

Università degli Studi di Milano – Bicocca

Corso di Laurea Magistrale di Data Science, a.a. 2020/2021



Le reazioni emotive degli utenti di Twitter alle notizie covid-19 in Italia

Report di Data Management, Giugno 2021

Daniele Quattrocchi – matricola 825418

Vittorio Menardo – matricola 812341

Raffaele Moretti – matricola 794537

Indice

1 Introduzione allo studio delle emozioni	5
2 Statistiche covid	10
3 Tweets covid	11
4 Data integration	11
5 Conclusioni	13
6 Riferimenti	13

ABSTRACT

Il COVID-19 ha avuto un forte impatto sulla sfera emotiva e sociale della vita quotidiana, al punto tale da ridefinire le nostre abitudini quotidiane, indipendentemente dall'età considerata. Ognuno di noi ha dovuto fare i conti con emozioni quali paura, stress, rabbia, noia/senso di limbo costante e in casi più gravi depressione. Queste emozioni si sono acuite soprattutto per coloro che, per sintomi/tampone risultato positivo, hanno contratto la malattia da Coronavirus (COVID-19). In questo progetto, abbiamo deciso di rilevare quali sono state le emozioni maggiormente espresse dagli utenti sulla piattaforma Twitter, analizzando esclusivamente tweets in lingua italiana, nel periodo compreso tra il 24 febbraio 2020¹ e il 2 maggio 2021. Dunque, la popolazione di riferimento sottoposta ad analisi è quella italiana. Dato che l'età minima per iscriversi a Twitter è di 13 anni, possiamo affermare che lo studio in questione si riferisce anche ad utenti con età inferiore a 18 anni. Tuttavia, per poter confrontare le reazioni degli utenti con le notizie in merito all'infezione da Coronavirus, abbiamo considerato il dataset in merito al bollettino ufficiale aggiornato al 31 maggio 2021, per poi estrarre i dati relativi al periodo di interesse. Integrate le informazioni di interesse per lo sviluppo del progetto, i risultati sono stati visualizzati mediante il software Tableau.

¹ Il 24 febbraio 2020 rappresenta la data in cui è stato pubblicato per la prima volta il bollettino della protezione civile in merito alla situazione del Coronavirus in Italia.

DISCLAIMER

Il progetto sviluppato è diviso in tre parti: la prima parte relativa allo studio delle emozioni, la seconda parte relativa alle statistiche in merito ai dati del bollettino ufficiale, la terza parte relativa ad analisi più approfondita di una componente della prima parte di studio. Dunque, di seguito verrà spiegata la ricerca e la gestione delle componenti della prima parte oggetto di studio.

1 Introduzione allo studio delle emozioni

La malattia da COVID-19 ha chiaramente portato ad un disagio emotivo. Nessuno è stato escluso dagli effetti emotivi di tale malattia, che ha colpito tutto il mondo. In particolare, l'idea alla base del nostro progetto si è concentrata nel valutare quali sono state le emozioni più espresse dalle persone durante il periodo temporale compreso tra il 24 febbraio 2020 e il 2 maggio 2021. A tal fine, abbiamo deciso di sviluppare il nostro progetto considerando la piattaforma Twitter ed analizzando esclusivamente i tweets in lingua italiana, in modo tale da poter valutare lo stato d'animo in Italia. Successivamente, abbiamo dovuto affrontare un problema fondamentale per lo studio in analisi, ossia come rilevare le emozioni sulla piattaforma. Per risolvere questo problema, abbiamo deciso di cercare tweets secondo i parametri di ricerca sopra descritti (lingua e riferimento temporale) che contenevano la parola COVID e la parola che indica l'emozione considerata. In questo modo, si evitano fenomeni di emozioni non correlate con la malattia da coronavirus. Successivamente, abbiamo affrontato il tema di quali parole, espressive dell'emozione, considerare in quanto è chiaro che gli utenti possono usare parole diverse per esprimere la stessa emozione. Per risolvere questo problema, abbiamo considerato la teoria della categorizzazione delle emozioni² dello psicologo Daniel Goleman, che dunque considera le seguenti categorie di emozioni:

- **Paura**, che contiene le parole: ansia, timore, nervosismo, preoccupazione, apprensione, cautela, esitazione, tensione, spavento, terrore; come stato psico-patologico, fobia e panico
- **Gioia**, che contiene le parole: felicità, godimento, sollievo, contentezza, beatitudine, diletto, divertimento, fierezza, piacere sensuale, esaltazione, estasi, gratificazione, soddisfazione, euforia, capriccio e, al limite estremo, entusiasmo maniacale
- **Amore**, che contiene le parole: accettazione, benevolenza, fiducia, gentilezza, affinità, devozione, adorazione, infatuazione, agape.
- **Sorpresa**, che contiene le parole: shock, stupore, meraviglia, trasecolamento.
- **Disgusto, che contiene le parole:** disprezzo, sdegno, aborrimiento, avversione, ripugnanza, schifo.
- **Vergogna**, che contiene le parole: senso di colpa, imbarazzo, rammarico, rimorso, umiliazione, rimpianto, mortificazione, contrizione.

Per lo studio di queste categorie, abbiamo deciso di considerare la parola che indica la categoria come elemento costituente della categoria stessa.

² Per ulteriori informazioni visita il seguente articolo: <https://www.giuliamartino.it/le-otto-emozioni-primarie/>

1.1 Tecniche utilizzate

Al fine di estrarre i tweets dalla piattaforma di Twitter, secondo i parametri di ricerca sopra descritti, abbiamo deciso di utilizzare la tecnica dello *scraping*, mediante la libreria Python *snsrape*. Tuttavia, riteniamo utile presentare al lettore quali sono state le decisioni che hanno reso necessario l'utilizzo di tale libreria.

Inizialmente abbiamo considerato le API che Twitter fornisce gratuitamente agli studenti, ricercatori e così via. Tuttavia, tali API sono soggette a forti restrizioni che impedivano di fatto lo sviluppo di tale progetto. Infatti, tali interfacce di programmazione consentono di scaricare tweets in tempo immediatamente reale al loro utilizzo o comunque fino a 7 giorni precedenti a quello odierno. Inoltre, esse hanno restrizioni sul numero di richieste al server per lo scaricamento dei tweets.

Dunque, i problemi che la tecnica di *scraping* doveva risolvere erano i seguenti:

- Accedere al server di Twitter per poter scaricare tweets storici (ossia Tweets pubblicati nei giorni, mesi, anni precedenti a quello attuale)
- Risolvere il problema del numero di richieste al server per scaricare i dati.

Per tale ragione, abbiamo deciso di considerare le API premium, che tuttavia presentano lo stesso problema delle API fornite gratuitamente in quanto:

- Tali API consentono di accedere ai dati del server fino a trenta giorni prima da quello attuale
- Il limite al numero di richieste permane
- Sono a pagamento

Dunque, per accedere ai dati storici senza limitazioni, è necessario pagare opzioni nell'*API premium* o *enterprises* che consentono di scaricare l'intero archivio di tweets.

Impossibilitati in questa soluzione, abbiamo deciso di affidarci allo *scraping* tramite librerie Python non ufficiali per lo scaricamento dei tweets. Dunque, non abbiamo utilizzato la libreria *Tweepy*. Questa scelta ha avuto un lato negativo: la mancata applicazione di uno streaming in tempo reale. Dunque, non è stato possibile utilizzare la tecnologia *Kafka*, tecnologia che consente di gestire in due tempi differenti (producer e consumer) i dati in streaming.

La soluzione successiva ricercata è stata dunque utilizzare la libreria *Selenium*. Tale libreria è molto complessa, consente di aprire dal notebook una pagina del browser microsoft edge, accedere a twitter e mandare richieste senza limite di scaricamento dei dati. Tuttavia, questa soluzione è stata ritenuta fin troppo complessa e controproducente per lo sviluppo di questo progetto. Per tale ragione, abbiamo optato per utilizzare la libreria *snscraper*. Tale libreria necessita di un'attenta configurazione iniziale, tuttavia è molto semplice fare *scraping* utilizzando questa libreria. È necessario solamente indicare il numero massimo di tweets da estrarre (`maxTweets`) e i parametri di ricerca dei tweets, come si evince dalla seguente immagine.

```
In [2]: maxTweets = 100000
tweets=[]

for i,tweet in enumerate(sntwitter.TwitterSearchScraper('covid paura lang:it until:2021-05-03 since:2020-02-24').get_items()):
    if i>maxTweets:
        break
    tweets.append({
        "data":pd.to_datetime(datetime.datetime.date(tweet.date)),
        "id":tweet.id,
        "username":tweet.user.username,
        "display_name":tweet.user.displayname,
        "testo":tweet.content,
        "reply_count":tweet.replyCount,
        "like_count":tweet.likeCount,
        "retweet_count":tweet.retweetCount,
        "quote_count":tweet.quoteCount,
    })
```

Il tempo che viene impiegato per scaricare i tweets dipende dal numero di tweets che effettivamente contengono i parametri di ricerca indicati. I dati estratti sono in formato JSON e li abbiamo opportunamente convertiti in formato CSV per poterli analizzare e pulire, per poi integrarli con altri dati (*fase data preparation: - data quality, - data integration*).

1.2 Preparazione dei dati - selezione emozioni

La spiegazione di questo paragrafo trova applicazione nello script 'selezione_emozioni'.

Prima di applicare la tecnica dello scraping per l'estrazione dei dati, abbiamo deciso di trovare un metodo per la selezione delle parole legate alle emozioni in ciascuna categoria. Tale funzione utilizza lo scraping ma senza estrarre effettivamente i tweets. Essa, infatti, conta solamente il numero di tweets associata a ciascuna parola, considerati i parametri di ricerca sopra descritti. Successivamente, tali parole chiave sono state ordinate in funzione del conteggio di tweets ad esse associate. Un esempio di quanto appena descritto per la categoria vergogna è mostrato nella seguente immagine:

9. Conteggio categoria 'vergogna'

```
In [18]: lista_di_parole = [
        'vergogna',
        'senso di colpa',
        'imbarazzo',
        'rammarico',
        'rimorso',
        'umiliazione',
        'rimpianto',
        'mortificazione',
        'contrizione']

        diz_conteggio = conteggio_tweets(lista_di_parole)
        diz_conteggio = {k: v for k, v in sorted(diz_conteggio.items(), key=lambda item: item[1], reverse=True)}
        diz_conteggio

Out[18]: {'vergogna': 9492,
          'imbarazzo': 489,
          'senso di colpa': 323,
          'rammarico': 205,
          'umiliazione': 94,
          'rimpianto': 91,
          'rimorso': 53,
          'mortificazione': 23,
          'contrizione': 3}
```

Per ciascuna categoria abbiamo deciso di considerare solamente le parole che avevano il numero maggiore di Tweets. I risultati sono i seguenti:

- Amore, per la categoria amore
- Dolore, per la categoria tristezza
- Gioia, per la categoria gioia
- Odio, per la categoria collera
- Paura per la categoria paura
- Schifo, per la categoria disgusto
- Shock, per la categoria sorpresa
- Vergogna, per la categoria Vergogna

1.3 Data quality

Per ciascuna parola sopra descritta si è proceduto ad estrarre i tweets e convertendo il formato JSON dei dati in formato CSV. Successivamente, si è proceduto ad affrontare il problema principale della pulizia dei dati. Infatti, successivamente, il dataset è stato pulito in quanto la ricerca dei tweet contenenti le parole covid e paura presenta un problema da non sottovalutare. La ricerca permette sì l'estrazione dei tweet contenenti la parola legata all'emozione insieme alla parola covid, tuttavia queste parole possono essere:

- Nel testo del tweet
- Nello username
- Nel display name

Per questa ragione, è probabile avere diverse combinazioni di quanto espresso sopra, portando all'estrazione di tweet di utenti che sulla piattaforma hanno il nome PAURA e nel testo parlano tutt'altro rispetto alla paura da covid. Nonostante ciò, tweet di questo tipo sono stati piuttosto rari nell'analisi di tutte le parole chiave. Per risolvere questo problema, abbiamo deciso di pulire il dataset tenendo solamente i tweet che contenevano le parole covid e paura solamente nel testo del tweet.

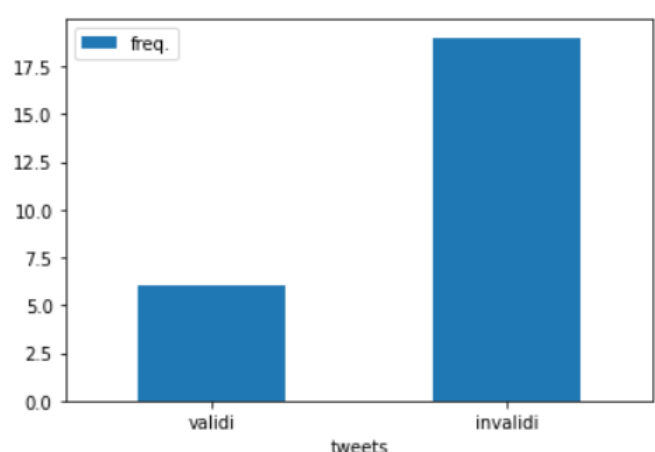
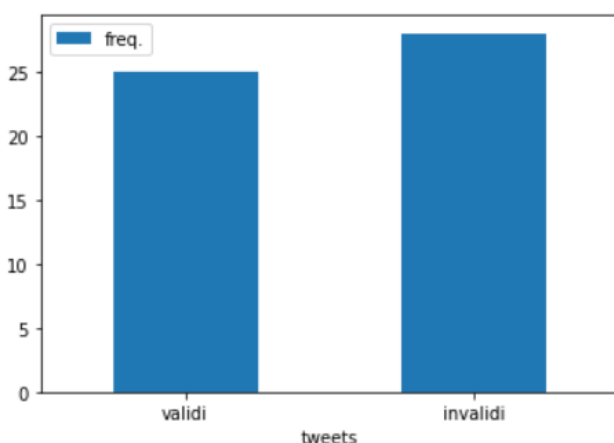
Successivamente, si è proceduto ad analizzare un campione dei tweet corrispondente all'1% dei tweet totali estratti. Dunque, per la parola paura sono stati analizzati circa 308 tweet. Il fine di questa procedura è individuare se nei tweet è presente ironia o negazione dell'emozione del tipo:

- *"Non ho paura"*, per la parola paura
- *"Mai una gioia"*, per la parola gioia
- *"L'amore prima del covid + emoji che ride"*, per la parola amore

e così via.

Per accettare la validità di tutto l'insieme di tweets, abbiamo deciso di studiare se il numero di tweets analizzati nel campione risultati come invalidi superavano il numero dei tweets classificati come validi.

Le parole Amore e Gioia sono risultate estremamente soggette all'ironia degli utenti, rendendo lo studio di queste emozioni non valido. Pertanto, non si è proceduto ulteriormente nell'analisi di queste due emozioni, tenendo solamente sei emozioni in totale sottoposte ad analisi. Di seguito, l'immagine di sinistra rappresenta l'analisi del campione dei tweets relativi alla parola amore, quella di destra per la parola gioia.



Successivamente, abbiamo proceduto a raggruppare i tweet per giorni, generando un conteggio per ogni giorno, relativo all'emozione analizzata (conteggio giornaliero non cumulato). Tuttavia, si è presentato un problema ossia una mancanza temporale tra i giorni raggruppati. In altri termini, per le parole chiave analizzate, vi sono stati giorni in cui non ci sono stati tweet in merito alle parole chiave considerate. Dunque, abbiamo risolto questo problema introducendo i giorni mancanti nel dataframe e assegnando a questi giorni il conteggio 0. Successivamente, i dati sono stati convertiti nella struttura JSON tramite un'opportuna funzione, che di fatto mantiene il tipo di dato come testo. Questo è stato necessario per poter caricare su MongoDB locale i dati testo che sono stati poi opportunamente riconosciuti/convertiti in formato JSON. Nella cartella allegata, i dati in questione sono nella sottocartella emozioni_db, che prende il nome del database creato in MongoDB locale.

1.4 MongoDB – Studio 3T

Per la gestione dei dati in formato JSON abbiamo deciso di utilizzare MongoDB, database non relazionale (NoSQL) di tipo documentale. Questa decisione è dovuta al fatto i tweets estratti sono in formato JSON. Nonostante avessimo potuto mantenere il formato CSV per operare la pulizia dei dataset, abbiamo deciso di mantenere il formato originale, per memorizzare i dati in MongoDB. Al fine di utilizzare tale database non relazionale, abbiamo proceduto a installare MongoDB localmente. Tuttavia, al posto che operare utilizzando la shell di MongoDB, si è deciso di installare STUDIO 3T, interfaccia GUI che permette di eseguire numerosi tasks in modo efficiente semplice e veloce. Infatti, tale interfaccia permette di:

- Utilizzare la query builder, componente che permette di costruire query sui dati senza dover conoscere il linguaggio di MongoDB.
- Importare/ esportare dati in formato nativo o non nativo su MongoDB (come CSV)
- Visualizzare i dati in formato tabellare o ad albero
- Modificare i dati direttamente tramite l'interfaccia
- Eseguire manualmente query con linguaggio SQL, che viene opportunamente convertito in linguaggio MongoDB

Queste sono le funzioni che principalmente abbiamo utilizzato tramite STUDIO 3T. Tuttavia, è opportuno evidenziare che la licenza in uso per questo progetto è stata fornita gratuitamente da tale azienda in quanto siamo studenti universitari. Tale licenza vale un anno dopo di che l'utilizzo di questa interfaccia è a pagamento. Il costo per l'utilizzo di STUDIO 3T è molto alto, in quanto è maggiormente previsto come utilizzo in grandi aziende e non da privato.

2 Statistiche covid

La descrizione di questo paragrafo è relativa al codice contenuto nello script 'italia_script_definitivo', nella sezione script della cartella allegata

Al fine di comprendere la reazione emotiva degli utenti alle notizie in merito al COVID-19, abbiamo deciso come enunciato nell'abstract del progetto a scaricare i dati ufficiali del bollettino della protezione civile, aggiornato al 31 maggio 2021. Di tale dataset sono state tenute solamente le colonne di principale interesse per lo studio e a scopo esplorativo, ossia:

- Nuovi positivi (giornalieri, non cumulati)
- Deceduti (cumulati)

Successivamente, sono stati estratti dal dataframe i record compresi nel periodo di analisi sopra descritto, per avere concomitanza con il periodo di analisi con i tweets relativi alle emozioni durante la pandemia.

A questo punto, i dati sui deceduti sono stati calcolati giornalmente, in quanto tale decisione è necessaria perché i conteggi dei tweets sono giornalieri. Questa decisione è dovuta al fatto che per poter analizzare lo stato d'animo della piattaforma è necessario considerare dati giornalieri non cumulati.

A fine esplorativo, si è deciso di importare sul notebook l'ultimo dataset di Our World in Data tramite Github, in modo tale da operare sui dati aggiornati. Di tale dataframe, abbiamo deciso di tenere solamente la colonna relativa ai casi in tutto il mondo di nuovi positivi al coronavirus, nominata 'casi_mondo'. Tale colonna presenta conteggi giornalieri in funzione del giorno di riferimento (i.e. non cumulato). Inoltre, da Our World In Data, tramite github, abbiamo importato i dati relativi alle vaccinazioni in Italia. Da questo dataset abbiamo preso la colonna 'total_vaccinations' (conteggio cumulato). Dopodiché si sono calcolati i dati giornalieri, come per quanto riguarda i dati sui deceduti sopra descritti.

Ottenuti i dataframe sopra descritti, abbiamo integrato i dati mediante la colonna 'data'. In questo modo abbiamo ottenuto il dataset finale per le statistiche COVID-19. Tali dati sono stati prontamente convertiti nella struttura JSON per poi essere importati localmente su MongoDB.

3 Tweets covid

La spiegazione di questo paragrafo trova applicazione nello script 'covid_tweets_script_definitivo'.

Per concludere l'analisi, a scopo esplorativo si è deciso di estrarre dalla piattaforma Twitter i tweets in lingua italiana contenente la parola covid, senza specificare altre parole di ricerca, tra il 24 febbraio 2020 e il 2 maggio 2021. Il codice è riportato nella cartella 'estrazione tweets covid'. I dati estratti sono più di tre milioni e il tempo impiegato per estrarli è stato di circa 13 ore.

I dati estratti sono stati innanzitutto puliti, tenendo solamente i tweets che contenevano la parola covid almeno nel testo del tweet. Successivamente tali dati sono stati raggruppati per giorno generando un conteggio giornaliero e sono stati poi convertiti nella struttura JSON (mantenendo comunque il tipo di dato testuale). In questo modo, i dati sono poi stati caricati in mongoDB locale. Nella cartella allegata, i dati in questione sono nella sottocartella covid_tweets_db, che prende il nome del database creato in mongoddb locale.

4 Data integration

Dopo aver esportato i dati in mongoDB locale, si è proceduto ad importare i conteggi relativi alle singole emozioni in uno script notebook, contenuto nella cartella 'integrazione e dati finali' e nominato 'integrazione_emozioni_script_definitivo'. Importati i dati, questi sono stati convertiti dal formato JSON al formato CSV, per poi essere mergiati sulla colonna 'data'. In questo modo si è ottenuta una tabella contenente per ciascun giorno il numero di conteggi di tweets contenenti l'emozione indicata nella colonna. Successivamente tali dati sono stati poi convertiti nella struttura JSON ed esportati in mongoDB locale. Nella cartella allegata, i dati in questione sono nella sottocartella dati_integrati_db, che prende il nome del database creato in mongoddb locale. Si mostra di seguito il dataframe ottenuto.

	data	nuovi_positivi	nuovi_deceduti	nuovi_vaccinati	casi_mondo	paura	dolore	vergogna	schifo	odio	shock	covid
0	2020-02-24	221	7	0	1766	74	3	11	17	4	1	4397
1	2020-02-25	93	3	0	2662	21	0	0	2	4	0	1799
2	2020-02-26	78	2	0	3075	25	0	3	2	0	0	1355
3	2020-02-27	250	5	0	4330	18	1	4	4	0	0	1302
4	2020-02-28	238	4	0	4442	13	1	3	2	0	1	1139
...
429	2021-04-28	13385	344	407080	2852389	65	4	30	14	3	1	6914
430	2021-04-29	14320	288	527247	2817990	65	6	17	17	9	0	6181
431	2021-04-30	13446	263	525920	2740812	71	3	12	12	10	8	6479
432	2021-05-01	12965	226	432490	2481260	65	4	11	12	2	1	5003
433	2021-05-02	9148	144	369395	2074469	59	3	39	33	7	5	5251

434 rows × 12 columns

Successivamente si è creato uno script analogo a questo indicato, dove i conteggi per ciascuna colonna sono stati resi cumulati (i.e. il conteggio del giorno 2 contiene anche il conteggio del giorno 1). Inoltre, si è proceduto ad un unpivot della tabella cumulata in modo da avere le emozioni come valori al quale sulla stessa riga vengono associati i conteggi giornalieri di tweet. Infine, la procedura di conversione del tipo di dato ed esportazione in mongoDB locale è analoga a quella descritta sopra. Questi dati creati sono fondamentali per la produzione del grafico race chart, che verrà spiegato opportunamente nel report di data vizualization. Nella cartella allegata, i dati in questione sono nella sottocartella data_per_viz, che prende il nome del database creato in mongoddb locale. Si mostra di seguito il dataframe ottenuto.

	data	emozioni	conteggio
0	2020-02-24	paura	74
1	2020-02-25	paura	95
2	2020-02-26	paura	120
3	2020-02-27	paura	138
4	2020-02-28	paura	151
...
2599	2021-04-28	shock	2236
2600	2021-04-29	shock	2236
2601	2021-04-30	shock	2244
2602	2021-05-01	shock	2245
2603	2021-05-02	shock	2250

2604 rows × 3 columns

Analogamente alla procedura indicata sopra, i dati contenenti i conteggi delle emozioni giornalieri non cumulati sono stati importati in uno script notebook contenuto nella stessa cartella ma nominato 'integrazione_emozioni_script_parte_2'. Inoltre, in questo script sono stati importati da mongoddb locale anche i dati relativi alle statistiche covid e i conteggi giornalieri dei tweets contenenti la parola covid (e non altre parole di ricerca). I dati sono stati convertiti in formato CSV, mergiati sulla colonna data. Tali dati sono stati poi calcolati in media mobile a 7 giorni e successivamente riconvertiti nella struttura JSON per poi essere esportati in mongoddb locale. Nella cartella allegata, i dati in questione sono nella sottocartella dati_integrati_db, che prende il nome del database creato in mongoddb locale. Si mostra di seguito il dataframe ottenuto.

	data	nuovi_positivi	nuovi_deceduti	nuovi_vaccinati	casi_mondo	paura	dolore	vergogna	schifo	odio	shock	covid
0	2020-02-24	221	7	0	1766	74	3	11	17	4	1	4397
1	2020-02-25	157	5	0	2214	48	2	6	10	4	0	3098
2	2020-02-26	131	4	0	2501	40	1	5	7	3	0	2517
3	2020-02-27	160	4	0	2958	34	1	4	6	2	0	2213
4	2020-02-28	176	4	0	3255	30	1	4	5	2	0	1998
...
429	2021-04-28	12860	323	368114	2591750	64	8	18	18	7	5	6798
430	2021-04-29	12613	312	386558	2587052	63	8	17	16	7	4	6508
431	2021-04-30	12425	301	403990	2571640	63	7	17	15	7	5	6422
432	2021-05-01	12303	287	410635	2559282	65	6	17	14	7	5	6295
433	2021-05-02	11730	277	425198	2533816	66	5	20	17	6	6	6241

434 rows × 12 columns

5 Conclusioni

Alla luce di quanto spiegato fin ora, possiamo concludere che siamo riusciti a rispondere alle domande che hanno originato questo progetto. In particolare, avremmo voluto considerare più emozioni per categoria, ma questo avrebbe richiesto analisi molto complesse, come la misurazione dell'andamento delle serie storiche di ciascuna parola nella categoria, al fine di valutare effettivamente se le parole di ciascuna categoria venivano usate come sinonimi. L'impossibilità di questa decisione è principalmente data dalla mancanza di conoscenze nell'ambito delle serie storiche. Tuttavia, possiamo ritenerci soddisfatti dei risultati ottenuti, in quanto siamo comunque riusciti a mostrare le emozioni effettivamente più espresse su Twitter. Interessante sarebbe il fatto di mostrare questi risultati ad un campione di persone che non usano Twitter, per valutare effettivamente se i risultati ottenuti possono essere estesi anche ad un gruppo più ampio di persone.

6 Riferimenti

www.mongodb.com

<https://pymongo.readthedocs.io/en/stable/>

<https://www.tableau.com/it-it>

<https://studio3t.com/>

<https://www.youtube.com/>

<https://twitter.com/?lang=it>