DS-6030 Homework Module 3

Tom Lever

06/08/2023

DS 6030 | Spring 2023 | University of Virginia

5. We now examine the differences between LDA and QDA.

   (a) If the Bayes decision boundary is linear, do we expect LDA or QDA to perform better on the training set? On the test set?

   If the Bayes decision boundary is linear, we expect Quadratic Discriminant Analysis to perform better on the training set. According to https://online.stat.psu.edu/stat508/lesson/9/9.2/9.2.8, "QDA, because it allows for more flexibility for the covariance matrix, tends to fit the data better than LDA". We expect Linear Discriminant Analysis to perform better on the test set as the Bayes decision boundary is linear and QDA might overfit the data / follow errors too closely / yield a small training Mean Squared Error but a large test MSE / work too hard to find patterns in the training data and pick up some patterns that are just caused by random chance rather than by true properties of the function relating predictors and response.

   (b) If the Bayes decision boundary is non-linear, do we expect LDA or QDA to perform better on the training set? On the test set?

   If the Bayes decision boundary is non-linear, we expect QDA to perform better on the training set and test set "because it allows for more flexibility for the covariance matrix".

   (c) In general, as the sample size n increases, do we expect the test prediction accuracy of QDA relative to LDA to improve, decline, or be unchanged? Why?

   According to https://cseweb.ucsd.edu/classes/sp12/cse151-a/lecture11-final.pdf, "Variance depends on the training set size. It decreases with more training data, and increases with more complicated classifiers". As the sample size n increases, we expect the test prediction accuracy of QDA relative to LDA to improve as QDA is a more complicated, flexible model than LDA with less bias and more variance than LDA and the variance of QDA decreases as sample size increases.

   (d) True or False: Even if the Bayes decision boundary for a given problem is linear, we will probably achieve a superior test error rate using QDA rather than LDA because QDA is flexible enough to model a linear decision boundary. Justify your answer.

   False. As above, we expect Linear Discriminant Analysis to perform better on the test set when the Bayes decision boundary is linear as QDA might overfit the data.

6. This question should be answered using the `Weekly` data set, which is part of the `ISLR2` 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?

```
library(ISLR2)
head(x = Weekly, n = 3)
```

```
#   Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
# 1 1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
# 2 1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
# 3 1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
```

The columns of `Weekly` are

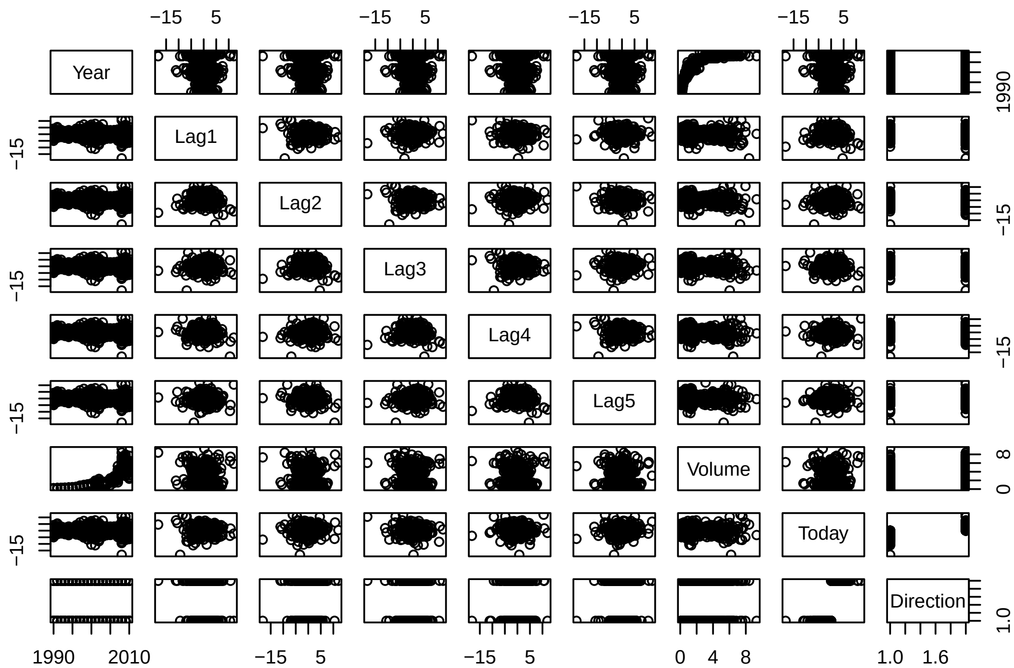
- $Year$: The year that the observation was recorded
- $Lag1$: Percentage returns 1 week previous
- $Lag2$: Percentage returns 2 weeks previous
- $Lag3$: Percentage returns 3 weeks previous
- $Lag4$: Percentage returns 4 weeks previous
- $Lag5$: Percentage returns 5 weeks previous
- $Volume$: Volume of stock market movement / volume of stock market activity / volume of shares traded / average number of daily shares traded in billions this week
- $Today$: Percentage return in the S&P500 this week
- $Direction$: Factor with levels `Down` and `Up` indicating whether the market had a positive or negative return on a given week

```
summary(Weekly)
```

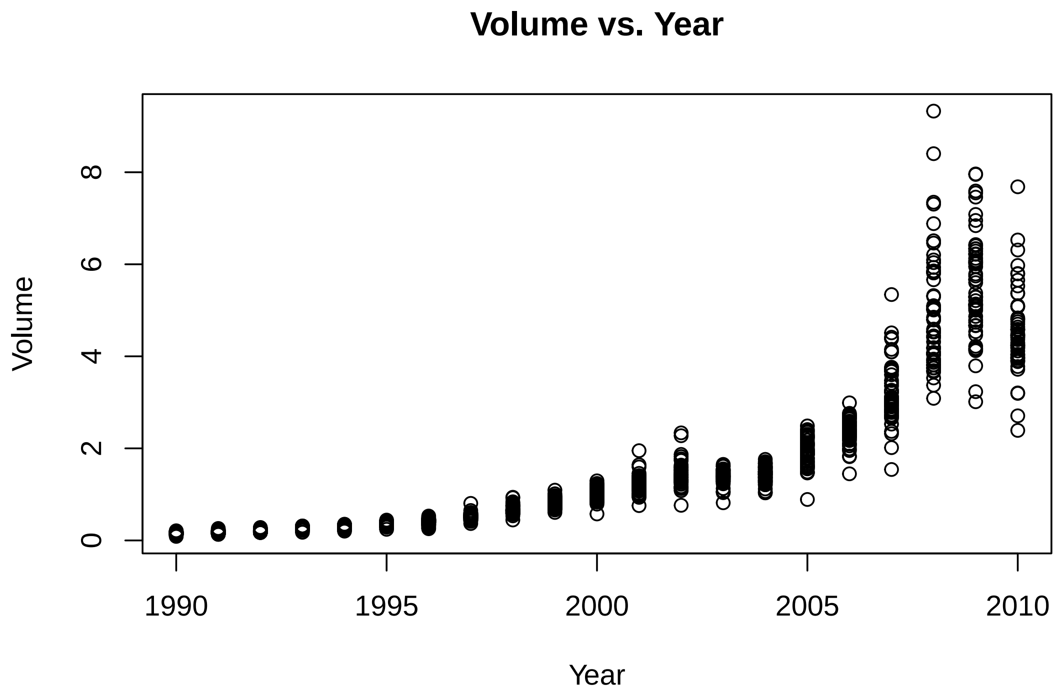
```
#       Year          Lag1               Lag2               Lag3
#  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
#  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
#  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
#  Mean   :2000   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
#
#
#
#
```

The years that an observation was recorded vary from 1990 to 2010. Minimum, first-quartile, median, mean, third-quartile, and maximum lags are similar across $Lag1$, $Lag2$, $Lag3$, $Lag4$, and $Lag5$ and $Today$. The market had a negative return for 484 weeks. The market had a positive return for 605 weeks. If we predicted that the market had a positive return for every one of the $1,089$ weeks, we would be correct $605/1,089 = 55.6$ percent of the time.

```
pairs(Weekly)
```




```r
plot(
    x = Weekly$Year,
    y = Weekly$Volume,
    xlab = "Year",
    ylab = "Volume",
    main = "Volume vs. Year"
)
```



$Volume$ seems to grow exponentially with $Year$.

```
library(TomLeversRPackage)
index_of_Direction <- get_index_of_column_of_data_frame(Weekly, "Direction")
data_frame_without_Direction <- Weekly[, -index_of_Direction]
correlation_matrix <- cor(data_frame_without_Direction)
analyze_correlation_matrix(correlation_matrix)
```

```
# Year
#     V+:  Year
#     V-:
#     H+:  Volume
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Lag1, Lag2, Lag3, Lag4, Lag5, Today
# Lag1
#     V+:  Lag1
#     V-:
#     H+:
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Year, Lag2, Lag3, Lag4, Lag5, Volume, Today
# Lag2
```

```
#     V+:  Lag2
#     V-:
#     H+:
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Year, Lag1, Lag3, Lag4, Lag5, Volume, Today
# Lag3
#     V+:  Lag3
#     V-:
#     H+:
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Year, Lag1, Lag2, Lag4, Lag5, Volume, Today
# Lag4
#     V+:  Lag4
#     V-:
#     H+:
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Year, Lag1, Lag2, Lag3, Lag5, Volume, Today
# Lag5
#     V+:  Lag5
#     V-:
#     H+:
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Year, Lag1, Lag2, Lag3, Lag4, Volume, Today
# Volume
#     V+:  Volume
#     V-:
#     H+:  Year
#     H-:
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Lag1, Lag2, Lag3, Lag4, Lag5, Today
# Today
#     V+:  Today
#     V-:
#     H+:
#     H-:
```

```
#     M+:
#     M-:
#     L+:
#     L-:
#     N:  Year, Lag1, Lag2, Lag3, Lag4, Lag5, Volume
```

$Volume$ has a high positive correlation with $Year$. All other pairs of variables have correlations that are negligible.

(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?

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

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

```
calculate_critical_value_zc(
    significance_level = 0.05,
    hypothesis_test_is_two_tailed = TRUE
)
```

```
# [1] 1.959964
```

A critical value $z_{\alpha/2=0.05/2} = 1.960$. The summary for the above logistic regression model provides test statistics for predictors. In parallel, the summary provides probabilities where each probability p is the probability that the magnitude $|z|$ of a random test statistic is greater than the magnitude $|z_0|$ of the appropriate test statistic. Because the magnitude of the test statistic for $Lag2$ is greater than the critical value, and the probability for this predictor is less than the significance level $\alpha = 0.05$, we reject the null hypothesis that $Lag2$ is insignificant in predicting the response in the context of the model and can be removed from the model. For $Lag2$ we have sufficient evidence to support the alternate hypothesis that the predictor is significant in predicting the response in the context of the model and cannot be removed from the model.

(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.

```
# vector_of_predicted_probabilities is a vector of predicted probabilities that
# an observation corresponds to the market having a positive return that week
vector_of_predicted_probabilities <- predict(
    object = logistic_regression_model,
    newdata = Weekly,
    type = "response"
)
map_of_binary_value_to_direction <- contrasts(x = Weekly$Direction)
map_of_binary_value_to_direction
```

```
#       Up
# Down  0
# Up    1
```

```
number_of_observations <- nrow(Weekly)
vector_of_predicted_directions <- rep("Down", number_of_observations)
condition <- vector_of_predicted_probabilities > 0.5
vector_of_predicted_directions[condition] = "Up"
confusion_matrix <- table(vector_of_predicted_directions, Weekly$Direction)
confusion_matrix
```

```
#
# vector_of_predicted_directions Down  Up
#                           Down   54  48
#                           Up    430 557
```

```
number_of_true_negatives <- confusion_matrix[1, 1]
number_of_false_negatives <- confusion_matrix[1, 2]
number_of_false_positives <- confusion_matrix[2, 1]
number_of_true_positives <- confusion_matrix[2, 2]
number_of_correct_predictions <-
    number_of_true_negatives + number_of_true_positives
fraction_of_correct_predictions <-
    number_of_correct_predictions / number_of_observations
fraction_of_correct_predictions
```

```
# [1] 0.5610652
```

The overall fraction of correct predictions is $611/1,089$. The confusion matrix is telling us that there are 48 false negatives and 430 false positives. A false negative is an instance of our logistic regression predicting that the market had a negative return on a week when the market had a positive return on that week. A false positive is an instance of our logistic regression predicting that the market had a positive return on a week when the market had a negative return on that week. The training error rate is 43.9 percent. For weeks when the market had a positive return, the model is correct 92.1 percent of the time / has a sensitivity, recall, hit rate, and True Positive Rate $TPR = \frac{TP}{P} = \frac{TP}{TP+FN} = \frac{557}{557+48} = 0.921$. For weeks when the market had a negative return, the model is correct 11.2 percent of the time / has a specificity, selectivity, and True Negative Rate $TNR = \frac{TN}{N} = \frac{TN}{TN+FP} = \frac{54}{54+430} = 0.112$.

(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).

```
condition <- (Weekly$Year >= 1990) & (Weekly$Year <= 2008)
Weekly_from_1990_to_2008_inclusive <- Weekly[condition, ]
condition <- (Weekly$Year > 2008) & (Weekly$Year <= 2010)
Weekly_from_2009_to_2010_inclusive <- Weekly[condition, ]
logistic_regression_model <- glm(
    formula = Direction ~ Lag2,
    data = Weekly_from_1990_to_2008_inclusive,
    family = binomial
)
# vector_of_predicted_probabilities is a vector of predicted probabilities that
# an observation corresponds to the market having a positive return that week
vector_of_predicted_probabilities <- predict(
    object = logistic_regression_model,
    newdata = Weekly_from_2009_to_2010_inclusive,
    type = "response"
)
map_of_binary_value_to_direction <-
    contrasts(x = Weekly_from_1990_to_2008_inclusive$Direction)
map_of_binary_value_to_direction
```

```
#      Up
# Down  0
# Up    1
```

```
number_of_observations <- nrow(Weekly_from_2009_to_2010_inclusive)
vector_of_predicted_directions <- rep("Down", number_of_observations)
condition <- vector_of_predicted_probabilities > 0.5
vector_of_predicted_directions[condition] = "Up"
confusion_matrix <- table(
    vector_of_predicted_directions,
    Weekly_from_2009_to_2010_inclusive$Direction
)
confusion_matrix
```

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

```
number_of_true_negatives <- confusion_matrix[1, 1]
number_of_false_negatives <- confusion_matrix[1, 2]
number_of_false_positives <- confusion_matrix[2, 1]
number_of_true_positives <- confusion_matrix[2, 2]
number_of_correct_predictions <-
    number_of_true_negatives + number_of_true_positives
fraction_of_correct_predictions <-
    number_of_correct_predictions / number_of_observations
fraction_of_correct_predictions
```

```
# [1] 0.625
```

The overall fraction of correct predictions is 65/104. The test error rate is 37.5 percent. For weeks when the market had a positive return, the model is correct 92.1 percent of the time / has a sensitivity, recall, hit rate, and True Positive Rate $TPR = \frac{TP}{P} = \frac{TP}{TP+FN} = \frac{56}{56+5} = 0.918$. For weeks when the market had a negative return, the model is correct 11.2 percent of the time / has a specificity, selectivity, and True Negative Rate $TNR = \frac{TN}{N} = \frac{TN}{TN+FP} = \frac{34}{34+9} = 0.209$.

(e) Repeat (d) using LDA.

```
library(MASS)
```

```
#
# Attaching package: 'MASS'

# The following object is masked from 'package:ISLR2':
#
#     Boston
```

```
LDA_model <- lda(
    formula = Direction ~ Lag2,
    data = Weekly_from_1990_to_2008_inclusive
)
LDA_model
```

```
# Call:
# lda(Direction ~ Lag2, data = Weekly_from_1990_to_2008_inclusive)
#
# Prior probabilities of groups:
#      Down        Up
# 0.4477157 0.5522843
#
# Group means:
#             Lag2
# Down -0.03568254
# Up    0.26036581
#
# Coefficients of linear discriminants:
#            LD1
# Lag2 0.4414162
```

According to https://www.andreaperlato.com/mlpost/linear-discriminant-analysis/, "Linear Discriminant Analysis was originally developed by R.A. Fisher to classify subjects into one of... two clearly defined groups. It was later expanded to classify subjects into more than two groups. [LDA] helps to find linear combination[s] of original variables that provide[s] the best possible separation between the groups."

According to http://strata.uga.edu/8370/lecturenotes/discriminantFunctionAnalysis.html, LDA for two groups "seeks a linear function that will maximum the differences among the groups... LDA will find an equation that maximizes the separation of the two groups using the variables measured for the cases in those two groups. If there are three variables in the data set (x, y, z), the discriminant function has the following linear form:

$$DF = a\,(x - \bar{x}) + b\,(y - \bar{y}) + c\,(z - \bar{z})$$

where a, b, and c are the coefficients (slopes) of the discriminant function. Each sample or case will therefore have a single value called its score.

Linear Discriminant Analysis "produces a number of discriminant functions (similar to principal components, and sometimes called axes) equal to the number of groups to be distinguished minus one."

For our LDA model, we have two groups. One group contains observations where each observation corresponds to a week when the market had a positive return. One group contains observations where each observation corresponds to a week when the market had a negative return. We have one predictor: $Lag2$.

"Coefficients of linear discriminants" "reports the coefficients of the discriminant function (a, b, and c). Because there are two groups, there are $2 - 1 = 1$ discriminant functions. Our one discriminant function

$$LD1 = \beta_{Lag2} \left(Lag2 - \bar{Lag2} \right) = 0.441 \left(Lag2 - 0.128 \right)$$

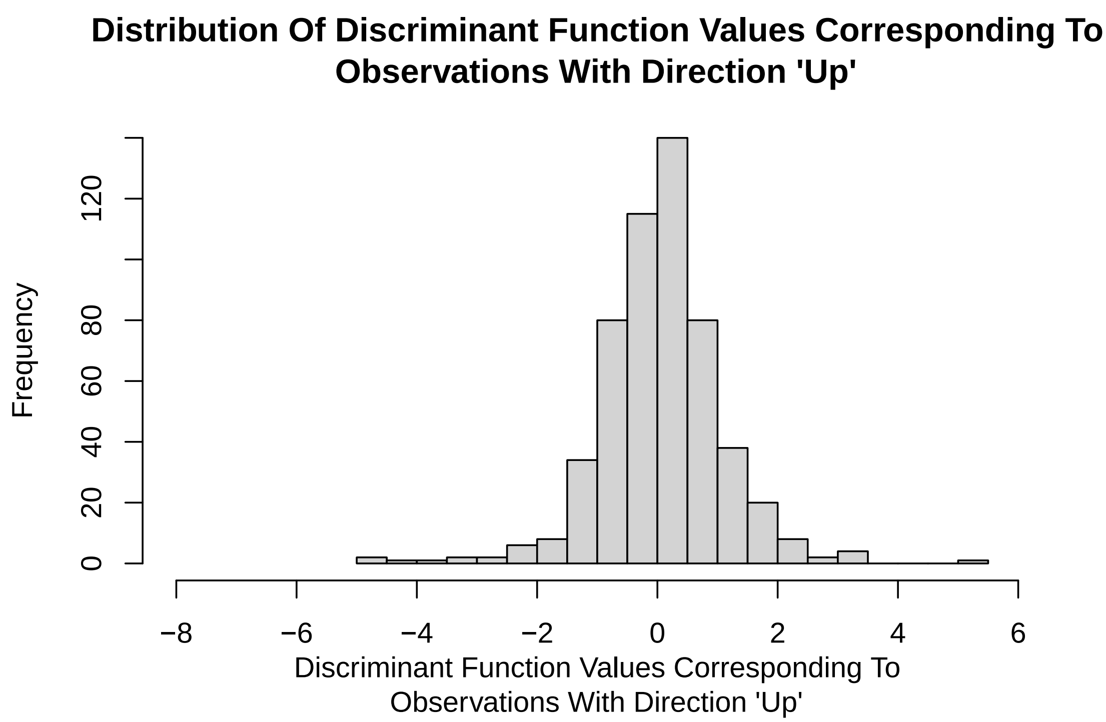
The group means are "average values of each of the variables for each of your groups." The mean value for $Lag2$ for our observations between 1990 and 2008 and for the group of weeks when the market had a positive return is 0.260. The mean value for $Lag2$ for our observations between 1990 and 2008 and for the group of weeks when the market had a negative return is -0.036. For a week when the market had a positive return, the return two weeks previously was likely positive. For a week when the market had a negative return, the return two weeks previously was likely negative.

"The prior probabilities of the groups... reflect... the proportion of each group within the dataset. In other words, if you had no measurements and the number of measured samples represented the actual abundances of the groups, the prior probabilities would describe the probability that any unknown sample would belong to each of the groups." The market had a positive return on a given week 55.2 percent of the time.

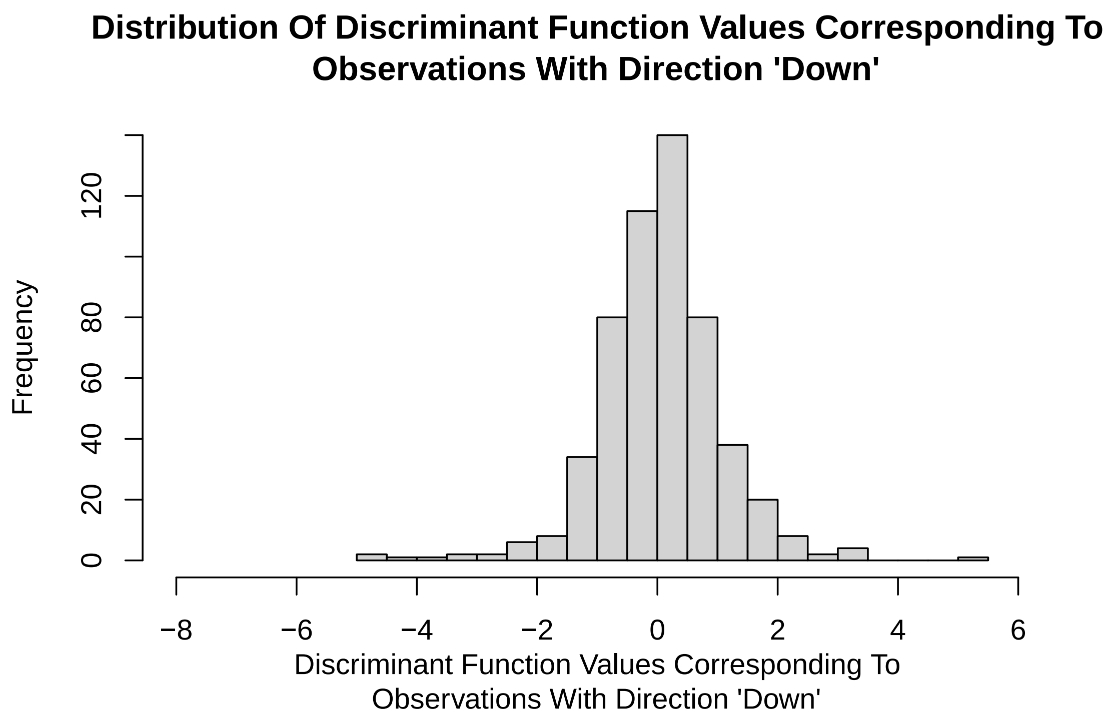
"Distribution Of Discriminant Function Values Corresponding To Observations With Direction 'Up'" and "Distribution of Discriminant Function Values Corresponding To Observations With Direction 'Down'" are plotted below. There is poor "separation of the groups" along discriminant function 1".

```
# The height of each version of the below histograms
# produced by `plot(LDA_model)` is about 2.25 inches.
prediction <- predict(LDA_model, newdata = Weekly_from_1990_to_2008_inclusive)
vector_of_discriminant_function_values <- prediction$x
training_observations_have_direction_Up <-
    Weekly_from_1990_to_2008_inclusive$Direction == "Up"
training_observations_have_direction_Down <-
    Weekly_from_1990_to_2008_inclusive$Direction == "Down"
indices_of_observations_with_direction_Up <-
    which(training_observations_have_direction_Up)
indices_of_observations_with_direction_Down <-
    which(training_observations_have_direction_Down)
vector_of_discriminant_function_values_corresponding_to_direction_Up <-
    vector_of_discriminant_function_values[
        indices_of_observations_with_direction_Up
    ]
vector_of_discriminant_function_values_corresponding_to_direction_Down <-
    vector_of_discriminant_function_values[
        indices_of_observations_with_direction_Up
    ]
hist(
    x = vector_of_discriminant_function_values_corresponding_to_direction_Up,
    xlim = c(-8, 6),
    breaks = 20,
    xlab = paste(
        "Discriminant Function Values Corresponding To\n",
        "Observations With Direction 'Up'",
```

```
        sep = ""
    ),
    ylab = "Frequency",
    main = paste(
        "Distribution Of Discriminant Function Values Corresponding To\n",
        "Observations With Direction 'Up'",
        sep = ""
    )
)
```




```
hist(
    x = vector_of_discriminant_function_values_corresponding_to_direction_Down,
    xlim = c(-8, 6),
    breaks = 20,
    xlab = paste(
        "Discriminant Function Values Corresponding To\n",
        "Observations With Direction 'Down'",
        sep = ""
    ),
    ylab = "Frequency",
    main = paste(
        "Distribution Of Discriminant Function Values Corresponding To\n",
        "Observations With Direction 'Down'",
        sep = ""
    )
)
```




```r
#ldahist(
#    data = vector_of_discriminant_function_values,
#    g = Weekly_from_1990_to_2008_inclusive$Direction
#)
#plot(LDA_model)
```

```r
prediction <- predict(LDA_model, newdata = Weekly_from_2009_to_2010_inclusive)
vector_of_directions <- prediction$class
confusion_matrix <-
    table(vector_of_directions, Weekly_from_2009_to_2010_inclusive$Direction)
confusion_matrix
```

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

```r
number_of_true_negatives <- confusion_matrix[1, 1]
number_of_false_negatives <- confusion_matrix[1, 2]
number_of_false_positives <- confusion_matrix[2, 1]
number_of_true_positives <- confusion_matrix[2, 2]
number_of_correct_predictions <-
    number_of_true_negatives + number_of_true_positives
fraction_of_correct_predictions <-
    number_of_correct_predictions / number_of_observations
fraction_of_correct_predictions
```

```
# [1] 0.625
```

The overall fraction of correct predictions is 65/104. Our Linear Discriminant Analysis model has fraction of correct predictions equal to that of our Logistic Regression model. The test error rate is 37.5 percent. For weeks when the market had a positive return, the model is correct 92.1 percent of the time / has a sensitivity, recall, hit rate, and True Positive Rate $TPR = \frac{TP}{P} = \frac{TP}{TP+FN} = \frac{56}{56+5} = 0.918$. For weeks when the market had a negative return, the model is correct 11.2 percent of the time / has a specificity, selectivity, and True Negative Rate $TNR = \frac{TN}{N} = \frac{TN}{TN+FP} = \frac{34}{34+9} = 0.209$.

(f) Repeat (d) using QDA.

```
QDA_model <- qda(
    formula = Direction ~ Lag2,
    data = Weekly_from_1990_to_2008_inclusive
)
QDA_model
```

```
# Call:
# qda(Direction ~ Lag2, data = Weekly_from_1990_to_2008_inclusive)
#
# Prior probabilities of groups:
#      Down        Up
# 0.4477157 0.5522843
#
# Group means:
#             Lag2
# Down -0.03568254
# Up    0.26036581
```

```
prediction <- predict(QDA_model, newdata = Weekly_from_2009_to_2010_inclusive)
vector_of_directions <- prediction$class
confusion_matrix <-
    table(vector_of_directions, Weekly_from_2009_to_2010_inclusive$Direction)
confusion_matrix
```

```
#
# vector_of_directions Down Up
#                 Down    0  0
#                 Up     43 61
```

```
number_of_true_negatives <- confusion_matrix[1, 1]
number_of_false_negatives <- confusion_matrix[1, 2]
number_of_false_positives <- confusion_matrix[2, 1]
number_of_true_positives <- confusion_matrix[2, 2]
number_of_correct_predictions <-
    number_of_true_negatives + number_of_true_positives
fraction_of_correct_predictions <-
    number_of_correct_predictions / number_of_observations
fraction_of_correct_predictions
```

```
# [1] 0.5865385
```

The overall fraction of correct predictions is 61/104. Our Quadratic Discriminant Analysis model predicts all observations as having direction "Up". Our Logistic Regression and Linear Discriminant Analysis models have fractions of correct predictions greater than that of our Quadratic Discriminant Analysis model. The test error rate is 41.3 percent. For weeks when the market had a positive return, the model is correct 92.1 percent of the time / has a sensitivity, recall, hit rate, and True Positive Rate $TPR = \frac{TP}{P} = \frac{TP}{TP+FN} = \frac{61}{61+0} = 1$. For weeks when the market had a negative return, the model is correct 0 percent of the time / has a specificity, selectivity, and True Negative Rate $TNR = \frac{TN}{N} = \frac{TN}{TN+FP} = \frac{0}{0+43} = 0$.

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

```
library(class)
number_of_observations_for_training <- nrow(Weekly_from_1990_to_2008_inclusive)
matrix_of_values_of_predictors_for_training <- matrix(
    data = Weekly_from_1990_to_2008_inclusive$Lag2
)
number_of_observations_for_testing <- nrow(Weekly_from_2009_to_2010_inclusive)
matrix_of_values_of_predictors_for_testing <- matrix(
    data = Weekly_from_2009_to_2010_inclusive$Lag2
)
vector_of_directions_for_training <- Weekly_from_1990_to_2008_inclusive$Direction
set.seed(1)
vector_of_directions <- knn(
    train = matrix_of_values_of_predictors_for_training,
    test = matrix_of_values_of_predictors_for_testing,
    cl = vector_of_directions_for_training,
    k = 1
)
confusion_matrix <-
    table(vector_of_directions, Weekly_from_2009_to_2010_inclusive$Direction)
confusion_matrix
```

```
#
# vector_of_directions Down Up
#                 Down   21 30
#                 Up     22 31
```

```
number_of_true_negatives <- confusion_matrix[1, 1]
number_of_false_negatives <- confusion_matrix[1, 2]
number_of_false_positives <- confusion_matrix[2, 1]
number_of_true_positives <- confusion_matrix[2, 2]
number_of_correct_predictions <-
    number_of_true_negatives + number_of_true_positives
fraction_of_correct_predictions <-
    number_of_correct_predictions / number_of_observations
fraction_of_correct_predictions
```

```
# [1] 0.5
```

The overall fraction of correct predictions is 52/104. Our Logistic Regression, Linear Discriminant Analysis, and Quadratic Discriminant Analysis models have fractions of correct predictions greater than that of our K Nearest Neighbors model. The test error rate is 50 percent. For weeks when the market had a positive return, the model is correct 50.8 percent of the time / has a sensitivity, recall, hit rate, and True Positive Rate $TPR = \frac{TP}{P} = \frac{TP}{TP+FN} = \frac{31}{31+30} = 0.508$. For weeks when the market had a negative return, the model is correct 48.8 percent of the time / has a specificity, selectivity, and True Negative Rate $TNR = \frac{TN}{N} = \frac{TN}{TN+FP} = \frac{21}{21+22} = 48.8$.

(h) Repeat (d) using naive Bayes. (skip this exercise)

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

Our Logistic Regression and Linear Discriminant Analysis models have the greatest overall fraction of correct predictions of 65/104. Our Quadratic Discriminant Analysis model has an overall fraction of correct predictions of 61/104. Our K Nearest Neighbors model has an overall fraction of correct predictions of 52/104.

(j) Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confu- sion 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.

14. 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.

(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?

(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?

(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?

(g) Perform naive Bayes 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? (skip this exercise)

(h) 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 `mpg01` in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?