Übung 15: scikit-learn

1. Genau wie NLTK stellt auch `scikit-learn` einige Datensets bereit. In dieser Übung sollen Sie mit dem 20newsgroups-Datenset arbeiten. Es enthält 20,000 Newsgroup-Dokumente aus 20 verschiedenen Kategorien enthält. Jedes Dokument ist genau einer Kategorie zugeordnet, die im Datenset annotiert ist. Das Datenset ist in Trainings- und Testdaten unterteilt, wobei alle Kategorien in Train und Test ungefähr gleich proportioniert vorkommen.

   a) Laden Sie nun die Trainingspartition von 20newsgroups. Sehen Sie sich dazu die Dokumentation der Funktion `sklearn.datasets.fetch_20newsgroups` an. Laden Sie die Trainingsdaten in eine Variable `newsgroups_train`.

   b) Die Dokumente sind nun in `newsgroups_train.data` als Liste gespeichert. Finden Sie heraus, wie viele Dokumente im Trainingsset sind.

2. Jetzt sollen Sie sich ein Trainingsset bauen, das nur Dokumente aus zwei Kagorien enthält und die Dokumente in Feature-Vektoren transformieren.

   a) Die Kategorien der Dokumente sind als Integer-IDs in `newsgroups_train.target` gespeichert. Überzeugen Sie sich, dass es genau so viele Kategorien wie Dokumente sind.

   b) `newsgroups_train.target_names` speichert die Namen der Kategorien. Lassen Sie sich die Kategorien anzeigen.

   c) Wählen Sie sich zwei Kategorien aus, die Ihnen nicht zu ähnlich erscheinen. Verwenden Sie dann nochmal `fetch_20newsgroups`, um nur die Trainingsdaten zu laden, die zu diesen beiden Kategorien gehören. Speichern sie diese in einer Variable `newsgroups_train_2cat`, und finden Sie heraus, wie viele Dokumente geladen wurden.

   d) Erstellen Sie jetzt für Ihr Trainingsset eine Dokument-Term-Matrix aus den Trainingsdaten (1 Reihe pro Dokument). In der Vorlesung haben wir die Klasse `sklearn.feature_extraction.text.CountVectorizer` verwendet. Man kann allerdings auch gewichtete Counts extrahieren. Verwenden Sie nun die Klasse `TfidfVectorizer`.

3. Jetzt können Sie einen Klassifizierer trainieren und auf Testdaten evaluieren.

   a) Verwenden Sie das „Flowchart" aus den Slides, um nach einem passenden Algorithmus zu suchen.

   b) Verwenden Sie den Algorithmus, um einen Klassifizierer zu trainieren, indem Sie die Dokument-Term-Matrix als Beobachtungen und die Kategorien (`newsgroups_train_cat2.target`) als Labels behandeln.

   c) Verwenden Sie jetzt wieder `fetch_20newsgroups()`, um die Testdaten aus denselben Kategorien zu laden. Erstellen Sie wieder eine Dokument-Term-Matrix `X_test`. Verwenden Sie nun das trainierte Modell, um die Testdaten zu klassifizieren. *Hinweis*: Verwenden Sie zum Vektorisieren dasselbe Objekt, mit dem Sie auch die Trainingsdaten vektorisiert haben. Rufen sie *nicht* noch einmal die `fit()`-Funktion auf, da sich sonst das Vokabular ändern würde.

   d) Evaluieren Sie die Accuracy Ihres Klassifizierers, indem Sie die Gold-Labels (`newsgroups_test_2cats.target`) mit den vom Klassifizierer ausgegebenen vergleichen. Schauen Sie in `sklearn.metrics` nach der Evaluierungsmetrik.

4. Wenn noch Zeit ist, können Sie folgendes ausprobieren:

   - Trainieren Sie ein Modell für zwei Kategorien, die sich sehr ähnlich sind
   - Trainieren Sie ein Modell mit mehr als zwei Kategorien!
   - Probieren Sie einen anderen Klassifizierer aus.