

# **Twitter Data Analysis**

Applied Data Science in Python - Partial Result

Project 5

**Sebastian Mohr**

Student-ID: 3188691

Wintersemester 2022

# 1 Loading in of data

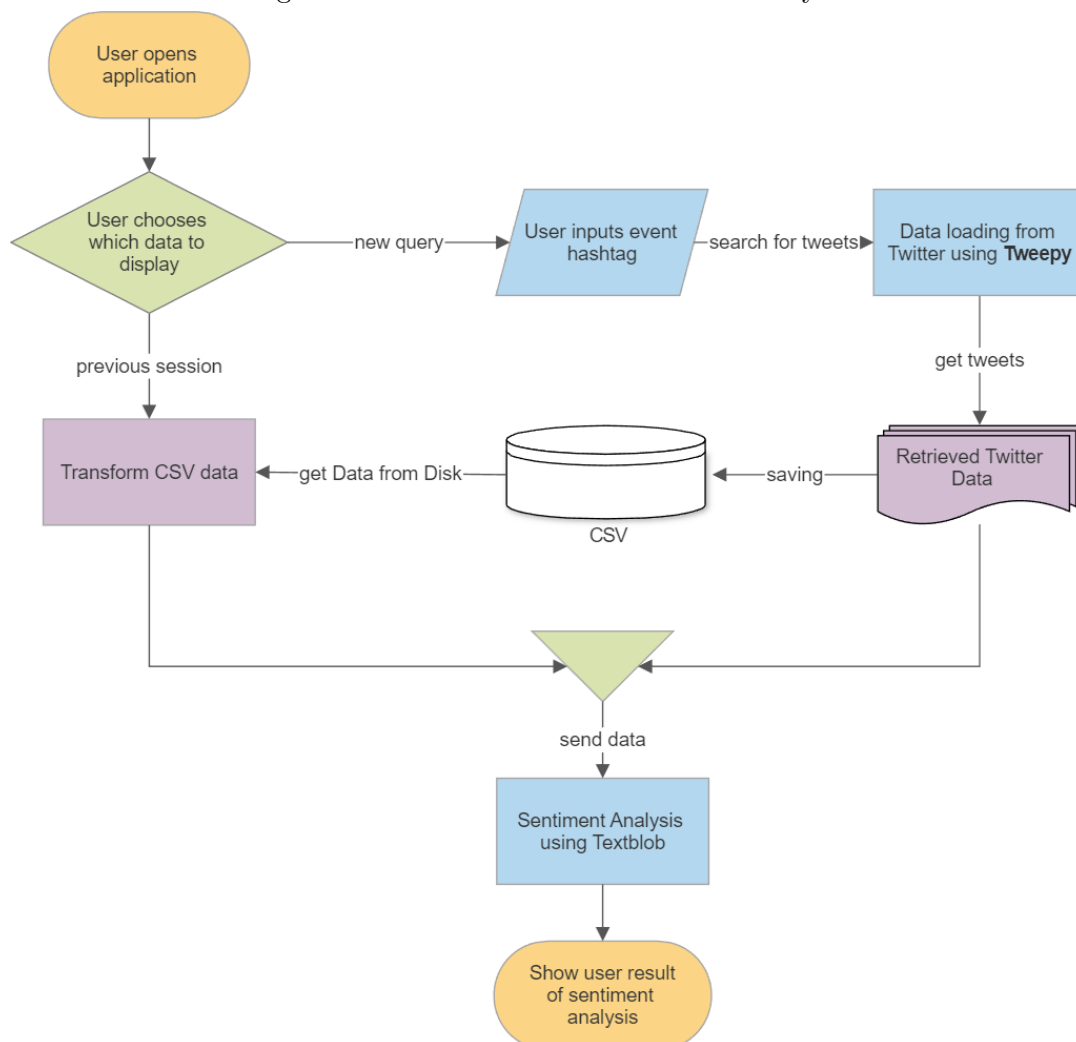
The data will be loaded through the Twitter API. The easiest way for this is to use the Python library **Tweepy** that provides different methods to access the Twitter API. The first step here will be the input of a specific events hashtag. After the tag has been put into the application, it will request a specific number of tweets for this event from the Twitter API. Currently I go with 50 tweets, but this number can be also set higher or lower, depending on preferences on how detailed the analyzed data should be.

To load the data, a **URL** has to be defined, which accesses the API at a specific endpoint. The URL also defines the query parameters, such as how many tweets are requested, which hashtags should be included in each tweet or which timeframe the tweet has to be created in. In **Tweepy** this process is done by creating a **Tweepy** Client, that includes the Bearer Token, through which the Twitter API can authenticate the current used dev-User. The defined query parameters then get put into the `client.search_recent_tweets(...)` method, that then starts an API call and retrieves all matching tweets. This method also takes in the definition of which fields of the tweets are necessary and will be included in the result. The data will then be stored in a CSV file to make it easily accessible in a later session.

## 2 Sentiment Analysis

After the tweets have been loaded into the application, they have to be analyzed to show them to the user. For this part of the project I will use the Python library **Textblob** that is specialized on natural language processing.

Figure 2.1: Flowchart of the sentiment analysis



After the user opens the application, they can choose to either start a new query by inputting a new hashtag or to view the previous session. When choosing the previous session, the stored data from the CSV will be loaded and transformed into a python object. When a new query is started, the application will load the data from twitter using **Tweepy**. After the data is ready, it gets sent into the sentiment analysis, that gets done through using **Textblob**. Afterwards the result gets displayed to the user.

### 3 Solution Examples

The goal is to give the user a feel on how the public opinion about a specific event is. For example if it's negative, the user can see which hashtags are trending alongside the event and what the peoples opinion about it is, by looking at top contributors in the requested data set.

To make the sentiment analysis easier readable for the user a bar will be displayed on top of the UI that shows the distribution between positive, neutral and negative tweets. `Textblob` also analyzes the polarity of the tweets which can be shown as well. The top users and hashtags will also be marked with a clearly visible color that indicates the sentiment of the respective tweets.

Figure 3.1: Mockup of the UI showing the Sentiment Analysis

