project2\_rf\_regression

EMVP

2023-05-02

#Q1  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0   
## ✔ readr 2.1.4 ✔ forcats 1.0.0   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(caret)

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

library(randomForest)

## randomForest 4.7-1.1  
## Type rfNews() to see new features/changes/bug fixes.  
##   
## Attaching package: 'randomForest'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine  
##   
## The following object is masked from 'package:ggplot2':  
##   
## margin

data("Boston", package = "MASS")  
  
nrow(Boston)

## [1] 506

#Q2  
set.seed(123)  
training.samples <- Boston$medv %>%  
 createDataPartition(p = 0.8, list = FALSE)  
train.data <- Boston[training.samples, ]  
test.data <- Boston[-training.samples, ]  
str(train.data)

## 'data.frame': 407 obs. of 14 variables:  
## $ crim : num 0.00632 0.02731 0.03237 0.06905 0.08829 ...  
## $ zn : num 18 0 0 0 12.5 12.5 12.5 12.5 12.5 0 ...  
## $ indus : num 2.31 7.07 2.18 2.18 7.87 7.87 7.87 7.87 7.87 8.14 ...  
## $ chas : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ nox : num 0.538 0.469 0.458 0.458 0.524 0.524 0.524 0.524 0.524 0.538 ...  
## $ rm : num 6.58 6.42 7 7.15 6.01 ...  
## $ age : num 65.2 78.9 45.8 54.2 66.6 96.1 85.9 82.9 39 56.5 ...  
## $ dis : num 4.09 4.97 6.06 6.06 5.56 ...  
## $ rad : int 1 2 3 3 5 5 5 5 5 4 ...  
## $ tax : num 296 242 222 222 311 311 311 311 311 307 ...  
## $ ptratio: num 15.3 17.8 18.7 18.7 15.2 15.2 15.2 15.2 15.2 21 ...  
## $ black : num 397 397 395 397 396 ...  
## $ lstat : num 4.98 9.14 2.94 5.33 12.43 ...  
## $ medv : num 24 21.6 33.4 36.2 22.9 27.1 18.9 18.9 21.7 19.9 ...

str(test.data)

## 'data.frame': 99 obs. of 14 variables:  
## $ crim : num 0.0273 0.0299 0.2112 0.2249 0.6298 ...  
## $ zn : num 0 0 12.5 12.5 0 0 0 0 0 75 ...  
## $ indus : num 7.07 2.18 7.87 7.87 8.14 8.14 8.14 8.14 5.96 2.95 ...  
## $ chas : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ nox : num 0.469 0.458 0.524 0.524 0.538 0.538 0.538 0.538 0.499 0.428 ...  
## $ rm : num 7.18 6.43 5.63 6.38 5.95 ...  
## $ age : num 61.1 58.7 100 94.3 61.8 84.5 94.1 100 68.2 15.8 ...  
## $ dis : num 4.97 6.06 6.08 6.35 4.71 ...  
## $ rad : int 2 3 5 5 4 4 4 4 5 3 ...  
## $ tax : num 242 222 311 311 307 307 307 307 279 252 ...  
## $ ptratio: num 17.8 18.7 15.2 15.2 21 21 21 21 19.2 18.3 ...  
## $ black : num 393 394 387 393 397 ...  
## $ lstat : num 4.03 5.21 29.93 20.45 8.26 ...  
## $ medv : num 34.7 28.7 16.5 15 20.4 18.2 12.7 14.5 18.9 34.9 ...

#Q3  
set.seed(123)  
model <- train(  
 medv ~., data = train.data, method = "rf",  
 trControl = trainControl("cv", number = 10)  
)  
# Best tuning parameter mtry  
model$bestTune

## mtry  
## 2 7

model

## Random Forest   
##   
## 407 samples  
## 13 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 366, 367, 366, 366, 366, 366, ...   
## Resampling results across tuning parameters:  
##   
## mtry RMSE Rsquared MAE   
## 2 3.635979 0.8577443 2.490950  
## 7 3.200443 0.8828334 2.206899  
## 13 3.342935 0.8705382 2.311711  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was mtry = 7.

#Q4  
predictions <- model %>% predict(test.data)  
head(predictions)

## 3 6 9 11 14 15   
## 34.86423 25.37893 18.68497 21.34806 20.37029 19.91423

# Compute the average prediction error RMSE  
RMSE(predictions, test.data$medv)

## [1] 3.107002

#Q5  
set.seed(123)  
rf <- randomForest(medv ~ ., data=Boston, ntree=500, mtry=7,keep.forest=FALSE,importance=TRUE)  
rf

##   
## Call:  
## randomForest(formula = medv ~ ., data = Boston, ntree = 500, mtry = 7, keep.forest = FALSE, importance = TRUE)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 7  
##   
## Mean of squared residuals: 9.876521  
## % Var explained: 88.3

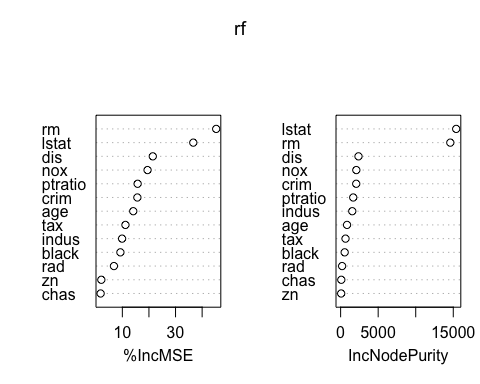
sqrt(rf$mse[500])

## [1] 3.142693

importance(rf)

## %IncMSE IncNodePurity  
## crim 15.650276 2077.63508  
## zn 2.096193 74.01045  
## indus 9.908350 1550.15527  
## chas 1.834505 89.85939  
## nox 19.480394 2090.06065  
## rm 45.281438 14604.65858  
## age 14.099022 854.13476  
## dis 21.446882 2379.79218  
## rad 6.842151 197.82982  
## tax 11.167872 651.73447  
## ptratio 15.750284 1678.73412  
## black 9.272033 538.92145  
## lstat 36.673347 15387.09079

#varImp(rf, type = 2)  
varImpPlot(rf)



#Q6  
12/3

## [1] 4

# 4 variables to select