SPRAWOZDANIE

Zajęcia: Nauka o danych II

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 8 Data 14.06.2025 Temat: „Implementacja zaawansowanych metod analizy tekstu w Pythonie” Wariant 1	Szymon Nycz Informatyka II stopień, niestacjonarne, 2 semestr, gr.1a TTO

1. Polecenie:

   Link do repozytorium:
   https://github.com/Maciek332/Semestr_3_Nycz/tree/main/NoD%20II

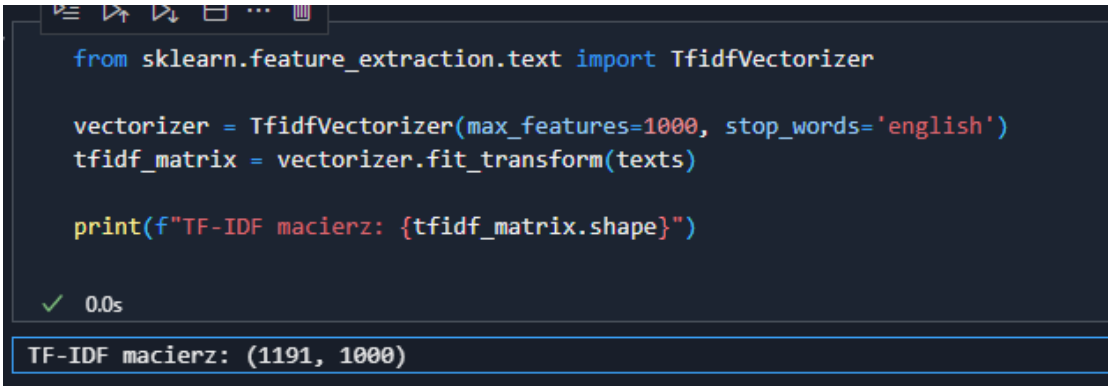
   Zadanie dotyczyło zastosowania wybranych metod analizy tekstu w Pythonie w celu wydobycia struktury i ukrytych zależności w danych tekstowych. W pierwszej kolejności posłużono się reprezentacją TF-IDF, która pozwala na uwzględnienie zarówno częstości występowania słów w dokumentach, jak i ich unikalności w całym korpusie, co umożliwia lepsze wyróżnienie słów kluczowych. Następnie wykorzystano modele osadzania słów w przestrzeni wektorowej (Word2Vec), które odwzorowują podobieństwa semantyczne między słowami, pozwalając m.in. na obliczanie ich podobieństw w sensie znaczeniowym.
   Kolejnym etapem była analiza tematyczna z użyciem modelu LDA, umożliwiająca automatyczne wykrycie dominujących tematów w zbiorze dokumentów. Pozwala to grupować teksty według ukrytych w nich wątków bez konieczności ich ręcznego czytania. Na zakończenie zastosowano redukcję wymiarowości metodą SVD, aby przedstawić dane tekstowe w uproszczonej przestrzeni i ułatwić ich wizualizację. Wszystkie te metody razem tworzą spójne podejście do eksploracji i analizy dużych zbiorów danych tekstowych.

2. Opis programu opracowanego

```
from sklearn.datasets import fetch_20newsgroups

data = fetch_20newsgroups(subset='train',
                          categories=['rec.autos', 'rec.sport.baseball'],
                          remove=('headers', 'footers', 'quotes'))

texts = data.data
print(f"Liczba dokumentów: {len(texts)}")
```

✓ 1m 7.8s

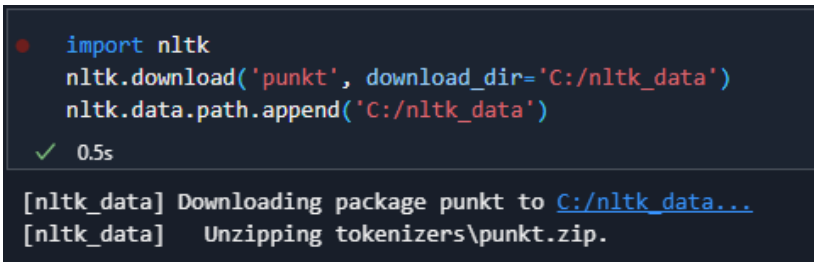
```
Liczba dokumentów: 1191
```



```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=1000, stop_words='english')
tfidf_matrix = vectorizer.fit_transform(texts)

print(f"TF-IDF macierz: {tfidf_matrix.shape}")
```

✓ 0.0s

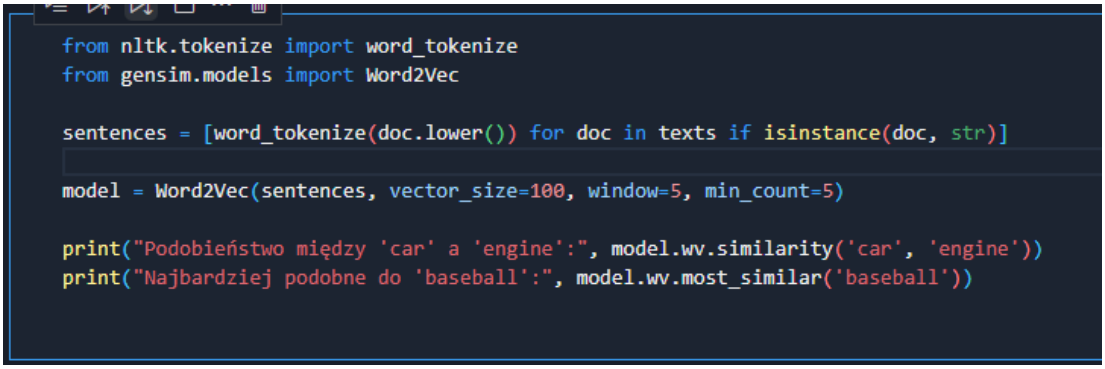
```
TF-IDF macierz: (1191, 1000)
```



```
import nltk
nltk.download('punkt', download_dir='C:/nltk_data')
nltk.data.path.append('C:/nltk_data')
```

✓ 0.5s

```
[nltk_data] Downloading package punkt to C:/nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
```

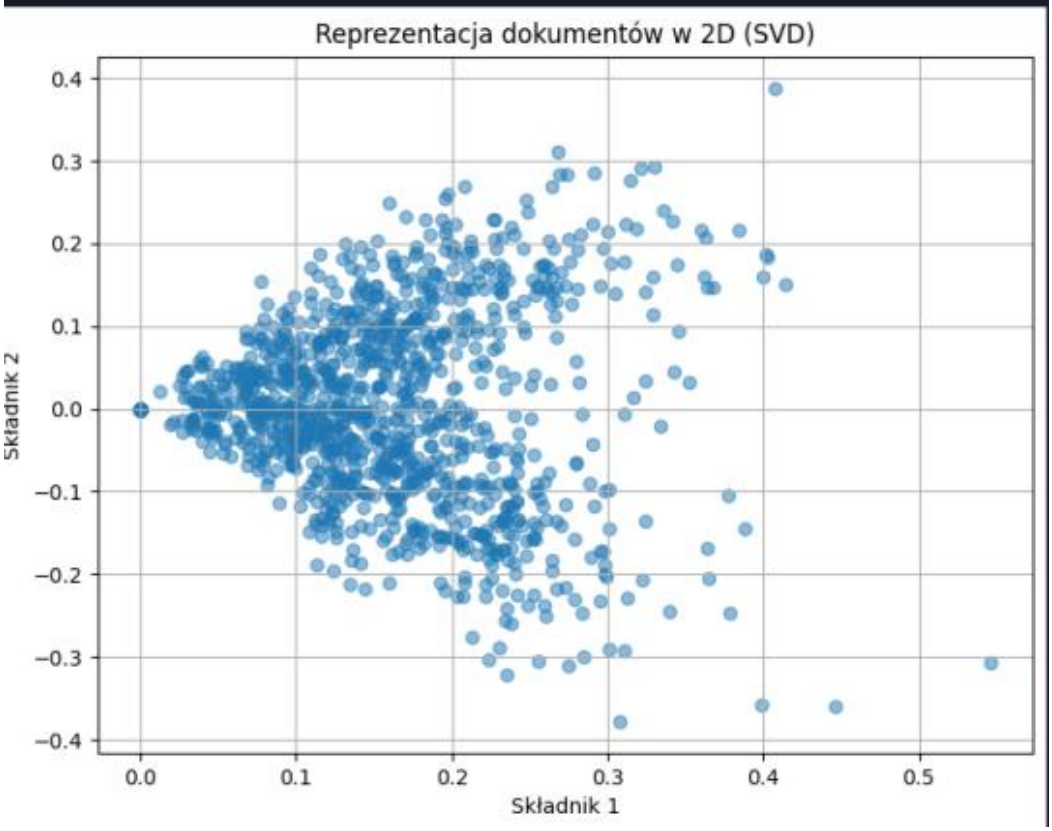


```
from nltk.tokenize import word_tokenize
from gensim.models import Word2Vec

sentences = [word_tokenize(doc.lower()) for doc in texts if isinstance(doc, str)]

model = Word2Vec(sentences, vector_size=100, window=5, min_count=5)

print("Podobieństwo między 'car' a 'engine':", model.wv.similarity('car', 'engine'))
print("Najbardziej podobne do 'baseball':", model.wv.most_similar('baseball'))
```

```python
from gensim.corpora.dictionary import Dictionary
from gensim.models import LdaModel

sentences = [s for s in sentences if len(s) > 5]

dictionary = Dictionary(sentences)
corpus = [dictionary.doc2bow(s) for s in sentences]

lda = LdaModel(corpus, num_topics=5, id2word=dictionary, passes=10)

for idx, topic in lda.print_topics():
    print(f"Temat {idx}: {topic}")
```

```python
from sklearn.decomposition import TruncatedSVD
import matplotlib.pyplot as plt

svd = TruncatedSVD(n_components=2)
X_2d = svd.fit_transform(tfidf_matrix)

plt.figure(figsize=(8, 6))
plt.scatter(X_2d[:, 0], X_2d[:, 1], alpha=0.5)
plt.title("Reprezentacja dokumentów w 2D (SVD)")
plt.xlabel("Składnik 1")
plt.ylabel("Składnik 2")
plt.grid(True)
plt.show()
```

0.6s



3. Wnioski

Realizacja tego zadania pokazała, że nowoczesne metody analizy tekstu stanowią niezwykle efektywne narzędzia do wydobywania wiedzy z dużych zbiorów dokumentów. Wykorzystanie TF-IDF umożliwiło skonstruowanie reprezentacji wektorowych dokumentów, w których kluczowe słowa miały większy wpływ na opis zawartości tekstu. Z kolei zastosowanie modelu Word2Vec pozwoliło uwzględnić semantyczne podobieństwa między słowami, co okazało się bardzo przydatne w zadaniach polegających na porównywaniu znaczenia lub wyszukiwaniu słów kontekstowo podobnych.

Analiza tematyczna przeprowadzona przy pomocy LDA ukazała, jak możliwe jest automatyczne grupowanie dokumentów według dominujących w nich tematów, co znacząco ułatwia eksplorację dużych zbiorów tekstów bez konieczności ich ręcznego czytania. Ostatni etap, czyli redukcja wymiarowości z wykorzystaniem SVD, pokazał, że dane tekstowe można skutecznie odwzorować w przestrzeni dwuwymiarowej, zachowując przy tym istotne informacje potrzebne do dalszej analizy czy wizualizacji.

Podsumowując, przeprowadzone eksperymenty potwierdziły skuteczność i uniwersalność omawianych metod w analizie danych tekstowych. Pozwoliły nie tylko na lepsze zrozumienie teoretycznych podstaw takich narzędzi jak TF-IDF, Word2Vec czy LDA, ale także na zdobycie praktycznych umiejętności ich implementacji oraz interpretacji wyników, co jest niezbędne we współczesnych zastosowaniach nauki o danych i przetwarzania języka naturalnego.