Assignment 3

Alex Haffner

QA1.

In a SVM hard margin, C is equal to infinity. This means there are no margin violations.

While a soft margin decreases the cost parameter and therefore allows for some margin violations and training error. Margins provide tradeoffs between the number of mistakes and the margin of violations in training data.

QA2. A cost parameter is a regularization parameter that decides how much an SVM should bend with the data. In SVM we can test different numbers of this parameter to allow so many or so little training violations.

QA3. The perceptron will not be activated because it does not reach the activation threshold.

(0.1) \*(0.8) = 0.08

11.1 \* (-0.2) = -2.22

0.08 + - 2.22 = - 2.14

-2.14 < 2.8

QA4. The alpha determines how fast the weights should be changed. The higher the alpha, the faster change of weights should happen. The goal is to find the fastest learning rate possible.

Part B

library(ISLR)

library(glmnet)

library(caret)

library(dplyr)

Carseats\_Filtered <- Carseats %>% select("Sales", "Price","Advertising","Population","Age","Income","Education")

View(Carseats\_Filtered)

#QB1.

library(kernlab)

library(rpart)

set.seed(123)

intrain <- createDataPartition(y = Carseats\_Filtered$Sales, p = 0.7, list = FALSE)

training <- Carseats\_Filtered[intrain,]

testing <- Carseats\_Filtered[-intrain,]

model <- train(Sales~., data = training, method = 'svmLinear',)

model

Text

Description automatically generated

The r-squared of the model is .32723.

Question 2

#QB2

set.seed(123)

trctrl <- trainControl(method = "repeatedcv", number = 5, repeats = 2)

grid <- expand.grid(C = c(0.1,0.5,0.1, 10))

svm\_linear\_grid <- train(Sales~., data = training, method = "svmLinear", trControl=trctrl, tuneGrid = grid, tuneLength = 10)

svm\_linear\_grid

Text, letter

Description automatically generated

#QB3

library(neuralnet)

#DataFrame

mydata = data.frame(Carseats\_Filtered)

summary(mydata)  
Table

Description automatically generated with medium confidence

#Scaled Data

Scale\_Model = preProcess(mydata, center = TRUE, scale = TRUE)

MyData\_Scaled= predict(Scale\_Model,mydata)

summary(MyData\_Scaled)

Table

Description automatically generated

#Question 2  
   
samplesize = 0.60 \*nrow(mydata)

set.seed(124)

index = sample(seq\_len( nrow ( mydata ) ), size = samplesize )

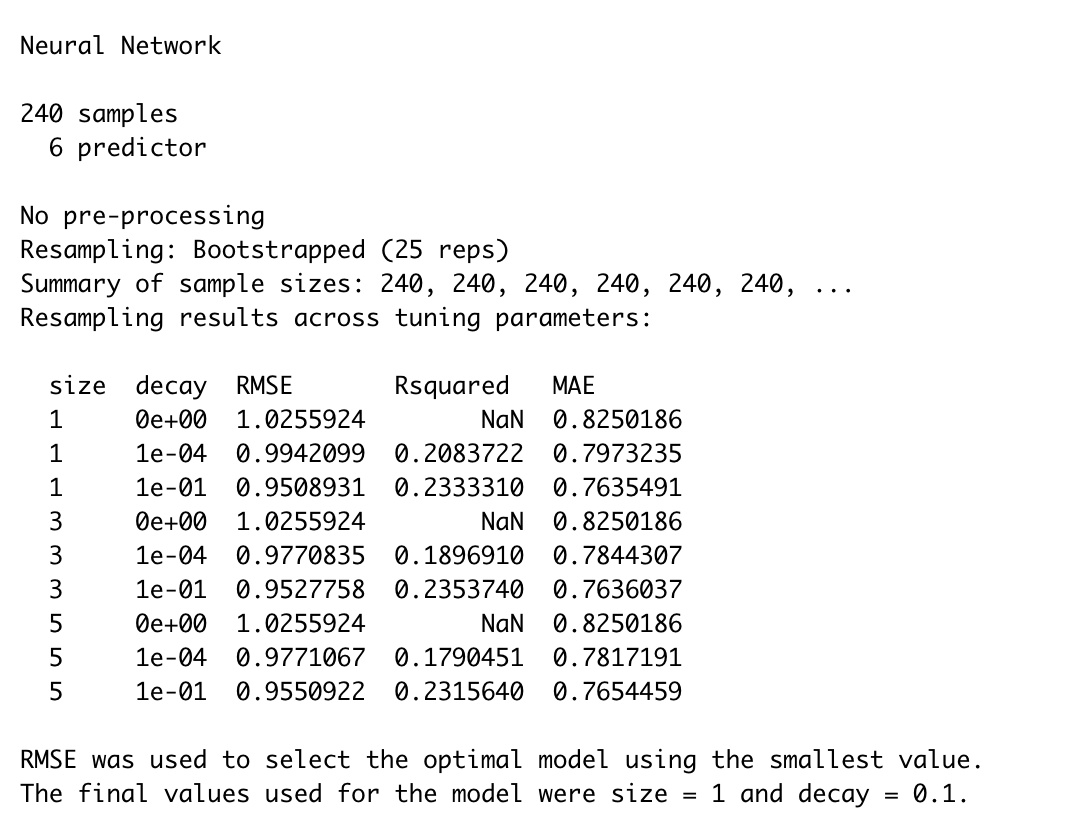
trainTrainsformed = MyData\_Scaled[index,]

testTransformed = (MyData\_Scaled[-index,])

#First Model

NN = train(Sales ~ Price + Advertising + Population + Age + Income + Education, data = trainTrainsformed, method="nnet")

NN



plot(NN)

Chart, line chart

Description automatically generated

The because my model did not turn out. I am using the best RMSE that isn't a Nan value to answer this question. The best model in my case would be the model with an RMSE of .9588 and R-squared of .2508.

#Question 4

grid <- expand.grid(Sales = 9, Price = 6.54, Population = 124, Advertising = 0, Age = 76, Income = 110, Education = 10)

prediction = predict(NN,grid, data = trainTrainsformed)

Application

Description automatically generated with low confidence

I am not sure this is correct. I emailed you to set up a time but never heard back before the assignment was due.