2019 학년도 1학기
DATA MINING
HW4

과목명	데이터마이닝
담당교수명	송종우 교수님
제출일	2019.04.10
학번	182STG27
이름	임지연

Ⅰ. Description

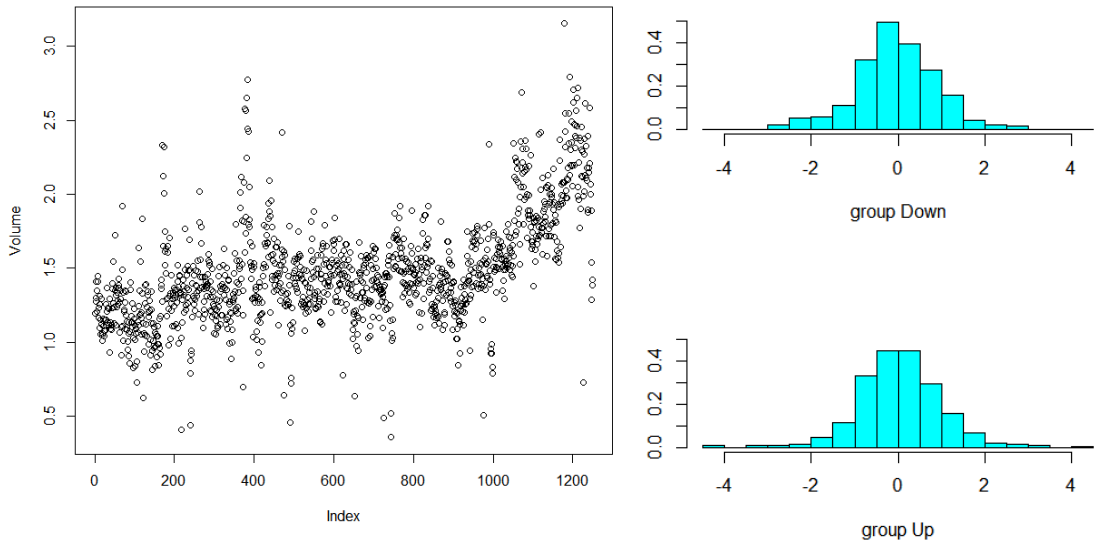
일반적으로 Classification 문제에 대해서 Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, K-Nearest Neighbors 모형을 주로 사용한다. 각각 모형별로 특징이 있으며 데이터에 따라서 성능이 좋은 모델이 달라지게 된다. 따라서 분석 시 상황에 맞게 가장 성능이 좋은 모델을 선택하게 된다. 본 과제에서는 R 에서 제공하는 함수인 glm, lda, qda, knn 함수를 이용하여 데이터를 적합해 본 후, 그 의미를 알고자 한다. Lab 에 나와있는 Smarket, Caravan 데이터에 대해서 분석할 수 있는 예제 코드를 실행해본 후, Example 4, 10,11 문제를 풀어보며 실제 데이터에 적합해보는 연습을 한다.

Ⅱ. Implementation

Question 1.

Lab : Logistic Regression, LDA, QDA, KNN 의 코드를 실행해보고 감상문을 써라

R에서 Smarket 데이터에 대해서 Logistic Regression을 적합하기 위해서 glm함수를 사용해 모든 변수를 포함한 모델을 적합해 본 결과 약 52%의 정확도가 나왔다. 또한 Lag1, Lag2 변수만 포함된 모형은 약 56 %의 정확도를 가졌다. 또한 MASS 패키지를 이용하여 LDA, QDA 모형을 Lag1, Lag2 변수만 포함된 모형에 적합해 본 결과 두 모델 모두 약 56%의 정확도를 갖는 것을 알 수 있었다. Class 패키지를 이용하여 knn 모형을 적합해 본 결과 k=1일 때 50%, k=3일 때 53%의 정확도를 갖는 것을 알 수 있었다. 아래 그래프는 각각 Smarket 데이터에 대한 Volume변수에 대한 그래프와 lda 모델의 적합결과이다.



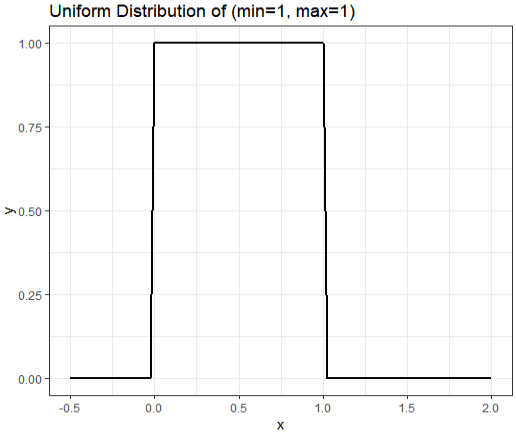
Question 2.

4.7 Excercises - Example 4, 10, 11 풀어라

[Example 4]

When the number of features p is large, there tends to be a deterioration in the performance of KNN and other local approaches that perform prediction using only observations that are near the test observation for which a prediction must be made. This phenomenon is known as the curse of dimensionality, and it ties into the fact that non- parametric approaches often perform poorly when p is large. We mensionality will now investigate this curse.

(a) Suppose that we have a set of observations, each with measurements on p = 1 feature, X. We assume that X is uniformly (evenly) distributed on [0, 1]. Associated with each observation is a response value. Suppose that we wish to predict a test observation's response using only observations that are within 10% of the range of X closest to that test observation. For instance, in order to predict the response for a test observation with X = 0.6, we will use observations in the range [0.55, 0.65]. On average, what fraction of the available observations will we use to make the prediction?



▶ [0,1] 범위를 갖는 균등분포를 가정할 때, [0.55, 0.65] 범위의 값을 사용한다면 pdf 의 면적이 곧 확률을 의미하기 때문에 관측치 중 $\frac{65-55}{1-0}$ = 10% 를 사용하는 것이다.

(b) Now suppose that we have a set of observations, each with measurements on p = 2 features, X1 and X2. We assume that (X1,X2) are uniformly distributed on [0, 1] × [0, 1]. We wish to predict a test observation's response using only observations that are within 10% of the range of X1 and within 10% of the range of X2 closest to that test observation. For instance, in order to predict the response for a test observation with X1 = 0.6 and X2 = 0.35, we will use observations in the range [0.55, 0.65] for X1 and in the range [0.3, 0.4] for X2. On average, what fraction of the available observations will we use to make the prediction?

▶ (0.65 - 0.55) x (0.4-0.3) = 1% 를 사용하는 것이다.

(c) Now suppose that we have a set of observations on p = 100 features. Again the observations are uniformly distributed on each feature, and again each feature ranges in value from 0 to 1. We wish to predict a test observation's response using observations within the 10% of each feature's range that is closest to that test observation. What fraction of the available observations will we use to make the prediction?

▶ 0.1^{100} 즉 거의 0 에 가까운 값이 된다. 따라서 이 의미는 p 값이 100 으로 매우 큰 경우 예측 가능한 관측치가 매우 적어진다는 것을 알 수 있다.

(d) Using your answers to parts (a)–(c), argue that a drawback of KNN when p is large is that there are very few training observations "near" any given test observation.

▶ P(차원수) 가 증가할 때 앞에서 볼 수 있듯 test 데이터와 가까운 주변 값이 기하급수적으로 감소한다.

(e) Now suppose that we wish to make a prediction for a test observation by creating a p-dimensional hypercube centered around the test observation that contains, on average, 10% of the training observations. For p = 1, 2, and 100, what is the length of each side of the hypercube? Comment on your answer.

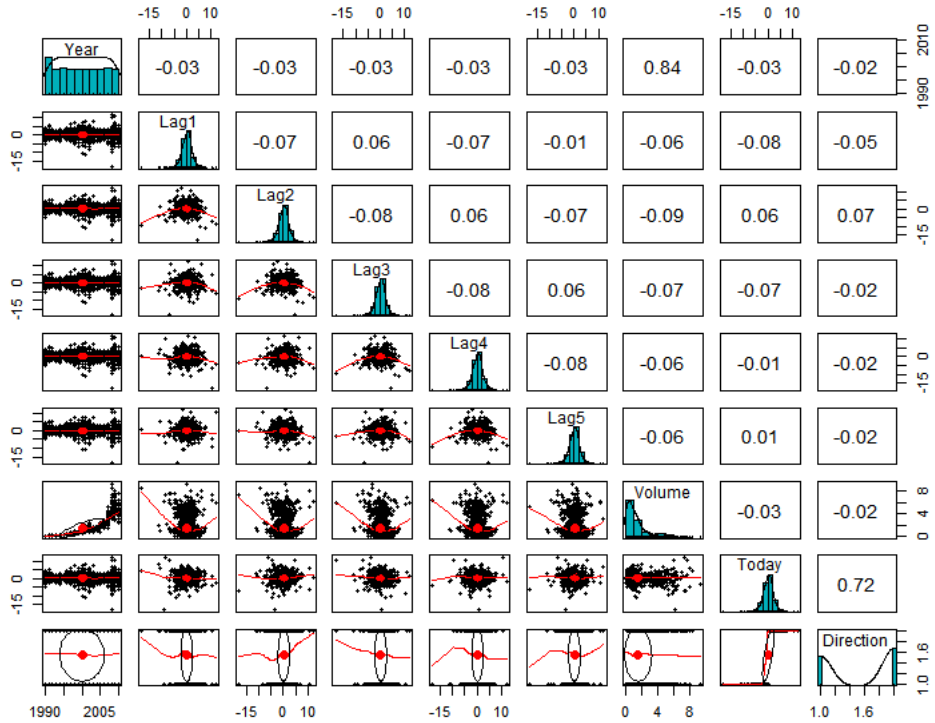
Note: A hypercube is a generalization of a cube to an arbitrary number of dimensions. When p = 1, a hypercube is simply a line segment, when p = 2 it is a square, and when p = 100 it is a 100-dimensional cube.

▶ p = 1 일 때, 0.1 , p = 2 일 때, $0.1^{1/2}$, p = 100 일 때, hypercube 의 각 면의 길이는 $0.1^{1/100}$ 값을 갖는다. 동일한 범위의 공간에 균일하게 분포된 데이터에서 p 값이 증가할수록 한 변의 길이가 점점 커지며 따라서 데이터를 포함할 확률이 작아지게 된다.

[Example 10]

This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1, 089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

(a) Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?



▶ 왼쪽 아래 부분의 각 변수들 끼리의 산점도를 그려본 결과 대체적으로 수평 그래프가 그려진다. 하지만 Year 변수와 Volume 변수의 경우에 양의 관계가 있어보인다.

(b) Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	0.267	0.086	3.106	0.002
Lag1	-0.041	0.026	-1.563	0.118
Lag2	0.058	0.027	2.175	0.03
Lag3	-0.016	0.027	-0.602	0.547
Lag4	-0.028	0.026	-1.05	0.294
Lag5	-0.014	0.026	-0.549	0.583
Volume	-0.023	0.037	-0.616	0.538

▶ Lag2 변수의 경우에만 p-value = 0.03 으로 0.05 보다 작은 값을 가지기 때문에 통계적으로 가장 유의미한 값을 가진다는 것을 알 수 있다.

(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

		Observe	
		Down	Up
Predict	Down	54	48
	Up	430	557

▶ Confusion Mastix 를 작성해 보았을 때, 정확도는 약 56%로 나타난다.

(d) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

		Observe	
		Down	Up
Predict	Down	9	5
	Up	34	56

▶ Confusion Mastix 를 작성해 보았을 때, 정확도는 약 62%로 나타난다.

(e) Repeat (d) using LDA.

		Observe	
		Down	Up
Predict	Down	9	5
	Up	34	56

▶ Confusion Mastix 를 작성해 보았을 때, 정확도는 약 62%로 나타난다.

(f) Repeat (d) using QDA.

		Observe	
		Down	Up
Predict	Down	0	0
	Up	43	61

▶ Confusion Mastix 를 작성해 보았을 때, 정확도는 약 59%로 나타난다.

(g) Repeat (d) using KNN with K = 1.

		Observe	
		Down	Up
Predict	Down	19	21
	Up	24	40

▶ Confusion Mastix 를 작성해 보았을 때, 정확도는 약 50%로 나타난다.

(h) Which of these methods appears to provide the best results on this data?

▶ Logistic, LDA 모델이 가장 좋은 성능을 나타낸다는 것을 알 수 있다.

(i) Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier.

k	accuracy
1	0.5
2	0.509615385
3	0.548076923
...	
11	0.557692308
12	0.557692308
13	0.596153846
14	0.567307692
15	0.586538462
16	0.576923077
17	0.586538462
18	0.567307692
19	0.567307692
20	0.586538462

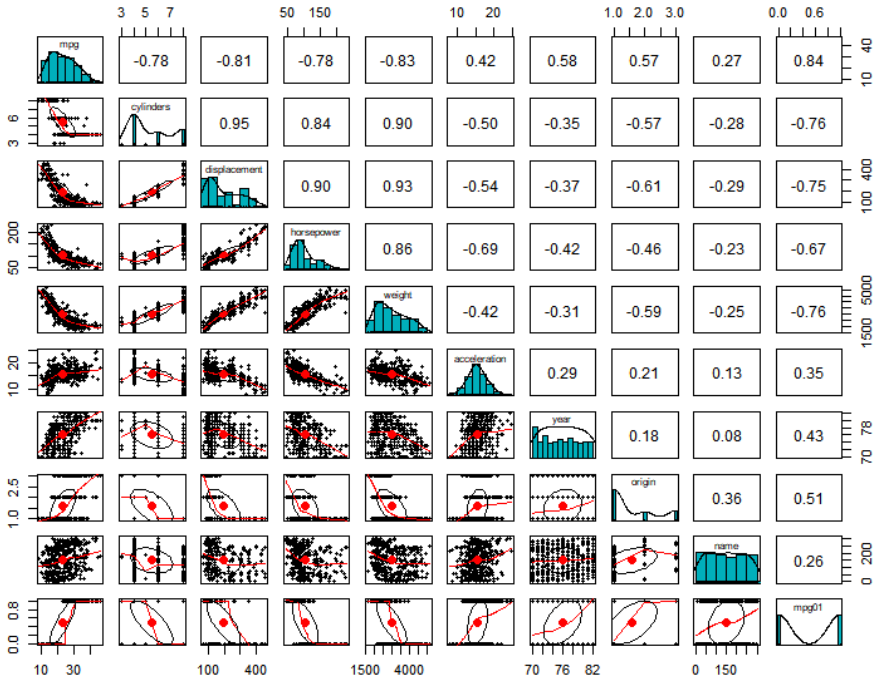
▶ k = 13 일때의 accuracy 값이 가장 크다. 따라서 가장 적합하다고 할 수 있다.

[Example 11]

In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set.

(a) Create a binary variable, mpg01, that contains a 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. Note you may find it helpful to use the data.frame() function to create a single data set containing both mpg01 and the other Auto variables.

(b) Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.



▶ 새로 생성한 mpg01 변수와 가장 상관관계가 높은 변수는 cylinders, displacement, horsepower, weight 변수의 상관계수 값이 가장 높은 것으로 보아 mpg01 변수의 예측에 유의한 영향을 미칠 것으로 예상된다.

(c) Split the data into a training set and a test set.

(d) Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

		Observe	
		0	1
Predict	0	49	1
	1	8	60

▶ Confusion Mastix 를 작성해 보았을 때, 오분류율은 약 7.6%로 나타난다.

(e) Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

		Observe	
		0	1
Predict	0	49	4
	1	8	57

▶ Confusion Mastix 를 작성해 보았을 때, 오분류율은 약 10.1%로 나타난다.

(f) Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

		Observe	
		0	1
Predict	0	47	9
	1	10	52

▶ Confusion Mastix 를 작성해 보았을 때, 오분류율은 약 16.1%로 나타난다.

(g) Perform KNN on the training data, with several values of K, in order to predict mpg01. Use only the variables that seemed most associated with pg01 in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?

k	accuracy
1	0.839
2	0.822
...	
28	0.898
29	0.915
30	0.915
31	0.907
32	0.915
33	0.907
...	
98	0.890
99	0.898
100	0.890

▶ k = 30 근처에서의 accuracy 값이 매우 높음을 알 수 있다.

Ⅲ. Discussion

이번 과제를 통해 크게 두 가지에 대한 공부를 할 수 있었다. 첫번째로는 대표적인 Classification 모형인 Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, K-Nearest Neighbors 모형을 R을 이용해서 적합 후 가장 성능이 좋은 모형을 찾는 일련의 과정 또는 방법에 대한 공부를 했다. 앞서 언급한 듯 데이터에 따라서 성능이 좋은 모델이 달라지게 된다. 본 과제에서 예제데이터로 분석해본 결과는 Linear Discriminant Analysis 모형이 가장 성능이 좋은 것으로 나타났다. 하지만 각각의 모형이 가지고 있는 특징이 있는데, 먼저 Logistic 모델은 multi-class 에 취약하지만 설명변수가 정량, 정성변수일 때 모두 사용 가능하다. LDA, QDA 모델은 multi-class 에 있어서 모수 추정이 안정적이지만 설명변수에 정량변수만 가능하며 정성변수는 포함시킬 수 없다. 따라서 상황에 따라 적절한 모형을 찾아서 사용하는 것이 필요할 것이다.

두번째로는, K-NN은 첫 번째 과제에서 살펴보았듯 차원의 저주 문제로 인해 차원(p)가 커질 때, 데이터들이 서로 너무 멀리 떨어져 있기 때문에 예측에 있어서 불안정한 결과 등 부정적인 영향이 있을 수 있다는 것을 값으로 계산하며 다시 공부하게 되었다.

Ⅳ. Appendix – R code

```
##Example 4
#a
x = runif(100)
library(ggplot2)
ggplot(data.frame(x=c(-0.5,2)), aes(x=x)) + stat_function(fun=dunif, args=list(min = 0, max = 1), colour="black", size=1)+   ggtitle("Uniform Distribution of (min=1, max=1)") +theme_bw()
(( 0.65 - 0.55)/(1-0))*100
#b
(( 0.65 - 0.55)*(0.4-0.3))*100
#c
0.1^(100)
##Example 10
# a
summary(Weekly)
library(psych)
pairs.panels(Weekly,
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,   # show density plots
             ellipses = TRUE )# show correlation ellipses
# b
glm.fit = glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, family = "binomial", data= Weekly)
summary(glm.fit)
#c
glm.probs = predict(glm.fit, Weekly, type = "response")
glm.pred = ifelse(glm.probs > 0.5, "Up", "Down")
table(glm.pred, Weekly$Direction)
mean(glm.pred == Weekly$Direction)
#d
train = Weekly %>% filter (Year %in% c(1990:2008) )
test = Weekly %>% filter (!(Year %in% c(1990:2008)) )
glm.fit = glm(Direction ~ Lag2, data=train, family="binomial")
glm.probs = predict(glm.fit, test, type="response")
```

```
glm.pred = ifelse(glm.probs > 0.5, "Up", "Down")
table(glm.pred, test$Direction)
mean(glm.pred==test$Direction)
#e
library(MASS)
lda.fit = lda(Direction ~ Lag2, data=train)
lda.pred = predict(lda.fit, test)$class
table(lda.pred, test$Direction)
mean(lda.pred==test$Direction)
#f
qda.fit = qda(Direction ~ Lag2, data=train)
qda.pred = predict(qda.fit, test)$class
table(qda.pred, test$Direction)
mean(qda.pred==test$Direction)
#g
set.seed(1)
train.X = as.matrix(train$Lag2); test.X = as.matrix(test$Lag2)
knn.pred = knn(train.X, test.X, train$Direction, k=1)
table(knn.pred, test$Direction)
mean(knn.pred == test$Direction)
#h #i
knn.pred = list(); aa = c()
for (i in 1:30){
set.seed(1)
knn.pred[[i]] <- knn(train.X, test.X, train$Direction, k=i)
aa[i] = mean(knn.pred[[i]] == test$Direction) }
mytable = data.frame(cbind("k" = 1:30, "accuracy" = aa))
##Example 11
#a
mpg01 = ifelse(Auto$mpg > median(Auto$mpg),1,0 )
myAuto = data.frame(Auto, mpg01)
#b
pairs.panels(myAuto, method = "pearson", # correlation methodhist.col = "#00AFBB",density = TRUE,   # show density plots
                ellipses = TRUE )# show correlation ellipses
#c
set.seed(1)
trainid = sample(1:nrow(myAuto), nrow(myAuto)*0.7 , replace=F)
train = myAuto[trainid,]
test = myAuto[-trainid,]
#d
lda.fit = lda(mpg01~cylinders+displacement+horsepower+weight, data=train)
lda.pred = predict(lda.fit, test)$class
table(lda.pred, test$mpg01)
mean(lda.pred != test$mpg01)
#e
qda.fit <- qda(mpg01~cylinders+displacement+horsepower+weight, data=train)
qda.pred <- predict(qda.fit, test)$class
table(qda.pred, test$mpg01)
mean(qda.pred != test$mpg01)
#f
glm.fit =glm(mpg01~cylinders+displacement+horsepower+weight, data=train, family=binomial)
glm.probs <- predict(glm.fit, test, type="response")
glm.pred <- ifelse(glm.probs > 0.5, 1, 0)
table(glm.pred, test$mpg01)
mean(glm.pred != test$mpg01)
#g
train.X = cbind(train$cylinders, train$displacement, train$horsepower, train$weight )
test.X = cbind(test$cylinders, test$displacement, test$horsepower, test$weight)
knn.pred = list(); aa = c()
for (i in 1:100){
set.seed(1)
knn.pred[[i]] <- knn(train.X, test.X, train$mpg01, k=i)
aa[i] = mean(knn.pred[[i]] == test$mpg01) }
mytable = data.frame(cbind("k" = 1:100, "accuracy" = aa))
```