assignment4_2.R

ramom

2023-06-13

```r
#setting up working directory
getwd()
```

```
## [1] "C:/Users/ramom/Desktop/MDS/Academic/1st Semester/MDS- 3- R/Assignments/Assignment4.2"
```

```r
setwd("C:/Users/ramom/Desktop/MDS/Academic/1st Semester/MDS- 3- R/Assignments/Assignment4.2")
getwd()
```

```
## [1] "C:/Users/ramom/Desktop/MDS/Academic/1st Semester/MDS- 3- R/Assignments/Assignment4.2"
```

```r
# Set the seed
set.seed(26)

#1. Generate a 1000 random data with 10 variables
#[five continuous: age (18 to 90 years),
#height (150 - 180 cm), weight (50 - 90 kg),
#income (10000 - 200000), diastolic blood pressure (70 - 170 mm Hg) and
#five categorical: sex (male/female), education (no education, primary,
#secondary, tertiary), place of residence (rural/urban),
#socio-economic status (low/medium/high) and exercise (yes/no)]
#using set.seed(your roll number and save it as SR object

#generating five continuous variable
age <- sample(18:90, 1000, replace = TRUE)
height <- sample(150:180, 1000, replace = TRUE)
weight <- sample(50:90, 1000, replace = TRUE)
income <- sample(10000:200000, 1000, replace = T)
diastolic_bp <- sample(70:170, 1000, replace = T)

#create categorical variables
sex <- sample(c("male", "female"), 1000, replace = TRUE)
education <- sample(c("no education", "primary", "secondary",
                      "tertiary"), 1000, replace = TRUE)
place_of_residence <- sample(c("rural", "urban"), 1000, replace = TRUE)
socioeconomic_status <- sample(c("low", "medium", "high"), 1000, replace = TRUE)
exercise <- sample(c("yes", "no"), 1000, replace = TRUE)

#create the dataframe
SR <- data.frame(age, height, weight, income, diastolic_bp, sex,
                 education, place_of_residence, socioeconomic_status, exercise)
head(SR)  #check the head of the dataframe
```

```
##   age height weight income diastolic_bp    sex    education place_of_residence socioeconomic_status e
## 1  81    170     90  21877          163   male no education              urban                 high
## 2  45    166     77  85722          116 female no education              rural                  low
## 3  89    173     59 189209          164   male      primary              rural               medium
## 4  60    171     60 198999          100   male      primary              urban                  low
## 5  68    165     52  19652           91 female      primary              urban               medium
## 6  58    166     72  69655           88   male    secondary              urban               medium
```

```r
str(SR)
```

```
## 'data.frame':    1000 obs. of  10 variables:
##  $ age                 : int  81 45 89 60 68 58 88 53 29 71 ...
##  $ height              : int  170 166 173 171 165 166 170 168 158 167 ...
##  $ weight              : int  90 77 59 60 52 72 69 87 55 77 ...
##  $ income              : int  21877 85722 189209 198999 19652 69655 114745 160929 180455 95714 ...
##  $ diastolic_bp        : int  163 116 164 100 91 88 87 92 153 163 ...
##  $ sex                 : chr  "male" "female" "male" "male" ...
##  $ education           : chr  "no education" "no education" "primary" "primary" ...
##  $ place_of_residence  : chr  "urban" "rural" "rural" "urban" ...
##  $ socioeconomic_status: chr  "high" "low" "medium" "low" ...
##  $ exercise            : chr  "yes" "no" "no" "yes" ...
```

```r
#changing categorical variable into factor
SR$sex <- as.factor(SR$sex)
SR$education <- as.factor(SR$education)
SR$place_of_residence <- as.factor(SR$place_of_residence)
SR$socioeconomic_status <- as.factor(SR$socioeconomic_status)
SR$exercise <- as.factor(SR$exercise)

str(SR)  #check the structure
```

```
## 'data.frame':    1000 obs. of  10 variables:
##  $ age                 : int  81 45 89 60 68 58 88 53 29 71 ...
##  $ height              : int  170 166 173 171 165 166 170 168 158 167 ...
##  $ weight              : int  90 77 59 60 52 72 69 87 55 77 ...
##  $ income              : int  21877 85722 189209 198999 19652 69655 114745 160929 180455 95714 ...
##  $ diastolic_bp        : int  163 116 164 100 91 88 87 92 153 163 ...
##  $ sex                 : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 2 2 2 1 ...
##  $ education           : Factor w/ 4 levels "no education",..: 1 1 2 2 2 3 3 3 2 4 ...
##  $ place_of_residence  : Factor w/ 2 levels "rural","urban": 2 1 1 2 2 2 2 1 1 2 ...
##  $ socioeconomic_status: Factor w/ 3 levels "high","low","medium": 1 2 3 2 3 3 3 2 1 2 ...
##  $ exercise            : Factor w/ 2 levels "no","yes": 2 1 1 2 2 2 1 2 2 1 ...
```

```r
#2. Randomly split the SR object data as SR.train (70%) and SR.test (30%) with
#replacement sampling and fit multiple linear regression with diastolic
#blood pressure as dependent variable and rest of variables as independent
#variable and get fit indices (R-Square, MSE, RMSE and MAE) for the SR.test data

#randomly split into 70% train and 30% test
ind <- sample(2, nrow(SR), replace = T, prob = c(0.7, 0.3))

#split into training and testing dataset
```

```r
SR.train <- SR[ind == 1, ]
SR.test <- SR[ind == 2, ]

#fitting the multiple linear regression model with diastolic_bp as
#dependent variable
mlr_model <- lm(diastolic_bp ~ ., data = SR.train)

#model accuracy for training data set
summary(mlr_model)
```

```
##
## Call:
## lm(formula = diastolic_bp ~ ., data = SR.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.401 -25.962   1.401  25.678  53.795
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.068e+02  2.302e+01   4.638 4.24e-06 ***
## age                         8.221e-03  5.307e-02   0.155    0.877
## height                      6.194e-02  1.296e-01   0.478    0.633
## weight                      5.056e-02  9.928e-02   0.509    0.611
## income                      1.239e-05  2.131e-05   0.582    0.561
## sexmale                     4.351e-01  2.307e+00   0.189    0.850
## educationprimary           -1.715e-01  3.144e+00  -0.055    0.957
## educationsecondary         -1.451e+00  3.259e+00  -0.445    0.656
## educationtertiary          -8.624e-01  3.308e+00  -0.261    0.794
## place_of_residenceurban    -4.371e-01  2.298e+00  -0.190    0.849
## socioeconomic_statuslow     2.512e+00  2.790e+00   0.900    0.368
## socioeconomic_statusmedium -3.144e+00  2.859e+00  -1.100    0.272
## exerciseyes                 1.951e+00  2.307e+00   0.846    0.398
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.78 on 667 degrees of freedom
## Multiple R-squared:  0.008215,   Adjusted R-squared:  -0.009628
## F-statistic: 0.4604 on 12 and 667 DF,  p-value: 0.9373
```

```r
#loading necessary library
library(dplyr)
library(caret)

predictions <- mlr_model %>%
  predict(SR.test)

#model accuracy for testing data set
data.frame(R2 = R2(predictions, SR.test$diastolic_bp),
           MSE = mean(mlr_model$residuals^2),
           RMSE = RMSE(predictions, SR.test$diastolic_bp),
           MAE = MAE(predictions, SR.test$diastolic_bp))
```

```
##           R2      MSE     RMSE      MAE
## 1 0.02563123 869.9229 27.75389 24.06366
```

```
#3.Fit the multiple linear regression model with Leave One Out Cross-Validation,
#k-fold cross validation, repeated k-fold cross validation methods and get fit
#indices for SR.test data and, compare the fit indices of supervised
#regression models fitted in step 2 and 3 above with careful interpretation

#Leave One Out Cross-Validation
# Define training control
train.control <- trainControl(method = "LOOCV")

# Train the model
model1 <- train(diastolic_bp ~ ., data = SR, method =
                  "lm", trControl = train.control)

#summarize the result
summary(model1)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.298 -25.127   0.743  25.632  56.138
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.009e+02  1.848e+01   5.458 6.08e-08 ***
## age                         2.148e-02  4.310e-02   0.498    0.618
## height                      6.339e-02  1.040e-01   0.609    0.542
## weight                      1.105e-01  8.020e-02   1.377    0.169
## income                      1.540e-05  1.735e-05   0.887    0.375
## sexmale                     1.497e+00  1.856e+00   0.807    0.420
## educationprimary           -5.891e-02  2.597e+00  -0.023    0.982
## educationsecondary         -1.879e+00  2.631e+00  -0.714    0.475
## educationtertiary          -2.406e+00  2.620e+00  -0.918    0.359
## place_of_residenceurban    -9.266e-01  1.852e+00  -0.500    0.617
## socioeconomic_statuslow     3.507e+00  2.239e+00   1.566    0.118
## socioeconomic_statusmedium -3.041e+00  2.283e+00  -1.332    0.183
## exerciseyes                 1.917e+00  1.856e+00   1.033    0.302
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.11 on 987 degrees of freedom
## Multiple R-squared:  0.01412,    Adjusted R-squared:  0.002131
## F-statistic: 1.178 on 12 and 987 DF,  p-value: 0.2941
```

```
print(model1)
```

```
## Linear Regression
##
```

```
## 1000 samples
##    9 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 999, 999, 999, 999, 999, 999, ...
## Resampling results:
##
##   RMSE      Rsquared      MAE
##   29.30333  8.492973e-05  25.59824
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
#Prediction with LOOCV and module accuracy
predictions1 <- model1 %>%
  predict(SR.test)

data.frame(R2 = R2(predictions1, SR.test$diastolic_bp),
           MSE = mean((predictions1 - SR.test$diastolic_bp)^2),
           RMSE = RMSE(predictions1, SR.test$diastolic_bp),
           MAE = MAE(predictions1, SR.test$diastolic_bp))
```

```
##           R2    MSE     RMSE      MAE
## 1 0.03548656 762.16 27.60725 23.97041
```

```r
# Fit multiple linear regression model using k-Fold Cross-Validation (k = 10)
lm_model_kfold <- train(diastolic_bp ~ .,
                        data = SR, method = "lm",
                        trControl = trainControl(method = "cv", number = 10))

#prediction with k fold cross validation
predictions_kfold <- lm_model_kfold %>%
  predict(SR.test)

#getting the required performance indicators
data.frame(R2_kfold = R2(predictions_kfold, SR.test$diastolic_bp),
           MSE_kfold = mean((predictions_kfold - SR.test$diastolic_bp)^2),
           RMSE_kfold = RMSE(predictions_kfold, SR.test$diastolic_bp),
           MAE_kfold = MAE(predictions_kfold, SR.test$diastolic_bp))
```

```
##     R2_kfold MSE_kfold RMSE_kfold MAE_kfold
## 1 0.03548656    762.16   27.60725  23.97041
```

```r
# Fit multiple linear regression model using Repeated k-Fold Cross-Validation
#(k = 10, repeats = 5)
lm_model_repkfold <- train(diastolic_bp ~ .,
                           data = SR, method = "lm",
                           trControl = trainControl(method = "repeatedcv",
                                                    number = 10, repeats = 5))
#getting the prediction using repeated k fold
predictions_repkfold <- lm_model_repkfold %>%
  predict(SR.test)
```

```r
#getting the required performance indicators for repeated k fold
data.frame(R2_repkfold = R2(predictions_repkfold, SR.test$diastolic_bp),
           MSE_repkfold = mean((predictions_repkfold - SR.test$diastolic_bp)^2),
           RMSE_repkfold = RMSE(predictions_repkfold, SR.test$diastolic_bp),
           MAE_repkfold = MAE(predictions_repkfold, SR.test$diastolic_bp))
```

```
##   R2_repkfold MSE_repkfold RMSE_repkfold MAE_repkfold
## 1  0.03548656       762.16      27.60725     23.97041
```

```r
#4. Fit KNN regression, Decision Tree regression, SVM regression and Neural
#Network regression using the same dependent and independent variables, get
#and compare fit indices of these models for SR.test data

# Fit KNN regression
knn_model <- train(diastolic_bp ~ ., data = SR.train,
                   method = "knn", trControl = trainControl(method = "none"))

#prediction using knn
predictions_knn <- knn_model %>%
  predict(SR.test)

#getting the required performance indicators for knn
data.frame(R2_knn = R2(predictions_knn, SR.test$diastolic_bp),
           MSE_knn = mean((predictions_knn - SR.test$diastolic_bp)^2),
           RMSE_knn = RMSE(predictions_knn, SR.test$diastolic_bp),
           MAE_knn = MAE(predictions_knn, SR.test$diastolic_bp))
```

```
##       R2_knn  MSE_knn RMSE_knn MAE_knn
## 1 0.01140901 892.1185 29.86835  24.275
```

```r
# Fit Decision Tree regression
dt_model <- train(diastolic_bp ~ ., data = SR.train,
                  method = "rpart", trControl = trainControl(method = "none"))

#prediction using decision tree
predictions_dt <- dt_model %>%
  predict(SR.test)

#getting the required performance indicators for decision tree
data.frame(R2_dt = R2(predictions_dt, SR.test$diastolic_bp),
           MSE_dt = mean((predictions_dt - SR.test$diastolic_bp)^2),
           RMSE_dt = RMSE(predictions_dt, SR.test$diastolic_bp),
           MAE_dt = MAE(predictions_dt, SR.test$diastolic_bp))
```

```
## Warning in cor(obs, pred, use = ifelse(na.rm, "complete.obs", "everything")): the standard deviation
```

```
##   R2_dt   MSE_dt  RMSE_dt   MAE_dt
## 1    NA 787.7586 28.06704 24.19944
```

```r
library(kernlab)
# Fit SVM regression
svm_model <- train(diastolic_bp ~ ., data = SR.train,
                   method = "svmRadial",
                   trControl = trainControl(method = "none"))

#prediction using SVM
predictions_svm <- svm_model %>%
  predict(SR.test)

#getting the required performance indicators for SVM
data.frame(R2_svm = R2(predictions_svm, SR.test$diastolic_bp),
           MSE_svm = mean((predictions_svm - SR.test$diastolic_bp)^2),
           RMSE_svm = RMSE(predictions_svm, SR.test$diastolic_bp),
           MAE_svm = MAE(predictions_svm, SR.test$diastolic_bp))
```

```
##      R2_svm  MSE_svm RMSE_svm  MAE_svm
## 1 0.0123419 785.2928 28.02308 24.19551
```

```r
# Fit Neural Network regression
nn_model <- train(diastolic_bp ~ ., data = SR.train,
                  method = "nnet", trControl = trainControl(method = "none"))
```

```
## # weights:  15
## initial  value 10727283.906718
## final  value 10657389.000000
## converged
```

```r
#prediction using neural net
predictions_nn <- nn_model %>%
  predict(SR.test)

#getting the required performance indicators for SVM
data.frame(R2_nn = R2(predictions_nn, SR.test$diastolic_bp),
           MSE_nn = mean((predictions_nn - SR.test$diastolic_bp)^2),
           RMSE_nn = RMSE(predictions_nn, SR.test$diastolic_bp),
           MAE_nn = MAE(predictions_nn, SR.test$diastolic_bp))
```

```
## Warning in cor(obs, pred, use = ifelse(na.rm, "complete.obs", "everything")): the standard deviation
```

```
##   R2_nn   MSE_nn  RMSE_nn   MAE_nn
## 1    NA 15347.77 123.8861 120.6688
```

```r
#from multiple linear regression model
# R2          MSE           RMSE         MAE
# 0.02563123  869.9229      27.75389     24.06366

#multiple linear regression model with Leave One Out Cross-Validation
# R2          MSE           RMSE         MAE
# 0.03548656  762.16        27.60725     23.97041
```

```r
# multiple linear regression model using k-Fold Cross-Validation (k = 10)
#R2_kfold       MSE_kfold       RMSE_kfold        MAE_kfold
#0.03548656   762.16          27.60725          23.97041


#multiple linear regression model using Repeated k-Fold Cross-Validation
#R2_repkfold        MSE_repkfold        RMSE_repkfold       MAE_repkfold
#0.03548656         762.16              27.60725            23.97041


#knn regression
#R2_knn     MSE_knn     RMSE_knn      MAE_knn
#0.01140901 892.1185    29.86835      24.275

#decision tree
#R2_dt   MSE_dt     RMSE_dt       MAE_dt
#NA      787.7586   28.06704      24.19944

#svm
#R2_svm     MSE_svm       RMSE_svm      MAE_svm
#0.01192118 786.2485      28.04012      24.19448

#neural network
#R2_nn   MSE_nn      RMSE_nn      MAE_nn
#NA      15347.77    123.8861     120.6688

#the best model has the highest R-Square value and the
#lowest MSE, RMSE, and MAE values.
# from the above analysis the linear model with LOOCV, k- fold CV and
#repeated k fold CV gives the highest R2 value (0.03548656) and lowest
#MSE, RMSE, and MAE values. So these model are the best model for prediction


#Predict diastolic blood pressure of a person with 50 years, 175mm height,
#80 kg weight, 90000 income, male, tertiary level education, living in urban
#area, medium socio-economic status and no exercise and
#interpret the result carefully

# From above we can use any of the model loocv or k fold cv or
#repeated k fold cv Linear Regression model as the best model
# Assuming the Multiple Linear Regression model as the best model
prediction<-predict(model1, newdata = data.frame(age = 50,
                                                 height = 175, weight = 80,
                                                 income = 90000, sex = "male",
                                                 education = "tertiary",
                                                 place_of_residence = "urban",
                                                 socioeconomic_status = "medium",
                                                 exercise = "no"))
prediction  #here the diastolic blood pressure based the given data is 118.389
```

```
##       1
## 118.389
```

```r
prediction_lm_K_fold<-predict(lm_model_kfold, newdata = data.frame(age = 50,
                                                height = 175, weight = 80,
                                                income = 90000, sex = "male",
                                                education = "tertiary",
                                                place_of_residence = "urban",
                                                socioeconomic_status = "medium",
                                                exercise = "no"))
prediction_lm_K_fold
```

```
##       1
## 118.389
```

```r
#7. Write a reflection of the assignment on your own words focusing on
#"what did I learn with this assignment?"

#Here in this assignment we have fitted the multiple linear regression model
#and get the required indicators for the analysis. We randomly create the
#variable data and fit different model and find the best model.

# The best model for predicting diastolic blood pressure is the multiple linear
# regression model with Leave One Out Cross-Validation (LOOCV), k-fold
# cross-validation, or repeated k-fold cross-validation. These models have
# the highest R-squared value and the lowest mean squared error (MSE), root
# mean squared error (RMSE), and mean absolute error (MAE) values compared to
# other models such as KNN regression, decision tree regression,
# SVM regression, and neural network regression.
```