МИНОБРНАУКИ РФ

ФГБОУ Тверской государственный технический университет

Факультет информационных технологий

Кафедра «Программное обеспечение»

Лабораторная работа №6

Дисциплина: «Анализ больших данных»

Вариант №7

Работу выполнил: студент группы

ПИН. РИС.21.06

Олимов Авазбек

Тверь, 2025

# Вариант задания

* Реализовать только общую часть задания на 10 песнях
* В поиске похожих элементов Word2Vec могут быть ошибки
* Модель можно Сложность: Rare не выгружать

# База данных

Собрано 13 текстов песен артиста Kanye West со сайта Genius.

Текста сохранены в файл kanye\_lyrics.txt

def get\_lyrics(url):

    try:

        response = requests.get(url)

        soup = BeautifulSoup(response.text, 'html.parser')

        title = soup.find('h1').text.strip()

        lyrics\_div = soup.find('div', class\_=lambda x: x and 'Lyrics\_\_Container' in x)

        lyrics = '\n'.join([p.text for p in lyrics\_div.find\_all('p')]) if lyrics\_div else ""

        return f"{title}\n{lyrics}"

    except Exception as e:

        print(f"Error fetching {url}: {str(e)}")

        return None

Признаки:

10 наиболее частых слов:

god 0.369395

hallelujah 0.315525

lord 0.215480

every 0.207785

oh 0.200089

sing 0.192393

uh 0.192393

need 0.161610

know 0.161610

im 0.153914

Проведены лемматизация и стемминг использованием библиотеки pymorphy3

- MorphAnalyzer

Лемматизированные слова:

['every', 'hour', '71', 'contributors', 'translations', 'türkçe', 'español', 'português', 'polski', 'italiano', 'deutsch', 'français', 'every', 'hour', 'lyrics', 'sing', 'every', 'hour', 'every', 'hour', 'til', 'the', 'power', 'every', 'minute', 'every', 'minute', 'of', 'the', 'lord', 'every', 'second', 'every', 'second', 'comes', 'sing', 'each', 'and', 'every', 'millisecond', 'down', 'we', 'need', 'you', 'we', 'need', 'you', 'sing', 'til', 'the']

Стеммированные слова:

['every', 'hour', '71', 'contributors', 'translations', 'türkçe', 'español', 'português', 'polski', 'italiano', 'deutsch', 'français', 'every', 'hour', 'lyrics', 'sing', 'every', 'hour', 'every', 'hour', 'til', 'the', 'power', 'every', 'minute', 'every', 'minute', 'of', 'the', 'lord', 'every', 'second', 'every', 'second', 'comes', 'sing', 'each', 'and', 'every', 'millisecond', 'down', 'we', 'need', 'you', 'we', 'need', 'you', 'sing', 'til', 'the']

Приведенные все слова к нижнему регистру.

# Загрузка текста

lyrics = load\_lyrics(file\_path)

lowercase\_lyrics = lyrics.lower()

Удалены все знаки препинания:

cleaned\_lyrics = remove\_stopwords(lyrics)

График слов

A close up of words

AI-generated content may be incorrect.

График t-SNE

A graph with numbers and letters

AI-generated content may be incorrect.

Модели

# Обучение модели Word2Vec

model = Word2Vec(sentences=tokenized\_sentences, vector\_size=100, window=5, min\_count=1, workers=4)

# Сохранение модели

model.save(r"C:\Users\avazb\Desktop\BigData\Lab6\kanye\_word2vec.model")

Найболее частые слова и применение tSNE:

# Получение 15 наиболее частых слов из модели

word\_freq = Counter(model.wv.index\_to\_key)

most\_common\_words = word\_freq.most\_common(15)

words = [word for word, \_ in most\_common\_words]

# Получение векторов для наиболее частых слов

word\_vectors = np.array([model.wv[word] for word in words])

# Применение t-SNE

tsne = TSNE(n\_components=2, random\_state=42, perplexity=min(5, len(words) - 1))

word\_vectors\_2d = tsne.fit\_transform(word\_vectors)

Вывод: В ходе работы успешно собраны и обработаны текста песен с использованием Python-библиотек (BeautifulSoup, pymorphy3, gensim, scikit-learn). Применение методов NLP (стемминг, лемматизация, word2vec и t-SNE) позволило эффективно извлечь лингвистические особенности и визуализировать семантические взаимосвязи в текстах.

Результаты: Получены:

1. очищенные текстовые данные (нормализация + удаление шумов)
2. векторные представления слов (word2vec-модель)
3. 2D-проекция семантических связей (t-SNE)
4. частотный анализ ключевых терминов

Модели и результаты сохранены в воспроизводимом формате для дальнейшего анализа.