Math 189: Homework 5

Alan Lui, Derek So, Xiangyu Wei

February 18, 2021

Introduction

This study looks into the a classification task focusing on automobiles' miles per gallon (MPG). We develop a model to predict whether a car gets high or low gas mileage based on the Auto data set of 392 observations on automobiles. Our analysis methods include using scatter plots and box plots to check the distributions, binarizing the MPG columns, and linear discriminant analysis (LDA) to build a model for the classification task.

Tasks & Analysis

The followings are the packages needed

```
library(ISLR)
library(tidyverse)
library(MASS)
```

Dataset

This Automobile dataset was taken from the StatLib library which is maintained at Carnegie Mellon University. It contains gas mileage, number of cylinders, engine horsepower, engine displacement, vehicle weight, acceleration time, model year, origin (1. American, 2. European, 3. Japanese), and names for 392 vehicles. The dataset was used in James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013) An Introduction to Statistical Learning with applications in R, www.StatLearning.com, Springer-Verlag, New York; and can be loaded from the `ISLR` package.

```
data(Auto)
glimpse(Auto)
```

```
## Rows: 392
## Columns: 9
## $ mpg          <dbl> 18, 15, 18, 16, 17, 15, 14, 14, 14, 15, 15, 14, 15, 14...
## $ cylinders    <dbl> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 6, 6, 6, ...
## $ displacement <dbl> 307, 350, 318, 304, 302, 429, 454, 440, 455, 390, 383,...
## $ horsepower   <dbl> 130, 165, 150, 150, 140, 198, 220, 215, 225, 190, 170,...
## $ weight       <dbl> 3504, 3693, 3436, 3433, 3449, 4341, 4354, 4312, 4425, ...
## $ acceleration <dbl> 12.0, 11.5, 11.0, 12.0, 10.5, 10.0, 9.0, 8.5, 10.0, 8....
## $ year         <dbl> 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70...
## $ origin       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, ...
## $ name         <fct> chevrolet chevelle malibu, buick skylark 320, plymouth...
```

Methods

After binarizing miles per gallon (MPG) based on the median, we used scatter plots and box plots to check which variables might be useful in predicting MPG category. We then separate the data into training set and testing set, and train a linear discriminant analysis (LDA) model using the selected features.

Analysis

1. Create a binary variable, `mgp01`, that contains 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. You can compute the median using the `median()` function.

Our first step is to binarize our data by creating a new variable mgp01, which equals 1 if mpg is greater than or equal to the median, and 0 otherwise. This simplifies the target predictions for the model we will create later.

```
Auto$mpg01 <- ifelse(Auto$mpg > median(Auto$mpg), 1, 0)
```

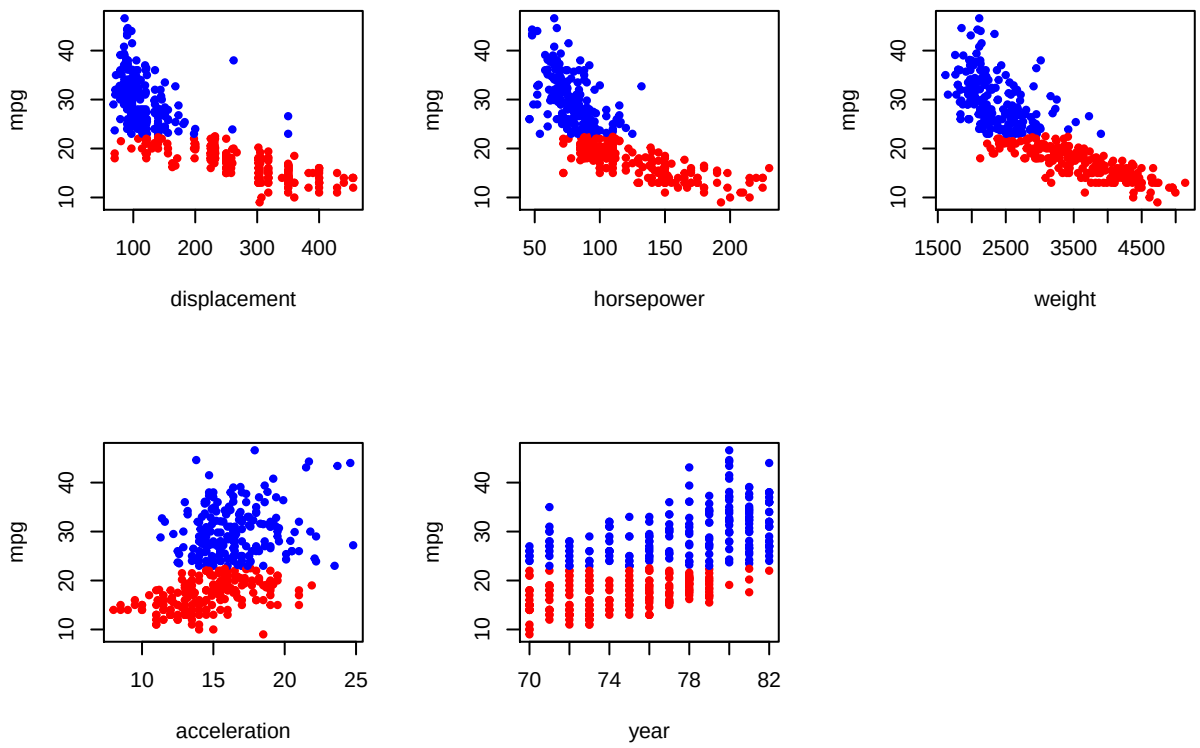
2. Explore the data graphically in order to investigate the association between mgp01 and the other features. Which of the other features seem most likely to be useful in predicting mgp01? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

We decide to exclude the two categorical variables `origin` and `name` are dropped since LDA uses a probabilistic model and does not work on categorical variables. The ordinal variable `cylinders` is also excluded for the same reason mentioned above.

```
Auto <- Auto[, -which(names(Auto) %in% c("origin","name","cylinders"))]
```

We first use a scatter plot to show the relationship between each kept variable (`displacement`, `horsepower`, `weight`, `acceleration`, `year`) with `mpg`. Red represents `mpg` that is smaller than or equals to the median, and blue represents `mpg` that is higher than the median.

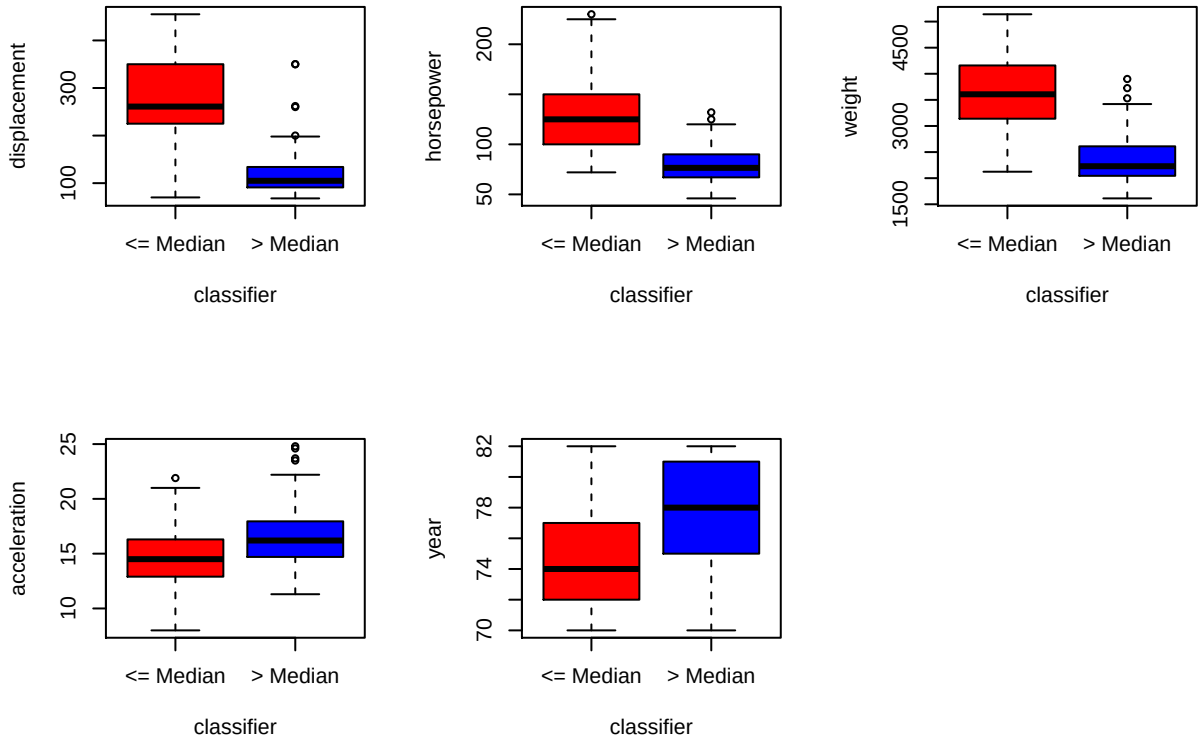
```
col_names = names(Auto)[-c(1,7)] # Auto column names
par(mfrow = c(2, 3)) # format the graphs
col_scale = ifelse(Auto$mpg > median(Auto$mpg), "blue", "red")
# loop through all columns to plot scatterplots
for(i in 1:length(col_names))
{
  plot(Auto[,i+1], Auto$mpg, pch = 20, col = col_scale,
       xlab = col_names[i], ylab = "mpg")
}
```



From the scatter plots above, we can see that there is a difference in the relationship of `mpg` larger than median (blue) with each variable & the relationship of `mpg` smaller than or equal to median (red) with each variable. For example, in `displacement`, `horsepower` and `weight` scatter plots, the slope for blue is more negative/steep than the slope for red. And in `acceleration` and `year` scatter plots, the intercept for blue is higher than the intercept for red.

Then, we use box plots to compare central tendencies and spreads of two `mpg01` category.

```
par(mfrow = c(2, 3)) # format the graphs
# loop through all columns to plot boxplots
for(i in (1:length(col_names)))
{
  df <- data.frame(y = Auto[,i+1], x = as.factor(Auto$mpg01)) # df to plot
  boxplot(data = df, y ~ x, col = c("red", "blue"),
          names = c("<= Median","> Median"),
          xlab = "classifier", ylab = col_names[i])
}
```



With box plots, we can see that `displacement`, `horsepower` and `weight` have distinctive distributions by binary `mpg01`, while `acceleration` and `year` seem to have closer and more similar distributions.

Thus, we decide to include all those 5 continuous variables into the model.

3. Split the data into a training set of size 300 and a test set of size 92.

After exploration, we split our data **randomly** into an approximate 75/25 train test split, or 300/92.

```
n_train <- 300
train_ind = sample(1:length(Auto$mpg01), n_train) # random sample 300
train = Auto[train_ind,-1] # get training set
test = Auto[-train_ind,-1] # get testing set
```

4. Perform LDA on the training data in order to predict mgp01 using the variables that seemed most associated with mgp01.

The following are some codes to train the data on training set using LDA based on `displacement`, `horsepower`, `weight`, `acceleration`, and `year`.

```
# separate dataset
Auto_mpg0 <- Auto[Auto$mpg <= median(Auto$mpg),]
Auto_mpg1 <- Auto[Auto$mpg > median(Auto$mpg),]
n_mpg0 <- nrow(Auto_mpg0)
n_mpg1 <- nrow(Auto_mpg1)
```

```
# sample mean vectors
Mean_mpg0 <- colMeans(train[train$mpg01 == 0,1:5])
Mean_mpg1 <- colMeans(train[train$mpg01 == 1,1:5])

# pooled sample covariance
S_mpg0 <- cov(train[train$mpg01 == 0,1:5])
S_mpg1 <- cov(train[train$mpg01 == 1,1:5])
S_pooled <- ((n_mpg0-1)*S_mpg0 +(n_mpg1-1)*S_mpg1)/(n_mpg0+n_mpg1-2)

# intercepts of LDA
S_inv <- solve(S_pooled)
alpha_mpg0 <- -0.5* t(Mean_mpg0) %*% S_inv %*% Mean_mpg0 + log(n_mpg0/n_train)
alpha_mpg1 <- -0.5* t(Mean_mpg1) %*% S_inv %*% Mean_mpg1 + log(n_mpg1/n_train)

# slope coefficients of LDA
beta_mpg0 <- S_inv %*% Mean_mpg0
beta_mpg1 <- S_inv %*% Mean_mpg1
```

5. Classify the test data. Discuss the results in terms of the proportion of correctly classified records.

We then predict training `mpg` category and testing `mpg` category using the LDA model we trained.

```
prediction.train <- c()
prediction.test <- c()
label <- c(0, 1)
# prediction for training set
for(i in 1:nrow(train)){
  #Read an ovservation in train/test data
  x.train <- t(train[i,1:5])

  #Calculate linear discriminant functions for each
  d_mpg0 <- alpha_mpg0 + t(beta_mpg0) %*% x.train
  d_mpg1 <- alpha_mpg1 + t(beta_mpg1) %*% x.train

  #Classify the observation to the class with highest function value
  d_vec <- c(d_mpg0, d_mpg1)
  prediction.train <- append(prediction.train, label[which.max( d_vec )])
}

# prediction for testing set
for(i in 1:nrow(test)){
  #Read an ovservation in train/test data
  x.test <- t(test[i,1:5])

  #Calculate linear discriminant functions for each
  d_mpg0 <- alpha_mpg0 + t(beta_mpg0) %*% x.test
  d_mpg1 <- alpha_mpg1 + t(beta_mpg1) %*% x.test

  #Classify the observation to the class with highest function value
  d_vec <- c(d_mpg0, d_mpg1)
  prediction.test <- append(prediction.test, label[which.max( d_vec )])
}
```

```r
# train accuracy
mean(train$mpg01 == prediction.train)
```

```
## [1] 0.89
```

```r
# test accuracy
mean(test$mpg01 == prediction.test)
```

```
## [1] 0.9347826
```

From the above, we can see that the accuracy rate for both training set and testing set is around **90%**.

Verification We also verify the results using `lda()` function from `MASS` package and the results are the same.

```r
model = lda(mpg01~., data = train) # build model
pred.train = predict(model, train)$class # predict train class
pred.test = predict(model, test)$class # predict test class
# train accuracy
mean(train$mpg01 == pred.train)
```

```
## [1] 0.89
```

```r
# test accuracy
mean(test$mpg01 == pred.test)
```

```
## [1] 0.9347826
```

Conclusion

Using a linear discriminant analysis (LDA) model, we created an effective classifier for high and low miles per gallon (MPG) based on automible data. The classification accuracy is around 90% which is verified by both manual calculation and package `lda()` function calculation. There are some concerns of overfitting/underfitting, but outside of a different train test split or removing more variables from the model, the results are satisfactory.

Discussion

Other models could shed more light on the relationship between some of these variables like a logistic regression, which would serve a similar purpose but have different outcomes in this scenario. Using a linear regression model rather than linear discriminant analysis was also a possibility, but is functionally much more different.

Appendix

Linear Discriminant Analysis (LDA)

The following is adapted from lecture 13 and lecture 14.

- Given a training dataset, we can estimate the linear discriminant function by:

$$\widehat{d}_k^L(\underline{x}) = -\frac{1}{2}\overline{\underline{x}}^{(k)\prime}\mathbf{S}^{-1}\overline{\underline{x}}^{(k)} + \overline{\underline{x}}^{(k)\prime}\mathbf{S}^{-1}\underline{x} + \log p_k$$

  where $\overline{\underline{x}}^{(k)}$ is the sample mean of group k, and $\mathbf{S}$ is the pooled sample covariance matrix.
- This is a function of the sample mean vectors, the pooled covariance matrix, and prior probabilities for g different populations.
- The estimated linear discriminant function can also be written as a linear form:

$$\begin{aligned}\widehat{d}_k^L(\underline{x}) &= \widehat{\alpha}_k + \widehat{\underline{\beta}}_k'\underline{x}\\ \widehat{\alpha}_k &= -\frac{1}{2}\overline{\underline{x}}^{(k)\prime}\mathbf{S}^{-1}\overline{\underline{x}}^{(k)} + \log p_k\\ \widehat{\underline{\beta}}_k &= \mathbf{S}^{-1}\,\overline{\underline{x}}^{(k)\prime}.\end{aligned}$$

Procedure for LDA: Given a training dataset which contains g populations, suppose we have a new observation $\underline{x}$ that is to be classified into one of the g populations, we can apply linear discriminant analysis as follows:

1. Choose priors p_k for $k = 1, 2, \ldots, g$.
2. Calculate sample mean vectors $\overline{\underline{x}}_k$ for $k = 1, 2, \ldots, g$.
3. Calculate pooled sample covariance matrix

$$\mathbf{S} = \frac{1}{N-g}\sum_{k=1}^{g}\sum_{i=1}^{n_k}\left(\underline{x}_i^{(k)} - \overline{\underline{x}}^{(k)}\right)\left(\underline{x}_i^{(k)} - \overline{\underline{x}}^{(k)}\right)',$$

4. Calculate coefficients

$$\widehat{\alpha}_k = -\frac{1}{2}\overline{\underline{x}}^{(k)\prime}\mathbf{S}^{-1}\overline{\underline{x}}^{(k)} + \log p_k$$

5. Calculate coefficients

$$\widehat{\underline{\beta}}_k = \mathbf{S}^{-1}\,\overline{\underline{x}}^{(k)}$$

6. Calculate

$$\widehat{d}_k^L(\underline{x}) = \widehat{\alpha}_k + \widehat{\underline{\beta}}_k'\underline{x}$$

   for $k = 1, 2, \ldots, g$.
7. Classify $\underline{x}$ into the group with highest value of $\widehat{d}_k^L(\underline{x})$.