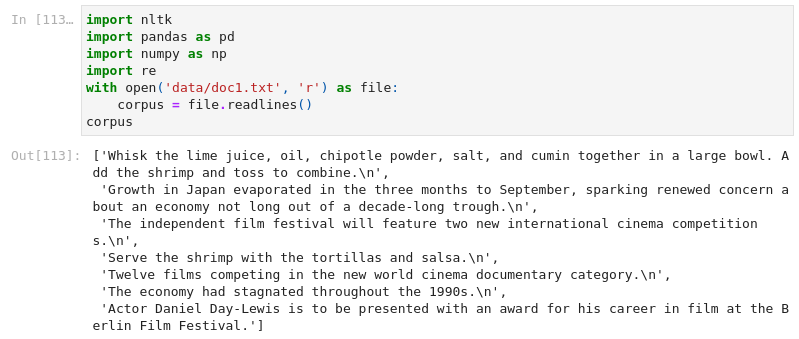
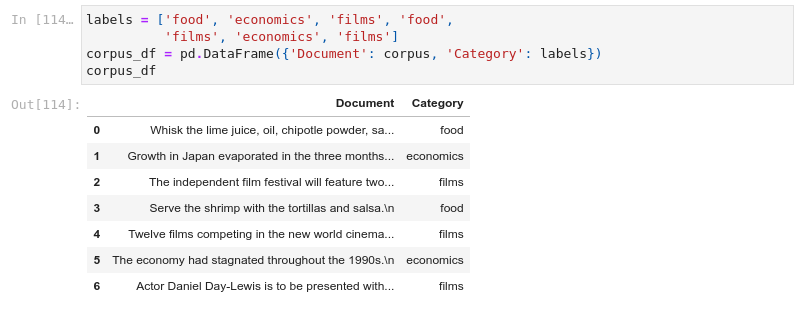
# Виконання

## Представити корпус як модель «Сумка слів». Вивести вектор для слова «film».

Зчитаємо текст з файлу та імпортуємо модулі nltk, pandas, numpy, re.

  
  
Рисунок 3.1 - Зчитування файлу

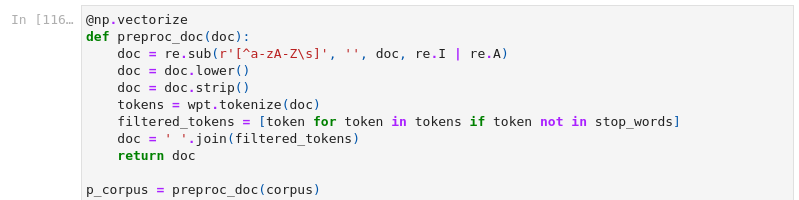
Задамо теми кожного документу.

  
  
Рисунок 3.2 - Теми документів

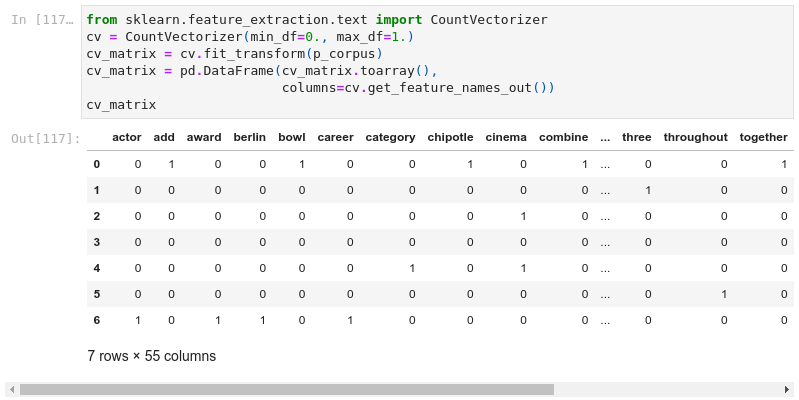
Визначимо стоп-слова англійської мови.

  
  
Рисунок 3.3 - Стоп-слова

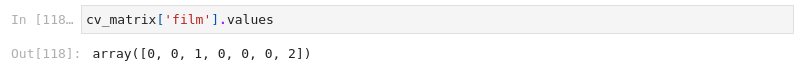
Визначимо функцію, що виконує попередню обробку документу. Застосуємо декоратор np.vectorize для того, щоб функція могла працювати з корпусами.

  
  
Рисунок 3.4 - Обробка документів

Представимо корпус як модуль "Сумка слів". Використаємо для цього клас CountVectorizer зі sklearn.feature\_extraction.text.

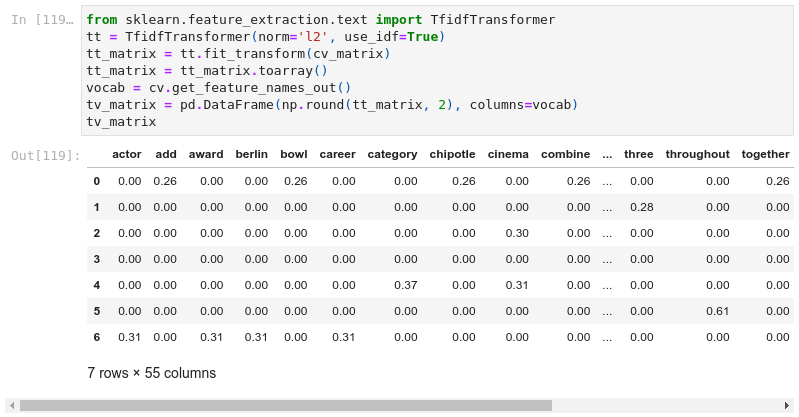
  
  
Рисунок 3.5 - Сумка слів

Виведемо вектор слова "film".

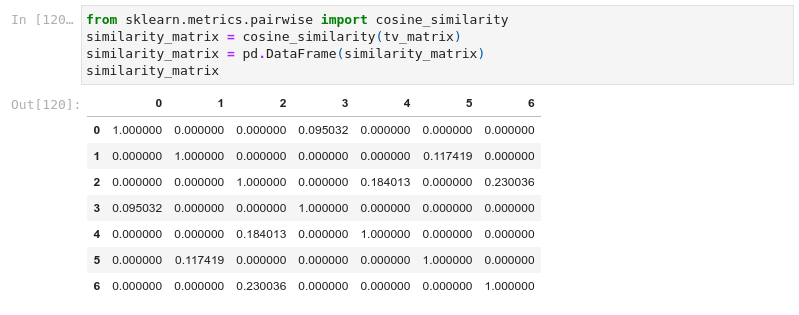
  
  
Рисунок 3.6 - Вектор слова "film"

## Представити корпус як модель TD-IDF. Спробувати кластеризувати документи за допомогою ієрархічної агломераційної кластеризації.

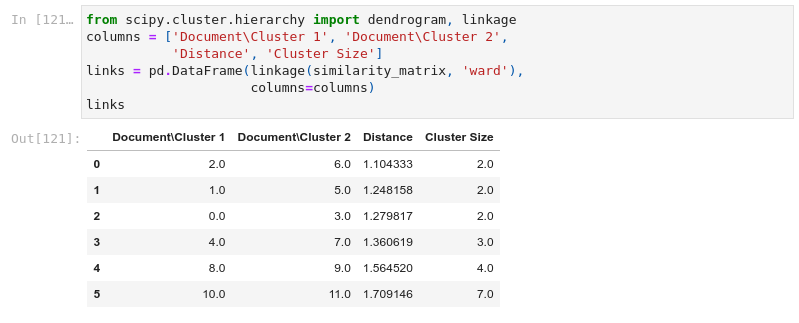
Перетворимо матрицю з частотою термінів на матрицю tfidf.

  
  
Рисунок 3.7 - Матриця TF-IDF

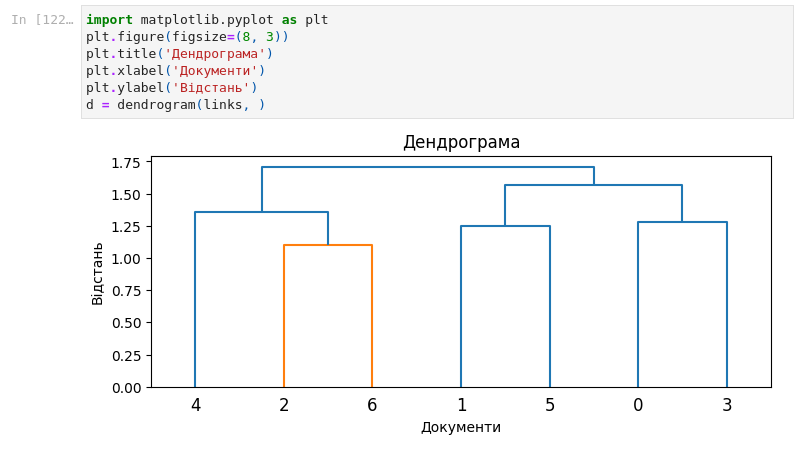
Обрахуємо подібність векторів за допомогою косинусної відстані.

  
  
Рисунок 3.8 - Матриця подібності

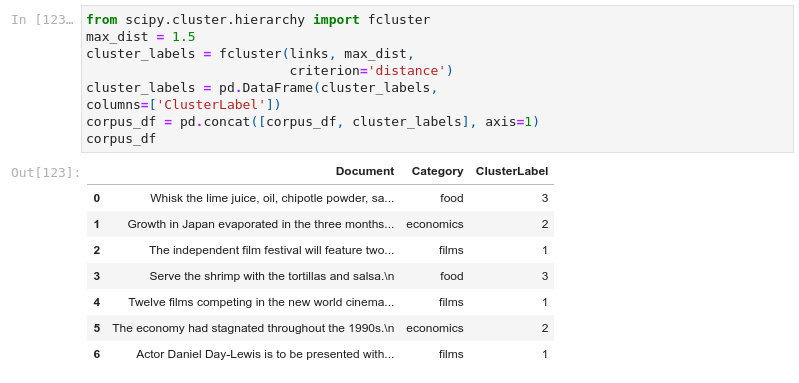
На основі матриці подібності побудуємо матрицю зв'язку.

  
  
Рисунок 3.9 - Матриця зв'язку

Візуалізуємо зв'язки за допомогою дендрограми.

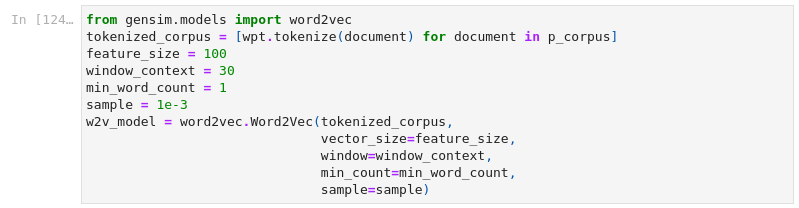
  
  
Рисунок 3.10 - Дендрограма

Отримаємо мітки кластерів. Бачимо, що кластери зливаються приблизно на відстані 1.5.

  
  
Рисунок 3.11 - Додавання міток

## Представити корпус як модель Word2Vec. Знайти подібні слова до слів shrimp, economy.

Побудуємо модель Word2Vec.

  
  
Рисунок 3.12 - Модель Word2Vec

Подивимося подібні слова до "shrimp", "economy".

  
  
Рисунок 3.13 - Виведення подібних слів