit on “dat1train.mm”. The stopping criterion used is the optimal model size as indicated by the model size having the minimum cross-validated deviance.

Please calculate the deviance on the validation set. Please extend the vector of model names and the vector of validation deviances to save these results for use in the overall model selection.

# create a formula for "y" in terms of sequences of variables in "vars.add"  
  
fmla.add.fnc<-function(i,vars.add){  
 vars.in<-vars.add[1:i]  
 return(as.formula(str\_c("y~",str\_c(vars.in,collapse="+"))))  
   
}  
  
# function to calculate validation set deviance  
  
deviance.valid<-function(m,dat.valid){  
 pred.m<-predict(m,dat.valid, type="response")  
-2\*sum(dat.valid$y\*log(pred.m)+(1-dat.valid$y)\*log(1-pred.m))  
}  
  
# Code to extract the variables added in order in a call to "step" with  
# direction equal to "forward"  
  
vars.get<-function(model.forward){  
 vars.add<-model.forward$anova[,1]  
 vars.add<-str\_replace(vars.add,"\\+ ","")  
 vars.add[1]<-1  
 return(vars.add)  
}  
  
# function to fit a sequence forward models with scope "fmla.this  
# on the data set "dat.this" and  
# return the deviance for each model on "dat.valid.this".  
  
deviance.get<-function(dat.this,fmla.this,dat.valid.this){  
 m.forward<-step(glm(y~1,data=dat.this,family="binomial"),scope=fmla.this, k=0,direction="forward",trace=0)  
  
# Collect the variables used in the order in which they were added  
vars.add<-vars.get(m.forward)  
   
# Apply "fmla.add.fnc" to each value of "i". This  
# gives the formulas for the models generated initial sequences of the variables in vars.add  
# Note that the first formula is for the model with just the intercept.  
   
fmlas<-apply(matrix(1:length(vars.add),ncol=1),1,  
 fmla.add.fnc,vars.add=vars.add)  
   
# Make a list of models corresponding to these formulas.  
models<-  
 lapply(fmlas,function(x){glm(x,data=dat.this,family="binomial")})  
   
 return(sapply(models,deviance.valid,dat.valid=dat.valid.this))  
}  
  
# Note the function "createFolds" from "caret" to put  
# an approximately equal proportions of "chd"=1 and "chd"=2 cases   
# in each fold.  
# Please run this code to check that your function performs as required:  
set.seed(12345678)  
 ind <- createFolds(factor(dat1train.mm$y), k = 10, list = FALSE)  
  
deviance.wrapper<-function(i,dat.w,ind.w,fmla.w){  
 return(deviance.get(dat.w[ind.w!=i,],fmla.w,  
 dat.w[ind.w==i,]))  
}  
  
deviance.sums.xv<-function(dat.this,fmla.this){  
 ind <- createFolds(factor(dat.this$y), k = 10, list = FALSE)  
 xv.mat<-apply(matrix(1:8,ncol=1),1,deviance.wrapper,dat.w=dat.this,  
 ind.w=ind,  
 fmla.w=fmla.this)  
 return(apply(xv.mat,1,sum))  
}  
  
# Calculate selected model size  
set.seed(12345678)  
fmla.mm<-as.formula(str\_c("y~",str\_c(names(dat1train.mm)[-1],collapse="+")))  
fwd<-deviance.sums.xv(dat1train.mm,fmla.mm)  
  
(model.size<-which.min(fwd))

## [1] 5

plot(fwd)



forward.model.xv<-step(glm(y~1,data=dat1train.mm,family="binomial"),scope=fmla.mm,  
 direction="forward",steps=model.size-1,k=0,trace=0)  
summary(forward.model.xv)

##   
## Call:  
## glm(formula = y ~ x5.x6 + x1 + x8 + x4, family = "binomial",   
## data = dat1train.mm)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.96162 -1.03332 -0.00548 1.04288 2.50704   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.01584 0.08992 0.176 0.860171   
## x5.x6 0.86011 0.11857 7.254 4.05e-13 \*\*\*  
## x1 0.55435 0.10448 5.306 1.12e-07 \*\*\*  
## x8 -0.32260 0.09158 -3.522 0.000428 \*\*\*  
## x4 -0.33809 0.10450 -3.235 0.001216 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 831.78 on 599 degrees of freedom  
## Residual deviance: 717.11 on 595 degrees of freedom  
## AIC: 727.11  
##   
## Number of Fisher Scoring iterations: 4

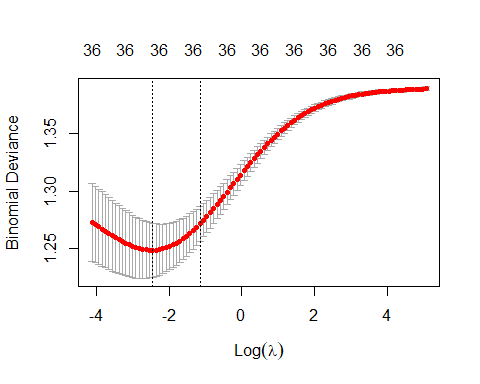
## solution   
#calculating for deviance on validation dataset  
  
validation.dev<-deviance.valid(forward.model.xv, dat1valid.mm)  
model.vec<-c(model.vec, "forward xv")  
dev.vec<-c(dev.vec, validation.dev)

In this case, backward selection with stopping based on cross-validation produces the same model as forward selection with stopping based on cross-validation. The code is available on request.

## Question 1, part 5, ridge regression

The code below uses cross-validated ridge regression as in cv.glmnet to fit ridge-penalized linear models of “y” as a function of the remaining variables and their pairwise interactions on the training set. Please report the deviance on the validation set of the model corresponding to “lambda.1se”, “lambda.min”. Please extend the vector of model names and the vector of validation deviances to save these results for use in the overall model selection.

# Format data for glmnet.  
  
Xtrain<-as.matrix(dat1train.mm[,-1])  
Xvalid<-as.matrix(dat1valid.mm[,-1])  
Xtest<-as.matrix(dat1test.mm[,-1])  
  
ytrain<-dat1train.mm$y  
yvalid<-dat1valid.mm$y  
ytest<-dat1test.mm$y  
  
# Fit ridge-penalized models  
set.seed(5678)  
cvfit = cv.glmnet(x=Xtrain, y=ytrain,alpha=0,family="binomial")  
plot(cvfit)



cvfit$lambda.min

## [1] 0.08590605

cvfit$lambda.1se

## [1] 0.3159951

# Calculate the mean squared error on the validation data for the "lambda.min" model and the "lambda.1se" model.  
  
cvpred<-predict(cvfit,Xvalid,c(cvfit$lambda.min,cvfit$lambda.1se),type="response")  
  
#model.vec<-c(model.vec,"ridge lambda.min")  
(validation.dev<-   
 -2\*sum(yvalid\*log(cvpred[,1])+(1-yvalid)\*log(1-cvpred[,1])))

## [1] 255.603

#dev.vec<-c(dev.vec,validation.dev)  
  
#model.vec<-c(model.vec,"ridge lambda.1se")  
  
# your code here  
#for lambda.min  
pred<-predict(cvfit,Xvalid,c(cvfit$lambda.min),type="response")  
  
dev.this<- -2\*sum(yvalid\*log(pred)+(1-yvalid)\*log(1-pred))  
(dev.vec<-c(dev.vec,dev.this))

## [1] 263.5624 253.4792 239.9764 255.6030

(model.vec<-c(model.vec,"lambda.min"))

## [1] "backward AIC" "forward AIC" "forward xv" "lambda.min"

summary(dev.this)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 255.6 255.6 255.6 255.6 255.6 255.6

# for lambda.1se  
  
pred1<-predict(cvfit,Xvalid,c(cvfit$lambda.1se),type="response")  
  
dev.this<- -2\*sum(yvalid\*log(pred1)+(1-yvalid)\*log(1-pred1))  
(dev.vec<-c(dev.vec,dev.this))

## [1] 263.5624 253.4792 239.9764 255.6030 258.1367

(model.vec<-c(model.vec,"lambda.1se"))

## [1] "backward AIC" "forward AIC" "forward xv" "lambda.min" "lambda.1se"

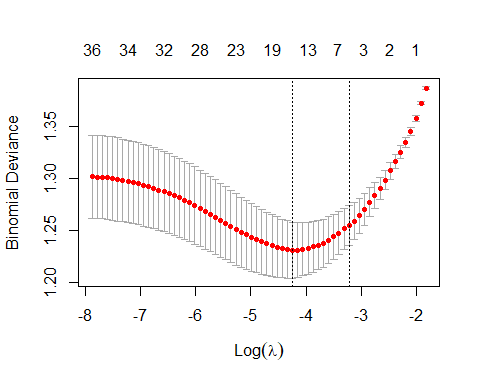
summary(dev.this)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 258.1 258.1 258.1 258.1 258.1 258.1

## Question 1, part 6

Please use cross-validated lasso regression as in cv.glmnet to fit lasso-penalized linear models of “y” as a function of the remaining variables and their pairwise interactions on the training set. Please report the deviance on the validation set of the models corresponding to “lambda.1se” and “lambda.min”. Please report the number of variables with non-zero coefficients for the models corresponding to “lambda.1se” and “lambda.min”. Please extend the vector of model names and the vector of validation deviances to save these results for use in the overall model selection.

set.seed(5678)  
  
#All the code below is mine  
   
fit.lasso<-cv.glmnet(x=Xtrain, y=ytrain, alpha=1, family="binomial")  
plot(fit.lasso)



# For lamda.min  
fit.lasso$lambda.min

## [1] 0.01432999

# coefficients for the model with lambda equal to "lambda.min"  
(coef.min<-coef(fit.lasso, s = "lambda.min"))

## 37 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 0.0385002641  
## x1 0.4183623930  
## x2 .   
## x3 0.0005614567  
## x4 -0.1968855805  
## x5 .   
## x6 .   
## x7 -0.0141514356  
## x8 -0.2331991393  
## x1.x2 0.0745006300  
## x1.x3 .   
## x1.x4 0.0150062205  
## x1.x5 0.0724814863  
## x1.x6 .   
## x1.x7 0.0132317667  
## x1.x8 .   
## x2.x3 .   
## x2.x4 .   
## x2.x5 .   
## x2.x6 .   
## x2.x7 0.0227033553  
## x2.x8 -0.1359951670  
## x3.x4 .   
## x3.x5 -0.0956934082  
## x3.x6 .   
## x3.x7 .   
## x3.x8 .   
## x4.x5 .   
## x4.x6 .   
## x4.x7 .   
## x4.x8 0.0435595517  
## x5.x6 0.6970493158  
## x5.x7 0.1161961033  
## x5.x8 0.0250180730  
## x6.x7 -0.1135977368  
## x6.x8 .   
## x7.x8 .

# names of the variables dropped in this model  
row.names(coef.min)[which(coef.min==0)]

## [1] "x2" "x5" "x6" "x1.x3" "x1.x6" "x1.x8" "x2.x3" "x2.x4" "x2.x5"  
## [10] "x2.x6" "x3.x4" "x3.x6" "x3.x7" "x3.x8" "x4.x5" "x4.x6" "x4.x7" "x6.x8"  
## [19] "x7.x8"

pred<-predict(fit.lasso,Xvalid,c(fit.lasso$lambda.min),type="response")  
dev.this<- -2\*sum(yvalid\*log(pred)+(1-yvalid)\*log(1-pred))  
(dev.vec<-c(dev.vec,dev.this))

## [1] 263.5624 253.4792 239.9764 255.6030 258.1367 247.9584

(model.vec<-c(model.vec,"lasso.lambda.min"))

## [1] "backward AIC" "forward AIC" "forward xv" "lambda.min"   
## [5] "lambda.1se" "lasso.lambda.min"

# For lamda.1se  
fit.lasso$lambda.1se

## [1] 0.03987405

# coefficients for the model with lambda equal to "lambda.1se"  
(coef.min<-coef(fit.lasso, s = "lambda.1se"))

## 37 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 0.016508731  
## x1 0.219365043  
## x2 .   
## x3 .   
## x4 -0.007439984  
## x5 .   
## x6 .   
## x7 .   
## x8 -0.103446782  
## x1.x2 .   
## x1.x3 .   
## x1.x4 .   
## x1.x5 .   
## x1.x6 .   
## x1.x7 .   
## x1.x8 .   
## x2.x3 .   
## x2.x4 .   
## x2.x5 .   
## x2.x6 .   
## x2.x7 .   
## x2.x8 .   
## x3.x4 .   
## x3.x5 .   
## x3.x6 .   
## x3.x7 .   
## x3.x8 .   
## x4.x5 .   
## x4.x6 .   
## x4.x7 .   
## x4.x8 .   
## x5.x6 0.544421465  
## x5.x7 0.015878083  
## x5.x8 .   
## x6.x7 .   
## x6.x8 .   
## x7.x8 .

# names of the variables dropped in this model  
row.names(coef.min)[which(coef.min==0)]

## [1] "x2" "x3" "x5" "x6" "x7" "x1.x2" "x1.x3" "x1.x4" "x1.x5"  
## [10] "x1.x6" "x1.x7" "x1.x8" "x2.x3" "x2.x4" "x2.x5" "x2.x6" "x2.x7" "x2.x8"  
## [19] "x3.x4" "x3.x5" "x3.x6" "x3.x7" "x3.x8" "x4.x5" "x4.x6" "x4.x7" "x4.x8"  
## [28] "x5.x8" "x6.x7" "x6.x8" "x7.x8"

pred<-predict(fit.lasso,Xvalid,c(fit.lasso$lambda.1se),type="response")  
dev.this<- -2\*sum(yvalid\*log(pred)+(1-yvalid)\*log(1-pred))  
(dev.vec<-c(dev.vec,dev.this))

## [1] 263.5624 253.4792 239.9764 255.6030 258.1367 247.9584 250.2248

(model.vec<-c(model.vec,"lasso.lambda.1se"))

## [1] "backward AIC" "forward AIC" "forward xv" "lambda.min"   
## [5] "lambda.1se" "lasso.lambda.min" "lasso.lambda.1se"

The number of non-zero coefficients in the lamda.min model is 18 including the intercept. In the lambda.1se model there are 6 including the intercept.

Note that the number of variables under consideration is too large for basic use of best subsets.

## Question 1, part 7

Of the backward model fit by AIC (part 2), the forward model fit by AIC (part 3), the forward model fit by cross-validation (part 4), the lambda.min and the lambda.1se ridge models (part 5), and the lambda.min and the lambda.1se lasso models (part 6), which model has the best deviance on the validation set? Please show the summary of the model fit on the combined training and validation sets using the variables, or the lambda in the case of penalized regression, that produced the selected model. What is the deviance of this new fitted model on the test set?

small.dev<-which(dev.vec==min(dev.vec))  
  
model.vec[small.dev] #forward xv model has the best deviance on the validation set (part 4)

## [1] "forward xv"

#fit model on combined training and validation sets  
dat1tv.mm<-bind\_rows(dat1train.mm,dat1valid.mm)  
  
forward.model.xv.test<-step(glm(y~1,data=dat1tv.mm,family="binomial"),scope=fmla.mm,  
 direction="forward",steps=model.size-1,k=0,trace=0)  
summary(forward.model.xv.test)

##   
## Call:  
## glm(formula = y ~ x5.x6 + x1 + x8 + x4, family = "binomial",   
## data = dat1tv.mm)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9969 -1.0441 0.4216 1.0287 2.4769   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.09476 0.07786 1.217 0.224   
## x5.x6 0.82886 0.10105 8.202 2.36e-16 \*\*\*  
## x1 0.55917 0.09241 6.051 1.44e-09 \*\*\*  
## x8 -0.36214 0.07863 -4.606 4.11e-06 \*\*\*  
## x4 -0.35325 0.08893 -3.972 7.12e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1106.74 on 798 degrees of freedom  
## Residual deviance: 955.64 on 794 degrees of freedom  
## AIC: 965.64  
##   
## Number of Fisher Scoring iterations: 4

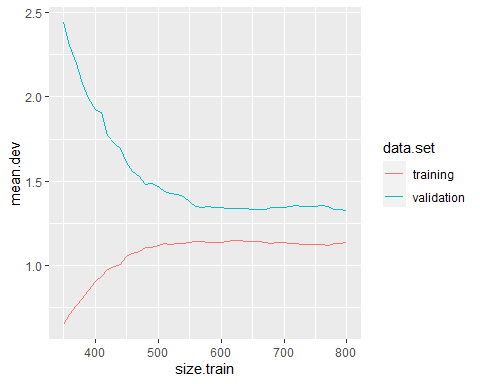
#Finding the MSE for forward xv model  
mean(forward.model.xv.test$residuals^2) #7.833858

## [1] 5.473023

Based on our model results displayed above, the best-performing model is the forward xv model based on the lowest deviance. The MSE for the model is 5.473023 after being fit on training and validation data sets.

Note: the plot above to assess the adequacy of the size of the training set has the drawback that an appearance of stability of the deviances for large subsets may be due in part to the fact that the larger models are based on data sets with many points in common. This plot is a good follow-up now that the test data has been used for its intended purpose.

temp<-bind\_rows(dat1train,dat1test)  
size.mat<-matrix(seq(350,nrow(temp),by=10),ncol=1)  
dev.by.size.get<-function(sz){  
 #dat.sz<-temp[sample(1:nrow(temp),sz),]  
 #m.this<-glm(fmla,data=dat.sz,family="binomial")  
 #dev.train.this<-dev.mean.get(m.this,dat.sz)  
 m.this<-glm(fmla,data=temp[1:sz,],family="binomial")  
 dev.train.this<-dev.mean.get(m.this,temp[1:sz,])  
 dev.valid.this<-dev.mean.get(m.this,dat1valid)  
 return(c(dev.train.this,dev.valid.this))  
}  
  
devs.mat<-apply(size.mat,1,dev.by.size.get)  
  
dat.devs<-data.frame(t(devs.mat))  
names(dat.devs)<-c("training","validation")  
dat.devs$size.train<-size.mat  
  
dat.devs<-pivot\_longer(dat.devs,cols=training:validation,   
 names\_to = "data.set",values\_to = "mean.dev")  
ggplot(dat.devs,aes(x=size.train,y=mean.dev,color=data.set))+geom\_line()



# Question 2, logistic regression

The code below generates a logistic regression model for the outcome variable “y”.

dat1tv.mm<-bind\_rows(dat1train.mm,dat1valid.mm)  
m<-glm(y~x1+x4+x5.x6+x8,dat1tv.mm,family = "binomial")

## Question 2, part 1

(5 points)

Please report the p-value for the Hosmer-Lemeshow test with 10 bins applied to this model.

Please report the p-value for the Le Cessie - van Houwelingen test with default parameters applied to this model.

Please comment on meaning of these tests regarding the validity of the model.

library(ResourceSelection)# for Hosmer-Lemeshow

## Warning: package 'ResourceSelection' was built under R version 4.1.3

## ResourceSelection 0.3-5 2019-07-22

library(rms)# for Le Cessie - van Houwelingen

## Warning: package 'rms' was built under R version 4.1.3

## Loading required package: Hmisc

## Warning: package 'Hmisc' was built under R version 4.1.3

## Loading required package: survival

##   
## Attaching package: 'survival'

## The following object is masked from 'package:caret':  
##   
## cluster

## Loading required package: Formula

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':  
##   
## src, summarize

## The following objects are masked from 'package:base':  
##   
## format.pval, units

## Loading required package: SparseM

##   
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':  
##   
## backsolve

#Hosmer-Lemeshow test  
m<-glm(y~x1+x4+x5.x6+x8,dat1tv.mm,family = "binomial")  
hoslem.test(m$y, fitted(m), g=10)

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: m$y, fitted(m)  
## X-squared = 14.578, df = 8, p-value = 0.06789

#Le Cessie - van Houwelingen test  
  
m2<-lrm(y~x1+x4+x5.x6+x8,x=TRUE,y=TRUE,data=dat1tv.mm)  
m2

## Logistic Regression Model  
##   
## lrm(formula = y ~ x1 + x4 + x5.x6 + x8, data = dat1tv.mm, x = TRUE,   
## y = TRUE)  
##   
## Model Likelihood Discrimination Rank Discrim.   
## Ratio Test Indexes Indexes   
## Obs 799 LR chi2 151.10 R2 0.230 C 0.729   
## 0 386 d.f. 4 R2(4,799)0.168 Dxy 0.459   
## 1 413 Pr(> chi2) <0.0001 R2(4,598.6)0.218 gamma 0.459   
## max |deriv| 4e-13 Brier 0.207 tau-a 0.229   
##   
## Coef S.E. Wald Z Pr(>|Z|)  
## Intercept 0.0948 0.0779 1.22 0.2236   
## x1 0.5592 0.0924 6.05 <0.0001   
## x4 -0.3533 0.0889 -3.97 <0.0001   
## x5.x6 0.8289 0.1011 8.20 <0.0001   
## x8 -0.3621 0.0786 -4.61 <0.0001   
##

resid(m2, 'gof') # the type 'gof' gives the Le Cessie - van Houwelingen

## Sum of squared errors Expected value|H0 SD   
## 165.6174728 165.1015462 0.5647762   
## Z P   
## 0.9135063 0.3609763

# fit statistic and probability

Both of these tests are measuring the goodness of fit for the logistic regression model. From the tests performed, the Hosmer-Lemeshow returns a p-value of .06789 indicating it is a good fit (a value below .05 indicates a poor fit). In the Le Cessie - Van Houwelingen test, we get a p-value of .36097 which also indicates a good fit.

## Question 2, part 2

(5 points)

Please report the deviance, confusion matrix, accuracy, precision (proportion of correctly predicted 1’s among the cases predicted to be 1’s by the model), recall (proportion of correctly predicted 1’s among cases with outcome equal to 1), and F1 (2(recall)(precision)/ (recall + precision)) for this model fitted on the combined training and validation data then applied to the test data, “dat1test.mm”.

#### The test deviance

anova(m,test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: y  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 798 1106.74   
## x1 1 27.978 797 1078.76 1.227e-07 \*\*\*  
## x4 1 14.604 796 1064.15 0.0001326 \*\*\*  
## x5.x6 1 86.421 795 977.73 < 2.2e-16 \*\*\*  
## x8 1 22.098 794 955.64 2.591e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#### Confusion matrix

probs<-predict(m, dat1test.mm,type="response")  
pred<-probs>=.5  
(confus.mat<-table(dat1test.mm$y,pred))

## pred  
## FALSE TRUE  
## 0 55 32  
## 1 23 91

#### Accuracy:

(accuracy<-sum(diag(confus.mat))/sum(confus.mat))

## [1] 0.7263682

#### Precision:

#below two lines for reference  
#dat1tv.mm<-bind\_rows(dat1train.mm,dat1valid.mm)  
#m<-glm(y~x1+x4+x5.x6+x8,dat1tv.mm,family = "binomial")  
(precision<-confus.mat[2,2]/sum(confus.mat[,2]))

## [1] 0.7398374

#### Recall:

recall<-(confus.mat[2,2]/sum(confus.mat[2,]))

#### F1:

(2\*recall\*precision/(recall+precision))

## [1] 0.7679325

# Question 3, count data

(10 points)

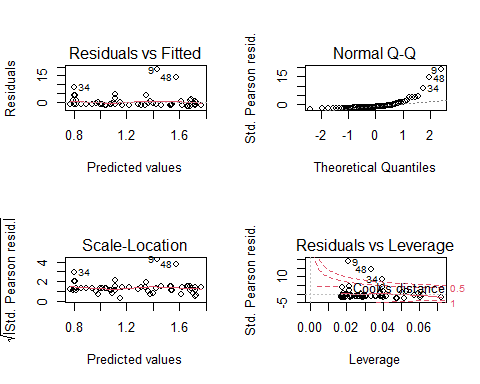
In “dat3”, loaded below, the values of “y” represent counts of occurrences of a phenomenon. Which model(s) seem most suited to predicting “y” from “x” among Poisson regression, quasipoisson regression, negative binomial regression, and linear regression? Please explain your reasoning on the basis of the data, summaries of the models, and other simple diagnostics you choose.

#### Poisson

load("dat3.RData")  
  
output.p <-glm(formula = y ~ x, data = dat3, family = poisson)  
(summary(output.p))

##   
## Call:  
## glm(formula = y ~ x, family = poisson, data = dat3)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.280 -2.354 -1.523 -0.025 10.853   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.02771 0.09561 10.749 < 2e-16 \*\*\*  
## x 0.51312 0.10867 4.722 2.34e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 503.11 on 59 degrees of freedom  
## Residual deviance: 480.19 on 58 degrees of freedom  
## AIC: 594.66  
##   
## Number of Fisher Scoring iterations: 6

### optional supplementary diagnostics ###  
par(mfrow=c(2,2))  
plot(output.p)



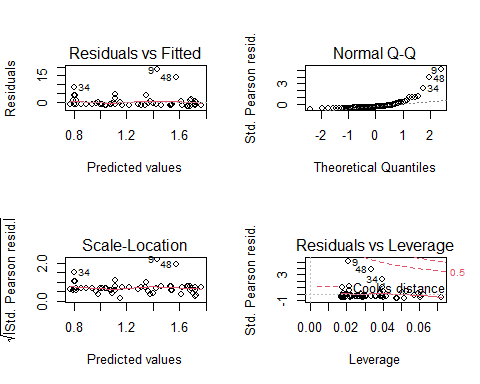
par(mfrow=c(1,1))

#### Quasipoisson

output.qp <-glm(formula = y ~ x, data = dat3, family = quasipoisson())  
(summary(output.qp))

##   
## Call:  
## glm(formula = y ~ x, family = quasipoisson(), data = dat3)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.280 -2.354 -1.523 -0.025 10.853   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.0277 0.3546 2.899 0.00528 \*\*  
## x 0.5131 0.4030 1.273 0.20804   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for quasipoisson family taken to be 13.7538)  
##   
## Null deviance: 503.11 on 59 degrees of freedom  
## Residual deviance: 480.19 on 58 degrees of freedom  
## AIC: NA  
##   
## Number of Fisher Scoring iterations: 6

### optional supplementary diagnostics ###  
par(mfrow=c(2,2))  
plot(output.qp)



par(mfrow=c(1,1))

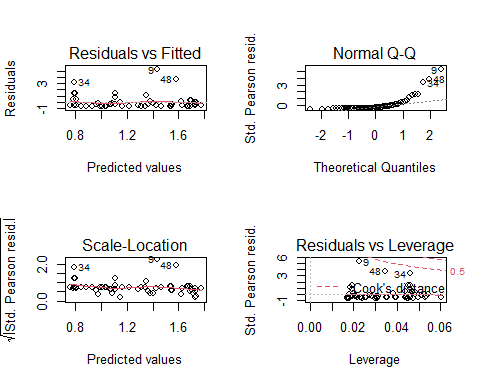
#### Negative binomial

Note that the standard deviation for the negative binomial is .

library(MASS)  
output.nb<- glm.nb(formula = y ~ x, data = dat3)  
summary(output.nb)

##   
## Call:  
## glm.nb(formula = y ~ x, data = dat3, init.theta = 0.3658084267,   
## link = log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.42134 -1.23526 -0.55752 -0.01138 2.15387   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.0208 0.2730 3.739 0.000185 \*\*\*  
## x 0.5249 0.3537 1.484 0.137857   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Negative Binomial(0.3658) family taken to be 1)  
##   
## Null deviance: 61.830 on 59 degrees of freedom  
## Residual deviance: 59.646 on 58 degrees of freedom  
## AIC: 271.97  
##   
## Number of Fisher Scoring iterations: 1  
##   
##   
## Theta: 0.3658   
## Std. Err.: 0.0861   
##   
## 2 x log-likelihood: -265.9700

### optional supplementary diagnostics ###  
par(mfrow=c(2,2))  
plot(output.nb)



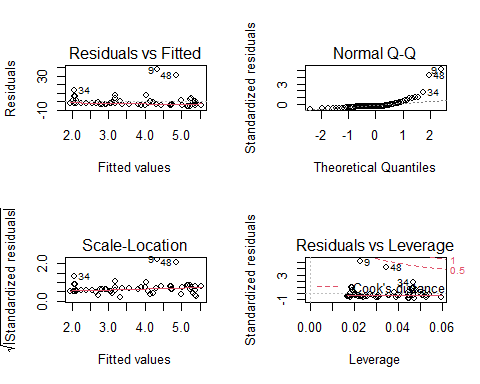
par(mfrow=c(1,1))

#### Linear regression

output.l<- lm(y ~ x, data = dat3)  
summary(output.l)

##   
## Call:  
## lm(formula = y ~ x, data = dat3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.242 -2.921 -2.123 -0.088 37.649   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.877 1.146 2.511 0.0148 \*  
## x 1.854 1.507 1.231 0.2234   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.423 on 58 degrees of freedom  
## Multiple R-squared: 0.02544, Adjusted R-squared: 0.008642   
## F-statistic: 1.514 on 1 and 58 DF, p-value: 0.2234

### supplementary diagnostics ###  
par(mfrow=c(2,2))  
plot(output.l)



par(mfrow=c(1,1))

From our model results, the Poisson model has the lowest p-values. However, our data distribution is skewed which would invalidate many of these models including the Poisson. Because of this, the Negative Binomial is the best model as this model is able to handle these skewed cases. Thus, we can make inferences from its output. Since it has an intercept p-value of 0.000185 \*\*\* it indicates a good fit.

# Question 4

(10 points)

The data frame “dat4” loaded below represents the number of incidents, “incidents” at facilities of different sizes,“exposure”, with different ratings, “rating”. A negative binomial model for “incidents” is fit below using log(exposure) as an offset and “rating” as an explanatory variable. Please fit a Poisson and a quasipoisson model for “incidents” using log(exposure) as an offset and “rating” as an explanatory variable. Which model provides the best information regarding these data?

load("dat4.RData")  
  
m.nb<-glm.nb(incidents~rating,offset(log(exposure)),data=dat4)  
summary(m.nb)

##   
## Call:  
## glm.nb(formula = incidents ~ rating, data = dat4, weights = offset(log(exposure)),   
## init.theta = 24.86514185, link = log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.5109 -1.5321 -0.7621 0.1363 5.3344   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.87143 0.06783 86.56 <2e-16 \*\*\*  
## rating -0.98831 0.02544 -38.85 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Negative Binomial(24.8651) family taken to be 1)  
##   
## Null deviance: 4612.75 on 29 degrees of freedom  
## Residual deviance: 127.99 on 28 degrees of freedom  
## AIC: 633.51  
##   
## Number of Fisher Scoring iterations: 1  
##   
##   
## Theta: 24.87   
## Std. Err.: 7.87   
##   
## 2 x log-likelihood: -627.513

m.p<-glm(incidents ~ rating, offset(log(exposure)), data = dat4, family = poisson)  
summary(m.p)

##   
## Call:  
## glm(formula = incidents ~ rating, family = poisson, data = dat4,   
## weights = offset(log(exposure)))  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -7.4453 -1.7515 -0.6961 0.3751 6.2280   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.92996 0.03645 162.67 <2e-16 \*\*\*  
## rating -1.01506 0.01841 -55.15 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 8383.51 on 29 degrees of freedom  
## Residual deviance: 226.94 on 28 degrees of freedom  
## AIC: 667.26  
##   
## Number of Fisher Scoring iterations: 4

m.qp<-glm(incidents ~ rating, offset(log(exposure)), data = dat4, family = quasipoisson())  
summary(m.qp)

##   
## Call:  
## glm(formula = incidents ~ rating, family = quasipoisson(), data = dat4,   
## weights = offset(log(exposure)))  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -7.4453 -1.7515 -0.6961 0.3751 6.2280   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.92996 0.10499 56.48 <2e-16 \*\*\*  
## rating -1.01506 0.05301 -19.15 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for quasipoisson family taken to be 8.293883)  
##   
## Null deviance: 8383.51 on 29 degrees of freedom  
## Residual deviance: 226.94 on 28 degrees of freedom  
## AIC: NA  
##   
## Number of Fisher Scoring iterations: 4

Of the models above, the best is the Poisson model as it has the lowest Std. Error out of the models while having comparable p-values. This indicates there is less variance around the fit.