Homework4\_Self\_correction

**Q1:** Wrong. I think I had an incorrect understanding about the question. I thought that the question meant X distributions for the two outcomes are different, inducing different variances, instead of the two distributions of Y.

**Q2**: Correct, but not perfect. My answer talks very little about the flexibility of the three methods. Specifically, for logistic regression, flexibility depends on the number of predictor variables used (more variables = higher flexibility); For LDA, flexibility depends on whether equal variances are assumed for each underlying distribution, but within the standard LDA there is no way to manipulate flexibility of the model; for kNN, flexibitliy is determined by K, such that lower values of K result in more flexible models.

**Q3**: Not exactly correct, but close to correct. The correct answer describes the extreme scenario where the number of predictors approaches the number of observations; my answer says that increasing the number of predictors (features) to some point will be increasingly degrade the performance of classifiers.

For the problems of kNN, the correct answer describes the scenario of high dimentionality where observations do not have enough nearby neighbors for the kNN. My answer says the same thing, but not that exact.

**Q4**: (a) Correct, not perfect: may add pairs(Weekly) to produce a matrix of scatterplot and pointing out that year and volume appear to have a relationship.

(b) Correct.

(c) Correct, but explanations are not enough: for example, pointing out that when the weeks the market goes up the logistic regression is right most of the time: 557/(557+48)=92.1%, whereas during weeks the market goes down, the logistic regression is wrong most of the time 54/(430+54)=11.2%.

(d) Correct.

(e) Correct.

(f) Correct.

(g) Incorrect. I made mistakes in the bracket of a function:i.e., cbind(Lag2)[Train]. That’s also why the program produced error in output.

**# I think I uploaded a wrong version of my code and all the answers in the following are blank. I am sorry.**

(h) Incorrect. Logistic regression and LDA methods provide similar test error rates.

(i)Sorry my answer is blank, so I pasted my answer here for correction.

*# kNN, k = 10*

knn.pred = knn(train.X, test.X, train.Direction, k = 10)

table(knn.pred, Direction.0910)

mean(knn.pred == Direction.0910)

*# 0.5769*

*# kNN, k=100*

knn.pred = knn(train.X, test.X, train.Direction, k = 100)

table(knn.pred, Direction.0910)

mean(knn.pred == Direction.0910)

*# 0.5577*

**# I think I uploaded a wrong version of my code and all the answers in the following are blank. I am sorry.**

**Q5**: for question (g), I found that I used a different set of number for K: 5, 20 and 120. The results turned out that when k=20, the knn classifier produced smallest test errors.

# (a)

library(ISLR)

summary(Auto)

attach(Auto)

mpg01 = rep(0, length(mpg))

mpg01[mpg > median(mpg)] = 1

Auto = data.frame(Auto, mpg01)

# (b)

cor(Auto[, -9])

pairs(Auto)

# The correlations between mpg, cylinders, displacement, horsepower and weight are relatively high

# (c)

train = (year%%2 == 0)

test = !train

Auto.train = Auto[train, ]

Auto.test = Auto[test, ]

mpg01.test = mpg01[test]

# (d)

# LDA

library(MASS)

lda.fit = lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto,

subset = train)

lda.pred = predict(lda.fit, Auto.test)

table(lda.pred$class==mpg01.test)

mean(lda.pred$class != mpg01.test)

# 0.1264 = 23/(23+159)

# (e)

# QDA

qda.fit = qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto,

subset = train)

qda.pred = predict(qda.fit, Auto.test)

table(qda.pred$class==mpg01.test)

mean(qda.pred$class != mpg01.test)

# 0.132 = 24/(24+158)

# (f)

# Logistic regression

glm.fit = glm(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto,

family = binomial, subset = train)

glm.probs = predict(glm.fit, Auto.test, type = "response")

glm.pred = rep(0, length(glm.probs))

glm.pred[glm.probs > 0.5] = 1

table(glm.pred == mpg01.test)

mean(glm.pred != mpg01.test)

# 0.121 = 22/(22+160)

# (g)

library(class)

train.X = cbind(cylinders, weight, displacement, horsepower)[train, ]

test.X = cbind(cylinders,weight, displacement, horsepower)[test, ]

train.mpg01 = mpg01[train]

set.seed(1)

# KNN(k=5)

knn.pred = knn(train.X, test.X, train.mpg01, k = 5)

table(knn.pred ==mpg01.test)

mean(knn.pred != mpg01.test)

# 0.148

# k=5, 14.8% test error rate.

# KNN(k=20)

knn.pred = knn(train.X, test.X, train.mpg01, k = 20)

table(knn.pred ==mpg01.test)

mean(knn.pred != mpg01.test)

# 0.132

#k=20, 13.2% test error rate and seems to produce the smallest test error rate.

# KNN(k=120)

knn.pred = knn(train.X, test.X, train.mpg01, k = 120)

table(knn.pred ==mpg01.test)

mean(knn.pred != mpg01.test)

# 0.170

#k=120, 17% test error rate.