Classification K-Nearest-Neighbors

Chris Bentz

14/10/2023

Description

k-nearest neighbor analyses of the feature vectors per character string (loaded from NaLaFi/results/features.csv). The results are stored in NaLaFi/results/KNN. Note that the number of characters has to be chosen manually (via num.char = ““), likewise, the features to be included can be chosen in the lines defining”estimations.subset” below. Also, subcorpora can be excluded via the “selected” object.

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)
library(ggExtra)
library(ggpubr)
```

Load Data

Load data table with values per text file.

```
# load estimations from stringBase corpus
estimations.df <- read.csv("~/Github/NaLaFi/results/features.csv")
#head(features.csv)
```

Exclude subcorpora (if needed). Choose the subcorpus to be excluded via “selected”.

```
selected <- c(#"heraldics",
              #"procunei",
              #"animal",
              #"random",
              #"morse",
              #"shuffled",
              #"weather",
              #"natural",
              #"pycode"
              )
estimations.df <- estimations.df[!(estimations.df$subcorpus %in% selected), ]
```

Split into separate files by length of chunks in characters.

```
# choose number of characters
num.char = 100
# subset data frame
estimations.df <- estimations.df[estimations.df$num.char == num.char, ]
nrow(estimations.df)
```

```
## [1] 3223
```

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",
                                       "rm.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
nrow(estimations.scaled)
```

```
## [1] 3223
```

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)]
nrow(estimations.training)
```

```
## [1] 2194
```

```
# Generate test set
estimations.test <- estimations.scaled[datasample == 2, 3:ncol(estimations.scaled)]
nrow(estimations.test)
```

```
## [1] 1029
```

Get training and test labels

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

Initialize data frame

```r
knn.results <- data.frame(k = numeric(0), accuracy = numeric(0), precision = numeric(0),
                          recall = numeric(0), f1 = numeric(0))
```

Building knn classifier

```r
# choose maximum number of neighbors n
n = 10
# run a loop over different numbers of neighbors up to n
for (k in 1:n){
  # knn estimation of labels
  predictions.knn <- knn(train = as.data.frame(estimations.training),
                         test = as.data.frame(estimations.test),
                         cl = training.labels, k = 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(predictions.knn, test.labels)
  # giving appropriate column names
  names(class.comparison) <- c("predicted", "observed")
  # inspecting our results table
  head(class.comparison)
  # get confusion matrix
  cm <- confusionMatrix(data = class.comparison$predicted,
                        reference = as.factor(class.comparison$observed))
  print(cm)
  # get precision, recall, and f1 from the output list of confusionMatrix()
  accuracy <- cm$overall['Accuracy']
  f1 <- cm[["byClass"]]["F1"]
  recall <- cm[["byClass"]]["Recall"]
  precision <- cm[["byClass"]]["Precision"]

  # prepare data frame with results
  local.results <- data.frame(k, accuracy, precision, recall, f1, row.names = NULL)
  local.results.rounded <- round(local.results, 2)
  # print(local.results.rounded)
  knn.results <- rbind(knn.results, local.results.rounded)
}
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction    non-writing writing
##   non-writing         412     110
##   writing             104     403
##
##                 Accuracy : 0.792
##                   95% CI : (0.7659, 0.8165)
##      No Information Rate : 0.5015
##      P-Value [Acc > NIR] : <2e-16
##
##                    Kappa : 0.584
##
##   Mcnemar's Test P-Value : 0.7325
##
##              Sensitivity : 0.7984
##              Specificity : 0.7856
##           Pos Pred Value : 0.7893
##           Neg Pred Value : 0.7949
##               Prevalence : 0.5015
##           Detection Rate : 0.4004
##     Detection Prevalence : 0.5073
##        Balanced Accuracy : 0.7920
##
##         'Positive' Class : non-writing
##
## Confusion Matrix and Statistics
##
##               Reference
## Prediction    non-writing writing
##   non-writing         412     112
##   writing             104     401
##
##                 Accuracy : 0.7901
##                   95% CI : (0.7639, 0.8146)
##      No Information Rate : 0.5015
##      P-Value [Acc > NIR] : <2e-16
##
##                    Kappa : 0.5802
##
##   Mcnemar's Test P-Value : 0.6339
##
##              Sensitivity : 0.7984
##              Specificity : 0.7817
##           Pos Pred Value : 0.7863
##           Neg Pred Value : 0.7941
##               Prevalence : 0.5015
##           Detection Rate : 0.4004
##     Detection Prevalence : 0.5092
##        Balanced Accuracy : 0.7901
##
##         'Positive' Class : non-writing
##
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    non-writing writing
##   non-writing         430     108
##   writing              86     405
##
##                Accuracy : 0.8115
##                  95% CI : (0.7862, 0.8349)
##     No Information Rate : 0.5015
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6229
##
##  Mcnemar's Test P-Value : 0.1316
##
##             Sensitivity : 0.8333
##             Specificity : 0.7895
##          Pos Pred Value : 0.7993
##          Neg Pred Value : 0.8248
##              Prevalence : 0.5015
##          Detection Rate : 0.4179
##    Detection Prevalence : 0.5228
##       Balanced Accuracy : 0.8114
##
##        'Positive' Class : non-writing
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    non-writing writing
##   non-writing         432     108
##   writing              84     405
##
##                Accuracy : 0.8134
##                  95% CI : (0.7882, 0.8368)
##     No Information Rate : 0.5015
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6268
##
##  Mcnemar's Test P-Value : 0.09694
##
##             Sensitivity : 0.8372
##             Specificity : 0.7895
##          Pos Pred Value : 0.8000
##          Neg Pred Value : 0.8282
##              Prevalence : 0.5015
##          Detection Rate : 0.4198
##    Detection Prevalence : 0.5248
##       Balanced Accuracy : 0.8133
##
##        'Positive' Class : non-writing
##
```

```
## Confusion Matrix and Statistics
## 
##               Reference
## Prediction    non-writing writing
##   non-writing         433     109
##   writing              83     404
##                                           
##                Accuracy : 0.8134          
##                  95% CI : (0.7882, 0.8368)
##     No Information Rate : 0.5015          
##     P-Value [Acc > NIR] : <2e-16          
##                                           
##                   Kappa : 0.6268          
##                                           
##  Mcnemar's Test P-Value : 0.0712          
##                                           
##             Sensitivity : 0.8391          
##             Specificity : 0.7875          
##          Pos Pred Value : 0.7989          
##          Neg Pred Value : 0.8296          
##              Prevalence : 0.5015          
##          Detection Rate : 0.4208          
##    Detection Prevalence : 0.5267          
##       Balanced Accuracy : 0.8133          
##                                           
##        'Positive' Class : non-writing     
## 
## Confusion Matrix and Statistics
## 
##               Reference
## Prediction    non-writing writing
##   non-writing         433     107
##   writing              83     406
##                                           
##                Accuracy : 0.8154          
##                  95% CI : (0.7903, 0.8386)
##     No Information Rate : 0.5015          
##     P-Value [Acc > NIR] : <2e-16          
##                                           
##                   Kappa : 0.6307          
##                                           
##  Mcnemar's Test P-Value : 0.0952          
##                                           
##             Sensitivity : 0.8391          
##             Specificity : 0.7914          
##          Pos Pred Value : 0.8019          
##          Neg Pred Value : 0.8303          
##              Prevalence : 0.5015          
##          Detection Rate : 0.4208          
##    Detection Prevalence : 0.5248          
##       Balanced Accuracy : 0.8153          
##                                           
##        'Positive' Class : non-writing     
## 
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    non-writing writing
##   non-writing         442     112
##   writing              74     401
##
##                Accuracy : 0.8192
##                  95% CI : (0.7943, 0.8423)
##     No Information Rate : 0.5015
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6384
##
##  Mcnemar's Test P-Value : 0.006668
##
##             Sensitivity : 0.8566
##             Specificity : 0.7817
##          Pos Pred Value : 0.7978
##          Neg Pred Value : 0.8442
##              Prevalence : 0.5015
##          Detection Rate : 0.4295
##    Detection Prevalence : 0.5384
##       Balanced Accuracy : 0.8191
##
##        'Positive' Class : non-writing
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    non-writing writing
##   non-writing         440     109
##   writing              76     404
##
##                Accuracy : 0.8202
##                  95% CI : (0.7954, 0.8432)
##     No Information Rate : 0.5015
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6404
##
##  Mcnemar's Test P-Value : 0.01864
##
##             Sensitivity : 0.8527
##             Specificity : 0.7875
##          Pos Pred Value : 0.8015
##          Neg Pred Value : 0.8417
##              Prevalence : 0.5015
##          Detection Rate : 0.4276
##    Detection Prevalence : 0.5335
##       Balanced Accuracy : 0.8201
##
##        'Positive' Class : non-writing
##
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    non-writing writing
##   non-writing         448     113
##   writing              68     400
##
##                Accuracy : 0.8241
##                  95% CI : (0.7994, 0.8469)
##     No Information Rate : 0.5015
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6481
##
##  Mcnemar's Test P-Value : 0.001074
##
##             Sensitivity : 0.8682
##             Specificity : 0.7797
##          Pos Pred Value : 0.7986
##          Neg Pred Value : 0.8547
##              Prevalence : 0.5015
##          Detection Rate : 0.4354
##    Detection Prevalence : 0.5452
##       Balanced Accuracy : 0.8240
##
##        'Positive' Class : non-writing
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    non-writing writing
##   non-writing         447     115
##   writing              69     398
##
##                Accuracy : 0.8212
##                  95% CI : (0.7964, 0.8441)
##     No Information Rate : 0.5015
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6423
##
##  Mcnemar's Test P-Value : 0.0009085
##
##             Sensitivity : 0.8663
##             Specificity : 0.7758
##          Pos Pred Value : 0.7954
##          Neg Pred Value : 0.8522
##              Prevalence : 0.5015
##          Detection Rate : 0.4344
##    Detection Prevalence : 0.5462
##       Balanced Accuracy : 0.8211
##
##        'Positive' Class : non-writing
##
```

```
print(knn.results)
```

```
##     k accuracy precision recall   f1
## 1   1     0.79      0.79   0.80 0.79
## 2   2     0.79      0.79   0.80 0.79
## 3   3     0.81      0.80   0.83 0.82
## 4   4     0.81      0.80   0.84 0.82
## 5   5     0.81      0.80   0.84 0.82
## 6   6     0.82      0.80   0.84 0.82
## 7   7     0.82      0.80   0.86 0.83
## 8   8     0.82      0.80   0.85 0.83
## 9   9     0.82      0.80   0.87 0.83
## 10 10     0.82      0.80   0.87 0.83
```

Write to file. Note that the file names have to be changed manually here.

```
write.csv(knn.results, file = paste("~/Github/NaLaFi/results/KNN/knn_results_ttr_rm",
                                    paste(num.char, ".csv", sep =""),
                                    sep =""), row.names = F)
```