Øving 3

2022-04-28

Problem 1

c) If the Bayes decision boundary is highly non-linear, would we expect the best value of K to be large or small? Why?

When $k \to \infty$, the model looks more linear. Therefore we want a rather small K, but not too small.

Problem 2

b) Explain the assumptions made to use linear discriminant analysis to classify a new observation to be a genuine or fake bank note. Write down the classification rule for a new observation.

In linear discriminant analysis we assume that the independent variables are normally distributed and that all of the classes has the same covariance matrix. If we do this assumption, we would classify a new observation into the discriminant function that is the largest.

We use the rule of classifying the observation to g if $\delta_g(x) - \delta_f(x) > 0$. Which can be written as

$$X_o^T \hat{\Sigma}^{-1}(\mu_g - \mu_f) - 1/2\mu_g^T \hat{\Sigma}^{-1}\mu_g + 1/2\mu_f^T \hat{\Sigma}^{-1}\mu_f + (\log \pi_g - \log \pi_f) > 0$$

Problem 3

a) On average, what fraction of people with an odds of 0.37 of defaulting on their credit card payment will in fact default?

$$\frac{p}{1-p} = 0.37 \to p = 0.37/1.37 = 0.270$$

So about 27% of people with an odds of 0.37 of defaulting on their credit card will in fact default.

b) Suppose that an individual has a 16% chance of defaulting on her credit card payment. What are the odds that she will default?

$$\frac{0.16}{1-0.16} = 0.19$$

Problem 4

Suppose we collect data for a group of students in a statistics class with variables x_1 = hours studied, x_2 = GPA, and Y = recieve an A. We fit a logistic regression and produce the estimated coefficients $\hat{\beta}_0 = -6$, $\hat{\beta}_1 = 0.05$, $\hat{\beta}_2 = 1$.

a) Estimate the probability that a student who studies for 40h and has an GPA of 3.5 gets an A.

$$P(Y = A, X_1 = 40, X_2 = 3.5) = \frac{e^{-6+0.05\cdot 40+1\cdot 3.5}}{1+e^{-6+0.05\cdot 40+1\cdot 3.5}} = 0.378$$

b) How many hours would the student in a) need to study to have a 50% probability of getting an A in the class?

$$\frac{e^{-6+0.05\cdot x+1\cdot 3.5}}{1+e^{-6+0.05\cdot x+1\cdot 3.5}} = 0.5 \rightarrow 0.05x = 2.5 \rightarrow x = 50$$

Problem 5 We have a two-class problem, with classes 0 = non-disease and 1 = disease, and a method $p(x)$ that produces probability of disease for a covariate x. In a population we have investigated N individuals and know the predicted probability of disease $p(x)$ and true disease status for these N.

a) We choose the rule $p(x) > 0.5$ to classify disease. Define the sensitivity and the specificity of the test.

b) Explain how you can construct a reciever operator curve (ROC) for your setting, and why that is a useful thing to do. In particular, why do we want to investigate different cut-offs of the probability of disease?

c) Assume that we have a competing method $q(x)$ that also produces probability of disease for a covariate x. We get the information that the AUC of the $p(x)$ method is 0.6 and the AUC of the $q(x)$ is 0.7. What is

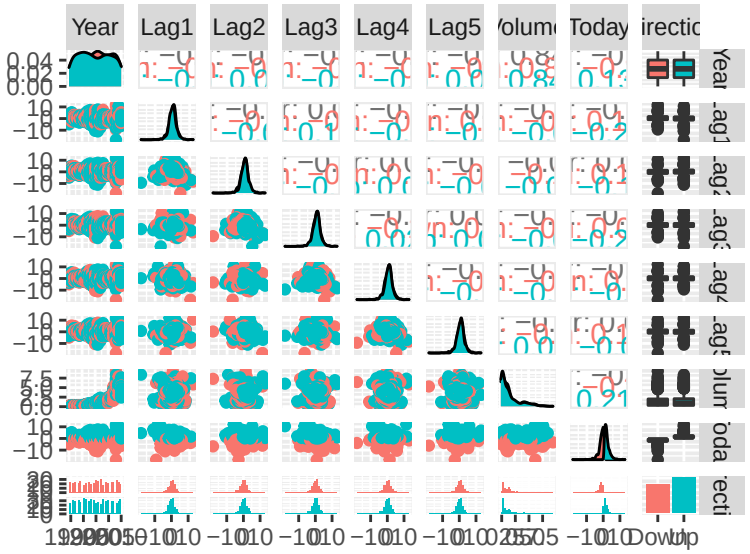
Problem 6

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

```
library(ISLR)
data("Weekly")

library(GGally)


ggpairs(Weekly, aes(color=Direction))
```



There seems so be a huge correlation between year and volume.

b) Use the full dataset to perform a logistic regression with Direction as response and the five lag variables plus Volume as predictors. Use the summary() function to print the results. Which of these predictors appeart to be of interest?

```
adj_weekly = Weekly[,-c(1,8)]

model = glm(as.factor(Direction) ~ ., data=adj_weekly, family="binomial")

summary(model)
```

```
##
## Call:
## glm(formula = as.factor(Direction) ~ ., family = "binomial",
##     data = adj_weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

We see that the lag2 variable has the lowest p-value, and this variable might be related to the direction response. However, it is not especially small (0.0296) and it might therefore just be small by chance.

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.

```
glm.probs = predict(model, type="response")
glm.preds = ifelse(glm.probs > 0.5, "Up","Down")
table(glm.preds, adj_weekly$Direction)
```

```
##
```

```
## glm.preds Down  Up
##      Down   54  48
##      Up    430 557
```

We see that the model performs well on putting the up values as up values, but performs bad at doing the same for down. It has a tendency to predict up rather than down. The overall performance would be:

```
perf = (54 + 557)/(54 + 557 + 430 + 48)
perf
```

```
## [1] 0.5610652
```

It has a 56,1% chance of predicting correct. The model is good at predicting when the market goes up, but bad at predicting when it goes down.

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.

```
#Dividing into training and testing data
train_id = (Weekly$Year < 2009)
train = Weekly[train_id,]
test = Weekly[!train_id,]

#Choosing only Direction and lag2 as variables
reduced_model = glm(Direction ~ Lag2,family = "binomial", data = train)

#Predicting based on the testing data
pred = predict(reduced_model, newdata = test, type = "response")

glm.preds = ifelse(pred > 0.5, "Up","Down")
table(glm.preds, test$Direction)
```

```
##
## glm.preds Down Up
##      Down    9  5
##      Up     34 56
```

Lets check the accuracy:

```
acc = (9+56)/(9 + 56 + 5 + 34)
acc
```

```
## [1] 0.625
```

We see that the performance is better.

e) Repeat d) using Linear discriminant analysis (LDA)

```
library(MASS)

model = lda(Direction ~ Lag2,data = train)
pred = predict(model, newdata = test, type="response")$class  #We have to choose class to get numeric


table(pred, test$Direction)
```

```
##
## pred   Down Up
##   Down    9  5
##   Up     34 56
```

f) Repeat d) using QDA

```
model = qda(Direction ~ Lag2,data = train)
lda.pred = predict(model, newdata = test, type="response")$class  #We have to choose class to get numer


table(lda.pred, test$Direction)
```

```
##
## lda.pred Down Up
##     Down    0  0
##     Up     43 61
```

This QDA classifier totally fails at predicting the Down trends, it only predicts "up" and is therefore a bad model for our problem.

g) Repeat d) using KNN with K = 1

```
#install.packages("class")
library(class)

K = 1

knn.train = as.matrix(train$Lag2)
knn.test = as.matrix(test$Lag2)

set.seed(123)

KNN_model = knn(train=knn.train, test = knn.test, cl = train$Direction, k = K, prob = T) #prob = T will
table(KNN_model, test$Direction)
```

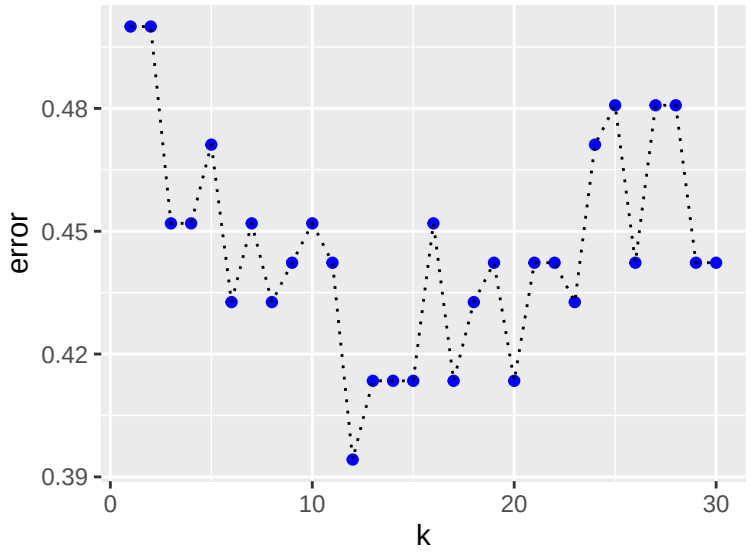
```
##
## KNN_model Down Up
##      Down   21 29
##      Up     22 32
```

h) Find the best value of K. Report the confusion matrix and overall fraction of correct predictions for this value of K.

```
K = 30
knn.error = rep(NA,K)

set.seed(234)
for(k in 1:K){
  knn.pred = knn(train = knn.train, tes = knn.test, cl = train$Direction, k = k)
  knn.error[k] = mean(knn.pred != test$Direction)
}

knn.error.df = data.frame(k=1:K, error = knn.error)
ggplot(knn.error.df, aes(x=k, y=error)) + geom_point(col = "blue") + geom_line(linetype = "dotted")
```



K = 12 seems to give the lowest error.

```
K = 12

KNN_model = knn(train=knn.train, test = knn.test, cl = train$Direction, k = K, prob = T)
table(KNN_model, test$Direction)
```

```
##
## KNN_model Down Up
##      Down   19 18
##      Up     24 43
```

We find the fraction of correct predictions.

```
acc = (19+43)/(19 + 43 + 18 + 24)
acc
```

```
## [1] 0.5961538
```

i) Which of these methods appear to provide the best results on this data?

The logistic regression with training and testing sets has the highest prediction accuracy.

j) Plot the ROC curves and calculate the UAC for the four methods. What can you say about the fit of these models?

```
KNNprobs = attributes(KNN_model)$prob

down = which(KNN_model == "Down")
KNNprobs[down] = 1 - KNNprobs[down]

library(pROC)
library(plotROC)

KNNroc = roc(response = as.numeric(test$Direction), predictor = KNNprobs, direction = "<")

#dat = data.frame(Direction = test$Direction, knn = KNNprobs, glm = glm.probs)
#dat_long = melt_roc(dat, "Direction", c("knn", "glm"))
 #NOT done
```