Aktueller Stand

David Fuhry

24/02/2021

Korpus

Der verwendete Korpus enthält Texte von sieben als konspirativ zu betrachtenden Websites sowie als Vergleichsdatensatz von je zwei journalistischen bzw. wissenschaftsjournalistischen Quellen.

Er setzt sich wie folgt zusammen:

site	dataset	n	mean_nchar
Alles Schall und Rauch	conspiracy	5351	5001
conrebbi	conspiracy	24	5609
deutschlandpranger	conspiracy	118	7527
fm-tv	conspiracy	128	9666
Frankfurter Rundschau	control	9987	3672
hinterderfichte	conspiracy	1083	4221
recentr	conspiracy	5041	4861
scienceblogs	control	9547	3254
scilogs	control	9546	5202
Spiegel Online	control	8673	4688
Watergate.tv	conspiracy	8123	2858

Die Verteilung in Bezug auf die Zielklasse ist mit etwa 1:2 nicht unbedingt super, aber sollte wohl passen.

Das Bereinigen ist aktuell relativ simpel gehalten, ich filtere u.a.:

- Nicht deutschsprachige Texte
- Sehr kurze Texte
- Wörter die komplett aus sehr selten vorkommenden Sonderzeichen bestehen
  - Hier sind wohl meist eher Fehler bei der Extraktion aus HTML Schuld
- Kleineres wie unterschiedliche Whitespaces, horizontale Linien, Unicode normalisierung, etc.

Ich filtere aktuell nicht auf Wortebene Stopwords o.ä., da die Features alle eher Stilbasiert sind.

Model mit Features aus der Literatur

Für ein erstes Modell habe ich nur Merkmale die in der qualitativen Literatur zu finden waren herangezogen, diese sind:

- Sentimentwerte
  - Als Summe, absolute Summe, Summe positiver Werte und Summe negativer Sentimente
- Zahl der Fragezeichen
- Zahl der Anführungszeichen
- Zahl von ‘scare quotes’, definiert als einzelnes Wort in Anführungszeichen#
- Durchschnitliche Länge von Zitaten
  - Nur Top-Level Zitate

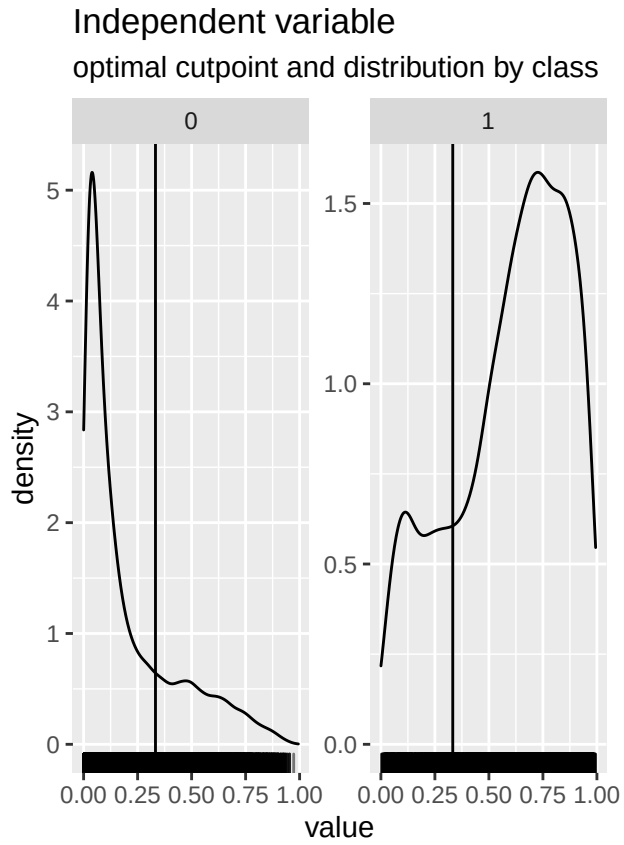
- Anteil von Zitaten am Gesamttext
- Anzahl von Zahlenangaben
- Anzahl von Negationswörtern

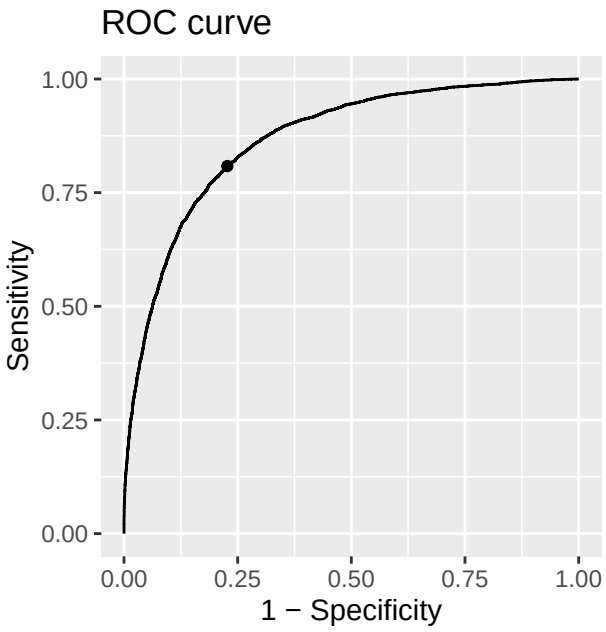
Es gibt noch ein paar in der Literatur genannte Eigenschaften von denen ich denke, dass sie prinzipiell automatisiert auswertbar wären, aber bei denen ich keine gute Lösung finden konte. So etwa die Anzahl von Modalwörtern, die Anzahl von passiven Verben und die Anzahl von Einheitenbezeichnungen.

Für die Sentimentanalyse nutze ich SentiWS mit Matching über Lemma und POS-Tags, die wiederrum aus Annotation mit spacy stammen. Alle Features wurden mit der Textlänge skaliert.

```
library(lightgbm)

set.seed(42)

inTraining <- as.vector(caret::createDataPartition(features$is_conspiracy, p = .66, list = FALSE))
training <- features[inTraining,]
validate  <- features[-inTraining,]

target <- training$is_conspiracy
training <- Matrix::Matrix(as.matrix(training[, -3:-1]), sparse = TRUE)

validate_target <- validate$is_conspiracy
validate <- Matrix::Matrix(as.matrix(validate[, -3:-1]), sparse = TRUE)

training <- lgb.Dataset(data = training, label = target)

pars = list(objective = "binary",
            learning_rate = 0.05,
            num_iterations = 10000L,
            max_depth = -1L,
            num_leaves = 50L,
            early_stopping_round = 10L)

lgb_test <- lgb.cv(params = pars,
                   data = training,
                   nfold = 10L)

pars$num_iterations <- round(lgb_test$best_iter + (lgb_test$best_iter / 10))

model <- lightgbm(data = training,
                  params = pars)

predicted <- predict(model, data = validate)
```

Das entstehende Model ist dann so irgendwie mittelgut, hier anhand der ROC-Kurve:





Sowie der confusion Matrix:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 9937 1289
##          1 2923 5442
##
##                Accuracy : 0.785
##                  95% CI : (0.7792, 0.7907)
##     No Information Rate : 0.6564
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5494
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8085
##             Specificity : 0.7727
##          Pos Pred Value : 0.6506
##          Neg Pred Value : 0.8852
##              Prevalence : 0.3436
##          Detection Rate : 0.2778
##    Detection Prevalence : 0.4270
##       Balanced Accuracy : 0.7906
##
```

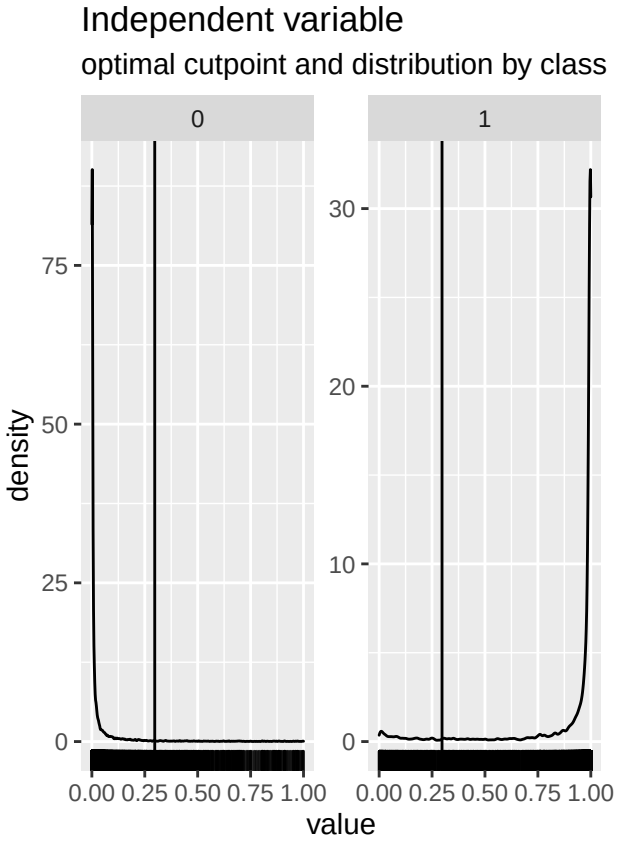
```
##        'Positive' Class : 1
##
```

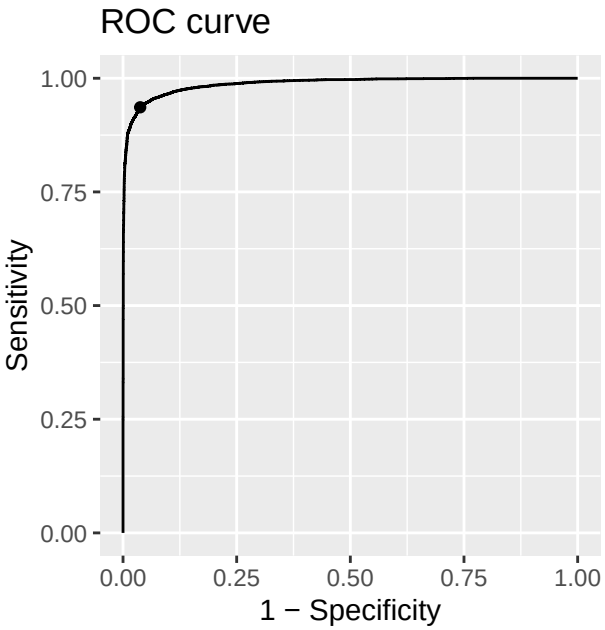
Erweitertes Modell

Für ein erweitertes Modell habe ich dann noch einige Features hinzugenommen:

- POS Tags
- Anteil von Sonderzeichen
- Anteil der Großbuchstaben
- Durchschnittliche Wortlänge

Das Modell wurde dann mit den gleichen Parametern trainiert wie das erste (Hyperparameter tuning steht noch an) und liefert deutlich bessere Ergebnisse:






```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 12374   430
##          1   486  6301
##
##                Accuracy : 0.9532
##                  95% CI : (0.9502, 0.9562)
##     No Information Rate : 0.6564
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8965
```

```
##
##  Mcnemar's Test P-Value : 0.06918
##
##             Sensitivity : 0.9361
##             Specificity : 0.9622
##          Pos Pred Value : 0.9284
##          Neg Pred Value : 0.9664
##              Prevalence : 0.3436
##          Detection Rate : 0.3216
##    Detection Prevalence : 0.3464
##       Balanced Accuracy : 0.9492
##
##        'Positive' Class : 1
##
```