Hochschule RheinMain
Prof. Dr. Adrian Ulges

30.04.2018

# Anwendungen der KI (SS 2018)

# Aufgabenblatt 3

---

**Aufgabe 3.1 (Logistische Regression in sklearn)**
Die Python-Bibliothek bietet zahlreiche Machine Learning - Modelle, unter anderem **logistische Regression**. Schauen Sie sich die Klasse `LogisticRegression` an: Mit `fit()` trainieren Sie das Modell, mit `predict()` erhalten Sie für neue Daten Vorhersagen. Dieses Tutorial bietet Ihnen kleine Beispiele.

**Aufgabe 3.2 (Dokumentklassifikation)**
Implementieren Sie einen Dokumentklassifikator, der `LogisticRegression` verwendet. Der Klassifikator soll automatisch Artikel der *New York Times* ihren Kategorien (wie *arts, technology, sports, ...*) zuordnen.

- Laden Sie `nytimes.zip` herunter und entpacken Sie die Datei. Im Ordner `data` finden Sie zwei Unterordner `train` und `test`. In diesen liegen 1435 bzw. 1437 Artikel der *New York Times*, sortiert in 8 Kategorien.
- In `classifier.py` finden Sie ein Programmgerüst, das Eingabedokumente per Kommandozeile einliest und vorverarbeitet (incl. tokenization, stemming, sowie Filtern von Stopwords und seltener Worte). Sie erhalten Bag-of-Words-Features für Ihren Klassifikator.
- Lesen Sie sich den Code in `classifier.py` durch *(die Textvorverarbeitung findet in preprocess_documents.py statt).*
- Starten Sie: `python classifier.py --train data/train/*/*`
  (die Textvorverarbeitung läuft durch, aber Sie erhalten einen `NotImplementedError`).
  *Anmerkung: Evtl. müssen Sie noch das Paket 'punkt' nachinstallieren: In der Python-Shell 'import nltk; nltk.download()' aufrufen, 'download(d)' auswählen und 'punkt' eingeben.*
- Unser Ziel ist es nun, die Klasse `DocumentClassifier` zu implementieren:
  - Verwenden Sie boolesche Bag-of-Words Features: Kommt ein Term nicht in einem Dokument vor, ist das entsprechende Merkmal 0, ansonsten 1 (d.h. die *Häufigkeit* des Vorkommens ist irrelevant).
  - Generieren Sie aus den Eingabedaten eine boolesche Term-Dokument-Matrix $X$.
  - Trainieren Sie in `train()` einen `LogisticRegression`-Klassifikator. Wenden Sie diesen in `apply()` an.

– Schreiben Sie außerdem etwas Code, um den trainierten Klassifikator zu speichern und später wieder zu verwenden. Hier bietet sich das Python-Modul `pickle` an.
– Trainieren Sie auf der Trainingsmenge. Testen Sie dann auf der Testmenge: `python classifier.py --apply data/test/*/*`. Messen Sie mit `error_rate()` die Fehlerrate.