R.r

rajanish

Wed May 16 19:24:23 2018

Simple package to build a Neural Network classifier for the Bank sales data.

@return confusion matrix table @export fn_nnet

@examples fn_nnet()

```
fn_nnet <- function() {
  #require("caret")
  suppressWarnings(suppressMessages(library("caret")))

  # Read the csv file and convert the dependent variable to binary class
  raw_data <-
    read.csv("/home/rajanish/Berkeley/DataScience/BankSales/data/bank.csv")
  raw_data$y <- ifelse(raw_data$y == "yes", 1, 0)

  # Conver the categorical data to one hot vectors and dependent variable to factors
  dmy <- dummyVars(" ~ .", data = raw_data, fullRank = T)
  vector_data <- data.frame(predict(dmy, newdata = raw_data))
  vector_data$y <- as.factor(vector_data$y)

  # Split the available data into Train and Test data
  ind <-
    sample(2,
           nrow(vector_data),
           replace = TRUE,
           prob = c(0.8, 0.2))
  train_data <- vector_data[ind == 1,]
  test_data <- vector_data[ind == 2,]

  # Build a Neural Network classification model fitControl <- trainControl(method = 'repeatedcv', numbe
  # 5, repeats = 5)
  features <- names(train_data)[!names(train_data) %in% "y"]
  model_nnet <-
    train(
      x = train_data[, features],
      y = train_data[, "y"],
      method = "nnet"  #,trControl=fitControl
      ,
      tuneLength = 2,
      maxit = 100,
      trace = FALSE
    )  ### increase the iterations later

  # Save the model to the project data folder
  model_file = "/home/rajanish/Berkeley/DataScience/BankSales/data/model_nnet.Rdata"
  save(model_nnet, file = model_file)
```

```
  # Use the model to predict dependent variable of test data
  predictions <-
    predict.train(object = model_nnet, test_data[, features], type = "raw")

  # Review Confusion Matrix and statistics
  conf_matrix <- confusionMatrix(predictions, test_data[, "y"])
  #print('The result')
  #print(conf_matrix$table)
  return (conf_matrix$table)
}
```