Laboratorium 2. Klasyfikacja danych tekstowych

Mając dane oznaczone etykietami, można wykonać prosty eksperyment klasyfikacji. Jednakże, surowy tekst napisany w języku naturalnym nie nadaje się do trenowania modeli. Aby było to możliwe konieczna jest wektoryzacja danych. Rozwiązanie tego laboratorium pozwoli poznać przykładowy sposób transformacji surowych danych tekstowych z etykietami do prostego modelu zdolnego do klasyfikacji.

Zadanie 2.1. Przygotowanie danych

Wczytaj dane z pliku `imdb.csv` i dla treści każdego dokumentu:

1. Zamień w tekście wszystkie duże litery na małe
2. Dokonaj tokenizacji z użyciem `TreebankWordTokenizer`[1]
3. Usuń z tokenów `stopwords`[2] (możesz też usunąć inne tokeny, które nie mają znaczenia np. "
")
4. Wykonaj stemming na tokenach z poprzedniego punktu z użyciem `PorterStemmer`[3]
5. Wykonaj lematyzację na tokenach z punktu 3. z użyciem `WordNetLemmatizer`[4]
6. Zapisz do tablic cztery zestawy tokenów:
   - Tokeny oryginalne
   - Tokeny bez stop words
   - Tokeny po stemmingu
   - Tokeny po lematyzacji

Zadanie 2.2. Klasyfikacja

Wczytaj zestawy tokenów z pierwszego zadania i dla każdego zestawu:

1. Podziel dane na treningowe (70%) oraz testowe (30%) z użyciem `train_test_split`[5]
2. Dokonaj wektoryzacji za pomocą metody `CountVectorizer`[6]
3. Wytrenuj model z użyciem klasyfikatora `MultinomialNB`[7] na danych treningowych (funkcja `fit()`)
4. Dokonaj predykcji modelu na danych testowych i zapisz wynik (funkcja `predict()`)
5. Wyświetl dokładność wyrażoną za pomocą metryki `accuracy_score`[8]
6. Porównaj uzyskane wyniki i zapisz w komentarzach spostrzeżenia

Zadanie 2.3. Eksperyment

Wczytaj zestawy tokenów z pierwszego zadania i dla każdego zestawu:

1. Dokonaj wektoryzacji z użyciem `TfidfVectorizer`[9]
2. Dane podziel na 5 foldów z użyciem funkcji `StratifiedKFold`[10]
3. Utwórz odpowiednie tablice na wyniki
4. Dla każdego foldu:
   - Wytrenuj model z użyciem klasyfikatora `MLPClassifier`[11] na danych **treningowych** (funkcja `fit()`)
   - Dokonaj predykcji modelu na danych **testowych** (funkcja `predict()`)
   - Zapisz w tablicy z punktu 5 dokładność wyrażoną za pomocą metryki `accuracy_score`
5. Wyznacz średnią dokładność oraz odchylenie standardowe i wyświetl wyniki

[1] <https://www.nltk.org/api/nltk.tokenize.treebank.html>
[2] <https://www.nltk.org/book/ch02.html>
[3] <https://www.nltk.org/api/nltk.stem.porter.html>
[4] <https://www.nltk.org/api/nltk.stem.wordnet.html>
[5] <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html>
[6] <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html>
[7] <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html>
[8] <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html>
[9] <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html>
[10] <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html>
[11] <https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html>