**Практическая работа №3**

Язык программирования: Python

Библиотеки: библиотеки обработки текстовых данных (https://webdevblog.ru/podhody-lemmatizacii-s-primerami-v-python/), Pandas (для работы с csv файлами)

Наборы данных (на выбор):

<https://www.kaggle.com/datasets/reihanenamdari/youtube-toxicity-data>

<https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis>

<https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health>

<https://www.kaggle.com/datasets/kashishparmar02/social-media-sentiments-analysis-dataset>

Задание:

Выполнить загрузку данных.

Выбрать категориальный признак для анализа. В некоторых наборах есть только один признак – sentiment или status, которые обозначают, что текст с положительным, негативным или нейтральным настроением.

Очистить тексты, токенизировать, лемматизировать и провести стемминг.

Посчитать частоту слов (частотный словарь) для каждой категории.

Визуализировать полученные данные (с помощью частотной диаграммы (гистограмма) и облака слов).

Выполнение:

1. Загрузка данных

Загрузить набор данных при помощи библиотеки Pandas. Библиотека позволяет работать со структурированными данными, в частности с csv файлами.

Более подробно про библиотеку написано в документации: https://pandas.pydata.org/docs/index.html

Сначала необходимо установить библиотеку (если еще не установлена). Далее в проект импортируется сама библиотека (pd – псевдоним (короткое название) для библиотеки):



Далее загружаются данные из файла. Указывается путь к файлу и другие параметры. Если в файле данные разделены с помощью запятой, то разделитель можно не указывать. В результате будет создан объект класса DataFrame.



С помощью свойства columns можно посмотреть список столбцов в полученной таблице. Для работы необходимы столбцы Text и метки.

Обращение к столбцам:



Обращение к строке по индексу (индекс может быть и строкой!):



Перебор строк (row – объект класса Series):



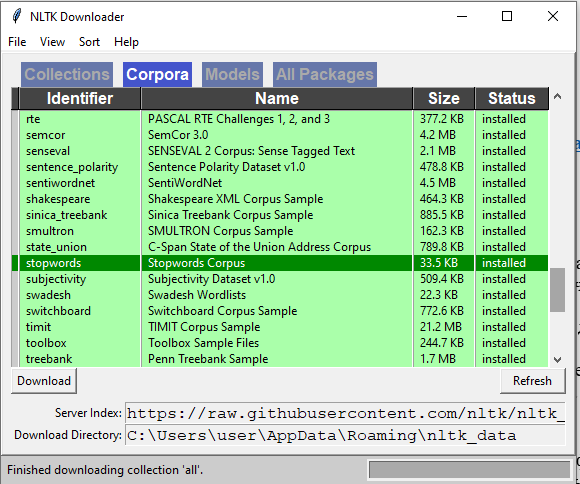
1. Предобработка текста, токенизация, лемматизация, стемминг

Сначала необходимо очистить текст от стоп-слов, лишних знаков, цифр, ссылок.

Для удаления стоп-слов можно воспользоваться библиотекой NLTK. Сначала необходимо вызвать загрузчик данных для библиотеки:



Далее в открывшемся окне выбираем необходимый пакет и загружаем:



Далее в программу можно загрузить сами слова, указав язык.



Токенизацию можно производить также с помощью библиотеки NLTK. В метод word\_tokenize передается строка, которая будет разбита на токены.



Слова привести к нижнему регистру.

Лемматизация с помощью библиотеки NLTK:



Работа с другими библиотеками описана по ссылке <https://webdevblog.ru/podhody-lemmatizacii-s-primerami-v-python/>. Необходимо для сравнения выбрать две библиотеки и выполнить работу по лемматизации текстов.

Стемминг необходимо произвести с помощью библиотеки NLTK, любыми двумя алгоритмами <https://spotintelligence.com/2022/12/14/stemming-python/>.

Для каждой записи в таблице должны получиться списки слов: без приведения к нормальной форме или получения основы (то есть без применения лемматизации и стемминга), два списка с разными библиотеками лемматизации, два – с двумя алгоритмами стемминга. Повторяющиеся слова не удалять!

1. Частота слов

Для каждой из категорий необходимо выделить частотные списки слов.

1. Визуализация

Построить частотную диаграмму и облако слов.

Для построения диаграммы можно воспользоваться стандартной библиотекой matplotlib либо какой-нибудь другой удобной вам.

Гистограмму можно построить с помощью методов bar и barh. Лучше использовать barh, чтобы слова были на оси OY (строки будут располагаться горизонтально). Данными для диаграммы должны являться слова и их частоты, которые были получены на предыдущем шаге. Для построения возьмите по 20 самых частых слов в категории и отсортируйте их по убыванию.

Как построить диаграмму по ссылке: <https://pythobyte.com/matplotlib-bar-plot-tutorial-and-examples-15307/>

Облако слов больше служит для быстро оценить содержание текстов, определить ключевые понятия и выделить основные темы. Более подробно как реализовать по ссылке: <https://skillbox.ru/media/code/vizualiziruy-eto-oblako-slov-na-python/>

Сравнить результаты по выборкам.

В чем плюсы и минусы лемматизации, стемминга?