randomForest—GenderClassificationProject.R

Jan

2023-07-26

```r
# Wczytanie potrzebnych bibliotek
library(caret)
```

```
## Warning: pakiet 'caret' został zbudowany w wersji R 4.3.1
```

```
## Ładowanie wymaganego pakietu: ggplot2
```

```
## Ładowanie wymaganego pakietu: lattice
```

```r
library(randomForest)
```

```
## Warning: pakiet 'randomForest' został zbudowany w wersji R 4.3.1
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Dołączanie pakietu: 'randomForest'
```

```
## Następujący obiekt został zakryty z 'package:ggplot2':
##
##     margin
```

```r
#Ustawienie ziarna generatora, po to żeby później porónać wyniki
set.seed(10)

# Wczytywanie danych z pliku csv oraz obejrzenie ich
Data <- read.csv("GenderClassification.csv", stringsAsFactors = TRUE)
View(Data)
str(Data)
```

```
## 'data.frame':    66 obs. of  5 variables:
##  $ Favorite.Color      : Factor w/ 3 levels "Cool","Neutral",..: 1 2 3 3 1 3 1 3 3 2 ...
##  $ Favorite.Music.Genre: Factor w/ 7 levels "Electronic","Folk/Traditional",..: 7 3 7 2 7 4 5 5 7 5
##  $ Favorite.Beverage   : Factor w/ 6 levels "Beer","Doesn't drink",..: 4 4 6 5 4 2 1 5 3 6 ...
##  $ Favorite.Soft.Drink : Factor w/ 4 levels "7UP/Sprite","Coca Cola/Pepsi",..: 1 2 2 3 2 3 2 3 1 2 .
##  $ Gender              : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
```

```r
#Zamiana typu zmiennych z factor na numeric(cleaning data), ponieważ ten typ jest wygodniejszy
Data$Favorite.Color <- as.numeric(Data$Favorite.Color)
Data$Favorite.Music.Genre <- as.numeric(Data$Favorite.Music.Genre)
Data$Favorite.Beverage <- as.numeric(Data$Favorite.Beverage)
Data$Favorite.Soft.Drink <- as.numeric(Data$Favorite.Soft.Drink)

#Skorzystanie z createDataPartion(library(caret)), żeby podzielić dane na training data i test data
TrainingDataSize <- createDataPartition(Data$Gender,
                                        p = 0.8,
                                        list = FALSE)
TrainingData <- Data[TrainingDataSize,]
TestData <- Data[-TrainingDataSize,]

#Korzystanie z algorytmu randomForest(drzew decyzyjnych)(library(randomForest))
modelr <- randomForest(formula = Gender ~ .,
                       data = Data)
#Przedstawienie modelu
print(modelr)
```

```
##
## Call:
##  randomForest(formula = Gender ~ ., data = Data)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 42.42%
## Confusion matrix:
##    F  M class.error
## F 21 12   0.3636364
## M 16 17   0.4848485
```