homework01

Jiapeng Wang

2023-09-26

First I give the answer to the question: the submodel height ~ reach + weight has the lowest RMSPE.

Here is my code:

```
# Setting Working Directory & Libraries ------------------------------------
# TODO: Set your working directory
setwd("E:/stat-visp/601")

# Data Preparation ---------------------------------------------------------
# TODO: Read the CSV file and perform any necessary data preparation steps
dat <- read.csv("boxers.csv")[,-1]

# Helper Functions ---------------------------------------------------------
# TODO: Define the generateTheFormula function to generate a model formula based on an integer
generateTheFormula <- function(i){
  varName <- c("reach", "chest.nor", "chest.exp", "weight", "fist")
  varsFlag <- as.integer(rev(intToBits(i)[1:5])) == 1
  currentVars <- varName[varsFlag]
  formulaNow <- as.formula(paste("height~-1+", paste(c("1",currentVars), collapse = "+"), sep = ""))
  return(formulaNow)
}

# Modeling: for a single model ---------------------------------------------

# TODO: Use the generateTheFormula function to get the current formula
formulaNow <- generateTheFormula(10)
error <- numeric(nrow(dat)) # Placeholder for saving the error

# TODO: Write a loop to calculate error for each observation

for(i in seq_len(nrow(dat))){
  # TODO: Fit the model and make predictions
  dat_leave_one <- dat[-i, ]
  model_lm <- lm(formulaNow, data = dat_leave_one)
  predicted_dat <- predict(model_lm, newdata = dat[i, ])
  error[i] <- abs(predicted_dat - dat$height[i])
}
# TODO: Compute the RMPSE for the current model
RMPSE <- sqrt(1 / nrow(dat) * sum(error^2))

  # A simpler way to write code...
  # TODO: Rewrite the for loop using sapply
```

```r
error <- sapply(seq_len(nrow(dat)), function(i) {
  dat_leave_one <- dat[-i, ]
  model_lm <- lm(formulaNow, data = dat_leave_one)
  predicted_height <- predict(model_lm, newdata = dat[i, ])
  return(abs(predicted_height - dat$height[i]))
})
RMPSE_simplified <- sqrt(1 / nrow(dat) * sum(error^2))
# Compute the RMPSE for the simplified version

  # No-brainer ...
  # TODO: Load the necessary library and fit the model using caret

library(caret)
# TODO: Fit the model using the train function from caret package and return RMSE
modelNow <- train(formulaNow, data = dat, method = "lm", trControl = trainControl(method = "LOOCV"))
RMSE_caret <- modelNow$results$RMSE

# Modeling: for all models ------------------------------------------

# TODO: Write code to fit all models and return RMSE for each

res <- sapply(1:31, function(i) {
  # TODO: Fit the model and return RMSE for each
  formulaNow <- generateTheFormula(i)
  error <- sapply(seq_len(nrow(dat)), function(j) {
    dat_leave_one <- dat[-j, ]
    model_lm <- lm(formulaNow, data = dat_leave_one)
    predicted_height <- predict(model_lm, newdata = dat[j, ])
    return(abs(predicted_height - dat$height[j]))
  })
  return(sqrt(1 / nrow(dat) * sum(error^2)))
})

# TODO: Find the model with minimum RMSE and print its formula

min_model <- which.min(res)
print(generateTheFormula(min_model))
```

```
## height ~ -1 + 1 + reach + weight
## <environment: 0x000002dc4ff70da0>
```