**Homework – 4**

**Question 1**

**Chart, scatter chart

Description automatically generated**

Consider a situation where a number of identical fruits are given different amounts of fertilizer (X1) and water(X2) and their growth was measured in some time-frame as following, this growth is captured by categorical target Y, which takes the value Yes if fruits have ripened > 1week, else No.

Below is the decision tree

X1 < 56





X2 < 50 R1

X2 < 40 R2



X2 < 42 X2 < 26



R5 R6 R3 R4

**Question 3**

# Generate sequences of values for m1 and m2

m1 <- seq(0 + 1e-06, 1 - 1e-06, length.out = 100)

m2 <- 1 - m1

# Calculate classification error, gini index, and entropy

class\_error <- 1 - apply(rbind(m1, m2), 2, max)

gini <- m1 \* (1 - m1) + m2 \* (1 - m2)

entropy <- -(m1 \* log(m1) + m2 \* log(m2))

# Plot the three metrics on the same graph

plot(m1, class\_error, type = "l", col = "black", xlab = "Proportion of Samples in Class 1 (m1)",

ylab = "Error Metrics", ylim = c(0, max(class\_error, gini, entropy)),

main = "Performance Metrics for Binary Classification")

lines(m1, gini, col = "blue")

lines(m1, entropy, col = "red")

# Add a legend to the graph

legend("topright", c("Classification Error", "Gini Index", "Entropy"),

col = c("black", "blue", "red"), lty = c(1, 1))

# Add a grid to the graph

grid()

# Save the graph to a file

png("performance\_metrics.png", width = 800, height = 600, units = "px", res = 300)

dev.off()

Chart

Description automatically generated

**Question 4**

X1 < = 1



X2 <= 1 5



X1 < = 0 15



3 X2 <= 0

10 0



B)

Chart

Description automatically generated with low confidence

**Question 5**

Given we have two classes Red and Green

10 estimates of P(Red| X) are

0.1, 0.15, 0.2, 0.2, 0.55, 0.6,0.6, 0.65, 0.7, 0.75

To combine results there are two approaches we can choose

1. **Majority Vote:** Out of 10 estimates 6 estimates have p > 0.5 which means the majority of the estimates classify X as Red.
2. **Average Probablity :**

Consider the average of 10 estimates

P = ( 0.1+0.15+0.2+0.2+0.55+0.6+0.6+ 0.65+ 0.7+ 0.75)/ 10 = 0.45

Since p <0.5

Class for X will be Green.

**Chapter 9**

**Exercises 1,2,3**

**1Ans.**

Code for plotting the line

X1 <- -10:10

X2 <- 3\*X1 + 1

df1 <- data.frame(X1, X2, line = "a")

grid <- expand.grid(X1 = seq(min(df1$X1), max(df1$X1), length.out = 200),

X2 = seq(min(df1$X2), max(df1$X2), length.out = 200))

mutate(region = case\_when(1 + 3\*X1 - X2 < 0 ~ "lt 0",

1 + 3\*X1 - X2 > 0 ~ "gt 0"))

ggplot(grid, aes(x = X1, y = X2)) +

geom\_tile(aes(fill = region), alpha = 0.5) +

geom\_line(data = df1, aes(x = X1, y = X2), size = 1, col = "#7CAE00") +

annotate("text", x = -5, y = 10, label = expression(1 + 3\*X[1] - X[2] < 0),

hjust = 0.5, size = 4, col = "blue") +

annotate("text", x = 5, y = -10, label = expression(1 + 3\*X[1] - X[2] > 0),

hjust = 0.5, size = 4, col = "red") +

scale\_fill\_manual(values = c("#F8766D", "#00BFC4")) +

scale\_x\_continuous(expand = c(0.01,0.01), breaks = seq(-10, 10, 2)) +

scale\_y\_continuous(expand = c(0.01,0.01), breaks = seq(-30, 30, 10)) +

theme(legend.position = "none") +

labs(title = expression(paste("Hyperplane Plot: ", 1 + 3\*X[1] - X[2] == 0)),

x = expression(paste("(", X[1], ")")),

y = expression(paste("(", X[2], ")")))

Chart

Description automatically generated

b) hyperplane sketch 2

A. Plot the line -2 + X1 + 2X2 = 0

X2 = 1 – ½(X1)

-2+X1+2X2>0⟺X2>1−1/2X1

* −2+X1+2X2<0⟺X2<1−1/2X1
* X1 <- -10:10
* X2 <- 1 - 0.5\*X1
* df2 <- data.frame(X1, X2, line = "b")
* grid <- grid %>%
* mutate(region = case\_when(-2 + X1 + 2\*X2 < 0 ~ "lt 0",
* -2 + X1 + 2\*X2 > 0 ~ "gt 0"))
* ggplot(grid, aes(x = X1, y = X2)) +
* geom\_tile(aes(fill = region), alpha = 0.5) +
* geom\_line(data = df2, aes(x = X1, y = X2), size = 1, col = "#C77CFF") +
* annotate("text", x = 2.5, y = 15, label = TeX("$-2 + X\_1 + 2X\_2 > 0$", output = "character"),
* hjust = 0.5, size = 4, parse = TRUE, col = "blue") +
* annotate("text", x = -2.5, y = -15, label = TeX("$-2 + X\_1 + 2X\_2 < 0$", output = "character"),
* hjust = 0.5, size = 4, parse = TRUE, col = "red") +
* scale\_fill\_manual(values = c("#00BFC4", "#F8766D")) +
* scale\_x\_continuous(expand = c(0.01,0.01), breaks = seq(-10, 10, 2)) +
* scale\_y\_continuous(expand = c(0.01,0.01), breaks = seq(-30, 30, 10))
* theme(legend.position = "none")

Chart

Description automatically generated

The following R code deals with it :

bind\_rows(df1, df2) %>%

ggplot(aes(x = X1, y = X2, col = line)) +

geom\_line(size = 1) +

scale\_color\_manual(values = c("#7CAE00", "#C77CFF"),

labels = unname(TeX(c("$1 + 3X\_1 − X\_2 = 0", "$−2 + X\_1 + 2X\_2 = 0")))) +

scale\_x\_continuous(breaks = seq(-10, 10, 2)) +

scale\_y\_continuous(breaks = seq(-30, 30, 10)) +

labs(x = TeX(r'($X\_1$)'),

y = TeX(r'($X\_2$)'),

col = "Hyperplane:") +

theme(legend.position = "bottom")

Chart, line chart

Description automatically generated

2. Non linear Decision Boundaries

Equation (1+X1)2+(2−X2)2=4

(X1−(−1))2+(X2−2)2=22

The following R code plots the decision boundaries

ggplot()

geom\_circle(data = data.frame(X1 = -1, X2 = 2, r = 2), aes(x0 = X1, y0 = X2, r = r), col = "mediumseagreen", size = 1) +

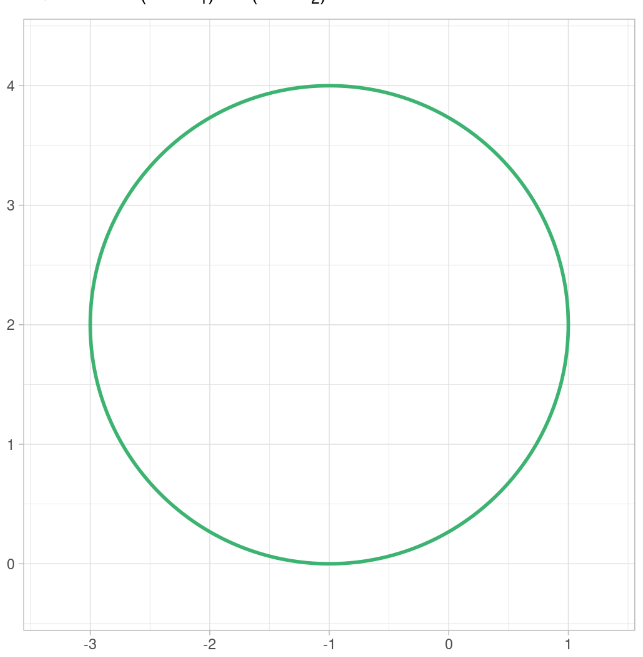
scale\_x\_continuous(expand = c(0.01,0.01), limits = c(-3.5, 1.5)) +

scale\_y\_continuous(expand = c(0.01,0.01), limits = c(-0.5, 4.5)) +

labs(title = TeX(r'(Curve Plot: $(1 + X\_1)^2 + (2 - X\_2)^2 = 4$)'),

coord\_fixed()

The curve plot is follows



b)

grid <- expand.grid(X1 = seq(-3.5, 1.5, length.out = 200),

X2 = seq(-0.5, 4.5, length.out = 200))

mutate(region = case\_when((1 + X1)^2 + (2 - X2)^2 > 4 ~ "gt 4",

TRUE ~ "le 4"))

ggplot() +

geom\_tile(data = grid, aes(x = X1, y = X2, fill = region), alpha = 0.5) +

geom\_circle(data = data.frame(X1 = -1, X2 = 2, r = 2),

aes(x0 = X1, y0 = X2, r = r), col = "mediumseagreen", size = 1) +

annotate("text", x = -1, y = 2, label = TeX("$(1 + X\_1)^2 + (2 - X\_2)^2 \\leq 4$", output = "character"),

hjust = 0.5, size = 4, parse = TRUE, col = "red") +

annotate("text", x = 0.5, y = 0, label = TeX("$(1 + X\_1)^2 + (2 - X\_2)^2 > 4$", output = "character"),

hjust = 0.5, size = 4, parse = TRUE, col = "blue") +

scale\_x\_continuous(expand = c(0.01,0.01), limits = c(-3.5, 1.5)) +

scale\_y\_continuous(expand = c(0.01,0.01), limits = c(-0.5, 4.5)) +

scale\_fill\_manual(values = c("#00BFC4", "#F8766D")) +

Chart, bubble chart

Description automatically generated

c) classification observations:

Ans.

new\_obs <- data.frame(X1 = c(0, -1, 2, 3), X2 = c(0, 1, 2, 8))

mutate(region = case\_when((1 + X1)^2 + (2 - X2)^2 > 4 ~ "gt 4, TRUE ~ "le 4"))

grid <- expand.grid(X1 = seq(-3.5, 3.5, length.out = 200),

X2 = seq(-0.5, 8.5, length.out = 200))

mutate(region = case\_when((1 + X1)^2 + (2 - X2)^2 > 4 ~ "gt 4", TRUE ~ "le 4"))

ggplot()

geom\_tile(data = grid, aes(x = X1, y = X2, fill = region), alpha = 0.5)

geom\_circle(data = data.frame(X1 = -1, X2 = 2, r = 2),

aes(x0 = X1, y0 = X2, r = r), col = "mediumseagreen", size = 1)

geom\_point(data = new\_obs, aes(x = X1, y = X2, col = region))

scale\_x\_continuous(expand = c(0.01,0.01), limits = c(-3.5, 3.5))

scale\_y\_continuous(expand = c(0.01,0.01), limits = c(-0.5, 8.5))

scale\_fill\_manual(values = c("#00BFC4", "#F8766D"))

scale\_color\_manual(values = c("blue", "red"))

Chart, bubble chart

Description automatically generated

d)Non-linear decision boundary

f(X)=0

f(X) = 0 implies

f(X) = (1 + X1)^2 + (2 – X2)^2 - 4

= X1^2 + 2 X1 + 1 + X2^2 - 4 X2 + 4 - 4

= X1^2 + X2^2 + 2 X1 - 4 X2 + 1

= (1)(X1^2) + (1)(X2^2) + (2)( X1) + (-4)( X2) + 1

= 0

From the final form you can see how f(X)= 0 is expressed in the form a1Z1+a2Z2+a3Z3+a4Z4+b=0 for (Z1,Z2,Z3,Z4)=(X12,X22,X1,X2) and is therefore linear

3.Maximal margin classifier

1. Plot observations

Chart, scatter chart

Description automatically generated

**b) M = sqrt(2)/4**

Chart, line chart

Description automatically generated

**C) f(X)=X2−X1+0.5**

**Red** class if f(X∗)>0and the **Blue** class if f(X∗)≤0.

 (β0,β1,β2)=(0.5,−1,1)

grid <- expand.grid(X1 = seq(0.5, 4.5,length.out = 200), X2 = seq(0.5, 4.5,length.out = 200))

mutate(region = case\_when(X2 - X1 + 0.5 < 0 ~ "lt 0",

X2 - X1 + 0.5 > 0 ~ "gt 0"))

ggplot(grid, aes(x = X1, y = X2)) +

geom\_tile(aes(fill = region), alpha = 0.5) +

geom\_point(data = df, aes(x = X1, y = X2, col = Y), size = 2) +

geom\_abline(slope = 1, intercept = -0.5) +

scale\_x\_continuous(expand = c(0.01,0.01), limits = c(0.5, 4.5)) +

scale\_y\_continuous(expand = c(0.01,0.01), limits = c(0.5, 4.5)) +

scale\_color\_manual(values = c("blue", "red")) +

scale\_fill\_manual(values = c("#F8766D", "#00BFC4")) +

theme(legend.position = "none")

Chart

Description automatically generated

**d)Margin**

ggplot(grid, aes(x = X1, y = X2)) +

geom\_tile(aes(fill = region), alpha = 0.5) +

geom\_abline(slope = 1, intercept = -0.5) +

geom\_abline(slope = 1, intercept = -1, linetype = 'dashed', col = 'grey40') +

geom\_abline(slope = 1, intercept = 0, linetype = 'dashed', col = 'grey40') +

geom\_point(data = df, aes(x = X1, y = X2, col = Y), size = 2) +

scale\_x\_continuous(expand = c(0.01,0.01), limits = c(0.5, 4.5)) +

scale\_y\_continuous(expand = c(0.01,0.01), limits = c(0.5, 4.5))

scale\_color\_manual(values = c("blue", "red"))

scale\_fill\_manual(values = c("#F8766D", "#00BFC4"))

theme(legend.position = "none")

**e) Support Vectors**

df$support <- c("No", "Yes", "Yes", "No", "Yes", "Yes", "No")

ggplot(grid, aes(x = X1, y = X2))

geom\_tile(aes(fill = region), alpha = 0.5)

geom\_abline(slope = 1, intercept = -0.5)

geom\_abline(slope = 1, intercept = -1, linetype = 'dashed', col = 'grey40')

geom\_abline(slope = 1, intercept = 0, linetype = 'dashed', col = 'grey40')

geom\_point(data = df, aes(x = X1, y = X2, col = Y, shape = support), size = 2)

geom\_text(data = df, aes(label = obs, vjust = -1), col = 'grey40')

scale\_x\_continuous(expand = c(0.01,0.01), limits = c(0.5, 4.5))

scale\_y\_continuous(expand = c(0.01,0.01), limits = c(0.5, 4.5))

scale\_shape\_manual(values = c('Yes' = 16, 'No' = 1))

scale\_color\_manual(values = c("red", "blue"), limits = c("Red", "Blue"))

scale\_fill\_manual(values = c("#F8766D", "#00BFC4"), labels = c("Red", "Blue"))

**f) Moving an observation**

Observation 7 does not play a critical role in determining the position of the maximum margin hyperplane because its location does not affect the separation between the two classes. Even if it moves slightly in any direction, the hyperplane equation of X2−X1+0.5=0 will still provide the best separation with a margin of M=2√4, which is the widest slab between the classes. Hence, the location of observation 7 is not a support vector and would have to move inside the margin to have any impact on the position of the maximum margin hyperplane.

**g) Non – optimal hyperplane**

Ans.) Here there is a hyperplane X2−0.75⋅X1−0.3=0

grid <- expand.grid(X1 = seq(0.5, 4.5,length.out = 200), X2 = seq(0.5, 4.5,length.out = 200))

mutate(region = case\_when(X2 - 0.75\*X1 - 0.3 < 0 ~ "lt 0", X2 - 0.75\*X1 - 0.3 > 0 ~ "gt 0"))

df$support <- ifelse(df$obs == 2, "Yes", "No")

ggplot(grid, aes(x = X1, y = X2))

geom\_tile(aes(fill = region), alpha = 0.5)

geom\_abline(slope = 0.75, intercept = 0.3)

geom\_abline(slope = 0.75, intercept = 0.1, linetype = 'dashed', col = 'grey40')

geom\_abline(slope = 0.75, intercept = 0.5, linetype = 'dashed', col = 'grey40')

geom\_point(data = df, aes(x = X1, y = X2, col = Y, shape = support), size = 2)

scale\_x\_continuous(expand = c(0.01,0.01), limits = c(0.5, 4.5))

geom\_text(data = df, aes(label = obs, vjust = -1), col = 'grey40')

scale\_y\_continuous(expand = c(0.01,0.01), limits = c(0.5, 4.5))

scale\_shape\_manual(values = c('Yes' = 16, 'No' = 1))

scale\_color\_manual(values = c("red", "blue"), limits = c("Red", "Blue"))

scale\_fill\_manual(values = c("#F8766D", "#00BFC4"), labels = c("Red", "Blue"))

Chart

Description automatically generated

**h) Non seperable data**

Ans. See below where I added observation 8, given by (X1,X2)=(2,3), meaning the two classes can no longer be separated by a hyperplane in two dimensions.

Chart, scatter chart

Description automatically generated

3)