**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи №8 з дисципліни

«Аналіз даних в інформаційних системах»

„**Аналіз текстів**”

**Виконав(ла)**

(шифр, прізвище, ім'я, по батькові)

*ІП-11 Тарасьонок Дмитро Євгенович*

**Перевірив**

(прізвище, ім'я, по батькові)

*Олійник Ю. А.*

Київ 2023

**ЗМІСТ**

[1 Мета лабораторної роботи 3](#_Toc135778108)

[2 Завдання 4](#_Toc135778109)

[2.1 Основне завдання 4](#_Toc135778110)

[2.2 Додаткове завдання 4](#_Toc135778111)

[3 Виконання основного завдання 5](#_Toc135778112)

[3.1 Побудувати та проаналізувати часовий ряд для статистики захворювань на Covid в двох сусідніх країнах по вашому вибору (дані взяти в інтернеті). 5](#_Toc135778113)

[3.2 Побудувати та проаналізувати часовий ряд для курсу гривня/долар або гривня/євро за останні 3 роки (дані взяти в інтернеті). 16](#_Toc135778114)

[4 Виконання додаткового завдання 29](#_Toc135778115)

[5 Висновок 41](#_Toc135778116)

# Мета лабораторної роботи

Мета роботи – ознайомитись з методами аналізу текстів.

# Завдання

## Основне завдання

Дані для виконання: текстові дані у форматі csv-файлів або дані з відкритих джерел (телеграм-канали, RSS-канали тощо). Приклад даних за посиланням

1. Нормалізація та попередня обробка даних.
2. провести очищення текстових даних від стоп-слів/тегів/розмітки;
3. виконати токенізацію текстових елементів;
4. провести лематизацію текстових елементів (можна використати бібліотеку Spacy - приклад роботи за посиланням). Зберегти результат в окремий файл.
5. Створити Bag of Words для всіх нормалізованих слів. Зберегти результат в окремий файл.
6. Порахувати метрику TF-IDF для 10 слів, що найчастіше зустрічаються в корпусі;

## Додаткове завдання

### Інтелектуальний аналіз текстів (+1 бал):

* провести сантимент аналіз (визначення емоційної тональності – позитивний / негативний) для даних ukr\_text.csv.
* провести категоризацію (визначення категорій тексту) даних методом LSA.

### Обробка даних оповідань А.К. Дойля та Е.По (+1 бал):

* Завантажити потрібні дані.
* Завантажити оповідання А.К. Дойля та Е.По з папки Texts/Task.
* Виконати попередню обробку текстів.
* Побудувати дві хмари слів, що використовують А.К. Дойль та Е.По.
* Який з письменників написав більш похмурі оповідання?

# Виконання основного завдання

Для виконання поставленого завдання було завантажено дані з Telegram-каналу «Адвокат Права». Для аналізу будуть використовуватися перша тисяча повідомлень. У першу чергу імпортуємо всі необхідні пакети.



Рисунок 3.1 – Імпортування необхідних пакетів

Після цього визначимо функції, за допомогою яких будемо обробляти повідомлення. Це будуть: видалення пунктуації, емодзі, стоп-слів, токенізація речень на слова та їх лематизація в залежності від мови.

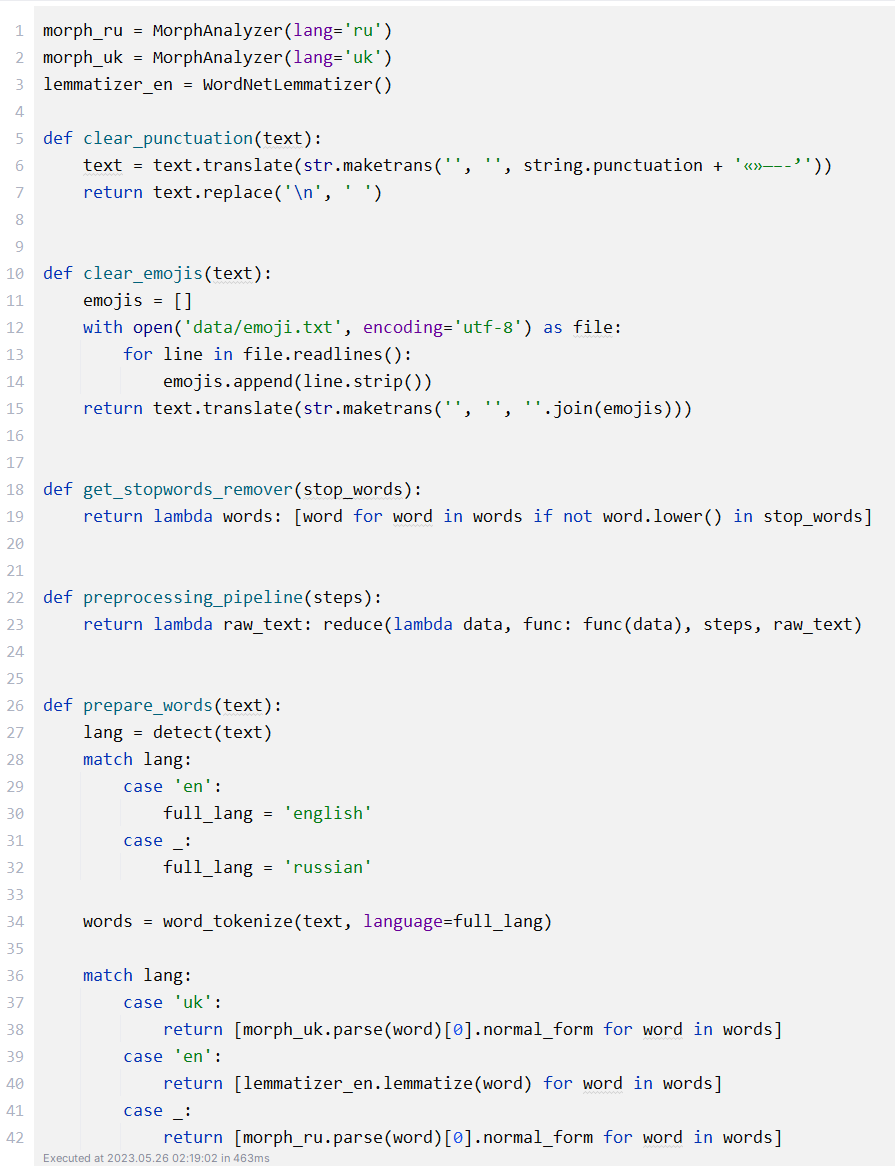


Рисунок 3.2 – Визначення функцій для попередньої обробки текстів

Наступним кроком прочитаємо файл із повідомленнями, запишемо їх у DataFrame.



Рисунок 3.3 – Завантаження повідомлень

Далі завантажимо список стоп-слів для української мови та створимо функцію, яка буде видаляти стоп-слова англійської, російської та української мов.



Рисунок 3.4 – Визначення стоп-слів

Об’єднаємо всі дії в один пайплайн для простоти роботи та застосуємо ці функції до кожного повідомлення.



Рисунок 3.5 – Попередня обробка текстів

Після застосування функцій обробки текстів отримали список слів для кожного повідомлення.

Далі об’єднаємо ці слова в один масив.

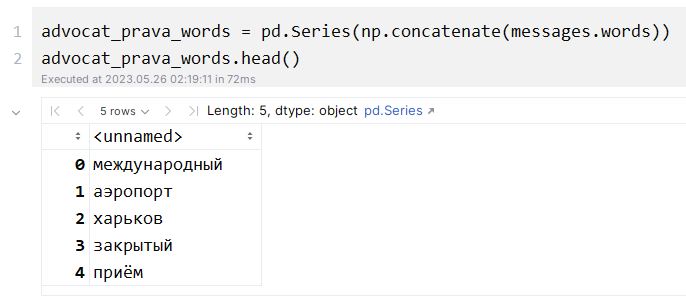


Рисунок 3.6 – 5 перших слів, що зустрічаються в корпусі

Запишемо ці слова у файл.



Рисунок 3.7 – Збереження слів у файл

Візуалізуємо частоту кожного зі слів.

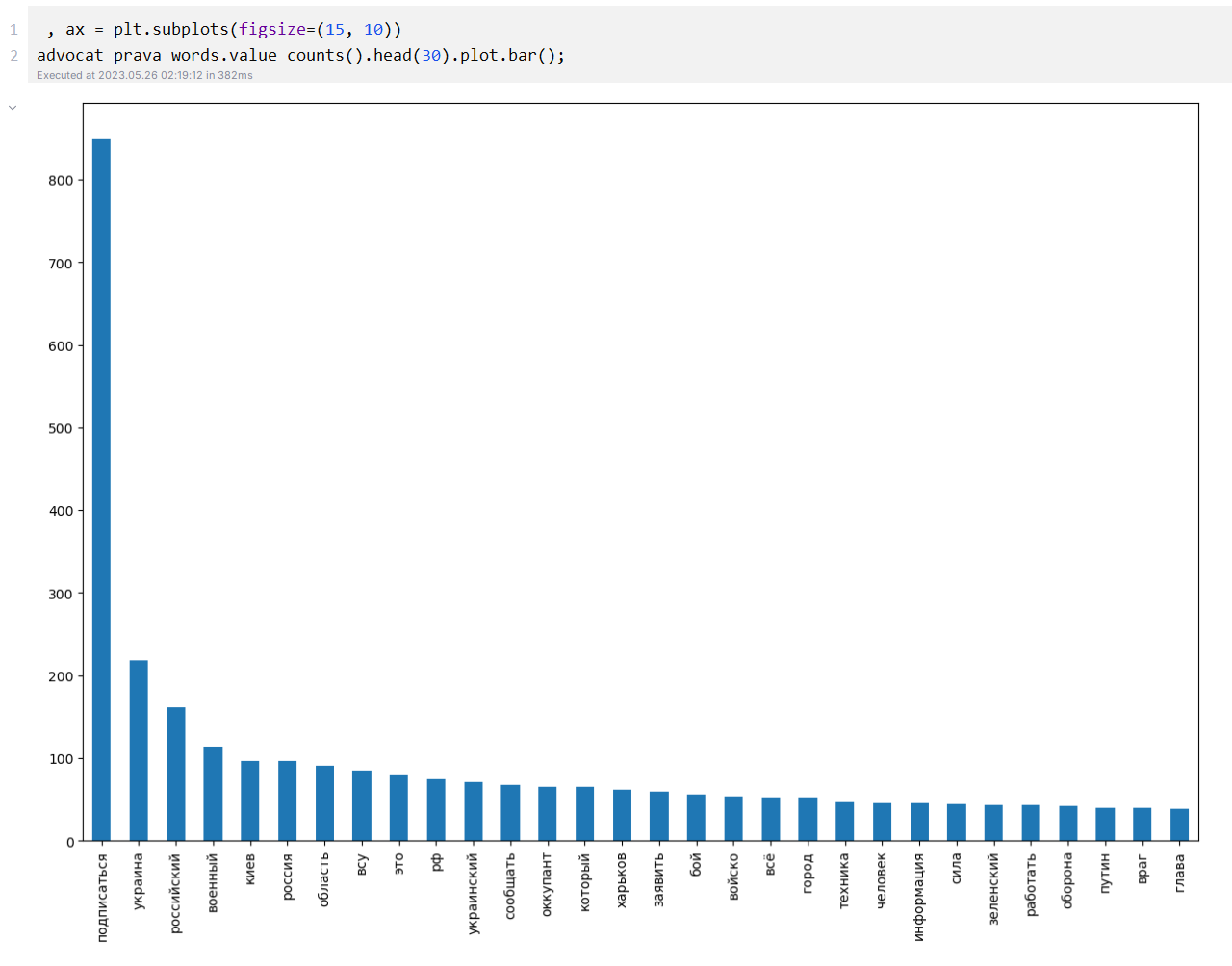


Рисунок 3.8 – Графік частоти кожного зі слів

Бачимо, що деякі стоп-слова не видалилися: «подписаться», «всё», «это». Видалимо і їх.

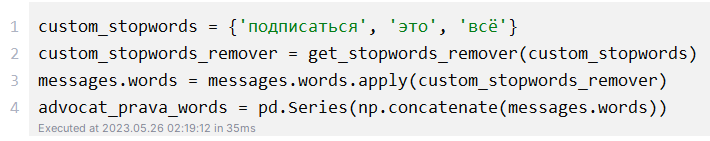


Рисунок 3.9 – Видалення додаткових стоп-слів

Наступним кроком зобразимо хмару слів.

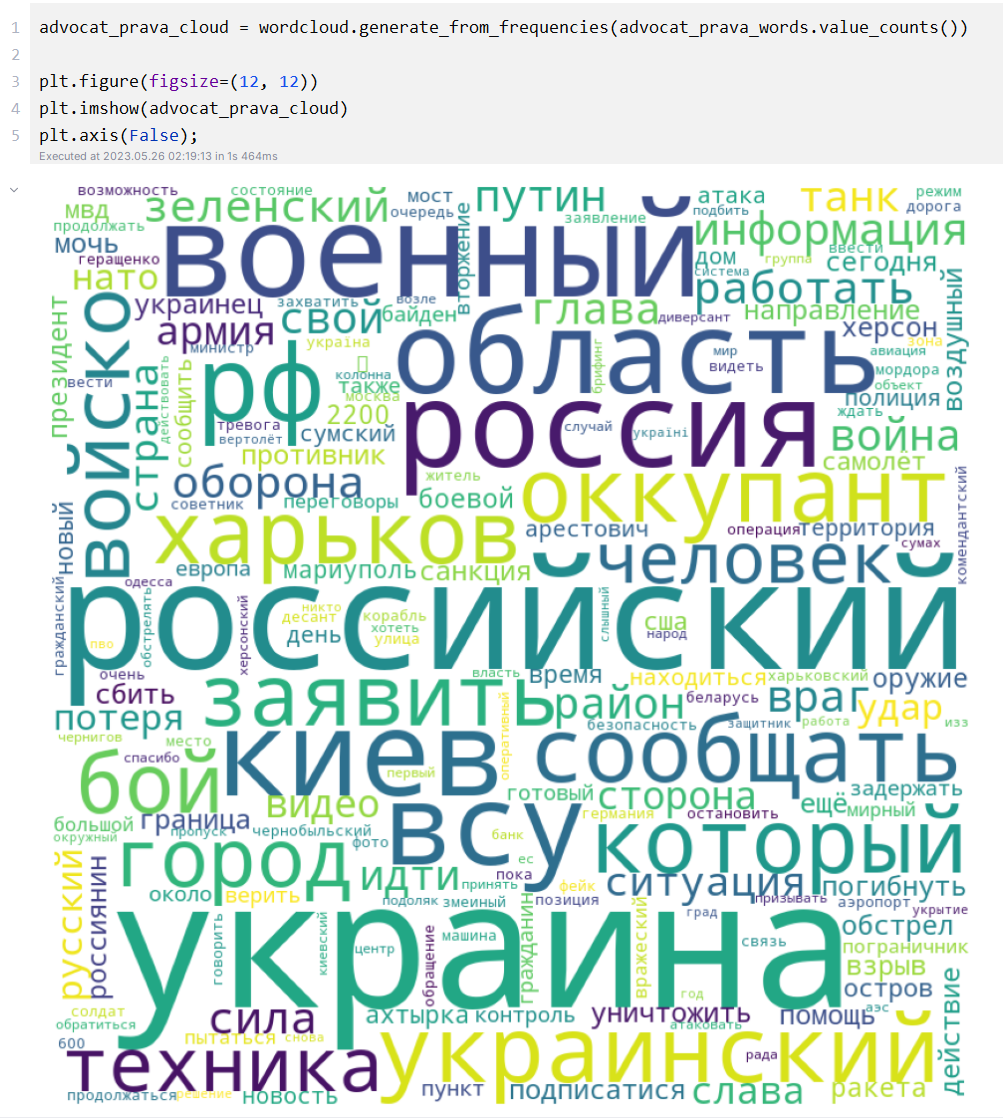


Рисунок 3.10 – Хмара слів для Telegram-каналу «Адвокат Права»

Тепер об’єднаємо очищені слова в речення просто додаючи між словами пробіли.

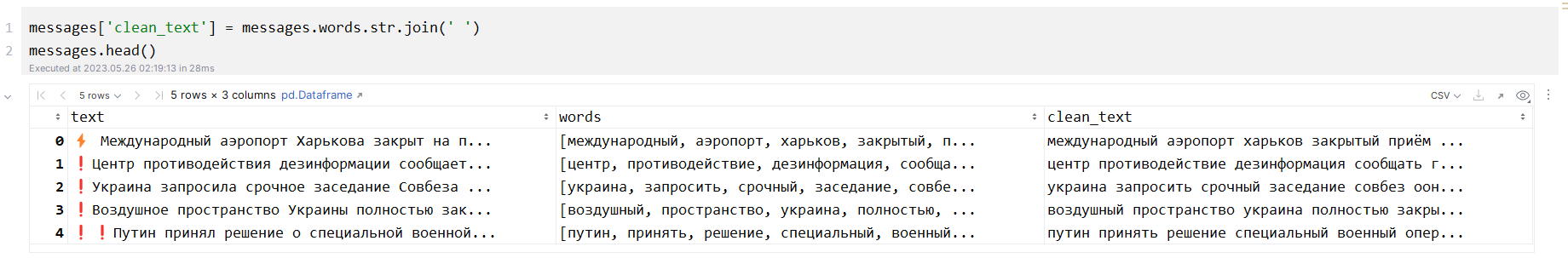


Рисунок 3.11 – Об’єднання слів у речення

Створимо Bag of Words для всіх нормалізованих речень, ознайомимося з тим, як виглядає результат роботи.

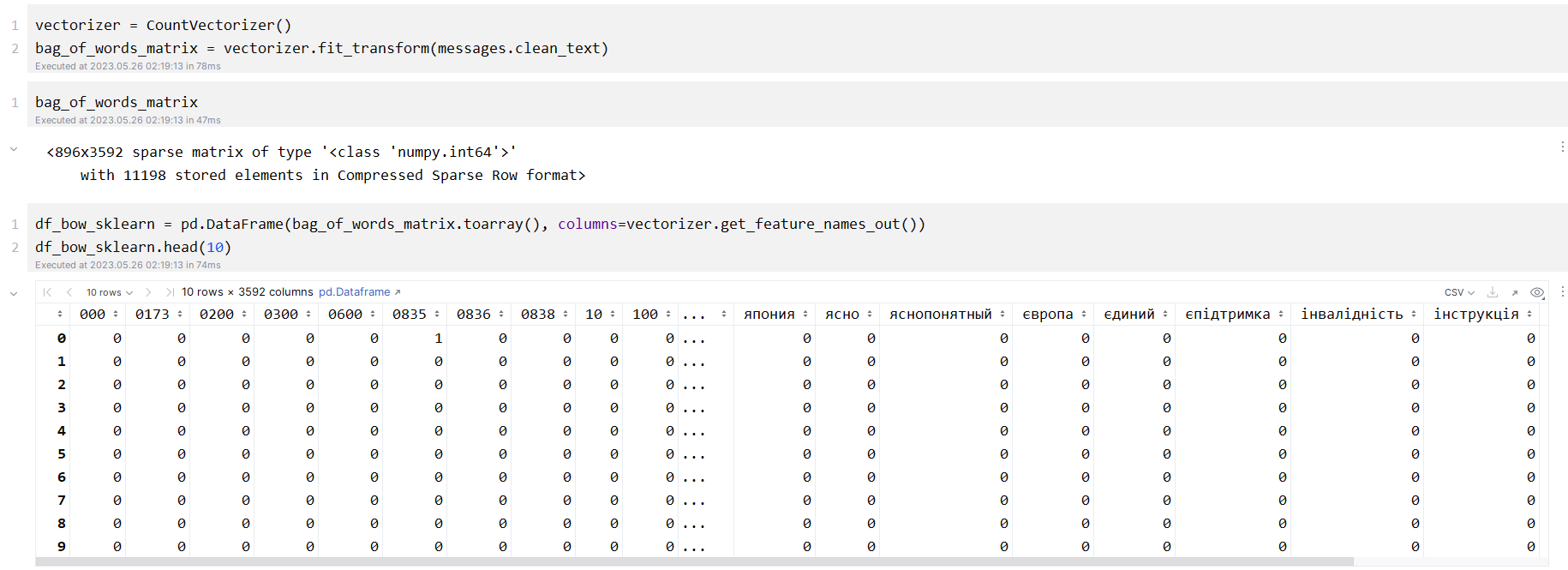


Рисунок 3.12 – Створення та навчання Bag Of Words

Збережемо отримані результати у файл.

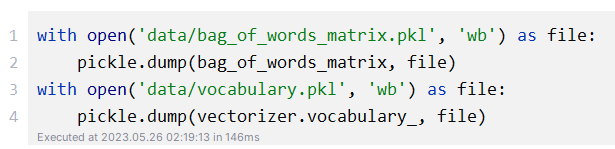


Рисунок 3.13 – Збереження Bag of Words у файл

Далі створимо TF-IDF векторизатор та навчимо його на наших нормалізованих реченнях.

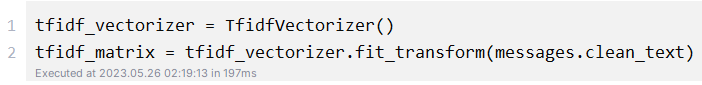


Рисунок 3.14 – Створення векторизатора для обчислення TF-IDF метрики

Отримаємо всі слова з векторизатора та порахуємо для них TF-IDF метрики для кожного речення.



Рисунок 3.15 – Обчислення TF-IDF метрики для 10 найбільш вживаних слів

# Виконання першого додаткового завдання

Для початку буде повторено всі ті ж дії для підготовки даних, що й при виконанні основного завдання.



Рисунок 4.1 – Імпортування необхідних пакетів



Рисунок 4.2 – Визначення функцій для попередньої обробки текстів



Рисунок 4.3 – Завантаження новин



Рисунок 4.4 – Завантаження стоп-слів для української мови

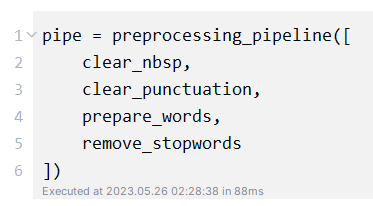


Рисунок 4.5 – Створення пайплайну для попередньої обробки

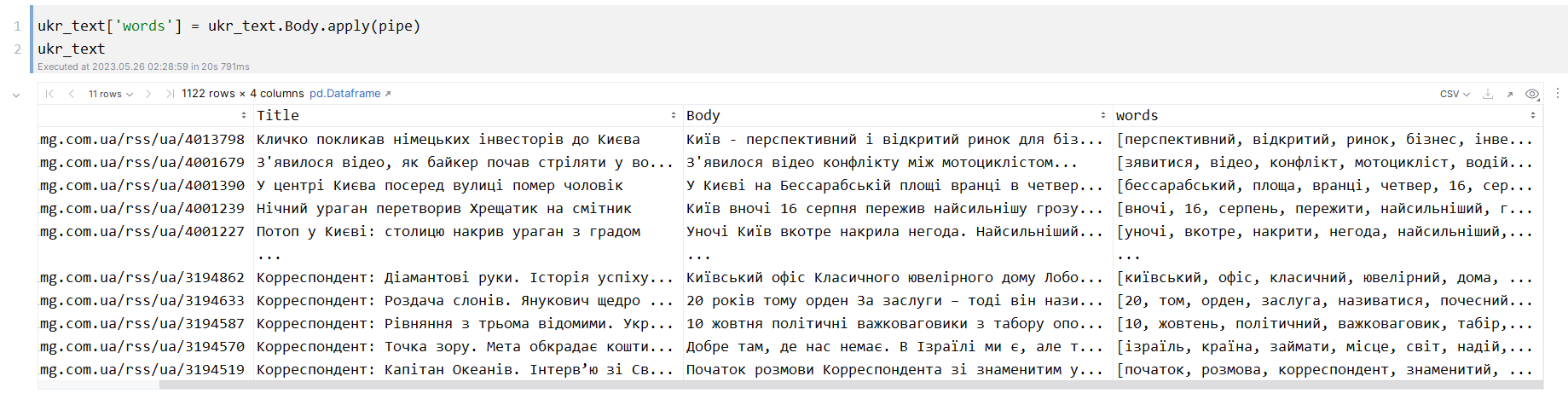


Рисунок 4.6 – Результат виконання попередньої обробки

Після того, як ми отримали масиви слів для кожної новини, можемо завантажити словник тональності та створити функцію, яка буде цю тональність обраховувати. Функція працює наступним чином: додає всі значення тональності, після чого ділить на загальну кількість слів у новині для того, щоб врівноважити цю оцінку для новин із різною кількістю слів.



Рисунок 4.7 – Завантаження словника тональності

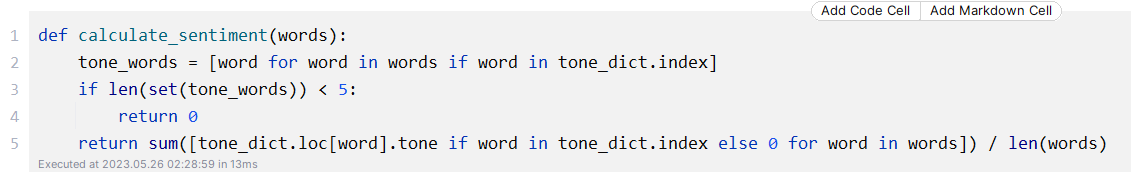


Рисунок 4.8 – Функція обрахунку тональності

Обрахуємо тональності.

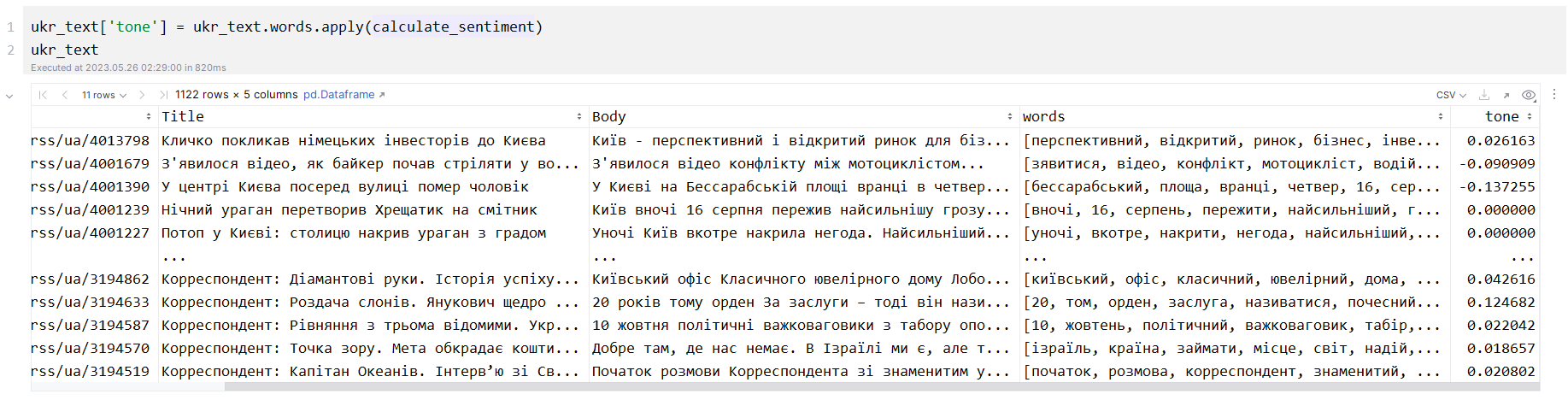


Рисунок 4.9 – Результат обрахунку тональності

Виведемо найбільш негативну та позитивну новини.

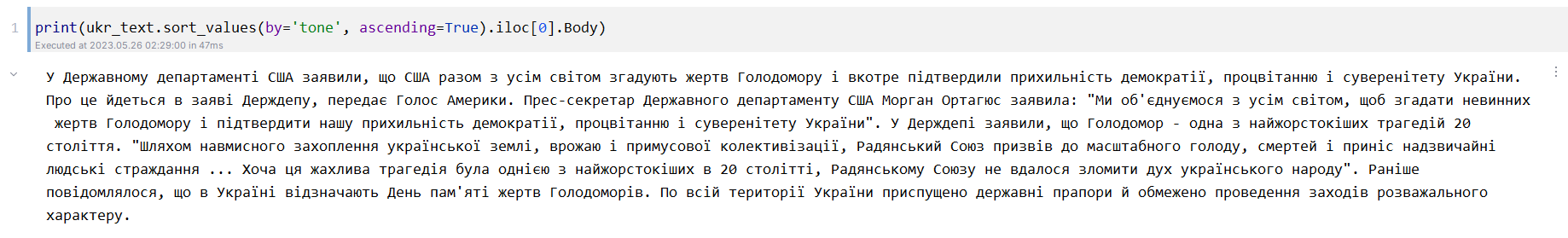


Рисунок 4.10 – Найбільш негативна новина

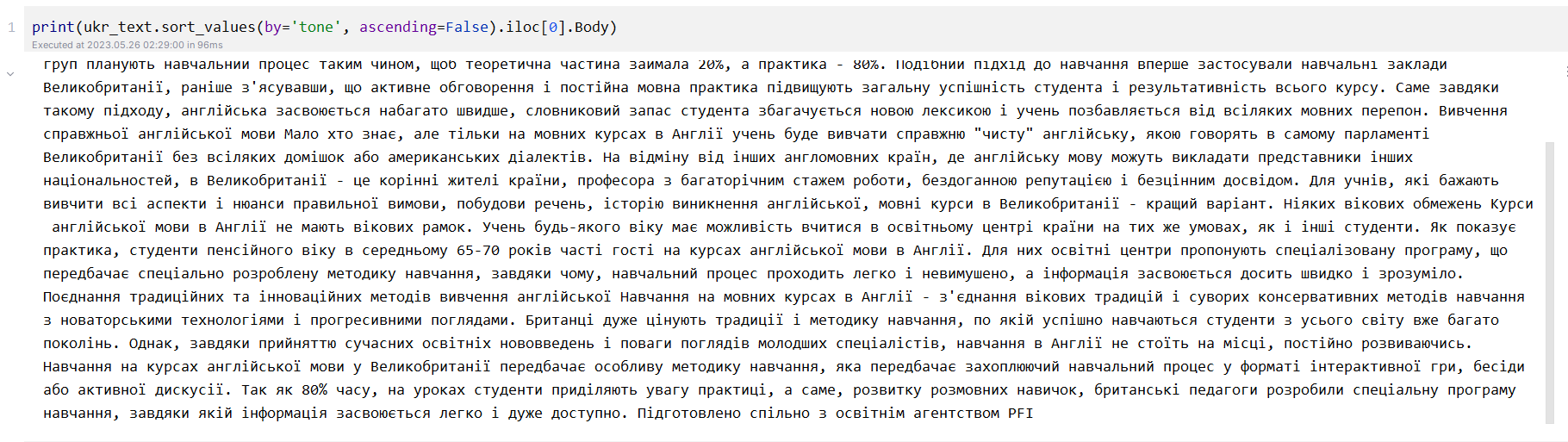


Рисунок 4.11 – Найбільш позитивна новина

Далі знову об’єднаємо слова кожної новини в новий текст, додавши між словами пробіли.

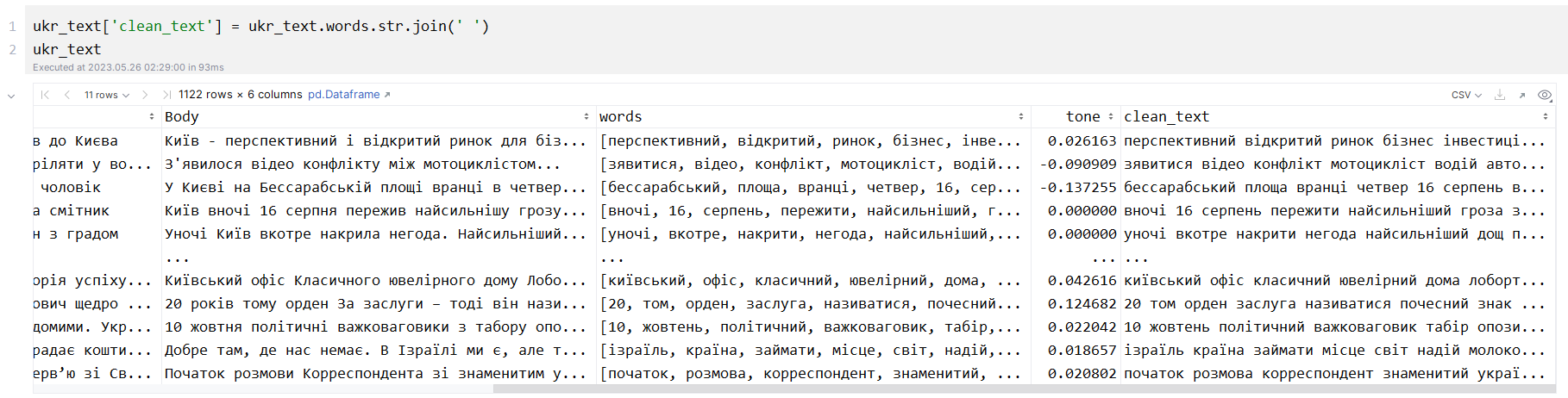


Рисунок 4.12 – Об’єднання слів у речення

Знову створимо векторизатор для обрахунку TF-IDF метрики.

Тепер створимо категоризатор. Встановимо кількість тем рівною 100, оскільки таке значення рекомендується для методу LSA в документації пакету scikit-learn. Навчимо цей перетворювач, виведемо матрицю для кожної новини.

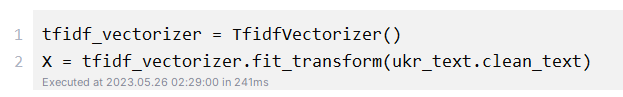


Рисунок 4.13 – Створення та навчання векторизатора для TF-IDF метрики



Рисунок 4.14 – Створення та навчання категоризатора

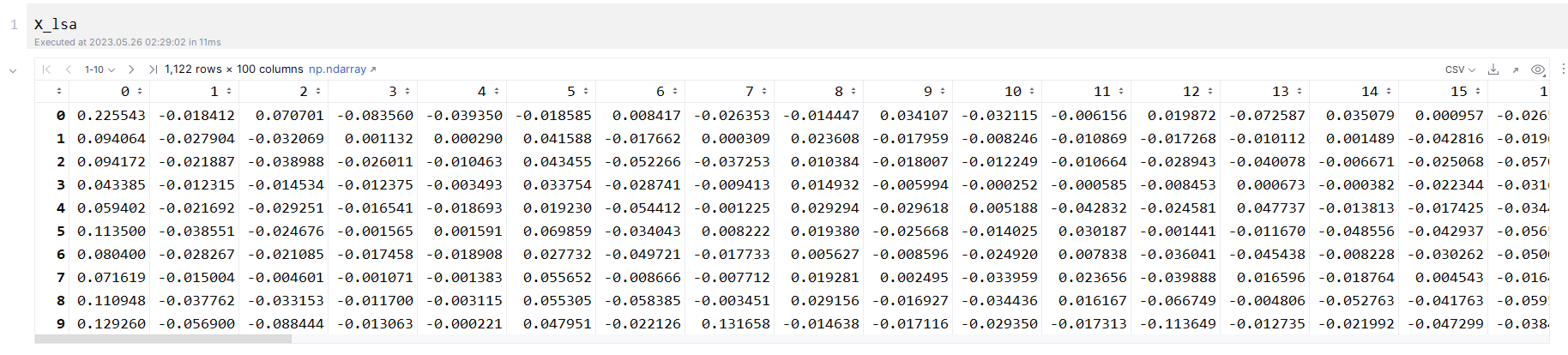


Рисунок 4.15 – Результат роботи категоризатора

Створимо функцію, яка буде показувати найважливіші слова для кожної категорії та виведемо їх для кожної категорії.

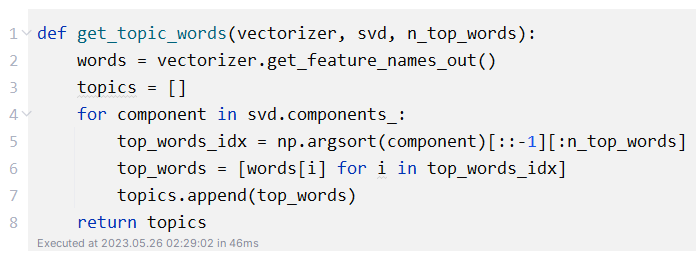


Рисунок 4.16 – Функція для виведення найважливіших слів для категорій

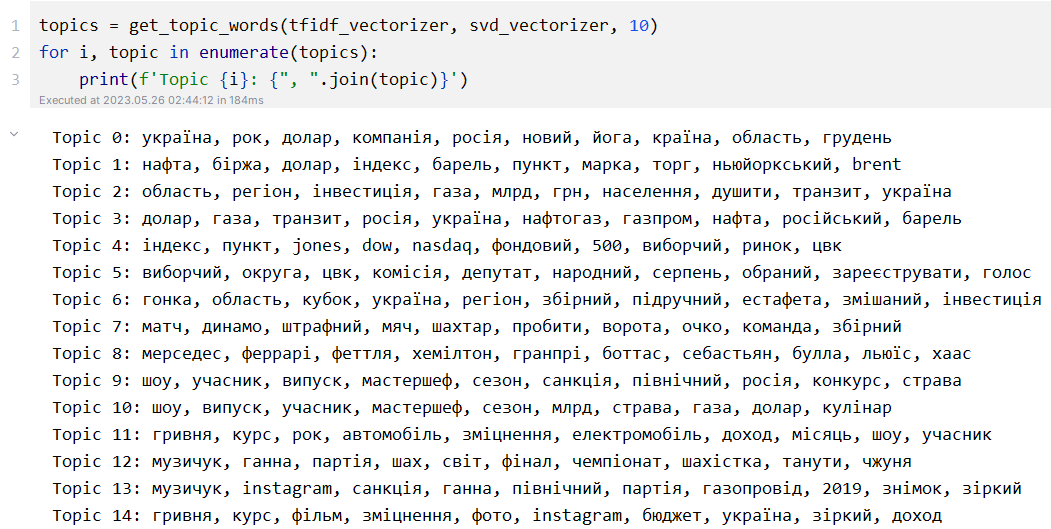


Рисунок 4.17 – Виведення найважливіших слів для категорій

Тепер додамо стовпець topic у наш набір даних, який буде вказувати, власне, на категорію, до якої належить кожна з новин.

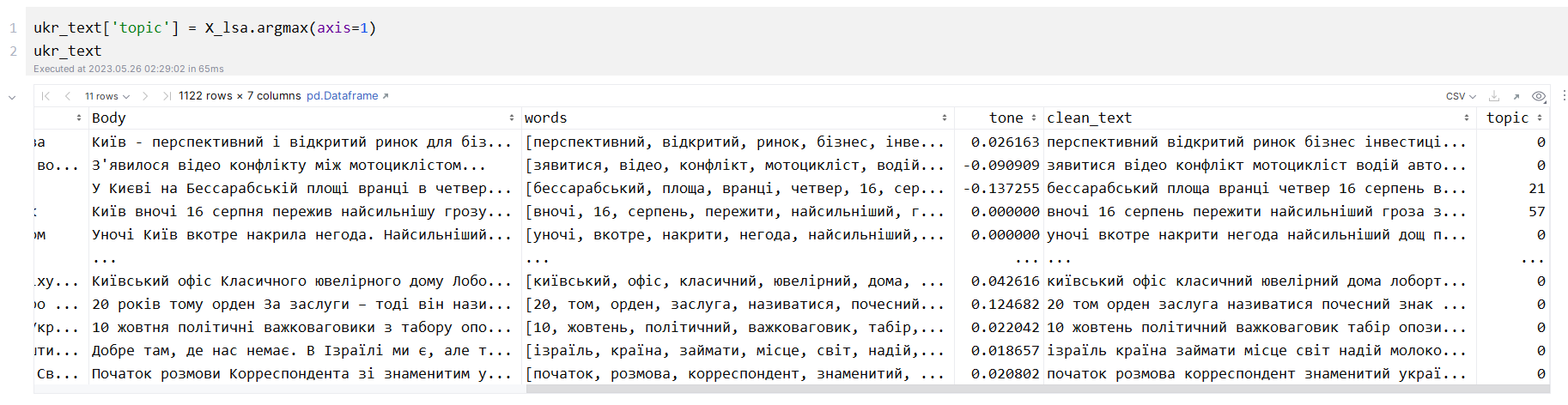


Рисунок 4.18 – Додавання стовпця з категорією до набору даних

Виведемо новини з категорією №6, яка дуже схожа на новини про футбол.

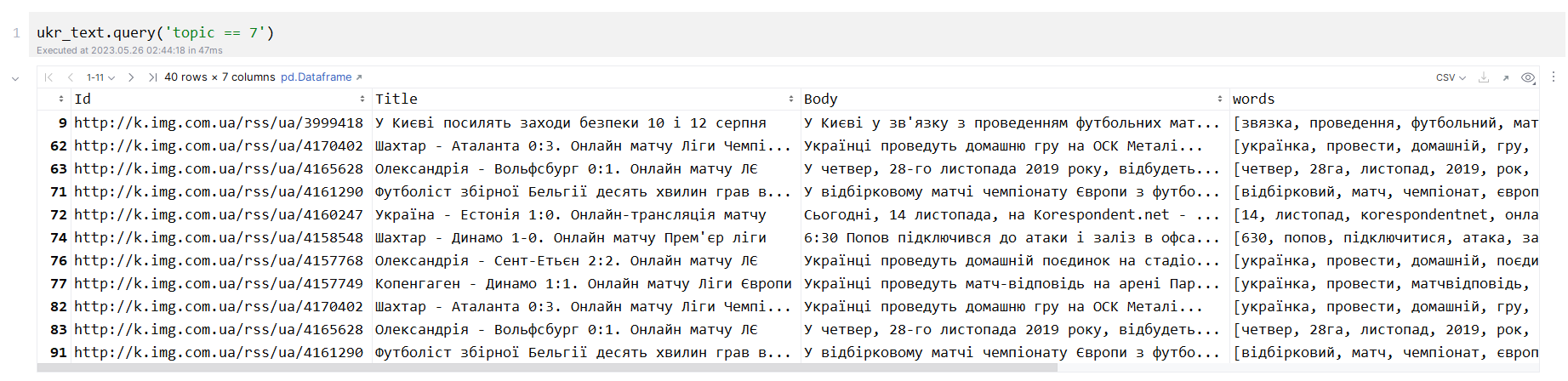


Рисунок 4.19 – Новини з категорією про футбол

Як бачимо, із цією категорією категоризатор впорався чудово. Можемо також вивести новини категорії №5, у яких, схоже, йдеться про вибори.

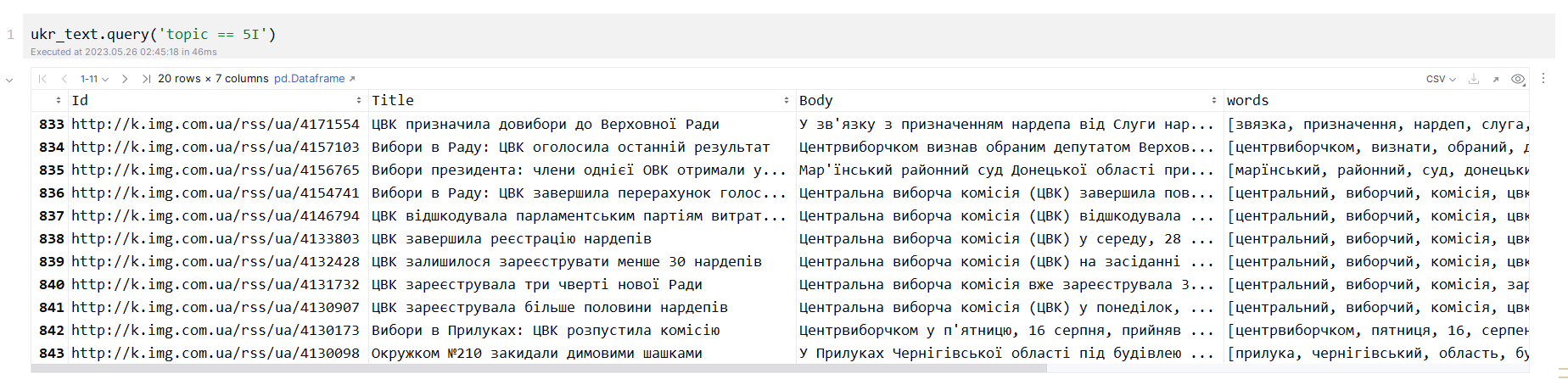


Рисунок 4.20 – Новини з категорією про вибори

І справді, всі новини якраз про вибори.

# Виконання другого додаткового завдання

Для початку повторимо всі дії з попередніх розділів.

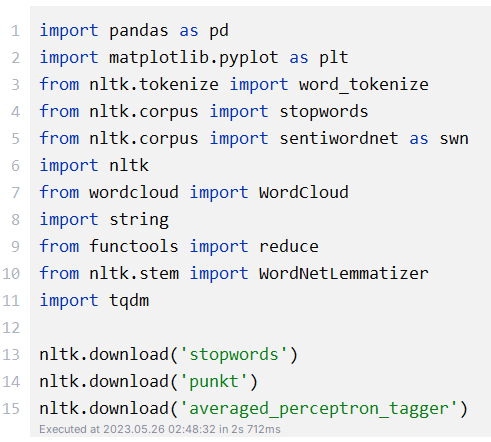


Рисунок 5.1 – Імпортування необхідних пакетів



Рисунок 5.2 – Визначення функцій для попередньої обробки текстів

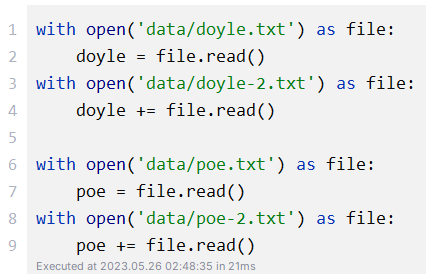


Рисунок 5.3 – Завантаження творів

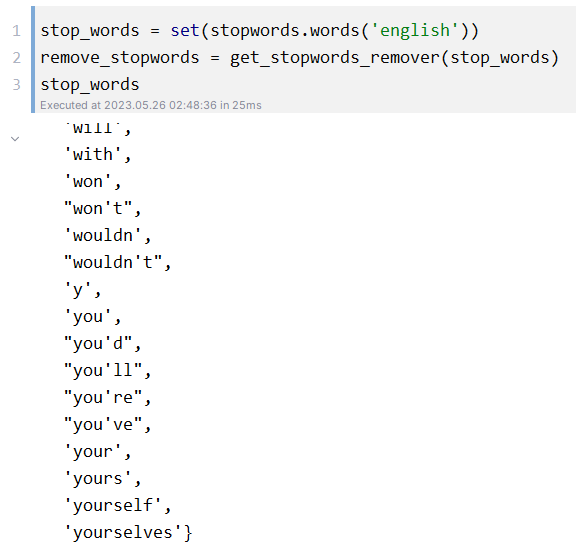


Рисунок 5.4 – Визначення стоп-слів для англійської мови

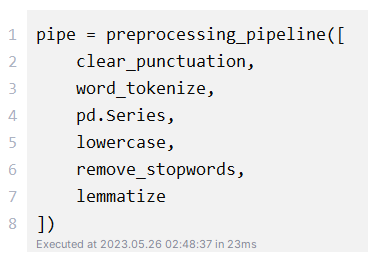


Рисунок 5.5 – Створення пайплайну для попередньої обробки текстів

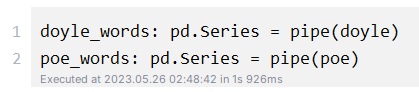


Рисунок 5.6 – Виконання попередньої обробки текстів

Виведемо графік частоти слів.

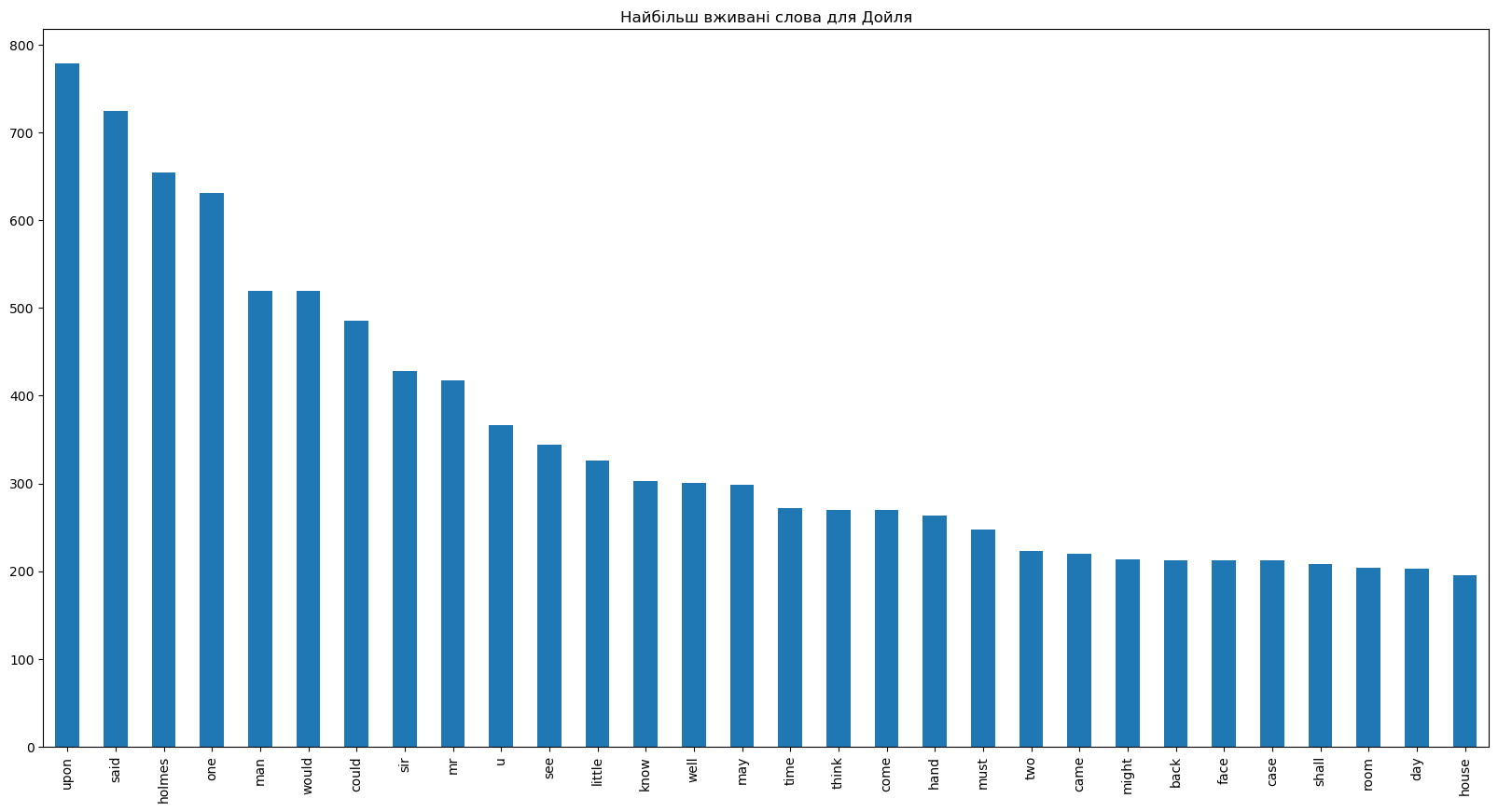


Рисунок 5.7 – Графік частот слів, що використовував Дойль

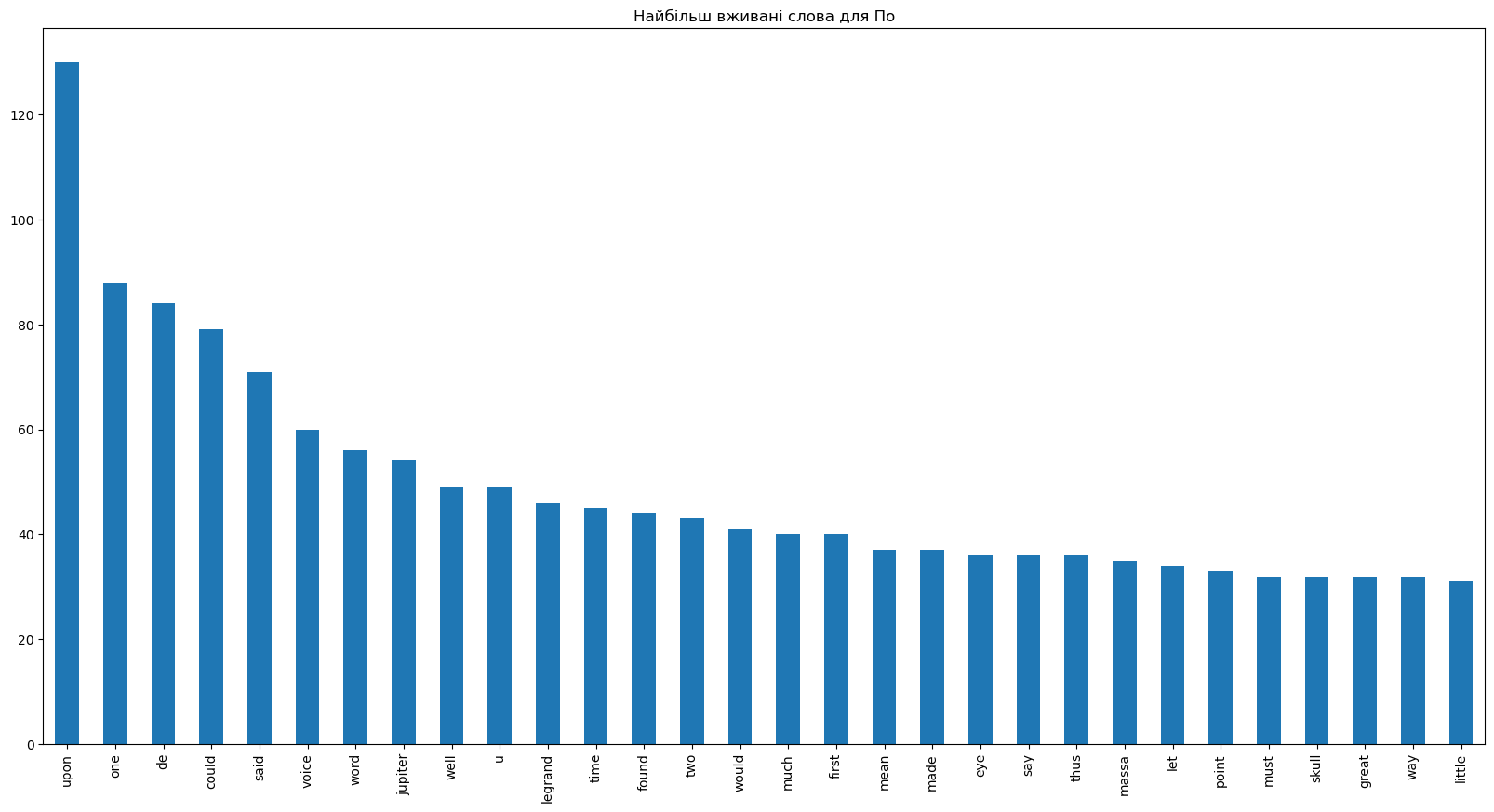


Рисунок 5.8 – Графік частот слів, що використовував По

Бачимо, на них велику кількість стоп-слів, що не були видалені. Видаляємо їх.

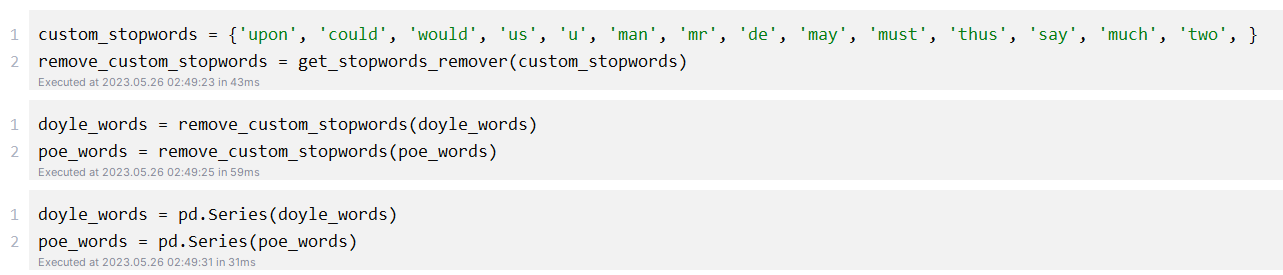


Рисунок 5.9 – Видалення додаткових стоп-слів

Будуємо хмари слів.

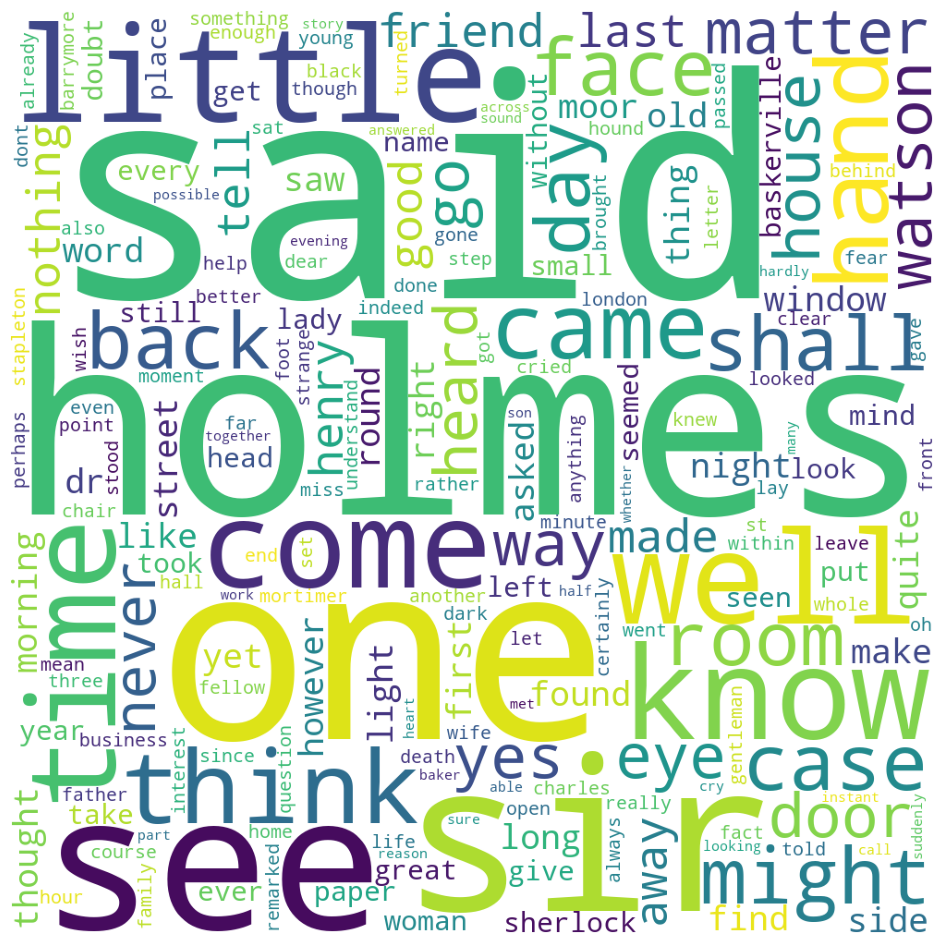


Рисунок 5.10 – Хмара слів творів Дойля



Рисунок 5.11 – Хмара слів творів По

Тепер обрахуємо «похмурість» для кожного з авторів. Для цього пройдемося кожним словом, що використовували автори, спочатку визначимо частину мови, до якої належить це слово, а потім за допомогою wordnet встановимо його тональність. Після обрахунку тональностей поділимо їх на загальну кількість слів, щоб збалансувати оцінку.



Рисунок 5.12 – Обрахунок похмурості для творів Дойля та По

Бачимо, що оцінки дуже близькі одне до одного, тому тут не можна сказати, що хтось писав більш похмурі оповідання, але все одно виведемо підсумок.

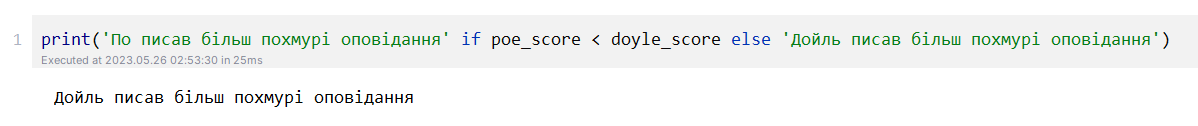


Рисунок 5.13 – Висновок про автора, який писав більш похмурі твори

# Висновок

У ході даної лабораторної роботи було отримано велику кількість практичних навичок із аналізу текстів. Я навчився виконувати попередню підготовку текстів: видалення розмітки, токенізацію слів, лематизацію. Навчився користуватися моделлю Bag of Words, обчислювати TF-IDF метрики. Також я провів обчислення тональності та категоризацію новин України за допомогою методу LSA. Для творів Дойля та По було намальовано хмару слів, обраховано тональність творів та отримано висновок, що Дойль писав більш похмурі твори, хоча різниця й невелика. При обчисленні тональності творів стикнувся з проблемою незбалансованості оцінок. Вирішив цю проблему шляхом ділення оцінки на загальну кількість слів у корпусі.