MATH 4323, Fall 2023, Homework # 3

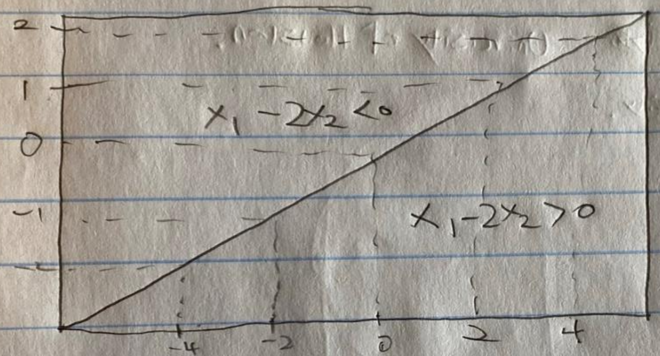
Name: Hyundoo Jeong

PSID:2212332.

1)

Hw3

1a)



i) $x_1=0$, $x_2=2$ lies on the region above hyplane because It is $-4<0$.

ii) $x_1=2$, $x_2=1$ lies on hyplane because it's equal

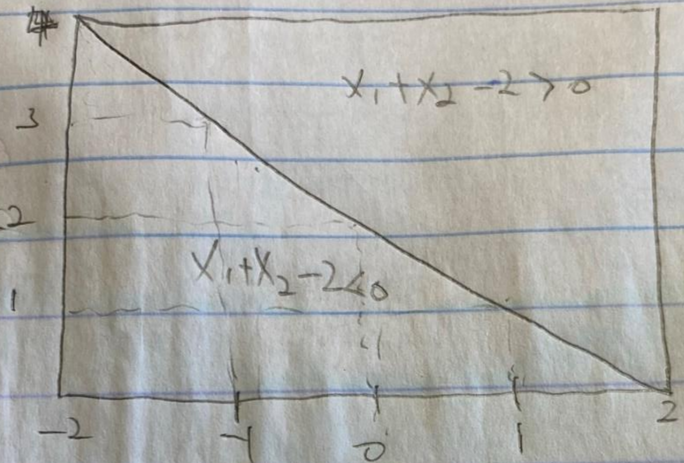
iii) $x_1=0$, $x_2=-2$ lies on the region below hyplane b

It is $4>0$.

iii) $x_1=0$, $x_2=-2$ lies on the region below hyplane becau

It is 4 > 0.

1b)



i) $X_1=1$, $X_2=1$ lies on hyperplane because It is equal

ii) $X_1=-1$, $X_2=-1$ lies on the region below hyperplane bec

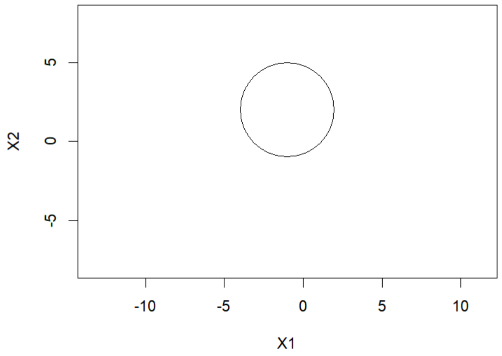
It is $-4<0$,

iii) $X_1=1$, $X_2=1$ lies on hyperplane because It is equa

2A)

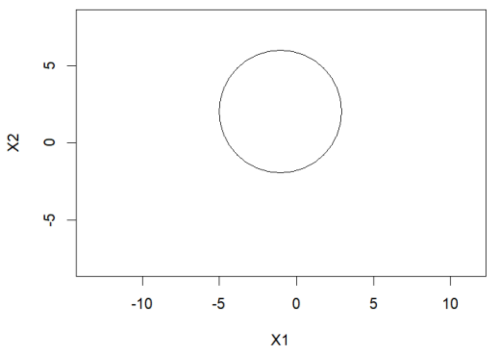
1. $(1 + X_1)^2 + (2 - X_2)^2 = 9.$

```
> plot(NA, NA, type = "n", xlim = c(-9, 7), ylim = c(-8, 8), asp = 1, xlab
 = "X1", ylab = "X2")
> symbols(c(-1), c(2), circles = c(3), add = TRUE, inches = FALSE)
```

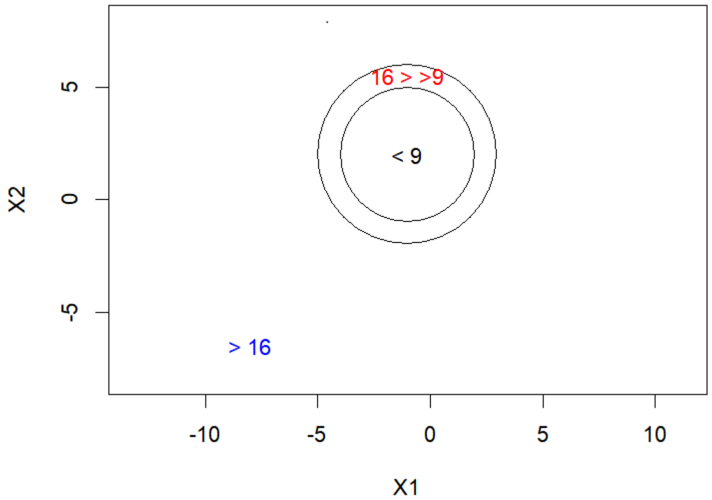


2. $(1 + X_1)^2 + (2 - X_2)^2 = 16.$

```
> plot(NA, NA, type = "n", xlim = c(-9, 7), ylim = c(-8, 8), asp = 1, xlab
 = "X1", ylab = "X2")
> symbols(c(-1), c(2), circles = c(4), add = TRUE, inches = FALSE)
```



2B)



2C)

f(0,0) => red

f(-1,1) =>red

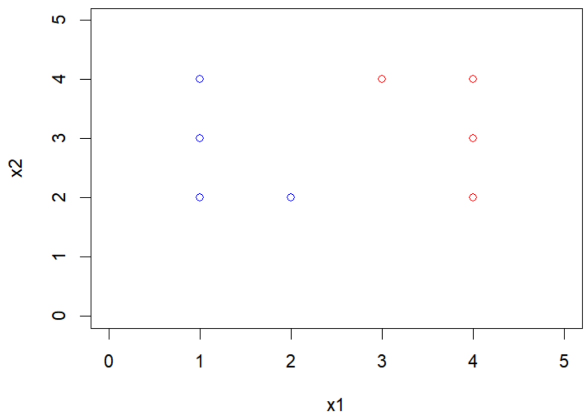
f(2,2)=>blue

f(3,4)=>red

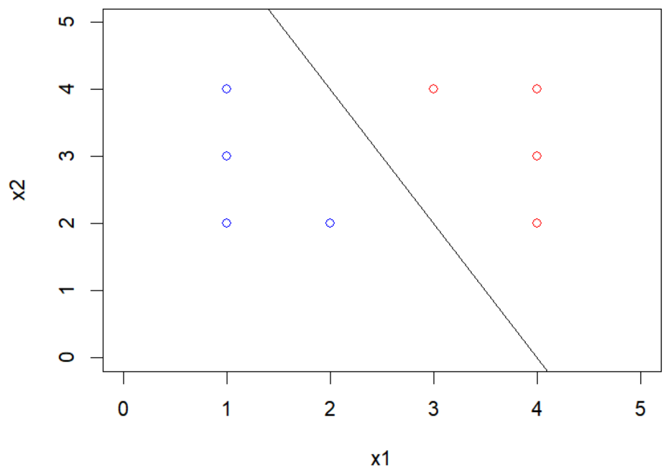
2D)

Answer: In terms of X1 and X2, it is not linear because degree is 2. However, in terms of X1,X1^2,X2, and X2^2, it is linear because degree is 1.

3A) We are given n = 8 observations in p = 2 dimensions. For each observation, there is an associated class label.



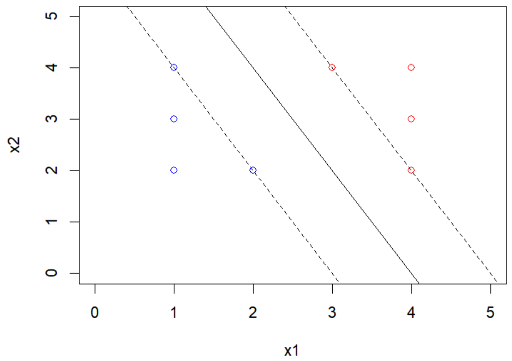
3B)



3C)

Answer: The classification rule is “Classify to red $-2X1-X2+8 < 0$ and classify to Blue otherwise.”

3D)



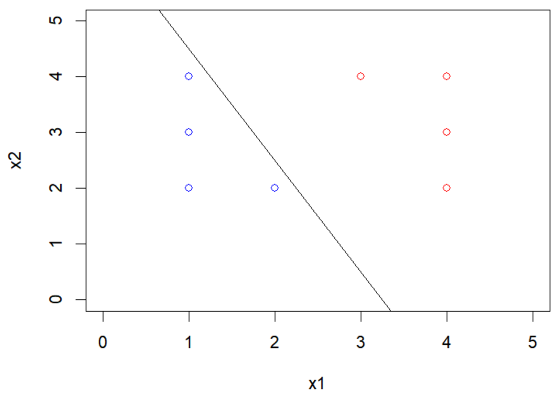
3E)

Answer: Supporting vectors are the points (2,2), (4,2), (1,4), (3,4) .

3F)

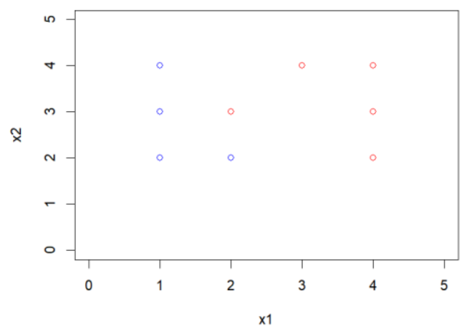
Answer: If we move to the point (1,3), we can’t change the maximal marginal hyperplane as it is a support vector.

3G)



Answer: Hyperplane equation is -2X1-X2+6.5

3H) Answer: red point(2,3) is added to the plot, two classes are obviously not separable.



4A)

```
> library(MASS)
> data("Boston")
> median_medv <- median(Boston$medv)
> Boston$medv01 <- ifelse(Boston$medv <= median_medv, 0, 1)
> Boston$medv <- NULL
> head(Boston)
```

4B)

Values of K:

1. K = 3(small value)
2. K = 5(moderate value)
3. K = 7(Larger value)

Subsets of Predictors:

1. Full set of all 13 predictors:
    - crim
    - zn
    - indus
    - chas
    - nox
    - rm
    - age
    - dis
    - rad
    - tax
    - ptratio
    - black
    - lstat
2. Subset based on logical considerations (a simplified subset for illustration):

- rm (average number of rooms per dwelling)
- lstat (percentage of lower status population)
- dis (weighted distance to employment centers)
- ptratio (pupil-teacher ratio)
- nox (nitrogen oxide concentration)

3. Subset based on correlation analysis (retaining one variable from highly correlated groups):
   - rm (average number of rooms per dwelling)
   - indus (proportion of non-retail business acres per town)
   - tax (property tax rate)
   - age (proportion of owner-occupied units built before 1940)
   - dis (weighted distance to employment centers)

4C)

Answer: We used k – fold cross-validation with different K = 3,2,1

- For K=3, the combination of predictors "lstat" and "rm" had the lowest test error (0.1442688).
- For K=2, the combination of predictors "lstat" and "rm" had the lowest test error (0.173913).
- For K=1, the combination of predictors "lstat" and "rm" had the lowest test error (0.1778656).

Therefore, for each value of K, the combination of predictors "lstat" and "rm" was the winning model with the lowest test error.

4D)

Answer: No, the variables in the Boston dataset are not on the same scale. We can deal with the problem using loocv.

4E)

Answer: No, it is not same scale compared to part (c).

5A)

```
> library(ISLR)
> data("Auto")
> head(Auto)
> Auto$mpglevel<- as.factor(ifelse(Auto$mpg > median(Auto$mpg), 1, 0))
> Auto$mpg<-NULL
```

5B)

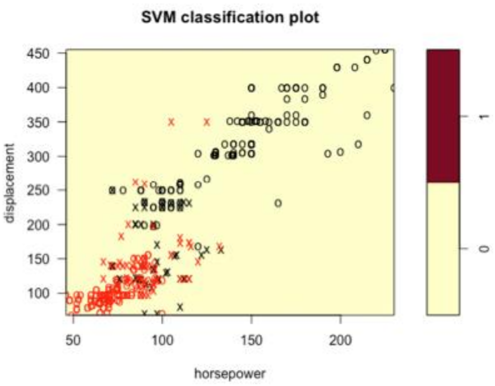
```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost
    5

- best performance: 0.08673077

- Detailed performance results:
   cost      error dispersion
1 1e-02 0.56115385 0.04344202
2 1e-01 0.17371795 0.07157577
3 1e+00 0.09942308 0.05029358
4 5e+00 0.08673077 0.05819036
5 1e+01 0.09435897 0.06043406
6 1e+02 0.08679487 0.05158759
7 1e+03 0.08929487 0.04871612
```

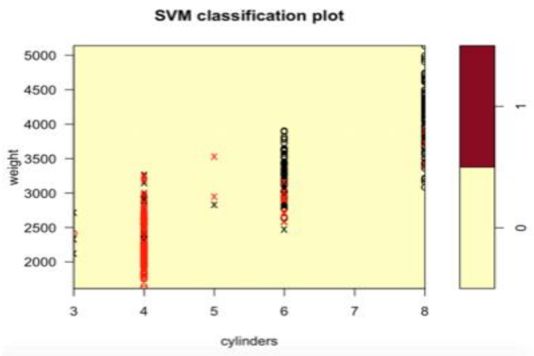
5C)

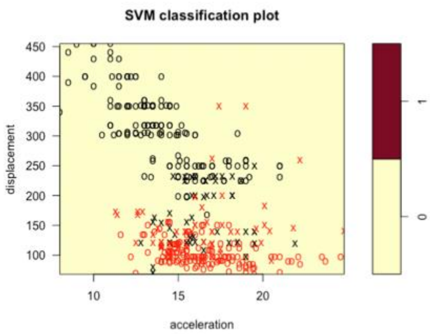
```
                true
predicted       0       1
          0     51      2
          1     10      55
```

Answer: For the best option cost = 5, the svm gave 12 misclassification errors as seen in the above confusion matrix.

5D)







6A)

Answer: Programming R code

library(ISLR)

set.seed(1)

train<- sample(nrow(OJ), 800)

OJ.train <- OJ[train,]

OJ.test <- OJ[-train,]

6B)

```
R 4.2.2 · ~/
Call:
svm(formula = Purchase ~ ., data = OJ.train, kernel = "linear", cost = 0.01)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  0.01

Number of Support Vectors:  432

 ( 215 217 )


Number of Classes:  2

Levels:
 CH MM
```

Answer: There are 432 support vectors. 432 are close to hyperplane. Changing the position of these 432 points slightly can affect the classification. results. All the points are classified into 2 distinct classes. 215 points are in near class 1, and 217 points are in near class 2.

6C)

```
> trainPredict <- predict(svm.linear, OJ.train)
> testPredict <- predict(svm.linear, OJ.test)
> mean(trainPredict != OJ.train$Purchase)
[1] 0.16625
> mean(testPredict != OJ.test$Purchase)
[1] 0.1814815
```

Answer: The training error is 0.16625 and test error is 0.1814815.

6D)

Answer:

```
> tune_out <- tune(svm, Purchase ~ ., data = OJ.train, ranges = list(cost
= 10^(-2:1)))
> best_cost <- tune_out$best.parameters$cost
> print(best_cost)
```

It is 1.

6E)

```
> svm_model_best <- svm(Purchase ~ ., data = OJ.train, cost = best_cost)
> train_pred_best <- predict(svm_model_best, train_data)
Error in predict.svm(svm_model_best, train_data) :
  object 'train_data' not found
> train_pred_best <- predict(svm_model_best, OJ.train)
> test_pred_best <- predict(svm_model_best, OJ.test)
> mean( train_pred_best != OJ.train$Purchase)
[1] 0.145
> mean(test_pred_best != OJ.test$Purchase)
[1] 0.1703704
> |
```

The new training error is 0.145.
The new test error is 0.1703704.