Test 2 Memo

Dr. Niladri Chakraborty

2024-05-18

```
#Loding the necessary packages

library(tidyverse)
library(MASS)
library(e1071)
library(neuralnet)
library(NeuralNetTools)
library(hrbrthemes)
library(lattice)
library(caret)
```

Q1.

Consider the 'mtcars' dataset in R.

(i) Add a short description and print the structure of the dataset.

(ii) Split the mtcars dataset into a training set and a testing set with a 70:30 ratio.

(iii) Fit Support Vector Machine (SVM) models to the training dataset with radial kernel and polynomial kernel considering 'mpg' as the dependent variable. Print the model summary for both models.

(iv) Using both models, predict the 'mpg' values for the test data.

(v) Evaluate and comment on the prediction accuracy of both models.

Ans.

The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

```
# Load the dataset
data("mtcars")

str(mtcars)

## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
```

```
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
summary(mtcars)
```

```
##       mpg             cyl             disp             hp       
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0  
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
##       drat             wt             qsec             vs        
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000  
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000  
##        am              gear            carb      
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
##  Median :0.0000   Median :4.000   Median :2.000  
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812  
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

```
# Split the mtcars data into training and testing datasets

set.seed(123)
trainIndex <- createDataPartition(mtcars$mpg, p = 0.7, list = FALSE, times =
1)
mtcarsTrain <- mtcars[trainIndex,]
mtcarsTest <- mtcars[-trainIndex,]
```

```r
# Create the model
svm_model1 <- svm(mpg ~ ., data = mtcarsTrain, type = "eps-regression",
kernel = 'radial')
# Print the model summary
summary(svm_model1)
```

```
##
## Call:
## svm(formula = mpg ~ ., data = mtcarsTrain, type = "eps-regression",
##     kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.1
##     epsilon:  0.1
##
##
## Number of Support Vectors:  23
```

```r
# Create the model
svm_model2 <- svm(mpg ~ ., data = mtcarsTrain, type = "eps-regression",
kernel = 'polynomial')
# Print the model summary
summary(svm_model2)
```

```
##
## Call:
## svm(formula = mpg ~ ., data = mtcarsTrain, type = "eps-regression",
##     kernel = "polynomial")
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  3
##       gamma:  0.1
##      coef.0:  0
##     epsilon:  0.1
##
##
## Number of Support Vectors:  16
```

```r
# Predict using the model
predictions1 <- predict(svm_model1, mtcarsTest)

#Print the predicted values
predictions1
```

```
##     Mazda RX4 Wag        Duster 360 Chrysler Imperial  Dodge Challenger
##          20.23258          14.63716          11.00264          18.53871
##     Porsche 914-2      Lotus Europa     Ford Pantera L      Ferrari Dino
##          21.89836          21.56823          17.12608          18.11284

# Predict using the model
predictions2 <- predict(svm_model2, mtcarsTest)

#Print the predicted values
predictions2

##     Mazda RX4 Wag        Duster 360 Chrysler Imperial  Dodge Challenger
##         20.166104         11.984652          9.841711         17.776325
##     Porsche 914-2      Lotus Europa     Ford Pantera L      Ferrari Dino
##         23.729165         22.595211         20.951850         21.145500

# Evaluate model
svm_accuracy1 <- postResample(predictions1, mtcarsTest$mpg)
svm_accuracy2 <- postResample(predictions2, mtcarsTest$mpg)
svm_accuracy1

##      RMSE  Rsquared       MAE
## 3.9164410 0.6287803 2.9609141

svm_accuracy2

##      RMSE  Rsquared       MAE
## 4.0240418 0.5270942 3.3696039
```

SVM model with radial kernel outperforms the SVM model with polynomial kernel across all three considered metrics suggesting it is the more accurate and reliable model for predicting or fitting the mtcars data.

Q2.

(i) **Fit another SVM model on the same training data you obtained in Q1 considering 'mpg' as the dependent variable and 'wt' and 'hp' as the independent variables. Use 'radial' kernel for the SVM model.**

(ii) **Using this new SVM model, create a contour plot for the 'mpg' values against the 'wt' and 'hp' values**

Ans.

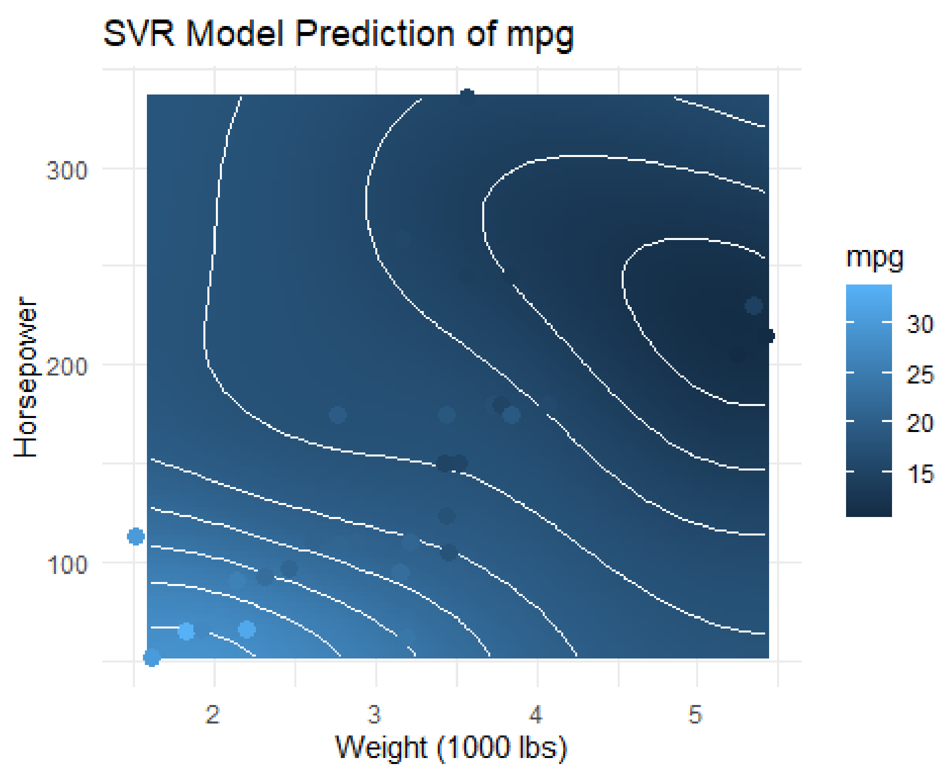
```
svm_model3 <- svm(mpg ~ wt+hp, data = mtcarsTrain, type = "eps-regression",
kernel = 'radial')

# Make predictions over a grid to plot
wt_seq <- seq(min(mtcarsTrain$wt), max(mtcarsTrain$wt), length.out = 100)
```

```
hp_seq <- seq(min(mtcarsTrain$hp), max(mtcarsTrain$hp), length.out = 100)
grid <- expand.grid(wt = wt_seq, hp = hp_seq)
grid$mpg <- predict(svm_model3, newdata = grid)

# Basic plot of the fitted surface
ggplot(grid, aes(x = wt, y = hp, fill = mpg)) +
  geom_tile() +
  geom_contour(aes(z = mpg), color = "white") +
  labs(title = "SVR Model Prediction of mpg", x = "Weight (1000 lbs)", y =
"Horsepower", fill = "mpg") +
  theme_minimal()+
  geom_point(data = mtcars, aes(x = wt, y = hp, color = mpg), size = 3)

## Warning: The following aesthetics were dropped during statistical
transformation: fill
## ℹ This can happen when ggplot fails to infer the correct grouping
structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



Q3.

(i) Fit a linear regression model for the 'mtcars' dataset with 'vs' as the dependent variable. Obtain the summary of the model output with a brief explanation.

(ii) Now create a regression line plot for the 'mpg' values against the 'wt' values.

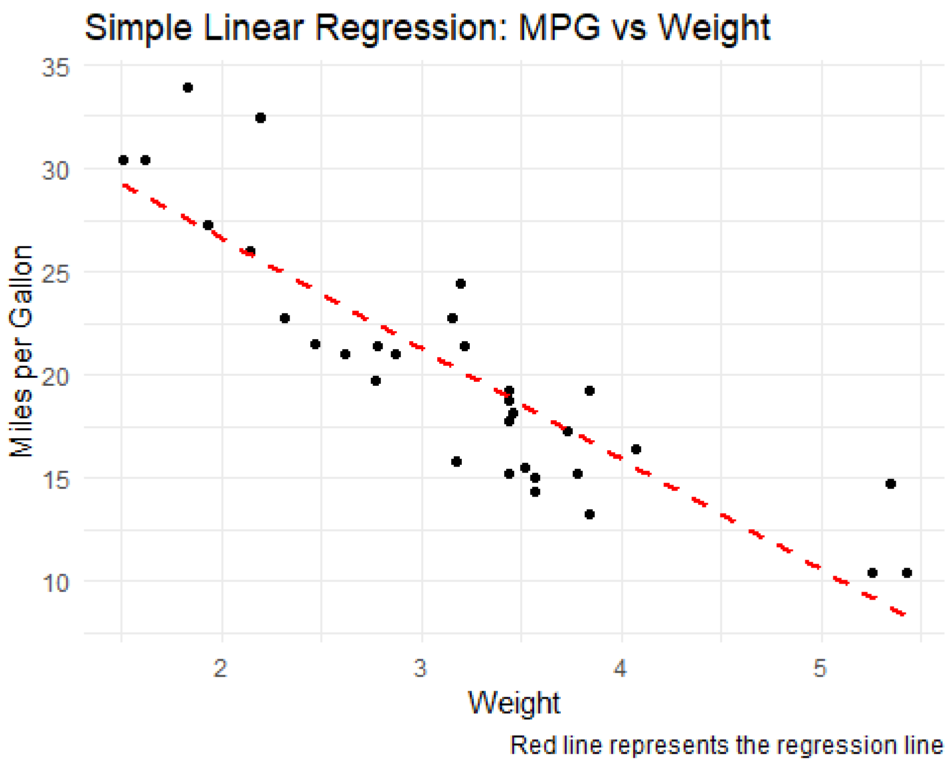
Ans.

```
# Fit the linear regression model
model <- glm(vs ~ mpg + wt + hp, data = mtcars, family = binomial)
# Print the model summary
summary(model)

##
## Call:
## glm(formula = vs ~ mpg + wt + hp, family = binomial, data = mtcars)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.61945   16.52453  -0.643   0.5205
## mpg           0.50291    0.48656   1.034   0.3013
## wt            3.87749    3.19255   1.215   0.2245
## hp           -0.09318    0.04318  -2.158   0.0309 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.860  on 31  degrees of freedom
## Residual deviance: 14.748  on 28  degrees of freedom
## AIC: 22.748
##
## Number of Fisher Scoring iterations: 8

ggplot(mtcars, aes(x=wt, y=mpg)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE, color="red", linetype="dashed") +
theme_minimal() +
  labs(title="Simple Linear Regression: MPG vs Weight", x="Weight", y="Miles
per Gallon", caption="Red line represents the regression line")

## `geom_smooth()` using formula = 'y ~ x'
```



Q4.

Consider the 'PlantGrowth' in R.

(i) Review the data structure and add a brief description of the dataset.

(ii) Fit an ANOVA model and obtain the model summary.

(iii) Are there significant differences in yields across various treatment conditions?

Ans. This dataset contains weight measurements for plants grown in three different treatment conditions. We will perform ANOVA to determine if there are statistically significant differences in the average plant weight across these treatment groups.

```
data(PlantGrowth)
str(PlantGrowth)

## 'data.frame':    30 obs. of  2 variables:
##  $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
##  $ group : Factor w/ 3 levels "ctrl","trt1",..: 1 1 1 1 1 1 1 1 1 1 ...

head(PlantGrowth)

##   weight group
## 1   4.17  ctrl
```

```
## 2   5.58  ctrl
## 3   5.18  ctrl
## 4   6.11  ctrl
## 5   4.50  ctrl
## 6   4.61  ctrl

summary(PlantGrowth)

##      weight       group
##  Min.   :3.590   ctrl:10
##  1st Qu.:4.550   trt1:10
##  Median :5.155   trt2:10
##  Mean   :5.073
##  3rd Qu.:5.530
##  Max.   :6.310

model <- aov(weight ~ group, data=PlantGrowth)
summary(model)

##             Df Sum Sq Mean Sq F value Pr(>F)
## group        2  3.766  1.8832   4.846 0.0159 *
## Residuals   27 10.492  0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model summary reflects significant differences in the the average plant weight across these treatment groups.

Q5.

Consider the ‘rock’ dataset.

(i) Review the data structure and add a brief description of the dataset.

(ii) Normalize the dataset.

(iii) Fit an Artificial Neural Network (ANN) model on the full dataset with ‘perm’ as the dependent variable. Plot the ANN model (use the NeuralNetTools package).

```
# Load the dataset
data("rock")

str(rock)

## 'data.frame':    48 obs. of  4 variables:
##  $ area : int  4990 7002 7558 7352 7943 7979 9333 8209 8393 6425 ...
##  $ peri : num  2792 3893 3931 3869 3949 ...
##  $ shape: num  0.0903 0.1486 0.1833 0.1171 0.1224 ...
##  $ perm : num  6.3 6.3 6.3 6.3 17.1 17.1 17.1 17.1 119 119 ...

head(rock)
```

```
##   area    peri     shape perm
## 1 4990 2791.90 0.0903296  6.3
## 2 7002 3892.60 0.1486220  6.3
## 3 7558 3930.66 0.1833120  6.3
## 4 7352 3869.32 0.1170630  6.3
## 5 7943 3948.54 0.1224170 17.1
## 6 7979 4010.15 0.1670450 17.1
```

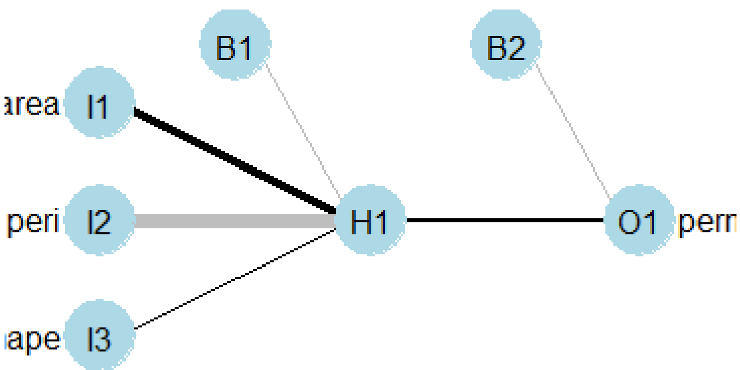
```r
summary(rock)
```

```
##       area            peri            shape              perm        
##  Min.   : 1016   Min.   : 308.6   Min.   :0.09033   Min.   :   6.30  
##  1st Qu.: 5305   1st Qu.:1414.9   1st Qu.:0.16226   1st Qu.:  76.45  
##  Median : 7487   Median :2536.2   Median :0.19886   Median : 130.50  
##  Mean   : 7188   Mean   :2682.2   Mean   :0.21811   Mean   : 415.45  
##  3rd Qu.: 8870   3rd Qu.:3989.5   3rd Qu.:0.26267   3rd Qu.: 777.50  
##  Max.   :12212   Max.   :4864.2   Max.   :0.46413   Max.   :1300.00
```

```r
# Normalize data
maxs <- apply(rock, 2, max)
mins <- apply(rock, 2, min)
scaled_rock <- as.data.frame(scale(rock, center = mins, scale = maxs - mins))

# Setting up the neural network
set.seed(123)
nn <- neuralnet(perm ~ area + peri + shape, data = scaled_rock)

# Plotting the neural network
plotnet(nn)
```



Q6. Create a function with 'for' loop that computes the factorial of a given number, n. The factorial of a number is the product of all positive integers up to that number. For example, the factorial of 5 is 5 * 4 * 3 * 2 * 1 = 120. using the function, calculate the factorial of 10.

Ans.

```
factorial_function <- function(n) {
  factorial = 1
  for (i in 1:n) {
    factorial <- factorial * i
  }
  return(factorial)
}
factorial_function(10)

## [1] 3628800
```

Q7. Write a repeat loop that continues to add random samples drawn from a standard normal distribution until the sum exceeds 10. Use seed value 100. Print the number of iterations required.

```
# Set the seed to ensure reproducible results
set.seed(100)

# Initialize total sum and iteration counter
total_sum <- 0
iterations <- 0

# Start the repeat loop
repeat {
    sample <- rnorm(1)
    total_sum <- total_sum + sample
    iterations <- iterations + 1
    if (total_sum > 10) {
        break  # Exit the loop if condition is met
    }
}

# Print the number of iterations required
iterations

## [1] 842
```