Math 189: hw 5 solution

TA

Feb 2021

References:

Math 189: Classification via Discriminant Analysis I, Professor Tucker S. McElroy, 2021 Winter.

Math 189: Classification via Discriminant Analysis II, Professor Tucker S. McElroy, 2021 Winter.

Auto Mileage

Data:

We develop a model to predict whether a given car gets high or low MPG based on the Auto data set. The dataset has 392 observations on automobiles. For each automobile, we record the miles per gallon (MPG), among other variables such as horsepower and weight. This data can be found in the **ISLR** package.

```
library(ISLR)
data(Auto)
```

Tasks

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.

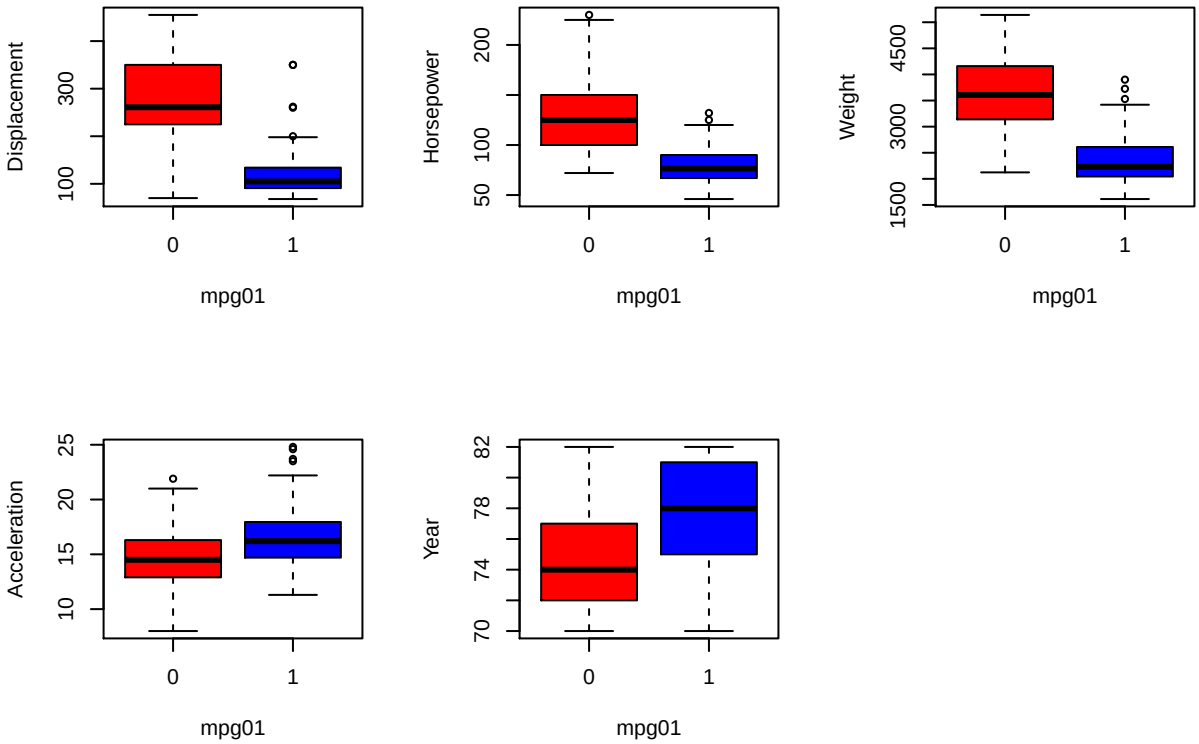
```
mpg.median = median(Auto$mpg)
mpg01 = factor(Auto$mpg >= mpg.median)
levels(mpg01) = c("0", "1")
```

Categorical variables in R are stored into a factor. We set the class of **mpg01** to be factor, and set its levels to be "0" and "1".

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 draw boxplots for five variables: **displacement**, **horsepower**, **weight**, **acceleration** and **year** versus the new binary variable **mpg01**. Scatterplots are also acceptable for this question.

```
par(mfrow = c(2, 3))
plot(Auto$displacement ~ mpg01, col = c("red", "blue"), ylab = "Displacement")
plot(Auto$horsepower ~ mpg01, col = c("red", "blue"), ylab = "Horsepower")
plot(Auto$weight ~ mpg01, col = c("red", "blue"), ylab = "Weight")
plot(Auto$acceleration ~ mpg01, col = c("red", "blue"), ylab = "Acceleration")
plot(Auto$year ~ mpg01, col = c("red", "blue"), ylab = "Year")
```



Based on the above five boxplots, it's evident that the distribution of these vriables are different for **mpg01 == 0** and **mpg01 == 1**, so it's reasonable to believe they are associated with the response. Regarding the other variables, **name** is not numerical, so we can't find a proper way to visualize it. The remaining two features **cylinders** and **origin** are oridinal and categorical accordingly, they may be associated with **mpg01**, but incorporating them into the LDA model will violate the multivariate normality assumption. For other classifiers like logistic regression, thers's a way to include categorical predictors by creating dummy variables, but for the probabilistic method LDA which relies heavily on the distribution assumption, it's inappropriate to include them. In summary, we exclude three features from the classification model: **cylinders**, **origin** and **name**.

Grading: Students can still get points if they include **cylinders** (by considering it as a continuous variable), but it's inappropriate to include **origin** or **name**. It's also wrong to use **mpg** as a predictor, since we can't predict something based on itself.

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

We create a random set of indices, and split the data based on these indices. The code is reproducible since we set a seed.

```
Auto$mpg01 = mpg01
n = nrow(Auto)
train_size = 300
test_size = 92
set.seed(2021)
index = sample(1:n, train_size)
data_train = Auto[index, ]
data_test = Auto[-index, ]
```

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

We set the prior distributions as sample proportions, then use the five variables selected in step 2 to train a LDA model.

```
n_0 = sum(data_train$mpg01 == 0)
n_1 = sum(data_train$mpg01 == 1)
p_0 = n_0 / train_size
p_1 = n_1 / train_size

Mean_0 = colMeans(data_train[data_train$mpg01 == 0, 3:7])
Mean_1 = colMeans(data_train[data_train$mpg01 == 1, 3:7])
S_0 = cov(data_train[data_train$mpg01 == 0, 3:7])
S_1 = cov(data_train[data_train$mpg01 == 1, 3:7])
S_pooled = ((n_0 - 1) * S_0 + (n_1 - 1) * S_1) / (n_0 + n_1 - 2)
S_inv = solve(S_pooled)

alpha0 = -0.5 *  t(Mean_0) %*% S_inv %*% Mean_0 + log(p_0)
alpha1 = -0.5 * t(Mean_1) %*% S_inv %*% Mean_1 + log(p_1)
alpha = c(alpha0, alpha1)

beta0 = S_inv %*% Mean_0
beta1 = S_inv %*% Mean_1
beta = cbind(beta0, beta1)
```

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

```
pred = rep(0, test_size)
for (i in 1:test_size) {
  ld.value = alpha + as.matrix(data_test[i, 3:7]) %*% beta
  pred[i] = ld.value[, 2] > ld.value[, 1]
}
sum(pred == data_test$mpg01) / test_size
```

```
## [1] 0.9456522
```

The prediction accuracy on the test set is 94.57%, which is fairly high. However, the result may be taken with some grain of salt due to two reasons: 1) The sample size 392 is relative small, especially for modern statistics; 2) The LDA model depends on the assumptions, which need to be justified. In addition, it's hard to incorporate categorical variables into the LDA model, but **cylinders** and **origin** may also be related to MPG. In the future, we can try other classifiers such as logistic regression for more comprehensive analysis.