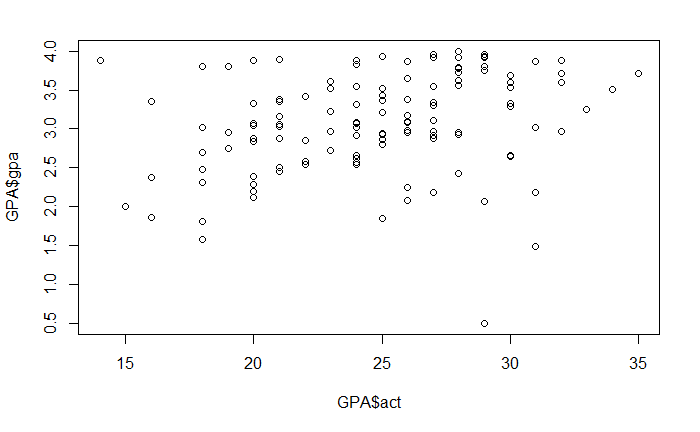
STAT 4360 Mini Project 4

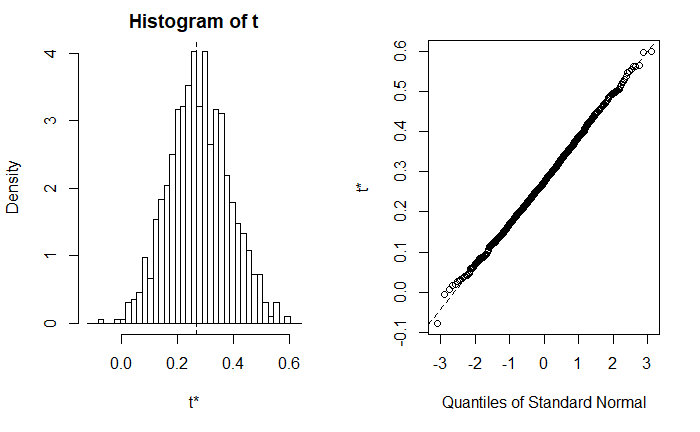
Name: Jaemin Lee

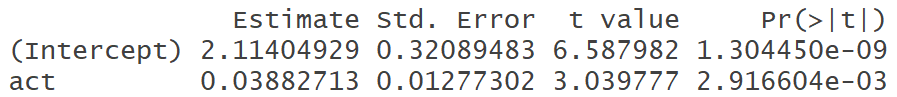
Section 1: Answers to the specific questions asked

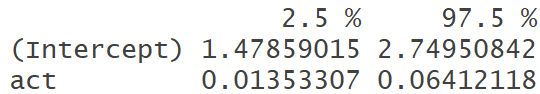
1. A) Based on the scatter plot, it seems like there's a positive linear relationship between gpa and act. There is an association between gpa and act (i.e., higher gpa and higher act score)



B) A point estimate of p is the correlation between GPA and ACT which is 0.27. The bootstrap estimates of bias is 0.007800885, bootstrap estimates of standard error is 0.1072831, and 95% confidence interval is (0.0728, 0.4917). This confidence interval tells us that if a large number of samples is collected (1000 replicates in our case) and a confidence interval is created for each sample, then approximately 95% of these intervals will contain the population parameter (correlation between GPA and ACT). Below shows the histogram of bootstrapped samples and the normality of the distribution. t is the matrix of random variables in statistics whereas t\* are the estimates of t random variables. Thus, the plot makes sense as it is the histogram of the random variable t and the x-axis contains the estimates of the random variables (the t\*s).



C) The least square estimates of the regression coefficients, and standard errors of the estimates are given below.

The 95% confidence interval of the coefficients are given below. Notice the 95% confidence interval for act is in (0.0135, 0.64). We will compare this with part D) later on.

Below plots represent linear regression assumptions and diagnostics.

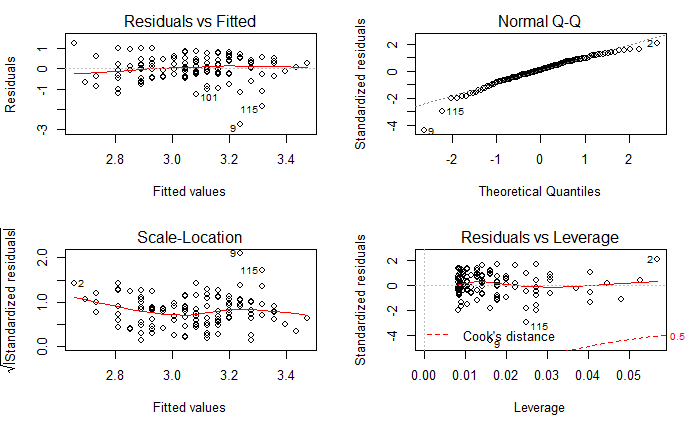
Linearity assumptions can be checked by inspecting the Residuals vs Fitted plot. Ideally, the residual plot will show no fitted pattern. That is, the red line should be approximately horizontal at zero. The presence of a pattern may indicate a problem with some aspect of the linear model. In our example, there is a slight curve in the residual plot. This suggests that we can’t assume the linear relationship between the predictors and the outcome variables.

The QQ plot of residuals can be used to visually check the normality assumption. The normal probability plot of residuals should approximately follow a straight line. In our case, there is a slight curvature and both ends don't really follow the dotted lines. This is happening due to the outliers (i.e. student 2, 9, and 115)

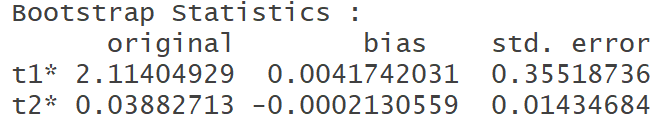
Homogeneity of variance assumption can be checked by examining the scale-location plot, also known as the spread-location plot. This plot shows if residuals are spread equally along the ranges of predictors. It’s good if you see a horizontal line with equally spread points. In our example, this is not the case.

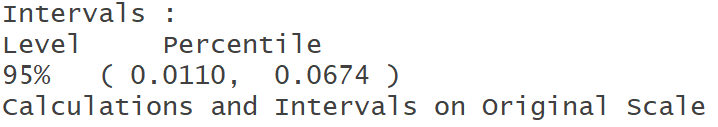
It can be seen that the variability (variances) of the residual points increases with the value of the fitted outcome variable, suggesting non-constant variances in the residuals errors (or heteroscedasticity). A possible solution to reduce the heteroscedasticity problem is to use a log or square root transformation of the outcome variable (y).

Residuals vs Leverage represents outliers and high leverage points. Student 2 and 115 are the two most extreme points with standardized residuals around 2 and below -3 respectively. Also, student 2 has a high leverage compared to other students



D) Below show the standard errors from nonparametric bootstrap on linear regression.

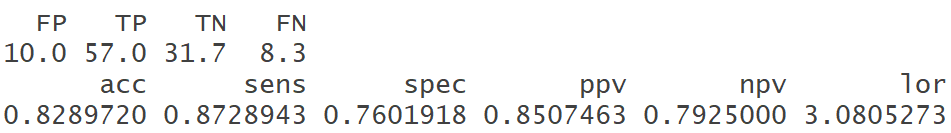


Below show the 95% confidence intervals using percentile bootstrap.

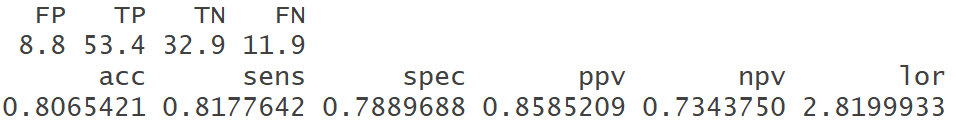
Notice the difference between the outputs from part C) and D). The bootstrap estimates for SE(beta hat 1) = 0.014 (corresponds to ACT), whereas the standard error estimates for beta hat 1 from part C) is 0.0127. The difference comes from the assumptions that linear regression makes when fitting the model. It makes an assumption that the error terms for each observation are uncorrelated with common variance. Then it estimates variance using RSS. However, bootstrap doesn't rely on any assumptions. Thus, it is more likely to give a more accurate estimates of SEs than linear regression. Hence, due to the more accurate SEs, it yields to a more accurate confidence interval for bootstrap than the least square. Notice from part C), the 95 % confidence interval is (0.0135, 0.064), whereas the 95% confidence intervals using percentile bootstrap is (0.011, 0.067).

1. A) code given in section 2 for changing StoreID into a categorical variable.

B) Below show the confusion matrix, accuracy, sensitivity, and specificity of LDA.

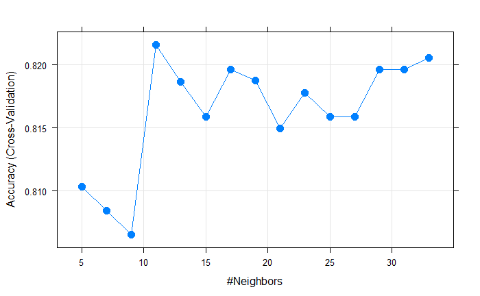
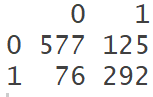


Misclassification rate = (FP + FN)/107 = (10 + 8.3)/107 = 0.171 and this is equivalent to the test error rate = 1 – accuracy = 1 – 0.829 = 0.171. Sensitivity = TP/(TP + FN) = 57/(57+8.3) = 0.87, which is consistent with the table above. Specificity = TN/(TN + FP) = 31.7/(31.7+10) = 0.76, which is consistent with the table as well.

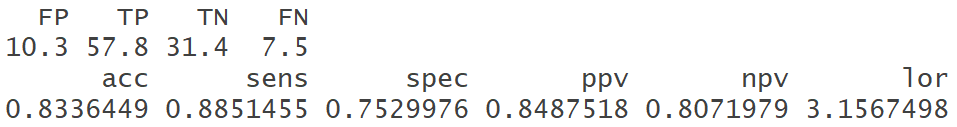
C) Below show the confusion matrix, accuracy, sensitivity, and specificity of QDA.

Misclassification rate = (FP + FN)/107 = (8.8+11.9)/107 = 0.193 and this is equivalent to the test error rate = 1 – accuracy = 1 – 0.807 = 0.193. Sensitivity = TP / (TP + FN) = 53.4/(53.4+11.9) = 0.818, which is consistent with the table above. Specificity = TN/(TN + FP) = 32.9/(32.9+8.8) = 0.789, which is consistent with the table as well.

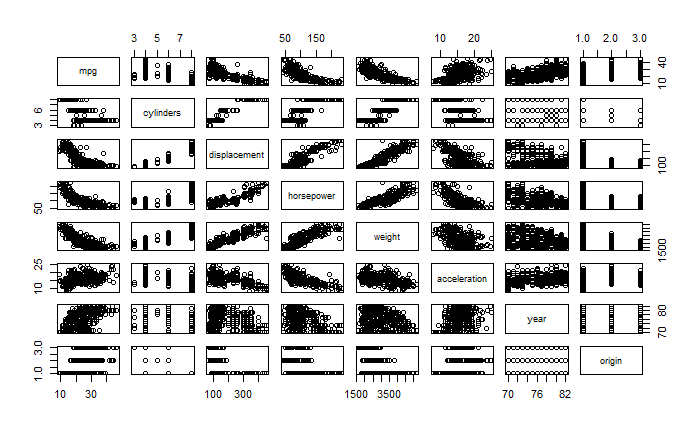
D) The optimal K for KNN using 10-folds cross validation is 11. The plot below shows that the accuracy is the highest (0.821) when K = 11. Also, the confusion matrix is shown below.



Misclassification rate = (FP + FN)/107 = (125+76)/1070 = 0.18 and this is equivalent to the test error rate = 1 – accuracy = 1 – 0.82 = 0.18. Sensitivity = TP / (TP + FN) = 557/(557+76) = 0.88. Specificity = TN/(TN + FP) = 292/(292+125) = 0.70.

E) Below show the confusion matrix, accuracy, sensitivity, and specificity of Logistic Regression.

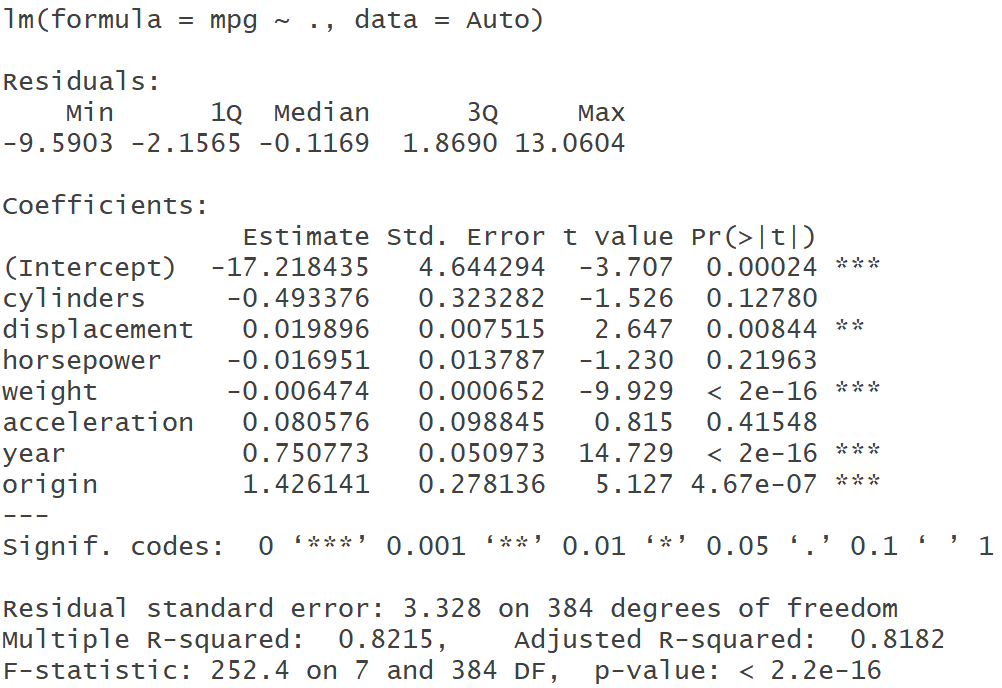
Misclassification rate = (FP + FN)/107 = (10.3+7.5)/107 = 0.166 and this is equivalent to the test error rate = 1 – accuracy = 1 – 0.833 = 0.166. Sensitivity = TP / (TP + FN) = 57.8/(57.8+7.5) = 0.885, which is consistent with the table above. Specificity = TN/(TN + FP) = 31.4/(31.4+10.3) = 0.753, which is consistent with the table as well.

F) Comparing the results in (B) – (E), the best classifier is logistic regression. It has the lowest misclassification rate and the highest accuracy.

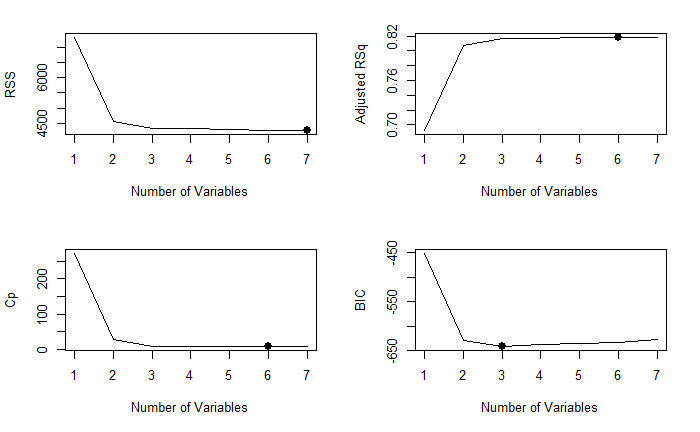
3. A)

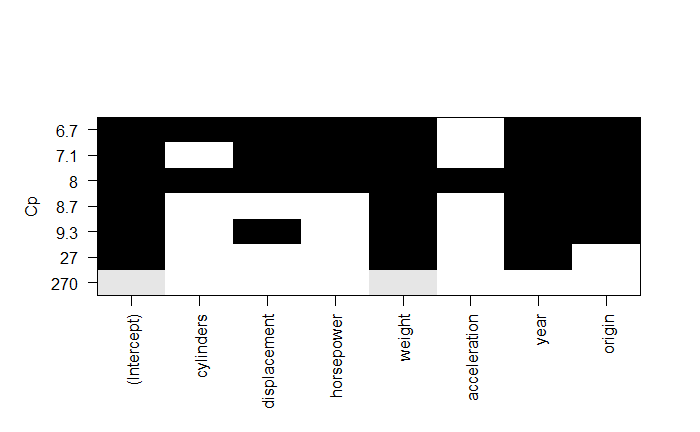
Based on the scatter plots above, it appears that mpg, cylinders, displacement, horsepower, weight, acceleration have associations.

B) Below represents the summary of multiple linear regression.

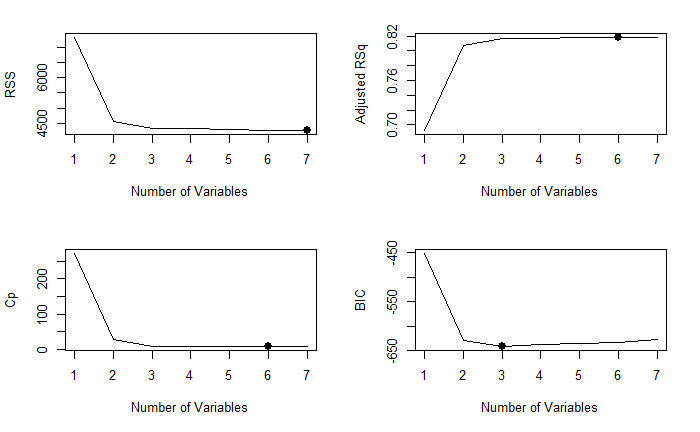


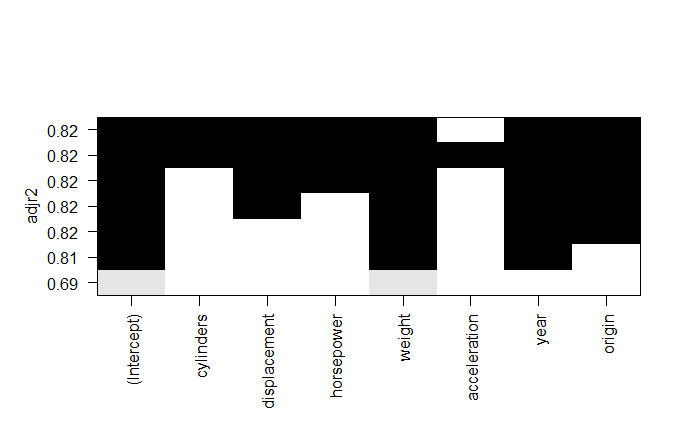
Notice adjusted R^2 is 0.8182 and displacement, weight, year, and origin are the significant variables.

C) Below shows the best model chosen using the best-subset selection method.

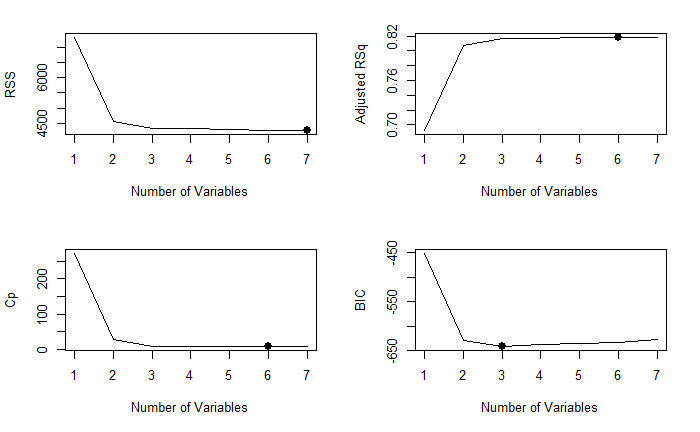
Residual sum of squares is the lowest for model 7. Adjusted R^2 is the highest for model 6. We want to look for the highest adjusted R^2 to check the goodness of fit for regression models that have different numbers of variables. Cp, which gives us the estimate of test MSE, is the lowest for model 6. BIC (Bayesian Information Criterion) is the lowest for model 3. Thus, model 6 appears to be the best using the best-subset selection method due to the highest adjusted R^2 (0.8183) and the lowest Cp. Model 6 contains cylinders, displacement, horsepower, weight, year, and origin as predictors.

The plot above shows which variables were significant in Cp. The top row of each plot contains a black square for each variable selected according to the optimal model associated with that statistic. For instance, we see that several models share a Cp close to 6.7. However, the model with the lowest Cp is the 6-variable model that contains only cylinders, displacement, horsepower, weight, year, and origin

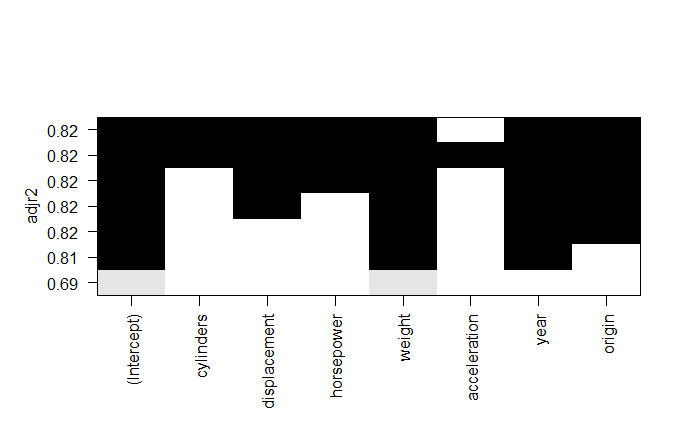
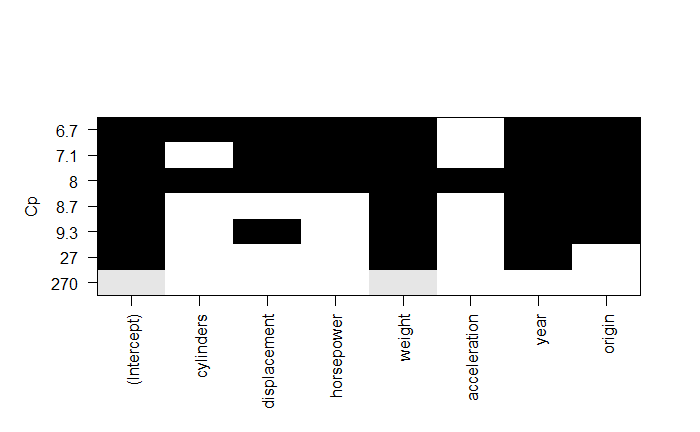
D) Below shows the best model chosen using forward stepwise selection.

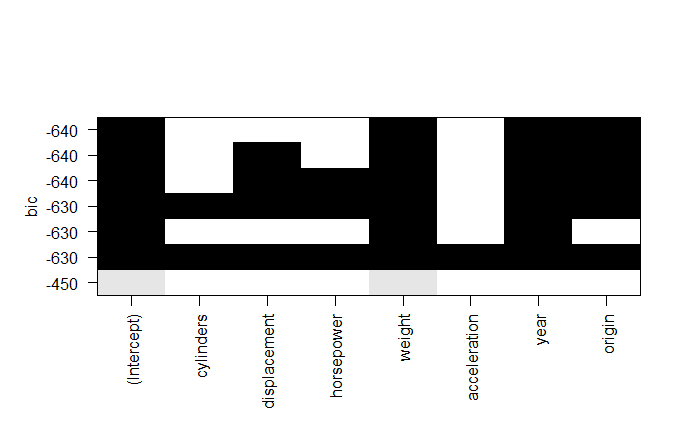
Interestingly, it shows the exact same result as the best-subset selection. Thus, the best model chosen using forward stepwise selection is the model with 6 variables again.

The top row of the above plot shows that adjusted R^2 is the highest when variables - cylinders, displacement, horsepower, weight, year, and origin are selected.

E) Below shows the best model chosen using backward stepwise selection.

Again, the plots look the same as the best-subset selection, thus as forward stepwise selection. Hence, the best model chosen using backward stepwise selection is the model with 6 variables – cylinders, displacement, horsepower, weight, year, and origin.



The above plots, again, are consistent with what we saw from best-subset selection and forward stepwise selection. We can clearly see that those 6 variables were selected again.

The top row of the above plot represents that three variables were selected using BIC. BIC is the lowest (-640) for weight, year, and origin. This is consistent with the 2 by 2 plots from parts (C) – (E) as their BICs are the lowest with the model with 3 variables.

F) Adjusted R^2 is the highest (0.8183) on the 6th model for best-subset selection, backward and forward stepwise selections. Of course, adjusted R^2 will go up as the model gets more complex. Also, Cp, which gives us the estimate of test MSE is the lowest on the model with 6 variables as well. Part (B), the linear model, has R^2 of 0.8182. Hence, based on the results from (B) – (E), I would suggest the model with 6 variables.

Section 2: R code

########## Question 1 ##########

GPA = read.csv("C:/Users/jaemi/Desktop/STAT 4360/Projects/Project 4/gpa.csv")

str(GPA)

### (a) scatter plot

plot(GPA$act, GPA$gpa)

#library(PerformanceAnalytics)

#chart.Correlation(GPA)

# looks like there's a positive linear relationship between gpa and act

# strong associated between gpa nad act (higher gpa = higher act)

# parameter of interest: p - population correlation between gpa and act

cor.fn <- function(x, indices) {

result <- cor(x[indices, "gpa"], x[indices, "act"])

return(result)

}

# point estimate of p

cor.fn(GPA, 1:nrow(GPA)) # 0.27

cor(GPA$gpa, GPA$act) # 0.27

# notice these two match

# perform bootstrap

library(boot)

# point estimate of p

library(MASS)

set.seed(1)

cor.boot <- boot(data = GPA, statistic = cor.fn, R = 1000)

cor.boot

# See the bootstrap distribution of correlation estimate

plot(cor.boot)

# Get a 95% confidence interval for correlation

boot.ci(cor.boot, type = "perc")

### (c)

# fit lm

gpa.lm = lm(gpa ~ act, data = GPA); summary(gpa.lm)$coefficients

# 95% CI of coefficeints

confint(gpa.lm, level= 0.95)

# Linear Regression Assumptions and Diagnostics

par(mfrow = c(2, 2))

plot(gpa.lm)

# d)

# applying bootstrap to LM

fit.fn <- function(data, index) {

result <- coef(lm(data[index, "gpa"] ~ data[index, "act"]))

return(result)

}

n = nrow(GPA)

fit.fn(GPA, 1:n)

# Estimates and SEs from LM fit

summary(lm(gpa ~ act, data = GPA))$coef

# perform bootstrap on LM

set.seed(1)

lm.boot = boot(data = GPA, statistic = fit.fn, R = 1000)

lm.boot

names(lm.boot)

sum(is.na(lm.boot$t))

boot.ci(lm.boot, conf = 0.95, index = 2, type = 'perc')

############ Question 2 ###########

library(ISLR)

# a)

# extract certain predictors

library(dplyr)

# extracting variables

OJ = OJ[,c(1, 3, 4, 5, 6, 7, 10)]

head(OJ)

str(OJ)

# let "1" indicate MM (Minute Maid) and "0" indicate CH (Citrus Hill).

old.purchase = c('CH', 'MM')

new.purchase = factor(c('0', '1'))

OJ$Purchase = new.purchase[match(OJ$Purchase, old.purchase)]

# recode StoreID as a categorical variable

old.ID = c("1", "2", "3", "4", "7")

new.ID = factor(c("1", "2", "3", "4", "7"))

OJ$StoreID = new.ID[match(OJ$StoreID, old.ID)]

str(OJ)

# b)

# perform LDA

# all data are used as training data

library(MASS)

# K-folds on LDA

library(crossval)

# classification examples

# set up lda prediction function

predfun.lda = function(train.x, train.y, test.x, test.y, negative)

{

require(MASS) # for lda function

lda.fit = lda(train.y ~., data = train.x)

ynew = predict(lda.fit, test.x)$class

# count TP, FP etc.

out = confusionMatrix(test.y, ynew, negative=negative)

return(out)

}

na.omit(OJ)

X = OJ[, 2:7] # predictors

head(X)

Y = (OJ[,1]) # response

head(Y)

set.seed(1)

cv.lda = crossval(predfun.lda, X, Y, K=10, B=1, negative = '1')

cv.lda$stat

diagnosticErrors(cv.lda$stat)

# c)

# perform QDA

# K= 10 fold validation on lda

library(crossval)

# classification examples

# set up lda prediction function

predfun.qda = function(train.x, train.y, test.x, test.y, negative)

{

require(MASS) # for lda function

qda.fit = qda(train.y ~., data = train.x)

ynew = predict(qda.fit, test.x)$class

# count TP, FP etc.

out = confusionMatrix(test.y, ynew, negative=negative)

return(out)

}

na.omit(OJ)

X = OJ[, 2:7] # predictors

head(X)

Y = (OJ[,1]) # response

head(Y)

set.seed(1)

cv.qda = crossval(predfun.qda, X, Y, K=10, B=1, negative = '1')

cv.qda$stat

diagnosticErrors(cv.qda$stat)

# d)

library(caret)

train.control <- trainControl(method = "cv")

as.numeric.factor = function(x) {

as.numeric(levels(x))[x]

}

OJ$StoreID = as.numeric.factor(OJ$StoreID)

str(OJ)

set.seed(2)

knn.fit <- train(Purchase~ ., method = "knn", tuneLength = 15, trControl = train.control, metric = "Accuracy", data = OJ)

knn.fit # optimal K = 11

knn.fit

plot(knn.fit, cex = 2, pch =20) # 0.821 accuracy rate

library(KODAMA)

knn.cv = knn.double.cv(OJ[,-1], OJ[,1], compmax = 11)

print(min(knn.cv$Q2Y))

print(which.min(knn.cv$Q2Y))

table(knn.cv$Ypred, OJ[,1])

plot(knn.cv$Q2Y,type = "o", ylab = "misclassification error")

# (e) using logistic regression

predfun.lr = function(train.x, train.y, test.x, test.y, negative)

{

lr.fit = glm(train.y ~., family = binomial, data = train.x)

lr.prob = predict(lr.fit, test.x, type = "response")

lr.pred = ifelse(lr.prob >= 0.5, "1", "0")

# count TP, FP etc.

out = confusionMatrix(test.y, lr.pred, negative=negative)

return(out)

}

na.omit(OJ)

X = OJ[, 2:7] # predictors

head(X)

Y = (OJ[,1]) # response

head(Y)

set.seed(1)

cv.lr = crossval(predfun.lr, X, Y, K=10, B=1, negative = '1')

cv.lr$stat

diagnosticErrors(cv.lr$stat)

# f)

# comparing restuls from b)-e)

err.lda = 1-diagnosticErrors(cv.lda$stat)[1]; print(err.lda) # 0.17

err.qda = 1-diagnosticErrors(cv.qda$stat)[1]; print(err.qda) # 0.19

err.lr = 1-diagnosticErrors(cv.lr$stat)[1]; print(err.lr) # 0.166

############## Question 3 #############

# a) Exploratory anlaysis

library(ISLR)

Auto

str(Auto)

# Take mpg as response and the remaining variables (except name) as predictors.

Auto = Auto[,-9]

str(Auto)

# check if there are any missing values

sum(is.na(Auto)) # no missing values

plot(Auto)

# looks like mpg, cylinders, dispacement, horsepower, weight, acceleration have associations

summary(Auto) # get the hang of the data set

# b) multiple linear regression using the least square method

lm.fit = lm(mpg ~ . , data = Auto)

summary(lm.fit)

# c) use best-subset selection to find the best model

library(leaps)

# Total number of predictors in the data

totpred = ncol(Auto) - 1; totpred

# full model

fit.full = regsubsets(mpg ~ ., data = Auto, nvmax = totpred)

fit.summary = summary(fit.full);fit.summary

names(fit.summary)

# check R^2

fit.summary$rsq

# check adjusted R^2

max(fit.summary$adjr2)

# Plot model fit measures for best model of each size against size

par(mfrow = c(2, 2))

# RSS

plot(fit.summary$rss, xlab = "Number of Variables", ylab = "RSS", type = "l")

which.min(fit.summary$rss) # 7

points(7, fit.summary$rss[7], cex = 2, pch = 20)

# Adjusted R^2

plot(fit.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")

which.max(fit.summary$adjr2) # 6

points(6, fit.summary$adjr2[6], cex = 2, pch = 20)

# CP

plot(fit.summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")

which.min(fit.summary$cp) # 6

points(6, fit.summary$cp[6], cex = 2, pch = 20)

# BIC

plot(fit.summary$bic, xlab = "Number of Variables", ylab = "BIC", type = "l")

which.min(fit.summary$bic) # 3

points(3, fit.summary$bic[3], cex = 2, pch = 20)

par(mfrow = c(1, 1))

plot(fit.full, scale = "r2")

plot(fit.full, scale = "adjr2")

plot(fit.full, scale = "Cp")

plot(fit.full, scale = "bic")

# Get coefficients of best model for a given size

coef(fit.full, 6)

# d) Forward stepwise selection

fit.fwd = regsubsets(mpg ~ ., data = Auto, nvmax = totpred, method = "forward")

fit.fwd.summary = summary(fit.fwd);fit.fwd.summary

names(fit.fwd.summary)

# check R^2

fit.fwd.summary$rsq

# check adjusted R^2

fit.fwd.summary$adjr2

# Plot model fit measures for best model of each size against size

par(mfrow = c(2, 2))

# RSS

plot(fit.fwd.summary$rss, xlab = "Number of Variables", ylab = "RSS", type = "l")

which.min(fit.fwd.summary$rss) # 7

points(7, fit.fwd.summary$rss[7], cex = 2, pch = 20)

# Adjusted R^2

plot(fit.fwd.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")

which.max(fit.fwd.summary$adjr2) # 6

points(6, fit.fwd.summary$adjr2[6], cex = 2, pch = 20)

# CP

plot(fit.fwd.summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")

which.min(fit.fwd.summary$cp) # 6

points(6, fit.fwd.summary$cp[6], cex = 2, pch = 20)

# BIC

plot(fit.fwd.summary$bic, xlab = "Number of Variables", ylab = "BIC", type = "l")

which.min(fit.fwd.summary$bic) # 3

points(3, fit.fwd.summary$bic[3], cex = 2, pch = 20)

par(mfrow = c(1,1))

plot(fit.fwd, scale = "r2")

plot(fit.fwd, scale = "adjr2")

plot(fit.fwd, scale = "Cp")

plot(fit.fwd, scale = "bic")

# e) Backward stepwise selection

fit.bwd = regsubsets(mpg ~ ., data = Auto, nvmax = totpred, method = "backward")

fit.bwd.summary = summary(fit.bwd);fit.bwd.summary

names(fit.bwd.summary)

# check R^2

fit.bwd.summary$rsq

# check adjusted R^2

fit.bwd.summary$adjr2

# Plot model fit measures for best model of each size against size

par(mfrow = c(2, 2))

# RSS

plot(fit.bwd.summary$rss, xlab = "Number of Variables", ylab = "RSS", type = "l")

which.min(fit.fwd.summary$rss) # 7

points(7, fit.fwd.summary$rss[7], cex = 2, pch = 20)

# Adjusted R^2

plot(fit.bwd.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")

which.max(fit.bwd.summary$adjr2) # 6

points(6, fit.bwd.summary$adjr2[6], cex = 2, pch = 20)

# CP

plot(fit.bwd.summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")

which.min(fit.bwd.summary$cp) # 6

points(6, fit.bwd.summary$cp[6], cex = 2, pch = 20)

# BIC

plot(fit.bwd.summary$bic, xlab = "Number of Variables", ylab = "BIC", type = "l")

which.min(fit.bwd.summary$bic) # 3

points(3, fit.bwd.summary$bic[3], cex = 2, pch = 20)

par(mfrow = c(1, 1))

plot(fit.bwd, scale = "r2")

plot(fit.bwd, scale = "adjr2")

plot(fit.bwd, scale = "Cp")

plot(fit.bwd, scale = "bic")