Module 4: Recommended Exercises

Statistical Learning V2021

alexaoh

01.02.2021

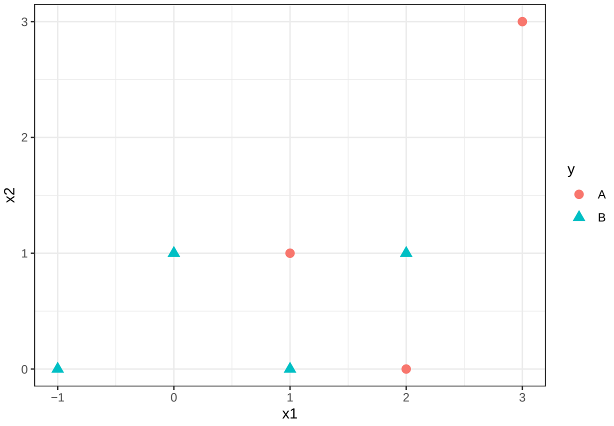
Problem 1: KNN (Exercise 2.4.7 in ISL textbook slightly modified)

The table and plot below provides a training data set consisting of seven observations, two predictors and one qualitative response variable.

```r
library(kableExtra)
knnframe = data.frame(x1 = c(3, 2, 1, 0, -1, 2, 1), x2 = c(3, 0, 1, 1,
    0, 1, 0), y = as.factor(c("A", "A", "A", "B", "B", "B", "B")))
print(knnframe)
```

```
#>   x1 x2 y
#> 1  3  3 A
#> 2  2  0 A
#> 3  1  1 A
#> 4  0  1 B
#> 5 -1  0 B
#> 6  2  1 B
#> 7  1  0 B
```

```r
library(ggplot2)
ggplot(knnframe, aes(x = x1, y = x2)) + geom_point(aes(color = y, shape = y),
    size = 3) + theme_bw()
```



We wish to use this data set to make a prediction for Y when $X_1 = 1, X_2 = 2$ using the K-nearest neighbors classification method.

a)

Calculate the Euclidean distance between each observation and the test point, $X_1 = 1, X_2 = 2$.

The Euclidean distances between each of the observations and the test point $(1, 2)$ are

```r
# Quick and dirty solution.
for (i in 1:7) {
    cat("Distance from point ", i, " ")
    cat(sqrt((1 - knnframe[i, 1])^2 + (2 - knnframe[i, 2])^2))
    cat("\n")
}
```

```
#> Distance from point  1  2.236068
#> Distance from point  2  2.236068
#> Distance from point  3  1
#> Distance from point  4  1.414214
#> Distance from point  5  2.828427
#> Distance from point  6  1.414214
#> Distance from point  7  2
```

b)

Use $P(Y = j \mid X = x_0) = \frac{1}{K} \sum_{I \in \mathcal{N}_0} I(Y = j)$ to predict the class of Y when $K = 1$, $K = 4$ and $K = 7$. Why is $K = 7$ a bad choice?

Without writing the calculations here, the predictions are as follows.

K = 1 When K = 1 the closest point to the test point is the third observation. Since this observation belongs to class A, $(1, 2)$ will also be classified as class A.

K = 4 When K = 4 the closest points are: 3, 4, 6 and 7. The probability of class A among these is $1/4$ and the probability of class B is $3/4$. Hence, $(1, 2)$ is classified as class B.

K = 7 When K = 7, all the points are taken into consideration. In this case the probability of class A is $3/7$ and the probability of class B is $4/7$, which means that $(1, 2)$ is classified as class B.

K = 7 is a bad choice because all the observations we have to our disposal are used at once. This is a bad tactic because the predicted value will be the same (class B) no matter which point on the board we want to classify. This will not give any more predictive or useful information than the distribution of all the point.

c)

If the Bayes decision boundary in this problem is highly non-linear, would we expect the best value for K to be large or small? Why?

When the Bayes decision boundary is highly non-linear we would expect the best value for K to be small, because in this case the method has high variance and low bias, which matches the non-linearity. When K is increased, the boundary becomes more linear, because the bias increases and the variance decreases.

Problem 2: Bank notes and LDA (with calculations)

To distinguish between genuine and fake bank notes measurements of length and diagonal of an image part of the bank notes have been made. For 1000 bank notes (500 genuine and 500 false) this gave the following values for the mean and the covariance matrix (using unbiased estimators), where the first value is the length of the bank note.

Genuine bank notes:

$$\bar{\mathbf{x}}_G = \begin{bmatrix} 214.97 \\ 141.52 \end{bmatrix} \text{ and } \hat{\boldsymbol{\Sigma}}_G = \begin{bmatrix} 0.1502 & 0.0055 \\ 0.0055 & 0.1998 \end{bmatrix}$$

Fake bank notes:

$$\bar{\mathbf{x}}_F = \begin{bmatrix} 214.82 \\ 139.45 \end{bmatrix} \text{ and } \hat{\boldsymbol{\Sigma}}_F = \begin{bmatrix} 0.1240 & 0.0116 \\ 0.0116 & 0.3112 \end{bmatrix}$$

a)

Assume the true covariance matrix for the genuine and fake bank notes are the same. How would you estimate the common covariance matrix?

I would use the pooled variance estimator, given by the covariance matrix for each class and the pooled estimator, as shown below.

- The covariance matrices for each class:

$$\hat{\boldsymbol{\Sigma}}_k = \frac{1}{n_k - 1} \sum_{i:y_i=k} (\mathbf{X}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{X}_i - \hat{\boldsymbol{\mu}}_k)^T$$

- Pooled version:

$$\hat{\boldsymbol{\Sigma}} = \sum_{k=1}^{K} \frac{n_k - 1}{n - K} \cdot \hat{\boldsymbol{\Sigma}}_k.$$

However, since the estimated covariance matrices for each class are given here, the first point is not needed. These estimations are used to calculate the pooled covariance in the second point. More specifically, the expression in this case turns out being

$$\hat{\boldsymbol{\Sigma}} = \frac{499}{998} \cdot \hat{\boldsymbol{\Sigma}}_G + \frac{499}{998} \cdot \hat{\boldsymbol{\Sigma}}_F = \frac{1}{2}(\hat{\boldsymbol{\Sigma}}_G + \hat{\boldsymbol{\Sigma}}_F) = \frac{1}{2}(\begin{bmatrix} 0.1502 & 0.0055 \\ 0.0055 & 0.1998 \end{bmatrix} + \begin{bmatrix} 0.1240 & 0.0116 \\ 0.0116 & 0.3112 \end{bmatrix}) = \begin{bmatrix} 0.13710 & 0.00855 \\ 0.00855 & 0.25550 \end{bmatrix}.$$

b)

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

We need to assume that the density functions of $\boldsymbol{X}$ of observations coming from each of the K classes (i.e. class conditional distributions) are normally distributed. In this case we will assume that observations from each of the two classes have bivariate normal distributions. Moreover, to use LDA we assume that the means may differ but the covariance matrices are assumed to being equal in all classes. The classification rule, or more specifically the classification boundary, is calculated in the following, by forcing equality of the discriminant functions of each of the classes

$$\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_G - \frac{1}{2} \boldsymbol{\mu}_G^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_G + \log \pi_G = \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_F - \frac{1}{2} \boldsymbol{\mu}_F^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_F + \log \pi_F,$$

$$\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_G - \boldsymbol{\mu}_F) - \frac{1}{2} \boldsymbol{\mu}_G^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_G + \frac{1}{2} \boldsymbol{\mu}_F^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_F + \log \pi_G - \log \pi_F = 0.$$

The linear function in $\boldsymbol{x}$ can be obtained by inserting the estimations from earlier, joined by an estimate of the prior probabilities.

c)

Use the method in b) to determine if a bank note with length 214.0 and diagonal 140.4 is genuine or fake. You can use R to perform the matrix calculations.

```
x <- matrix(c(214, 140.4), ncol = 1)

# Estimations of prior probabilities.
prior.G <- 0.5
prior.F <- 0.5

x.G <- matrix(c(214.97, 141.52), ncol = 1)
x.F <- matrix(c(214.82, 139.45), ncol = 1)
Sigma.G <- matrix(c(0.1502, 0.0055, 0.0055, 0.1998), nrow = 2)
Sigma.F <- matrix(c(0.124, 0.0116, 0.0116, 0.3112), nrow = 2)

Sigma <- 0.5 * (Sigma.G + Sigma.F)
Sigma.inv <- solve(Sigma)
t(x) %*% Sigma.inv %*% (x.G - x.F) - 0.5 * t(x.G) %*% Sigma.inv %*% x.G +
    0.5 * t(x.F) %*% Sigma.inv %*% x.F + log(prior.G) - log(prior.F)
```

```
#>           [,1]
#> [1,] -1.215085
```

Now, since the calculated value is negative, it means that, given the values $(214.0, 140.4)^T$, the probability is higher for it being fake. Hence, the note is classified as fake!

d)

What is the difference between LDA and QDA? Use the classification rule for QDA to determine the bank note from c). Do you obtain the same result? You can use R to perform the matrix calculations.

The difference between LDA and QDA is that in QDA the covariances of each of the different classes are allowed to be different. This makes QDA more flexible than LDA, but we still assume Gaussian distributions with different means each class. The discrimnant functions are now given by

$$\begin{aligned}\delta_k(\boldsymbol{x}) &= -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log\pi_k \\ &= -\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log\pi_k.\end{aligned}$$

Using the same procedure as in c) (subtracting the discriminant function of false from true) gives the following answer

```
delta.G <- -0.5 * t(x) %*% solve(Sigma.G) %*% x + t(x) %*% solve(Sigma.G) %*%
    x.G - 0.5 * t(x.G) %*% solve(Sigma.G) %*% x.G - 0.5 * log(det(Sigma.G)) +
    log(prior.G)
delta.F <- -0.5 * t(x) %*% solve(Sigma.F) %*% x + t(x) %*% solve(Sigma.F) %*%
    x.F - 0.5 * t(x.F) %*% solve(Sigma.F) %*% x.F - 0.5 * log(det(Sigma.F)) +
    log(prior.F)
delta.G - delta.F
```

```
#>           [,1]
#> [1,] -1.542967
```

It is apparent that the conclusion does not change; the note should still be classified as false.

Problem 3: Odds (Exercise 4.7.9 in ISL textbook)

This problem has to do with *odds*.

a)

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

An odds of 0.37 corresponds to a defaulting probability of $\frac{0.37}{1+0.37} = \frac{37}{137} \approx 0.27$. This means that the fraction of people, on average, which will default is approximately 0.27.

b)

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

A probability of 16% corresponds to odds of $\frac{0.16}{1-0.16} = \frac{4}{21} \approx 0.19$.

Problem 4: Logistic regression (Exercise 4.7.6 in ISL textbook)

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

a)

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

The estimation is

$$\hat{p} = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2)} = \frac{\exp(-6 + 0.05 \cdot 40 + 1 \cdot 3.5)}{1 + \exp(-6 + 0.05 \cdot 40 + 1 \cdot 3.5)} \approx 0.3775.$$

b)

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

Using some algebra, the estimated hours are

$$\hat{x}_1 = \frac{1}{\hat{\beta}_1}(\log\left(\frac{p}{1-p}\right) - \hat{\beta}_0 - \hat{\beta}_2 x_2) = \frac{1}{0.05}(\log\left(\frac{0.5}{1-0.5}\right) - (-6) - 1 \cdot 3.5) = 50 \text{ hours}$$

Problem 5: Sensitivity, specificity, ROC and AUC

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 to disease. Define the sensitivity and the specificity of the test.

In the following we will denote a 'positive' as having tested positively for the disease(=1).

The sensitivity of the test is the proportion of correctly classified positive observations. More precisely, the number of true positives (number of predicted diseased that are actually diseased) divided by the actual positives in the population. This the same as (1-Type2(false negative)).

The specificity of the test is the proportion of correctly classified negative observations. More precisely, the number of true negatives (number of predicted non-diseased that are actually non-diseased) divided by the actual negatives in the population. This is the same as (1-Type1(false positive)).

b)

Explain how you can construct a receiver 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?

A ROC curve can be constructed by plotting sensitivity against (1-specificity) for all possible values of the probability-threshold for classification. This is useful since then we can get a visual impression of which threshold gives the best values for the sensitivity and the power of the test. We want to investigate this because perhaps we want to find the cut-off that maximizes both the sensitivity and the specificity at once. This depends on the use-case of course, because sometimes one might want prioritize one over the other.

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 $q(x)$-method is 0.7. What is the definition and interpretation of the AUC? Would you prefer the $p(x)$ or the $q(x)$ method for classification?

The AUC is the area under the ROC curve. Hence, it gives the overall performance of the test for all possible thresholds. A higher area (1 is max = means that the test has a perfect fit for all thresholds) means that the specificity and the sensitivity are collectively smaller than when the area is lower. I would prefer q for classification in this case, since the AUC is higher.

Data analysis with R

For the following problems, you should check out and learn how to use the following R functions: `glm()` (`stats` library), `lda()`, `qda()` (`MASS` library), `knn()` (`class` library), `roc()` and `auc()` (`pROC` library).

Problem 6 (Exercise 4.7.10 in ISL textbook - modified)

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 numerical and graphical summaries of the `Weekly` data. Do there appear to be any patterns?

```r
library(ISLR)
library(GGally)
```

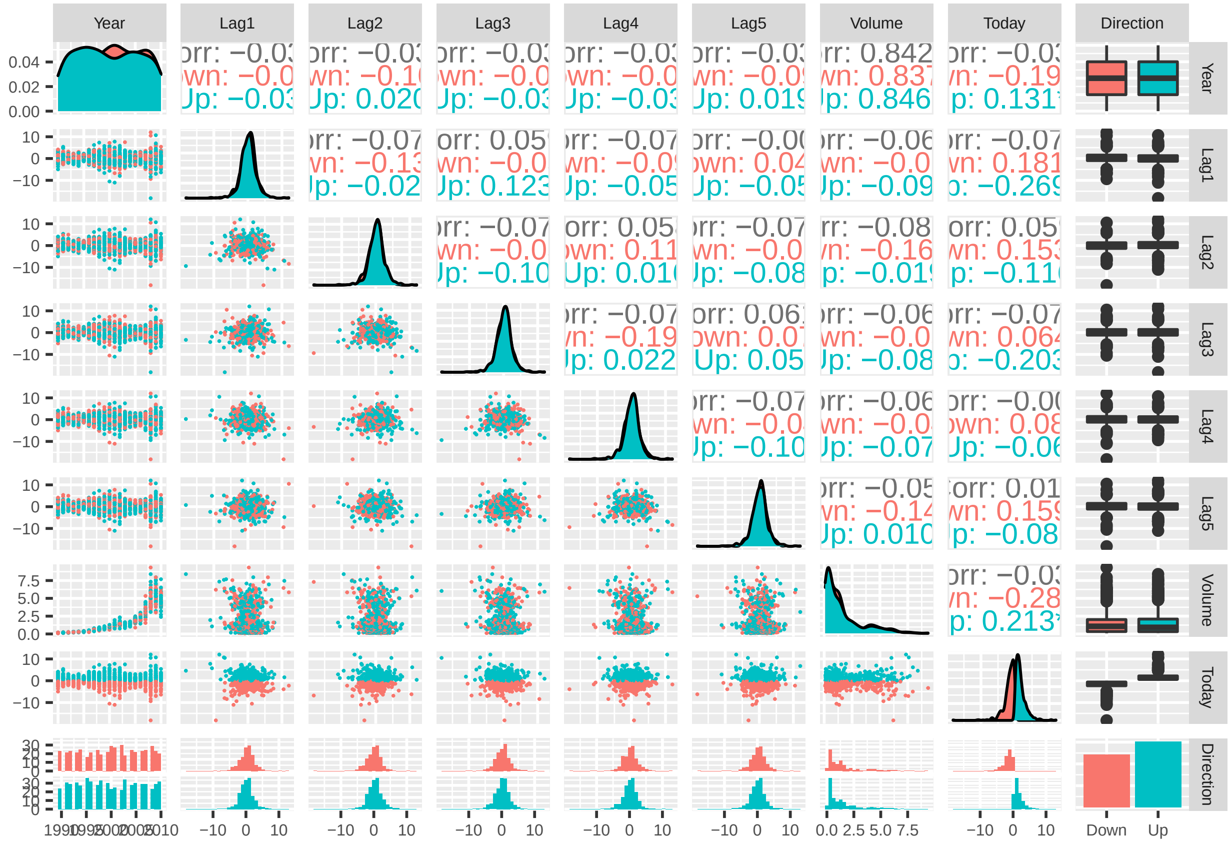
```r
data("Weekly")
summary(Weekly)
```

```
#>       Year           Lag1               Lag2               Lag3        
#>  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950  
#>  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580  
#>  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410  
#>  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472  
#>  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090  
#>  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260  
#>       Lag4               Lag5              Volume            Today         
#>  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950  
#>  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540  
#>  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410  
#>  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499  
#>  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050  
#>  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260  
#>  Direction 
#>  Down:484  
#>  Up  :605  
#>            
#>            
#>            
#>            
```

```r
cor(Weekly[, -9])
```

```
#>               Year         Lag1        Lag2        Lag3         Lag4
#> Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
#> Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
#> Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
#> Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
#> Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
#> Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
#> Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
#> Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
#>                Lag5      Volume         Today
#> Year   -0.030519101  0.84194162 -0.032459894
#> Lag1   -0.008183096 -0.06495131 -0.075031842
#> Lag2   -0.072499482 -0.08551314  0.059166717
#> Lag3    0.060657175 -0.06928771 -0.071243639
#> Lag4   -0.075675027 -0.06107462 -0.007825873
#> Lag5    1.000000000 -0.05851741  0.011012698
#> Volume -0.058517414  1.00000000 -0.033077783
#> Today   0.011012698 -0.03307778  1.000000000
```

```r
ggpairs(Weekly, aes(color = Direction), lower = list(continuous = wrap("points",
    size = 0.01))) + theme(text = element_text(size = 7))
```



All correlations in the correlation matrix are pretty small, except for the correlation between `Year` and `Volume`, which is ≈ 0.842. This means that they are highly correlated. The same is seen from the pairs-plot (including the correlations from the figure). In the pairs-plot between `Year` and `Volume` it looks almost like the year increases quadratically with the volume. Hence, there appears to be a pattern of positive correlation between the volume of shares traded and the year that the observation was recorded.

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. Which of these predictors appear to be of interest?

```
glm.fit <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
    data = Weekly, family = binomial)
summary(glm.fit)
```

```
#>
#> Call:
#> glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
#>     Volume, family = binomial, data = 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
```

I would say that there is no clear evidence that any of the predictors have a real association with `Direction`, but the smallest p-value among these is associated with `Lag2`. `Lag1` could perhaps also be of some interest.

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.

```r
glm.probs_Weekly = predict(glm.fit, type = "response")
glm.preds_Weekly = ifelse(glm.probs_Weekly > 0.5, "Up", "Down")
table(predicted = glm.preds_Weekly, true = Weekly$Direction)
```

```
#>          true
#> predicted Down  Up
#>      Down   54  48
#>      Up    430 557
```

```r
mean(glm.preds_Weekly == Weekly$Direction)  # Calculate fraction of correct predictions.
```

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

From the confusion matrix, it can be calculated that the overall fraction of correct predictions is

$$\frac{557+54}{430+557+48+54} = \frac{611}{1089} \approx 0.561.$$

The confusion matrix is telling us the following about the types of mistakes made by the logistic regression

- Sensitivity: $\frac{557}{48+557} \approx 0.921$
- Specificity: $\frac{54}{54+430} \frac{27}{242} \approx 0.112$

It is clear that the test is a lot better at predicting when the market goes `Up` vs 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 (that is, the data from 2009 and 2010).
R-hints: use the following code to divide into test and train set. For predicting the direction of the test set, use `newdata = Weekly_test` in the `predict()` function.

```
Weekly_trainID = (Weekly$Year < 2009)
Weekly_train = Weekly[Weekly_trainID, ]
Weekly_test = Weekly[!Weekly_trainID, ]

glm.fit2 <- glm(Direction ~ Lag2, data = Weekly_train, family = binomial)

glm.probs_Weekly2 = predict(glm.fit2, newdata = Weekly_test, type = "response")
glm.preds_Weekly2 = ifelse(glm.probs_Weekly2 > 0.5, "Up", "Down")
table(predicted = glm.preds_Weekly2, true = Weekly_test$Direction)
```

```
#>          true
#> predicted Down Up
#>      Down    9  5
#>      Up     34 56
```

The overall fraction of correct predictions is

$$\frac{56+9}{56+9+5+34} = \frac{5}{8} = 0.625.$$

e)

Repeat d) using LDA.

```
library(MASS)
lda.Weekly <- lda(Direction ~ Lag2, data = Weekly_train)
lda.Weekly_pred <- predict(lda.Weekly, newdata = Weekly_test)$class
lda.Weekly_prob <- predict(lda.Weekly, newdata = Weekly_test)$posterior
table(predicted = lda.Weekly_pred, true = Weekly_test$Direction)
```

```
#>          true
#> predicted Down Up
#>      Down    9  5
#>      Up     34 56
```

The overall fraction of correct predictions is

$$\frac{56+9}{56+9+5+34} = \frac{5}{8} = 0.625,$$

which is the same as for logistic regression (since this is a binary classification problem with one covariate).

f)

Repeat d) using QDA.

```
qda.Weekly <- qda(Direction ~ Lag2, data = Weekly_train)
qda.Weekly_pred <- predict(qda.Weekly, newdata = Weekly_test)$class
qda.Weekly_prob <- predict(qda.Weekly, newdata = Weekly_test)$posterior
table(predicted = qda.Weekly_pred, true = Weekly_test$Direction)
```

```
#>          true
#> predicted Down Up
#>      Down    0  0
#>      Up     43 61
```

The overall fraction of correct predictions is

$$\frac{61}{61+43} = \frac{61}{104} \approx 0.587.$$

g)

Repeat d) using KNN with $K = 1$.

R-hints: plug in your variables in the following code to perform KNN. The argument `prob=T` will provide the probabilities for the classified direction (which you will need later). When there are ties (same amount of Up and Down for the nearest neighbors), the `knn` function picks a class at random. We use the `set.seed()` function such that we don't get different answers for each time we run the code.

```r
library(class)
knn.train <- as.matrix(Weekly_train$Lag2)
knn.test <- as.matrix(Weekly_test$Lag2)
set.seed(123)
knn.model <- knn(train = knn.train, test = knn.test, cl = Weekly_train$Direction,
    k = 1, prob = T)
table(predicted = knn.model, true = Weekly_test$Direction)
```

```
#>          true
#> predicted Down Up
#>      Down   21 29
#>      Up     22 32
```

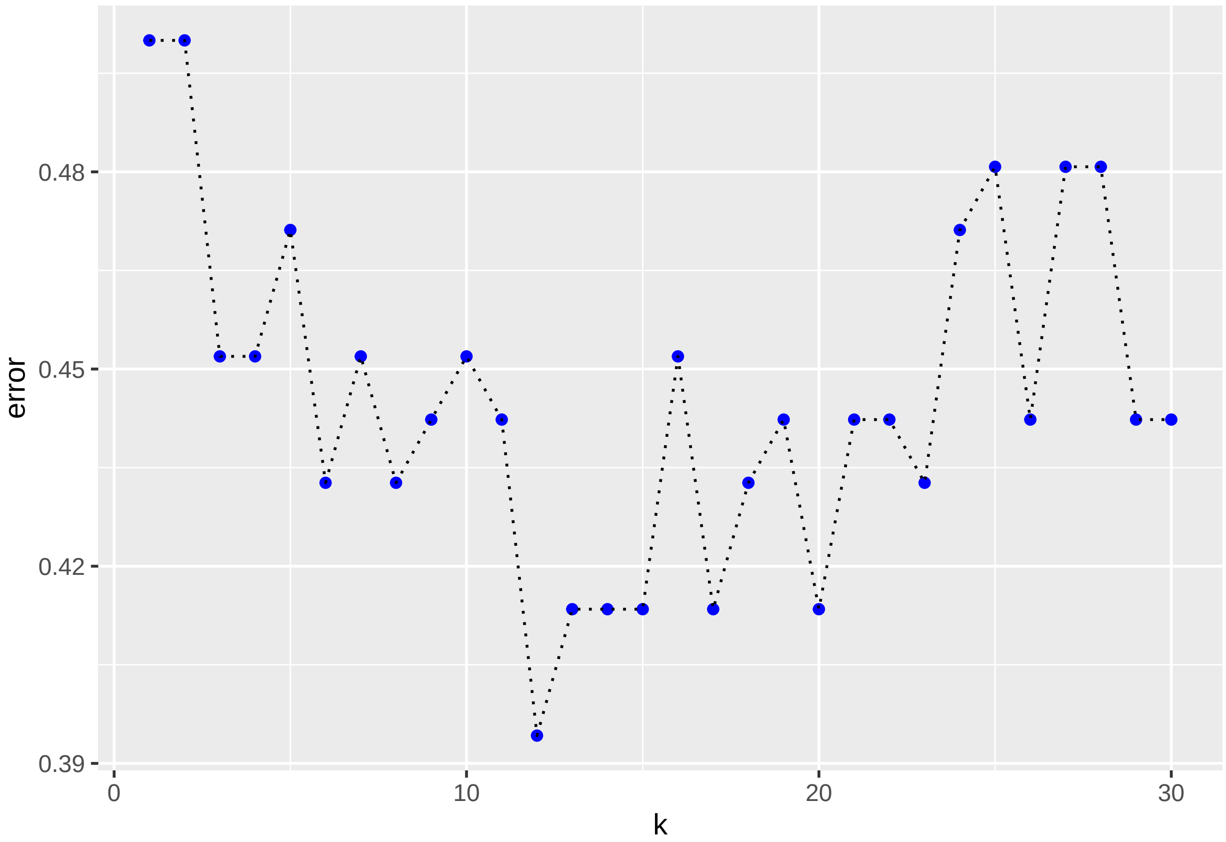
The overall fraction of correct predictions is

$$\frac{21+32}{21+32+29+22} = \frac{53}{104} \approx 0.510.$$

h)

Use the following code to find the best value of K. Report the confusion matrix and overall fraction of correct predictions for this value of K.

```r
# knn error:
K = 30
knn.error = rep(NA, K)
set.seed(234)
for (k in 1:K) {
    knn.pred = knn(train = knn.train, test = knn.test, cl = Weekly_train$Direction,
        k = k)
    knn.error[k] = mean(knn.pred != Weekly_test$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")
```



The plot shows that $K = 12$ is the best value of K.

The confusion matrix for $K = 12$ is

```
knn12 <- knn(train = knn.train, test = knn.test, cl = Weekly_train$Direction,
    k = 12, prob = T)
table(predicted = knn12, true = Weekly_test$Direction)
```

```
#>          true
#> predicted Down Up
#>      Down   19 18
#>      Up     24 43
```

```
mean(knn.pred == Weekly_test$Direction)
```

```
#> [1] 0.5576923
```

i)

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

It looks like logistic regression/LDA, which are the same in this case, give the best results on this data.

j)

Plot the ROC curves and calculate the AUC for the four methods (using your the best choice for KNN). What can you say about the fit of these models?

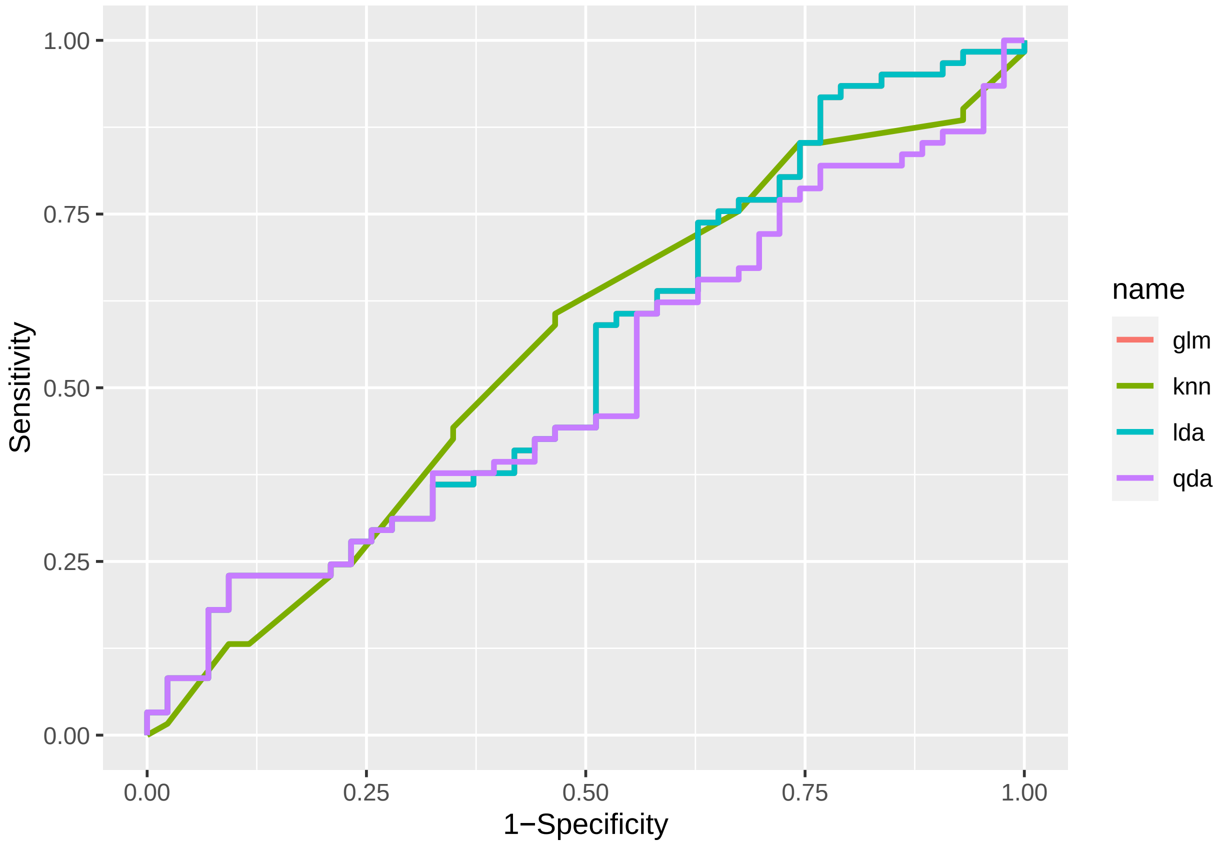
R-hints:

- For KNN you can use `knn(...,prob=TRUE)` to get the probability for the classified direction. Note that we want $P(Direction = Up)$ when plotting the ROC-curve, so we need to modify the probabilities returned from the `knn` function.

```
# get the probabilities for the classified class
knn12.probs = attributes(knn12)$prob
# since we want the probability for Up, we need to take 1-p for the
# elements that gives probability for Down
down = which(knn12 == "Down")
knn12.probs[down] = 1 - knn12.probs[down]
```

- Use the following code to produce ROC-curves:

```
library(pROC)
library(plotROC)
glmroc <- roc(response = Weekly_test$Direction, predictor = glm.probs_Weekly2,
    direction = "<")
ldaroc <- roc(response = Weekly_test$Direction, predictor = lda.Weekly_prob[,
    2], direction = "<")
qdaroc <- roc(response = Weekly_test$Direction, predictor = qda.Weekly_prob[,
    2], direction = "<")
knnroc <- roc(response = Weekly_test$Direction, predictor = knn12.probs,
    direction = "<")
dat <- data.frame(Direction = Weekly_test$Direction, glm = glm.probs_Weekly2,
    lda = lda.Weekly_prob[, 2], qda = qda.Weekly_prob[, 2], knn = knn12.probs)
dat_long <- melt_roc(dat, "Direction", c("glm", "lda", "qda", "knn"))
ggplot(dat_long, aes(d = D, m = M, color = name)) + geom_roc(n.cuts = F) +
    xlab("1-Specificity") + ylab("Sensitivity")
```



```
# glm is very similar to lda, so the roc-curve for glm is not shown.

auc(glmroc)
```

```
#> Area under the curve: 0.5463
```

```
auc(ldaroc)
```

```
#> Area under the curve: 0.5463
```

```
auc(qdaroc)
```

```
#> Area under the curve: 0.5086
```

```
auc(knnroc)
```

```
#> Area under the curve: 0.5555
```

All these ROC are very close to the diagonal, which is in correspondence with the AUC being approximately 0.5. This means that none of these classifiers are any good, since they are very close to the "no information"-classifier.