ISLR | Chapter 9 Exercises

Marshall McQuillen

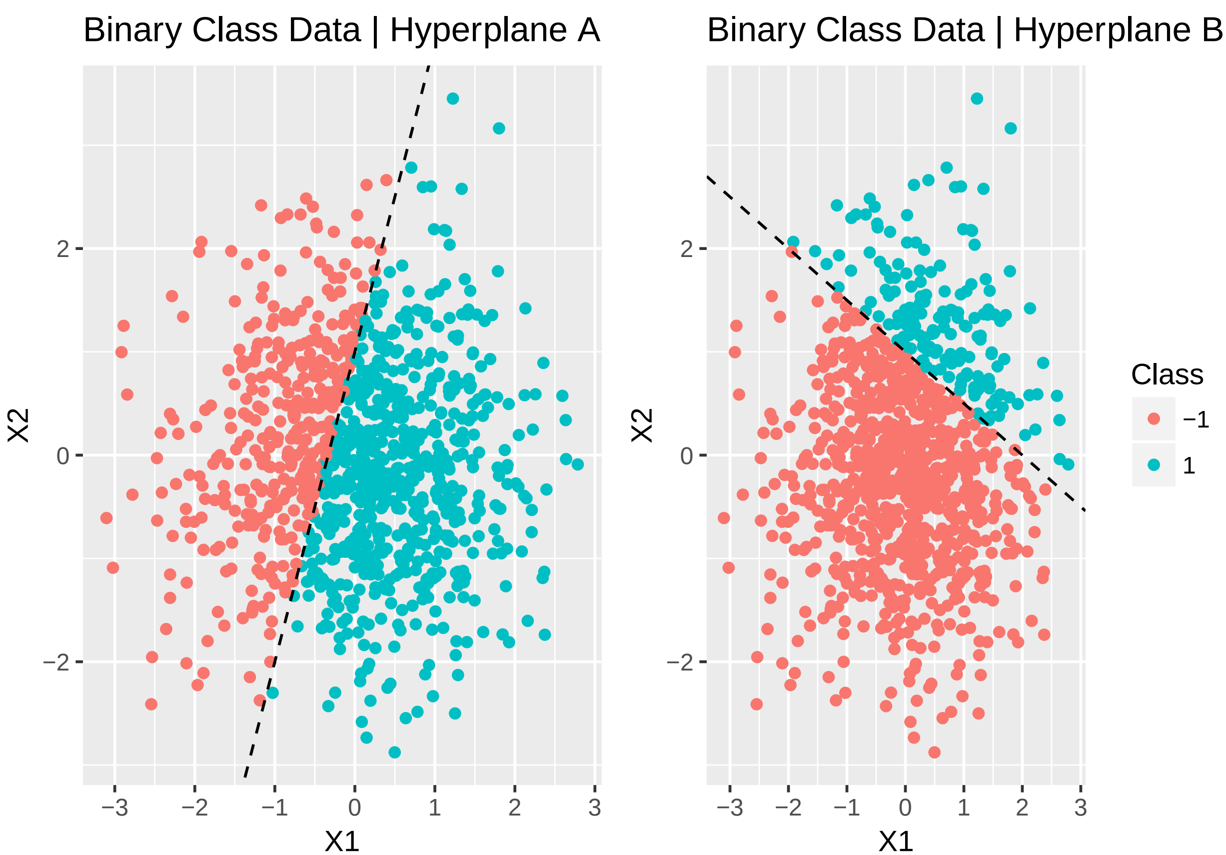
11/17/2018

Conceptual

1

- **A & B**.

```r
suppressPackageStartupMessages(library(gridExtra))
suppressPackageStartupMessages(library(ggplot2))

x <- matrix(rnorm(1000*2), ncol = 2)

hyperplane1 <- 1 + 3*x[, 1] - x[, 2]
y <- ifelse(hyperplane1 > 0, 1, -1)

hyperplane2 <- -2 + x[, 1] + 2*x[, 2]
y2 <- ifelse(hyperplane2 > 0, 1, -1)

par(mfrow = c(1, 2))

plot1 <- ggplot(data.frame(x = x,
                   y = factor(y, levels = c(-1, 1))), aes(x.1, x.2, color=y)) +
            geom_point(show.legend = FALSE) +
            geom_abline(intercept = 1,
                        slope = 3,
                        linetype = 'dashed') +
            ggtitle("Binary Class Data | Hyperplane A") +
            xlab("X1") +
            ylab("X2") +
            labs(color = "Class")

plot2 <- ggplot(data.frame(x = x,
                   y = factor(y2, levels = c(-1, 1))), aes(x.1, x.2, color=y)) +
            geom_point() +
            geom_abline(intercept = 1,
                        slope = -0.5,
                        linetype = 'dashed') +
            ggtitle("Binary Class Data | Hyperplane B") +
            xlab("X1") +
            ylab("X2") +
            labs(color = "Class")

grid.arrange(plot1, plot2, ncol = 2)
```

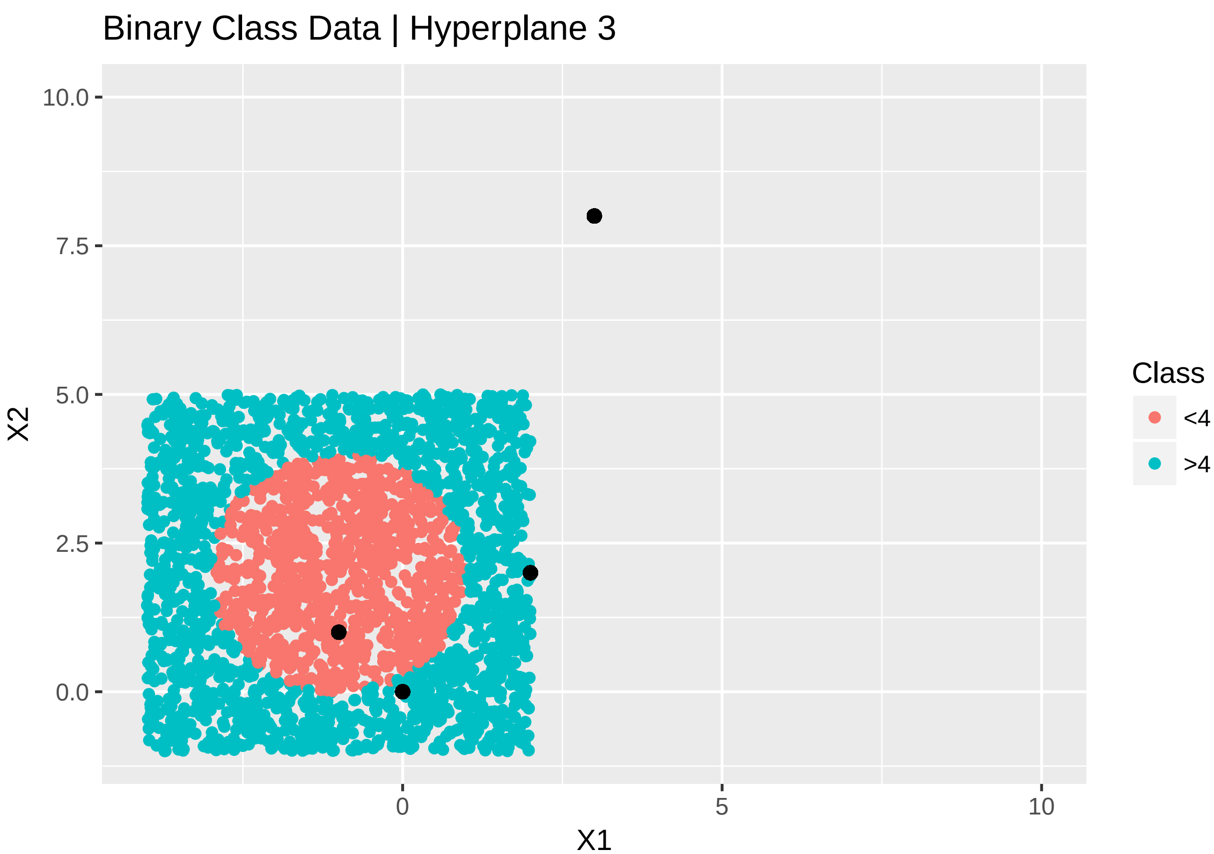


2

- **A, B & C**. The hyperplane is the circle encompassing the pink/orange data points below, where those data points are the ones who's value, when plugged into the equation $f(X_1, X_2) = (1 + X_1)^2 + (2 - X_2)^2 - 4$, will be negative. The blue data points output value of the above equation would be positive. The 4 data points plotted in black are those requested in part **C**, and it is clear which class they would fall into.

```
x_1 <- runif(2500, min = -4, max = 2)
x_2 <- runif(2500, min = -1, max = 5)

hyperplane3 <- (1 + x_1)^2 + (2 - x_2)^2 - 4
y3 <- factor(ifelse(hyperplane3 > 0, ">4", "<4"))

df <- data.frame(x1 = x_1,
                 x2 = x_2,
                 y = y3)

ggplot(df, aes(x1, x2, color=y)) +
    geom_point() +
    geom_point(x = 0,
               y = 0,
               col = 'black',
               cex = 2) +
    geom_point(x = -1,
```

```
             y = 1,
             col = 'black',
             cex = 2) +
  geom_point(x = 2,
             y = 2,
             col = 'black',
             cex = 2) +
  geom_point(x = 3,
             y = 8,
             col = 'black',
             cex = 2) +
  ggtitle("Binary Class Data | Hyperplane 3") +
  scale_y_continuous(limits = c(-1, 10)) +
  scale_x_continuous(limits = c(-4, 10)) +
  xlab("X1") +
  ylab("X2") +
  labs(color = "Class")
```



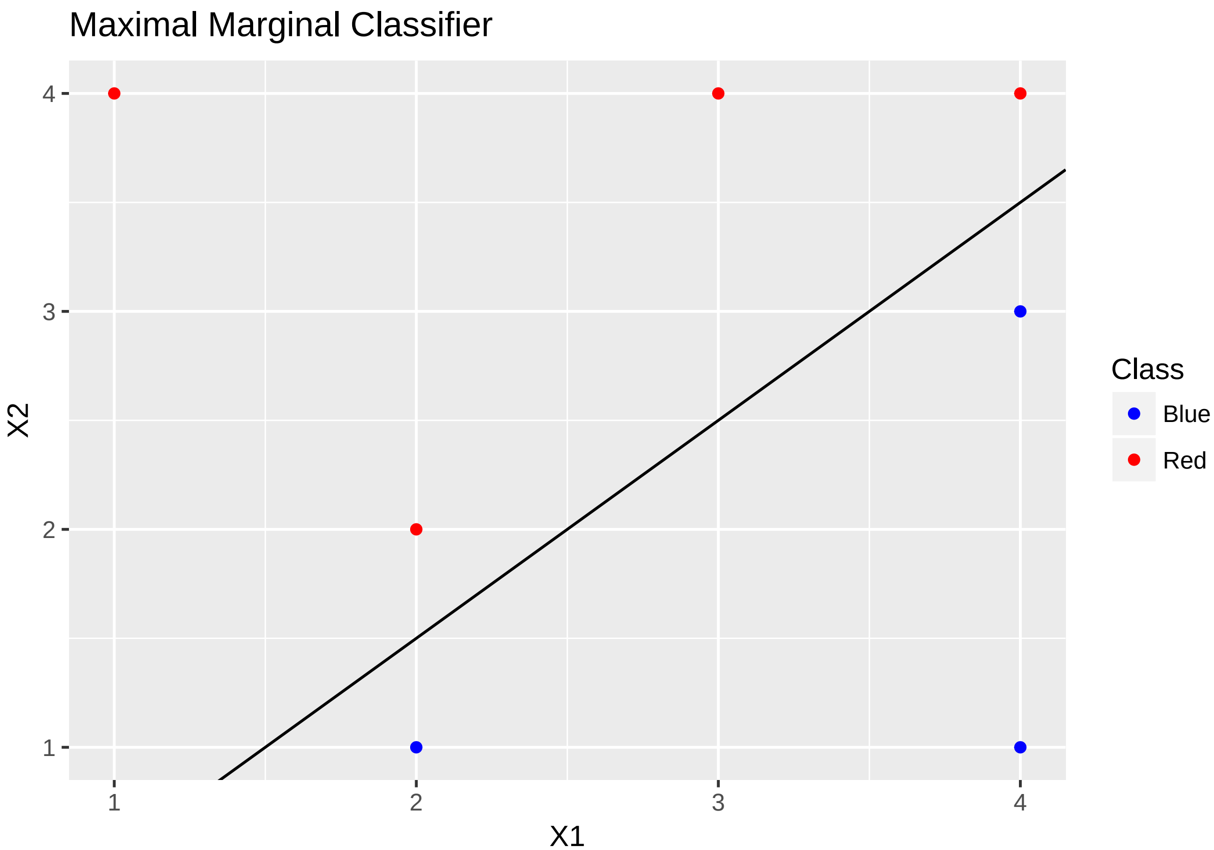
- **D**. One can see that the equation given in the text is non-linear with regard to X_1 and X_2. However, when expanded and refactored, it is clear that the hyperplane is linear with regard to to X_1, X_2, X_1^2 and X_2^2.

$$
\begin{align}
(1 + X_1)^2 + (2 - X_2)^2 &= 4 \\
(1 + X_1)^2 + (2 - X_2)^2 - 4 &= 0 \\
(1 + 2X_1 + X_1^2) + (4 - 4X_2 + X_2^2) - 4 &= 0 \\
1 + 2X_1 + X_1^2 - 4X_2 + X_2^2 &= 0 \\
\end{align}
$$

3

- **A & B**. The separating hyperplane has the equation $X_1 - X_2 - 0.5 = 0$.

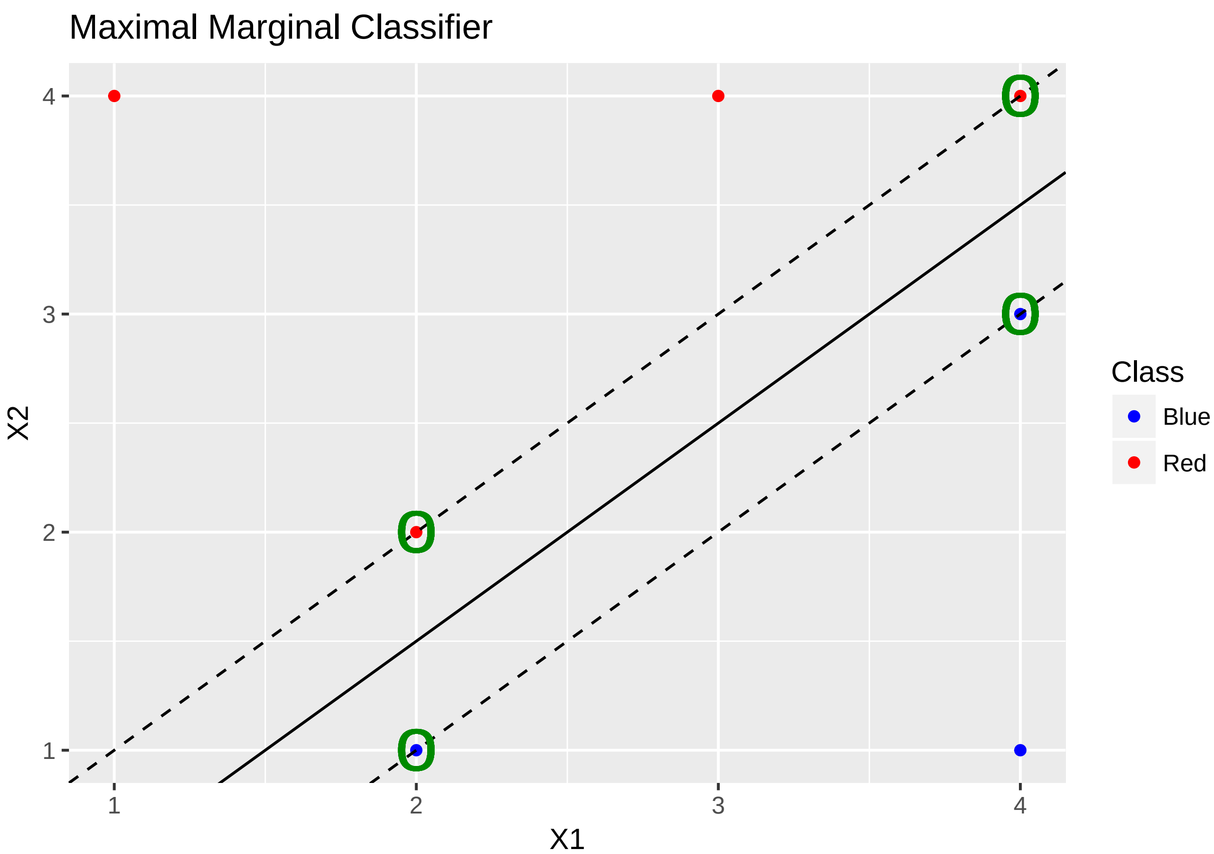
```r
df <- data.frame(x_1 = c(3,2,4,1,2,4,4),
                 x_2 = c(4,2,4,4,1,3,1),
                 y = c(rep("Red", 4), rep("Blue", 3)))

ggplot(df, aes(x_1, x_2, color = y)) +
    geom_point() +
    geom_abline(intercept = -0.5,
                slope = 1,
                linetype = 'solid') +
    scale_color_manual(values = c("Red" = 'red', "Blue" = 'blue')) +
    ggtitle("Maximal Marginal Classifier") +
    xlab("X1") +
    ylab("X2") +
    labs(color = 'Class')
```



- **C**. Classify a test observation x_t to Red if $(1)X_1 + (-1)X_2 - 0.5 < 0$, otherwise classify to Blue.
- **D**. The support vectors are those data points circled in green, the edge of the margins are the dashed lines.

```
ggplot(df, aes(x_1, x_2, color = y)) +
    geom_point() +
    geom_abline(intercept = -0.5,
                slope = 1,
                linetype = 'solid') +
    geom_abline(intercept = -0,
                slope = 1,
                linetype = 'dashed') +
    geom_abline(intercept = -1,
                slope = 1,
                linetype = 'dashed') +
    geom_point(x = 2,
               y = 2,
               color = 'green4',
               size = 10,
               shape = "o") +
    geom_point(x = 2,
               y = 1,
               color = 'green4',
               size = 10,
```

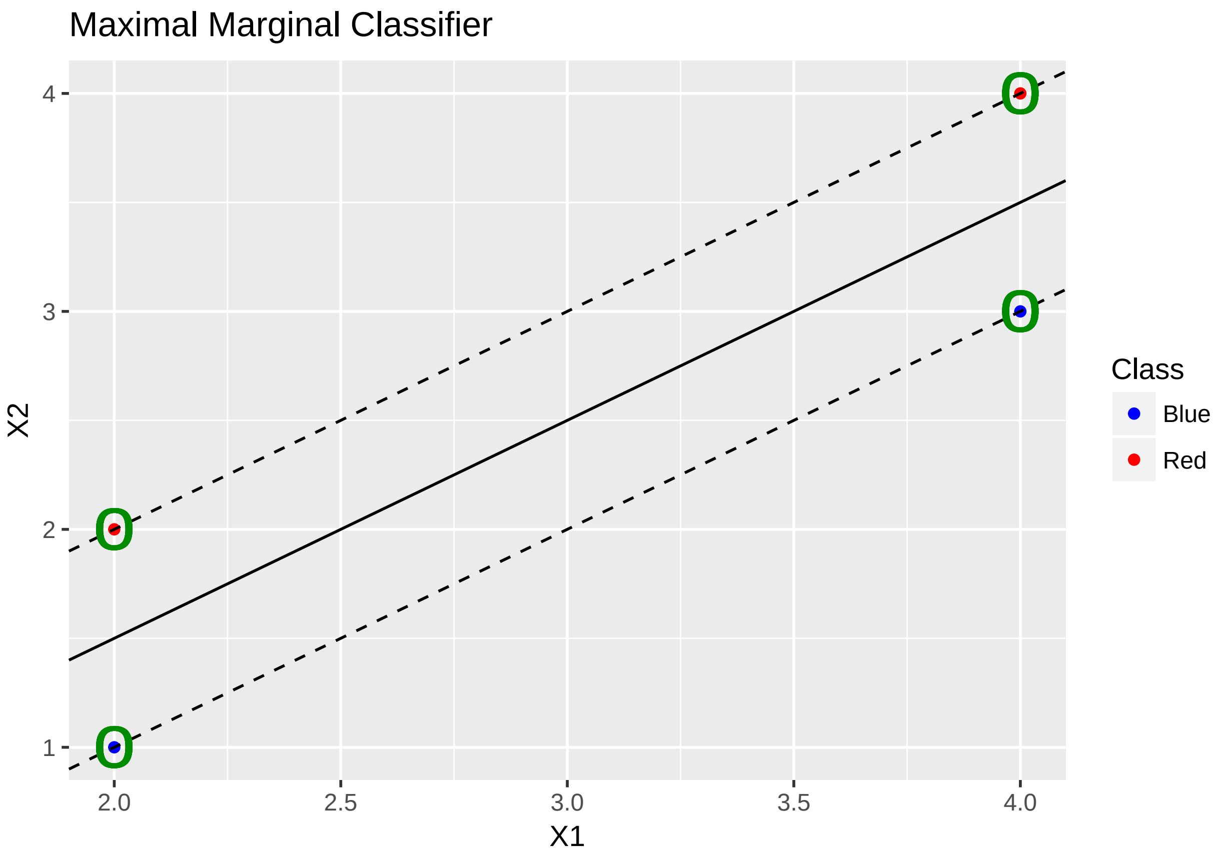
```r
              shape = "o") +
  geom_point(x = 4,
             y = 3,
             color = 'green4',
             size = 10,
             shape = "o") +
  geom_point(x = 4,
             y = 4,
             color = 'green4',
             size = 10,
             shape = "o") +
  scale_color_manual(values = c("Red" = 'red', "Blue" = 'blue')) +
  ggtitle("Maximal Marginal Classifier") +
  xlab("X1") +
  ylab("X2") +
  labs(color = 'Class')
```



- **F**. Not only can the seventh observation be removed from the data set without affecting the hyperplane, **any observation that is not a support vector can be removed without distorting the hyperplane,** as shown in the plot below.

```r
sub_df <- df[-c(1, 4, 7), ]
```

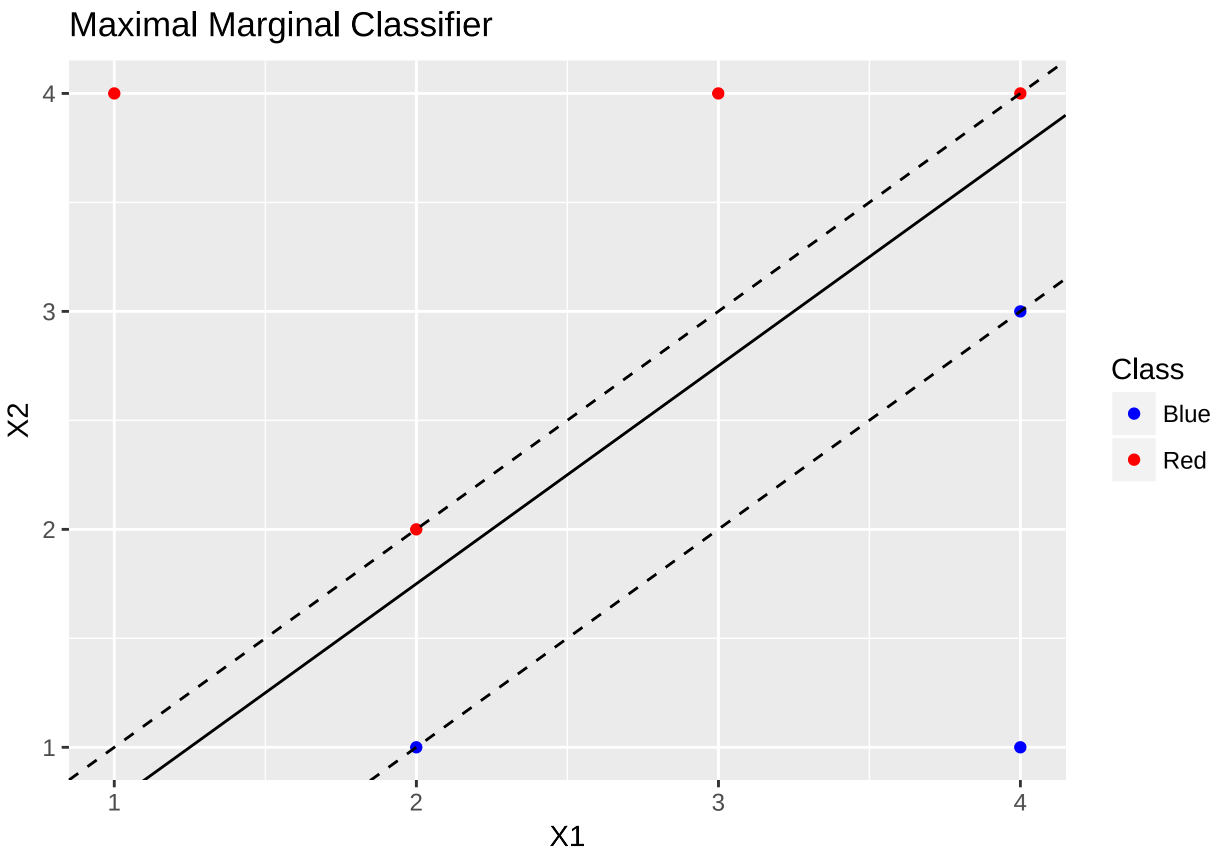
```r
ggplot(sub_df, aes(x_1, x_2, color = y)) +
    geom_point() +
    geom_abline(intercept = -0.5,
                slope = 1,
                linetype = 'solid') +
    geom_abline(intercept = -0,
                slope = 1,
                linetype = 'dashed') +
    geom_abline(intercept = -1,
                slope = 1,
                linetype = 'dashed') +
    geom_point(x = 2,
               y = 2,
               color = 'green4',
               size = 10,
               shape = "o") +
    geom_point(x = 2,
               y = 1,
               color = 'green4',
               size = 10,
               shape = "o") +
    geom_point(x = 4,
               y = 3,
               color = 'green4',
               size = 10,
               shape = "o") +
    geom_point(x = 4,
               y = 4,
               color = 'green4',
               size = 10,
               shape = "o") +
    scale_color_manual(values = c("Red" = 'red', "Blue" = 'blue')) +
    ggtitle("Maximal Marginal Classifier") +
    xlab("X1") +
    ylab("X2") +
    labs(color = 'Class')
```



- **G**. The below hyperplane, with the equation $(1)X_1 + (-1)X_2 - 0.25$ would not be the **maximal** separating hyperplane because the margin between the red support vectors and the hyperplane is smaller than the margin between the blue support vectors and the machine.

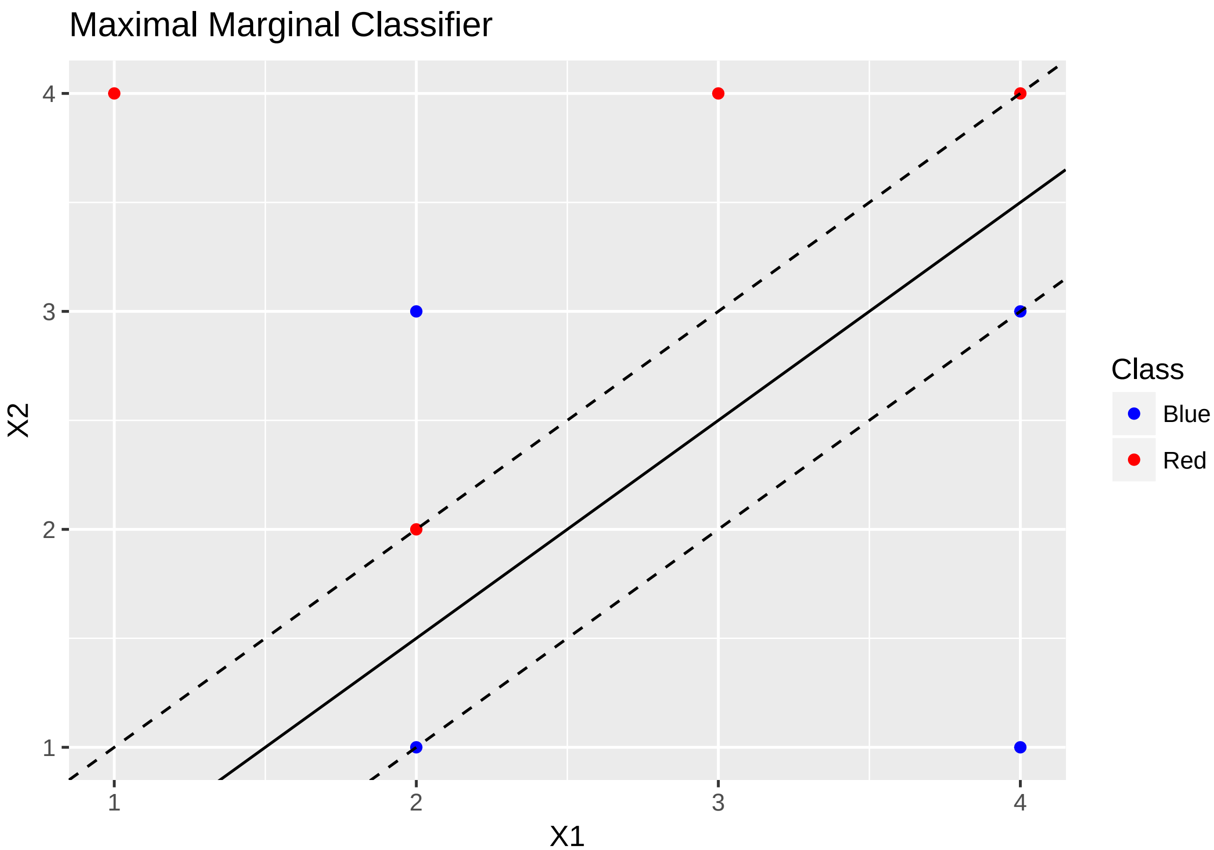
  The reason that this is less desirable than the Maximal Marginal Classifier is because the hyperplane, *in this scenario*, is favoring the blue data points, **with no "supporting" evidence (pun intended)**. It is giving them (the blue support vectors) a wider berth than is necessary, at the cost of the berth to the red support vectors.

```r
ggplot(df, aes(x_1, x_2, color = y)) +
    geom_point() +
    geom_abline(intercept = -0.25,
                slope = 1,
                linetype = 'solid') +
    geom_abline(intercept = -0,
                slope = 1,
                linetype = 'dashed') +
    geom_abline(intercept = -1,
                slope = 1,
                linetype = 'dashed') +
    scale_color_manual(values = c("Red" = 'red', "Blue" = 'blue')) +
    ggtitle("Maximal Marginal Classifier") +
    xlab("X1") +
    ylab("X2") +
    labs(color = 'Class')
```



- **H**. A data set where the Maximal Marginal Classifier does not exist.

```
new.point <- data.frame(x_1 = 2,
                        x_2 = 3,
                        y = "Blue")
df <- rbind(df, new.point)

ggplot(df, aes(x_1, x_2, color = y)) +
    geom_point() +
    geom_abline(intercept = -0.5,
                slope = 1,
                linetype = 'solid') +
    geom_abline(intercept = -0,
                slope = 1,
                linetype = 'dashed') +
    geom_abline(intercept = -1,
                slope = 1,
                linetype = 'dashed') +
    scale_color_manual(values = c("Red" = 'red', "Blue" = 'blue')) +
    ggtitle("Maximal Marginal Classifier") +
    xlab("X1") +
    ylab("X2") +
    labs(color = 'Class')
```


Maximal Marginal Classifier
X2
4
3
2
1
X1
1
2
3
4
Class
Blue
Red