Homework 4

Kunaal Raghav

Install and Library Packages

```r
#install.packages("pROC")
library(pROC)
```

```
Warning: package 'pROC' was built under R version 4.5.2

Type 'citation("pROC")' for a citation.


Attaching package: 'pROC'

The following objects are masked from 'package:stats':

    cov, smooth, var
```

```r
library(ISLR2)
```

```
Warning: package 'ISLR2' was built under R version 4.5.2
```

```r
library(MASS)
```

```

Attaching package: 'MASS'

The following object is masked from 'package:ISLR2':

    Boston
```

Stock Market Prediction (Exercise 4.8.13)

Question A

```r
data("Weekly")

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

```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5, family = binomial,
    data = Weekly)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.23029    0.06203   3.712 0.000205 ***
Lag1        -0.04010    0.02635  -1.522 0.128125
Lag2         0.06015    0.02674   2.249 0.024503 *
Lag3        -0.01508    0.02664  -0.566 0.571381
Lag4        -0.02677    0.02643  -1.013 0.311082
Lag5        -0.01349    0.02636  -0.512 0.608894
---
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: 1486.7  on 1083  degrees of freedom
AIC: 1498.7

Number of Fisher Scoring iterations: 4
```

Right now the only variable that seems to be statistically significant is Lag2, or the percentage returns for the previous two weeks.

Part B

```r
probs <- predict(Directions, type = "response")
preds <- ifelse(probs > 0.5, "Up", "Down")
table(preds, Weekly$Direction)
```

```
preds  Down  Up
  Down   49  41
  Up    435 564
```

```
mean(preds == Weekly$Direction)
```

```
[1] 0.5629017
```

On one hand, there are a lot of false negatives, meaning that predictions are unusually high for up/ raises in the markets, where as they actually come to be heavily in the downs/ negative.

Part C

```
train <- Weekly$Year <= 2008
test  <- Weekly$Year > 2008

fit_lag2 <- glm(Direction ~ Lag2, data = Weekly, family = binomial, subset = train)
probs_test <- predict(fit_lag2, newdata = Weekly[test, ], type = "response")
preds_test <- ifelse(probs_test > 0.5, "Up", "Down")

table(preds_test, Weekly$Direction[test])
```

```
preds_test Down Up
      Down    9  5
      Up     34 56
```

```
mean(preds_test == Weekly$Direction[test])
```

```
[1] 0.625
```

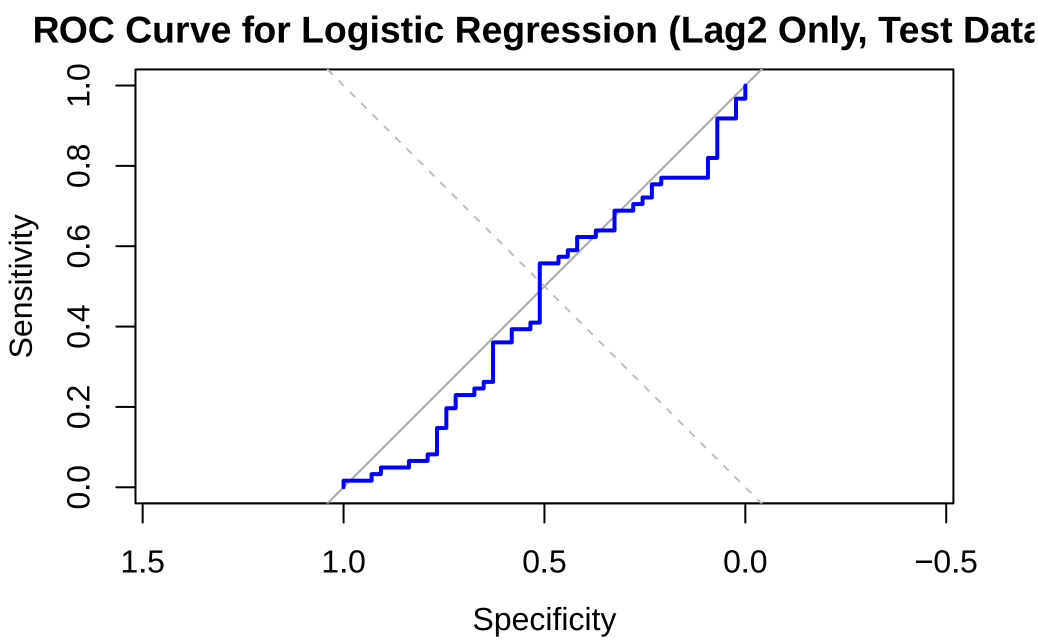
Considering that this model has a 62.5%, we can say that this model is more accurate. With the extra variables that were removed, its safe to say that there is a pattern of over fitting in the previous model

Part D

```r
# Train/test split
train <- Weekly$Year <= 2008
test  <- Weekly$Year > 2008
fit_lag2 <- glm(Direction ~ Lag2, data = Weekly,
                family = binomial, subset = train)
probs_test <- predict(fit_lag2, newdata = Weekly[test, ], type = "response")
actual_test <- Weekly$Direction[test]
roc_obj <- roc(actual_test, probs_test, levels = c("Down", "Up"))
```

```
Setting direction: controls > cases
```

```r
plot(roc_obj, col = "blue", lwd = 2,
     main = "ROC Curve for Logistic Regression (Lag2 Only, Test Data)")
abline(0, 1, lty = 2, col = "gray")
```



The model, while slightly better than a random prediction, which sits at .5, is not well established enough to properly predict down markets.

LDA vs QDA (Exercise 4.8.5)

Part A

On the training data set, since the model is more flexible, QDA is better used

On the test data set, since the model matches that classifer, LDA is better , as it will produce a lower test error

Part B

In the training set, QDA will have a lower error rate, because it will fit the model better than LDA, due to flexibility

With a nonlinear setting in the Bayes decision boundary, QDA will preform better due to low bias

Part C

With increases in the n size, test performance of QDA should improve, as QDA has high variance. Small n sizes means the variance hurts test performance

Part D

The aforementioned statement is wrong, as , the Bayes Classifier, in its Linear form, matches LDAS which has low bias and lower variance (as compared to QDA).

Non Uniform Prior (Exercise 4.8.7)

The following is the probability that we are setting up for calculation

$$P(Dividend|X = 4)$$

$$\mu_1 = 10$$

$$\mu_2 = 0$$

$$\sigma = 6$$

$$\sigma^2 = 36$$

Prior Probability issuing a dividend:

$$P(D) = .8$$

Prior Probability not issuing a dividend:

$$P(N) = .2$$

Observed Profit

$$X = 8$$

The first step is calculating for the following equations:

$$f_1(4) = f(X = 4|D)$$

$$f_1(4) = f(X = 4|N)$$

We use the density function to calculate. The density function is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}^{(-\frac{(x-\mu)^2}{2\sigma^2})}$$

For dividend companies, the filled in equation is as follows:

$$f_2(4) = \frac{1}{6\sqrt{2\pi}}^{(-\frac{(4-10)^2}{72})}$$

The transformed equation becomes:

$$f_1(4) = \frac{1}{6\sqrt{2\pi}}e^{-.5}$$

For non-dividend companies, the filled in equation is as follows

$$f_2(4) = \frac{1}{6\sqrt{2\pi}}^{(-\frac{(4-0)^2}{72})}$$

The transformed equation becomes:

$$f_2(4) = \frac{1}{6\sqrt{2\pi}}e^{-.22}$$

Then we plug in the Bayes Theorem:

$$P(D|X=4) = \frac{f_1(4)P(D)}{f_1(4)P(D) + f_2(4)P(N)}$$

The equationm, simplified, now becomes the following:

$$P(D|X=4) = \frac{.8(e^{-.5})}{(.8(e^{-.5}))(.2(e^{-.22})} = .752$$

Thus we can claim the following : The probability the company will issue a dividend given last year's profit was 4% is approximately 75%

If you want, I can also show you the exact R code that reproduces this result or help you generalize the formula for any value of X.

Stock Market Predictions Part 3 (Exercise 4.8.13)

Part A

```
data("Weekly")
train <- Weekly$Year <= 2008
test  <- Weekly$Year > 2008
lda_fit <- lda(Direction ~ Lag2, data = Weekly, subset = train)
lda_pred <- predict(lda_fit, Weekly[test, ])$class
table(lda_pred, Weekly$Direction[test])
```

```
lda_pred Down Up
    Down    9  5
    Up     34 56
```

```
mean(lda_pred == Weekly$Direction[test])
```

```
[1] 0.625
```

Part B

```r
qda_fit <- qda(Direction ~ Lag2, data = Weekly, subset = train)

qda_pred <- predict(qda_fit, Weekly[test, ])$class

table(qda_pred, Weekly$Direction[test])
```

```
qda_pred Down Up
    Down    0  0
    Up     43 61
```

```r
mean(qda_pred == Weekly$Direction[test])
```

```
[1] 0.5865385
```

Part C

With the provided accuracy measures, shown above, I would most likely suggest LDA, as it results in the highest accuracy, at 62%