Exercise-create_model.R

jamia

2022-11-08

```
#Name: Jamia Begum
#NIU: 1676891



library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#install.packages("caret")
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.2

## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
#Boston



#Split the data in 70% (train) and 30% (test)
## 75% of the sample size
smp_size <- floor(0.75 * nrow(Boston))
set.seed(123)
train_ind <- sample(seq_len(nrow(Boston)), size = smp_size)
#seq_len(nrow(Boston)) gives a vector of 1:nrow(Boston)
#train_ind gives the random row numbers of 70% of the size

train <- Boston[train_ind, ]
test <- Boston[-train_ind, ]
#train
#test


#creating the model
#Stepwise regression is a procedure we can use to build a regression model
#from a set of predictor variables by entering and removing predictors in a
#stepwise manner into the
#model until there is no statistically valid reason to enter or remove any more.

#We will fit a multiple linear regression model using median house value (mdev)
#as our response variable and all of the other 13 variables
#in the dataset as potential predictors variables.


#performing both-direction stepwise regression using step function
model<-step(lm(medv ~ ., data = train), trace = 0, direction = "both")

#view final model

model
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
##     tax + ptratio + black + lstat, data = train)
##
## Coefficients:
## (Intercept)         crim           zn         chas          nox           rm
##   38.898906    -0.103016     0.054315     3.660510   -16.331036     3.350842
##         dis          rad          tax      ptratio        black        lstat
##   -1.560359     0.292992    -0.010568    -0.889102     0.006808    -0.596534
```

```r
#The goodness of fit of a statistical model
#describes how well it fits a set of observations.
#one of the goodness of fit is the Root Mean Squared Error (RMSE) value,
#which measures the average prediction error made by the model in predicting
#the outcome for an observation. That is, the average difference between
#the observed known outcome values and the values predicted by the model.
```

```r
#The lower the RMSE, the better the model.

gof<-RMSE(fitted(model), train$medv)
gof
```

```
## [1] 4.64805
```

```r
#validate the model in the test dataset and compare
predictions<-predict(model, newdata = test)

compare <- data.frame(Model = c("model on train data",
                                "model on test data"),
                 rmse = round(c(gof,RMSE(predictions, test$medv)),6))



compare
```

```
##                  Model     rmse
## 1 model on train data 4.648050
## 2  model on test data 4.883027
```

```r
#since the RMSE value is bit more for the test data,so our predictive model
#is a bit overfitting
```