**Практическая работа №4**

Язык программирования: Python

Задание:

Реализовать поиск фильмов с похожим описанием.

1. Загрузка данных из файла **tmdb\_5000\_movies.csv**. Понадобятся только столбцы **original\_title** и **overview**. Есть несколько фильмов без какого либо описания – их нужно удалить.
2. Предобработка текстов:
   1. Произвести очистку текста.
   2. Очистить от стоп-слов.
   3. Лемматизировать слова (см. предыдущую работу).
3. Векторизация текстов. Используйте библиотеку **scikit-learn** и класс **TfidfVectorizer**. Метод **fit\_transform** принимает корпус текстов и преобразует его в матричное представление, где строки представляют тексты, а столбцы – все встречающиеся в этих текстах слова. Для каждого слова в тексте определяется его метрика TF-IDF. Таким образом, каждый текст будет представлен вектором, размер которого будет равен размеру словаря всех слов, которые встречаются в во всех текстах.
4. Отображение векторов текстов в двухмерном пространстве. Используйте метод **t-SNE**, который позволяет уменьшить размерность векторов до необходимой. Метод также реализован в библиотеке scikit-learn в виде класса TSNE. Получившиеся преобразованные векторы отобразить любой библиотекой для визуализации.
5. Реализовать функционал для поиска похожих текстов. Для этого необходимо сравнивать векторы с помощью косинусной меры сходства векторов и евклидова расстояния. Они также есть в виде метода **cosine\_similarity** и **euclidean\_distances** в библиотеке scikit-learn.

Чем ближе два вектора, тем более похожие тексты, представляемые этими векторами. Косинусное сходство определяется в диапазоне от 0 до 1, от наименее схожих к наиболее схожим соответственно. Результирующее значение евклидова расстояния варьируется от 0 до бесконечности, где 0 указывает на идентичные векторы, а большие значения указывают на большее различие между векторами.

1. Для примеров поиска брать фильмы из имеющихся данных (векторы для сравнения уже подготовлены), а также взять новые фильмы (описание фильма на английском языке можно отыскать на сайте IMDB) и векторизовать их (используя тот же векторизатор, что использовался для векторизации текстов из файла (метод transform)).

Объясните, в чем заключается суть TF-IDF.

Зачем выполнять векторизацию текстов?

Какие тексты можно сравнивать?

Какие меры сходства еще применимы для текстов?