558 Homework 2

Corbin Charpentier

4/26/2022

```
#  include=FALSE}
rm(list=ls())

pp <- function(...) {
    print(paste0(...))
}
```

1. In this problem, we will make use of the Auto data set, which is part of the ISLR2 package.

(a) Fit a least squares linear model to the data, in order to predict mpg using all of the other predictors except for name. Present your results in the form of a table. Be sure to indicate clearly how any qualitative variables should be interpreted.

```
lm_model <- lm(
    mpg ~ cylinders + displacement + horsepower + weight + acceleration + year + factor(origin),
    data=Auto
)
summary(lm_model)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + factor(origin), data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0095 -2.0785 -0.0982  1.9856 13.3608
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.795e+01  4.677e+00  -3.839 0.000145 ***
## cylinders       -4.897e-01  3.212e-01  -1.524 0.128215
## displacement     2.398e-02  7.653e-03   3.133 0.001863 **
## horsepower      -1.818e-02  1.371e-02  -1.326 0.185488
## weight          -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## acceleration     7.910e-02  9.822e-02   0.805 0.421101
## year             7.770e-01  5.178e-02  15.005  < 2e-16 ***
## factor(origin)2  2.630e+00  5.664e-01   4.643 4.72e-06 ***
## factor(origin)3  2.853e+00  5.527e-01   5.162 3.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.307 on 383 degrees of freedom
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

I chose to only interpret `origin` as a qualitative variable since its values do not have any relationship with each other, unlike `cylinders` or `year` where there is a notion of "increasing". Each class of `origin` appears to be significant, including class 1, which we verify by looking at the significance of the y-intercept.

(b) What is the (training set) mean squared error of this model?

```
MSE <- mean((Auto$mpg - predict(lm_model, Auto))^2)
```

We used all the data to train. Mean squared error is: 10.682.

(c) What gas mileage do you predict for a Japanese car with three cylinders, displacement 100, horsepower of 85, weight of 3000, acceleration of 20, built in the year 1980?

```
auto_row <- as.data.frame.list(list(0, 3, 100, 85, 3000, 20, 80, 3, "Foo"), col.names=colnames(Auto))
predicted_mpg <- predict(lm_model, newdata=auto_row)
```

We predict 27.895 for this vehicle.

(d) On average, holding all other covariates fixed, what is the difference between the mpg of a Japanese car and the mpg of an American car?

Based in the coefficient of the Japanese dummy variable and the fact that American vehicles are the basis of the `origin` variable, the MPG of a Japanese care, on average, is more fuel efficient by ~2.8 miles per gallon.

(e) On average, holding all other covariates fixed, what is the change in mpg associated with a `10`-unit change in horsepower?

We see that the coefficient for the `horsepower` variable is `-1.818e-02`. Multiplying that by `10`, the unite change, the MPG change associated with a 10-unit change in `horsepower` is `-0.181`. Said a different way, for every increase of `10 hoursepower`, MPG is expected to drop by `0.181`.

2. Consider using only the origin variable to predict mpg on the Auto data set. In this problem, we will explore the coding of this qualitative variable.

(a) First, code the origin variable using two dummy (indicator) variables, with Japanese as the default value. Write out an equation like (3.30) in the textbook, and report the coefficient estimates. What is the predicted mpg for a Japanese car? for an American car? for a European car?

The linear model is as follows:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i \text{ th vehicle is from America} \\ \beta_0 + \beta_2 + \epsilon_i & \text{if } i \text{ th vehicle is from Europe} \\ \beta_0 + \epsilon_i & \text{if } i \text{ th vehicle is from Japan} \end{cases}$$

```
# Filter out all the variables we don't need
summary(lm(mpg ~ relevel(factor(origin), ref="3"), data=Auto))
```

```
##
## Call:
## lm(formula = mpg ~ relevel(factor(origin), ref = "3"), data = Auto)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.451  -5.034  -1.034   3.649  18.966
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         30.4506     0.7196  42.314  < 2e-16 ***
## relevel(factor(origin), ref = "3")1 -10.4172     0.8276 -12.588  < 2e-16 ***
## relevel(factor(origin), ref = "3")2  -2.8477     1.0581  -2.691  0.00742 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.396 on 389 degrees of freedom
## Multiple R-squared:  0.3318, Adjusted R-squared:  0.3284
## F-statistic:  96.6 on 2 and 389 DF,  p-value: < 2.2e-16
```

R's `lm()` function already does the dummy variable creation for us–we merely need to specify the default value. Using only the `origin` variable as a predictor, the expected MPG of a Japanese care is `30.4506`MPG, the y-intercept of the model. The predicted MPGs for European and American cars, respectively, are `30.4506 - 10.4172 = ~20.14` and `30.4506 - 2.8577 = ~27.65`

(b) Now, code the origin variable using two dummy (indicator) variables, with American as the default. Write out an equation like (3.30) in the textbook, and report the coefficient estimates. What is the predicted mpg for a Japanese car? for an American car? for a European car?

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i \text{ th vehicle is from Europe} \\ \beta_0 + \beta_2 + \epsilon_i & \text{if } i \text{ th vehicle is from Japan} \\ \beta_0 + \epsilon_i & \text{if } i \text{ th vehicle is from America} \end{cases}$$

```
# Filter out all the variables we don't need
summary(lm(mpg ~ relevel(factor(origin), ref="1"), data=Auto))
```

```
##
## Call:
## lm(formula = mpg ~ relevel(factor(origin), ref = "1"), data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.451  -5.034  -1.034   3.649  18.966
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         20.0335     0.4086  49.025   <2e-16 ***
## relevel(factor(origin), ref = "1")2  7.5695     0.8767   8.634   <2e-16 ***
## relevel(factor(origin), ref = "1")3 10.4172     0.8276  12.588   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.396 on 389 degrees of freedom
## Multiple R-squared:  0.3318, Adjusted R-squared:  0.3284
## F-statistic:  96.6 on 2 and 389 DF,  p-value: < 2.2e-16
```

The results are identical to problem 3b, except, here, the y-intercept represents MPG for American vehicles.

(c) Now, code the origin variable using two variables that take on values of +1 or -1. Write out an equation like (3.30) in the textbook, and report the coefficient estimates. What is the predicted mpg for a Japanese car? for an American car? for a European car?

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 - \beta_2 + \epsilon_i & \text{if } i\text{th vehicle is from America} \\ \beta_0 - \beta_1 + \beta_2 + \epsilon_i & \text{if } i\text{th vehicle is from Europe} \\ \beta_0 - \beta_1 - \beta_2 + \epsilon_i & \text{if } i\text{th vehicle is from Japan} \end{cases}$$

```
dfOrigin <- dplyr::select(Auto, c(mpg, origin))

dfWeird <- dfOrigin %>% mutate(
    american = ifelse(origin == 1, 1, -1),
    european = ifelse(origin == 2, 1, -1),
    japanese = ifelse(origin == 3, 1, -1),
)

summary(lm(mpg ~ american + european, data=dfWeird))
```

```
##
## Call:
## lm(formula = mpg ~ american + european, data = dfWeird)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.451  -5.034  -1.034   3.649  18.966
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  23.8182     0.4384  54.335  < 2e-16 ***
## american     -5.2086     0.4138 -12.588  < 2e-16 ***
## european     -1.4238     0.5290  -2.691  0.00742 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.396 on 389 degrees of freedom
## Multiple R-squared:  0.3318, Adjusted R-squared:  0.3284
## F-statistic:  96.6 on 2 and 389 DF,  p-value: < 2.2e-16
```

Predicted MPG for an American car = `23.8182 - (-5.2086) - (-1.5238) = 30.5506` Predicted MPG for a Japanese car = `23.8182 + (-5.2086) - (-1.5238) = 20.1334` Predicted MPG for a Japanese car = `23.8182 - (-5.2086) + (-1.5238) = 27.503`

(d) Finally, code the origin variable using a single variable that takes on values of 0 for Japanese, 1 for American, and 2 for European. Write out an equation like (3.30) in the textbook, and report the coefficient estimates. What is the predicted mpg for a Japanese car? for an American car? for a European car?

$$y_i = \beta_0 + \beta_1 x_{i1} + x_{i2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 * 1 + \epsilon_i & \text{if } i\text{th vehicle is from America} \\ \beta_0 + \beta_1 * 2 + \epsilon_i & \text{if } i\text{th vehicle is from European} \\ \beta_0 + \beta_1 * 3 + \epsilon_i & \text{if } i\text{th vehicle is from Japanese} \end{cases}$$

```
summary(lm(mpg ~ origin, data=Auto))
```

```
##
## Call:
```

```
## lm(formula = mpg ~ origin, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.2416  -5.2533  -0.7651   3.8967  18.7115
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.8120     0.7164   20.68   <2e-16 ***
## origin        5.4765     0.4048   13.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.447 on 390 degrees of freedom
## Multiple R-squared:  0.3195, Adjusted R-squared:  0.3177
## F-statistic: 183.1 on 1 and 390 DF,  p-value: < 2.2e-16
```

Predicted MPG for American car = `14.8120 + 5.4765(1) = 20.2885` Predicted MPG for European car = `14.8120 + 5.4765(2) = 25.765` Predicted MPG for Japanese car = `14.8120 + 5.4765(3) = 31.2415`

(e) Comment on your results in (a)-(d).

It makes sense that the results from 5a and 5b are exact matches since we're merely changing the order commutative arithmetic. It also makes sense that the predicted MPGs for 5c are very close to 5a and 5b, albeit with different coefficients (since the associated coefficient is reflected across zero instead of dropped out when the category isn't present), since a salient difference still exists between the presence and un-presence of a category (1 if its there, -1 if it's not) and we still have a variable for each category.

This change for 5d when we encode the qualitative variable as quantitative. Instead of encoding a qualitative state as summation of on-or-off variables, we encode it as a series of discrete intervals on the Real number line. OLS simply fits a model that interpolates through the three categories.

3. Fit a model to predict mpg on the Auto dataset using origin and horsepower, as well as an interaction between origin and horsepower. Present your results, and write out an equation like (3.35) in the textbook. On average, how much does the mpg of a Japanese car change with a one-unit increase in horsepower? How about the mpg of an American car? a European car?

$$y_i \approx \beta_0 + \beta_4 \times \text{horsepower}_i + \begin{cases} \beta_2 + \beta_5 \times \text{horsepower}_i \text{ if } i\text{th car is European} \\ \beta_3 + \beta_6 \times \text{horsepower}_i \text{ if } i\text{th car is Japanese} \\ 0 \text{ if } i\text{th car is American} \end{cases}$$

```
summary(lm(mpg ~ factor(origin) + horsepower + factor(origin):horsepower, data=Auto))
```

```
##
## Call:
## lm(formula = mpg ~ factor(origin) + horsepower + factor(origin):horsepower,
##     data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7415  -2.9547  -0.6389   2.3978  14.2495
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                34.476496   0.890665  38.709  < 2e-16 ***
## factor(origin)2            10.997230   2.396209   4.589 6.02e-06 ***
## factor(origin)3            14.339718   2.464293   5.819 1.24e-08 ***
## horsepower                 -0.121320   0.007095 -17.099  < 2e-16 ***
## factor(origin)2:horsepower -0.100515   0.027723  -3.626 0.000327 ***
## factor(origin)3:horsepower -0.108723   0.028980  -3.752 0.000203 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.422 on 386 degrees of freedom
## Multiple R-squared:  0.6831, Adjusted R-squared:  0.679
## F-statistic: 166.4 on 5 and 386 DF,  p-value: < 2.2e-16
```

Average MPG change per unit change in in horsepower for... American cars = `-0.121320` European cars = `-0.121320 + -0.100515 = -0.221835` Japanese cars = `-0.121320 + -0.108723 = -0.230043`

4. 5. and 6

SEE HAND-WRITTEN SECTION BELOW TYPED REPORT.

7. In this problem, you will generate data with $p = 2$ features and a qualitative response with $K = 3$ classes, and $n = 50$ observations per class. You will then apply linear discriminant analysis to the data.

(a) Generate data such that the distribution of an observation in the kth class follows a $N(\mu_k, \Sigma)$ distribution, for $k = 1, ..., K$. That is, the data follow a bivariate normal distribution with a mean vector μ_k that is specific to the kth class, and a covariance matrix Σ that is shared across the K classes. Choose Σ and $\mu_1, ..., \mu_K$ such that there is some overlap between the K classes, i.e. no linear decision boundary is able to perfectly separate the training data. Specify your choices for Σ and $\mu_1, ..., \mu_K$.

```
K <- 3 # Number of qualitative response classes
p <- 2 # Number of predictors
n <- 50 # Number of observations per class

gen_7_data_for_class <- function(klass, mus, covmat) {
    x1x2 <- as.data.frame.matrix(mvrnorm(n=n, mu=mus, Sigma=covmat)) %>%
        bind_cols(replicate(n, factor(klass)))
    colnames(x1x2) <- c("x1", "x2", "class")
    x1x2
}

gen_7_data <- function() {
    covmat <- matrix(c(1, 0.3, 0.3, 1), 2, 2)
    gen_7_data_for_class(1, c(-2, -2), covmat) %>%
        bind_rows(gen_7_data_for_class(2, c(0, 0), covmat)) %>%
        bind_rows(gen_7_data_for_class(3, c(2, 2), covmat))
}


df7a <- gen_7_data()

head(df7a)
```

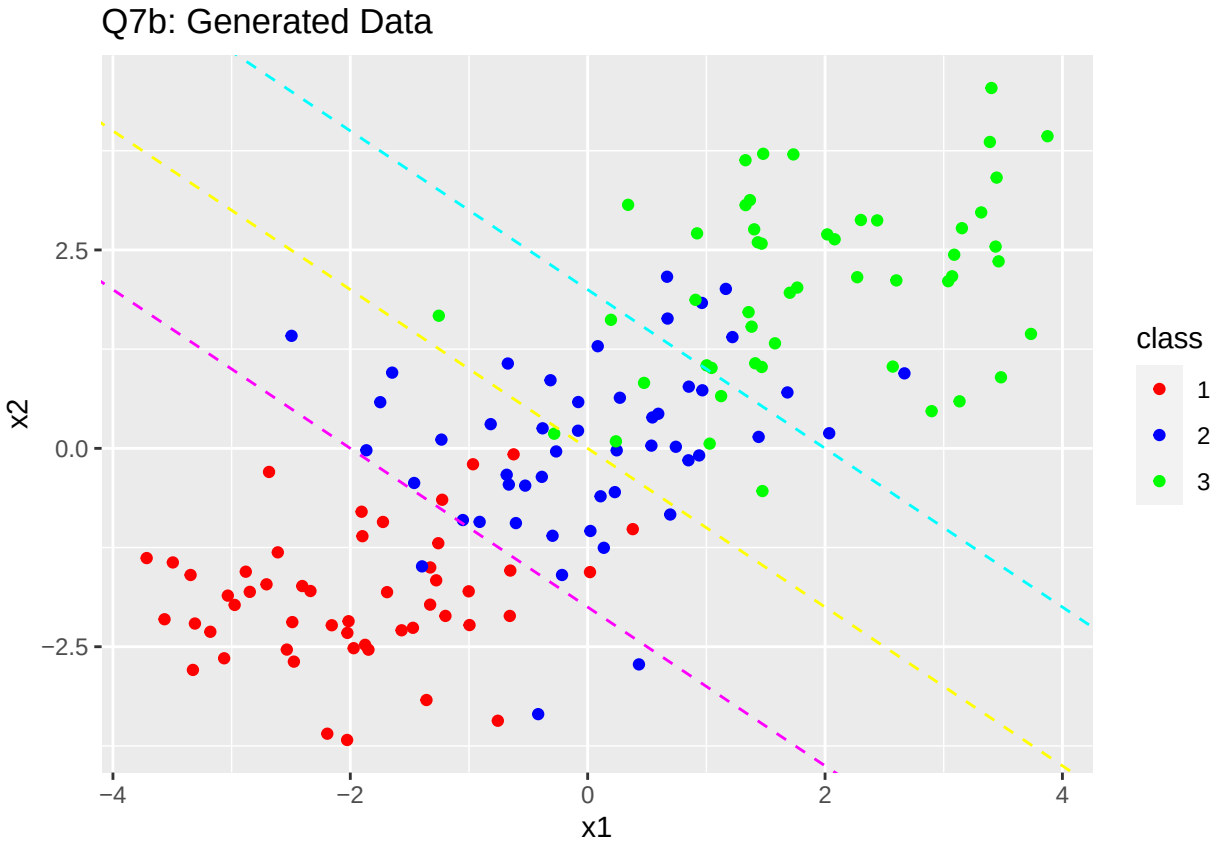
```
##           x1        x2 class
```

```
## 1 -3.7159397 -1.382782     1
## 2 -0.9955467 -2.226914     1
## 3 -3.0319852 -1.855949     1
## 4 -3.1794001 -2.311633     1
## 5 -1.1985497 -2.111443     1
## 6 -1.2762281 -1.663077     1
```

$$\Sigma_k = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix} \text{ for } k = 1, 2, 3$$
$$\boldsymbol{\mu_1} = (-2, -2)^T$$
$$\boldsymbol{\mu_2} = (0, 0)^T$$
$$\boldsymbol{\mu_3} = (2, 2)^T$$

(b) Plot the data, with the observations in each class displayed in a different color. Compute and display the Bayes decision boundary (or Bayes decision boundaries) on this plot. This plot should look something like the right-hand panel of Figure 4.6 in the textbook (although no need to worry about shading the background, and also you don't need to display the LDA decision boundary for this sub-problem — you will do that in the next sub-problem). Be sure to label which region(s) of the plot correspond to each class.

```
qplot(x1, x2, main="Q7b: Generated Data", color=class, data=df7a) +
    scale_color_manual(values=c("1"="red",
                                "2"="blue",
                                "3"="green")) +
    geom_abline(intercept=-2, slope=-1, color="magenta", linetype="dashed") + # k=1, k=2 boundary
    geom_abline(intercept=0, slope=-1, color="yellow", linetype="dashed") +      # k=1, k=3 boundary
    geom_abline(intercept=2, slope=-1, color="cyan", linetype="dashed")       # k=2, k=3 boundary
```



Magenta line: $k = 1, k = 2$ Boundary

Yellow line: $k = 1, k = 3$ Boundary

Cyan line: $k = 2, k = 3$ Boundary

(For the work to generate this lines, see hand written section at the end of this report)

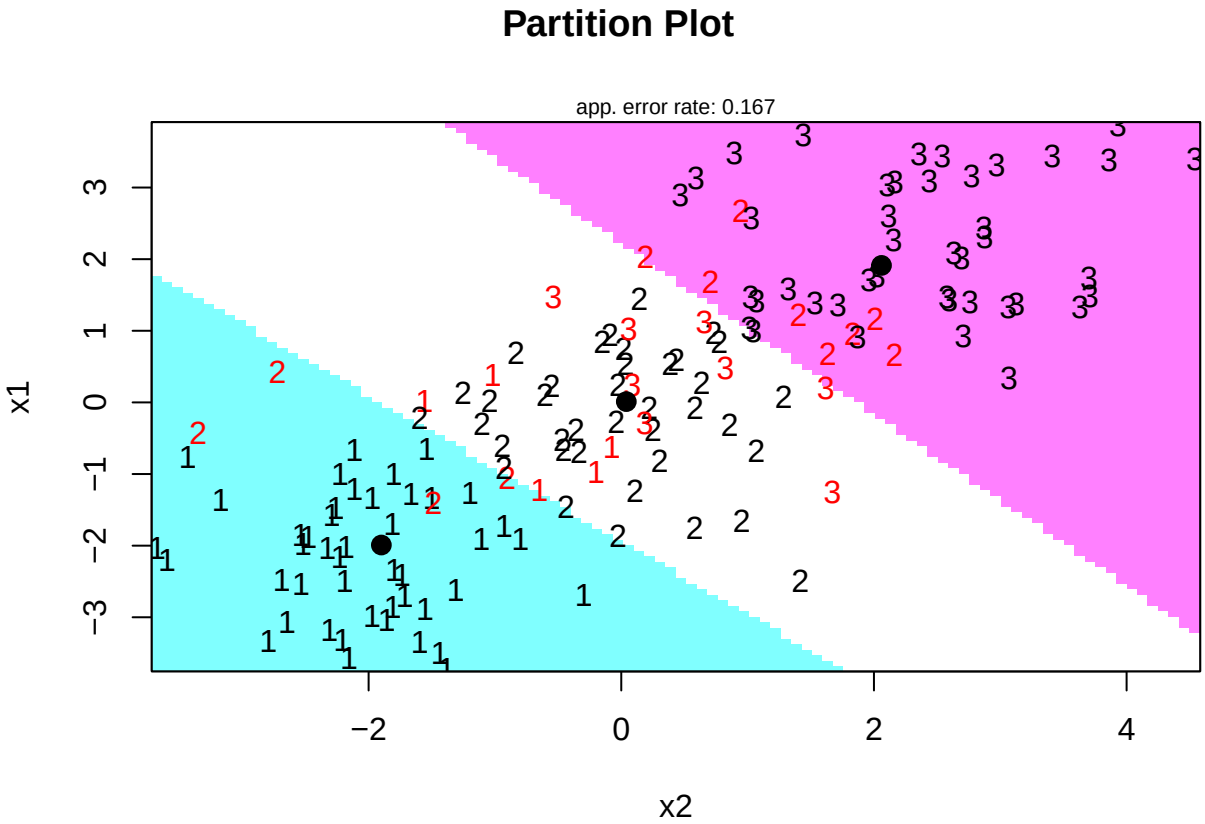
(c) Fit a linear discriminant analysis model to the data, and make a plot that displays the observations as well as the decision boundary (or boundaries) corresponding to this fitted model. How does the LDA decision boundary (which can be viewed as an estimate of the Bayes decision boundary) compare to the Bayes decision boundary that you computed and plotted in (b)?

```
ldaModel <- lda(class ~ ., data=df7a)
print(ldaModel)
```

```
## Call:
## lda(class ~ ., data = df7a)
##
## Prior probabilities of groups:
##         1         2         3
## 0.3333333 0.3333333 0.3333333
##
## Group means:
##            x1          x2
## 1 -1.99141227 -1.89895515
## 2  0.01124364  0.03985938
## 3  1.91264400  2.05994410
```

```
##
## Coefficients of linear discriminants:
##          LD1        LD2
## x1 0.5737602 -0.7943620
## x2 0.6317473  0.7833749
##
## Proportion of trace:
##    LD1    LD2
## 0.9997 0.0003
```

```
partimat(class ~ ., data=df7a, method="lda")
```



The plane is separated into three partitions. The blue region is the decision area for $k = 1$ The white region is the decision area for $k = 2$ The magenta region is the decision area for $k = 3$

The estimates are "not bad", meaning in the neighborhood of the data, the boundaries run nearly parallel with a slope of 1, as they should. The y-intercepts also appear to closely align with the Bayes' boundary lines.

(d) Report the KxK confusion matrix for the LDA model on the training data. The rows of this confusion matrix represent the predicted class labels, and the columns represent the true class labels. (See Table 4.4 in the textbook for an example of a confusion matrix.) Also, report the training error (i.e. the proportion of training observations that are misclassified).

```
testAndPrint <- function(model, df) {
    df['prediction'] <- predict(model, df)$class
    conf_mat <- confusion_matrix(
        targets=df$class,
        predictions=df$prediction
    )
```

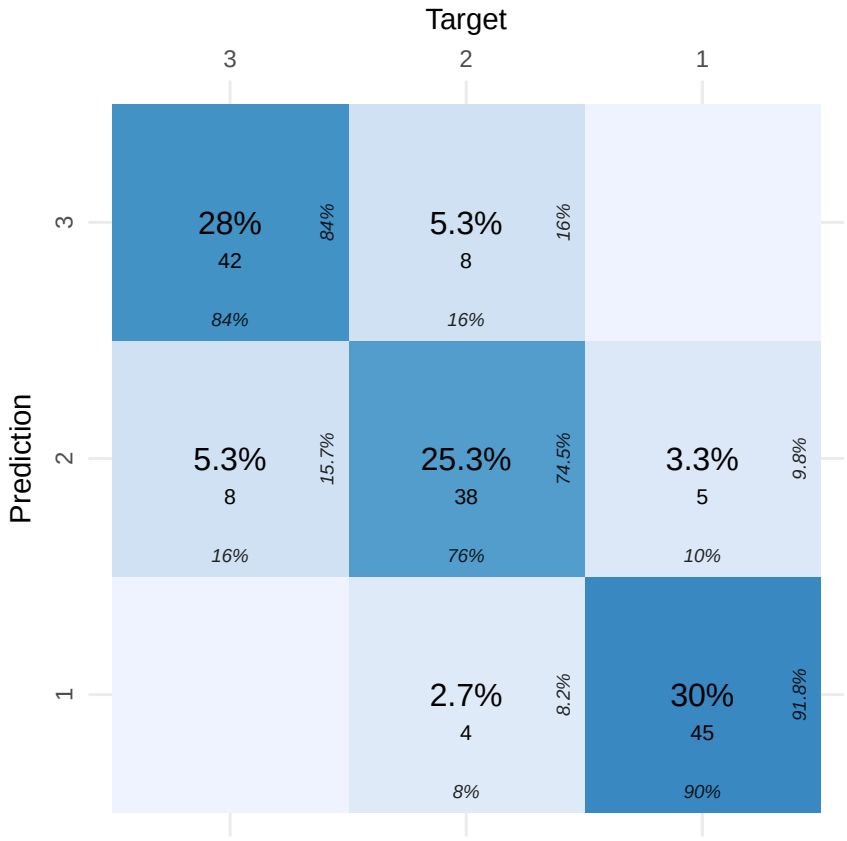
```
    # Now compute the training error
    perr <- 1 - sum(df$class == df$prediction)/150
    pp("Prediction error ", perr)

    conf_mat
}

conf_mat <- testAndPrint(ldaModel, df7a)
```

```
## [1] "Prediction error 0.166666666666667"
```

```
plot_confusion_matrix(conf_mat$`Confusion Matrix`[[1]])
```



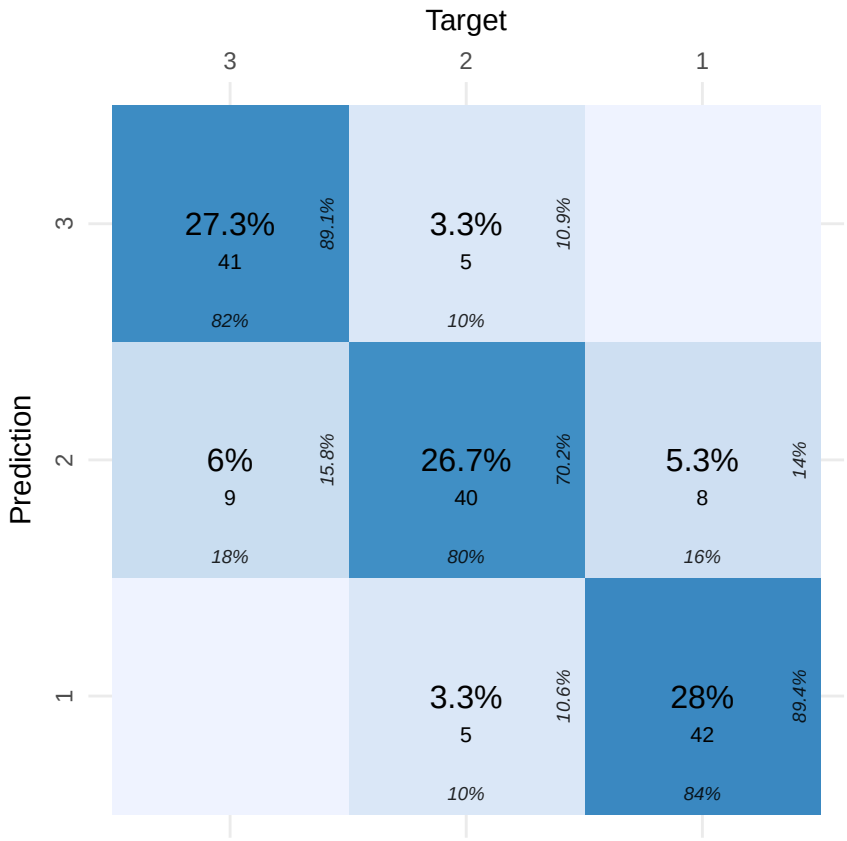
(e) Generate $n = 50$ test observations in each of the K classes, using the bivariate normal distributions from (a). Report the KxK confusion matrix, as well as the test error, that results from applying the model fit to the training data in (c) to your test data.

```
df7aTest <- gen_7_data()

conf_mat <- testAndPrint(ldaModel, df7aTest)
```

```
## [1] "Prediction error 0.18"
```

```
plot_confusion_matrix(conf_mat$`Confusion Matrix`[[1]])
```



(f) Compare your results from (d) and (e), and comment on your findings.

The results are highly comparable, which makes sense because the testing and training set were generated using the same distribution. The differences between the training and testing prediction results can be explained by the model bias, variance, and random noise. We also see that, because the Class 2 region is bounded on two sides, it has the highest error rate (Class 2 points spread into both the regions for Class 1 and Class 3).

8. In this problem, you will apply quadratic discriminant analysis to the data from Q7.

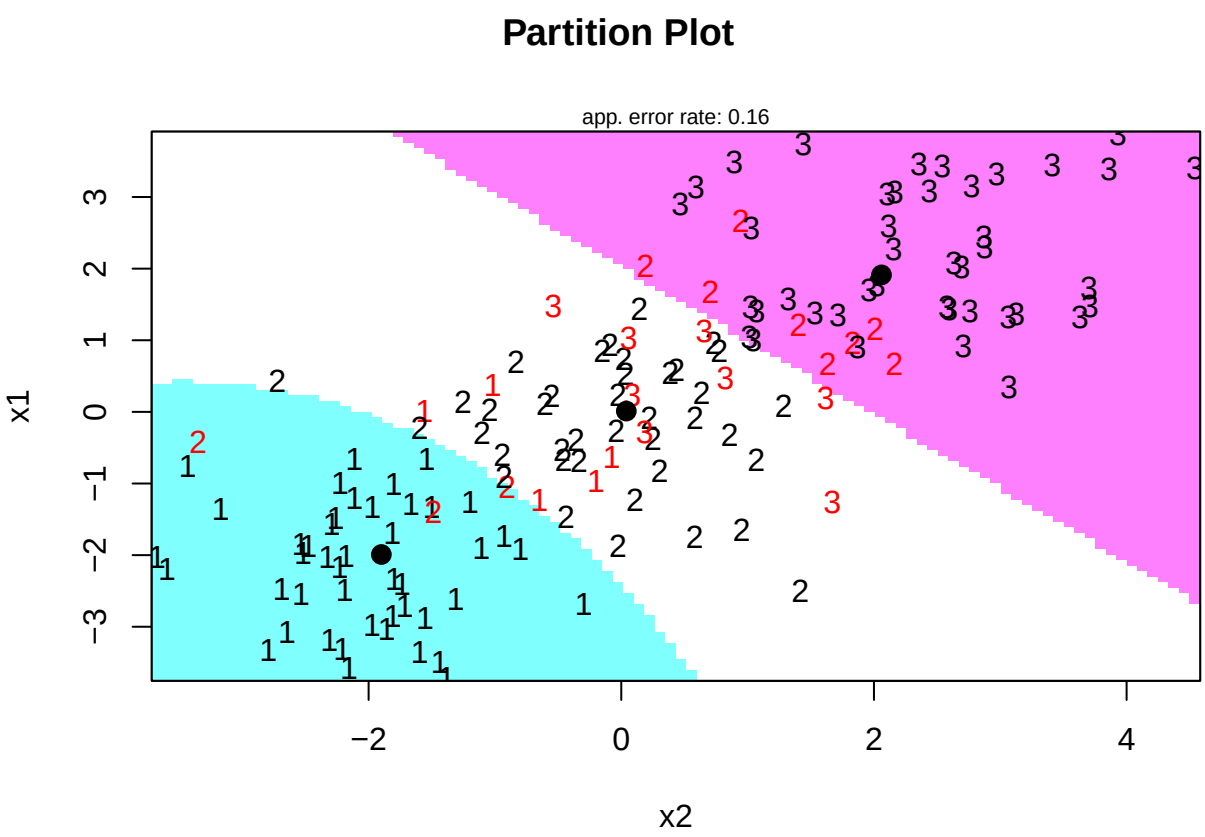
(a) Fit a quadratic discriminant analysis model to the training data from Q7,and make a plot that displays the observations as well as the QDA decision boundary (or boundaries) corresponding to this fitted model. Be sure to label which region(s) of the plot correspond to each class. How does the QDA decision boundary compare to the Bayes decision boundary that you computed in Q7(b)?

```
qdaModel <- qda(class ~ ., data=df7a)
print(qdaModel)
```

```
## Call:
## qda(class ~ ., data = df7a)
##
## Prior probabilities of groups:
##         1         2         3
## 0.3333333 0.3333333 0.3333333
##
```

```
## Group means:
##            x1          x2
## 1 -1.99141227 -1.89895515
## 2  0.01124364  0.03985938
## 3  1.91264400  2.05994410
```

```
partimat(class ~ ., data=df7a, method="qda")
```



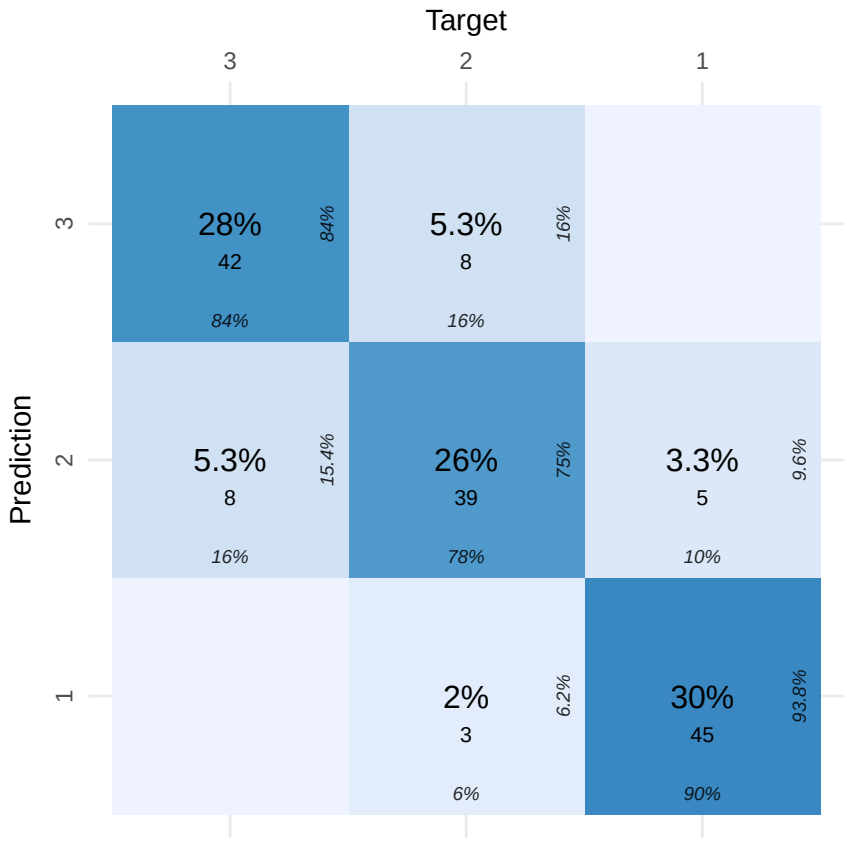
The plane is separated into three partitions. The blue region is the decision area for $k = 1$ The white region is the decision area for $k = 2$ The magenta region is the decision area for $k = 3$

(b) Report the KxK confusion matrix for the QDA model on the training data, as well as the training error.

```
conf_mat <- testAndPrint(qdaModel, df7a)
```

```
## [1] "Prediction error 0.16"
```

```
plot_confusion_matrix(conf_mat$`Confusion Matrix`[[1]])
```

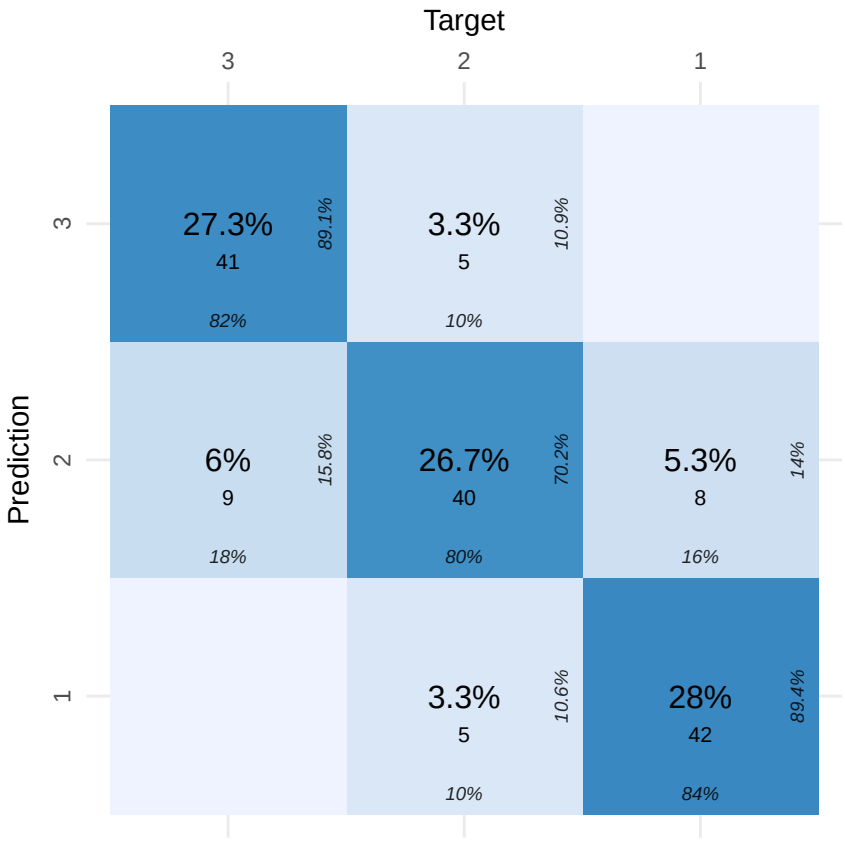


(c) Repeat (b), but this time using the test data generated in Q7. (That is, apply the model fit to the training data in (a) in order to calculate the test error.)

```
conf_mat <- testAndPrint(ldaModel, df7aTest)
```

```
## [1] "Prediction error 0.18"
```

```
plot_confusion_matrix(conf_mat$`Confusion Matrix`[[1]])
```



(d) Compare your results in (b) and (c), and comment on your findings.

Again, we expect these results to be high comparable (which they are) because the data is iid. It's a matter of random chance whether more points for a class in the test set land in that class's decision boundary established by the training set. The error rates between test and training sets would continue to converge as n increases.

(e) Which method had smaller *training error* in this example: LDA or QDA? Comment on your findings.

QDA. Intuitively, this may be because the generated data for each class becomes more sparse as distance increases from the mean, equally in all directions. This means that distribution of data is roughly circular around the mean (i.e. the boundary is not a line). Consequently, a quadratic function can better approximate this boundary than a linear one.

(f) Which method had smaller *test error* in this example: LDA or QDA? Comment on your findings.

QDA, for the same reasons as Q8e

9.

SEE HAND-WRITTEN SECTION BELOW

Homework 2

DATA 558

Corban Charpentier

(4) a. $Y = \beta_0 + \beta_1 X_1 + \varepsilon \Rightarrow Y = -165.1 + 4.8(69) = \boxed{142}$

b. OLS: $\hat{\beta} = (x^T x)^{-1} x^T y$ If each height in the training set is divided by 12

$\beta_0 : -165.1$ since β_0, the y-intercept, is the value of the predictor when x is 0, it will remain unchanged as x is scaled

$\beta_1 : 4.8(12) = 57.6$ As x is scaled down, its coefficient, β_1, will be scaled up to compensate

c. Given $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$

Suppose $x \in R^{n \times 2}$ and $x_{i2} = 12 x_{i1}$ ie $x \in \begin{bmatrix} x_{11} & 12 \cdot x_{11} \\ x_{21} & 12 \cdot x_{21} \\ \vdots & \vdots \\ x_{n1} & 12 x_{n1} \end{bmatrix}$ $y \in R^{n \times 1}$

$x^T y \rightarrow (2 \times n) \cdot (n \times 1) = 2 \times 1$

$\beta_0 = -165.1$ (one possible solution)

$\beta_1 = 4.8$

$\beta_2 = 57.6$

β_1 and β_2 are perfectly correlated, so the addition of either doesn't add any new information to the model

$\hat{\beta} = (x^T x)^{-1} x^T y$

Not that the input matrix will not be linearly independent (rank 1), so a pseudo inverse will be required, to find a single solution.

d. Training error will be the same for all of them since they are all equivalent models.

---

(5) a. $p = 1$ $X \in R$ $Y \in \{1, 2\}$ class $1 \sim N(\mu, \sigma^2)$ class $2 \sim \text{Unif}[-2, 2]$ Derive the expression of the Bayes decision boundary: x s.t. $P(Y=1 \mid X=x) = P(Y=2 \mid X=x)$

$P(X=x \mid Y=1) = f_1(x)$

$P(Y=1 \mid X=x) = P(Y=2 \mid X=x) \Rightarrow \frac{\pi_1 f_1(x)}{\pi_1 f_1(x) + \pi_2 f_2(x)} = \frac{\pi_2 f_2(x)}{\pi_1 f_1(x) + \pi_2 f_2(x)}$

$\Rightarrow \pi_1 f_1(x) = \pi_2 f_2(x)$

$-2 \le x \le 2 : \pi_1 \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left(\frac{-1}{2} \cdot \frac{(x-\mu)^2}{\sigma^2}\right) = \frac{1}{4}\pi_2$

$\Rightarrow \exp\left(\frac{-1}{2} \cdot \frac{(x-\mu)^2}{\sigma^2}\right) = \frac{\pi_2 \sigma \sqrt{2\pi}}{4\pi_1}$

↓ (Next page) ↓

$$\Rightarrow -\frac{1}{2}\cdot\frac{(x-\mu)^2}{\sigma^2} = \ln\left(\frac{\pi_2\sigma\sqrt{2\pi}}{\pi_1}\right)$$

$$\Rightarrow x = \sqrt{-2\sigma^2\ln\left(\frac{\pi_2\sigma\sqrt{2\pi}}{\pi_1}\right)} + \mu$$

$x \notin (-2, x>2$: undefined since we'd have $\ln(0)$

b. Assume $\mu=0, \sigma=1, \pi_1=0.45 \Rightarrow \pi_2=0.55$

$$x = \sqrt{-2(1)^2\cdot\ln\left(\frac{0.55(1)(\sqrt{2\pi})}{1(0.45)}\right)} + 0 \approx 1.93$$

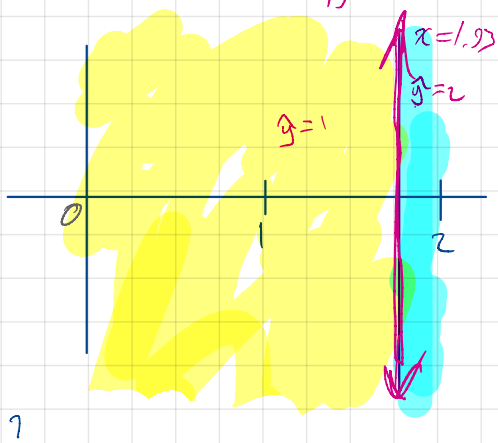
Now lets see which class yields a higher probability for $x=1$

$$f_1(1) = \frac{1}{\sqrt{2\pi}}\cdot\exp\left(\frac{(1-0)^2}{2}\right) = \frac{1}{\sqrt{2\pi}}\cdot\exp\left(\frac{1}{2}\right) \approx 0.66$$

$$P(Y=1|X=1) = \frac{\pi_1 f_1(1)}{\pi_1 f_1(1) + \pi_2 f_2(1)} = \frac{0.45(0.66)}{0.45(0.66)+0.55\left(\frac{1}{4}\right)} = 0.68$$

$$P(Y=2|X=1) = \frac{\pi_2 f_2(1)}{\pi_1 f_1(1) + \pi_2 f_2(1)} = \frac{0.55\left(\frac{1}{4}\right)}{0.45(0.66)+0.55\left(\frac{1}{4}\right)} = 0.316$$

$$\Rightarrow \begin{cases} 0 \le x < 1.93 \rightarrow 1 = \hat{y} \\ 1.93 \le x \le 2 \rightarrow 2 = \hat{y} \end{cases}$$



C. Observe n training observations $(x_1, y_1), \ldots, (x_n, y_n)$

Q: How to use these values to estimate μ, σ, π_1?

Let $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $T_k = \{(x_i, y_i) : y_i = k, (x_i, y_i) \in S\}$ $k \in \{1, 2\}$

card(·) be the cardinality of a set

$$\pi_1 = \frac{\text{card}(T_1)}{\text{card}(S)}$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\mu)^2}, \quad x_i \text{ from } (x, y) \in T_1$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad x \text{ from } (x, y) \in T_1$$

d. Given $X = x_0$, estimate $P(Y=1 \mid X=x_0)$

Let $f_1(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-1}{2} \cdot \frac{(x-\mu)^2}{\sigma^2}\right)$

$\pi_1, \mu, \sigma \equiv$ their values from problem (5c)

$\pi_2 = 1 - \pi_1$

$$P(Y=1 \mid X=x_0) = \frac{\pi_1 f_1(x_0)}{\pi_1 f_1(x_0) + \pi_2 f_2(x_0)}$$ (no unknowns)

---

(6)

a. Logistic Regression: observe $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$ log-odds $= 0.7$

Find $P(Y=1 \mid X=x)$

$P(Y=1 \mid X=x)$

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x_1 + \dots \beta_p x_p = 0.7$$

$$\Rightarrow \frac{p(x)}{1-p(x)} = \exp(0.7)$$

$$\Rightarrow p(x) = \frac{\exp(0.7)}{1+\exp(0.7)}$$

$$\Rightarrow P(Y=1 \mid X=x) = \frac{\exp(0.7)}{1+\exp(0.7)}$$

for $x = (x_1, \dots, x_p)^T$

b. $x^* = (x_1+1, x_2-1, x_3+2, x_4 \dots, x_p)^T$ Find $P(Y=1 \mid X=x^*)$

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1(x_1+1) + \beta_2(x_2-1) + \beta_3(x_3+2) + \dots + \beta_p x_p - (\beta_1 - \beta_2 + 2\beta_3) = 0.7$$

$$\Rightarrow \log\left(\frac{p(x^*)}{1-p(x^*)}\right) = 0.7 + \beta_1 - \beta_2 + 2\beta_3$$

$$\Rightarrow P(Y=1 \mid X=x^*) = \frac{\exp(0.7 + \beta_1 - \beta_2 + 2\beta_3)}{1+\exp(0.7 + \beta_1 - \beta_2 + 2\beta_3)}$$

For problems (7) and (8), see code

(7) b. Calculate the Bayes decision boundaries

$$P(Y=k|X=x) = \frac{P(X=x|Y=k)\,P(Y)}{P(X)} \qquad P(Y)=\frac{1}{3} \qquad P(X)=?$$

Let $\Sigma_k = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}$ for $k=1,2,3$

$\mu_1 = (-2,-2)^T$, $\mu_2 = (0,0)^T$, $\mu_3 = (2,2)^T$

k=1, k=2 Decision Boundary:

$$\frac{P(X=x|Y=1)\cdot\frac{1}{3}}{P(X)} = \frac{P(X=x|Y=2)\cdot\frac{1}{3}}{P(X)}$$

$$\Rightarrow P(X=x|Y=1) = P(X=x|Y=2)$$

$$\Rightarrow \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)\right)$$

$$\Rightarrow -\tfrac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) = -\tfrac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)$$

$$\Rightarrow (x-\mu_1)^T(x-\mu_1) = (x-\mu_2)^T(x-\mu_2)$$

$$\Rightarrow (x_1+2)^2 + (x_2+2)^2 = (x_1-0)^2 + (x_2-0)^2$$

$$\Rightarrow x_1^2 + 4x_1 + 4 + x_2^2 + 4x_2 + 4 = x_1^2 + x_2^2$$

$$\Rightarrow 4x_1 + 4x_2 = -8$$

$$\Rightarrow \boxed{x_2 = -x_1 - 2}$$

(a b) $\begin{pmatrix} c \\ d \end{pmatrix}$ = ac + bd

$(x_1-\mu_{11} \quad x_2-\mu_{12}) \begin{pmatrix} x_1-\mu_{12} \\ x_2-\mu_{12} \end{pmatrix}$

k=1, k=3 Boundary

we can start here

$$(x-\mu_1)^T(x-\mu_1) = (x-\mu_3)^T(x-\mu_3)$$

$$\Rightarrow (x_1+2)^2 + (x_2+2)^2 = (x_1-2)^2 + (x_2-2)^2$$

$$\Rightarrow x_1^2 + 4x_1 + 4 + x_2^2 + 4x_2 + 4 = x_1^2 - 4x_1 + 4 + x_2^2 - 4x_2 + 4$$

$$\Rightarrow \boxed{x_2 = -x_1}$$

k=2, k=3 Boundary

$$(x-\mu_2)^T(x-\mu_2) = (x-\mu_3)^T(x-\mu_3)$$

$$\Rightarrow (x_1+0)^2 + (x_2+0)^2 = (x_1-2)^2 + (x_2-2)^2$$

$$\Rightarrow x_1^2 + x_2^2 = x_1^2 - 4x_1 + 4 + x_2^2 - 4x_2 + 4$$

$$\Rightarrow -4x_1 - 4x_2 + 8 = 0$$

$$\Rightarrow \boxed{x_2 = -x_1 + 2}$$

4 Derive an expression for the ridge regression estimates

Let $y \in \mathbb{R}^{n\times 1}$, $x \in \mathbb{R}^{n\times p}$, $\beta \in \mathbb{R}^{p\times 1}$, $R(\beta, y, x)$ be the ridge regression estimate

$$\text{Loss}(x, y, \beta) = \|y - x\beta\|^2 + \lambda \beta^T\beta$$

($\|\cdot\|^2$: 2-norm; $\lambda\beta^T\beta$: this is a constant)

$$= (y - x\beta)^T(y - x\beta) + \lambda\beta^T\beta$$

$$= y^Ty - y^Tx\beta - \beta^Tx^Ty + \beta^Tx^Tx\beta + \lambda\beta^T\beta$$

$$\frac{\partial \text{Loss}}{\partial \beta} = 0 - y^Tx - y^Tx + 2x^Tx\beta + 2\lambda\beta$$

$$= -2y^Tx + 2x^Tx\beta + 2\lambda\beta$$

✡ Now set to 0 and solve for β:

$$2x^Tx\beta + 2\lambda\beta = 2y^Tx$$

$$\Rightarrow (2x^Tx + 2\lambda)\beta = 2y^Tx$$

$$\Rightarrow \boxed{\beta = (x^Tx + \lambda)^{-1}y^Tx} = R(\beta, y, x)$$