**Eric Agyemang:**

CODES FOR HW3

library(ISLR)

attach(College)

set.seed(11)

#Randomly splitting data into trainig and test set in 7:3 ratio

subset<-sample(nrow(College),nrow(College)\*0.7)

train<-College[subset,]

test<-College[-subset,]

ls.full<-lm(Apps~.,data=train)

summary(ls.full)

predicted.apps<-predict(ls.full,test)

testerror<-mean((test$Apps-predicted.apps)^2)

testerror

#create matrix for training set and test set

train.mat<-model.matrix(Apps~.,data=train)

test.mat<-model.matrix(Apps~.,data=test)

#defining grid to covering all the range of lambda.This will e used to find best value of lambda

grid<-10^seq(4,-2,length=100)

#fitting the ridge regression model

library(glmnet)

ridge<-glmnet(train.mat,train$Apps,alpha=0,lambda=grid,thresh = 1e-12)

#doing cross validation on model

cv.ridge<-cv.glmnet(train.mat,train$Apps,alpha=0,lambda=grid,thresh=1e-12)

#finding the lambda for which cv error is minimum on training data

bestlam.ridge<-cv.ridge$lambda.min

bestlam.ridge

#using the lambda value obtained from cross validation for the ridge model directly on test data set to get the predicted values

pred.newridge<-predict(ridge,s=bestlam.ridge,newx =test.mat)

#Mean Square Error calculation

mean((test$Apps-pred.newridge)^2)

lasso<-glmnet(train.mat,train$Apps,alpha=1,lambda=grid,thresh = 1e-12)

#doing cross validation on model

cv.lasso<-cv.glmnet(train.mat,train$Apps,alpha=1,lambda=grid,thresh=1e-12)

#finding the lambda for which cv error is minimum on training data

bestlam.lasso<-cv.lasso$lambda.min

bestlam.lasso

#using the lambda value obtained from cross validation for the lasso model directly on test data set to get the predicted values

pred.newlasso<-predict(lasso,s=bestlam.lasso,newx =test.mat)

#Mean Square Error calculation

mean((test$Apps-pred.newlasso)^2)

#Non zero Coefficienct estimates

predict(lasso,s=bestlam,type="coefficients")

coef.lasso <- predict(fit.lasso, type="coefficients", s=lambda)[1:ncol(College),]

coef.lasso[coef.lasso != 0]

library(pls)

pcrmodel<-pcr(Apps~.,data=train,scale=TRUE,validation="CV")

#plotting Mean square error (MSEP) for different number of components

validationplot(pcrmodel,val.type="MSEP")

predict.pcr<-predict(pcrmodel,test,ncomp=17)

mean((test$Apps-predict.pcr)^2)

plsrmodel<-plsr(Apps~.,data=train,scale=TRUE,validation="CV")

#plotting Mean square error (MSEP) for different number of components

validationplot(plsrmodel,val.type="MSEP")

predict.plsr<-predict(plsrmodel,test,ncomp=10)

mean((test$Apps-predict.plsr)^2)

#Least Square model

test.avg <- mean(test$Apps)

lm.r2 <- 1 - mean((predicted.apps - test$Apps)^2) / mean((test.avg - test$Apps)^2)

#Ridge model

ridge.r2 <- 1 - mean((pred.newridge - test$Apps)^2) / mean((test.avg - test$Apps)^2)

#Lasso model

lasso.r2 <- 1 - mean((pred.newlasso - test$Apps)^2) / mean((test.avg - test$Apps)^2)

#PCR model

pcr.r2 <- 1 - mean((predict.pcr - test$Apps)^2) / mean((test.avg - test$Apps)^2)

#PLS model

pls.r2 <- 1 - mean((predict