# 4. Разработка web-приложения для сбора данных с Twitter

Как было описано выше, для прогнозирования событий с помощью Twitter требуются данные записей пользователей. Сбор данных может быть бесконечным процессом, если простым перебором искать нужные твиты, так как количество пользователей, по данным на конец ноября 2022 года составляет примерно 260 000 000 активных юзеров [112]. Наличие приложения для сбора данных значительно упрощает решение данной задачи. Есть большое количество профессиональных инструментов для сбора, часть которых была описана во второй главе данной работы. Но все они обладают своими особенностями: либо имеют платную подписку, либо требуют авторизации, а значит, предоставления своих данных, либо сложны в использовании, либо же не могут обойти API Twitter и блокируются на этапе повторного сбора.

В связи с этим формируется цель исследования - разработка своего web-приложения для пользования в локальной сети – хорошее решение. Для этого не требуются ни регистрация (если не предусмотрена самим разработчиком), ни оплаты, так как используются публичные инструменты, имеющиеся в общем бесплатном доступе, а также просты в использовании.

Исходя из вышеназванных параметров, были поставлены условия для написанного приложения:

1. Понятность кода – пользователь, кто желает дополнить приложение по своим целям при открытии кода, должен понимать, о чем речь идет в конкретном отрезке программы;
2. Кроссплатформенность – приложение должно работать на базе любой операционной системы (Windows, Linux, MacOS, ChromeOS и другие ОС);
3. Анонимность – приложение не требует авторизации и не блокируется API Twitter при сборе данных;
4. Бесплатность – утилита будет опубликована в открытый доступ.

Исходя из первых двух параметров, наиболее подходящим языком для написания программы является «Python». Разработка кода будет происходить в общедоступном ПО «Visual Studio Code», а само приложение будет собрано на localhost в браузере с помощью фреймворка «Streamlit».

Требуются определить типы данных, которые будут собираться программой. Исходя из разбора исследований прогнозирования, приведенных в третьей главе, главными параметрами будут: дата и ID твита, его текст и ID пользователя. Для корректного набора данных среди параметров поиска будут интервал времени твитов (дата начала и дата окончания) и ключевые слова.

Для начала работы импортируются данные из библиотек datetime, streamlit, snscrape.modules.twitter и pandas:

import snscrape.modules.twitter as sntwitter

import pandas as pd

import streamlit as st

import datetime as dt

Модуль datetime предоставляет классы для управления датами и временем, которые в будущем используются для определения временных рамок выборки твитов [113]. Модуль streamlit позволяет импортировать классы для последующего графического представления приложения через данный фреймворк [114]. Snscrape - парсер для служб социальных сетей (SNS). Он собирает такие вещи, как профили пользователей, хэштеги или ищет и возвращает обнаруженные элементы, например, соответствующие сообщения [115]. Аналитический модуль pandas здесь будет использоваться для отбора данных из представленного массива

Следом задаются массив для записи данных и задается его размерность:

maxTweets = 100

tweets\_list1 = []

Следом, для сбора данных используется TwitterSearchScraper для определения ключевых фраз с целью последующей выборки, а также добавление в список данных найденных твитов.

for i, tweet in enumerate(sntwitter.TwitterSearchScraper('from:').get\_items()):

    if i > maxTweets:

        break

    tweets\_list1.append(

        [tweet.date, tweet.id, tweet.content, tweet.user.username])

Записываются в первую строку шапка таблицы, а также формируется таблица формата «.csv», куда будут записываться данные из списка:

tweets\_df1 = pd.DataFrame(tweets\_list1, columns=['TweetTime', 'Tweet`s ID', 'txt', 'User'])

tweets\_df1.head()

tweets\_df1.to\_csv('tweets.csv', sep=',', index=False)

Для графического представления web-приложения, как писалось выше, используется фреймворк «Streamlit». Ниже приведен отрезок кода, где задаются параметры для данного фрейма:

st.title("Streamlit Twitter Scrapper")

today = dt.date.today()

tomorrow = today + dt.timedelta(days=1)

start\_date = st.date\_input("Start",today)

end\_date = st.date\_input("End",tomorrow)

if start\_date < end\_date:

    st.success("Start: '%s'\n\nEnd: '%s'" % (start\_date , end\_date))

else :

    st.error('Incorrect Value: End must fall after Start')

tweeter\_trending = st.text\_input("Enter text", "Type")

В конце задается кнопка действия, нажатие на которую выполняет процесс сбора и заполняет данные внутри таблицы.

if (st.button('OK')):

    result = tweeter\_trending.title()

    st.success(result)

После компиляции кода в терминале прописывается команда:

**streamlit run main.py**

В терминале появляется следующее сообщение и запускается графическое приложение кода:

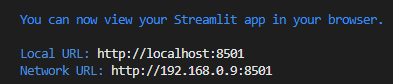




Рисунок 8 – Интерфейс веб-приложения

Теперь следует рассмотреть пример работы. Приложение подразумевает сбор данных в преддверии каких-либо событий. Например, ближайшее на момент написания работы событие – рождество. Вводя данные 2022/11/30; 2022/12/15; «Christmas» можно получить сведения, о чем рассуждали в разное время пользователи в ожидании данного события»:

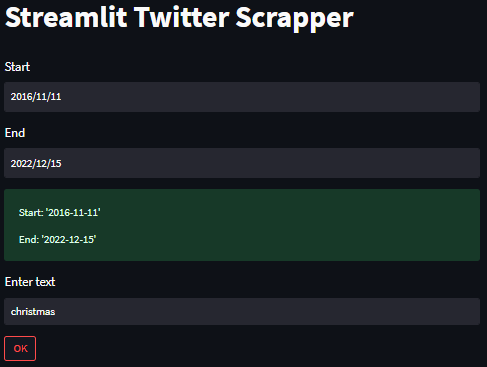


Рисунок 9 – Введение условий в приложение

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 10 – Результат запроса из приложения

Таким образом, разработанное в рамках данного исследования web-приложение, содержащее описанные выше элементы, способно искать твиты пользователей по ключевой фразе за определенный промежуток времени и представлять их в виде таблицы, при этом примитивно понятна в использовании, легко подается редактированию, не требует авторизации, доступен на любой платформе, так как для работы самого web-приложения требуется только браузер, а также обходит блокировку API за счет класса TwitterSearchScraper из модуля snscrape.modules.twitter.