WST3 - Assignment 2 - Classification

Als Programmiersprache haben wir Python verwendet und für die verschiedenen Klassifizierer wurde das Paket sklearn herangezogen. Wir haben uns nicht auf binäre Klassifizierer beschränkt (wie in der Angabe gefordert, sondern versuchen direkt alle Klassen zu klassifizieren. Dieses Vorgehen wirkt sich natürlich auf die Qualität der Ergebnisse aus, da die Klassifizierer weniger genau sind. Für die Kennzahlen haben wir immer alle Klassen einzeln betrachtet.

Team

- Dominik Blaha - S1910454002
- David Piringer - S1910454024

Ausgabe des Python-Skripts

```
## Processing input ...
length of ctd = 118
length of ttd = 118
## Splitting training and test dataset...
length of training ctd = 82
length of testing ctd = 36
length of training ttd = 82
length of testing ttd = 36
first member of training dataset:
    ['biological and chemical basics', '0', '0', '0', '1']
first member of testing dataset:
    ['basics of statistics', '1', '0', '0', '0']
## Evaluating classification performance using the Rocchio classifier ...
~~~~~~~ Rocchio (euclidean) ~~~~~~~
########## TRAIN ##########
------- CLASS 0 -------
  Precision: 0.45
     Recall: 0.375
  F-Measure: 0.4090909090909091
-----------------------
------- CLASS 1 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 2 -------
  Precision: 0.043478260869565216
     Recall: 0.2
  F-Measure: 0.07142857142857142
-----------------------
------- CLASS 3 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
```

```
-----------------------
########## TEST ##########
------- CLASS 0 -------
  Precision: 0.71875
     Recall: 0.9583333333333334
  F-Measure: 0.8214285714285714
-----------------------
------- CLASS 1 -------
  Precision: 0.125
     Recall: 0.25
  F-Measure: 0.16666666666666666
-----------------------
------- CLASS 2 -------
  Precision: 0.2222222222222222
     Recall: 0.4
  F-Measure: 0.2857142857142857
-----------------------
------- CLASS 3 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
~~~~~~~ Rocchio (manhatten) ~~~~~~~
########## TRAIN ##########
------- CLASS 0 -------
  Precision: 0.45
     Recall: 0.375
  F-Measure: 0.4090909090909091
-----------------------
------- CLASS 1 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 2 -------
  Precision: 0.043478260869565216
     Recall: 0.2
  F-Measure: 0.07142857142857142
-----------------------
------- CLASS 3 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
########## TEST ##########
------- CLASS 0 -------
  Precision: 0.5714285714285714
     Recall: 0.5
  F-Measure: 0.5333333333333333
-----------------------
------- CLASS 1 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
```

```
-----------------------
------- CLASS 2 -------
  Precision: 0.1
     Recall: 0.4
  F-Measure: 0.16000000000000003
-----------------------
------- CLASS 3 -------
  Precision: 0.05
     Recall: 0.3333333333333333
  F-Measure: 0.08695652173913045
-----------------------
## Evaluating classification performance using the kNN classifier ...
~~~~~~~ kNN (n_neighbors = 1) ~~~~~~~
########## TRAIN ##########
------- CLASS 0 -------
  Precision: 0.45
     Recall: 0.375
  F-Measure: 0.4090909090909091
-----------------------
------- CLASS 1 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 2 -------
  Precision: 0.043478260869565216
     Recall: 0.2
  F-Measure: 0.07142857142857142
-----------------------
------- CLASS 3 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
########## TEST ##########
------- CLASS 0 -------
  Precision: 0.5
     Recall: 0.3333333333333333
  F-Measure: 0.4
-----------------------
------- CLASS 1 -------
  Precision: 0.14285714285714285
     Recall: 1.0
  F-Measure: 0.25
-----------------------
------- CLASS 2 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 3 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
```

```
-----------------------
~~~~~~~ kNN (n_neighbors = 3) ~~~~~~~
########## TRAIN ##########
------- CLASS 0 -------
  Precision: 0.47619047619047616
     Recall: 0.4166666666666667
  F-Measure: 0.4444444444444445
-----------------------
------- CLASS 1 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 2 -------
  Precision: 0.045454545454545456
     Recall: 0.2
  F-Measure: 0.07407407407407407
-----------------------
------- CLASS 3 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
########## TEST ##########
------- CLASS 0 -------
  Precision: 0.6451612903225806
     Recall: 0.8333333333333334
  F-Measure: 0.7272727272727272
-----------------------
------- CLASS 1 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 2 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 3 -------
  Precision: 0.07142857142857142
     Recall: 0.3333333333333333
  F-Measure: 0.11764705882352941
-----------------------
~~~~~~~ kNN (n_neighbors = 10) ~~~~~~~
########## TRAIN ##########
------- CLASS 0 -------
  Precision: 0.6333333333333333
     Recall: 0.7916666666666666
  F-Measure: 0.7037037037037038
-----------------------
------- CLASS 1 -------
  Precision: 0.0
     Recall: 0.0
```

```
  F-Measure: 0.0
-----------------------
------- CLASS 2 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 3 -------
  Precision: 0.06666666666666667
     Recall: 0.3333333333333333
  F-Measure: 0.1111111111111111
-----------------------
########## TEST ##########
------- CLASS 0 -------
  Precision: 0.6666666666666666
     Recall: 1.0
  F-Measure: 0.8
-----------------------
------- CLASS 1 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 2 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 3 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
~~~~~~~ kNN (n_neighbors = 15) ~~~~~~~
########## TRAIN ##########
------- CLASS 0 -------
  Precision: 0.6470588235294118
     Recall: 0.9166666666666666
  F-Measure: 0.7586206896551724
-----------------------
------- CLASS 1 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 2 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 3 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
```

```
-----------------------
########## TEST ##########
------- CLASS 0 -------
  Precision: 0.6666666666666666
     Recall: 1.0
  F-Measure: 0.8
-----------------------
------- CLASS 1 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 2 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 3 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
## Evaluating classification performance using naive bayes ...
~~~~~~~ Naive Bayes ~~~~~~~
########## TRAIN ##########
------- CLASS 0 -------
  Precision: 0.45
     Recall: 0.375
  F-Measure: 0.4090909090909091
-----------------------
------- CLASS 1 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
------- CLASS 2 -------
  Precision: 0.043478260869565216
     Recall: 0.2
  F-Measure: 0.07142857142857142
-----------------------
------- CLASS 3 -------
  Precision: 0.0
     Recall: 0.0
  F-Measure: 0.0
-----------------------
########## TEST ##########
------- CLASS 0 -------
  Precision: 0.75
     Recall: 1.0
  F-Measure: 0.8571428571428571
-----------------------
------- CLASS 1 -------
  Precision: 0.25
     Recall: 0.5
```

```
  F-Measure: 0.3333333333333333
-----------------------
------- CLASS 2 -------
  Precision: 0.2
     Recall: 0.2
  F-Measure: 0.20000000000000004
-----------------------
------- CLASS 3 -------
  Precision: 0.14285714285714285
     Recall: 0.3333333333333333
  F-Measure: 0.2
-----------------------
```

Ergebnisse

Die Klassifizierer werden immer mit ihren Trainings- und einem komplementären Testset ausgeführt. Uns ist nicht genau klar, warum die Klassifizierer bei ihren eigenen Trainingsset immer sehr schlecht abschneiden und bei dem Testset besser. Es ergibt Sinn, dass Klasse 0 immer die größte Präzision aufweist, da diese Klasse (mit Abstand) am meisten im Trainingsset vertreten ist. Unsere Reihung der Klassifizierer sieht wie folgt aus:

1. Naive Bayes
2. Rocchio
3. kNN

Bei kNN ist die Anzahl der Nachbarn sehr entscheidend. Uns ist aufgefallen, dass der kNN-Klassifizierer (mit unseren Einstellungen) manche Klassen gar nicht erkennen kann. Da wir alle 4 Klassen klassifizieren und Klasse 1 dabei die meisten Dokumente beinhält, resultieren höhere Werte von `k` in einer höheren Wahrscheinlichkeit die Dokumente der Klasse 1 zuzuordnen.

Bei Rocchio haben wir die zwei Distanzmaße `euclidean` und `manhattan` ausprobiert und `euclidean` funktioniert am besten.

Anhang

- GitHub-Repository