Università degli Studi di Milano Bicocca

Scuola di Scienze

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di Laurea Magistrale in Data Science

Twitter Polarization and Vaccines

Armanini Justin – 830103 – j.armanini@campus.unimib.it
Caiffa Emanuele – 872515 – e.caiffa@campus.unimib.it
Palomba Eleonora – 876479 – e.palomba4@campus.unimib.it
Zayeva Nataliya – 867981 – n.zayeva@campus.unimib.it

Abstract

Background: Per tutta la durata del 2021, il tema delle vaccinazioni, con tutte le sue controversie, i dubbi e le perplessità di buona parte dei cittadini, ha occupato un ruolo di rilievo all'interno del dibattito pubblico. In un simile scenario i social network, che con le opportune premesse possono essere considerati uno specchio della società, rappresentano una buona opportunità per cercare di studiare la percezione del fenomeno e le opinioni che le persone hanno in merito.

Obiettivi: Il progetto qui presentato è stato realizzato allo scopo di provare a rispondere alle seguenti domande di ricerca:

- Chi sono gli utenti italiani più "influenti" in materia di vaccini? Essi variano nel corso delle settimane?
- Quale è il lessico più frequente quando si parla di vaccini?
- È possibile osservare una "polarizzazione" del dibattito? Esistono comunità di pro-vax e no-vax che restano ancorate alle proprie posizioni rifiutando il confronto con idee contrastanti?

Metodo: La realizzazione dell'architettura necessaria per la raccolta dei dati utili alle analisi ha affrontato il problema della *velocità* e del *volume* dei Big Data.
Sono stati analizzati i tweet di utenti italiani tra Febbraio e Maggio 2021. È stato applicato un algoritmo di community detection per individuare le comunità pro-vax e no-vax tenendo in considerazione le interazioni reciproche degli utenti. Tali interazioni sono state usate anche per poter stimare l'influenza degli utenti nella piattaforma.

Risultati: Le analisi condotte hanno permesso di osservare:

- Chi sono gli utenti più influenti e come questi sono cambiati nel corso delle settimane;
- Le parole più ricorrenti nei tweet che parlano di vaccini;
- Che esiste effettivamente una polarizzazione: come confermato da diversi studi, la possibilità di accedere ad informazioni su internet senza alcuna mediazione implica la tendenza da parte degli utenti a segregarsi all'interno di gruppi in cui è vengono confermate le proprie convinzioni, escludendo così qualsiasi possibilità di confronto.

Contenuti

1. Introduzione

Nel secondo decennio del XXI secolo, l'affermazione dei social network ha portato ad una vera e propria rivoluzione nel mondo della comunicazione. Ai canali tradizionali, quali tv, radio e giornali, se ne sono affiancati di nuovi, come i social network per l'appunto, e più in generale internet. Si pensi che secondo il rapporto sul consumo di informazione di Agcom del 2018 [1], internet rappresentava la principale fonte di informazione per il 26,3% della popolazione italiana. Nel 2021 si può supporre che questo dato sia rimasto invariato se non addirittura aumentato. Tra i fattori che possono spiegare questa ascesa vi è sicuramente il fatto che i social network rappresentano una vera e propria finestra immediata sul mondo, veicolando le informazioni in maniera di gran lunga più veloce rispetto ai canali tradizionali. Si potrebbe persino affermare che l'informazione non sia più mediata, cioè che giunga all'utente in maniera diretta.

Il web ha quindi stravolto le dinamiche della comunicazione. La struttura stessa della comunicazione è cambiata: da un modello verticale, in cui l'informazione era trasmessa dai media e recepita dal pubblico, si è passati ad uno reticolare e bidirezionale, in cui l'utente finale non si limita a riceverla in maniera quasi passiva, ma può interagire con l'informazione, può condividerla, commentarla, cercarne di altre a sostegno o smentita della prima. Gli algoritmi alla base dei sistemi di raccomandazione possono poi personalizzare l'esperienza che ciascuno intrattiene con l'informazione. I "mi piace", i "retweet", le condivisioni e le interazioni innescano una catena di eventi che fanno in modo che a Mario Rossi siano proposte, con "priorità" maggiore, informazioni diverse rispetto a quelle proposte al suo vicino Roberto Verdi, mentre invece il TG1 è trasmesso in maniera identica nelle case di entrambi, il che porta alla nascita del cosiddetto fenomeno delle "camere dell'eco". Tutti questi fattori si collegano poi al tema delle fake news e della cosiddetta "post-verità", gettando le basi per tutta serie di questioni sotto innumerevoli punti di vista, da quello sociale a quello economico, passando per quello psicologico, politico e tanti altri.

Il progetto nasce dalla consapevolezza della portata del fenomeno appena descritto, così come dalla volontà di volerlo contestualizzare rispetto ad un altro tema, particolarmente sentito nel corso del 2021, ovvero quello dei vaccini contro il virus COVID19. In merito alla questione delle vaccinazioni, e in particolare rispetto a come questo argomento possa polarizzare il dibattito nei social network è stato realizzato un interessante lavoro di Schmidt et al. [2], al quale il progetto qui presentato si ispira per quanto concerne le analisi. In particolare, fissato come perimetro di interesse il social network *Twitter*, l'obiettivo dello studio è quello di rispondere alle seguenti domande di ricerca:

- Chi sono gli utenti italiani più "influenti" in materia di vaccini? In altri termini, quali sono gli utenti che hanno un maggior seguito e una maggiore risonanza all'interno della piattaforma?
- Quali sono le parole più ricorrenti nei tweet degli italiani quando si parla di vaccini? Con che frequenza si parla di morti o effetti collaterali? È più ricorrente l'aspetto sanitario? Oppure quello politico-economico?
- Esiste una polarizzazione del dibattito? Ovvero, è possibile individuare gruppi con idee contrastanti che tendono a isolarsi interagendo soltanto con chi la pensa alla stessa maniera?

Tutti i quesiti sono stati affrontati con granularità settimanale, allo scopo di osservare eventuali cambiamenti nel corso del

tempo e verificare se i fatti ed eventi di cronaca in Italia e nel mondo potessero avere un'influenza sugli utenti più seguiti (i), sul taglio della comunicazione (ii) e sulla polarizzazione della piattaforma (iii). La scelta della granularità settimanale è dettata dal fatto che è stata ritenuta essere quella più adatta nel recepire eventuali fenomeni temporali: una granularità giornaliera sarebbe stata condizionata dal fatto che, rispetto al numero di contagi giornaliero, il sabato e la domenica ne registrano sempre un numero inferiore, il che li rende giorni "non equiparabili" agli altri sotto questo aspetto. Allo stesso modo, la granularità mensile è stata esclusa perché eccessivamente estesa per poter intercettare trend e fenomeni interessanti.
La finestra di tempo analizzata va dal 22 Febbraio 2021 al 23 Maggio 2021. Si tratta di un intervallo arbitrario, ma ritenuto comunque interessante se lo si considera come fase inziale della campagna vaccinale in Italia per la maggior parte della popolazione.

Il contributo originale di questa ricerca è quello di cercare di adattare lo studio [2] al contesto di Twitter anziché Facebook, e quello di provare a fornire una misura quantitativa della polarizzazione osservata.

La struttura del paper è la seguente:
Nella Sezione 2 verrà descritta l'architettura software realizzata per poter raccogliere ed elaborare i dati necessari. Nella Sezione 3 verrà descritta l'origine dei dati raccolti, come sono stati organizzati e modellati. Nella Sezione 4 verranno illustrate le analisi svolte e i principali risultati ottenuti. Infine, nella Sezione 5, alla luce di quanto osservato nella Sezione 4, si proverà a rispondere alle domande di ricerca presentate precedentemente.

2. Architettura

Allo scopo di raccogliere i dati e svolgere le analisi necessarie è stata implementata l'architettura riportata in Figura 1.
Come da vincolo di progetto, la realizzazione ha dovuto preoccuparsi di affrontare due delle tre "V" dei Big Data (Volume, Velocità, Varietà). Sono state affrontate Volume e Velocità.

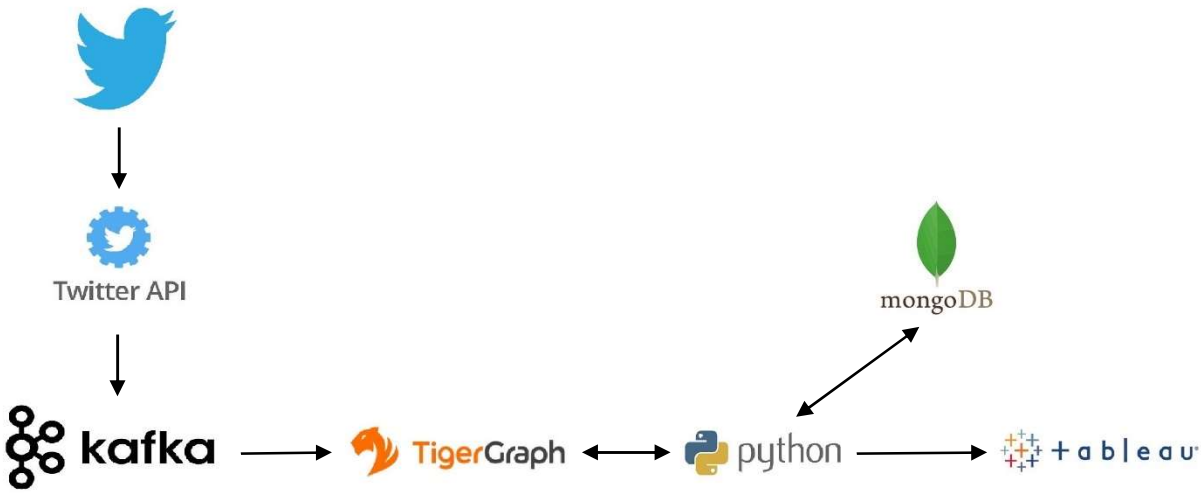

Figura 1. Architettura della pipeline implementata per la raccolta e analisi dei dati.

Twitter API

Per raccogliere i tweet sono state usate le API di *Twitter*. Questo ha consentito una raccolta dei dati in streaming, realizzando così la V della velocità. La velocità è stata realizzata tramite acquisizione dei dati. L'elaborazione è stata fatta solo in un secondo momento, dopo che tutti i dati necessari sono stati raccolti. Questo perché, per gli obiettivi del progetto, non è stato necessario fornire i risultati delle analisi in tempo reale.

Apache Kafka

Allo scopo di disaccoppiare la fase di acquisizione dei tweet con quella di memorizzazione vera e propria, è stato utilizzato *Apache Kafka* [3] come tecnologia per rendere asincrone le due fasi. *Apache Kafka* è una piattaforma per gestire dati in streaming distribuita e organizzata secondo un meccanismo a coda di tipo *publish and subscribe*. Nell'architettura proposta è stato creato un nodo *Kafka* gestito tramite *Apache Zookeeper* [4] e al suo interno è stato creato un topic "tweets". Grazie a questa tecnologia è stato possibile acquisire i tweet in maniera veloce, con un ritardo di poche decine di secondi rispetto a quando sono stati pubblicati, per poi poterli memorizzare tramite *TigerGraph*.

TigerGraph

TigerGraph [5] è un DBMS a grafo distribuito con storage engine nativo. La scelta di utilizzare un modello a grafo è dettata dal fatto che esso ben si presta a modellare reti di relazioni come quelle che si presentano all'interno dei social network. Soprattutto in casi di reti con diversi livelli di profondità, una soluzione relazionale provocherebbe il fenomeno di *join bombing*, con il rischio di incorrere in un decadimento delle prestazioni se non addirittura in un crash del DBMS stesso.

Il linguaggio per interrogare *TigerGraph* è *GSQL* che, come suggerisce il nome, utilizza una sintassi SQL-like. *GSQL* è un *pattern matching query language* di tipo dichiarativo.
L'architettura interna del DBMS *TigerGraph* prevede un *Graph Storage Engine* (GSE) e un *Graph Process Engine* (GPE). In particolare il GPE fornisce una funzionalità di partitioning dei dati in maniera automatica su più nodi, realizzando così lo scaling-out della soluzione al bisogno. Inoltre esso permette di effettuare l'elaborazione dati e utilizzo di algoritmi built-in in maniera veloce ed efficiente. Ciò è possibile grazie al fatto che esso è sviluppato in C++, ed è progettato per supportare la logica *MapReduce*, il che permette di distribuire il calcolo sui vari nodi, per poi aggregare i risultati ottenuti dai diversi task paralleli e ottenere un singolo output.

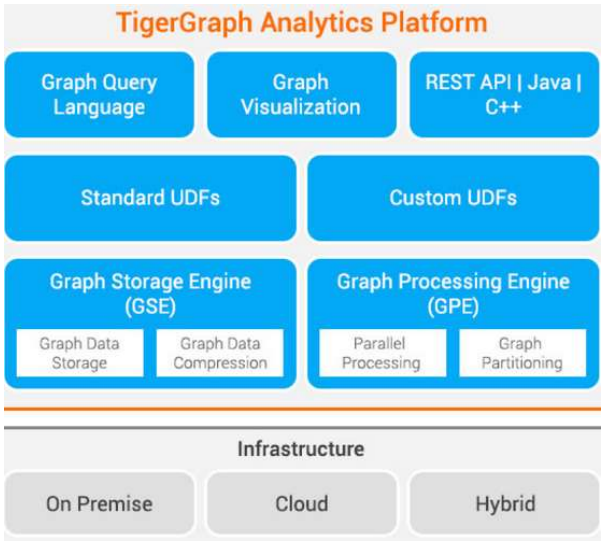

Figura 2. Architettura interna del DBMS TigerGraph

L'unico svantaggio di *TigerGraph* rispetto ad altre soluzioni è che lo schema dei dati non è *free,* ma deve essere definito all'inizio in fase di progettazione.
La parte di configurazione del DBMS e di definizione dello schema dei dati sono state effettuate tramite *GraphStudio*, un'interfaccia grafica che permette di interagire in maniera semplice e intuitiva. L'interazione con i dati (query di inserimento e interrogazione) è stata poi effettuata tramite codice Python. In

particolare per le API è stato utilizzato il design pattern della OOP chiamato "*Adapter Pattern*" (o anche "*Wrapper*") tramite l'implementazione della classe *tigergraph_connection* allo scopo di rendere ancora più facile l'interazione con il DBMS tramite codice.
Nella pipeline è stata utilizzata una versione di *TigerGraph* implementata in cloud su una macchina disponibile tramite il servizio *AWS* e sfruttando:

- 2 nodi *Partition* per la distribuzione orizzontale;
- 2 nodi *Replication* per la duplicazione dei dati.

In questo modo è stata aumentata sia l'affidabilità tramite la replica dei dati, sia le performance aumentando il grado di parallelismo.

È opportuno precisare che la fase di acquisizione ha fatto uso di Tigergraph "centralizzato" poiché l'unica soluzione a poter essere implementata utlizzando un'istanza gratuita di *AWS*. Considerato il tempo di quasi 3 mesi, utilizzare Tigergraph distribuito, che richiede come requisito minimo una macchina *AWS* con un costo orario minimo di circa 1.20$, avrebbe rappresentato un'ingente spesa. Per questa ragione, solo nel momento in cui l'acquisizione dei dati è stata interrotta è stato migrato l'intero database su un una nuova macchina virtuale, distribuito, e utilizzato per il tempo necessario per le analisi (al momento della scrittura di questo paper, vengono regalati 25$ al momento dell'iscrizione). Questo accorgimento, dovuto ad una questione meramente economica, non ha richiesto alcuna modifica dal punto di vista implementativo, in quanto le query vengono eseguite nello stesso modo, sia che si tratti di una soluzione distribuita o meno.

MongoDB

MongoDB [6] è un DBMS documentale, non completamente open-source, che supporta nativamente i file in formato JSON, quindi caratterizzato da strutture annidate e NoSQL like, ossia senza una struttura tabellare tipica dei datawarehouse tradizionali. *MongoDB* è stato scelto come database documentale visto il suo ampio utilizzo in diverse industrie, da *KPMG* a *Microsoft*, passando anche per agenzie governative e banche statunitensi ed inglesi. Vi sono diversi tipi di licenze d'utilizzo, nell'architettura proposta è stata utilizzata la "community server", ad utilizzo gratuito.
Per il progetto è stato utilizzato il metodo "*sharding*" per gestire la scalabilità orizzontale dei dati. Trattasi di un metodo che, attraverso l'aggiunta di nodi, permette al sistema di distribuire sia i dati che il carico applicativo. I sistemi classici, che non sono distribuiti orizzontalmente, infatti, possono mettere alla prova la capacità di un singolo server. Per esempio, un high query rate, ossia una frequenza alta di query, può esaurire la capacità della CPU e della RAM del server.
Lo sharded cluster, ovvero il risultato dello *sharding* consiste nelle seguenti componenti:

- *shards* (o *replica sets*): uno *shard* è un contenitore che memorizza un sottoinsieme dei dati della collection per la quale è stato impostato lo sharding. Ciascuno shard, a propria volta, si compone di uno o più nodi *mongod* che contengono lo stesso sottoinsieme di dati;
- *routers* (o *mongos*): un router è il componente che conosce come i dati sono stati distribuiti tra gli *shard* e di conseguenza mappa le query sia di scrittura e lettura, fornendo un'interfaccia tra le applicazioni client e i cluster sharded;
- *config server*: componente che contiene i metadati e le impostazioni dei cluster sharded.

I documenti sono distribuiti attraverso gli *shards* grazie alle chiavi, o "*shard keys*", e sono organizzati in "*chunks*", ovvero gruppi logici di documenti con limite superiore ed inferiore basato sulla *shard key*. La scalabilità orizzontale avviene proprio grazie alle *keys*, che consentono di specificare l'allocazione dei dati su più nodi. Il sistema utilizzato è stato configurato con:

- due *shard cluster* in replica set, ciascuno con tre nodi *mongod;*
- un *config server* in replica set con tre nodi;
- un *router* per gestire il tutto.

Tale configurazione è stata scelta per gestire al meglio il trade-off fra risorse a disposizione e scalabilità orizzontale.

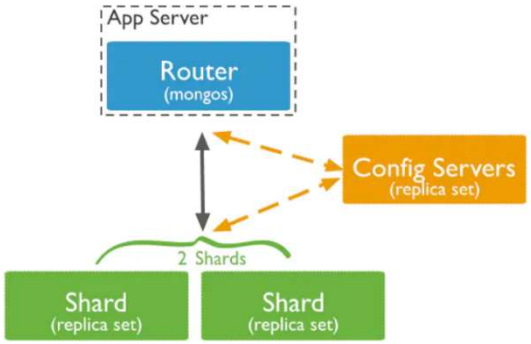

Figura 3. Architettura implementata tramite sharding.

L'utilizzo sia di *TigerGraph* che di *MongoDB*, configurati entrambi in maniera non centralizzata, ha permesso di realizzare la V del volume. Semplicemente andando ad aumentare i nodi è possibile scalare l'architettura per poter gestire dati di dimensioni maggiori.

Nota
È stato fatto un tentativo con *Apache Spark* per scalare anche l'elaborazione dei dati, ma dovendo trattare dati in forma di grafo, il modulo *GraphFrames* di *Spark* si è rivelato essere povero di metodi e funzionalità da utilizzare per le analisi prefissate, in particolare per la fase di clustering su grafo. Per questa ragione è stato scartato. Tuttavia diversi accorgimenti sono stati presi utilizzando array di *numpy* e matrici sparse di *scipy* che memorizzano solo gli archi del grafo effettivamente presenti ed espongono metodi efficienti. In questo modo, pur non essendo soluzioni distribuite di elaborazione, si è voluto comunque dimostrare che è stato considerato il problema del volume e sono state adottate delle misure per limitare l'occupazione di memoria.

3. Dati

I dati utilizzati si riferiscono ai tweet e agli utenti attivi sulla piattaforma nel periodo che va dal 22 Febbraio 2021 al 23 Maggio 2021. Essi sono stati acquisiti in streaming tramite API di *Twitter* in formato JSON. È stato applicato come filtro per acquisire tali tweet il vincolo che questi dovessero contenere la parola "vaccino/i". Nel momento in cui si scarica un tweet, le API forniscono un gran numero di informazioni, sia riguardo il tweet, sia riguardo gli utenti che sono coinvolti nella sua pubblicazione (l'autore, utenti menzionati, utenti che hanno scritto il tweet a cui quello corrente risponde ecc…). Tuttavia si è scelto di memorizzare solo le informazioni ritenute essere effettivamente utili per le analisi previste.

Le informazioni estratte dai tweet sono state memorizzate su *Tigergraph* secondo il modello di dati riportato in Figura 4.

Per il nodo *User* sono stati definiti i seguenti attributi:

- *Id*: identificativo numerico che Twitter attribuisce a ciascun utente;
- *Created*_at: data di creazione del profilo utente. In fase di stesura del progetto si pensava di poter utilizzare questo attributo per identificare eventuali profili *fake* o *bot,* che è risaputo vengono creati

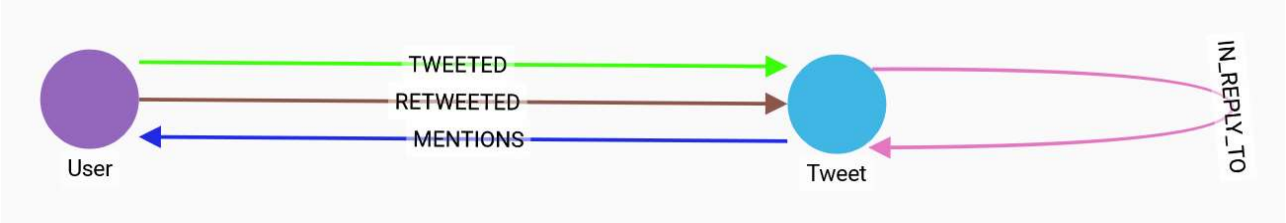

Figura 4. Modello dei dati.

pochi giorni prima di produrre spam o fake news. Tuttavia per mancanza di tempo l'idea è stata abbandonata;

- *Screen_name*: nome dell'utente, consiste in una stringa alfanumerica, preceduta dalla @;
- *Classe*: di default fissato come "NON_INFLUENTE" consiste in un campo booleano calcolato che indica l'influenza dell'utente cui si riferisce ("INFLUENTE" o "NON_INFLUENTE"). La logica con cui viene calcolato, che utilizza diverse metriche basate sulle interazioni e le attività del singolo utente, verrà approfondita nella Sezione 4.

Per il nodo *Tweet* sono stati definiti i seguenti attributi:

- *Id*: identificativo numerico usato da Twitter;
- *Created_at*: data di pubblicazione del tweet, necessario per svolgere analisi temporali;
- *Text*: contenuto del testo del tweet, di tipo stringa.

Per nessuna delle relazioni sono stati definiti degli attributi.

Infine i dati risultanti dalle elaborazioni intermedie sono stati memorizzati su *MongoDB*.
Per quanto riguarda le metriche calcolate per poter stimare l'influenza degli utenti per ciascuna settimana, esse sono state memorizzate in semplici documenti, organizzati nella collection "*metriche*".
Le relazioni tra utenti influenti e non, invece, sono state memorizzate in documenti organizzati nella collection "*links*". Il significato e uso di questi dati verrà approfondito nella Sezione 4.

4. Analisi e risultati

Per polarizzazione si intende la "segregazione" degli utenti in comunità distaccate. In altri termini si parla di polarizzazione nel caso in cui sia possibile osservare gruppi di utenti che presentano numerose interazioni con utenti appartenenti allo stesso gruppo e poche, se non addirittura nulle, con utenti di gruppi diversi. Questa definizione rispecchia quella più tecnica di "*cluster*" nell'ambito dell'apprendimento non supervisionato.

L'obiettivo principale delle analisi è stato esattamente quello di individuare e quantificare, se presente, la polarizzazione su *Twitter*.

Le analisi svolte sono state replicate, in maniera identica dal punto di vista metodologico, per tutte le settimane in esame, dove ogni settimana va dalle 00:00 del Lunedì fino alle 23:59 della Domenica. In questo modo è stato possibile svolgere un'analisi che per ciascun intervallo di tempo considerasse le stesse condizioni (es. sabati e domeniche con minor numero di contagi registrati per via del minor numero di tamponi effettuati), di modo da poter costruire una serie temporale di dati da studiare.

Il procedimento può essere riassunto nei seguenti punti:

1. Classificazione di utenti influenti e non;

2. Costruzione del grafo bipartito che mette in relazione utenti influenti con quelli non influenti;
3. Costruzione del grafo di "proiezione" a partire dal grafo del punto precedente, che considera come nodi soltanto gli utenti influenti;
4. Applicazione dell'algoritmo di Clustering su grafo "*Louvain*" [7] per fare *community detection*;
5. Calcolo della polarizzazione degli utenti non influenti.

Rispetto al lavoro di Schmidt et al. [2], dove la distinzione in utente e pagina (equivalente di "utente influente", tanto che d'ora in avanti i due termini verranno usati come sinonimi) è intrinseca della struttura di *Facebook* stesso, nel caso di *Twitter* è stato necessario introdurre il Punto 1.
Pe risolvere tale problema è stata presa ispirazione dal lavoro di Pal & Counts [8] in cui si individuano delle metriche utili per distinguere utenti influenti da quelli non.
Per poter calcolare tali metriche è stato necessario ricavare le seguenti *features,* per ciascun utente*:*

- **OT1**: numero originale di tweet pubblicati dall'utente;
- **CT1**: numero di tweet conversazionali (in risposta ad altri tweet);
- **CT2**: numero di tweet conversazionali in risposta a tweet pubblicati dall'utente;
- **RT1**: numero di retweet che l'utente ha pubblicato rispetto a tweet scritti da altri utenti;
- **RT2**: numero di tweet dell'utente che sono stati retwittati da altri utenti;
- **RT3**: numero di utenti che hanno retwittato i tweet dell'utente;
- **M1**: numero di menzioni effettuate dall'autore;
- **M2**: numero di utenti menzionati dall'autore;
- **M3**: numero di menzioni dell'autore da parte di altri utenti;
- **M4**: numero di utenti che menzionano l'utente considerato.

Si noti che queste *features* sono state ricavate tramite query al DBMS.
A questo punto sono state calcolate le seguenti *metriche complesse*:

- **TS** (*Topical signal*):
$$\frac{OT1 + CT1 + RT1}{\#TWEETS}$$
stima di quanto l'autore è stato coinvolto nella discussione dell'argomento del suo tweet (vaccini in questo caso);

- **SS** (*Signal strength*):
$$\frac{OT1}{OT1 + RT1}$$
indica quanto è forte il *Topical signal* dell'autore. Può essere intesa come misura dell'"originalità" dei suoi tweet;

- **CS** *(Non-Chat Signal*):
$$\frac{OT1}{OT1 + CT1} + \lambda \frac{CT1 - CT2}{CT1 + 1}$$
quanti tweet sono stati scritti dall'autore sull'argomento e quant'è sviluppata la conversazione con altri utenti. Lambda è stata fissata pari a 0.7;

- **RI** (*Retweet Impact*):
$$RT2 \cdot \log RT3$$
indica l'impatto del contenuto generato dall'autore;

- **MI** (*Mention Impact*):
$$M3 \cdot \log M4 - M1 \cdot \log M2$$

esprime quanto è stato menzionato l'autore prendendo in considerazione l'argomento di interesse.

Successivamente, per ciascuna settimana e per ciascun utente è stata calcolata la media delle *metriche complesse*. Infine, avendo ordinato gli utenti per tale media, sono stati classificati come utenti influenti quelli aventi media superiore al 95-esimo percentile. Tale media è stata poi utilizzata come variabile "*Influenza*" nelle infografiche.
La divisione in utenti influenti e non ha permesso di realizzare il grafo bipartito che mette in relazione utenti influenti con quelli non influenti che hanno retwittato i contenuti dei primi (Punto 2).

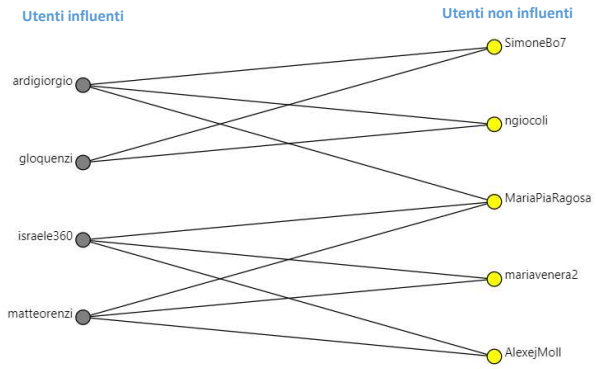

Figura 5. Grafo bipartito. Il primo insieme di nodi è quello degli utenti influenti. Il secondo è quello degli utenti non influenti. L'arco sta a rappresentare il fatto che un utente non influente ha retwittato almeno un tweet di quel determinto utente influente.

Nell'esempio in figura significa che, nella settimana in analisi, *@SimoneBo7* (utente non influente) ha retwittato i tweet sia di *@ardigiorgio* che di *@gloquenzi* (entrambi influenti).

Da questo grafo è stato ottenuto il grafo di "proiezione" (Punto 3), ovvero un grafo completo i cui nodi sono soltanto gli utenti influenti e gli archi pesati con il numero di utenti non influenti che i due nodi collegati condividono nel grafo bipartito. In figura non si rappresentano gli archi di peso 0, ottenendo quindi un grafo non completo. Questo ha permesso di rappresentare la topologia della rete di Twitter sulla base dell'intensità delle interazioni.

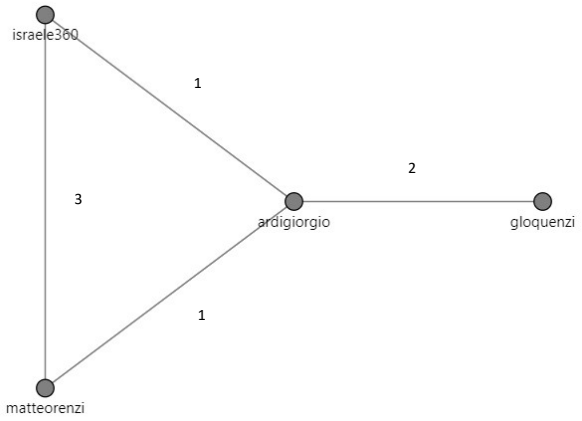

Figura 6. Grafo di proiezione. I nodi rappresentano gli utenti influenti e gli archi sono pesati con il numero di utenti in comune nel grafo di Figura 5.

È stato quindi applicato l'algoritmo di clustering *Louvain* tramite la libreria *scikit-network* [9] [10] implementata in Python (Punto 4).

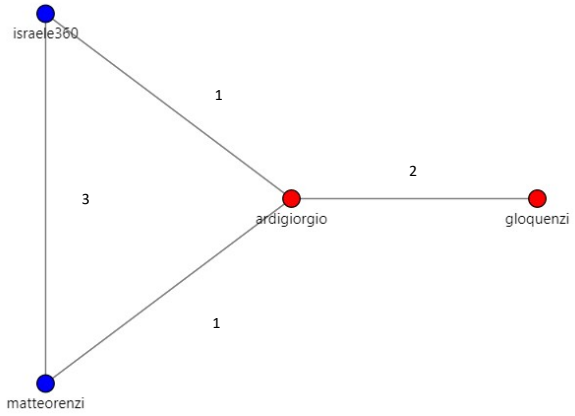

Figura 7. Grafo di proiezione dopo l'applicazione dell'algoritmo di clustering.

I risultati hanno messo in evidenza l'esistenza di due grandi cluster in quasi tutte le settimane. Soltanto nelle settimane x e y sono stati ricavati rispettivamente tre e quattro cluster. Per questi due casi, in seguito a valutazione manuale, sono stati uniti i cluster in modo da potersi rincodurre a una situazione con due soli cluster.
Non disponendo di ulteriori dati che permettessero una validazione supervisionata, la bontà dei risultati è stata

valutata sia utilizzando la modularità [11] sia ispezionando manualmente il contenuto dei cluster. Per ciascun cluster è stato estratto un campione casuale di utenti influenti per i quali sono stati valutati manualmente i relativi tweet e retweet. Quest'analisi ha riscontrato che, pur non essendo perfetti, i cluster ottenuti coincidevano con le due community che ci si aspettava esistessero: pro-vax e no-vax.

A questo punto è stato possibile, per ciascun utente non influente (che rappresentano per ciascuna settimana il 95% della totalità degli utenti) calcolarne la polarizzazione (Punto 5). Preso l'utente u, sia *x* il numero di retweet di tweet i cui autori appartengono alla community pro-vax, e *y* il numero dei retweet (sempre di *u*) rispetto a tweet pubblicati da autori della community no-vax: la polarizzazione di *u* è data da:

$$p(u) = \frac{x - y}{x + y}$$

Questa formula implica che un utente con $p(u)$ vicina a +1 sarà polarizzato verso la community pro-vax, mentre un utente la cui $p(u)$ è vicina a -1 sarà polarizzato verso la community no-vax. Infine un utente la cui $p(u)$ è prossima a zero sarà "neutrale" rispetto al dibattito. Tale calcolo di polarizzazione è stato applicato soltanto agli utenti che avessero fatto almeno 10 retweet nella settimana di interesse.

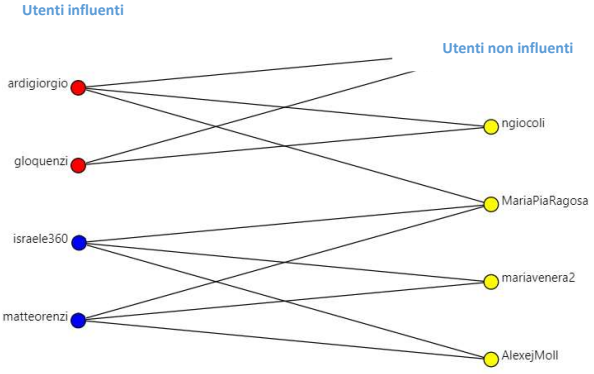

Figura 8. Grafo bipartito dopo aver fatto community detection sugli utenti influenti.

Ad esempio, riprendendo l'esempio (semplificato) dopo il clustering, si ha che l'utente *@MariaPiaRagosa* ha avuto due interazioni con la communiy pro-vax *(@matteorenzi* e *@israele360*) e una con quella no-vax *(@ardigiorgio*)

$$p(@MariaPiaRagosa) = \frac{2 - 1}{2 + 1} = \frac{1}{3}$$

Ne segue che, seppur leggermente, l'utente non influente *@MariaPiaRagosa* sia polarizzata verso la community pro vaccino.

Infine tramite Kernel Density Estimation si è ottenuta una stima della funzione di densità di probabilità (essendo la polarizzazione valore continuo, non discreto). Tramite opportuno test statistico [12] si è rifiutata, in tutte le settimane, l'ipotesi di unimodalità della distribuzione. È possibile osservare la stessa cosa anche graficamente dalle visualizzazioni riportate in Tableau (Twitter polarization and Vaccines | Tableau Public).

Si riporta di seguito soltanto la Funzione di Densità di Probabilità (PDF) riferita alla prima e alla seconda settimana, a fine di chiarimento:

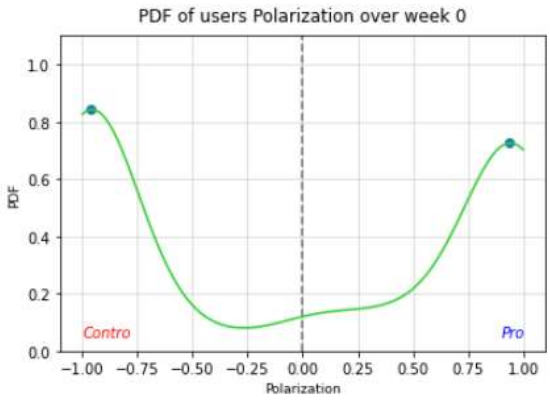

Figura 9. PDF della polarità per la prima settimana di analisi.

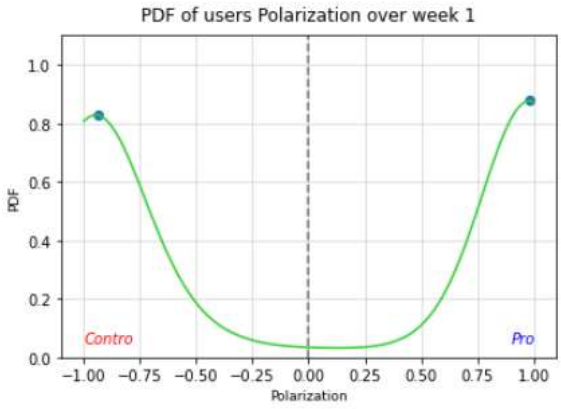

Figura 10. PDF della polarità per la seconda settimana di analisi

In entrambi gli esempi è evidente come la maggiore densità di probabilità (i.e le due mode) si registri per valori di polarità prossimi a -1 e +1, mentre per i valori nel mezzo ci sia una densità di probabilità decisamente inferiore.

Per quanto riguarda la seconda domanda di ricerca, invece, sono stati estratti tutti i termini presenti all'interno di ognuno dei tweet raccolti e si è provveduto a sottoporli ad un processo di normalizzazione dei lemmi: tutte le declinazioni verbali sono state uniformate all'infinito, i plurali uniformati nella forma singolare e sono stati rimossi congiunzioni, avverbi o aggettivi non particolarmente rilevanti o significativi per lo scopo.
Inoltre, per focalizzare meglio la ricerca intorno al tema "vaccini" sono stati consultati alcuni siti web [13] [14] che consentono di ottenere un glossario di tutti i termini correlati ad un argomento specifico.
In aggiunta, è stata effettuata anche una sommaria lettura periodica dei post twittati dagli utenti italiani che contenevano l'hashtag "#vaccino". Grazie a queste due modalità operative, è stata stilata una lista di circa 140 termini che potenzialmente avrebbero potuto ottenere frequenze elevate, per i quali si è provveduto a contare le occorrenze all'interno dell'insieme di termini estratto dai tweet.

5. Conclusioni

Lo studio effettuato ha permesso di rispondere in modo esaustivo alle domande di ricerca poste inizialmente. In particolare, è stato possibile osservare:

- Chi sono gli utenti più influenti e come questi sono cambiati nel corso delle settimane;
- Le parole più ricorrenti nei tweet che parlano di vaccini;
- Che esiste effettivamente una polarizzazione: come confermato da diversi studi, la possibilità di accedere ad informazioni su internet senza alcuna mediazione implica la tendenza da parte degli utenti a segregarsi all'interno di gruppi in cui vengono confermate le proprie convinzioni, escludendo così qualsiasi possibilità di confronto.

Dall'analisi è infatti emerso che:

- Gli utenti con polarità positiva e negativa che hanno avuto più influenza, nel corso delle diverse settimane considerate, sono risultati Roberto Burioni e Claudio Borghi. Per quanto riguarda gli utenti neutrali non è stato possibile individuare un unico utente più influente degli altri, in quanto, come si può vedere dalle classifiche, si sono succeduti sia utenti (come Matteo Salvini, Nino Cartabellotta, ...), che testate giornalistiche (come Repubblica, il Fatto Quotidiano, …)
- Durante le prime settimane considerate, i termini utilizzati più di frequente all'interno dei tweet sono risultati "morte" e "AstraZeneca" , e questo potrebbe essere dovuto alla registrazione dei primi casi di trombosi avvenuti a marzo. Nei mesi successivi, le parole più utilizzate risultano invece

essere “Pfizer”, “AstraZeneca” e “prenotazione”, probabilmente a causa dell’avanzamento della campagna vaccinale e della stessa anche agli over 50 e 40.

- Si è osservato, per tutte le settimane e in maniera più o meno intensa, che la PDF presenta una forte concentrazione di utenti nelle due posizione più estreme (pro e no-vax). Questo dato permette di concludere che la polarizzazione sia particolarmente accentuata, ovvero che la maggior parte degli utenti sia ancorata ad una delle due posizioni e abbia pochissime, se non nulle, interazioni con quella opposta. Inoltre, si può osservare un costante aumento della numerosità della community dei “contro”, fino a raggiungere il picco della densità di probabilità nella 20esima settimana.
  Osservando l’infografica di Tableau a pagina 7, è possibile concludere, che la numerosità della community dei “pro” e dei “contro” risulta, nel complesso, molto simile durante le prime settimane, mentre quella dei “neutrali” resta pressoché bassa fino alla settimana 12, in cui subisce un lieve aumento. Inoltre, a partire da questa settimana in poi è possibile osservare un graduale, ma costante aumento della numerosità della community dei “contro”, fino a raggiungere il picco della densità di probabilità nella 20esima settimana. La community dei “pro”, invece, ha subìto un calo considerevole nel corso delle settimane. Un’osservazione che si potrebbe fare è che questa tendenza si sia verificata durante la settimana immediatamente successiva ai primi di casi di trombosi.

La difficoltà maggiore è stata quella di riuscire a individuare una statistica in grado di riassumere la bimodalità in maniera quantitativa. Nonostante esistano diversi indici in letteratura, e alcuni di essi siano stati calcolati (*bimodality ratio*, *bimodality coefficient*, statistica utilizzata nel dip test) in realtà nessuno è stato ritenuto soddisfacente per questo scopo specifico.
Pertanto si è deciso di limitarsi a valutare la polarizzazione soltanto graficamente. Per analisi grafica più approfondita si rimanda alle visualizzazioni di Tableau.

Per quanto riguarda gli sviluppi futuri si potrebbe pensare di approfondire l’idea di analizzare ed individuare eventuali profili “fake” o “bot” come accennato. Allo stesso tempo si potrebbe pensare di estendere l’orizzonte di analisi su scala mondiale, e non limitarsi al solo contesto italiano.

Riferimenti Bibliografici

[1] AGCOM, «Rapporto sul consumo di infromazione,» 2018.

[2] A. L. Schmidt, F. Zollo, A. Scala, C. Betsch e W. Quattrociocchi, Polarization of the vaccination debate on Facebook, 2018.

[3] «Apache Kafka,» [Online]. Available: https://kafka.apache.org/.

[4] «Apache Zookeeper,» [Online]. Available: https://zookeeper.apache.org.

[5] «TigerGraph,» [Online]. Available: https://docs.tigergraph.com/.

[6] «MongoDB,» [Online]. Available: https://docs.mongodb.com/.

[7] «Louvain Method,» [Online]. Available: https://en.wikipedia.org/wiki/Louvain_method.

[8] A. Pal e S. Counts, Identifying Topical Authorities in Microblogs, 2011.

[9] «scikit-network,» [Online]. Available: https://scikit-network.readthedocs.io/en/latest/index.html.

[10] T. Bonald , N. de Lara , Q. Lutz e B. Charpentier, Scikit-network: Graph Analysis in Python, 2020.

[11] U. Brandes, D. Delling, M. Gaertler, R. G̈orke, M. Hoefer, Z. Nikoloski e D. Wagner, On Modularity Clustering, 2008.

[12] J. A. Hartigan e P. M. Hartigan, The Dip Test of Unimodality, 1985.

[13] «Bologna Today,» [Online]. Available: https://www.bolognatoday.it/attualita/parole-pandemia-glossario-coronavirus.html.

[14] «Ethos,» [Online]. Available: https://www.ethosnet.it/5-settembre-2019-il-processo-della-valutazione-dei-rischi-nelle-cantine-vinicole/.