ADM Assignmnet 3

Harish Kunaparaju

2023-04-29

```
library(ISLR)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-6
```

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(kernlab)
```

```
##
## Attaching package: 'kernlab'

## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
Carseats_Filtered <- Carseats %>% select("Sales", "Price", "Advertising","Population","Age","Income","E
```

**QB1. Build a linear SVM regression model to predict Sales based on all other attributes ("Price", "Advertising", "Population", "Age", "Income" and "Education"). Hint: use caret train() with method set to "svmLinear". What is the R-squared of the model?

```
myctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
set.seed(33)
svm_Linear <- train(Sales~., data = Carseats_Filtered, method = "svmLinear",
                    trControl=myctrl,
                    preProcess = c("center", "scale"),
                    tuneLength = 10)
svm_Linear
```

```
## Support Vector Machines with Linear Kernel
##
## 400 samples
##   6 predictor
##
## Pre-processing: centered (6), scaled (6)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 360, 360, 360, 360, 360, 360, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   2.266556  0.3702692  1.826177
##
## Tuning parameter 'C' was held constant at a value of 1
```

In the above model, I added a train control section that adds cross validation to the model.the R square of the model is 0.3702692,The tuning parameter "c" was held constant at a value of 1

**QB2 Customize the search grid by checking the model's performance for C parameter of 0.1,.5,1 and 10 using 2 repeats of 5-fold cross validation.

```
grid <- expand.grid(C = c(0.1,0.5,1,10))
myctrl2 <- trainControl(method = "repeatedcv", number = 5, repeats = 2)
svm_Linear_Grid <- train(Sales~., data = Carseats_Filtered, method = "svmLinear",
                         trControl=myctrl2,
                         preProcess = c("center", "scale"),
                         tuneGrid = grid,
                         tuneLength = 10)
svm_Linear_Grid
```

```
## Support Vector Machines with Linear Kernel
##
## 400 samples
##   6 predictor
##
## Pre-processing: centered (6), scaled (6)
## Resampling: Cross-Validated (5 fold, repeated 2 times)
## Summary of sample sizes: 320, 320, 319, 320, 321, 320, ...
## Resampling results across tuning parameters:
##
##   C     RMSE      Rsquared   MAE
##    0.1  2.261443  0.3572556  1.814723
##    0.5  2.262722  0.3572088  1.815372
##    1.0  2.262918  0.3571380  1.815725
##   10.0  2.263355  0.3571151  1.815768
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was C = 0.1.
```

#In this step we have added a search grid at the desired points. We have also adjusted the cross validation as instructed. The interesting outcome is that as we increase C, the change decreases. The difference between 1 and 10 is much less than between 0.1 and 0.5. With this, we can see that **0.1** is the best C available.

** QB3 Train a neural network model to predict Sales based on all other attributes (“Price”, “Advertising”, “Population”, “Age”, “Income” and “Education”). Hint: use caret train() with method set to “nnet”. What is the R-square of the model with the best hyper parameters (using default caret search grid) – hint: don’t forget to scale the data.

```
set.seed(333)
neuraldata<-train(Sales~., data=Carseats_Filtered,method='nnet',linout=TRUE,
          preProcess = c("center", "scale"),
          trace=FALSE)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```
print(neuraldata)
```

```
## Neural Network
##
## 400 samples
##   6 predictor
##
## Pre-processing: centered (6), scaled (6)
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 400, 400, 400, 400, 400, 400, ...
## Resampling results across tuning parameters:
##
##   size  decay  RMSE      Rsquared   MAE
##   1     0e+00  2.378434  0.3152778  1.893636
##   1     1e-04  2.327953  0.3363996  1.847815
##   1     1e-01  2.319733  0.3402767  1.839524
##   3     0e+00  2.624000  0.2262713  2.070419
##   3     1e-04  2.770440  0.2361570  2.052908
##   3     1e-01  2.470122  0.2764282  1.975647
##   5     0e+00  2.941597  0.1858532  2.239928
##   5     1e-04  2.672187  0.2164144  2.143118
##   5     1e-01  2.635138  0.2279019  2.120872
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were size = 1 and decay = 0.1.
```

#The model selected size = 1 and decay 0.1 as the most optimal model using RMSE. The specific Rsquared for this model is decay 1e-01 the closest Rsquared 0.3402767 for a model with RMSE of 2.319733

**QB4 Consider the following input: • Sales=9 • Price=6.54 • Population=124 • Advertising=0 • Age=76 • Income= 110 • Education=10 What will be the estimated Sales for this record using the above neuralnet model?

```
Sales <- c(9)
Price <- c(6.54)
Population <- c(124)
Advertising <- c(0)
Age <- c(76)
Income <- c(110)
Education <- c(10)
Test <- data.frame(Sales, Price, Population, Advertising, Age, Income, Education)
```

#Now that we have established the test data that is needed, it is time to predict using our network.

```
Pred_sales <- predict(neuraldata, Test)
Pred_sales
```

```
##        1
## 11.46031
```

#The resulting Pred_sales variable would contain a vector of predicted Sales values, one for each row in the Test data frame, based on the neural network model's estimates of the relationship between the predictor variables and the Sales response variable.The sales predicted value would be "11.46031"