# **Progetto**

Il progetto permette di effettuare il recupero di tweet relativi allo specifico hashtag indicato al momento dell'esecuzione.

I tweet raccolti possono essere poi filtrati per rimuovere messaggi superflui e duplicati per poi essere sottoposti ad emotion recognition e ad una ricerca delle parole più frequenti.

## **Preparazione**

### **Python packages**

Creare un progetto nel quale inserire i file .py e .ui del codice sorgente e installare i packages necessari per il funzionamento delle varie componenti dell'applicazione:

* pandas
* nltk
* tweepy
* feel-it

pip install pandas

pip install nltk

pip install tweepy

pip install feel-it

### **Autenticazione**

Per accedere a Twitter è necessario creare un developer account le cui credenziali e token di accesso devono essere inserite all'interno di *scraping.py*

self.consumer\_key = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

self.consumer\_secret = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

self.access\_token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

self.access\_token\_secret = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

Questo account funzionerà da punto di partenza per la raccolta dei tweet

## **How to use**

### **Scraping**

Per iniziare la raccolta dei tweet sono possibili di metodi:

* Tramite interfaccia grafica (avviando ***main.py***)
* Tramite terminale (avviando ***scraping.py***)

python3 scraping.py

In entrambi i casi verrà poi richiesto di inserire l'hashtag su cui effettuare la ricerca e la data, che andrà a comporre il nome del file csv risultante.

Inserire hashtag sul quale effettuare la ricerca:

> #foo

Inserire la data nel formato yyyy-mm-dd:

> 2022-01-01

Il file risultante si chiamerà #foo\_2022-01-01.csv e contiene i 100 tweet più recenti contenente l'hashtag.

### **Filtering**

Il processo di filtraggio si impegna a rimuovere i tweet di spam o duplicati (retweet).

Prima dell'avvio assicurarsi che il file csv da filtrare si trovi nella stessa cartella di *filter.py* e che sia presente anche un file spam.csv che può essere utilizzato per contenere messaggi di spam che si desidera rimuovere col filtering.

Per avviarlo, eseguire da terminale ***filter.py***

python3 filter.py

Viene qui richiesto di digitare 1 per selezionare l'opzione di filtering e poi di inserire il nome del file csv da filtrare

Digitare 1 per eseguire filtering, 2 per aggiungere ID:

> 1

Inserire il nome del file da filtrare:

> #foo\_2022-01-01.csv

Vengono prodotti in output 2 file csv:

* #foo\_2022-01-01.csv\_spam.csv
* #foo\_2022-01-01.csv\_spam.csv\_dup.csv

Il primo contiene i tweet risultanti dopo la rimozione dei messaggi di spam ed il secondo quelli ulteriormente filtrati per rimuovere tutti i duplicati.

### **Merging**

Dopo che sono stati raccolti tweet in più occasioni e abbiamo una moltitudine di file csv filtrati, usiamo funzionalità di merging per riunire tutti i messaggi raccolti.

Tutti i file raccolti devono essere raccolti all'interno di una cartella che va collocata nella stessa posizione in cui troviamo *merger.py*

Per avviarlo, eseguiamo da terminale ***merger.py***

python3 merger.py

Indicare il nome della cartella in cui sono i file da unire:

> Bar

Viene prodotto il file Bar\_merged.csv all'interno della cartella Bar.

**IMPORTANTE:**

In seguito alla sua creazione è necessario effettuare una ridefinizione della struttura del file csv.

Per fare ciò avviamo *filter.py*, assicurandoci che Bar\_merged.csv si trovi nella sua stessa posizione, e selezioniamo l'opzione 2. Viene richiesto di il nome del file ed in output otteniamo Bar\_merged.csv\_id.csv:



### 

### 

### **Conteggio delle parole**

Partendo da un csv correttamente formattato e completo (vedere il file output dello step Merging), si inizia il processo di analisi delle parole più comuni all'interno dei tweet.

Prima dell'avvio assicurarsi che il file csv si trovi nella stessa cartella di *words-analyser.py*

Per avviarlo, eseguire da terminale ***words-analyzer.py***

python3 words-analyzer.py

Inserire il nome del file di cui si esegue il word count:

> Bar\_merged.csv\_id.csv

Vengono prodotti in output 2 file csv:

* Bar\_merged.csv\_id.csv\_token.csv
* Bar\_merged.csv\_id.csv\_token.csv\_words\_count.csv

Il primo è un file di *transizione* nel quale ogni tweet è suddiviso in lessemi escludendo tutte le componenti superflue del linguaggio ed i termini ritenuti non necessari.

Queste informazioni sono usate per la creazione del secondo csv, nel quale le parole contate vengono presentate in ordine di frequenza, affiancate dall'ID del tweet in cui sono state trovate. Parole con frequenza troppo bassa non vengono considerate.