

# Classification with Logistic Regression

Chris Bentz

16/01/2023

## 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(dplyr)
library(class)
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/features.csv")
#head(features.csv)
```

Exclude subcorpora (if needed).

```
#selected <- c("natural")
#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 = 1000
# subset data frame
estimations.df <- estimations.df[estimations.df$num.char == num.char, ]
```

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",
                                         "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], scale(estimations.subset[2: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
train <- estimations.scaled[datasample == 1, 1:ncol(estimations.scaled)]
# Generate test set
test <- estimations.scaled[datasample == 2, 1:ncol(estimations.scaled)]
```

## Building logistic regression model

The following code to run a logistic regression is adopted from <https://datasciencedojo.com/blog/logistic-regression-in-r-tutorial/> (last accessed 16.01.2023).

```
# logistic regression estimation of labels
log.model <- glm(corpus ~ ., data = train, family = binomial(link = "logit"))
summary(log.model)
```

```
##
## Call:
## glm(formula = corpus ~ ., family = binomial(link = "logit"),
##      data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -5.0865   0.0010   0.2399   0.3453   1.8896
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.1691     0.2767   7.839 4.54e-15 ***
## huni.chars     0.6460     0.5340   1.210 0.226407
## hrate.chars  -1.9298     0.3584  -5.385 7.25e-08 ***
## ttr.chars     3.3325     0.9289   3.588 0.000334 ***
## rm.chars     -3.3707     0.4078  -8.265 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1367.48  on 1243  degrees of freedom
```

```
## Residual deviance: 498.62 on 1239 degrees of freedom
## AIC: 508.62
##
## Number of Fisher Scoring iterations: 8
```

## Prediction

```
log.predictions <- predict(log.model, test, type = "response")
head(log.predictions)
```

```
##      6430      6436      6439      6441      6451      6453
## 0.8069365 0.9411588 0.9432922 0.9460837 0.9454989 0.9186915
```

Assign a label according to the rule that the label is “non-writing” if the prediction probability is  $>0.5$ , else assign “writing”.

```
log.prediction.rd <- ifelse(log.predictions > 0.5, "writing", "non-writing")
head(log.prediction.rd, 10)
```

```
##      6430      6436      6439      6441      6451
## "writing" "writing" "writing" "writing" "writing"
##      6453      6454      6461      6464      6465
## "writing" "writing" "non-writing" "non-writing" "non-writing"
```

## Model evaluation

```
# creating a dataframe from known (true) test labels
test.labels <- data.frame(test$corpus)
# combining predicted and known classes
class.comparison <- data.frame(log.prediction.rd, test.labels)
# giving appropriate column names
names(class.comparison) <- c("predicted", "observed")
# inspecting our results table
head(class.comparison)
```

```
##      predicted observed
## 6430    writing non-writing
## 6436    writing non-writing
## 6439    writing non-writing
## 6441    writing non-writing
## 6451    writing non-writing
## 6453    writing non-writing
```

```
# get confusion matrix
cm <- confusionMatrix(class.comparison$predicted,
                      reference = class.comparison$observed)
# print(cm)
# get precision, recall, and f1 from the output list of confusionMatrix()
f1 <- cm[["byClass"]][["F1"]]
recall <- cm[["byClass"]][["Recall"]]
precision <- cm[["byClass"]][["Precision"]]
```

```
# prepare data frame with results
```

```
lr.results <- data.frame(precision, recall, f1, row.names = NULL)
```

```
lr.results.rounded <- round(lr.results, 2)
```

```
print(lr.results)
```

```
## precision recall f1
```

```
## 1 0.9423077 0.7205882 0.8166667
```

Write to file.

```
write.csv(lr.results, file = "~/Github/NaLaFi/results/LR/results_logReg.csv", row.names = F)
```