Jared Yu
Module 6 Assignment
Data Mining

1. This problem involves hyperplanes in two dimensions.
   a. Sketch the hyperplane $1 + 3X_1 - X_2 = 0$. Indicate the set of points for which $1 + 3X_1 - X_2 > 0$, as well as the set of points for which $1 + 3X_1 - X_2 < 0$.

Ans:
The equation $1 + 3X_1 - X_2 = 0$ can be rewritten as $X_2 = 3X_1 + 1$. The inequality $1 + 3X_1 - X_2 > 0$ can likewise be rewritten as $X_2 < 3X_1 + 1$ and $1 + 3X_1 - X_2 < 0$ can be changed to $X_2 > 3X_1 + 1$. We can denote the first inequality as being in class Blue, and the second inequality being in class Red. This can be seen below in Figure 1, where blue dots represent areas of the graph where the class is Blue, and the red dots represent class Red. The hyperplane is the black line going diagonally through the graph.

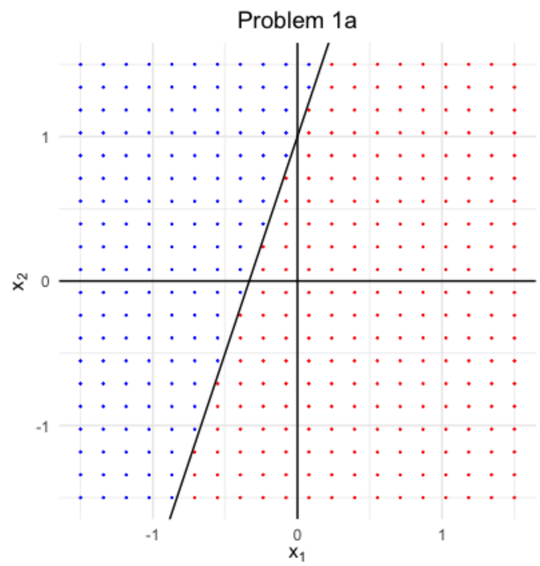


Figure 1 The above figure shows the sketch of the hyperplane in Problem 1 a). It separates the upper region into class Blue and the lower region into class Red as denoted by the colored points on either side of the line.

   b. On the same plot, sketch the hyperplane $-2 + X_1 + 2X_2 = 0$. Indicate the set of points for which $-2 + X_1 + 2X_2 > 0$, as well as the set of points for which $-2 + X_1 + 2X_2 < 0$.

Ans:
The equation $-2 + X_1 + 2X_2 = 0$ can be rewritten as $X_2 = -0.5X_1 + 1$. The inequality $-2 + X_1 + 2X_2 > 0$ can be rewritten as $X_2 > -0.5X_1 + 1$, and the inequality $-2 + X_1 + 2X_2 < 0$ can be rewritten as $X_2 < -0.5X_1 + 1$. We can denote the first inequality as being in class Purple, and the second inequality being in class Green. This can be seen below in Figure 2, it includes the same points as in part a) of the problem. Additionally, a second line has been added in red that represents this second hyperplane from part b). The upper region represents the class Purple and the lower region represents the class Green.

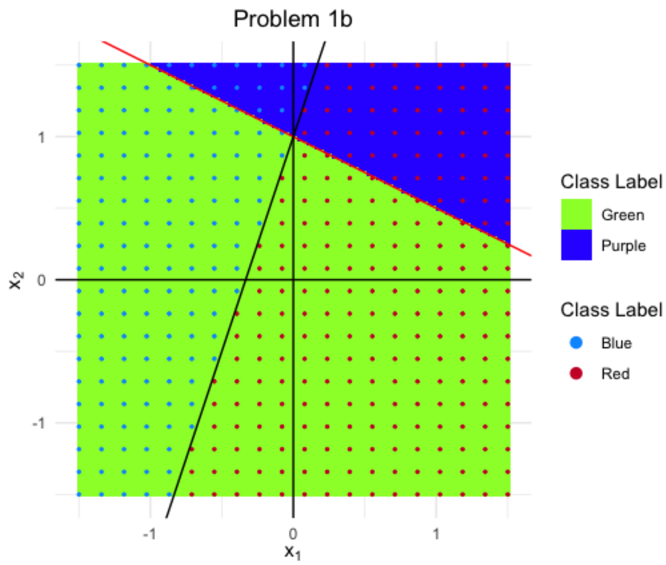


Figure 2 The above figure shows the updated plot for Problem 1 b). The points remain the same as before, while an additional filled region is used to help distinguish the two classes in part b). This is distinguished by the red line representing the hyperplane in part b). The upper region is class Purple and the lower region is class Green as denoted by the colors.

2. We have seen that in $p = 2$ dimensions, a linear decision boundary takes the form $\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$. We now investigate a non-linear decision boundary.
   a. Sketch the curve

$$(1 + X_1)^2 + (2 - X_2)^2 = 4.$$

Ans: (Reference: [1], [2])
The curve in part a) is actually that of a circle. It gives a circle with radius 2 with a center at $(-1, 2)$, given the formula, $(x - h)^2 + (y - k)^2 = r^2$. This circle is plotted below in Figure 3.

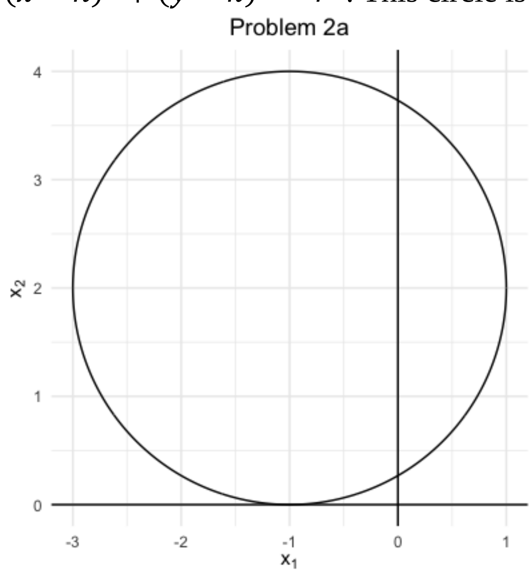


Figure 3 The above figure shows the circle that corresponds to the equation in part a).

   b. On your sketch, indicate the set of points for which

$$(1 + X_1)^2 + (2 - X_2)^2 > 4,$$

as well as the set of points for which

$$(1 + X_1)^2 + (2 - X_2)^2 \leq 4.$$

<u>Ans:</u> (Reference: [3])

The first inequality, $(1 + X_1)^2 + (2 - X_2)^2 > 4$, is an inequality for points outside of the circle. While the second inequality, $(1 + X_1)^2 + (2 - X_2)^2 \leq 4$, indicates points on or within the circle. We can denote the points outside the circle belonging to class Blue, and the points inside or on the circle belonging to class Red. This can be seen below in Figure 4.

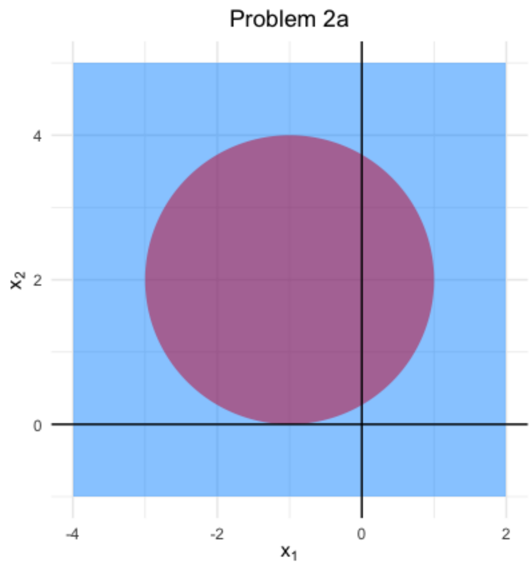


Figure 4 The above figure shows the color coded version of the previous Figure 3 plot. The area on or within the circle are colored red and belong to class Red. The area outside the circle is colored blue and belongs to class Blue.

c. Suppose that a classifier assigns an observation to the blue class if

$$(1 + X_1)^2 + (2 - X_2)^2 > 4,$$

and to the red class otherwise. To what class is the observation (0,0) classified? (-1,1)? (2,2)? (3,8)?

<u>Ans:</u>

The observation (0,0) can be seen to lie outside the circle and so it belongs to the blue class. The observation (-1,1) can be seen within the red circle so it belongs to the red class. The observation (3,8) is far outside the red circle, so it belongs to the blue class.

d. Argue that while the decision boundary in (c) is not linear in terms of X_1 and X_2, it is linear in terms of X_1, X_1^2, X_2, and X_2^2.

<u>Ans:</u> (Reference: [1], [4], [5])

Looking at $(1 + X_1)^2 + (2 - X_2)^2 > 4$, we can see from previous parts that after plotting it is clearly not a line, but rather a circle. Since it is not a straight line, it is a nonlinear inequality. We can further expand $(1 + X_1)^2 + (2 - X_2)^2 > 4$ as follows,

$$(1 + X_1)^2 + (2 - X_2)^2 > 4$$
$$\rightarrow 1 + X_1 + X_1^2 + 4 - 2X_2 + X_2^2 > 4$$
$$\rightarrow X_1^2 + X_2^2 + X_1 - 2X_2 + 5 > 4$$
$$\rightarrow X_1^2 + X_2^2 + X_1 - 2X_2 > -1.$$

From this, it is evident that this new inequality which is the expanded form of the previous one is linear in terms of X_1, X_1^2, X_2, and X_2^2. The reason is that it fits the format of a linear inequality (i.e., $a_1x_1 + a_2x_2 + \cdots + a_nx_n > b$) now. This would be more obvious if for example we renamed X_1^2 and X_2^2 to X_3 and X_4 respectively.

3. Here we explore the maximal margin classifier on a toy data set.
    a. We are given $n = 7$ observations in $p = 2$ dimensions. For each observation, there is an associated class label.
    Sketch the observations.

Ans:
The data is plotted below in Figure 5. The points associated with class Red are colored in red and those associated with class Blue are colored in blue.

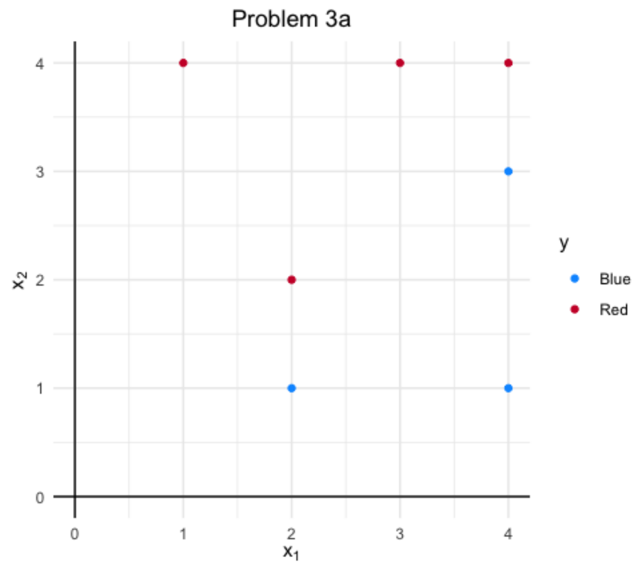


Figure 5 The above figure shows the plotted points for the data in Problem 3. Points with the color red are associated with class Red and blue points are associated with class Blue.

    b. Sketch the optimal separating hyperplane, and provide the equations for this hyperplane (of the form (9.1)).

Ans: (Reference [6], [7])
Looking at Figure 5, it is clear that the support vectors for the Red class would be (2, 2) and (4, 4), while the support vectors for the Blue class would be (2, 1) and (4, 1). The textbook states that the optimal separating hyperplane in this case is the line that is equidistant from these support vectors. The points equidistant from these pairs of points are (2, 1.5) and (4, 3.5). Plugging these into the equation for a line between two points, we find

$$m = \frac{3.5 - 1.5}{4 - 2} = \frac{2}{2} = 1$$

$$y = x + b$$

$$b = y - x = 3.5 - 4 = -0.5$$

Therefore, it follows that the equation for the line separating these two classes is $y = x - 0.5$. This is plotted below in Figure 6.

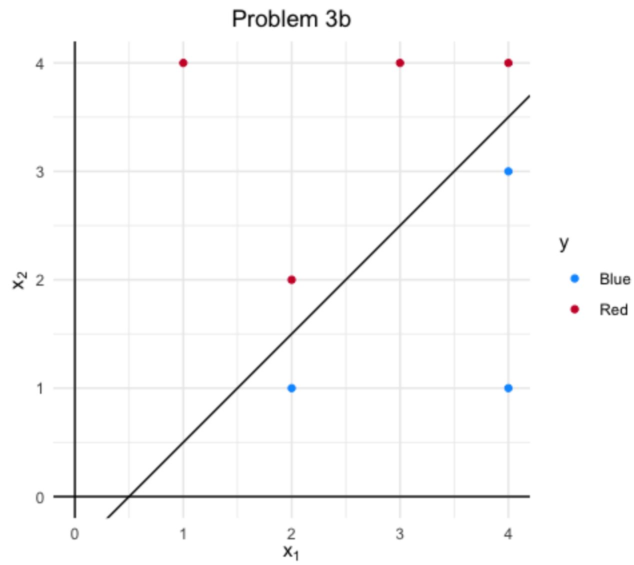


Figure 6 The above figure shows the same points from part a), but the hyperplane from the maximal margin classifier is also shown separating the two colors.

c. Describe the classification rule for the maximal margin classifier. It should be something along the lines of "Classify to Red if $\beta_0 + \beta_1 X_1 + \beta_2 X_2 > 0$, and classify to Blue otherwise." Provide the values for β_0, β_1, and β_2.

Ans:

The maximal margin classifier in part b) can be stated as follows,

"Classify to Red if $0.5 - X_1 + X_2 > 0$, and classify to Blue otherwise."

d. On your sketch, indicate the margin for the maximal margin hyperplane.

Ans:

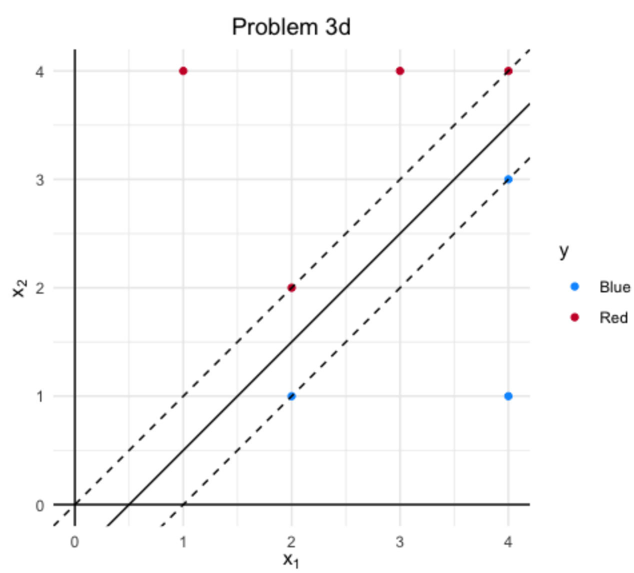


Figure 7

e. Indicate the support vectors for the maximal margin classifier.

Ans:

Below in Figure 8 is the same plot from part d). The difference is that some green lines are added to indicate the support vectors which lie along the margin and are all equidistance from the hyperplane separating the two classes.

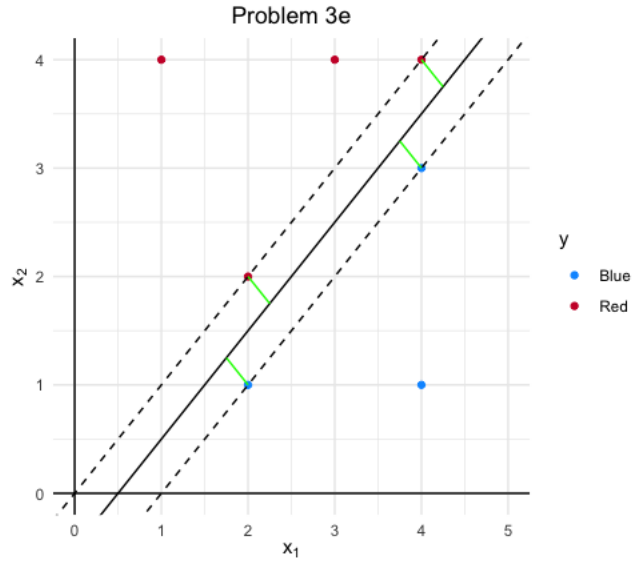


Figure 8 The above figure shows the same plot from Figure 7, except some green lines are added to identify the four support vectors. These are the points that lie on the margin and are equidistant from the hyperplane.

f. Argue that a slight movement of the seventh observation would not affect the maximal margin hyperplane.

Ans:

The seventh observation is located at $(x_1 = 4, x_2 = 1)$ and is classified as Blue. Looking back to Figure 8, it can be seen that this is one of the three Blue observations. Out of the three, it is the only one that is not a support vector for the hyperplane. Furthermore, it is quite a distance away from the margin itself.

g. Sketch a hyperplane that is *not* the optimal separating hyperplane, and provide the equation for this hyperplane.

Ans:

The following hyperplane has the equation,

$$X_2 = -0.5X_1 + 3.5.$$

It can be seen below in Figure 9. Looking at the hyperplane, it is clearly not optimal in terms of being a maximal margin classifier. It misclassifies two points, where the red point at (2, 2) and the blue point at (4, 3) are misclassified according to this hyperplane.

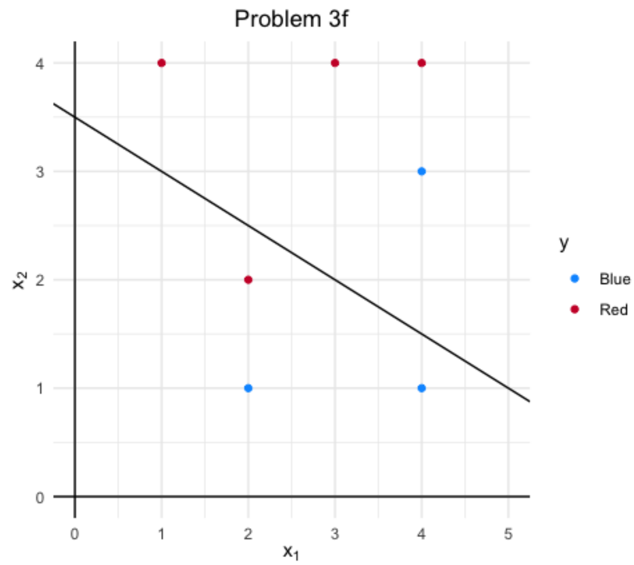


Figure 9 The above figure shows a non-optimal hyperplane, where the equation of the line is stated above.

h. Draw an additional observation on the plot so that the two classes are no longer separable by a hyperplane.

<u>Ans:</u>

A new Blue observation is added at (2,3). It can be seen below in Figure 10. It is located on the other side of the classifier for the Blue class. It is also beyond the margin itself. So, it can be seen now that the two classes are no longer separated by a hyperplane.

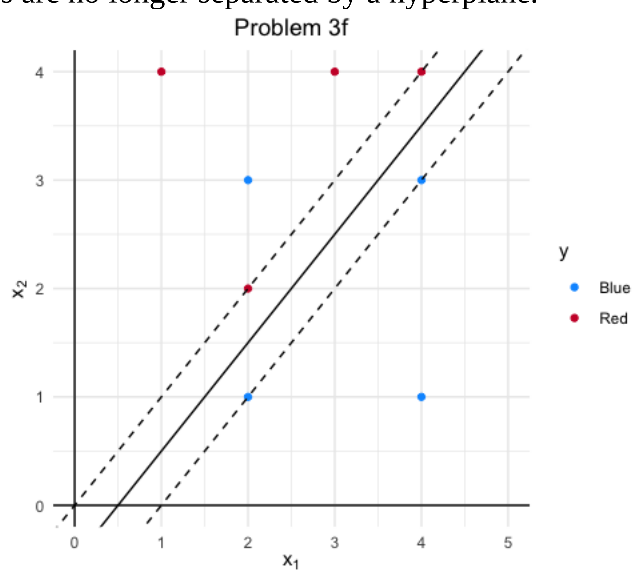


Figure 10 The above figure shows the new observation at (2,3). It belongs to class Blue and violates the maximum margin classifier.

4. Generate a simulated two-class data set with 100 observations and two features in which there is a visible but non-linear separation between the two classes. Show that in this setting, a support vector machine with a polynomial kernel (with degree greater than 1) or

a radial kernel will outperform a support vector classifier on the training data. Which technique performs best on the test data? Make plots and report training and test error rates in order to back up your assertions.

Ans: (Reference: [1], [8])

To generate the data, the following two equations were used

$$X_2 = 1.5X_1^2 \pm (0.5 + Z).$$

In the case of the $\pm$ sign being $+$, it corresponds with class 1 and class 2 in the case of the sign being $-$. The Z term is some random variable with a standard normal distribution to help give some random scatter to the pattern. The X_1 term is also drawn from a standard normal distribution. Both classes have 50 observations each. A plot of the points can be seen below in Figure 11. There is a clear non-linear separation between the two classes.

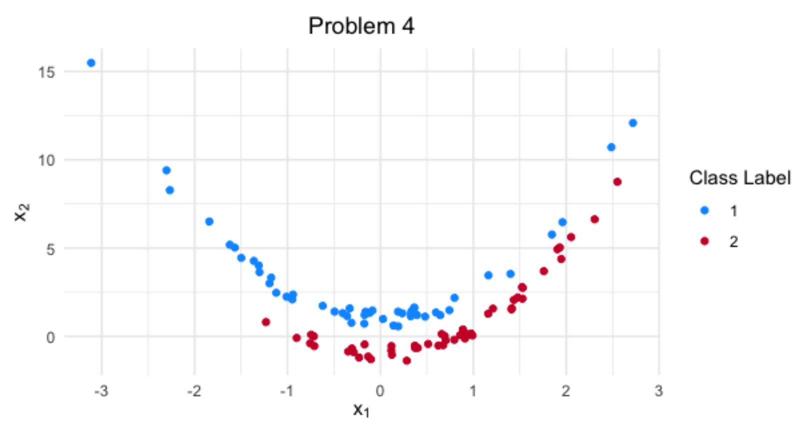
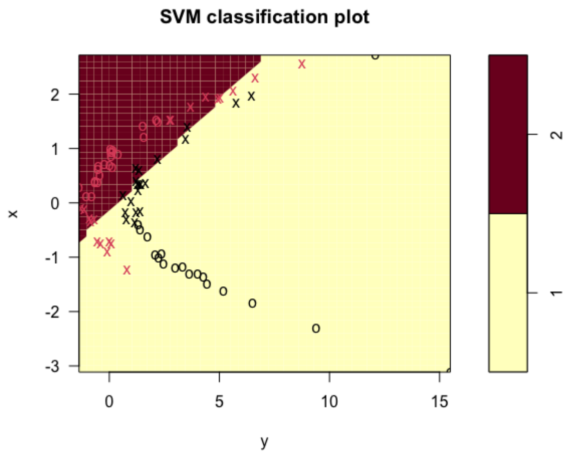


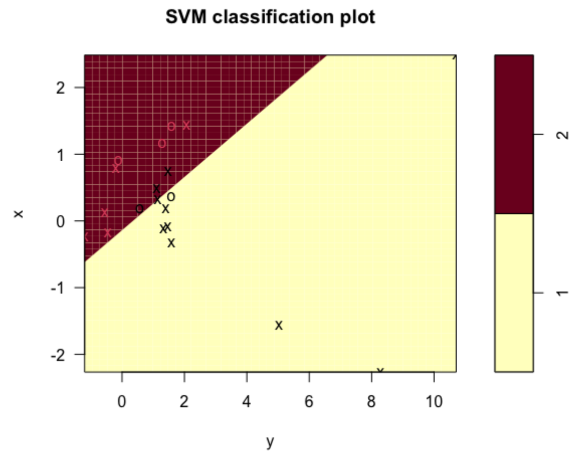
Figure 11 The above figure shows the generated dataset for the two classes. There is a clear non-linear separation between the two classes. Class 1 is colored blue and class 2 is colored red.

After the data has been generated, a simple 80/20 split was done, where 80 observations went to the training set and 20 observations went to the testing set. These observations were randomly sampled from the total. Using the training examples, they were all used to train a set of three different support vector machine (SVM) models using the 'e1071' package in R. The three models trained used a linear, polynomial, and radial kernel respectively. They were all tuned with 'cost=10' and in the case of the radial kernel, there is the additional 'gamma=1' parameter.

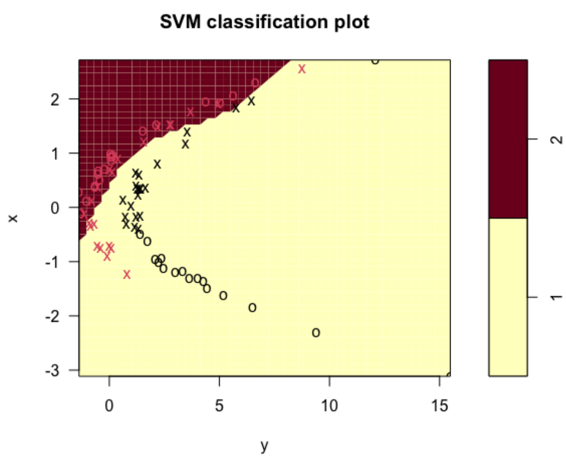
Below in Table 1, the plots for each of the different SVM models can be seen. From top to bottom are the linear, polynomial, and radial kernel respectively. The left column shows the training data, while the right column shows the test data. It can be seen that the polynomial and radial kernel are capable of fitting the non-linear separation between the two classes, while the linear kernel simply fits a straight line.



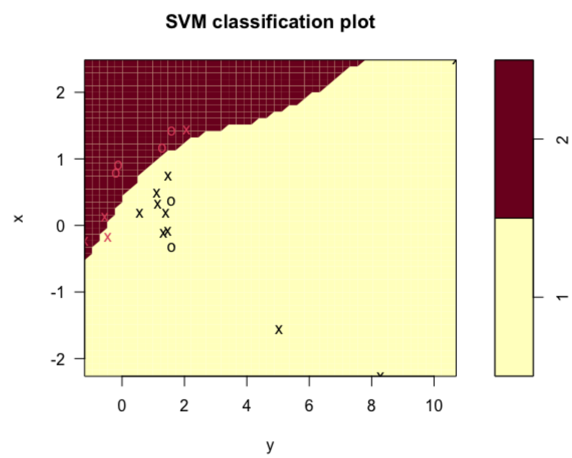
(a)



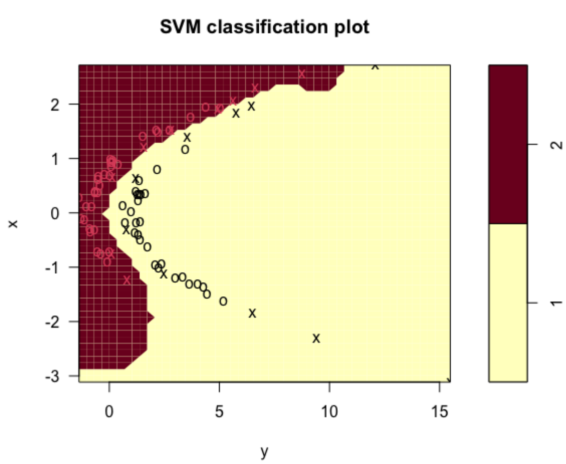
(b)



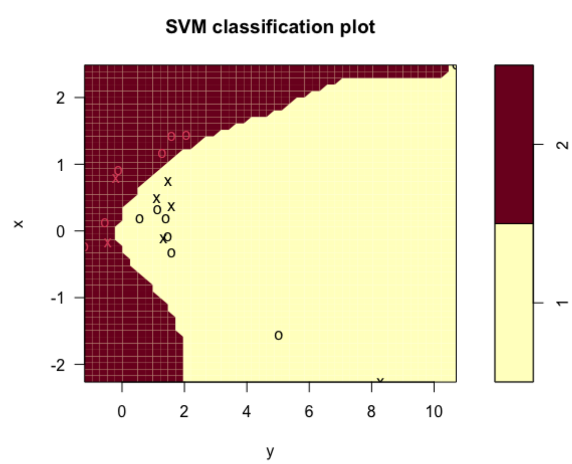
(c)



(d)



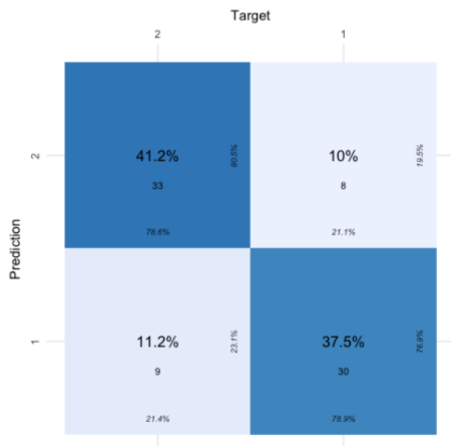
(e)



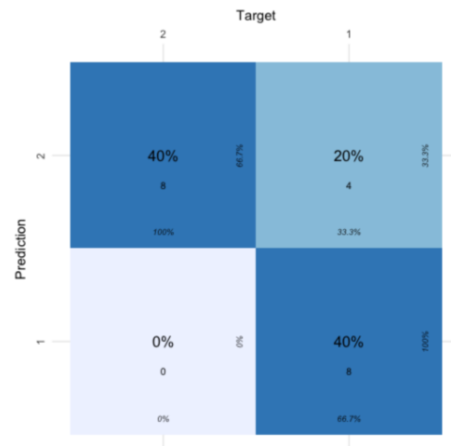
(f)

Table 1 The above table shows the plots of the SVM models based on the linear, polynomial, and radial kernel. From top to bottom are the linear, polynomial, and radial kernels respectively. The left column shows the plots on the training data, while the right column shows the plots on the testing data.

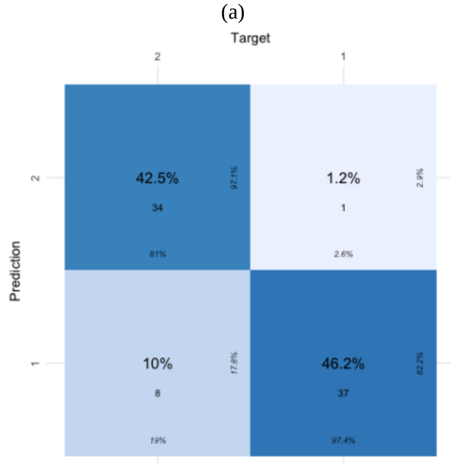
After fitting the model on the training data and testing it on the testing set, some confusion matrices were generated for the training and testing data. The confusion matrices can be seen below in Table 2. The table shows from top to bottom the confusion matrices for the linear, polynomial, and radial kernels respectively. The left column and right columns show the confusion matrices for the training and testing data respectively.



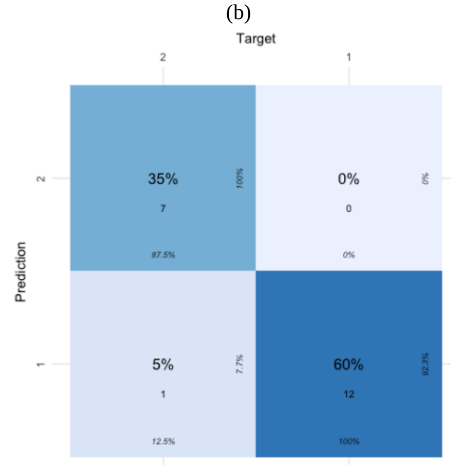
(a)



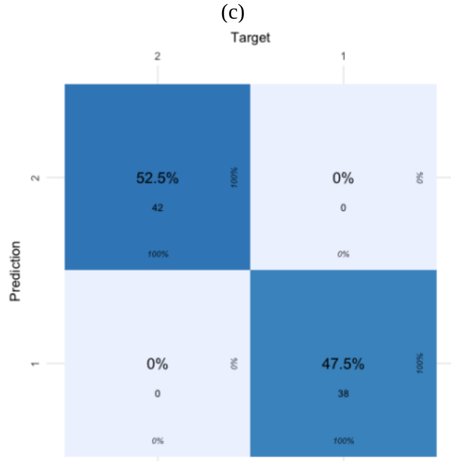
(b)



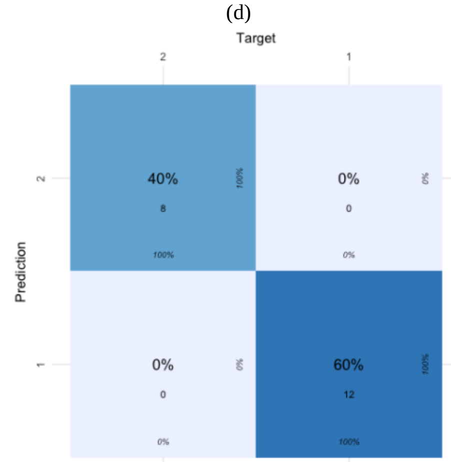
(c)



(d)



(e)



(f)

Table 2 The above table shows the confusion matrices from top to bottom for the linear, polynomial, and radial kernels respectively. The left column shows the plots on the training data, while the right column shows the plots on the testing data.

It can be seen that the best performance across training and testing data is with the radial kernel, where there are zero errors. In second place is the polynomial kernel with 9 errors in the training set and 1 error in the testing set. In last place is the linear kernel with 17 errors in the training set and 4 errors in the testing set. The error rates are shown together below in Table 3. The error rate is calculated to be the percentage of misclassified observations out of the total training and testing dataset sizes (i.e., 80 and 20).

	Training Error Rate	Testing Error Rate
Linear Kernel	21.2%	20%
Polynomial Kernel	11.2%	5%
Radial Kernel	0%	0%

Table 3 The above table shows the error rate for the training (left) and testing set (right). The rows indicate from top to bottom the linear, polynomial, and radial kernels.

Reference:

[1] https://blog.princehonest.com/stat-learning/

[2] https://www.purplemath.com/modules/sqrcircle.htm

[3] https://www.varsitytutors.com/algebra_ii-help/graphing-circular-inequalities

[4] https://sciencing.com/identify-linear-nonlinear-equations-5895035.html

[5] https://en.wikipedia.org/wiki/Linear_inequality

[6] https://piazza.com/class/kc0jkwru805u1?cid=130

[7] https://www.mesacc.edu/~scotz47781/mat150/notes/eqn_line/Equation_Line_Two_Points_Notes.pdf

[8] https://cran.r-project.org/web/packages/cvms/vignettes/creating_a_confusion_matrix.html

Code Appendix

```
library(tidyverse); library(e1071); library(cvms); library(tibble); library(broom)

### Problem 1a
q1a <- function(x1, x2) {
  return(1 + 3*x1 - x2)
}

x1s <- seq(-1.5, 1.5, length.out = 1e2)
x2s <- seq(-1.5, 1.5, length.out = 1e2)

# Reference: https://selbydavid.com/2018/01/09/neural-network/
decision_grid <- expand.grid(x1 = x1s, x2 = x2s)
decision_grid_mat <- data.matrix(decision_grid[,c('x1','x2')])
q1a_grid_output <- mapply(q1a, decision_grid_mat[,1], decision_grid_mat[,2])
```

```r
decision_grid$class_label <- factor(x = q1a_grid_output > 0, labels = c('Blue', 'Red'))

# Reference: https://stackoverflow.com/questions/40675778/center-plot-title-in-ggplot2
theme_update(plot.title = element_text(hjust = 0.5))

# Reference: https://stackoverflow.com/questions/43129280/color-points-with-the-color-as-a-column-in-
ggplot2
p1 <- ggplot(data = decision_grid, aes(x = x1, y = x2)) +
 geom_point(colour = decision_grid$class_label, size = 0.1) +
 # Reference: http://www.sthda.com/english/wiki/ggplot2-add-straight-lines-to-a-plot-horizontal-vertical-
and-regression-lines
 geom_abline(intercept = 1, slope = 3) +
 geom_hline(yintercept = 0) + geom_vline(xintercept = 0) +
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 1a') +
 theme_minimal() + theme(aspect.ratio = 1, plot.title = element_text(hjust = 0.5))

### Problem 1b
x1s <- seq(-1.5, 1.5, length.out = 1e2)
x2s <- seq(-1.5, 1.5, length.out = 1e2)
q1b <- function(x1, x2) {
  return(-2 + x1 + 2*x2)
}
decision_gridb <- expand.grid(x1 = x1s, x2 = x2s)
decision_grid_matb <- data.matrix(decision_gridb[,c('x1','x2')])
q1b_grid_output <- mapply(q1b, decision_grid_matb[,1], decision_grid_matb[,2])
decision_gridb$class_label <- factor(x = q1b_grid_output > 0, labels = c('Green', 'Purple'))

# Shrink the # of points in the geom_point
x1s <- seq(-1.5, 1.5, length.out = 20)
x2s <- seq(-1.5, 1.5, length.out = 20)
decision_grid <- expand.grid(x1 = x1s, x2 = x2s)
decision_grid_mat <- data.matrix(decision_grid[,c('x1','x2')])
q1a_grid_output <- mapply(q1a, decision_grid_mat[,1], decision_grid_mat[,2])
decision_grid$class_label <- factor(x = q1a_grid_output > 0, labels = c('Blue', 'Red'))

p2 <- ggplot(data = decision_grid, aes(x = x1, y = x2)) +
 # Reference: http://www.math-evry.cnrs.fr/_media/members/cambroise/teaching/tp_r_part1_corrected.pdf
 geom_tile(data = decision_gridb, aes(fill = class_label)) +
 # Reference: http://www.cookbook-r.com/Graphs/Colors_(ggplot2)/
 # Reference: https://stackoverflow.com/questions/25195869/how-to-change-color-palette-of-geom-tile-in-r-
ggplot2
 # Reference: https://stackoverflow.com/questions/25176399/scale-fill-discrete-and-scale-fill-manual-
legend-options-confusion
 scale_fill_manual(values=c("#99FF33", "#3300FF"), name = 'Class Label') +
 # Reference: https://stackoverflow.com/questions/47023781/how-to-add-a-legend-for-two-geom-layers-in-one-
ggplot2-plot
 geom_point(aes(colour = class_label), size = 0.5) +
 scale_color_manual(values=c("#0099FF", "#CC0033"), name = 'Class Label') +
 geom_abline(intercept = 1, slope = 3) +
 geom_abline(intercept = 1, slope = -0.5, color = 'red') +
 geom_hline(yintercept = 0) + geom_vline(xintercept = 0) +
 theme_minimal() + theme(aspect.ratio = 1, plot.title = element_text(hjust = 0.5)) +
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 1b') +
 guides(shape = FALSE,
          colour = guide_legend(override.aes = list(
            fill = c("#99FF33", "#3300FF"),
            size = c(3, 3),
            shape = c(16, 16))))

### Problem 2
### part (a)
# Reference: https://stackoverflow.com/questions/6862742/draw-a-circle-with-ggplot2
plot_circle <- function(center = c(0,0), diameter = 1, npoints = 1e3){
  r = diameter / 2
  tt <- seq(0,2*pi,length.out = npoints)
  xx <- center[1] + r * cos(tt)
  yy <- center[2] + r * sin(tt)
```

```r
  return(data.frame(x = xx, y = yy))
}
circle_data <- plot_circle(center = c(-1, 2), diameter = 2*2, npoints = 1e3)

p3 <- ggplot(circle_data, aes(x,y)) +
 geom_path() +
 # Reference: https://stackoverflow.com/questions/21294196/how-do-i-make-my-facets-perfectly-square
 theme(aspect.ratio = 1) +
 geom_hline(yintercept = 0) + geom_vline(xintercept = 0) +
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 2a') +
 theme_minimal() + theme(aspect.ratio = 1, plot.title = element_text(hjust = 0.5))

### part (b)
# (-4,5), (2,5), (-4,-1), (1,-1)
fill_area <- data.frame(x1 = c(-4), x2 = c(2),
                        y1 = c(-1), y2 = c(5), class_label = c('Blue'))
circle_data$class_label <- 'Red'

p4 <- ggplot() +
 # Reference: http://sape.inf.usi.ch/quick-reference/ggplot2/geom_rect
 # Reference: https://stackoverflow.com/questions/50343911/remove-border-from-geom-rect-using-ggplot2
 # Reference: https://stackoverflow.com/questions/31599146/ggplot2-change-geom-rect-colour-in-a-stacked-
barplot
 geom_rect(data = fill_area,
           mapping = aes(xmin = x1, xmax = x2, ymin = y1, ymax = y2), fill = '#0099FF',
           color = NA, alpha = 0.5) +
 geom_polygon(data = circle_data, mapping = aes(x, y), fill = "#CC0033", alpha = 0.5) +
 # Reference: https://stackoverflow.com/questions/21294196/how-do-i-make-my-facets-perfectly-square
 geom_hline(yintercept = 0) + geom_vline(xintercept = 0) +
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 2a') +
 # Reference: https://stackoverflow.com/questions/45346885/center-plot-title-in-ggplot2-using-theme-bw
 theme_minimal() + theme(aspect.ratio = 1, plot.title = element_text(hjust = 0.5)) +
 scale_colour_manual(name = 'the colour',
                     values =c('#0099FF'='black','#CC0033'='red'), labels = c('c2','c1'))

### Problem 3
### part (a)
df <- data.frame(x1 = c(3,2,4,1,2,4,4),
                 x2 = c(4,2,4,4,1,3,1),
                 y = rep(c('Red', 'Blue'), each = 4)[-8])

p5 <- ggplot(data = df, mapping = aes(x = x1, y = x2, colour = y)) +
 geom_hline(yintercept = 0) + geom_vline(xintercept = 0) +
 geom_point(aes(colour = y), size = 1.5) +
 scale_color_manual(values=c("#0099FF", "#CC0033"), name = expression(y)) +
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 3a') +
 theme_minimal() + theme(aspect.ratio = 1, plot.title = element_text(hjust = 0.5))

### part (b)
p6 <- ggplot(data = df, mapping = aes(x = x1, y = x2, colour = y)) +
 geom_hline(yintercept = 0) + geom_vline(xintercept = 0) +
 geom_point(aes(colour = y), size = 1.5) +
 geom_abline(slope = 1, intercept = -0.5) +
 scale_color_manual(values=c("#0099FF", "#CC0033"), name = expression(y)) +
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 3b') +
 theme_minimal() + theme(aspect.ratio = 1, plot.title = element_text(hjust = 0.5))

### part (d)
p7 <- ggplot(data = df, mapping = aes(x = x1, y = x2, colour = y)) +
 geom_hline(yintercept = 0) + geom_vline(xintercept = 0) +
 geom_point(aes(colour = y), size = 1.5) +
 geom_abline(slope = 1, intercept = -0.5) +
 geom_abline(slope = 1, intercept = -1, linetype = 'dashed') +
 geom_abline(slope = 1, intercept = 0, linetype = 'dashed') +
 scale_color_manual(values=c("#0099FF", "#CC0033"), name = expression(y)) +
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 3d') +
```

```r
 theme_minimal() + theme(aspect.ratio = 1, plot.title = element_text(hjust = 0.5))

### part (e)
# y = x - 0.5
# y - y1 = (x - x1)
# (2,2)
# y - 2 = x - 2
# y = x
d=data.frame(x=c(1,2,5,6,8), y=c(3,6,2,8,7), vx=c(1,1.5,0.8,0.5,1.3), vy=c(0.2,1.3,1.7,0.8,1.4))

margin_points <- data.frame(
  x = c(2, 2, 4, 4),
  y = c(2, 1, 4, 3),
  vx = c(0.25, -0.25, 0.25, -0.25),
  vy = c(-0.25, 0.25, -0.25, 0.25)
)

p8 <- ggplot(data = df, mapping = aes(x = x1, y = x2, colour = y)) +
 geom_hline(yintercept = 0) + geom_vline(xintercept = 0) +
 geom_point(aes(colour = y), size = 1.5) +
 geom_abline(slope = 1, intercept = -0.5) +
 geom_abline(slope = 1, intercept = -1, linetype = 'dashed') +
 geom_abline(slope = 1, intercept = 0, linetype = 'dashed') +
 # Reference: http://sape.inf.usi.ch/quick-reference/ggplot2/geom_segment
 geom_segment(data=margin_points, mapping=aes(x=x, y=y, xend=x+vx, yend=y+vy), size=0.5, color="green") +
 scale_color_manual(values=c("#0099FF", "#CC0033"), name = expression(y)) +
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 3e') +
 xlim(0, 5) + theme_minimal() +
 theme(aspect.ratio = 1, plot.title = element_text(hjust = 0.5))

### part (g)
p9 <- ggplot(data = df, mapping = aes(x = x1, y = x2, colour = y)) +
 geom_hline(yintercept = 0) + geom_vline(xintercept = 0) +
 geom_point(aes(colour = y), size = 1.5) +
 geom_abline(slope = -0.5, intercept = 3.5) +
 scale_color_manual(values=c("#0099FF", "#CC0033"), name = expression(y)) +
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 3f') +
 xlim(0, 5) + theme_minimal() +
 theme(aspect.ratio = 1, plot.title = element_text(hjust = 0.5))

### part (h)
df2 <- data.frame(x1 = c(3,2,4,1,2,4,4,2),
                  x2 = c(4,2,4,4,1,3,1,3),
                  y = rep(c('Red', 'Blue'), each = 4))

p10 <- ggplot(data = df2, mapping = aes(x = x1, y = x2, colour = y)) +
 geom_hline(yintercept = 0) + geom_vline(xintercept = 0) +
 geom_point(aes(colour = y), size = 1.5) +
 geom_abline(slope = 1, intercept = -0.5) +
 geom_abline(slope = 1, intercept = -1, linetype = 'dashed') +
 geom_abline(slope = 1, intercept = 0, linetype = 'dashed') +
 scale_color_manual(values=c("#0099FF", "#CC0033"), name = expression(y)) +
 labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 3f') +
 xlim(0, 5) + theme_minimal() +
 theme(aspect.ratio = 1, plot.title = element_text(hjust = 0.5))

### Problem 4
set.seed(111)
x1_class1 <- rnorm(50)
x1_class2 <- rnorm(50, mean = 0.5)
x2_class1 <- 1.5 * x1_class1^(2) + 0.5 + runif(50)
x2_class2 <- 1.5 * x1_class2^(2) + - 0.5 - runif(50)

df <- data.frame(x = c(x1_class1, x1_class2),
                 y = c(x2_class1, x2_class2),
                 z = rep(c(1,2), each = 50))
```

```r
p11 <- ggplot(df, aes(x, y, color = factor(z))) +
  geom_point() +
  scale_color_manual(values=c("#0099FF", "#CC0033"), name = 'Class Label') +
  theme_minimal() +
  theme(aspect.ratio = 0.55, plot.title = element_text(hjust = 0.5)) +
  labs(x = expression(x[1]), y = expression(x[2]), title = 'Problem 4')

set.seed(666)
train_idx <- sample(seq(1,1e2), 80)

data.train <- data.frame(x = df[train_idx,c('x')],
                         y = df[train_idx,c('y')],
                         z = as.factor(df$z[train_idx]))

data.test <- data.frame(x = df[-train_idx,'x'],
                        y = df[-train_idx,'y'],
                        z = as.factor(df$z[-train_idx]))

svm.linear <- svm(z~., data=data.train, kernel="linear", cost=10)
plot(svm.linear, data.train)
table(df$z[train_idx], predict(svm.linear, data.train))

svm.poly = svm(z~., data=data.train, kernel="polynomial", cost=10)
plot(svm.poly, data.train)
table(df$z[train_idx], predict(svm.poly, data.train))

svm.radial = svm(z~., data=data.train, kernel="radial", gamma=1, cost=10)
plot(svm.radial, data.train)
table(df$z[train_idx], predict(svm.radial, data.train))

plot(svm.linear, data.test)
plot(svm.poly, data.test)
plot(svm.radial, data.test)
table(df$z[-train_idx], predict(svm.linear, data.test))
table(df$z[-train_idx], predict(svm.poly, data.test))
table(df$z[-train_idx], predict(svm.radial, data.test))

linear_cm_train <- tidy(table(tibble("target"=df$z[train_idx],
                                      "prediction"=predict(svm.linear, data.train))))
plot_confusion_matrix(linear_cm_train,
                      targets_col = "target",
                      predictions_col = "prediction",
                      counts_col = "n")

poly_cm_train <- tidy(table(tibble("target"=df$z[train_idx],
                                    "prediction"=predict(svm.poly, data.train))))
plot_confusion_matrix(poly_cm_train,
                      targets_col = "target",
                      predictions_col = "prediction",
                      counts_col = "n")

radial_cm_train <- tidy(table(tibble("target"=df$z[train_idx],
                                      "prediction"=predict(svm.radial, data.train))))
plot_confusion_matrix(radial_cm_train,
                      targets_col = "target",
                      predictions_col = "prediction",
                      counts_col = "n")


linear_cm <- tidy(table(tibble("target"=df$z[-train_idx],
                                "prediction"=predict(svm.linear, data.test))))
plot_confusion_matrix(linear_cm,
                      targets_col = "target",
                      predictions_col = "prediction",
                      counts_col = "n")
```

```r
poly_cm <- tidy(table(tibble("target"=df$z[-train_idx],
                              "prediction"=predict(svm.poly, data.test))))
plot_confusion_matrix(poly_cm,
                      targets_col = "target",
                      predictions_col = "prediction",
                      counts_col = "n")

radial_cm <- tidy(table(tibble("target"=df$z[-train_idx],
                                "prediction"=predict(svm.radial, data.test))))
plot_confusion_matrix(radial_cm,
                      targets_col = "target",
                      predictions_col = "prediction",
                      counts_col = "n")
```