Classification K-Nearest-Neighbors

Chris Bentz

12/01/2023

Load Packages

If the libraries are not installed yet, you need to install them using, for example, the command: install.packages("ggplot2"). For the Hrate package this is different, since it comes from github. The devtools library needs to be installed, and then the install_github() function is used.

```
library(ggplot2)
library(dplyr)
library(class)
library(gridExtra)
library(gmodels)
library(caret)
```

Load Data

Load data table with values per text file.

```
# load estimations from stringBase corpus
estimations.df <- read.csv("~/Github/NaLaFi/results/estimation10chars.csv")
# alternatively: "~/Github/NaLaFi/results/estimation100chars.csv"
# "~/Github/NaLaFi/results/estimation1000chars.csv"
#head(estimations10.df.)
```

Select relevant columns of the data frame, i.e. the measures to be included in classification and the ''corpus" or ''subcorpus" column.

```
estimations.subset <- estimations.df[c("corpus", "subcorpus", "huni.chars", "hrate.chars", "ttr.chars",
```

Remove NAs (whole row)

```
estimations.subset <- na.omit(estimations.subset)
```

Center and scale the data

```
estimations.scaled <- cbind(estimations.subset[1:2], scale(estimations.subset[3:ncol(estimations.subset)
```

Create Training and Test Sets

```
# Generating seed
set.seed(1234)
# Randomly generating our training and test samples with a respective ratio of 2/3 and 1/3
datasample <- sample(2, nrow(estimations.scaled), replace = TRUE, prob = c(0.67, 0.33))
# Generate training set
estimations.training <- estimations.scaled[datasample == 1, 3:ncol(estimations.scaled)]
# Generate test set
estimations.test <- estimations.scaled[datasample == 2, 3:ncol(estimations.scaled)]
```

Get training and test labels

```
# Generate training labels
training.labels <- estimations.scaled[datasample == 1, 1]
# Generate test labels
test.labels <- estimations.scaled[datasample == 2, 1]
```

Building knn classifier

```
# choose k
k = 4
estimations.knn <- knn(train = estimations.training, test = estimations.test, cl = training.labels, k =
```

Model Evaluation

```
# creating a dataframe from known (true) test labels
test.labels <- data.frame(test.labels)

# combining predicted and known classes
class.comparison <- data.frame(estimations.knn, test.labels)

# giving appropriate column names
names(class.comparison) <- c("predicted", "observed")

# inspecting our results table
head(class.comparison)
```

```
##     predicted    observed
## 1     writing non-writing
## 2 non-writing non-writing
## 3     writing non-writing
## 4     writing non-writing
## 5 non-writing non-writing
## 6     writing non-writing
```

```r
# get confusion matrix
cm <- confusionMatrix(class.comparison$predicted,
                      reference = class.comparison$observed,
                      positive = "writing")
print(cm)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    non-writing writing
##   non-writing          15       9
##   writing              26     290
##
##                Accuracy : 0.8971
##                  95% CI : (0.8597, 0.9272)
##     No Information Rate : 0.8794
##     P-Value [Acc > NIR] : 0.180637
##
##                   Kappa : 0.4089
##
##  Mcnemar's Test P-Value : 0.006841
##
##             Sensitivity : 0.9699
##             Specificity : 0.3659
##          Pos Pred Value : 0.9177
##          Neg Pred Value : 0.6250
##              Prevalence : 0.8794
##          Detection Rate : 0.8529
##    Detection Prevalence : 0.9294
##       Balanced Accuracy : 0.6679
##
##        'Positive' Class : writing
##
```

```r
# get precision, recall, and f1 from the output list of confusionMatrix()
f1 <- cm[["byClass"]]["F1"]
recall <- cm[["byClass"]]["Recall"]
precision <- cm[["byClass"]]["Precision"]
```

Prepare and safe data frame with results

```r
# combine in data frame
knn.results <- data.frame(k, precision, recall, f1, row.names = NULL)
knn.results.rounded <- round(knn.results, 2)
print(knn.results.rounded)
```

```
##   k precision recall   f1
## 1 4      0.92   0.97 0.94
```

Write to file.

```r
write.csv(knn.results.rounded, file = "~/Github/NaLaFi/results/knn/knn_results.csv")
```