Brief Solution of ISyE 7406 Homework 3

General Peer Grader Guidance: When grading your peers, you will not only learn how to improve your future homework submissions but you will also gain deeper understanding of the concepts in the assignments. When assigning scores, consider the responses to the questions given your understanding of the problem and using the solutions as a guide. Moreover, please add comments to your review, e.g., explain why the student missed the points when deducting points, or explain why you think the students do well and which parts impress you most. Ensure your comments are specific to questions and the student responses in the assignment.

For peer grading of homework #3, there are three aspects

1. **Technical Part.** Most students knew how to implement our methods in R and were able to obtain R results. However, there might have several common main technical concerns:

   (i) While we did not explicitly mention the cross validation (CV) in this HW itself as in HW#1 and HW#2, it is not robust to conclude the "best" method based on a single split of the data set into the training and testing datasets. Such a single split is not robust and not convincing. You should evaluate different methods based on CV, either K-fold CV or Monte Carlo CV will be fine. Note that the result of a single split can be vary different, given that we have a small dataset. Thus you should look at the logic flow instead of numerical values, e..g, the numerical values in our solution set are just one of many possibilities.

   (ii) Some students might blindly apply the methods without discussing whether they are suitable to this specific data set or not, e.g., please check the model assumptions.

   (iii) It might be useful to do variable selection if you want your results to be as good as possible. There are generally two approaches: one is to use the variable/model selection technique such as stepwise algorithm with AIC/BIC criterion or LASSO method, and the other is to select those important exploratory variables in the sense of being highly correlated with the response Y variable. Both will be acceptable in this homework, as both have been widely used in practice. Since we have shown the first approach in HW#1-HW#2, here we illustrate the second approach. Again, both approaches are acceptable and should receive full credits.

2. **Interpretation Part**: Most students now provide some brief comments to discuss the plots, figures or tables, or make a claim which classification method is better. However, statistical analysis or data mining should be treated as the tools, not the final outcomes, in your future career. In other words, you can also discuss how to apply your results/methods in the context of building or buying cars with high gas mileage, depending on whether you work for the car manufacturing or customers. Just report numerical numbers is not enough, need to interpret the results.

3. **Presentation Part**: Your report should be easy to read — imagine your parents, future boss can understand your report at the high-level and know your conclusions (even though they may not understand some specific technical or mathematical details). Also please please do not just copy and paste R codes or outputs in the main body or the report — you may include them in the appendix if you think they are very important. You can re-write those R

Table 1: The correlation matrix.

	mpg01	cylinders	displacement	horsepower	weight	acceleration	year	origin
mpg01	1.00	-0.76	-0.75	-0.67	-0.76	0.35	0.43	0.51
cylinders	-0.76	1.00	0.95	0.84	0.90	-0.50	-0.35	-0.57
displacement	-0.75	0.95	1.00	0.90	0.93	-0.54	-0.37	-0.61
horsepower	-0.67	0.84	0.90	1.00	0.86	-0.69	-0.42	-0.46
weight	-0.76	0.90	0.93	0.86	1.00	-0.42	-0.31	-0.59
acceleration	0.35	-0.50	-0.54	-0.69	-0.42	1.00	0.29	0.21
year	0.43	-0.35	-0.37	-0.42	-0.31	0.29	1.00	0.18
origin	0.51	-0.57	-0.61	-0.46	-0.59	0.21	0.18	1.00

output in a way that any intelligent people (without any statistical or data mining training) can understand you.

Below are some general technical comments to each section of your data analysis report.

(i) Introduction.

Includes problem statement and motivation of your project such as interesting or meaningful implications in practice. At the last part you can mention the structure of your project.

(ii) Exploratory data analysis:

You can include
1. Description of data file, data format.
2. The most important graphs, plots or tables that will help us to understand data sets (some relevant though not crucial graphs, plots, or tables might be included in the appendix)

For this specific dataset, boxplots and scatterplots are drawn in Figure 1 and 2 to find relationship between the response variable "mpg01" and predictors. Since it is hard to see the relations between mpg01 and the other variables, these plots can be put in the appendix.

Also, the correlation matrix is

Based on those analysis, we might want to select those significant features, which are cylinders, displacement, horsepower, weight, and origin. However, please note that there are no unique answers, and any other reasonable choices (including all variables) are also correct.

(iii) Classification methods and results:

You can include
Simple description of each methods
For KNN, we choose k using cross validation.
Cross validation with 100 repetitions to obtain averaged testing accuracy or one time test result.

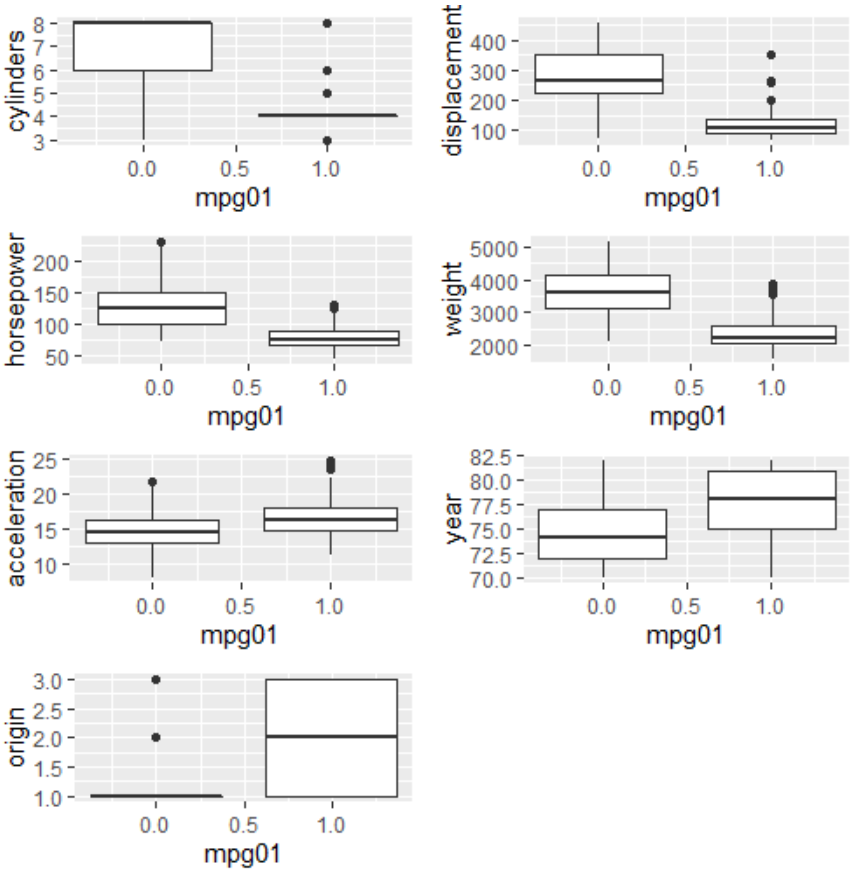

Figure 1: Boxplot.

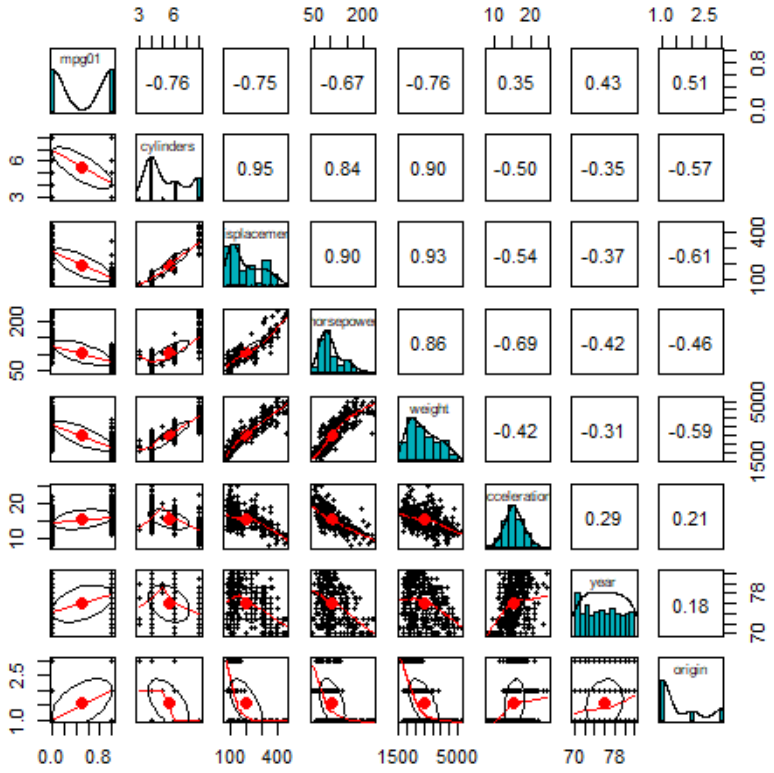

Figure 2: Scatterplot.

Table 2: Training and Testing Errors (One split)

Training errors				
LDA	QDA	Naive Bayes	Logistic Regression	KNN(K=3)
0.102	0.099	0.099	0.099	0.076
Testing errors				
LDA	QDA	Naive Bayes	Logistic Regression	KNN(K=3)
0.128	0.103	0.103	0.103	0.103

Table 3: Training and Testing Errors for KNN with different k (One split)

Training error									
$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
0.000	0.079	0.076	0.085	0.079	0.102	0.105	0.091	0.105	0.108
Testing error									
$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
0.154	0.205	0.103	0.154	0.128	0.128	0.128	0.128	0.128	0.128

Table 4: Average errors with 100 iterations

Training average errors				
LDA	QDA	Naive Bayes	Logistic Regression	KNN(K=3)
0.0993	0.1019	0.0977	0.0999	0.0795
Testing average errors				
LDA	QDA	Naive Bayes	Logistic Regression	KNN(K=3)
0.1077	0.1079	0.1010	0.1072	0.1254

It is important to note that the actual numerical values will likely be different based on the student's choice of variables, random seed, and computer (Mac/Win/Unix).

(v) findings

You can include
1. relevant findings based on tables and numbers you found in your results;
2. T-test to confirm the best performing method;
3. based on your best method/model, what are the crucial factors when building or buying cars with high gas mileage?

Table 5: Standard deviation of errors with 100 iterations

Training average errors				
LDA	QDA	Naive Bayes	Logistic Regression	KNN(K=3)
0.1128	0.0852	0.0892	0.0984	0.1228
Testing average errors				
LDA	QDA	Naive Bayes	Logistic Regression	KNN(K=3)
0.0478	0.0459	0.0448	0.0520	0.0535

Table 6: A comparison of QDA vs other methods

p-values of paired t-test			
LDA	Naive bayes	Logistic Regression	KNN
9.163e-16	0.001275	8.451e-05	1.134e-12

R codes

```
### Read the data
Auto1 <- read.table(file = "Auto.csv", sep = ",", header=T)

### Do the classification
mpg01 = I(Auto1$mpg >= median(Auto1$mpg))
Auto = data.frame(mpg01, Auto1[,-1]); ## replace column "mpg" by "mpg01".

### Graphical analysis
## scatter plot
scat_cylinders <- ggplot(Auto1, aes(y = mpg, x = cylinders)) + geom_point()
scat_displacement <- ggplot(Auto1, aes(y = mpg, x = displacement)) + geom_point()
scat_horsepower <- ggplot(Auto1, aes(y = mpg, x = horsepower)) + geom_point()
scat_weight <- ggplot(Auto1, aes(y = mpg, x = weight)) + geom_point()
scat_acceleration <- ggplot(Auto1, aes(y = mpg, x = acceleration)) + geom_point()
scat_year <- ggplot(Auto1, aes(y = mpg, x = year)) + geom_point()
scat_origin <- ggplot(Auto1, aes(y = mpg, x = origin)) + geom_point()

#install.packages('gridExtra')
library(gridExtra)
grid.arrange(scat_cylinders, scat_displacement, scat_horsepower, scat_weight,
             scat_acceleration, scat_year, scat_origin, ncol=2)
# grid.arrange(scat_cylinders, scat_displacement, scat_horsepower, scat_weight, ncol=4)
# grid.arrange(scat_acceleration, scat_year, scat_origin, ncol=3)
```

```
## box plot
box_cylinders <- ggplot(Auto1, aes(y = cylinders, x = mpg01)) + geom_boxplot()
box_displacement <- ggplot(Auto1, aes(y = displacement,x = mpg01)) + geom_boxplot()
box_horsepower <- ggplot(Auto1, aes(y= horsepower,x = mpg01)) + geom_boxplot()
box_weight <- ggplot(Auto1, aes(y= weight, x = mpg01)) + geom_boxplot()
box_acceleration <- ggplot(Auto1, aes(y=acceleration, x = mpg01)) + geom_boxplot()
box_year <- ggplot(Auto1, aes(y = year, x = mpg01)) + geom_boxplot()
box_origin <- ggplot(Auto1, aes(y = origin, x = mpg01)) + geom_boxplot()

grid.arrange(box_cylinders, box_displacement, box_horsepower, box_weight,
             box_acceleration, box_year, box_origin, ncol=2)
# grid.arrange(box_cylinders, box_displacement, box_horsepower, box_weight, ncol=4)
# grid.arrange(box_acceleration, box_year, box_origin, ncol=3)



#The corrlation table
round(cor(Auto),2)

### scatter plot matrix
pairs(Auto, pch = 10)

# A more fancy scatter plot matrix
library(psych)
pairs.panels(Auto,
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
)


###cylinders, displacement, horsepower, weight, and origin are the most important variables
Auto2 = Auto[,c(1:5,8)]


### Split the data to train and test
n = dim(Auto2)[1] ### total number of observations
n1 = round(n/10) ### number of observations randomly selected for testing data
set.seed(19930419); ### set the random seed
flag = sort(sample(1:n, n1))

Auto2train = Auto2[-flag,]
Auto2test = Auto2[flag,]
Auto2train$mpg01 <- as.factor(Auto2train$mpg01);
```

```
###LDA

library(MASS)
fit1 <- lda(Auto2train[,2:6], Auto2train[,1])
##
pred1 <- predict(fit1,Auto2train[,2:6])$class
mean( pred1  != Auto2train$mpg01)
##   0.09348442 for miss.class.train.error

mean( predict(fit1,Auto2test[,2:6])$class != Auto2test$mpg01)
## 0.1538462 for miss.class.test.error

## QDA
fit2 <- qda(Auto2train[,2:6], Auto2train[,1])
##
pred2 <- predict(fit2,Auto2train[,2:6])$class
mean( pred2!= Auto2train$mpg01)
##  0.09065156 for miss.class.train.error

mean( predict(fit2,Auto2test[,2:6])$class != Auto2test$mpg01)
## 0.2051282 for miss.class.test.error

## Naive Bayes

library(e1071)
fit3 <- naiveBayes(Auto2train[,2:6], Auto2train[,1])
pred3 <- predict(fit3, Auto2train[,2:6]);
mean( pred3 != Auto2train$mpg01)
##   0.09348442 for miss.class.train.error

mean( predict(fit3,Auto2test[,2:6]) != Auto2test$mpg01)
##  0.2051282 for miss.class.test.error

# logistic

library(nnet)
fit4 <- multinom( mpg01 ~ cylinders + displacement + horsepower + weight + origin,
data=Auto2train)
summary(fit4);
pred4<- predict(fit4, Auto2train[,2:6])
mean( pred4 != Auto2train$mpg01)
##  0.09065156 for miss.class.train.error

mean(predict(fit4,Auto2test[,2:6]) != Auto2test$mpg01)
```

```
##   0.1794872 for miss.class.test.error

# KNN
library(class)
tetrain=c(1:10)
tetest=c(1:10)
for (i in 1:10){
pred5  <- knn(train = Auto2train[,2:6], test = Auto2train[,2:6], cl = Auto2train[,1], k=i)
tetrain[i]<-round(mean( pred5 != Auto2train[,1]),3)

##tetrain for k=1:10
## 0.000 0.074 0.074 0.074 0.082 0.085 0.088 0.093 0.091 0.093
tetest[i]<-round(mean( knn(train = Auto2train[,2:6], test = Auto2test[,2:6],
cl = Auto2train[,1], k=i) != Auto2test[,1]),3)
##tetest for k=1:10
## 0.205 0.179 0.154 0.154 0.179 0.205 0.179 0.179 0.179 0.205
}
tetrain
tetest


### Average preformance

B= 100; ### number of loops
TrainErrALL = NULL
TestErrALL = NULL ### Final TE values

n = dim(Auto2)[1] ### total number of observations
n1 = round(n/10) ### number of observations randomly selected for testing data

for (b in 1:B){
  flag = sort(sample(1:n, n1))
  Auto2train = Auto[-flag,]
  Auto2test = Auto[flag,]
  Auto2train$mpg01 <- as.factor(Auto2train$mpg01);

  #(i) LDA
  fit1 <- lda(Auto2train[,2:6], Auto2train[,1])
  pred1 <- predict(fit1,Auto2train[,2:6])$class
  trainerr1 <- mean( pred1  != Auto2train$mpg01)
  testerr1 <- mean( predict(fit1,Auto2test[,2:6])$class != Auto2test$mpg01)

  #(ii) QDA
  fit2 <- qda(Auto2train[,2:6], Auto2train[,1])
  pred2 <- predict(fit2,Auto2train[,2:6])$class
```

```
  trainerr2 <- mean( pred2!= Auto2train$mpg01)
  testerr2 <- mean( predict(fit2,Auto2test[,2:6])$class != Auto2test$mpg01)

  #(iii) Naive Bayes
  fit3 <- naiveBayes(Auto2train[,2:6], Auto2train[,1])
  pred3 <- predict(fit3, Auto2train[,2:6]);
  trainerr3 <- mean( pred3 != Auto2train$mpg01)
  testerr3 <- mean( predict(fit3,Auto2test[,2:6]) != Auto2test$mpg01)

  #(iv) Logisic Regression
  fit4 <- multinom( mpg01 ~ cylinders + displacement +
  horsepower + weight, data=Auto2train)
  summary(fit4);
  pred4<- predict(fit4, Auto2train[,2:6])
  trainerr4 <- mean( pred4 != Auto2train$mpg01)
  testerr4 <- mean(predict(fit4,Auto2test[,2:6]) != Auto2test$mpg01)

  #(v) KNN
  pred5  <- knn(train = Auto2train[,2:6], test = Auto2train[,2:6],
  cl = Auto2train[,1], k=3)
  trainerr5 <- mean( pred5 != Auto2train[,1])
  testerr5 <- mean( knn(train = Auto2train[,2:6], test = Auto2test[,2:6],
  cl = Auto2train[,1], k=3) != Auto2test[,1])

  TrainErrALL = rbind( TrainErrALL,
  cbind(trainerr1, trainerr2, trainerr3, trainerr4, trainerr5) );
  TestErrALL = rbind( TestErrALL,
  cbind(testerr1, testerr2, testerr3, testerr4, testerr5) );
}

dim(TrainErrALL) ### This should be a Bx7 matrices
dim(TestErrALL)
colnames(TrainErrALL) <- c("mod1", "mod2", "mod3", "mod4", "mod5");
colnames(TestErrALL) <- c("mod1", "mod2", "mod3", "mod4", "mod5");
## You can report the sample mean and sample variances for the five models
round(apply(TrainErrALL, 2, mean),4)
round(sqrt(apply(TrainErrALL, 2, var)),4)
round(apply(TestErrALL, 2, mean),4)
round(sqrt(apply(TestErrALL, 2, var)),4)

## compare QDA with others
t.test(TestErrALL[,1], TestErrALL[,2],paired=TRUE)
#p 1.003e-06
t.test(TestErrALL[,3], TestErrALL[,2],paired=TRUE)
# p 0.0005635
```

```
t.test(TestErrALL[,4], TestErrALL[,2],paired=TRUE)
# p 4.927e-07
t.test(TestErrALL[,5], TestErrALL[,2],paired=TRUE)
# p 1.817e-06
```