Università Politecnica delle Marche

Facoltà di Ingegneria

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

---

Tesi di Laurea Magistrale

Un approccio basato sulla Social Network Analysis per lo studio del ciclo di vita delle challenge di TikTok

A Social Network Analysis based approach to study the lifespan of TikTok challenges

Relatore	Candidato
Prof. Domenico Ursino	Lorenzo Giuliani

Anno Accademico 2020-2021

Indice

Introduzione

I social network

Negli ultimi 15 anni, con l'avvento del cosiddetto "*Web 2.0*", ed il propagarsi di una migliore connettività alla rete Internet in ogni angolo del pianeta, sono venute alla luce alcune piattaforme virtuali, oggi note come "*social network*", in grado di mettere in comunicazione gli utilizzatori, e permettere loro di interagire tramite diverse modalità, sia testuali che multimediali. I social network, con il passare degli anni si sono sensibilmente evoluti; oggi tendono sempre più a configurarsi come degli aggregatori di informazioni, accogliendo al loro interno tutti i contributi, in diversa natura, resi liberamente disponibili da un enorme numero di utenti.

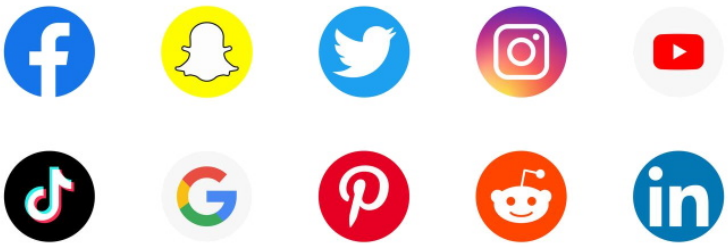

Figura 1: Loghi di alcuni dei social network più utilizzati. *Fonte: dreamstime.com*

Il social network, quindi, diventa un vero e proprio specchio della società reale, dove i milioni di utenti giornalieri tendono a creare delle "comunità", sulla base di interessi e passioni comuni, che amplificano le possibilità relazionali del singolo individuo ed alimentano la formazione di piattaforme di conoscenze condivise, in grado di agevolare e potenziare la ricerca di informazioni, una delle principali funzioni della rete.

L'analisi delle dinamiche sociali che avvengono all'interno di una piattaforma di social network apre un ampio ventaglio di possibilità, e semplifica molti studi che, per mancanza di dati e difficoltà nel reperirli, non sarebbero sostenibili nel mondo reale. Nel corso degli anni, infatti, sono stati pubblicati molteplici *paper*, negli ambiti più disparati, che trattano le reti sociali, evidenziando le loro caratteristiche, ed elencandone pregi e pericoli.

La ricerca

La presente tesi si colloca in tale contesto culturale. Essa, infatti, prende come riferimento la piattaforma TikTok, un social network che, negli ultimi anni ha ottenuto un successo prorompente, contando, in data odierna, più di 700 milioni di utenti attivi, con oltre 2 miliardi di download totali. In esso, gli utenti iscritti caricano filmati di breve durata, spesso imitando contenuti già presenti sulla piattaforma, dai quali prendono ispirazione; in questo modo nascono i trend, serie di video accomunati da un'unica tematica, realizzati anche in modo molto simile tra di loro. La ricerca descritta nella presente tesi propone una nuova metodologia che è stata adottata per studiare da un punto di vista analitico e quantitativo, tali eventi. Il nuovo metodo distingue i trend *positivi*, ovvero quelli composti da video che generano effetti positivi nei propri autori, o in chi passivamente vede e ne reagisce ai contenuti, dai *trend negativi*, o *pericolosi*, cioè quelli che, al contrario, propongono contenuti con azioni che possono danneggiare chi le esegue e di cui ne è sconsigliata l'imitazione, o più semplicemente, contenuti moralmente discutibili, poco adatti ai più piccoli.

L'obiettivo di questo lavoro consiste nella ricerca di *pattern*, ovvero sequenze ben definite, e ripetute, che distinguano univocamente l'evoluzione nel tempo delle due categorie di trend precedentemente elencate. In particolare, viene applicata la Social Network Analysis per cercare, se presenti, differenze strutturali tra le reti che descrivono le interazioni tra gli utenti della piattaforma che hanno partecipato ad alcuni trend di riferimento, accuratamente selezionati a priori. La ricerca segue un approccio iterativo: ad ogni passo si analizzano i risultati ottenuti in precedenza, al fine di generare risultati utili a nuove considerazioni. La tesi è strutturata in modo tale da evidenziare tutti i passaggi dell'intero lavoro svolto, di cui ogni capitolo rappresenta una fase.

Nel complesso, la tesi è strutturata come di seguito specificato:

- Il primo capitolo fornisce al lettore una panoramica sull'ambito di ricerca, descrivendo TikTok, il social network di interesse, ed il suo funzionamento. Successivamente vengono mostrati alcuni dati atti a dimostrare quanto la piattaforma sia efficace e pervasiva, motivo per cui, al giorno d'oggi, è uno dei social media più utilizzati al mondo. In conclusione, viene definito che cos'è una "*challenge*", ovvero il concetto di riferimento di tutto il lavoro.
- Il secondo capitolo si apre con una digressione storica sulla Social Network Analysis e la teoria dei grafi. Dal momento che tutta la ricerca si basa su tali concetti, vengono anche fornite al lettore alcune informazioni di base sugli oggetti matematici che saranno trattati a seguire, dalle strutture dati alle metriche di valutazione dei risultati.
- Il terzo capitolo sancisce l'inizio della ricerca. Qui sono trattate le tecnologie utilizzate durante il progetto e, successivamente, vengono descritte l'architettura software e le soluzioni adottate per l'acquisizione, la trasformazione, e la costruzione del dataset iniziale, nonché le difficoltà affrontate durante le prime fasi.
- Nel quarto capitolo sono descritti i modelli utilizzati per la rappresentazione dei dati, legati alla teoria dei grafi, e la loro semantica. In questa fase sono mostrati ed analizzati tutti i grafi generati a partire dal dataset di cui prima.
- Nel quinto capitolo si analizza la dinamica nel tempo delle reti, passando da un'analisi statica ad una dinamica; si definisce l'algoritmo di segmentazione del *lifespan* delle challenge, e si individuano le caratteristiche principali che le caratterizzano.

- Nel sesto capitolo si categorizzano e si raggruppano gli intervalli precedentemente identificati, sulla base delle loro caratteristiche, e si cercano, se presenti, i *pattern* desiderati, raggiungendo, in questo modo, l'obiettivo della ricerca.
- L'ultimo capitolo, infine, analizza la letteratura correlata al progetto, elencando alcune delle ricerche condotte negli ultimi anni, in diversi ambiti, per quanto riguarda la piattaforma TikTok.

In conclusione, viene dedicato uno spazio alla discussione dei risultati, con uno sguardo ad eventuali sviluppi futuri che possono esser promossi a partire dai risultati ottenuti fino a questo momento.

1. Introduzione a TikTok

1.1 Il Social Network TikTok

TikTok (Figura 1.1), noto in Cina come *Douyin*, è un social network lanciato nel settembre 2016 basato sulla condivisione di video divertenti ed istantanei. In quanto social network, TikTok è fruibile sia in maniera limitata mediante browser, sia, soprattutto, da applicazione mobile su dispositivi *Android* ed *iOS*. TikTok si appoggia sulla relativa piattaforma finalizzata alla gestione dei rapporti sociali, permettendo la comunicazione e condivisione per mezzi testuali e multimediali tra gli utenti iscritti e favorendo la nascita di comunità di persone che condividono interessi.

Figura 1.1: Logo dell'applicazione TikTok. *Fonte: TikTok.com*

TikTok riprende ed estende le funzionalità della dismessa applicazione *Musical.ly*, lanciata nel 2014 dagli stessi autori di TikTok Alex Zhu e Luyu Yang, e con la quale si è fusa nell'agosto 2018 a seguito di un aggiornamento ad opera dell'azienda cinese *ByteDance* che l'aveva acquistata nel 2017. Questa applicazione permetteva agli utilizzatori di realizzare video a tema musicale della durata massima di 15 secondi; gli utenti potevano produrre video in *lip sync* e/o danzando sulle note di una canzone a loro scelta e modificare il risultato, prima della pubblicazione, alterando la velocità di riproduzione o aggiungendo un numero limitato di filtri e/o effetti speciali.

A differenza di *Musical.ly*, in TikTok vi è una maggior eterogeneità di contenuti; qui vi sono proposti sia video a tema musicale, ove gli utenti si esibiscono in performance di *lip sync*, di canto o di ballo, che video di tutt'altra tematica e genere, permettendo di adattare audio e suoni divenuti famosi sia sullo stesso TikTok che su altre piattaforme, e/o di registrare audio inediti ed originali. Considerato che uno degli obiettivi pricipali di questo social network è quello di stimolare la creatività degli utenti, TikTok mette a disposizione un grande quantitativo di funzionalità accattivanti, quali filtri, effetti di bellezza e sticker, da applicare tramite un tool di editing, che consentono di personalizzare ulteriormente il risultato prodotto. Inoltre, mentre su *Musical.ly* la durata del video era limitata ai 15 secondi, TikTok ammette durate ben superiori (con gli ultimi aggiornamenti sono permessi video di 180 secondi), consentendo all'utenza di esprimere anche concetti più articolati o mettere in scena rappresentazioni più elaborate.

All'interno dell'applicazione i video sono presentati con risoluzione *Full HD* ed aspect ratio 9:16 (1080×1920 pixel); si tratta, quindi, di video allungati che si adattano alla visualizzazione full-screen in verticale degli smartphone (Figura 1.2). Questa configurazione sfrutta al massimo l'ergonomicità del dispositivo mobile, rendendo possibile l'utilizzo dell'applicazione con una sola mano, grazie anche alla ormai consolidata *gesture* "*swipe-up*", che consiste nello scorrimento del dito di una mano dalla parte bassa di un display touch-screen a quella alta, e che nel caso dell'applicazione in questione consente di passare dalla visualizzazione di un contenuto a quella di un contenuto successivo. I video vengono prodotti automaticamente dalla piattaforma; per questo si dice che l'esperienza di visualizzazione è passiva.

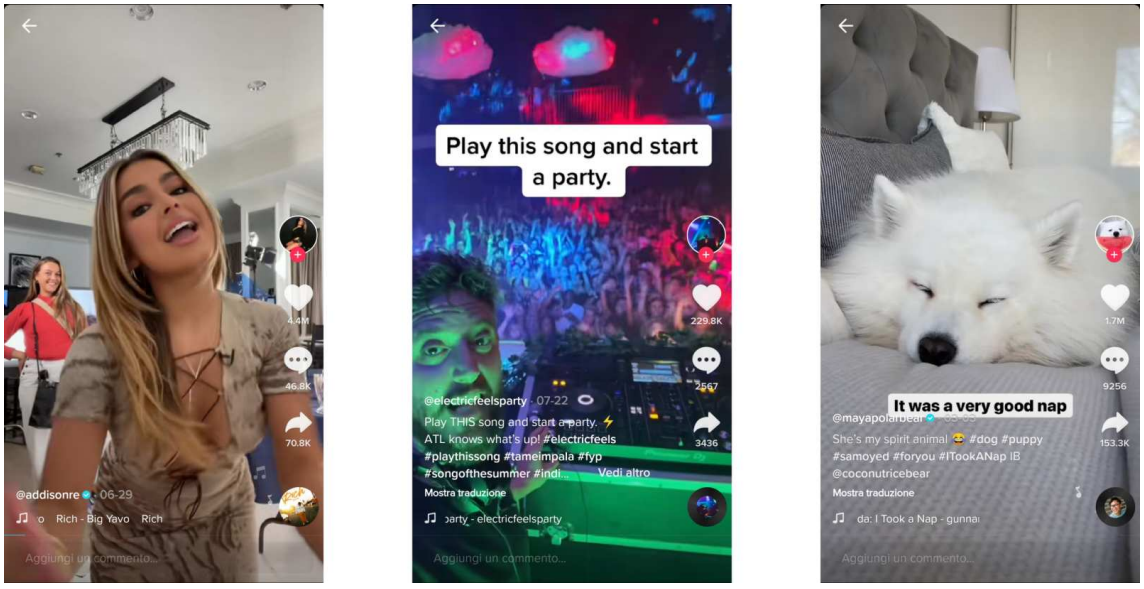

Figura 1.2: Interfaccia dell'applicazione. *Fonte: Autoprodotta*

Queste caratteristiche hanno permesso a TikTok di raccogliere immediato consenso e diventare in tempi molto rapidi la principale applicazione di social media al mondo, relativamente a diverse categorie. Entro poco meno di un anno dal lancio della versione occidentale, TikTok è divenuta l'applicazione *iOS* più scaricata ed utilizzata al mondo. Con il chiudersi dell'anno 2017, infatti, l'applicazione era già stata scaricata circa 130 milioni di volte (Figura 1.3), per poi acquisire sempre maggior popolarità, registrando un tasso di crescita pressoché esponenziale. Ad agosto 2019, considerando sia i dati forniti dall'*Apple Store* che quelli relativi al *Google Play Store*, TikTok ha superato la soglia di un miliardo di download in tutto il mondo. Attualmente la piattaforma conta circa 700 milioni di utenti attivi, 5.4 milioni dei quali provengono dall'Italia, con oltre 2 miliardi di download totali.

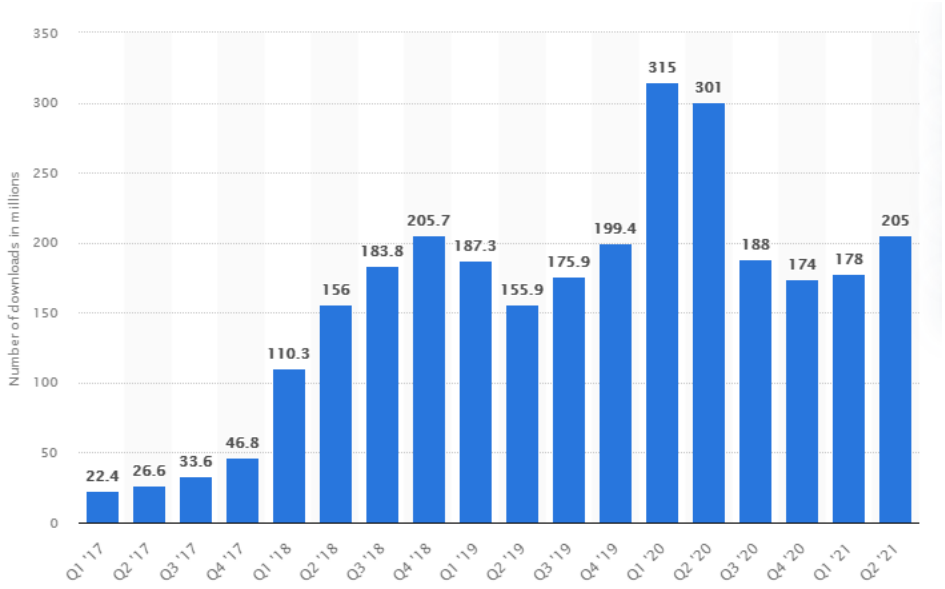

Figura 1.3: Numero di download di TikTok, suddivisi per trimestre. *Fonte: Statista.com*

1.2 Funzionamento di TikTok

Per poter usufruire dei servizi offerti dalla piattaforma TikTok è necessario, come prima cosa, registrarsi, scegliendo un nome utente univoco con il quale identificarsi all'interno della piattaforma stessa. L'applicazione è organizzata a sezioni accessibili mediante un menù orizzontale posto nella parte inferiore dell'interfaccia; una volta effettuato il login, l'utilizzatore è reindirizzato alla sezione "Home" (altresì nota come *"For You Page"*), ovvero una raccolta di video consigliati all'utente da un motore di Intelligenza Artificiale interno alla piattaforma sulla base degli interessi, delle preferenze e dei comportamenti di quest'ultimo. Questa prima sezione si presenta come uno scroll infinito, dove i differenti contenuti sono riprodotti in maniera passiva ed in successione dalla piattaforma.

L'utente può interagire lasciando "mi piace", toccando in rapida successione due volte lo schermo, commentando il video con una propria opinione od un pensiero, oppure condividendo il video stesso sia con utenti nella piattaforma che verso servizi esterni. Qualora l'autore del contenuto fruito abbia abilitato i relativi permessi, egli può opzionalmente interagire con il video creando, a sua volta, un nuovo contenuto multimediale di tipologia "reazione", oppure "duetto". Nel primo caso il nuovo video si apre riproponendo parzialmente il video originale al quale viene legato il nuovo video; nel secondo caso sia il video originale che quello derivato vengono mostrati appaiati uno accanto all'altro.

La sezione "Discover" ("Scopri" in lingua italiana) elenca i contenuti più in voga del momento, mostrando la miniatura di diversi video che hanno registrato un elevato numero di interazioni, separandoli in base ad un *hashtag* caratterizzante, ovvero una parola chiave che rappresenta il loro contenuto, oppure in base all'audio utilizzato. Qui è anche possibile effettuare una ricerca all'interno della piattaforma, specificando una o più parole chiave ed il filtro da utilizzare per la restituzione dei risultati; è, infatti, possibile cercare specifici utenti in base allo *username*, a determinati *hashtag*, a video, a suoni, con la possibilità di specificarne la data di pubblicazione.

Le altre due sezioni "Inbox" ("In arrivo") e "Me" ("Profilo") sono dedicate alla gestione social della piattaforma; la prima raccoglie tutte le notifiche ricevute e fornisce il servizio di messaggistica istantanea; la seconda rappresenta la pagina personale dell'utente, aggregando i video caricati, i video piaciuti e permettendo di modificare le impostazioni del profilo e dell'applicazione stessa.

Contenuti originali possono essere prodotti a partire da un *tap* sull'elemento centrale del menù principale. All'apertura dell'utilità video editor (Figura 1.4) si avvia la fotocamera dello smartphone e vengono proposte delle impostazioni principali di registrazione, oltre che degli effetti applicabili a priori, alcuni dei quali distorcono e modificano sensibilmente le immagini generate, grazie all'impiego della realtà aumentata. L'applicazione mobile TikTok permette, inoltre, di accelerare, rallentare, o modificare, mediante un ulteriore filtro, il suono o la musica di sottofondo selezionabile all'interno di una vasta gamma di generi musicali.

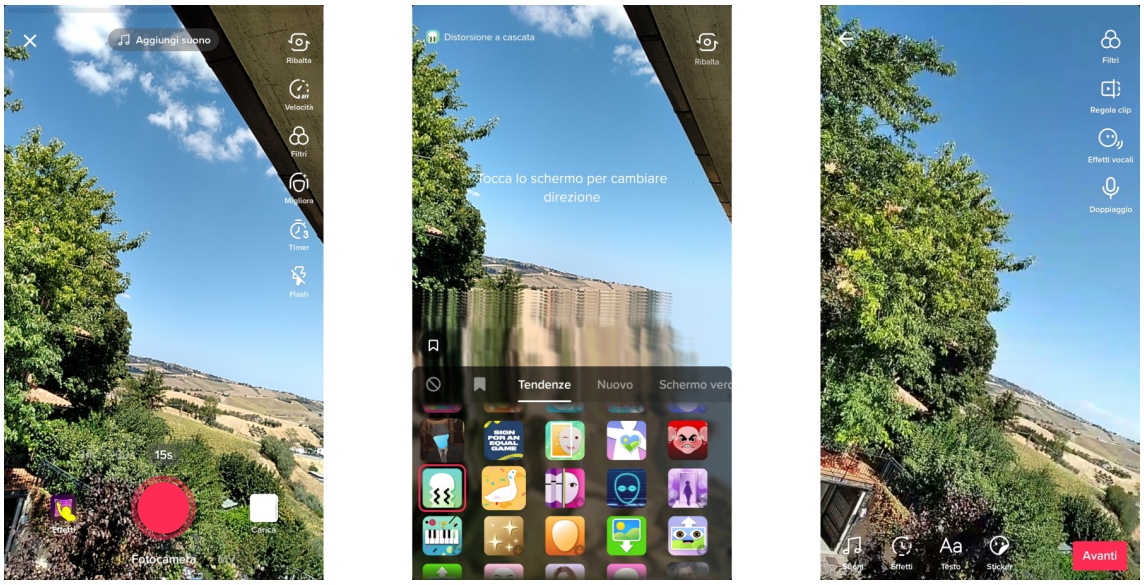

Figura 1.4: TikTok video editor. *Fonte: autoprodotta*

L'applicazione consente agli utenti di configurare i propri account come "privati". Il contenuto di questi account rimane a disposizione di TikTok, ma è visualizzabile soltanto dagli utenti autorizzati dal titolare, che, similmente ad altri social network, possono scegliere se rendere il profilo pubblico o privato, ovvero se interagire soltanto con gli amici, tramite commenti, messaggi o video di "reazione" e "duetti".

1.3 Il target della piattaforma

Date le sue caratteristiche, TikTok è diventata da subito molto popolare sulla fascia di popolazione dei giovani e, soprattutto, dei giovanissimi, appartenenti alla cosiddetta "generazione Z", che identifica i nati tra il 1996 ed il 2009.

Questi rappresentano il target ideale dell'applicazione poiché abituati fin dall'infanzia ad un'esperienza utente simile a quella fornita da TikTok; sono, infatti, cresciuti, a differenza della popolazione meno giovane, con l'avvento delle tecnologie a schermo capacitivo, quali smartphone e tablet, i quali hanno ridisegnato l'esperienza utente classica in qualcosa di molto più interattivo e visivo. Oltre a questo, come già visto in precedenza, TikTok stimola la creatività dei propri utenti più giovani offrendo loro un posto dove esprimersi in ciò che sanno fare meglio. Nel febbraio 2020 gli utenti statunitensi compresi nella fascia di età 10-29 rappresentavano il 62% dell'intera *userbase* della piattaforma (fonte Statista.com, 2020).

Tuttavia, con il passare del tempo, TikTok ha registrato un numero sempre crescente di utenti nella fascia di età meno giovane, un aumento che si è consolidato per tutto l'anno 2020 (Figura 1.5), incrementando, quindi, l'età media generale degli utilizzatori del social network. In questo periodo si sono iscritti prevalentemente utenti con fascia d'età compresa tra 31 e 35 anni, ciò è indice del

fatto che TikTok sta iniziando a coprire anche una diversa fascia di utenti rispetto ai primi anni di funzionamento.

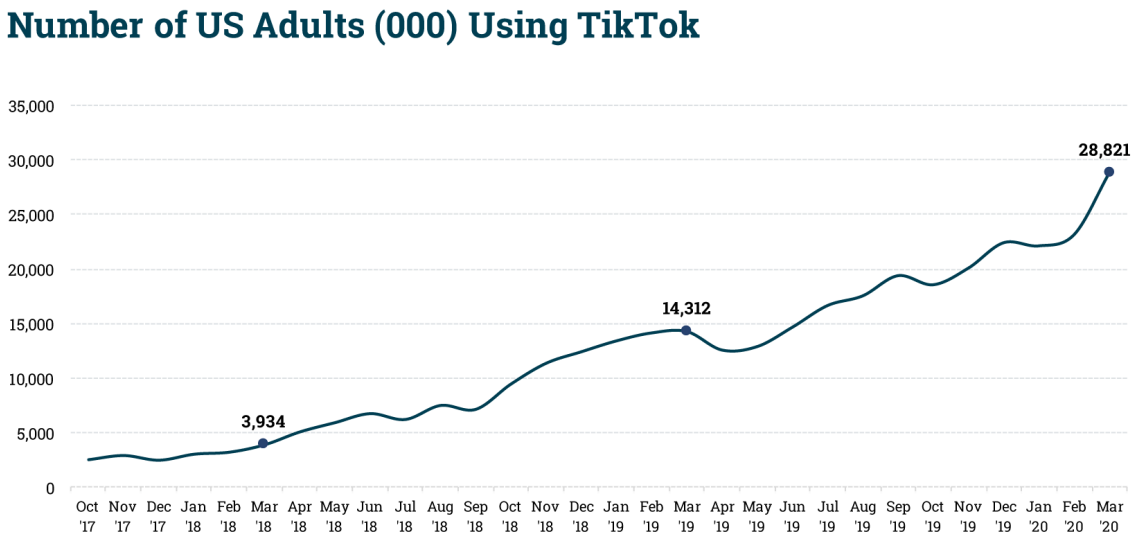

Figura 1.5: Incremento di utenti statunitensi di età avanzata. *Fonte: MarketingCharts.com*

Infatti, nel marzo 2021, gli utenti statunitensi compresi nella fascia di età 10-29 rappresentavano solo il 47.4% degli utilizzatori statunitensi totali, vedendo una diminuzione del 15% circa rispetto all'anno precedente (figura 1.6). I giovanissimi continuano a rappresentare, in ogni caso, la fascia di popolazione più attiva, pari al 25% del totale.

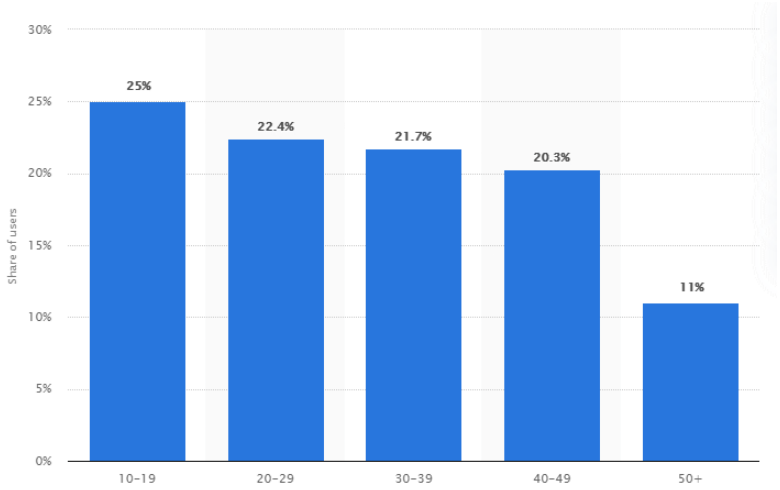

Figura 1.6: Utenti di TikTok per fascia di età, marzo 2021 - USA. *Fonte: Statista.com*

1.4 TikTok: i trend e le challenge

All'interno della piattaforma TikTok è possibile individuare specifici insiemi di video, pubblicati da un grande numero di utenti distinti, che definiscono un *trend* o una *challenge*. Un trend virale non è altro che una sequenza di video legati da una tematica e/o un suono comune riproposta con diverse interpretazioni da un elevato numero di utenti in un certo intervallo di tempo. I trend nascono spesso involontariamente, per divertimento, degli stessi utenti che, dopo aver interagito con un video

originale, ne ripropongono lo stesso contenuto semplicemente imitando le azioni dell'autore, e così generando diverse catene di video affini tra loro. I trend sono facilmente identificabili mediante uno o più *hashtag* inclusi nella descrizione del video, ovvero parole chiave anticipate dal carattere "#", oppure attraverso un suono caratterizzante, tipico dei trend a tema musicale.

Un esempio di trend virale è stato "*#TimeWarpScan*", popolare nel 2020. Questo *hashtag* identifica una serie di video in cui gli autori utilizzano in maniera artistica l'omonimo filtro, caratterizzato da una banda che scorre gradualmente sullo schermo durante l'acquisizione del filmato bloccando, al suo passaggio, il colore dei pixel in corrispondenza alla sua posizione generando, così, una composizione. Un altro trend è stato "*#SavageLove*", in cui tutti gli utenti si esibivano in una particolare coreaografia sulle note di Savage Love, canzone pubblicata nel 2020 dall'artista Jason Derulo.

Una challenge, invece, è un particolare trend che consiste in una sfida virale a cui gli utenti di TikTok son chiamati a partecipare. Nei relativi video gli autori mostrano il loro tentativo di portare a termine il determinato obiettivo. A differenza dei trend, le challenge sono tipicamente più strutturate e specificano maggiormente nel dettaglio quali devono essere i contenuti del video; esse nascono proprio con l'obiettivo di diventare virali, in molti casi in qualità di promozione pubblicitaria.

Essendo, in molti casi, complicato distinguere i contenuti di un trend da quelli di una challenge, data la loro labile differenza, ed essendo, in molti casi, il concetto di trend la generalizzazione del concetto di challenge, con il proseguire della trattazione di questa tesi i due concetti verranno uniti ed i due termini verranno considerati sinonimi. Si indicherà quindi con challenge la partecipazione di un utente di TikTok ad un evento virale, indipendentemente dal fatto che questo richieda il raggiungimento di un risultato o meno.

Secondo una ricerca svolta da Johannes Ahlse, Felix Nilsson, Nina Sandström, riportata nell'articolo *"It's time to TikTok"*, ci sono molteplici motivazioni che spingono un utente a partecipare ad una challenge. Tutti questi aspetti vengono posti all'interno di una piramide con un approccio *top-down*. La figura deve, quindi, essere letta dall'alto verso il basso, ovvero a partire dall'aspetto maggiormente motivante a quello meno importante (Figura 1.7).

Figura 1.7: Motivazioni principali alla partecipazione ad una challenge *Fonte: Jonkoping University*

I motivi principali sono sette:

1. *Entertainment*: il divertimento di cui si può giovare nel cercare di raggiungere un obiettivo definito.

2. *Personal identity*: la possibilità di esibire la propria personalità e valori alla community degli utenti di TikTok.
3. *Socializing*: la possibilità di competere nel raggiungimento dell'obiettivo della challenge con i propri amici.
4. *Status*: poiché le challenge sono spesso avviate da ragazzi molto giovani, che sono attualmente la fetta più grande della *userbase* di TikTok, i loro coetanei non si sentono fuori luogo nel competere.
5. *Convenience*: più una challenge è semplice nell'essere portata a termine più un ragazzo è interessato a parteciparvi.
6. *Structure*: la struttura di una particolare challenge piaciuta ad un utente, in termini di musiche utilizzate e azioni da compiere, può invitarlo a parteciparvi.
7. *Information seeking*: molti giovani partecipano ad una challenge con l'obiettivo di rimanere aggiornati su cosa vada di moda sulla piattaforma o meno.

Si noti come la ricerca di informazioni ("information seeking"), sia l'aspetto meno motivante alla partecipazione di una challenge. Tra gli aspetti più rilevanti c'è il divertimento e la volontà di socializzazione, elementi alla base della cultura di un teenager.

1.4.1 Quando le challenge diventano pericolose

Come si è visto, la partecipazione ad una challenge è una forma di svago a disposizione degli utenti di TikTok e può avere dei ricontri positivi negli stessi partecipanti. Nel corso della storia di TikTok, tuttavia, sono diventate virali alcune challenge il cui contenuto poteva diventare pericoloso per l'utente che decideva di emularne le azioni. Un esempio è la "*#CornCobChallenge*", che consisteva nel rimuovere con i propri denti i semi di una pannocchia di mais mentre questa roteava velocemente alimentata da un motore elettrico, tipicamente un trapano. Un'altra challenge pericolosa è stata la "*#CerealChallenge*", ove i partecipanti riempivano il più possibile la propria bocca con alimenti quali fiocchi d'avena e cereali per colazione, con il rischio di rimanere soffocati.

In alcuni casi degli sfortunati tentativi di partecipazione ad una challenge, come la nota "*#BenadrylChallenge*", consistente nel filmare la propria reazione all'assunzione dell'omonimo farmaco senza prescrizione, sono terminati con il decesso del soggetto partecipante.

Nell'ultimo periodo, TikTok ha potenziato i controlli relativi alla verifica di conformità dei contenuti pubblicati, bloccando e censurando con maggior celerità gli eventi virali potenzialmente pericolosi; per questo motivo, attualmente, le challenge sono prevalentemente innocue. Ad oggi, se un video pubblicato mostra comportamenti potenzialmente pericolosi ma rispettosi delle politiche della piattaforma, come, ad esempio, azioni svolte da/con la supervisione di un professionista, l'applicazione mostra, nella parte inferiore dello schermo, un banner specificante il fatto che il contenuto riporti "comportamenti pericolosi", scoraggiandone l'emulazione dei più giovani.

1.5 Altre problematiche

Oltre ai problemi derivanti dalla partecipazione ad una challenge pericolosa, è necessario far riferimento anche ad altre problematiche che possono insorgere con l'utilizzo della piattaforma. Alcuni svantaggi o possibili problematiche dell'utilizzo del social network sono i seguenti:

- *Accounts fake*: c'è la possibilità che qualche malintenzionato crei account fasulli, che non corrispondano alla reale identità della persona stessa. Questo può mettere in pericolo gli utenti che entrano in contatto con questo tipo di account, in particolar modo se si tratta di adolescenti e, soprattutto, bambini.
- *Abuso di utilizzo e dipendenza*: questo social è stato pensato per catturare l'attenzione completa dell'utente, grazie ai video full screen ed alla navigazione passiva; per questo motivo è importante avere il pieno controllo sull'utilizzo della piattaforma. Il rischio che si incorra in

una forma di dipendenza è molto alto, anche a causa dell'utilizzo dell'intelligenza artificiale che propone contenuti di forte interesse per l'utente, il quale, incuriosito dal prossimo video che verrà proposto, avrà difficoltà nello smettere di usufruire della piattaforma.

- *Contenuti offensivi e calo dell'autostima*: uno dei problemi principali che si è riscontrato nei giovani è la crescente insicurezza e l'innalzamento dei numeri relativamente alla depressione e ai suicidi. Molto spesso si cercano di inseguire canoni estetici irraggiungibili o non realistici che vengono proposti sui social, e questo contribuisce al calo dell'autostima personale degli adolescenti. In particolare, in questa piattaforma specifica, c'è un inasprimento di odio e contenuti offensivi, che vanno ad acuire le problematiche sociali già presenti.

2. Social Network Analysis

2.1 Social Network Analysis

Per Social Network Analysis si intende una metodologia di analisi delle relazioni sociali la cui origine è da attribuire agli studi effettuati dallo psichiatra rumeno Jacob Levi Moreno (1889-1974) sulla "sociometria", la scienza sociale che studia i metodi di rilevazione e misurazione delle relazioni che intercorrono all'interno di una comunità, e dallo psicologo austriaco Fritz Heider (1896-1988) sulla "teoria dell'equilibrio cognitivo", che spiega come le relazioni di "sentimento", o di gradimento tra individui, sono equilibrate se la valenza affettiva (di segno positivo o negativo) all'interno di un sistema si moltiplica in un risultato positivo. Quando, per qualche motivo, l'equilibrio nel sistema viene a mancare, le persone cercano immediatamente di ripristinare una condizione di coerenza, ossia uno stato di equilibrio.

Le loro intuizioni furono successivamente riprese dagli statunitensi Frank Harary (1921-2005), matematico, e Dorwin Cartwright (1915-2008), psicologo, i quali formalizzarono, tramite la teoria dei grafi, un potente metodo per lo studio delle strutture e dinamiche sociali. Ancora successivamente, si introdusse l'uso dei sociogrammi, una metodologia di indagine usata ancora oggi nelle scienze dell'educazione che, tramite l'uso di questionari, si pone l'obiettivo di ricostruire le posizioni degli individui all'interno di un gruppo, per la valutazione delle proprietà informali dei grafi.

Un fondamentale sviluppo alla Social Network Analysis venne dato da due ricercatori del dipartimento di antropologia sociale della Manchester University (John Barnes e Siegfried Nadel), i quali focalizzarono l'attenzione sulle relazioni date dal potere e dai conflitti tra gli individui, piuttosto che su quelle relazioni date dalle configurazioni sociali pre-imposte, come, ad esempio, potevano esserlo i rapporti gerarchici, mostrando come le relazioni visibili in una rete sociale possono essere molto diverse da quelle reali. Oltre a ciò, questi ricercatori diedero anche la definizione di "cricca".

Gli studi sulla Social Network Analysis furono, poi, portati avanti dal sociologo britannico J. Clyde Mitchell (1918-1995), il quale introdusse la differenza tra rete "completa" e rete ego-centrica, oltre all'utilizzo degli indici per l'analisi. In tempi più recenti, gli studi di Harrison Colyar White sostennero che la ricerca di proprietà del network non andava fatta basandosi su categorie

prestabilite, ma sulle reali proprietà della rete e, in questo contesto, il sociologo Mark Granovetter dimostrò l'importanza dei legami deboli (anche noti come "*weak ties*") all'interno dei network.

Tra i vari tipi di analisi, una teoria originale conosciuta come "teoria del mondo piccolo", proposta da Stanley Milgram (1933-1984), si è rivelata ricca di interessanti spunti di ricerca; essa afferma che, in una rete sociale qualunque, il percorso di relazioni più breve necessario per collegare tra di loro due individui è, solitamente, molto piccolo rispetto alla dimensione della rete. Questa definizione viene spesso associata al concetto *six degrees of separation* (anche se l'autore non la citò mai direttamente), che fissa il numero massimo di passaggi a sei. Un'altra teoria interessante è quella legata alla definizione del numero di Dunbar, introdotto dall'antropologo Robin Dunbar (1947). Questo è una quantificazione numerica del limite cognitivo teorico relativo al numero di persone con cui un individuo è in grado di mantenere relazioni sociali stabili, ossia relazioni nelle quali egli conosce l'identità di ciascuna persona e come queste persone si relazionano con ciascuna delle altre. Le stime sul valore del numero di Dunbar oscillano tra 100 e 250, ma l'approssimazione adoperata solitamente è 150. Tale numero rafforza, quindi, il concetto di "mondo piccolo", deducibile dalla omonima teoria precedentemente menzionata.

Ai giorni nostri, l'analisi dei Social Network interessa i più disparati campi di ricerca sociali, psicologici, antropologici, matematici, fisici, economici e matematici, oltre che numerose applicazioni nella computer science. La Social Network Analysis studia, infatti, le relazioni che sussistono all'interno di un gruppo di generici elementi, e permette di comprendere quali legami li uniscono. Essa consente di identificare le entità più influenti in un gruppo e di scoprire quali di esse rappresentano l'aggancio necessario per mettere in contatto due o più elementi che, altrimenti, non avrebbero la possibilità di comunicare tra loro. Ad un livello più alto, la Social Network Analysis mostra com'è fatta una rete: ci si può chiedere se le relazioni siano biunivoche e perlopiù omogenee, oppure se il gruppo osservato è formato da sottogruppi o nicchie. Si può comprendere quanto comunicano queste comunità tra loro, e quali sono i membri che costituiscono un ponte, ovvero un passaggio utile e strategico che permette di raggiungere un'altra comunità.

L'ovvio legame tra una grande mole di dati e le piattaforme online, dove i network prendono forma, sta spingendo sempre più a considerare l'analisi delle reti sociali come una nuova, importante, e tuttora poco esplorata, branca di indagine.

2.2 Rappresentazione di una rete sociale

Come visto, la Social Network Analysis, detta anche "teoria delle reti sociali", rappresenta lo studio delle relazioni e trova applicazione in diverse scienze. Il termine "*social*" è, ad oggi, inteso in senso lato, non più con diretto riferimento alle relazioni sociali tra esseri umani, bensì, più genericamente, alle relazioni tra individui, entità ed attori di una relazione di qualunque natura.

Nella seconda metà del XX secolo, gli statunitensi Frank Harary e Dorwin Cartwright hanno individuato nella "teoria dei grafi" il miglior modo per rappresentare una rete sociale. Questa è la disciplina che si occupa dello studio dei grafi, oggetti discreti che permettono di schematizzare una grande varietà di situazioni e processi, e spesso di consentirne delle analisi in termini quantitativi e algoritmici, adattandosi perfettamente alla Social Network Analysis moderna.

La paternità di questa teoria è attribuita ad Eulero, uno dei più importanti matematici del '700, che la usò per rispondere al quesito di geometria topologica noto come "sette ponti di Königsberg": esso trattava di un problema in cui si considerava una reale città, all'epoca situata in Prussia, oggi exclave russa sulla costa baltica con il nome Kaliningrad, all'interno della quale vi erano sette ponti che collegavano i quattro quartieri della città. L'obiettivo era quello di scoprire se fosse possibile attraversare tutti i ponti una sola volta in sequenza, passeggiando da un punto di inizio e concludendo il percorso nel punto da cui si è iniziato. Eulero dimostrò, tramite l'utilizzo della teoria dei grafi, che tale cammino non era possibile.

Per farlo, egli rappresentò (Figura 2.1) la mappa della città utilizzando dei punti (i "nodi", le zone della città) collegati da linee (gli "archi", i ponti), ed enunciò la definizione di *circuito euleriano*: un qualsiasi grafo è percorribile se e solo se ha tutti i nodi di grado pari, o due di essi sono di grado dispari; per percorrere un grafo "possibile" con due nodi di grado dispari, è necessario partire da uno di essi, e si terminerà sull'altro nodo dispari.

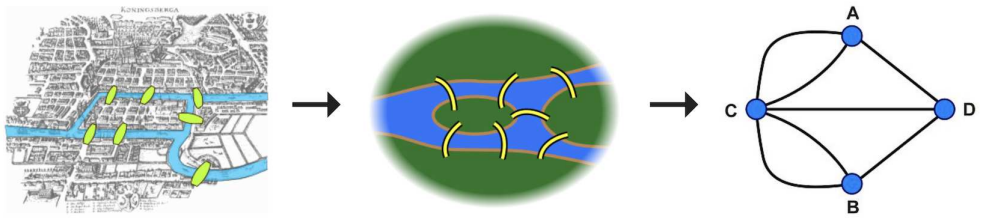

Figura 2.1: Problema dei sette ponti di Königsberg. *Fonte: wikipedia.com*

Il percorso desiderato non è percorribile poiché tutte le aree della città sono collegate tra di loro da un numero dispari di ponti.

Nel XX secolo gli studi sulla teoria dei grafi proseguirono sulla dimostrazione del cosiddetto "problema dei quattro colori", il quale afferma che data una superficie piana divisa in regioni connesse, come ad esempio una carta geografica politica, sono sufficienti quattro colori per colorare ogni regione facendo in modo che regioni adiacenti non abbiano lo stesso colore. Il problema è dimostrabile rappresentando la mappa attraverso un grafo, associando ad ogni regione un nodo; due nodi sono connessi tra loro se e solo se le due regioni corrispondenti hanno un segmento di bordo in comune.

Altri risultati sulla teoria dei grafi si ebbero sempre nel XX secolo con l'enunciazione dei *cammini hamiltoniani*: un cammino in un grafo è detto hamiltoniano se esso tocca tutti i vertici del grafo una e una sola volta. Il nome fu attribuito in onore del matematico William Rowan Hamilton (1805–1865), il quale ideò un gioco matematico sul bordo di un dodecaedro; chiaramente l'astrazione che ne segue è la rappresentazione dei vertici della figura in nodi del grafo, mentre gli spigoli sono rappresentati dai collegamenti tra nodi.

Alla luce di ciò risulta chiaro come i grafi siano strumenti molto potenti ed adattabili a molteplici situazioni, come, ad esempio, alla tipica rappresentazione delle relazioni tra utenti di un social network quale TikTok, ove, a seconda del taglio che si vuole dare allo studio, i nodi e gli archi del grafo che ne consegue possono assumere diversi significati.

2.2.1 I grafi

Come anticipato in precedenza, una base comune per i programmi dell'analisi delle reti sociali è l'approccio matematico della teoria dei grafi, che fornisce un linguaggio formale per la descrizione delle reti e dei loro caratteri. L'elemento caratterizzante è il grafo, un insieme di elementi, detti nodi o vertici, che possono essere collegati a due a due da delle linee chiamate archi o lati o spigoli; la presenza di un arco indica una relazione tra i due nodi interessati.

Più formalmente, si definisce grafo non orientato una coppia $G = (V, E)$ in cui V è un insieme finito di elementi, detti nodi o vertici, ed E è una famiglia di coppie non ordinate di elementi appartenenti a V. Agli elementi di E, che sono detti archi o lati, è possibile associare una funzione matematica $f : E \to \mathbb{R}$; in questo caso gli archi vengono detti archi pesati, e la funzione esprime il valore che l'arco assume nella rappresentazione. Se una coppia $(i, j) \in E$ compare almeno due volte all'interno dell'insieme si parla di archi multipli. Se, per concludere, le coppie in E sono ordinate allora si parla di grafo orientato; in questo caso l'ordine definisce la direzione della relazione

(Figura 2.2). Se la relazione è asimmetrica il grafo risulterà orientato, altrimenti il grafo sarà non orientato.

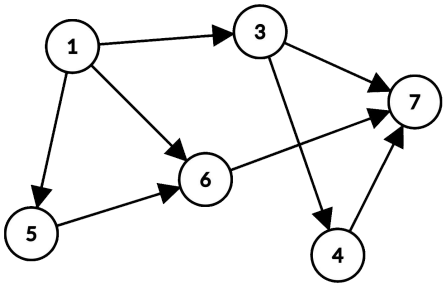

Figura 2.2: Rappresentazione di un grafo orientato. *Fonte: autoprodotta*

All'interno di un grafo possono essere individuate delle caratteristiche interessanti, che possono assumere una diversa semantica a seconda del contesto in cui sono valutate. Un percorso ("*walk*") è una sequenza di nodi e linee, non necessariamente tutti distinti, che descrive un tragitto da un nodo di partenza ad un nodo di destinazione. Un percorso con nodi e linee tutti distinti prende il nome di sentiero ("*path*"). Un percorso chiuso in cui ogni linea e ogni nodo sono inseriti in sequenza una ed una sola volta, tranne il nodo di origine, si definisce ciclo ("*cycle*"). Possono esistere due o più percorsi tra due nodi con lunghezze differenti, e quelli generalmente usati negli algoritmi sono i percorsi più brevi ("*shortest path*"): se il grafo non è pesato si valuta la distanza geodetica, che rappresenta il numero di archi che separano due nodi, altrimenti, se il grafo è pesato, bisogna valutare, in aggiunta al numero di archi, anche la somma dei pesi degli archi stessi. Il diametro, invece, è la lunghezza del percorso più lungo che collega coppie di nodi ("*largest path*").

2.2.2 Indici della rete

Con la teoria dei grafi sono state introdotte diverse metriche relative ad alcune proprietà il cui studio permette una maggiore comprensione del grafo in esame. Il grado di un nodo ("*degree*") è una funzione che indica il numero di archi adiacenti al nodo stesso, che, nel caso di un grafo orientato, possono essere entranti ("*indegree*") oppure uscenti ("*outdegree*"). Questo concetto si collega a quello di eccentricità, definita come il massimo valore assunto dalla funzione grado di ogni nodo del grafo.

Un'altra metrica molto utilizzata è il coefficiente di clustering. Esso esprime la misura di quanto i nodi di un grafo tendono ad essere connessi fra loro, formando sottografi completi. Il coefficiente di clustering locale C_i di un vertice $v_i \in V$ è dato dal numero di collegamenti fra i membri di N_i, definito come l'insieme dei nodi direttamente connessi a v_i, diviso il numero di collegamenti potenziali fra loro.

La densità di una rete descrive, invece, il livello di saturazione delle relazioni fra i nodi. In altri termini, essa misura quante siano le connessioni attive fra quelle che, dato il numero di nodi, potrebbero potenzialmente sussistere. In particolare, il grado di densità può essere definito come $D = \frac{2 \cdot |E|}{|V| \cdot (|V|-1)}$. La densità di un grafo assume valori compresi tra 0 ed 1 e la si può ricollegare facilmente al concetto di probabilità; essa misura la probabilità che una qualsiasi coppia di nodi sia adiacente, mentre la connessione di un grafo dipende dalla distribuzione degli archi tra i nodi.

Nello specifico della Social Network Analysis, può essere utile misurare il potere o l'influenza dei componenti sulla base delle loro connessioni: una misura di questo tipo calcola l'indice di centralità del componente. Questo fattore, tuttavia, può essere valutato in diversi modi, in funzione delle diverse proprietà che si tengono in considerazione. Quattro tra le più importanti sono:

- *Degree centrality*: è la più semplice forma di centralità, misura "l'importanza" di un nodo in funzione del grado che esso assume. Questa metrica permette di individuare le celebrità di una comunità, che generalmente sono in numero limitato ed hanno un grado significativamente più elevato rispetto a tutti gli altri elementi nella rete.
- *Closeness centrality*: focalizza la propria attenzione su quanto un attore è vicino agli altri. Un attore, quindi, è tanto più centrale nella rete quanto più è nella posizione di interagire velocemente con gli altri attori. Il suo valore è inversamente proporzionale alla distanza geodetica: meno si è distanti dagli altri nodi e più si è centrali.
- *Betweenness centrality*: definisce il ruolo di mediatore all'interno della rete. I nodi che si collocano nella posizione di intermediari, ovvero localizzati sui percorsi che collegano coppie di nodi non adiacenti, possono esercitare un importante potere di controllo sul flusso delle informazioni.
- *Eigenvector centrality*: rappresenta il ruolo di "eminenza grigia" all'interno della rete, ovvero un decisore potente che opera segretamente ed in modo non ufficiale. Il valore di questa metrica è tanto maggiore quanto più il nodo è connesso a molti nodi che sono, a loro volta, centrali.

Nella ricerca delle proprietà caratterizzanti di un grafo, esistono diversi algoritmi, come il *PageRank*, ideato e descritto nel relativo paper dai fondatori di Google, che forniscono ulteriori misure utili. Nello specifico, *PageRank* valuta l'importanza di ogni nodo del grafo, basando il calcolo sul numero di archi entranti su di esso ed il valore del nodo d'origine all'interno del grafo stesso; l'assunzione fondamentale è che un nodo è importante quanto più sono importanti i nodi che vi si collegano.

2.2.3 Le principali strutture

Nell'analisi di una rete, oltre a valutare le metriche già elencate, risulta utile analizzare la struttura della rete al fine di studiarne la topologia. Quando si parla di social network, infatti, è possibile individuare diverse componenti tra di loro scollegate, facenti riferimento a distinte comunità di elementi.

Un grafo si dice connesso se esiste un percorso tra ogni coppia di nodi nel grafo, altrimenti, lo si definisce sconnesso; un elevato numero di componenti sconnesse può indicare, a seconda del contesto, una scarsa coesione tra gli attori della rete. Si definiscono ponte ("*bridge*") e punto di separazione ("*cutpoint*"), rispettivamente, un arco ed un nodo che, se soppressi, sconnettono il grafo. Un nodo è definito raggiungibile se esiste un percorso che lo collega agli altri nodi, indipendentemente dalla sua lunghezza e dagli intermediari attraversati dal percorso. Un nodo isolato, al contrario, è definito come non raggiungibile e la sua distanza dagli altri è infinita.

Altre strutture utili dettagliano gli insiemi di nodi ed archi. Tra queste, il componente fondamentale è la triade, definito come un insieme di tre nodi (Figura 2.3):

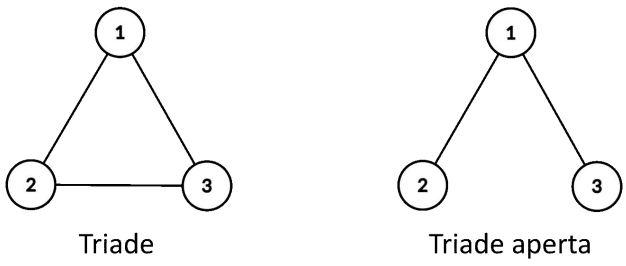

Figura 2.3: Principali tipologie di triadi. *Fonte: autoprodotta*

Una triade è chiusa se tutti i nodi compresi nella struttura sono connessi tra loro. Questa indica un collegamento molto forte tra le entità coinvolte, poiché esse hanno due vie di comunicazione; nell'esempio in Figura 2.3 l'oggetto 1 può comunicare con l'oggetto 2 sia direttamente sia attraverso il terzo oggetto. Alternativamente, le triadi possono essere aperte, ed in questo caso si parla di *buchi strutturali*, se il numero di archi che connettono i nodi è pari a 2. L'elemento di collegamento tra i due nodi non comunicanti è un *cutpoint* ed, in alcuni contesti, è di notevole interesse, in quanto può mettere in contatto due intere comunità.

Una clique, o cricca, (Figura 2.4) è un sottoinsieme completo massimale del grafo; i nodi che la compongono sono tra di loro tutti connessi. Una clique è essenzialmente definita per l'insieme delle triadi che la compongono, rendendo quest'ultimo l'elemento fondante della struttura.

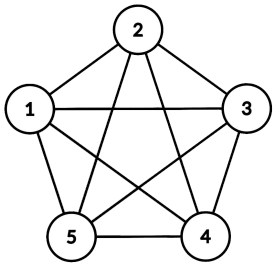

Figura 2.4: Rappresentazione di una clique a cinque nodi. *Fonte: autoprodotta*

Il problema di individuazione delle clique è *NP-hard*; per questo motivo la loro ricerca all'interno di un grafo, specie se questo è di grande dimensione, risulta molto onerosa e si preferisce rilassare il problema individuando i k-core. Un k-core è un gruppo massimale di nodi ciascuno dei quali connessi ad almeno k altri nodi del gruppo.

Un'ultima struttura molto importante è l'ego-network, ovvero il sottografo dei vicinati di un nodo con grado elevato. All'interno di un grafo di grandi dimensioni, il nodo centrale di una ego-network è un *cutpoint* tra un numero $n \geq 2$ di nodi tra loro non connessi; esso rappresenta l'unico punto di comunicazione tra un elevato numero di attori. A seconda della tipologia della rete, queste particolari strutture possono assumere una semantica differente, ed il loro studio è di notevole interesse.

3. Costruzione del dataset di riferimento

3.1 Strumenti utilizzati

Durante la realizzazione del progetto sono stati utilizzati diversi software di terze parti a supporto del flusso di lavoro adottato, consentendo di sviluppare e condividere software, nonché di visualizzare efficacemente i risultati prodotti.

Tra questi, *Git* è un sistema distribuito di controllo di versione open-source, progettato per gestire efficientemente qualsiasi tipo di progetto di qualunque dimensione. Esso è caratterizzato da alcuni elementi chiave, tra cui:

- Supporto allo sviluppo non lineare: il software offre funzionalità quali diramazione e fusione ("*branching*" e "*merging*") e comprende strumenti specifici per visualizzare e navigare una cronologia di sviluppo non lineare.
- supporto allo sviluppo distribuito: *Git* fornisce ad ogni sviluppatore una copia locale, detta *repository*, dell'intera cronologia di sviluppo. Eventuali modifiche possono essere trascritte, tramite opportuni comandi, nella copia di ogni sviluppatore, che le importa da un servizio *hosting* remoto come diramazioni di sviluppo, e le fonde allo stesso modo di una diramazione sviluppata localmente. La scelta del *provider* di servizio *hosting* remoto di *repository* è ricaduta su *GitHub*, la più nota piattaforma di *hosting* per progetti software.
- Autenticazione crittografica della cronologia: viene memorizzata una cronologia delle operazioni, in modo tale che il nome di una modifica particolare ("*commit*") dipenda dalla completa cronologia di sviluppo che conduce a tale modifica. Una volta che un *commit* è stato pubblicato, non è più possibile modificare una versione del progetto precedente, senza che ciò venga notato.

Il software utilizzato per tutto lo sviluppo del codice è *Visual Studio Code*. Esso è un editor di codice sorgente, scritto in linguaggio *TypeScript*, sviluppato da *Microsoft* per tutti i principali sistemi operativi. Esso include nativamente il supporto per tutti i principali linguaggi di programmazione, un motore di debugging altamente configurabile, un'interfaccia per *Git* integrata, *IntelliSense* e *refactoring* assistito del codice. Tuttavia, le sue funzionalità possono essere estese mediante l'installazione di estensioni gratuite disponibili in un *repository* centralizzato, che lo rendono un *Integrated Development Environment* molto potente. Infatti, ad Agosto 2021, *Visual Studio Code* si

è posizionato come strumento di sviluppo più popolare tra gli sviluppatori che hanno partecipato allo "*Stack Overflow 2021 Developer Survey*"; il 71.06% degli 82,277 partecipanti al sondaggio hanno dichiarato di utilizzarlo.

In ultimo, per una visualizzazione più chiara delle reti prodotte, è stato adottato *Gephi*, un software open-source per l'analisi e la visualizzazione delle reti sociali, scritto in *Java* e basato sulla piattaforma *NetBeans*. Il suo sviluppo è stato avviato nel 2008 dagli studenti dell'*Université de technologie de Compiègne*, ed oggi è mantenuto dal *Gephi Consortium*, una società francese no-profit.

3.1.1 Python e le librerie

Il linguaggio di programmazione utilizzato per lo sviluppo del progetto è *Python* (Figura 3.1). Questo è un linguaggio di programmazione ad alto livello orientato ad oggetti e multi-paradigma, particolarmente adatto, tra gli altri usi, per sviluppare applicazioni distribuite, scripting e system testing. Python risulta estremamente efficiente, anche se è considerato un linguaggio interpretato; i programmi vengono automaticamente compilati in un formato chiamato bytecode prima di essere eseguiti. Questo formato è più compatto e più efficiente, e garantisce, quindi, prestazioni molto elevate. Inoltre, diverse strutture dati, funzioni e moduli della *standard library* di Python sono implementati interamente in linguaggio *C*, per essere ancora più performanti; questa è una collezione di oltre 200 moduli installati direttamente con la distribuzione di Python scelta, utilizzati per svolgere diversi compiti, come, ad esempio, l'interazione con il sistema operativo ed il file system.

Figura 3.1: Logo del linguaggio Python. *Fonte: python.org*

La sua sintassi elegante lo rende un linguaggio di programmazione estremamente semplice da leggere ed utilizzare; il suo design si basa sul principio del *least astonishment*, cioè della "minima sorpresa", rendendolo particolarmente adatto sia ai neofiti che agli utenti più esperti. Date le sue caratteristiche, e vista la sua versatilità, Python è stato prontamente apprezzato dalla community e negli ultimi anni è stato soggetto ad una diffusione esponenziale. Python, infatti, si è arricchito di un elevato numero di librerie di terze parti facilmente integrabili, modularmente, mediante l'utilizzo di un package manager (come, ad esempio, *pip*).

Tra queste librerie di terze parti, quelle che sono risultate più utili durante lo sviluppo del progetto comprendono moduli di gestione dei dati, di visualizzazione e di analisi matematica. Le principali librerie utilizzate sono `Pandas`, `NetworkX`, `Matplotlib` e `Scikit-Learn`.

`Pandas` è una libreria software ad alto livello per la manipolazione e l'analisi dei dati, che si basa su `NumPy`, un'altra libreria che aggiunge supporto a grandi matrici e array multidimensionali, oltre che una vasta collezione di funzioni matematiche per poter operare efficientemente su queste strutture dati. In particolare, `Pandas` offre strutture dati e operazioni per manipolare tabelle numeriche e serie temporali. Il nome deriva dal termine "*panel data*", definizione econometrica di dataset che include osservazioni su più periodi di tempo, per gli stessi individui. Questa libreria è stata utilizzata durante tutte le fasi di gestione e manipolazione dei dati, sfruttando la sua caratteristica di ottima gestione dei file in formato `.csv` (*Comma Separated Values*).

`NetworkX` è una libreria per la creazione, la manipolazione e lo studio della struttura, della dinamica e delle funzioni di reti complesse. Essa permette di operare con formati di dati sia standard che particolari, e mette a disposizione dell'utilizzatore un grande numero di funzioni che permettono di analizzare quantitativamente i principali indici di rete, di estrarre strutture come

componenti sconnesse o fortemente connesse, di costruire nuovi modelli di rete, di progettare nuovi algoritmi e di disegnare reti, permettendo infinite possibilità di personalizzazione.

`Matplotlib` è una libreria ad alto livello, basata su `NumPy`, per la creazione di visualizzazioni statiche, animate e interattive in Python. Nell'ambito del progetto, `Matplotlib` è stata molto utile in fase di visualizzazione di grafici.

Per concludere, `Scikit-learn` è un modulo ad alto livello open-source, che fornisce un grande numero di algoritmi di apprendimento automatico; essa contiene metodi di classificazione (tra cui *naive bayes*, *k-nearest neighbour*, *support vector machine* e *random forest*), regressione e clustering. Come tutte le principali librerie di analisi, `Scikit-learn` è basata sulle librerie `NumPy` e `SciPy`, una libreria open-source per la distribuzione di algoritmi e strumenti matematici, contenente moduli per l'algebra lineare, l'integrazione matematica, algoritmi per il calcolo di transformate di *Fourier*, e molti altri strumenti utili nelle scienze e nell'ingegneria.

3.2 L'idea nell'acquisizione dei dati

Il progetto, realizzato in collaborazione con il *DII*, Dipartimento di Ingegneria dell'Informazione dell'Università Politecnica delle Marche, riguarda l'analisi di dinamiche dei comportamenti degli utenti della piattaforma TikTok. Per ottenere i dati necessari all'analisi, è possibile interfacciarsi con l'*API* (*Application Programming Interface*) distribuita dalla piattaforma stessa.

Le API, coordinate da un *API Management System* (Figura 3.2), sono degli strumenti messi a disposizione dalle organizzazioni che consentono di accedere ad uno specifico *pool* di funzioni accessorie ad un programma. A seconda dei privilegi in possesso all'utente che effettua la richiesta, esse consentono di interrogare il server ospitante l'applicazione per ottenere informazioni non visibili altrimenti, per inserire nuove informazioni o per modificare quelle già presenti. Per richiedere al programma di effettuare determinate operazioni, è necessario seguire delle regole, dettate dalle API stesse.

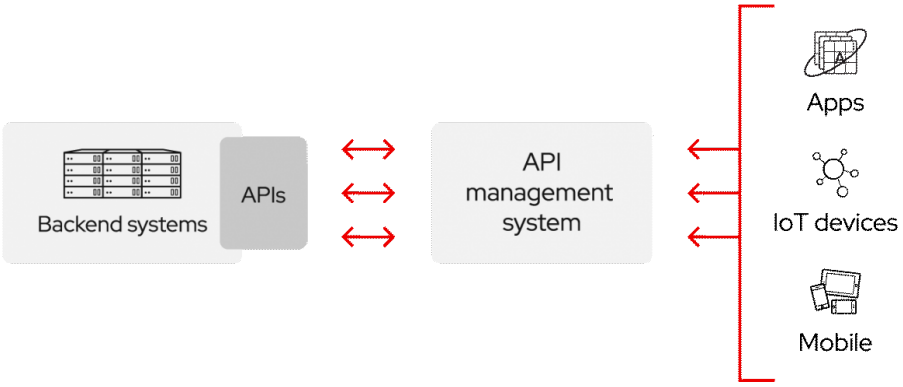

Figura 3.2: API workflow. *Fonte: redhat.com*

Tutte le API messe a disposizione dai servizi web moderni consentono di essere interrogate tramite *HTTP* (*HyperText Transfer Protocol*), un protocollo a livello applicativo usato come principale sistema per la trasmissione delle informazioni sul web; proprio per questo motivo, ogni dispositivo che dispone di un *access point* verso la rete Internet può interfacciarsi con tali API e comunicare con loro.

Nello specifico del progetto, è possibile leggere le informazioni sui video caricati dagli utenti tramite una richiesta di tipo "*GET*" all'*URL* "`https://m.tiktok.com/api/`", aggiungendo a tale stringa il giusto percorso per ottenere i dati desiderati. In caso di richiesta correttamente formulata, l'API di TikTok restituisce al client i dati in formato *JSON* (*JavaScript Object Notation*), adatto

all'interscambio di dati tra applicazioni client/server. Una tipica risposta è, quindi, strutturata come segue:


```
{
    "id":"6969483540358581505",
    "desc":"#fyp",
    "createTime":1622709341,
    ...
    "video":{
        "id":"6969483540358581505",
        "duration":13,
        "format":"mp4",
        ...
    },
    "author":{
        "id":"6913108625435132933",
        "uniqueId":"danathebee",
        "verified":false,
        ...
    },
    "music":{
        "id":"6958985308046232325",
        "title":"Levan Polkka (Cover)",
        ...
    },
    "originalItem":false,
    ...
    "authorStats":{
        "followingCount":31,
        "followerCount":4200000,
        "videoCount":199,
        ...
    },
    ...
}
```


3.2.1 Il wrapper TikTokApi: la libreria utilizzata

Lato client, si è utilizzato il modulo *TikTokApi*, un *wrapper* Python non ufficiale per interagire con l'API di TikTok, sviluppato dallo studente statunitense David Teather. Nell'eseguire le richieste, TikTokApi utilizza *Playwright*, una libreria Python per automatizzare i browser *Chromium*, *Firefox* e *WebKit* con una singola interfaccia.

Alla Versione 3.9.2, utilizzata durante lo svolgimento dell'analisi, il modulo mette a disposizione un gran numero di metodi specifici per sfruttare le diverse funzionalità distribuite dall'API di TikTok; la libreria contiene metodi per acquisire informazioni sui video di tendenza e consente di effettuare richieste per *hashtag*, per musica, per autore e per singolo video. Inoltre, TikTokApi permette di ottenere una serie di informazioni riguardo le statistiche sui singoli utenti. Ad ogni richiesta effettuata, il wrapper restituisce al chiamante tutte le informazioni impacchettate in strutture dati di semplice consultazione tramite Python.

3.3 Definizione delle challenge

Scelti gli strumenti con cui condurre la ricerca, è opportuno selezionare un certo numero di challenge non pericolose, o positive, bilanciato da uno stesso numero di challenge pericolose, o negative. Le challenge positive corrispondono a trend che hanno un impatto positivo sulle persone, che non coinvolgono azioni pericolose e che, quindi, risultano divertenti ed interessanti. Viceversa, poiché i trend particolarmente pericolosi risultano censurati dalla piattaforma TikTok stessa, come già

evidenziato, si son considerati come "negativi" quei trend che risultano a cattivo impatto sugli utenti che decidono di parteciparvi, che possono arrecare danni indirettamente a cose o persone, e che portano tematiche sensibili e/o moralmente discutibili.

Tra le varie possibilità, sono state individuate le seguenti sette challenge positive:

- ***#bussitchallenge***: trend creato da ragazze che si presentano senza trucco ed in tenuta casalinga, per poi passare improvvisamente, a seguito di un breve transizione, ad essere vestite al meglio, sistemate in trucco e parrucco; la musica che identifica la challenge è "Buss It", prodotta da Erika Banks.
- ***#copinesdancechallenge***: esecuzione di una particolare coreografia di ballo sulle note di "Fly", di Aya Nakamura.
- ***#emojichallenge***: sfida che consiste nel simulare le *emoji* utilizzate nelle chat attraverso le espressioni facciali e l'ausilio di oggetti di uso quotidiano. Alla challenge non è associata alcuna musica specifica.
- ***#itookanap***: si riprende una persona o un animale domestico che sta riposando in posizioni divertenti. Al trend è associata l'omonima musica pubblicata dall'utente "gunnarolla".
- ***#colpiditesta***: la persona che viene ripresa deve cercare di colpire una palla virtuale con la propria testa, simulando dei veri e propri palleggi, senza farla cadere a terra. La challenge non ha alcuna musica specifica associata.
- ***#boredinthehouse***: trend divenuto popolare nel 2020 durante l'esplosione dell'epidemia mondiale di *COVID-19*, in cui le persone si mostravano annoiate in casa, intente ad effettuare azioni quotidiane. Il trend prende il nome dall'omonima canzone "Bored in the House" di Tyga.
- ***#plankchallenge***: sfida di resistenza in cui gli utenti devono eseguire passi di danza basati su esercizi fisici. Le musiche utilizzate in questa challenge sono a discrezione del gusto soggettivo dell'utente, per questo motivo non c'è un unico audio che la identifica.

Per quanto riguarda, invece, le challenge considerate negative, sono state scelte le seguenti:

- ***#silhouettechallenge***: trend identificato da un mashup delle canzoni "Put your head on my shoulder", di Paul Anka, e "Streets", di Doja Cat, prodotto da Giulia Di Nicolantonio. La prima parte del video presenta l'autore del video vestito, ma, a seguito della transizione audio, questo appare parzialmente svestito ed offuscato da un filtro rosso, che lo fa apparire in ombra. Il pericolo consiste nel fatto che è stato individuato il modo per eliminare a posteriori il filtro applicato, con la possibilità di mettere a rischio la privacy degli utenti che decidono di partecipare a questo trend. Sebbene questo sia nato per combattere ogni forma di discriminazione legata al peso e alla forma del corpo, non è auspicabile che le proprie foto in intimo possano circolare sul web.
- ***#bugsbunnychallenge***: ragazze distese sul ventre simulano di avere orecchie da coniglio, utilizzando i piedi e una prospettiva di ripresa che permette loro di ottenere la sagoma corretta. Tuttavia, alcuni utenti realizzano il video in abiti succinti, mostrando parti intime con il concludersi del video. La musica che identifica il trend è l'omonima "Bugs Bunny", degli artisti russi GERDA & DARI.
- ***#strippatiktok***: in questa challenge, spogliarellisti e spogliarelliste mostrano la loro vita ed i loro guadagni, portando contenuti poco adatti all'utenza più giovane. Questa challenge non si identifica con un unico audio.
- ***#firewroks***: trend che consiste nel far esplodere fuochi d'artificio in prossimità di soggetti ed oggetti. Questo può essere potenzialmente dannoso per chi si trova vicino ad un'esplosione indotta da persone non professioniste, poco consce del pericolo della loro azione. Anche questa challenge non si identifica con un unico audio.
- ***#fightchallenge***: challenge in cui adolescenti, ragazzi o adulti ne riprendono altri mentre combattono. È considerato un trend negativo in quanto l'emulazione delle azioni proposte nei

relativi video da parte di utenti non professionisti può arrecare danno alle persone coinvolte. Nuovamente, non vi è alcun audio specifico che caratterizza il trend.

- ***#sugarbaby***: alcuni giovani utenti mostrano il lusso ottenuto dalla frequentazione con persone benestanti, spesso anziane. Viene considerata negativa per l'influenza che potrebbe avere sugli adolescenti e le nuove generazioni, oltre ai pericoli che una frequentazione di questo tipo potrebbe portare. Anche in questo caso la challenge non si identifica con un audio specifico, è molto frequente che il suono del video corrisponda all'audio registrato dal microfono dello smartphone.
- ***#updownchallenge***: è un gioco di coppia in cui si afferrano mani e piedi del partner e si cerca di ribaltare la posizione attuale da terra. Il rischio è quello di ferirsi sbattendo la testa nel momento in cui ci si capovolge. Non vi è un unico audio che identifica il trend.

In totale, quindi, sono state selezionate 14 challenge.

3.4 Algoritmo per la raccolta dati

Per proseguire, è stato progettato un primo algoritmo in grado di scaricare i dati relativi ai singoli video appartenenti alle diverse challenge, e di organizzarli in un formato facilmente manipolabile tramite la libreria Pandas. Lo schema fondamentale di acquisizione comprende tre fasi principali riassumibili nel seguente modo:

1. richiesta dei dati relativi ai video di una specifica challenge, tramite l'*hashtag* associato;
2. estensione del dataset attraverso l'acquisizione di ulteriori dati a partire dagli autori dei video acquisiti durante la prima fase;
3. manipolazione dei dati e generazione del dataset finale.

L'intera procedura è stata automatizzata e centralizzata nel file `"main.py"`, eseguibile da riga di comando, contenuto nella *working directory* principale (Figura 3.3). Il suddetto file fa uso del parametro **`-c`**, che specifica la chiave della challenge di cui si intende ottenere dati.

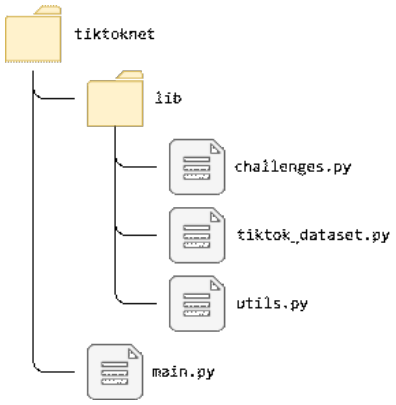

Figura 3.3: Struttura della working directory per l'acquisizione dei dati. *Fonte: autoprodotta*

All'atto dell'esecuzione dello script `main.py`, viene inizializzata una nuova connessione all'API di TikTok tramite il comando `TikTokApi.get_instance`.

La prima fase dell'algoritmo fa uso del metodo `byHashtag`, definito nel wrapper utilizzato, per recuperare i dati forniti dall'API, a partire dalle chiavi contenute in `challenges.py`. Quest'ultimo, è un file di supporto contenente un dizionario dove sono specificati diversi *hashtag* relativi ad un'unica challenge; è comune, infatti, individuare, all'interno dell'applicazione TikTok, diversi *hashtag* che fanno riferimento ad un unico trend. Questo è particolarmente vero nel caso di challenge pericolose, ove, per ingannare l'algoritmo di verifica dei contenuti, e, quindi, per evitare problemi legati al proprio account, gli utenti variano leggermente la sintassi delle parole chiave

utilizzate. Un esempio di ciò è rappresentato dal trend "*#sugarbaby*", individuato anche, tra altri, dagli *hashtag* "*#suggarbaby*" ed "*#suggarbabby*", intenzionalmente con sintassi scorretta.

Al termine di questo processo si ottiene una lista di video, arricchita da un grande numero di informazioni aggiuntive, relazionati tra loro solo per l'*hashtag* utilizzato. Lavorando in un'ottica di Social Network Analysis, le informazioni ottenute fino a questo punto non sono sufficienti per generare alcun tipo di rete. La seconda fase dell'algoritmo ha, quindi, lo scopo di scaricare nuovi dati con il successivo intento di realizzare dei collegamenti tra gli autori dei video in base alla loro eventuale interazione reciproca. Ciò permette di realizzare le reti di interesse, la cui semantica verrà approfondita successivamente.

Il secondo passo dell'algoritmo consiste nell'analizzare le caratteristiche degli autori dei video ottenuti durante la prima fase. La funzione `userLiked`, contenuta nel wrapper, restituisce, dopo aver interrogato l'API, l'elenco dei video con cui l'autore ha interagito assegnando un "mi piace". La funzione è stata usata con due scopi differenti: inizialmente per capire quali utenti hanno impostato la lista dei video piaciuti come "pubblica" e, successivamente, per recuperare le vere e proprie informazioni relative ai video piaciuti. Infatti, come discusso in precedenza, TikTok fornisce a ciascun utente la possibilità di nascondere al pubblico, mediante opportune impostazioni di privacy, diverse informazioni, tra cui il suddetto elenco; ciò impedisce all'API di recuperare nuovi dati, bloccando, di fatto, nella sua ricostruzione *bottom-up*, una possibile catena di interazione tra autori.

3.4.1 Problematiche nell'acquisizione dei dati

Durante lo sviluppo delle prime due fasi dell'algoritmo sono venute alla luce diverse problematiche relative all'utilizzo degli strumenti fino ad ora discussi. Il primo problema riguarda la quantità di informazioni che, ad ogni richiesta, l'API di TikTok fornisce, e la modalità con cui esse vengono erogate. La quantità di dati forniti non supera mai i 5000 video per specifico *hashtag*, e non è possibile dichiarare né un intervallo temporale di ricerca né ulteriori filtri, in quanto la ricerca per *hashtag* dell'API si basa esclusivamente su un'unica parola chiave, e restituisce una specifica selezione di video che, se richiamata a distanza di breve tempo, non varia mai. Questa caratteristica non ha permesso di costruire dataset molto grandi ed accurati, costringendo ad un ridimensionamento della portata del progetto e delle reti.

Oltre a ciò, l'API di TikTok è molto lenta nel fornire i risultati di una *query*; anche quando i dati restituiti sono nulli, come nel caso in cui venga effettuata una ricerca sui video piaciuti da un utente che, tuttavia, ha impostato la lista associata come "privata", l'API impiega all'incirca 1 secondo a rispondere, e l'attesa aumenta proporzionalmente all'aumentare dei dati restituiti.

Un ulteriore fattore limitante è dettato dall'applicazione di TikTok stessa. Infatti, ad ogni registrazione di un utente, la lista dei video piaciuti dello stesso viene inizializzata come "privata"; di conseguenza, un altissimo numero di utenti, stimato pari all'85%, mediante il metodo di supporto `checkAvLikedPerc` contenuto in `utils.py`, mantengono privata, per volontà o mancata conoscenza, tale lista. Ciò non permette all'algoritmo di ampliare il dataset generato e trovare nuovi collegamenti con cui alimentare le reti.

In ultimo, si è dovuti intervenire sia nella configurazione del wrapper utilizzato, sia sul codice sorgente dello stesso. Di default, una volta instanziato il wrapper mediante il già citato metodo `TikTokApi.get_instance`, questo effettua richieste al dominio `m.tiktok.com`. Tuttavia, poiché il wrapper utilizzato non è ufficiale ed il suddetto dominio rifiuta spesso le richieste da parte di servizi non ufficiali, è stato necessario utilizzare il parametro `use_test_endpoints=True`, che reinderizza la richiesta al dominio di test `t.tiktok.com`, meno protetto e maggiormente permissivo. Oltre a ciò, il wrapper esegue le proprie richieste attraverso il browser *Playwright*; quest'ultimo smette regolarmente di funzionare dopo un elevato numero di richieste, interrompe la sua esecuzione restituendo un errore con la diretta conseguenza che tutti i dati fino ad allora processati vengono

persi. Non essendo presente, all'interno del file `browser.py` della libreria, una funzione di reset è stato necessario implementarla come segue:

```
def clean_playwright():
    global playwright
    playwright.stop()
    playwright=None
```

In questo modo, una volta restituito l'errore a *runtime* da parte del browser, è possibile intervenire resettando e riavviando sia *Playwright* che l'istanza del wrapper stesso (`api`):

```
...
except:
    browser.clean_playwright()
    api.clean_up()
    api=TikTokApi.get_instance(use_test_endpoints=True)
```

3.4.2 Data processing: fase di ETL

La terza fase dell'algoritmo è relativa a tutte le operazioni di estrazione, manipolazione e salvataggio dei dati in opportune strutture. A differenza delle prime due fasi sequenziali, questa è realizzata per tutta l'esecuzione dello *script*, e chiude l'algoritmo.

In un processo di *data analytics*, cioè di raccolta ed analisi di grandi volumi di dati al fine di estrarre informazioni nascoste, questa fase prende il nome di "*Extract, Transform and Load*" (*ETL*). ETL è, a sua volta, un processo composto da tre passaggi:

1. ***Extract***: i dati grezzi, inizialmente estratti da una vasta gamma di sorgenti, quali database esistenti, registri di attività, report su anomalie e prestazioni di applicazioni ed eventi di sicurezza, devono essere caricati in destinazioni che gli analisti possono utilizzare come punto di partenza per la seconda fase del processo. Nello specifico del progetto, è necessario convertire i dati in formato *JSON*, ottenuti interrogando l'API di TikTok, in un oggetto di tipo `DataFrame`, una struttura dati bi-dimensionale in grado di ospitare dati eterogenei in forma tabellare, definita dalla libreria `Pandas`. Per fare ciò, all'interno del file `utils.py` si è implementata la funzione `datasetHelper`, riportata di seguito:

   ```
   import pandas as pd
   def datasetHelper(tiktoks):
       rows = []
       for tiktok in tiktoks:
           _row = {}
           for key in tiktok:
               elem = tiktok.get(key)
               if isinstance(elem, dict) or isinstance(elem, list):
                   if isinstance(elem, list):
                       elem = elem[0]
                   for elem_k in elem:
                       _row[key + "_" + elem_k] = str(elem.get(elem_k)).replace(";", "")
               else:
                   _row[key] = str(elem).replace(";", "")
           rows.append(_row)
       return pd.json_normalize(rows)
   ```

2. ***Transform***: la fase di trasformazione del processo di ETL consiste nel manipolare i dati estratti al primo passo al fine di prepararli al caricamento, o salvataggio, nella destinazione finale prestabilita, sia essa un software di data warehousing o un'architettura di altra natura. Tra le operazioni più comuni vi sono attività di pulizia dei dati, per rimuovere dati sporchi, poco utili o ridondanti, attività di integrazione tra informazioni ottenute da sorgenti distinte e creazione di nuovi campi calcolati a partire da quelli estratti in origine. Nello specifico

del progetto, le operazioni di trasformazione affrontate sono molteplici, e riguardano sia operazioni di pulizia, per la rimozione di campi poco utili ottenuti mediante l'interrogazione all'API di TikTok, sia il calcolo di nuovi attributi, per definire la semantica delle reti. I dati ottenuti originariamente, infatti, sono stati arricchiti di 3 campi molto utili per le successive analisi:

- `likedBy_id`: rappresenta il codice identificativo di un utente che ha interagito con lo specifico video assegnando "mi piace".
- `likeBy_secUid`: rappresenta un secondo codice identificativo dell'utente che ha interagito con lo specifico video.
- `likedBy_uniqueId`: rappresenta lo *username* dell'utente che ha assegnato "mi piace" al video corrente.

Per ogni elemento del `DataFrame` di `Pandas`, i tre campi corrispondono allo stesso utente. L'algoritmo di calcolo funziona come riportato di seguito:

(a) Dato l'elenco degli autori di video relativi ad una specifica challenge, con privacy sulla lista dei video piaciuti impostata come "pubblica", l'algoritmo inizia interrogando l'API di TikTok al fine di scaricare informazioni su tutti i video che sono piaciuti a ciascuno di loro.

(b) Si cercano corrispondenze tra gli *hashtag* dei video contenuti nella lista "piaciuti" con quelli definiti nel file `challenges.py`, per stabilire se l'utente abbia interagito, o meno, con un video relativo alla challenge in questione.

(c) Se vi sono corrispondenze, si cerca all'interno del dataset ottenuto al primo passo dell'algoritmo generale di estrazione dei dati; un video già presente viene arricchito dei nuovi campi con le informazioni sull'autore corrente, un video non presente viene aggiunto al `DataFrame`. Al termine della procedura, i video che non hanno ricevuto interazioni risultano avere valori nulli per i tre campi calcolati.

Altre operazioni di trasformazione riguardano il ridimensionamento del `DataFrame`, che contiene inizialmente tutti i campi estratti mediante l'API di TikTok, passando da 106 attributi ad un totale di 27, compresi i tre campi calcolati, selezionando accuratamente i più informativi.

3. ***Load***: l'ultima fase del processo di ETL prevede, in genere, il caricamento dei dati estratti e trasformati in una nuova destinazione. Nel caso del progetto in questione, i dati sono sempre salvati, mediante il metodo `to_csv` fornito dalla libreria `Pandas`, in file di testo espresso in formato `.csv`.

3.5 Struttura del dataset di riferimento

Alla luce delle operazioni effettuate, il dataset di riferimento è costituito da un insieme di file in formato `.csv`, uno per ciascun trend individuato. Ciascuna riga del dataset individua un video e le informazioni legate ad esso; in questa fase, il set di dati contiene anche informazioni sui video che non hanno ricevuto interazioni. Nella Tabella 3.1 e 3.2 viene riportata la suddivisione dei video per challenge positive e negative, rispettivamente.

Tabella 3.1: *Suddivisione dei video per challenge positive.*

Challenge	**Numero video**
#bussitchallenge	4903
#copinesdancechallenge	4586
#emojichallenge	4956

#itookanap	4661
#colpiditesta	4966
#boredinthehouse	4449
#plankchallenge	4244

Tabella 3.2: *Suddivisione dei video per challenge negative.*

Challenge	**Numero video**
#silhouettechallenge	3524
#bugsbunnychallenge	6237
#strippatiktok	8254
#firewroks	6240
#fightchallenge	13902
#sugarbaby	6617
#updownchallenge	14556

Ciascuna riga del dataset è costituita da 27 attributi.

3.5.1 Definizione degli attributi

I campi contenuti nel dataset sono riportati, insieme alla corrispettiva descrizione, nella Tabella 3.3.

Tabella 3.3: *Definizione degli attributi del dataset.*

Nome campo	**Tipo campo**	**Descrizione campo**
`id`	`string`	Codice identificativo del video.
`createTime`	`datetime`	Data di creazione del video.
`video_id`	`string`	Codice identificativo del video. I campi `video_id` e `id` corrispondono.
`video_duration`	`int64`	Durata del video in secondi.
`author_id`	`string`	Codice identificativo dell'autore del video.
`author_uniqueId`	`string`	Identificativo assegnato all'autore del video in fase di registrazione.
`author_nickname`	`string`	Nome utente dell'autore del video.
`author_verified`	`bool`	Rappresenta la possibilità o meno che l'autore del video sia verificato, ovvero che rappresenti un personaggio di spicco all'interno della società (possibile *influencer*).

Tabella 3.3: *Definizione degli attributi del dataset.*

Nome campo	**Tipo campo**	**Descrizione campo**
`author_secUid`	`string`	Stringa identificativa dell'autore del video.
`music_id`	`int64`	Codice identificativo del brano audio utilizzato nel video.
`music_title`	`string`	Titolo del brano audio utilizzato nel video.
`music_authorName`	`string`	Nome dell'autore del brano utilizzato nel video.
`stats_diggCount`	`int64`	Numero di "mi piace" assegnati al video.
`stats_shareCount`	`int64`	Numero di volte in cui il video è stato condiviso.
`stats_commentCount`	`int64`	Numero di commenti al video.
`stats_playCount`	`int64`	Numero totale di visualizzazioni del video.
`duetinfo_duetFromId`	`string`	Identificativo del video duetto, presente solo se il video in questione è di tipo "duetto".
`authorStats_diggCount`	`int64`	Numero totale di "mi piace" assegnati dall'autore del video.
`authorStats_followingCount`	`int64`	Numero totale di utenti seguiti dall'autore del video.
`authorStats_followerCount`	`int64`	Numero totale di seguaci dell'utente.
`authorStats_heartCount`	`int64`	Somma del numero di "mi piace" assegnati ai video caricati dall'utente.
`authorStats_videoCount`	`int64`	Numero totale di video caricati dall'utente.
`duetEnabled`	`bool`	*Flag* che determina se il video può essere duettato da un altro utente della piattaforma o meno.
`originalVideo`	`float64`	Il valore 1 identifica il video originale della challenge, se esiste, il valore 0 identifica tutti gli altri.
`likedBy_id`	`string`	Codice identificativo dell'utente che ha interagito con lo specifico video assegnando "mi piace".
`likedBy_secUid`	`string`	Secondo codice identificativo dell'utente che ha interagito con lo specifico video assegnando "mi piace".
`likedBy_uniqueId`	`string`	Nickname dell'utente che ha assegnato "mi piace" al video corrente.

4. Un modello per la rappresentazione delle challenge

4.1 Semantica delle reti

Come anticipato in precedenza, è necessario definire una precisa semantica con cui descrivere le reti che saranno create a partire dal dataset di riferimento. Un grafo, infatti, è costituito da un insieme di nodi ed archi, ove, questi ultimi, esprimono un concetto di relazione tra due nodi. La relazione che si imposta è quella di interazione; in particolare si realizzano dei collegamenti in funzione dei campi calcolati "`likedBy_*`" ("*piaciuto da*"). Ogni riga del dataset corrisponde ad un video relativo ad una challenge di TikTok caricato da uno specifico autore; questo ha, eventualmente, ricevuto un "mi piace" da uno o più utenti che, a loro volta, hanno caricato un video partecipando alla stessa challenge. Tra i due utenti coinvolti si instaura, pertanto, una relazione che si traduce in un arco orientato tra il secondo autore ed il primo.

Più formalmente, definito l'insieme C di tutte le challenge selezionate, si associa un grafo $G_i = (V_i, E_i)$ ad una challenge $C_i \in C$. V_i è l'insieme dei nodi del grafo, ed ogni nodo v_{ij} corrisponde ad un autore a_{ij} che ha caricato un video in C_i. E_i, invece, è l'insieme degli archi di G_i. Ogni arco $(v_{ij}, v_{ik}) \in E_i$ indica che l'utente a_{ik} ha interagito mettendo "mi piace" al video caricato dall'autore a_{ij}, considerando però il timestamp t_{ij}, cioè la data ed ora di caricamento del video associato al nodo v_{ij}, che deve precedere il corrispondente t_{ik}. La presenza di questo arco restituisce l'idea di espansione della challenge C_i verso nuovi utenti, che reagiscono con approvazione al trend e decidono di parteciparvi.

In questo modo non si realizza un'unica rete che abbraccia indiscriminatamente tutte le challenge selezionate, bensì si crea una rete per ognuna delle challenge stesse, con la possibilità di studiarle singolarmente e raggiungere lo scopo principale della ricerca, ovvero capire se si vengono a creare differenze strutturali, tra le reti di challenge positive, o non pericolose, e quelle di challenge negative, o pericolose, durante tutta la fase di espansione delle challenge stesse.

4.2 Reti delle challenge

Dopo aver definito la semantica delle reti, è necessario progettare un algoritmo di supporto che, a partire dal dataset di riferimento, permetta di generare i grafi relativi alle singole challenge in

analisi. Per fare ciò, la *working directory* è stata ampliata con due file (Figura 4.1).

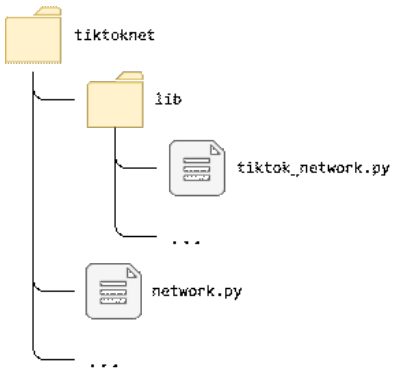

Figura 4.1: Struttura aggiornata della working directory. *Fonte: autoprodotta*

Il file "network.py", analogamente a quanto accade con il file "main.py" analizzato nel capitolo precedente, centralizza ed esegue il calcolo delle reti delle challenge attraverso il metodo graphCalculation, definito nel file "tiktok_network.py", il quale, oltre ad altre importanti strutture dati, restituisce un oggetto di tipo DiGraph, definito nella libreria NetworkX, che rappresenta un grafo orientato.

Il grafo in output alla funzione graphCalculation può essere sia mostrato sullo schermo, tramite libreria Matplotlib, sia salvato in formato .gexf ("*Graph Exchange XML Format*"), un linguaggio di *markup* utilizzato per la descrizione di reti complesse, e dei loro metadati associati, compatibile con il software *Gephi*, il quale permette di studiare la rete in modo più approfondito, oppure di ottenere visualizzazioni più accattivanti rispetto a quelle ottenibili con Matplotlib. L'algoritmo per la generazione della rete funziona in due fasi riassumibili nel seguente modo:

1. filtraggio dei dati originali secondo le specifiche del chiamante;
2. dichiarazione delle strutture fondamentali della rete, quali nodi ed archi, definizione del *layout* della rete ed instanziazione del grafo diretto.

Come prima cosa, infatti, è necessario rimuovere i nodi isolati, i quali non danno contributo informativo alla ricerca. Dopo aver caricato in memoria il dataset relativo ad una challenge, viene effettuato un filtraggio sulle righe che presentano valore nullo sui campi calcolati "likedBy_*", in quanto esse rappresentano video che non sono piaciuti a nessuno degli autori coinvolti nella ricerca, e l'utente che ha caricato il video in questione, a meno che non abbia caricato anche un secondo video relativo alla stessa challenge che sia piaciuto ad almeno uno degli altri autori, non presenterebbe alcun collegamento in ingresso. Oltre a ciò, il parametro opzionale di tipo booleano "remAutoLikes" permette di indicare se le righe del dataset che corrispondono ad utenti che interagiscono con loro stessi, mettendo "mi piace" ad un loro stesso video, debbano essere rimosse. Nella teoria dei grafi, tale relazione si traduce in un "cappio", ovvero un arco in cui gli estremi coincidono; poiché un utente che interagisce con se stesso è poco utile allo studio dell'espansione del grafo, di default il parametro è impostato come "*vero*", ed i cappi vengono rimossi.

Il tempo è un parametro fondamentale nella ricerca; perché è di fondamentale interesse lo studio dell'evoluzione della rete generata a partire dai video di una challenge. Per effettuare al meglio questo studio, l'algoritmo accetta due parametri, lifespanCond e intervals, che permettono di effettuare un campionamento del grafo generale rispetto ad uno specifico arco temporale. In particolare, quei parametri permettono di indicare dei valori percentuali di *lifespan*, ovvero il tempo di vita della challenge, al fine di estrarre la rete risultante definita per quei valori. Il *lifespan* del trend viene calcolato sulla base del campo "createTime" del dataset, e cioè il timestamp di caricamento

del video in questione. L'intervallo del tempo di vita del trend, quindi, è definito per differenza tra la data e l'ora di caricamento dell'ultimo video e del primo video caricato; dividendo il valore ottenuto per 100 si ottiene la durata percentuale del trend. Il parametro `lifespanCond` permette di definire una maschera che campiona il grafo sull'intervallo $[t\%, 100\%]$, mentre il parametro `intervals` consente di definire anche l'estremo superiore $[t_{min}\%, t_{max}\%]$.

Terminata la fase di filtraggio, è necessario individuare all'interno del dataset le strutture fondamentali che costituiscono il grafo. A tal fine, si effettua una scansione sul dataset filtrato, e, per ogni riga, i valori dei campi `author_id` e `likedBy_id` vengono prima inseriti in un insieme dei nodi, e successivamente come valori, rispettivamente, delle chiavi `target` e `source` di un dizionario, a sua volta inserito in un insieme degli archi. L'arco così definito permette di essere orientato secondo la direzione della relazione di interazione tra i due utenti. Inoltre, si definisce un'altra struttura dati, denominata `labels`, che ospita il valore dei *nickname* degli utenti, in modo tale che questi possano essere mostrati a schermo per rendere il grafo più leggibile, come nell'esempio in Figura 4.2.

```
nodes = {"68640117433567390061",
         "60685466300104233501"}

edges = {"source": "60685466300104233501",
         "target": "68640117433567390061"}

labels = {"68640117433567390061": "john45",
          "60685466300104233501": "nick1" }
```

Figura 4.2: Esempio di definizione di un grafo su dati fittizi. *Fonte: autoprodotta*

La funzione `graphCalculation` riceve come parametro anche `colorCriteria`, responsabile della dichiarazione del *layout* del grafo in essa; vengono inizializzate due strutture dati che rappresentano, nodo per nodo, le coordinate x ed y di posizione ed il colore. Se `colorCriteria` corrisponde a `"createTime"`, l'unica stringa attualmente compatibile, il *layout* del grafo rappresenta la distribuzione dei nodi nel tempo; i nodi che corrispondono ad autori che hanno caricato video meno recentemente appaiono neri e centrati rispetto agli assi del piano cartesiano; quelli che hanno caricato video meno recentemente appaiono, invece, distanziati rispetto all'origine degli assi, posizionati con un angolo casuale, e con una colorazione in scala di grigio tendente verso colori sempre più chiari; i nodi per cui non è possibile individuare una data di pubblicazione, cioè quelli che hanno interagito mettendo "mi piace" al video di un utente ed hanno caricato a loro volta un video inerente alla challenge in questione che, però, non ha ricevuto interazioni dagli utenti della ricerca, appaiono con una colorazione tendente al giallo. L'autore che ha caricato il video originale della challenge viene rappresentato con il colore rosso.

Infine, questa funzione restituisce in output diverse strutture dati, tra cui il grafo, il *layout*, il dataset originale e filtrato, ed alcune statistiche fondamentali sulla struttura della rete, utili in prima battuta per individuare differenze fondamentali tra i grafi relativi a challenge positive e quelli relativi a challenge negative.

4.2.1 Le reti delle challenge positive

Come prima attività, quindi, è stata utilizzata la funzione `graphCalculation` per generare le reti relative alle challenge positive scelte, senza partizioni su base temporale. Di seguito sono mostrati i grafi risultanti dall'elaborazione; come visto, il loro *layout* evidenzia, challenge per challenge, la distribuzione dei video nel tempo:

- ***#bussitchallenge***: la rete è mostrata in Figura 4.3.

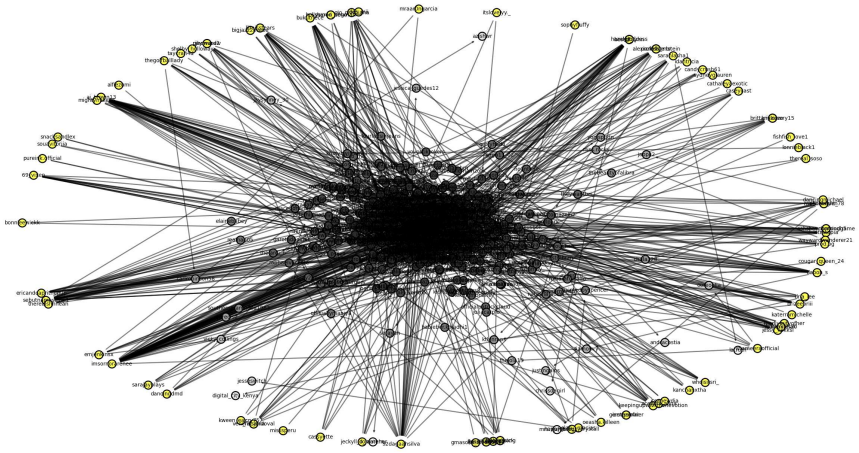

Figura 4.3: Grafo di ***#bussitchallenge***. *Fonte: autoprodotta*

Numero di nodi	618
Numero di archi	708
Grado medio dei nodi	1.146
Coefficiente di clustering medio	0.0047
Densità della rete	0.0019

- ***#copinesdancechallenge***: la rete è mostrata in Figura 4.4.

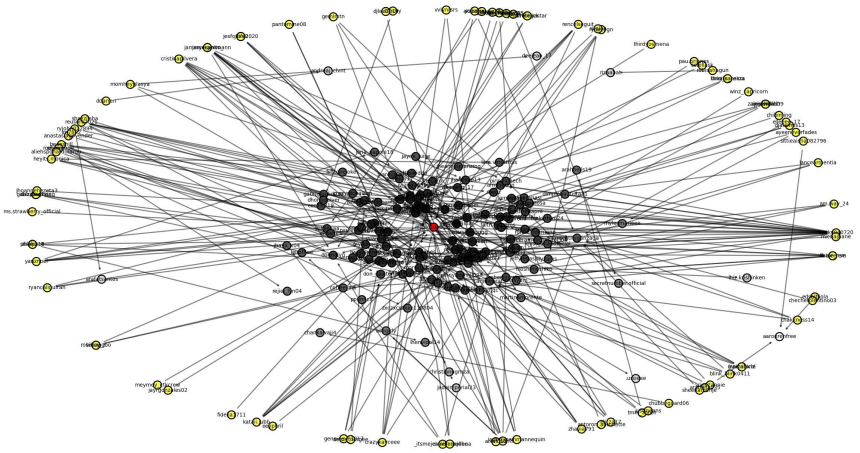

Figura 4.4: Grafo di ***#copinesdancechallenge***. *Fonte: autoprodotta*

Numero di nodi	237
Numero di archi	226
Grado medio dei nodi	0.9536
Coefficiente di clustering medio	0
Densità della rete	0.004

- ***#emojichallenge***: la rete è mostrata in Figura 4.5.

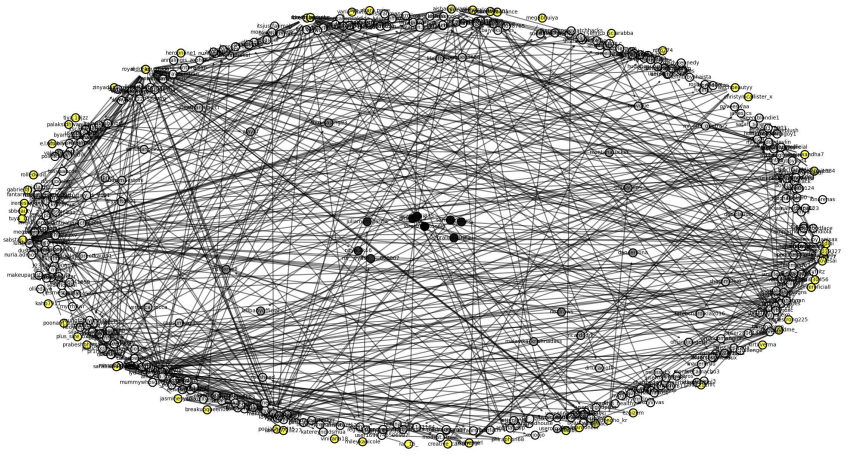

Figura 4.5: Grafo di ***#emojichallenge***. *Fonte: autoprodotta*

Numero di nodi	440
Numero di archi	498
Grado medio dei nodi	1.132
Coefficiente di clustering medio	0.0053
Densità della rete	0.0026

- ***#itookanap***: la rete è mostrata in Figura 4.6.

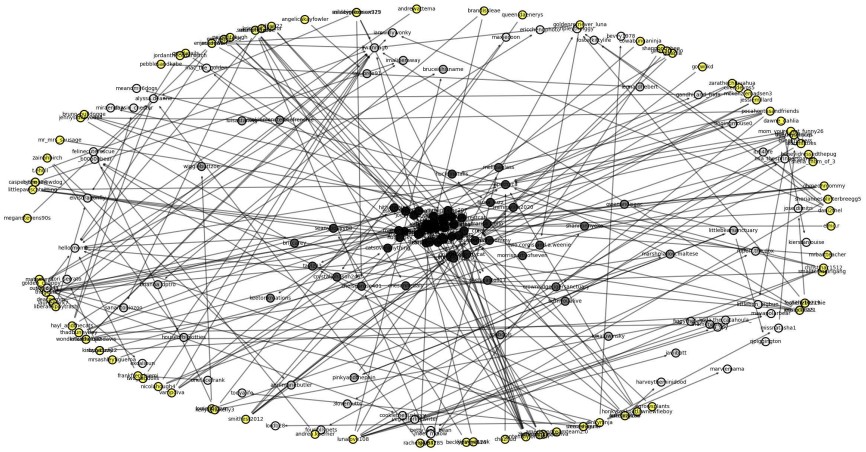

Figura 4.6: Grafo di ***#itookanap***. *Fonte: autoprodotta*

Numero di nodi	219
Numero di archi	201
Grado medio dei nodi	0.9178
Coefficiente di clustering medio	0
Densità della rete	0.0042

- ***#colpiditesta***: la rete è mostrata in Figura 4.7.

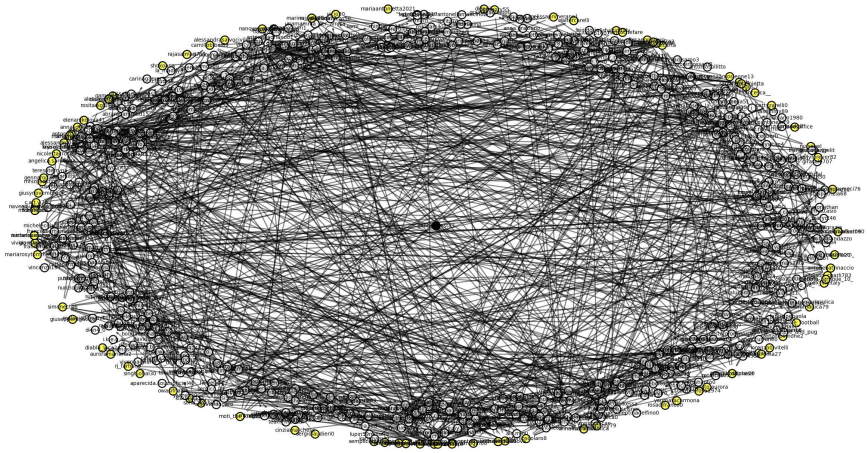

Figura 4.7: Grafo di ***#colpiditesta***. *Fonte: autoprodotta*

Numero di nodi	691
Numero di archi	843
Grado medio dei nodi	1.22
Coefficiente di clustering medio	0.0015
Densità della rete	0.0018

- ***#boredinthehouse***: la rete è mostrata in Figura 4.8.

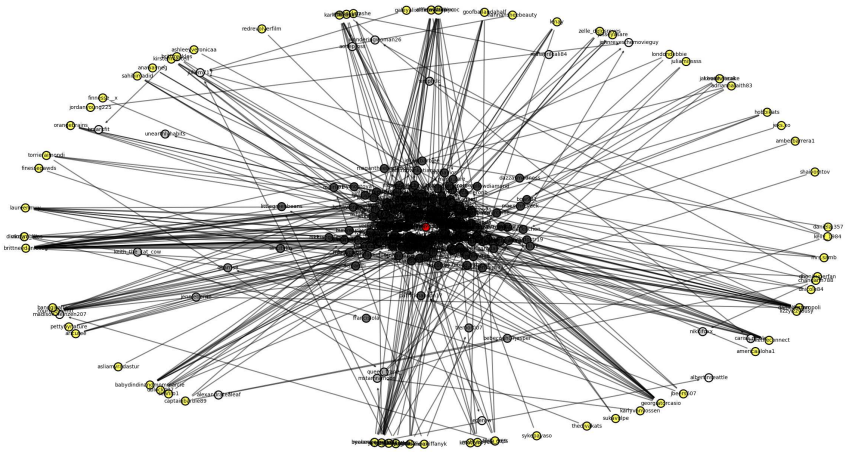

Figura 4.8: Grafo di ***#boredinthehouse***. *Fonte: autoprodotta*

Numero di nodi	306
Numero di archi	309
Grado medio dei nodi	1.01
Coefficiente di clustering medio	0.0018
Densità della rete	0.0033

- ***#plankchallenge***: la rete è mostrata in Figura 4.9.

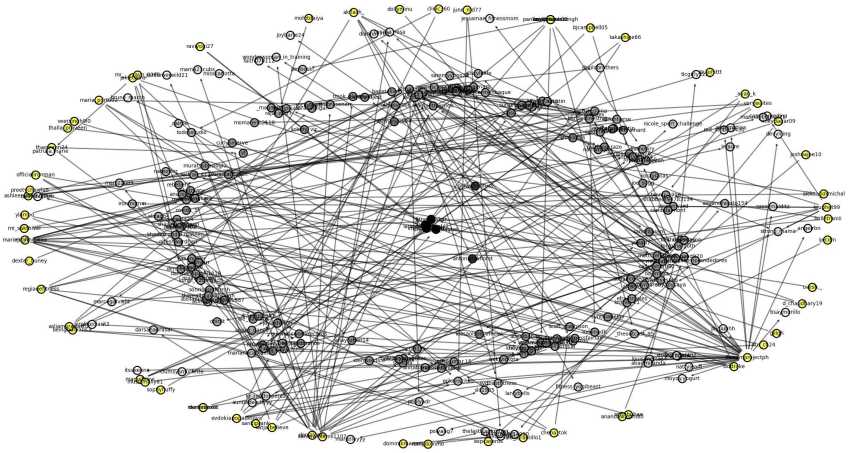

Figura 4.9: Grafo di ***#plankchallenge***. *Fonte: autoprodotta*

Numero di nodi	271
Numero di archi	266
Grado medio dei nodi	0.9815
Coefficiente di clustering medio	0.0079
Densità della rete	0.0036

A titolo esemplificativo, in Figura 4.10 è riportata una visualizzazione alternativa di ***#itookanap***, creata tramite il software *Gephi*.

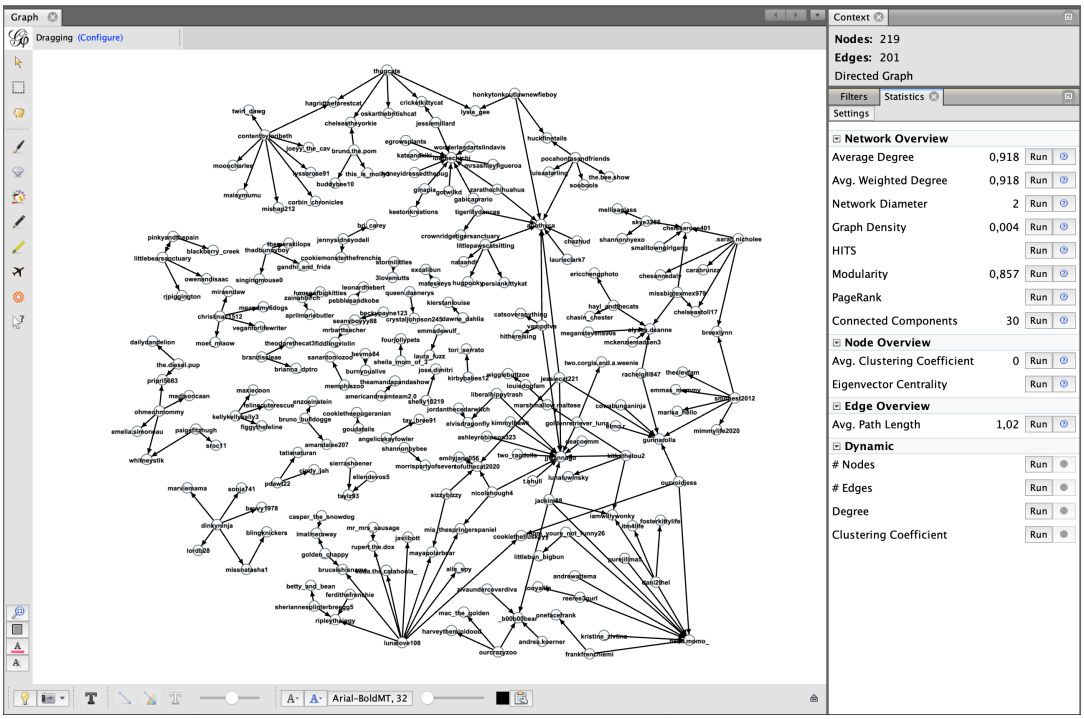

Figura 4.10: Visualizzazione tramite *Gephi* di ***#itookanap***. *Fonte: autoprodotta*

Il software permette di ottenere delle visualizzazioni molto chiare delle reti; il suo pannello principale è costituito da una tavolozza, dove viene renderizzato il grafo, che permette di interagire con gli elementi, spostando e personalizzando, con infinite possibilità, i nodi e gli archi. Un pannello situato a destra dello schermo mostra alcune metriche fondamentali e permette di effettuare dei filtraggi al fine di ottenere dei campioni di rete con caratteristiche simili tra loro.

4.2.2 Le reti delle challenge negative

La stessa operazione di generazione dei grafi è stata riproposta per i dati delle challenge negative:

- ***#silhouettechallenge***: la rete è mostrata in Figura 4.11.

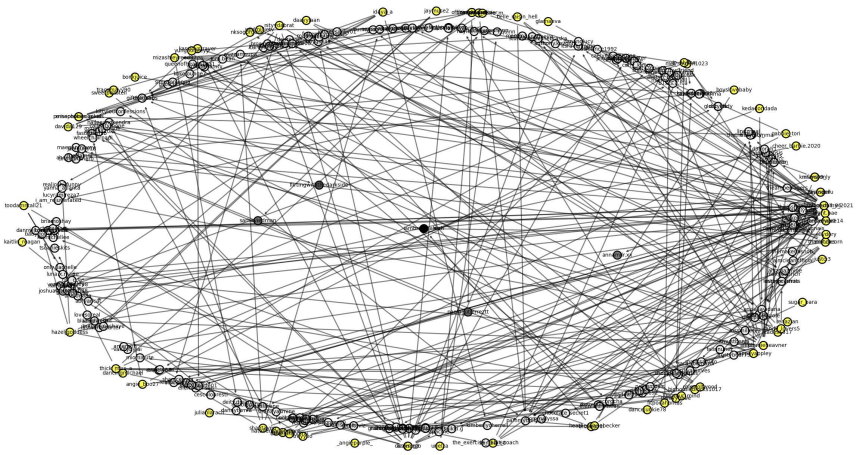

Figura 4.11: Grafo di ***#silhouettechallenge***. *Fonte: autoprodotta*

Numero di nodi	262
Numero di archi	259
Grado medio dei nodi	0.9811
Coefficiente di clustering medio	0
Densità della rete	0.0037

- ***#bugsbunnychallenge***: la rete è mostrata in Figura 4.12.

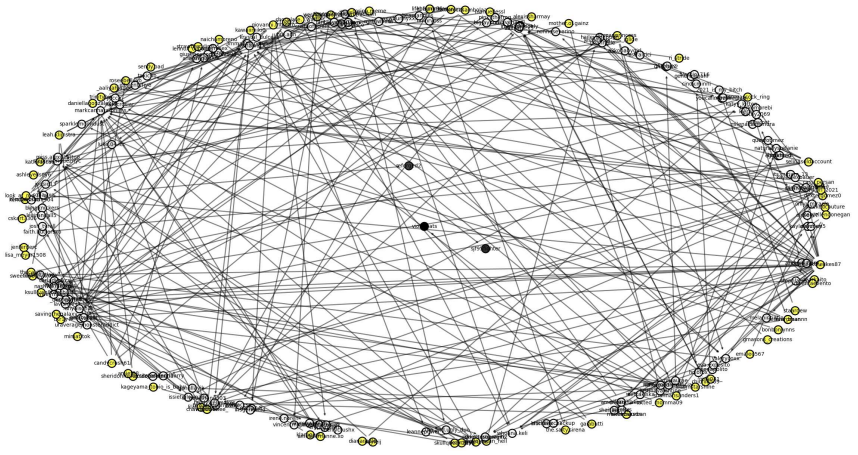

Figura 4.12: Grafo di ***#bugsbunnychallenge***. *Fonte: autoprodotta*

Numero di nodi	212
Numero di archi	239
Grado medio dei nodi	1.1273
Coefficiente di clustering medio	0
Densità della rete	0.0053

- ***#strippatiktok***: la rete è mostrata in Figura 4.13.

Figura 4.13: Grafo di ***#strippatiktok***. *Fonte: autoprodotta*

Numero di nodi	297
Numero di archi	519
Grado medio dei nodi	1.7475
Coefficiente di clustering medio	0.0025
Densità della rete	0.0059

- ***#firewroks***: la rete è mostrata in Figura 4.14.

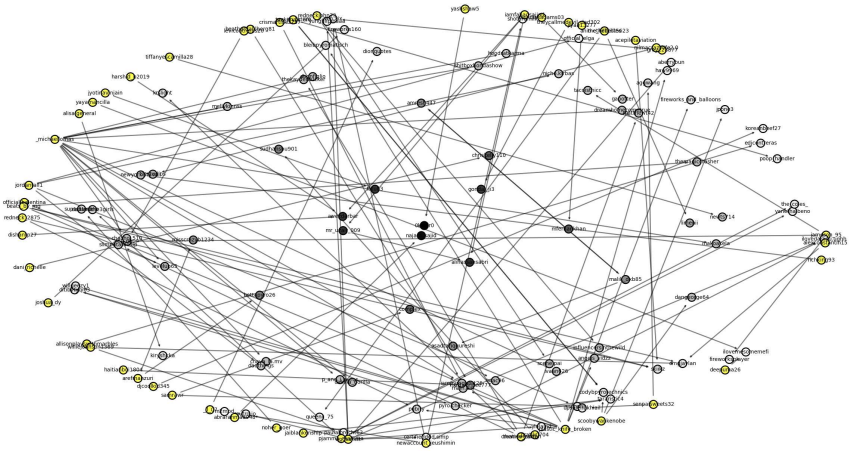

Figura 4.14: Grafo di ***#firewroks***. *Fonte: autoprodotta*

Numero di nodi	141
Numero di archi	111
Grado medio dei nodi	0.7872
Coefficiente di clustering medio	0.0083
Densità della rete	0.0056

- ***#fightchallenge***: la rete è mostrata in Figura 4.15.

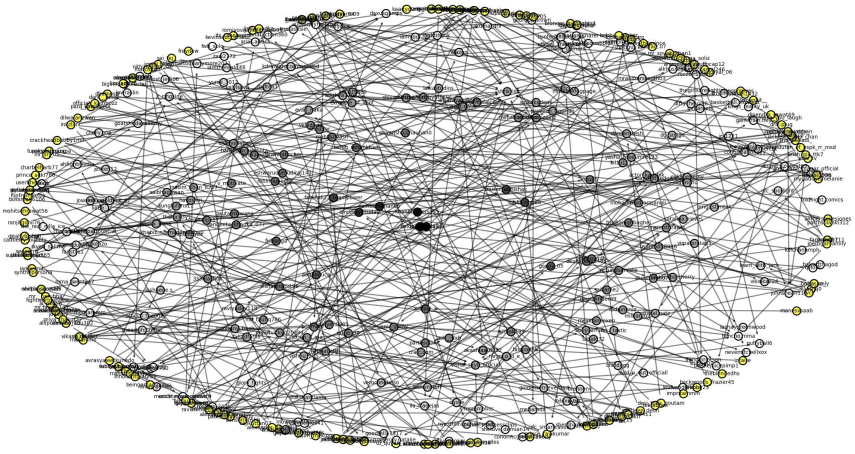

Figura 4.15: Grafo di ***#fightchallenge***. *Fonte: autoprodotta*

Numero di nodi	409
Numero di archi	339
Grado medio dei nodi	0.8288
Coefficiente di clustering medio	0.0009
Densità della rete	0.002

- ***#sugarbaby***: la rete è mostrata in Figura 4.16.

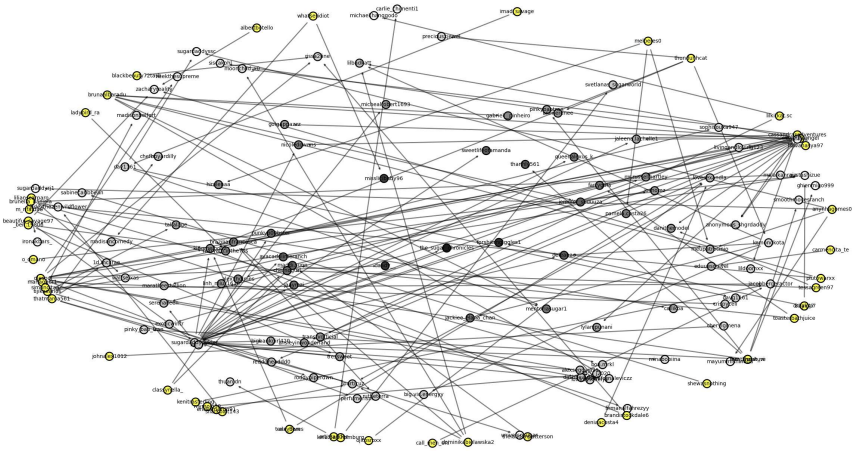

Figura 4.16: Grafo di ***#sugarbaby***. *Fonte: autoprodotta*

Numero di nodi	151
Numero di archi	143
Grado medio dei nodi	0.947
Coefficiente di clustering medio	0.0036
Densità della rete	0.0063

- ***#updownchallenge***: la rete è mostrata in Figura 4.17.

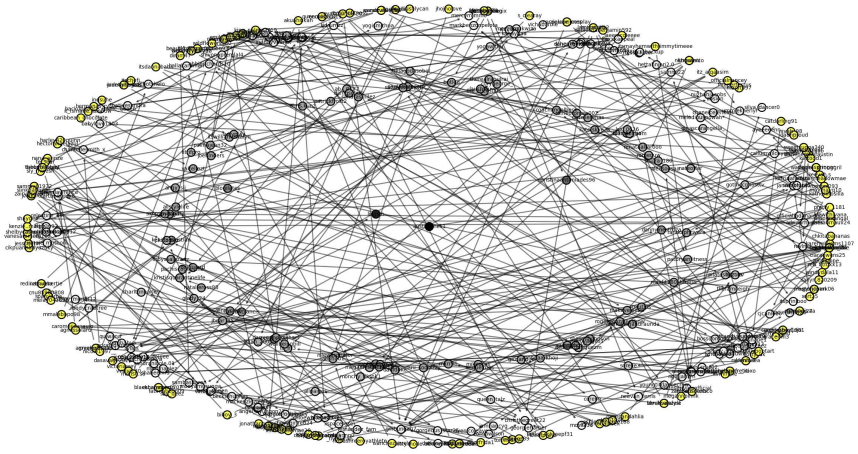

Figura 4.17: Grafo di ***#updownchallenge***. *Fonte: autoprodotta*

Numero di nodi	375
Numero di archi	294
Grado medio dei nodi	0.784
Coefficiente di clustering medio	0.0112
Densità della rete	0.0021

4.3 Analisi sui dati delle reti

Oltre alle informazioni sulle metriche di base delle reti, è utile effettuare un'analisi aggiuntiva sui dati utilizzati per la generazione delle stesse. Infatti, il dataset di riferimento è costituito da molte informazioni che riguardano la natura dei video e degli autori che hanno partecipato alle challenge.

Per questo motivo è stato ralizzato uno *script* Python, identificato dal nome `pre_analysis.py`, che fa uso della libreria `Pandas` al fine di calcolare alcune caratteristiche sui dati aggregati di challenge non pericolose e challenge pericolose. In questa fase, le metriche più interessanti sono relative a tre argomenti principali:

- *Informazioni sui video delle challenge*: queste metriche valutano le informazioni relative ai video delle challenge e misurano il numero di interazioni, in termini assoluti, di "mi piace", commenti e condivisioni. Esse misurano, anche, la durata media dei video caricati ed il numero medio per challenge di diversi audio utilizzati. Intuitivamente, un trend non caratterizzato da un audio particolare presenta un numero maggiore di audio distinti utilizzati rispetto ad un trend che, invece, è definito per una musica specifica.

- *Informazioni sugli autori dei video*: le metriche relative agli utenti che hanno partecipato alla challenge misurano, in media, come questi utilizzano il Social Network. Si analizzano, quindi, ancora una volta in termini assoluti, la quantità di video che gli utenti hanno caricato sulla piattaforma TikTok, il numero totale di "mi piace" che hanno ricevuto ed assegnato, il numero di seguaci e di utenti seguiti. Gli *influencer* sono tipicamente individuabili per il loro elevato numero di seguaci e di interazioni ricevute.
- *Informazioni generali sul lifespan delle challenge*: queste metriche esplicitano, in termini percentuali, la distribuzione del numero di nodi sul tempo di vita della challenge.

L'esecuzione dell'algoritmo ha restituito i risultati mostrati nelle Tabelle 4.15, 4.16 e 4.17.

Tabella 4.15: *Tabella delle metriche sui video delle challenge.*

Metrica	**Challenge positive**	**Challenge negative**
Durata media dei video (in secondi)	20.13	20.01
Numero medio di suoni distinti usati nei video	162.57	113.57
Numero medio di "mi piace" ricevuti	155,516.29	211,189.94
Numero medio di commenti ricevuti	1,542.51	2,227.82
Numero medio di condivisioni ricevute	5,156.11	5,567.92
Numero medio di visualizzazioni ottenute	1,350,133.96	1,700,197.71

Tabella 4.16: *Tabella delle metriche sugli autori delle challenge.*

Metrica	**Challenge positive**	**Challenge negative**
Numero medio di "mi piace" totali assegnati dall'autore	14,402.29	11,411.39
Numero medio di "mi piace" totali ricevuti dall'autore	7,004,933.38	10,082,467.23
Numero medio di seguaci dell'autore	385,590.42	398,168.93
Numero medio di utenti seguiti dall'autore	1,164.56	678.63
Numero medio di video pubblicati dall'autore	351.52	273.27

Tabella 4.17: *Tabella delle metriche sul lifespan delle challenge.*

Metrica	**Challenge positive**	**Challenge negative**
Durata media di una challenge (in giorni)	405	620.43
Percentuale di video caricati nel primo 5% di *lifespan* della challenge	2.58%	0.77%
Percentuale di video caricati nel primo 25% di *lifespan* della challenge	27.59%	2.63%
Percentuale di video caricati nel primo 50% di *lifespan* della challenge	34.32%	7.97%
Percentuale di video caricati nel primo 75% di *lifespan* della challenge	44.19%	23.34%

4.3.1 Caratteristiche individuate

Dai risultati estratti si nota come, mediamente, le challenge negative siano caratterizzate da un *lifespan* più longevo; una motivazione dietro a ciò risiede nel fatto che i video delle challenge negative tendono, solitamente, ad attrarre maggiormente l'attenzione delle persone, soprattutto se si parla degli utenti più giovani, i quali interagiscono con i video maggiormente e per un periodo di tempo molto lungo. Infatti, per quanto riguarda la quantità di interazioni, intese in termini di numero di "mi piace", di condivisioni, di commenti e di visualizzazioni ricevute, i video di challenge negative registrano valori sensibilmente superiori rispetto a quelli ottenuti da challenge positive.

Questo aspetto si ritrova anche affrontando un'analisi quantitativa relativamente agli utenti coinvolti nella ricerca. Si nota immediatamente come gli autori di video relativi a challenge negative registrino un numero nettamente maggiore di "mi piace" ricevuti, con, addirittura, un numero di seguaci medio inferiore. Questo significa che molti utenti, oltre a quelli che seguono gli autori in questione, ed in maniera maggiore rispetto alla controparte dei video delle challenge positive, hanno interagito con il suddetto video pur non seguendo l'autore.

Per quanto riguarda, invece, le metriche di base relative alle reti, estratte al passaggio precedente, le differenze non sono marcate. Nelle Tabelle 4.18 e 4.19 sono riportati, per comodità, i valori ottenuti.

Tabella 4.18: *Tabella riassuntiva delle metriche delle reti di challenge positive.*

Challenge	**Numero di nodi**	**Numero di archi**	**Grado medio dei nodi**	**Coefficiente di clustering medio**	**Densità della rete**
#bussitchallenge	618	708	1.146	0.0047	0.0019
#copinesdancechallenge	237	226	0.9536	0	0.004
#emojichallenge	440	498	1.132	0.0053	0.0026
#itookanap	219	201	0.9178	0	0.0042
#colpiditesta	691	843	1.22	0.0015	0.0018
#boredinthehouse	306	309	1.01	0.0018	0.0033
#plankchallenge	271	266	0.9815	0.0079	0.0036
Valore medio	**397**	**436**	**1.0515**	**0.0049**	**0.0031**

Tabella 4.19: *Tabella riassuntiva delle metriche delle reti di challenge negative.*

Challenge	**Numero di nodi**	**Numero di archi**	**Grado medio dei nodi**	**Coefficiente di clustering medio**	**Densità della rete**
#silhouettechallenge	262	259	0.9811	0	0.0037
#bugsbunnychallenge	212	239	1.1279	0	0.0053
#strippatiktok	297	519	1.7475	0.0025	0.0059
#firewroks	141	111	0.7872	0.0083	0.0056
#fightchallenge	409	339	0.8288	0.0009	0.002
#sugarbaby	151	143	0.947	0.0036	0.0063
#updownchallenge	375	294	0.784	0.0112	0.0021
Valore medio	**264**	**272**	**1.029**	**0.0038**	**0.0044**

Fatta eccezione per la dimensione delle reti, che indica un contributo sensibilmente maggiore e più definito in termini di video di challenge positive rispetto alle challenge negative, le altre metriche presentano valori pressoché simili.

5. Definizione di intervalli e feature

5.1 Algoritmo di definizione degli intervalli

Il tema principale della ricerca riguarda l'analisi del comportamento delle challenge nel tempo e l'individuazione di differenze, e somiglianze, tra l'evoluzione di reti che rappresentano le challenge positive, identificate dai sette trend già discussi, e quelle negative, altresì già individuate. Finora sono stati analizzati i dati statici, ottenuti mediante gli algoritmi di acquisizione dei dati, e si sono identificate le prime importanti differenze tra le due diverse classi di challenge di TikTok. Il passo successivo è quello di analizzare la loro dinamicità nel tempo, per studiarne le variazioni. Analizzare l'andamento in un certo intervallo di tempo può consentire di determinare degli istanti di vita comuni tra i trend, come, ad esempio, la fase di creazione, il momento in cui la challenge diventa virale o quello in cui la sua popolarità si riduce, fino ad annullarsi. Ognuno di questi momenti può avere caratteristiche differenti che contraddistinguono una challenge dalle altre.

In Tabella 4.17, mostrata nel capitolo precedente, si nota che l'espansione delle challenge positive presenta, attorno alle fasi centrali dell'intero ciclo di vita dei trend stessi, un tasso di crescita medio molto basso, dopo, però, aver registrato un'importante partecipazione da parte degli utenti nel primo quartile del *lifespan*. Le challenge negative mantengono, invece, riguardo il tasso di crescita sul numero di video che contribuiscono alle challenge, valori percentuali molto bassi, almeno fino al terzo quartile del tempo di vita; la partecipazione di nuovi utenti alla challenge, però, esplode con una tendenza esponenziale nel periodo immediatamente precedente alla fase di acquisizione dei dati. Tale evento, che corrisponde, approssimativamente, all'ultimo 7% dell'intero ciclo di vita dei trend, si presenta anche per quanto riguarda l'evoluzione delle challenge positive; ciò è dovuto principalmente dal fatto che l'API di TikTok, quando interrogata, fornisce il maggior numero possibile di informazioni riguardo i video caricati nei momenti che precedono l'interrogazione, oltre agli altri risultati meno recenti, seppur questi abbiano registrato poche interazioni. Per questo motivo, l'esplosione di video caricati nelle ultime fasi percentuali del ciclo di vita è un elemento comune a tutti i trend. In ogni caso, la differenza principale che si nota è che, come si vede chiaramente in Figura 5.1, le challenge positive, in media, presentano sempre un maggior numero di video in tutti gli istanti del ciclo di vita, rispetto a quelle negative. Questo è un elemento fondamentale che consente di definire il punto di partenza delle prossime analisi.

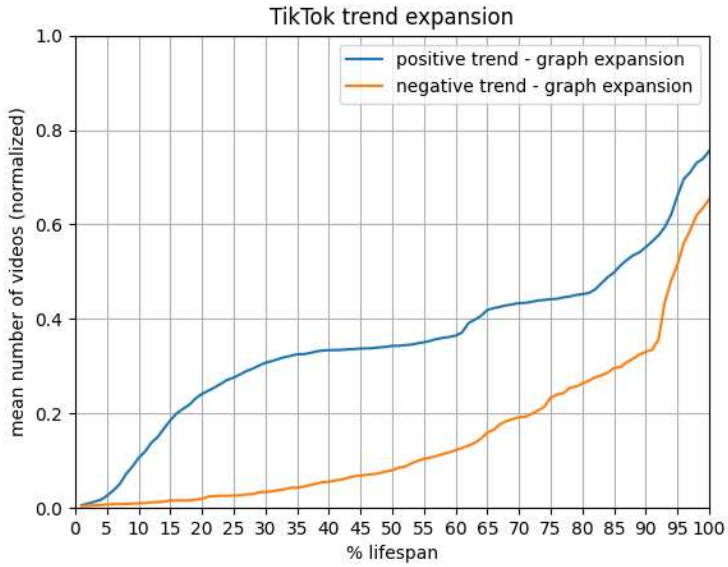

Figura 5.1: Andamento di challenge a confronto. *Fonte: autoprodotta*

Individuata la principale differenza nell'evoluzione di challenge appartenenti alle due diverse classi, è necessario approfondire la conoscenza dei dati relativi ai singoli trend, in ottica di evoluzione temporale delle reti; l'idea è quella di definire un approccio matematico con cui analizzare l'espansione, in termini di numero di nodi, del grafo di una challenge al fine di individuare degli intervalli con cui suddividere il suo *lifespan*.

In particolar modo, per sintetizzare i diversi momenti di vita delle challenge, si può ricorrere allo strumento matematico della "*derivata*", che, applicata al modello che descrive la partecipazione degli utenti di TikTok alla challenge in esame, restituisce informazioni sulla sua variazione; la derivata $f'(x)$ di una funzione $f(x)$ fornisce l'andamento di $f(x)$, descrivendo quando questa cresce o descresce. Dato $X = \{x_1, ..., x_N\}$ i punti in cui $f'(x) = 0$, per il teorema di Fermat, sono quelli di massimo e minimo relativi della funzione $f(x)$. Questi punti possono essere utili per delimitare gli intervalli temporali di interesse, con cui suddividere le diverse challenge.

In Figura 5.2 è riportato il grafico di espansione di una challenge esemplificativa.

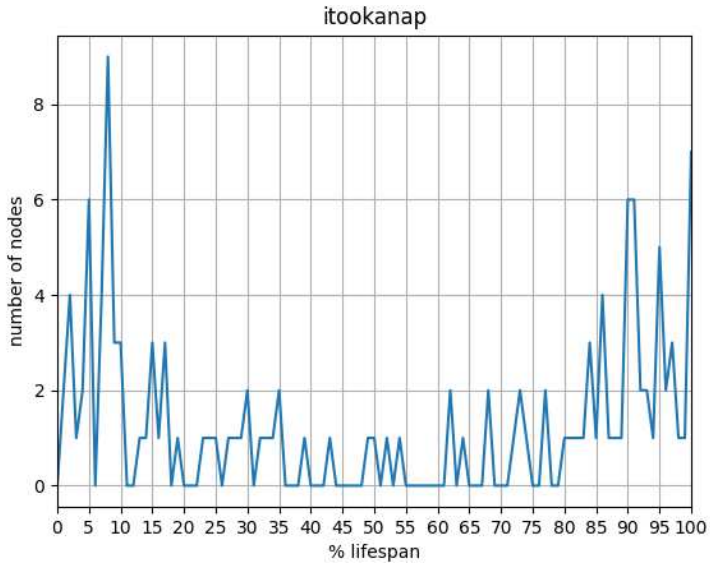

Figura 5.2: Numero di nodi di ***#itookanap*** nel tempo. *Fonte: autoprodotta*

Come si può facilmente intuire, non ha grande significato analitico calcolare la derivata su una funzione che rappresenta questo insieme di punti; visivamente si nota che i punti di massimo e minimo relativi sono molti, e ottenere troppi intervalli difficilmente caratterizzabili non è utile al fine della ricerca, poiché non si avrebbero congruenze tra intervalli di challenge diverse.

Per questo motivo, si è sviluppato un algoritmo in linguaggio Python, identificato dal nome `lifespan_analysis.py`, che opera nel seguente modo:

1. Carica in memoria il dataset utilizzato per la creazione del grafo di una challenge, tramite il metodo `graphCalculation`, già discusso in precedenza.
2. Effettua un'aggregazione dei dati, in particolare delle informazioni riguardanti il numero di nodi per intervallo percentuale del *lifespan* della challenge, riducendo la granularità ad $\frac{1}{5}$ dell'originale; il numero di intervalli percentuali diminuisce da 100 a 20. In questo modo si riducono a priori il numero di possibili massimi e minimi relativi.
3. Genera la funzione da derivare instanziando un oggetto di tipo `UnivariateSpline`, dichiarato nella libreria `Scipy`, il quale rappresenta l'interpolazione dell'insieme di 20 punti ottenuti al passaggio precedente. Passando da un insieme di valori discreti ad un insieme di valori definiti in un intervallo continuo in $\mathbb{R}$, si può calcolare la derivata della suddetta funzione, utilizzando il metodo `derivative`, fornito da `Scipy`.
4. Calcola le radici della derivata utilizzando il metodo `roots`, anch'esso distribuito dalla libreria `Scipy`, ottenendo una lista di valori che corrispondono ai massimi e minimi relativi della funzione che interpola i punti che descrivono l'espansione del grafo della challenge.

5.1.1 Risultati ottenuti

I risultati ottenuti dall'elaborazione dell'algoritmo sono molto interessanti. Di seguito si riportano i grafici delle funzioni interpolate, generate per ogni challenge, con i relativi punti estremanti. Gli estremi degli intervalli sono approssimati sul valore percentuale più prossimo:

- ***#bussitchallenge***: si possono individuare 4 intervalli; essi sono definiti dall'insieme $\{]0, 15],]15, 60],]60, 90],]90, 100]\}$. Il grafico è mostrato in Figura 5.3.

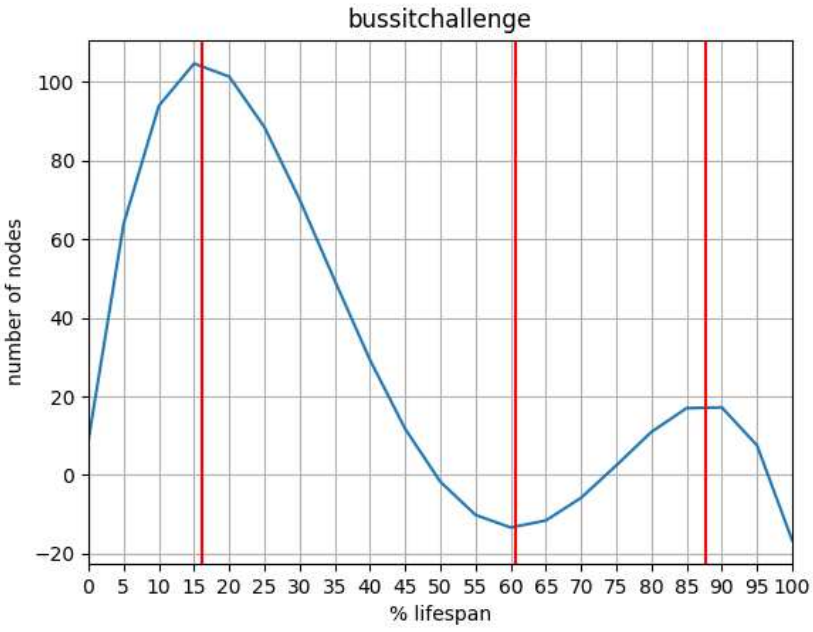

Figura 5.3: Challenge positiva: ***#bussitchallenge***. *Fonte: autoprodotta*

- ***#copinesdancechallenge***: si possono individuare 4 intervalli; essi sono definiti dall'insieme di valori $\{]0, 20],]20, 65],]65, 90],]90, 100]\}$. Il grafico è mostrato in Figura 5.4.

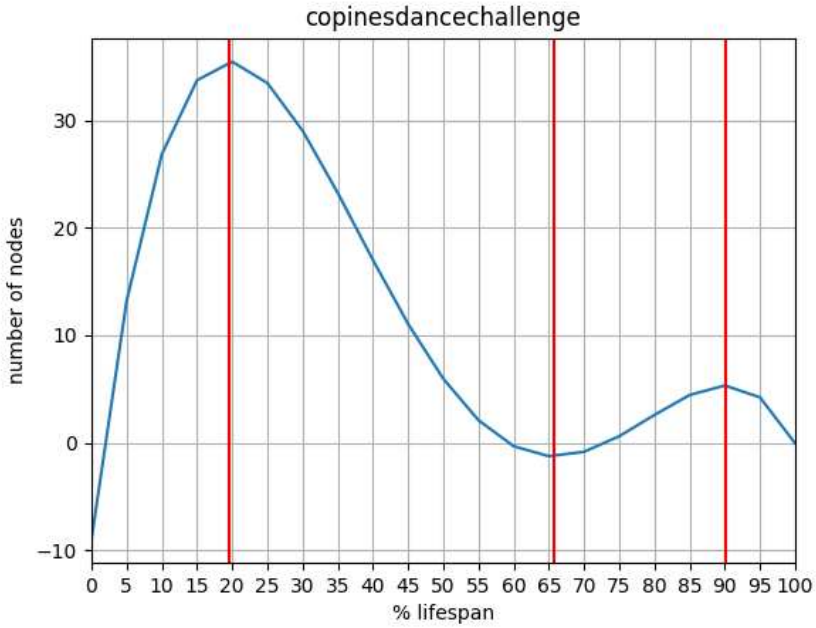

Figura 5.4: Andamento ed intervalli di ***#copinesdancechallenge***. *Fonte: autoprodotta*

- ***#emojichallenge***: si possono individuare 4 intervalli; essi sono definiti dall'insieme $\{]0, 10],]10, 35],]35, 65],]65, 100]\}$. Il grafico è mostrato in Figura 5.5.

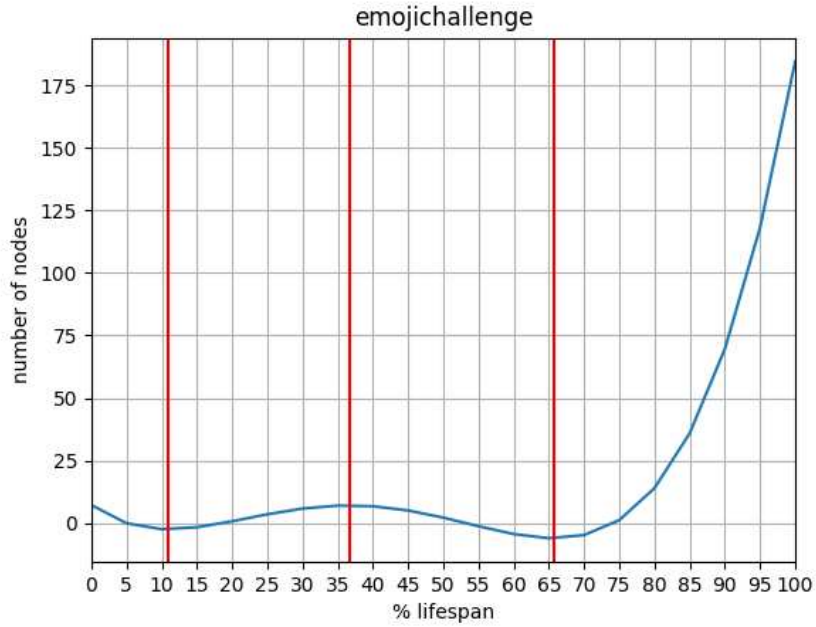

Figura 5.5: Andamento ed intervalli di ***#emojichallenge***. *Fonte: autoprodotta*

- ***#itookanap***: si possono individuare 3 intervalli; essi sono definiti dall'insieme $\{]0, 10],]10, 55],]55, 100]\}$. Il grafico è mostrato in Figura 5.6.

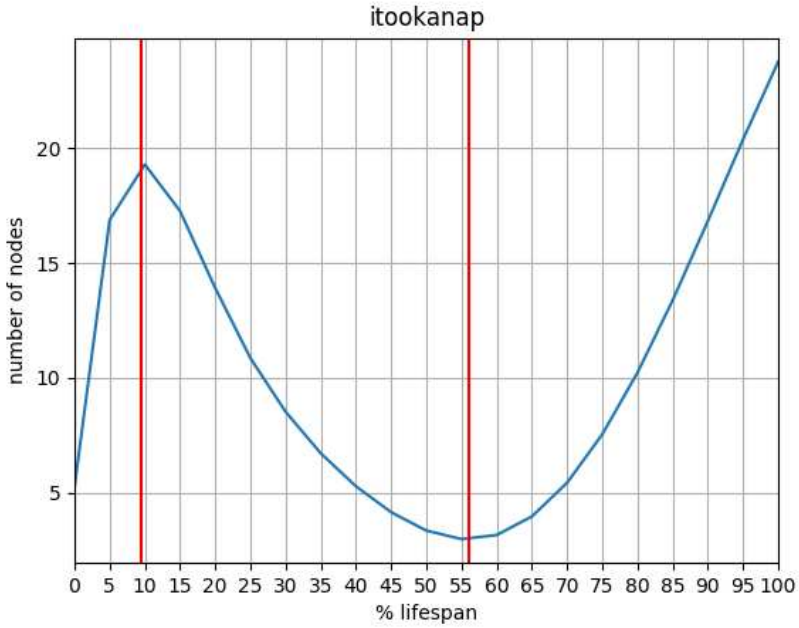

Figura 5.6: Andamento ed intervalli di ***#itookanap***. *Fonte: autoprodotta*

- ***#colpiditesta***: si possono individuare 4 intervalli; essi sono definiti dall'insieme $\{]0, 15],]15, 45],]45, 95],]95, 100]\}$. Il grafico è mostrato in Figura 5.7.

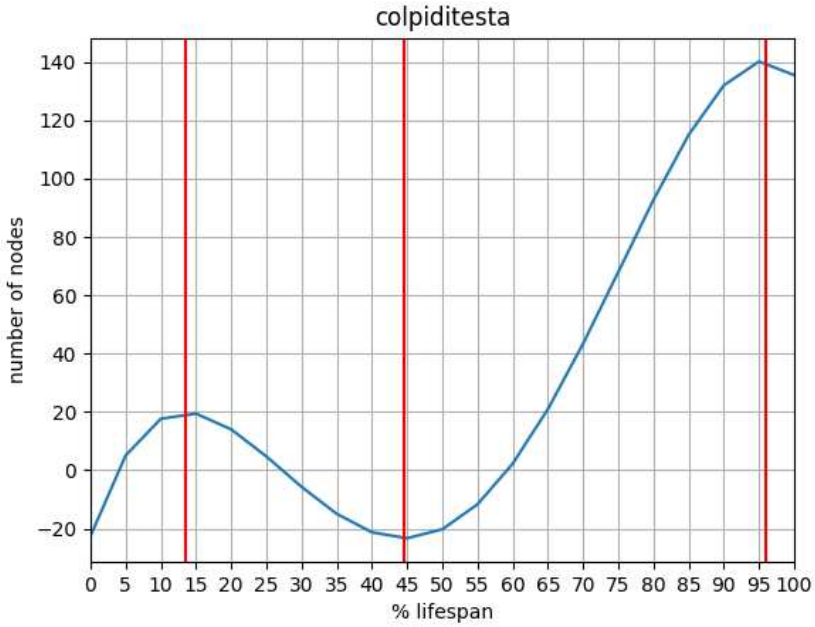

Figura 5.7: Andamento ed intervalli di ***#colpiditesta***. *Fonte: autoprodotta*

- ***#boredinthehouse***: si possono individuare 4 intervalli; essi sono definiti dall'insieme $\{]0, 15],]15, 60],]60, 90],]90, 100]\}$. Il grafico è mostrato in Figura 5.8.

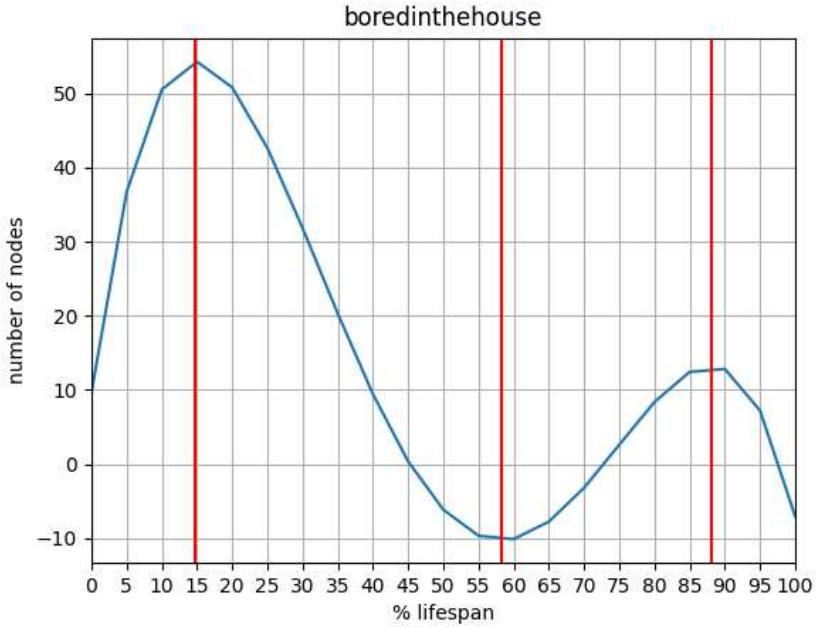

Figura 5.8: Andamento ed intervalli di ***#boredinthehouse***. *Fonte: autoprodotta*

- ***#plankchallenge***: si possono individuare 3 intervalli; essi sono definiti dall'insieme $\{]0, 15],]15, 70],]70, 100]\}$. Il grafico è mostrato in Figura 5.9.

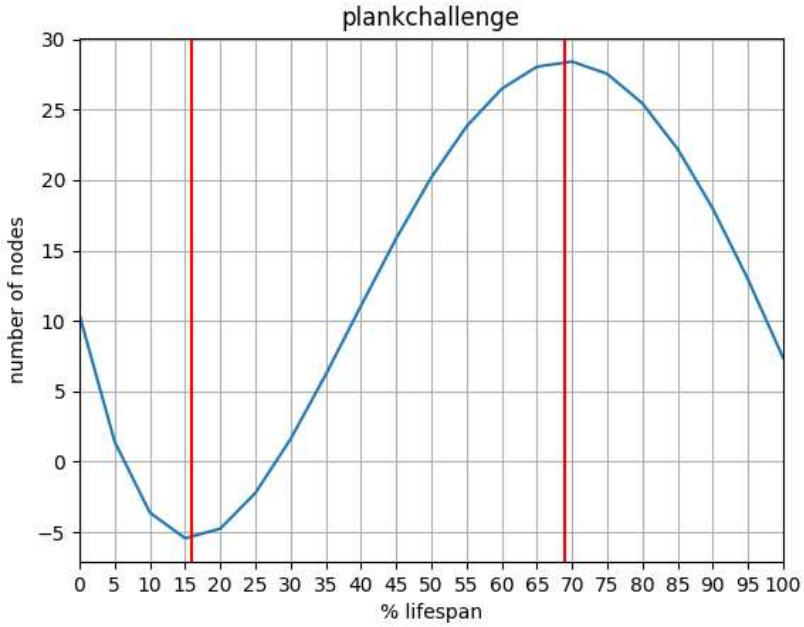

Figura 5.9: Andamento ed intervalli di ***#plankchallenge***. *Fonte: autoprodotta*

- ***#silhouettechallenge***: si possono individuare 3 intervalli; essi sono definiti dall'insieme $\{]0, 20],]20, 55],]55, 100]\}$. Il grafico è mostrato in Figura 5.10.

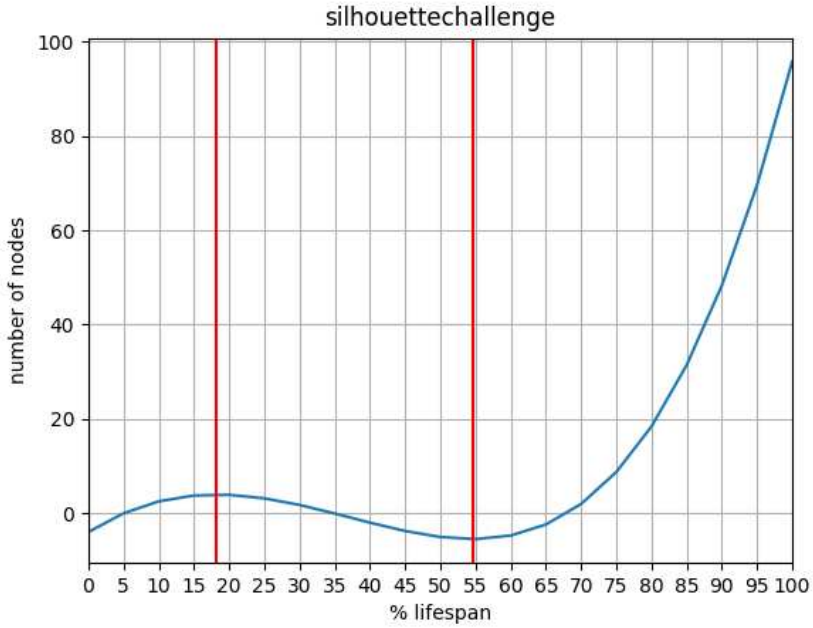

Figura 5.10: Andamento ed intervalli di ***#silhouettechallenge***. *Fonte: autoprodotta*

- ***#bugsbunnychallenge***: si possono individuare 4 intervalli; essi sono definiti dall'insieme $\{]0, 10],]10, 40],]40, 70],]70, 100]\}$. Il grafico è mostrato in Figura 5.11.

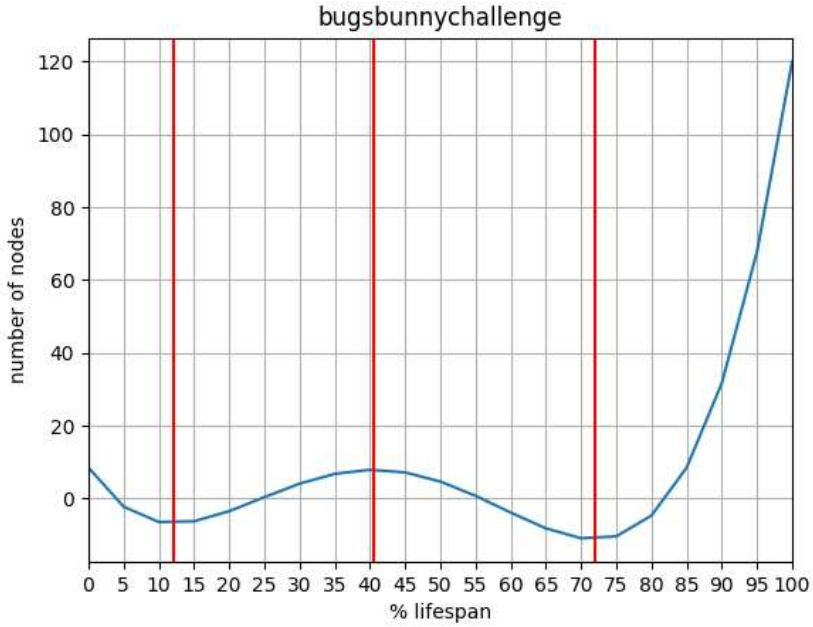

Figura 5.11: Andamento ed intervalli di ***#bugsbunnychallenge***. *Fonte: autoprodotta*

- ***#strippatiktok***: si possono individuare 3 intervalli; essi sono definiti dall'insieme $\{]0, 5],]5, 25],]25, 100]\}$. Il grafico è mostrato in Figura 5.12.

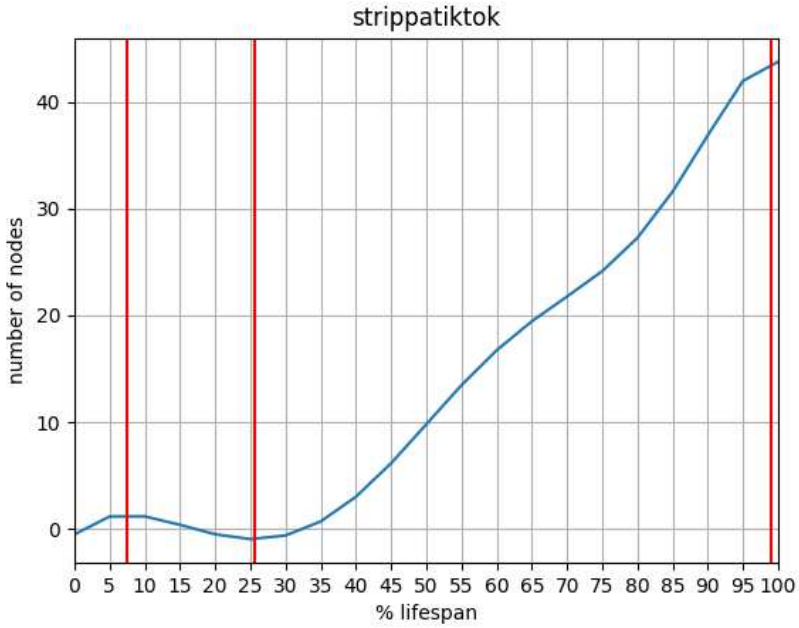

Figura 5.12: Andamento ed intervalli di ***#strippatiktok***. *Fonte: autoprodotta*

- ***#firewroks***: si possono individuare 3 intervalli; essi sono definiti dall'insieme $\{]0, 15],]15, 25],]25, 100]\}$. Il grafico è mostrato in Figura 5.13.

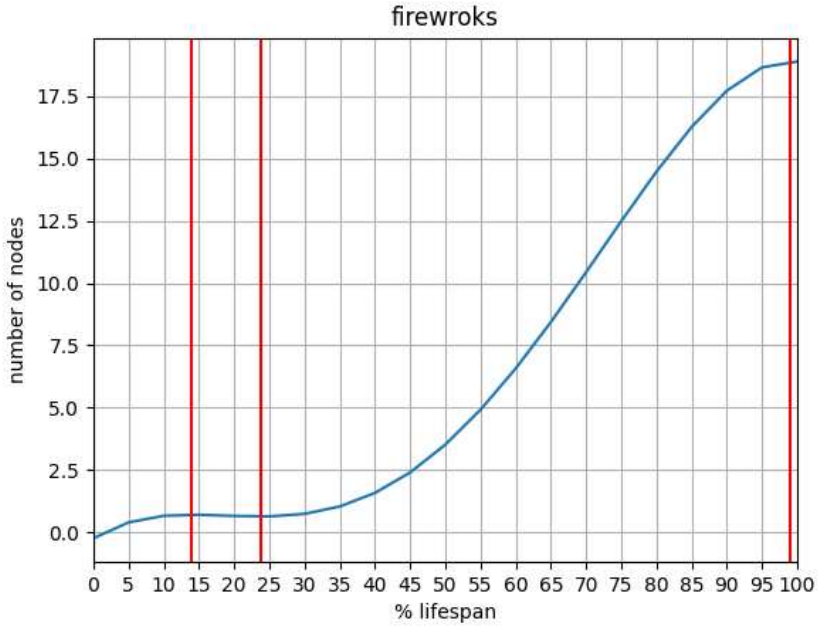

Figura 5.13: Andamento ed intervalli di ***#firewroks***. *Fonte: autoprodotta*

- ***#fightchallenge***: si possono individuare 4 intervalli; essi sono definiti dall'insieme $\{]0, 10],]10, 60],]60, 75],]75, 100]\}$. Il grafico è mostrato in Figura 5.14.

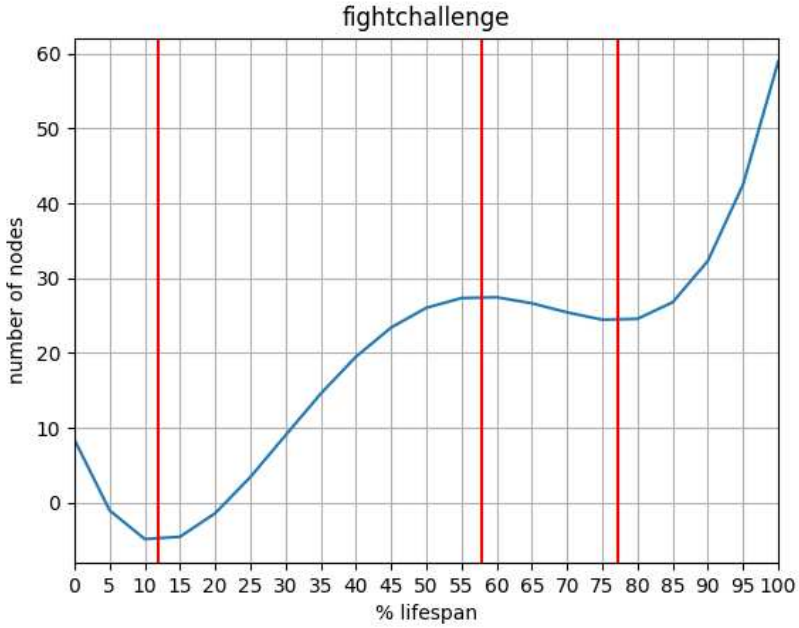

Figura 5.14: Andamento ed intervalli di ***#fightchallenge***. *Fonte: autoprodotta*

- ***#sugarbaby***: si possono individuare 4 intervalli; essi sono definiti dall'insieme $\{]0, 10],]10, 35],]35, 60],]60, 100]\}$. Il grafico è mostrato in Figura 5.15.

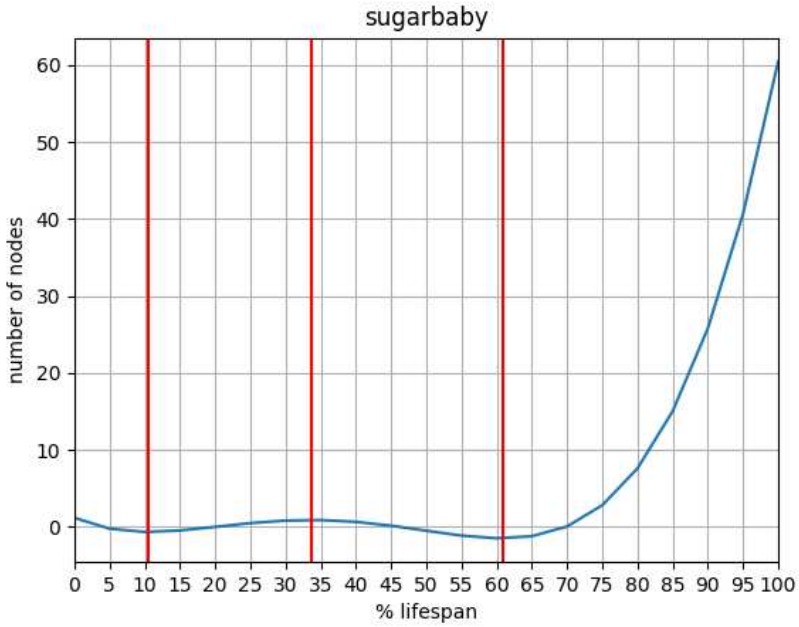

Figura 5.15: Andamento ed intervalli di ***#sugarbaby***. *Fonte: autoprodotta*

- ***#updownchallenge***: si possono individuare 3 intervalli; essi sono definiti dall'insieme $\{]0, 10],]10, 30],]30, 100]\}$. Il grafico è mostrato in Figura 5.16.

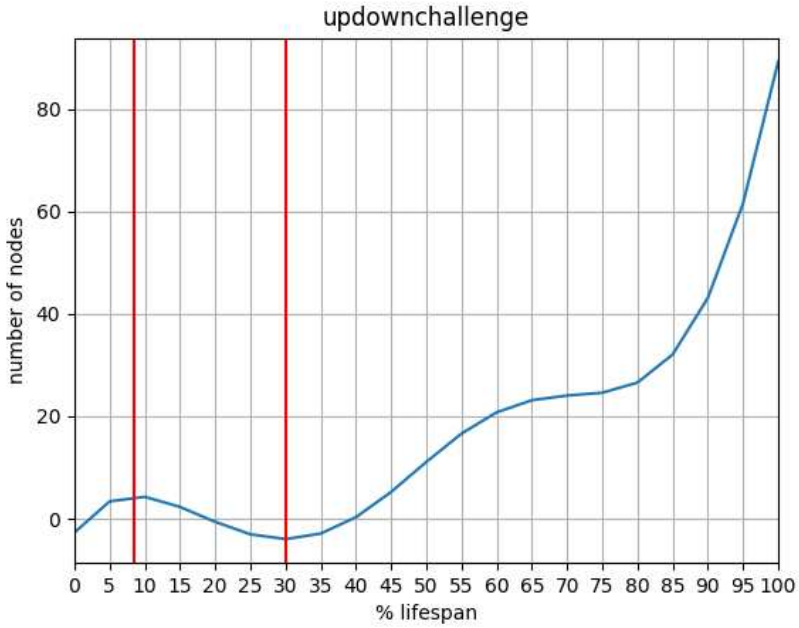

Figura 5.16: Andamento ed intervalli di ***#updownchallenge***. *Fonte: autoprodotta*

Intuitivamente, si nota come gli intervalli siano consecutivi ed ordinabili, secondo il valore percentuale del *lifespan* della challenge; ne consegue che essi possono essere rinominati in base alla propria posizione. Ad esempio, l'intervallo $]0, 10]$ corrisponde all'intervallo numero 1 per la challenge ***#itookanap***, $]10, 55]$ corrisponde all'intervallo numero 2 e così via. In Tabella 5.1 sono riportati i risultati ottenuti, in formato sintetico.

Tabella 5.1: *Tabella riassuntiva sui risultati dell'analisi.*

Challenge	**Intervallo**	**Tendenza**
#bussitchallenge	1 - $]0, 15]$	+
	2 - $]15, 60]$	-
	3 - $]60, 90]$	+
	4 - $]90, 100]$	-
#copinesdancechallenge	1 - $]0, 20]$	+
	2 - $]20, 65]$	-
	3 - $]65, 90]$	+
	4 - $]90, 100]$	-
#emojichallenge	1 - $]0, 10]$	-
	2 - $]10, 35]$	+
	3 - $]35, 65]$	-
	4 - $]65, 100]$	+
	1 - $]0, 10]$	+

#itookanap	2 -]10, 55] 3 -]55, 100]	- +
#colpiditesta	1 -]0, 15] 2 -]15, 45] 3 -]45, 95] 4 -]95, 100]	+ - + -
#boredinthehouse	1 -]0, 15] 2 -]15, 60] 3 -]60, 90] 4 -]90, 100]	+ - + -
#plankchallenge	1 -]0, 15] 2 -]15, 70] 3 -]70, 100]	- + -
#silhouettechallenge	1 -]0, 20] 2 -]20, 55] 3 -]55, 100]	+ - +
#bugsbunnychallenge	1 -]0, 10] 2 -]10, 40] 3 -]40, 70] 4 -]70, 100]	- + - +
#strippatiktok	1 -]0, 5] 2 -]5, 25] 3 -]25, 100]	+ - +
#firewroks	1 -]0, 15] 2 -]15, 25] 3 -]25, 100]	+ - +
#fightchallenge	1 -]0, 10] 2 -]10, 60] 3 -]60, 75] 4 -]75, 100]	- + - +
#sugarbaby	1 -]0, 10] 2 -]10, 35] 3 -]35, 60] 4 -]60, 100]	- + - +
#updownchallenge	1 -]0, 10] 2 -]10, 30] 3 -]30, 100]	+ - +

5.2 Individuazione delle feature

Una volta individuati tutti gli intervalli di riferimento, è necessario definire alcune metriche, o *feature*, che li descrivano in modo opportuno. Questa operazione è fondamentale al fine del

proseguimento della ricerca poiché le *feature* sono gli elementi che permettono di classificare gli intervalli, individuando somiglianze o discrepanze tra le diverse entità in challenge di tipologie distinte.

L'obiettivo della ricerca consiste nell'identificare differenze strutturali che si vengono a creare nell'espansione del grafo della challenge; per questo motivo si sono scelte *feature* che riguardano esclusivamente metriche, sia di base che calcolate, della rete stessa. Chiaramente, questi elementi sono valutati per tutti i singoli intervalli, in modo da poter associare ad un momento della vita della challenge i dati che lo caratterizzano.
L'elenco delle *feature* prese in considerazione è il seguente:

- numero di nodi della rete;
- numero di archi della rete;
- densità della rete;
- degree centrality media della rete;
- eigenvector centrality media della rete;
- numero di componenti connesse della rete;
- numero di nodi coinvolti nella componente connessa con dimensione massima;
- grado medio dei nodi della rete, di cui viene calcolata anche la deviazione standard;
- valore del PageRank medio della rete;
- closeness centrality media della rete;
- coefficiente di clustering medio della rete;
- raggio della componente connessa con il massimo numero di nodi, per ciascuna rete;
- diametro della componente connessa con il massimo numero di nodi, per ciascuna rete;
- indegree centrality media della rete;
- percentuale di nodi che appartengono alla componente connessa di dimensione massima, rispetto al numero totale di nodi della rete;
- eccentricità media della rete, ovvero la media, calcolata per ogni nodo del grafo, del massimo grado tra tutti i nodi collegati al nodo corrente;
- lunghezza media di tutti i percorsi della rete;
- numero di ego-network della rete; nell'algoritmo di calcolo delle metriche, per convenzione, una ego-network è definita da almeno 4 nodi e 3 archi, che collegano una stella di tre nodi ad un punto centrale;
- numero di nodi nella ego-network con il massimo numero di nodi, per ciascuna rete;
- numero medio di nodi che compongono le ego-network della rete.

Dopo aver calcolato i valori delle *feature*, si è generata la matrice di correlazione delle metriche selezionate. Questa matrice è una tabella di dimensione $n \times n$, ove n è il numero delle variabili indipendenti che, per ogni cella, riporta il valore di correlazione tra la coppia di metriche che la individuano. La correlazione ρ_{xy} è una misura statistica che valuta la relazione tra due variabili x ed y, tale che a ciascun valore della prima corrisponda un valore della seconda, seguendo una certa regolarità. Essa si definisce come:

$$-1 \leq \rho_{xy} = \frac{\delta_{xy}}{\delta_x \delta_y} = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_x y^2}} \leq 1$$

Per analizzare più nel dettaglio la matrice di correlazione, possiamo suddividerla in tre parti: una diagonale principale e due triangoli, uno uno superiore ed uno inferiore. Le correlazioni sulla diagonale principale, ovvero quella che va dalla cella in alto a sinistra a quella in basso a destra, sono tutte uguali al valore 1. Ciò indica, semplicemente, che la correlazione di una variabile con se stessa è sempre massima. Spostando l'attenzione alle altre coppie di variabili, l'indice di correlazione appare sempre due volte. I valori presenti nel triangolo inferiore della matrice sono, infatti, gli stessi riportati nelle celle presenti nel triangolo superiore della stessa matrice.

Questo perché l'indice di correlazione è una misura statistica di tipo simmetrico che non tiene conto dell'ordine con cui le variabili vengono inserite all'interno della formula.

In Figura 5.17 viene riportata la matrice di correlazione risultante dalle *feature* considerate per ogni intervallo.

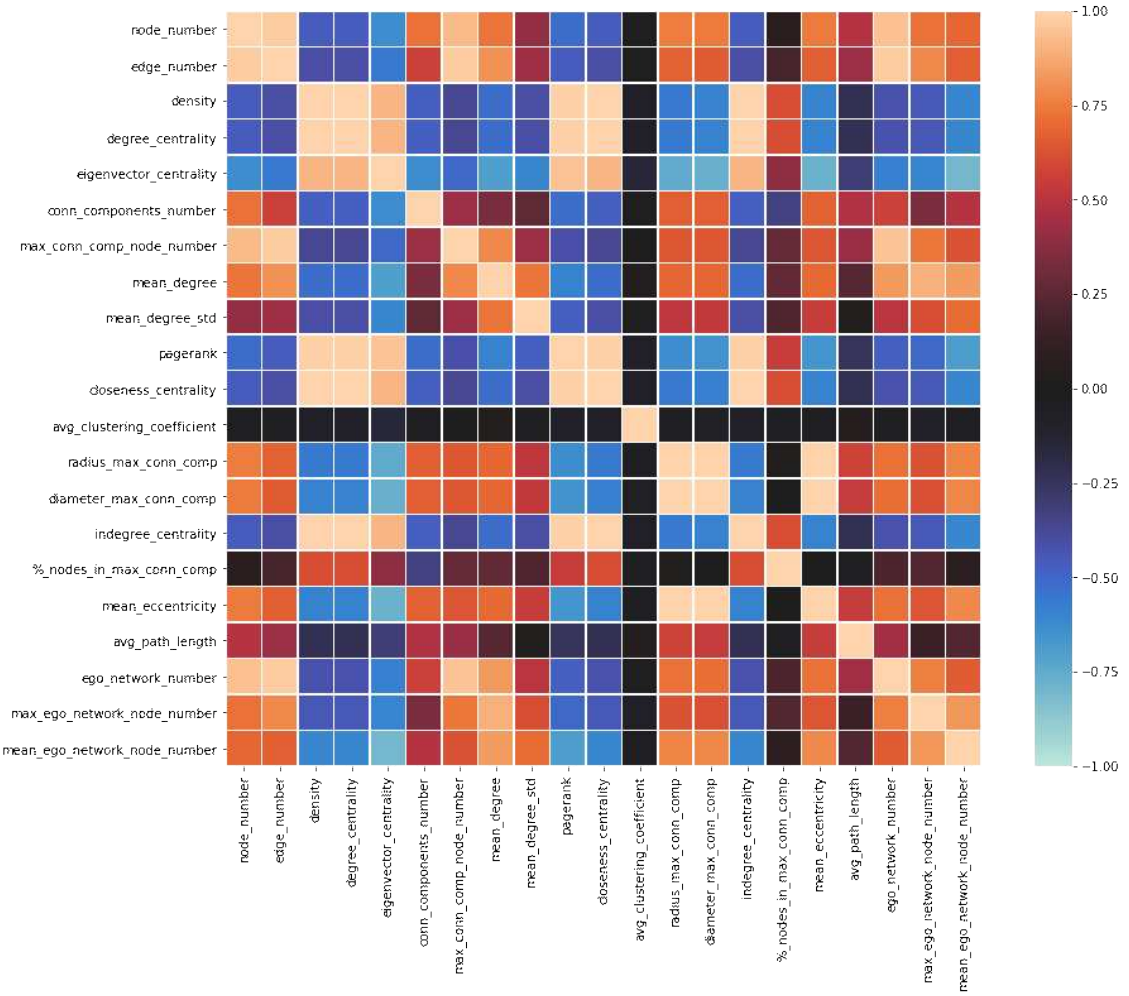

Figura 5.17: Matrice di correlazione. *Fonte: autoprodotta*

Come si vede dalla *heatmap*, gran parte delle coppie di *feature* individuate sono strettamente correlate, con valori molto prossimi a -1 o 1. Effettuare un'analisi considerando entrambi gli elementi di una coppia di metriche molto correlate tra di loro è una procedura scorretta, in quanto i valori di un elemento che si correla ad un altro è deducibile dal primo, e la *feature* non aggiunge informazioni all'analisi.

5.2.1 Riduzione dello spazio delle feature

Prima di procedere, quindi, con l'analisi, è opportuno analizzare le coppie di *feature* al fine di individuare quelle che sono maggiormente correlate. Successivamente, è necessario individuare la combinazione di metriche che minimizza i valori nelle celle della matrice di correlazione, costruendo, di fatto, un sottospazio delle *feature*.

Per questo motivo, dapprima si sono rimosse le *features* che si correlano con tutte le altre per valori $-1 \leq \rho \leq -0.75$, oppure $0.75 \leq \rho \leq 1$. Successivamente si sono analizzate le singole coppie di metriche e si è individuato il sottoinsieme delle stesse che soddisfa i requisiti prestabiliti. La

procedura è stata effettuata senza ausilio di script poiché, oltre alla riduzione della correlazione, si volevano mantenere le *feature* più significative per uno studio in ottica di Social Network Analysis. La matrice di correlazione derivante da questa operazione è mostrata in Figura 5.18.

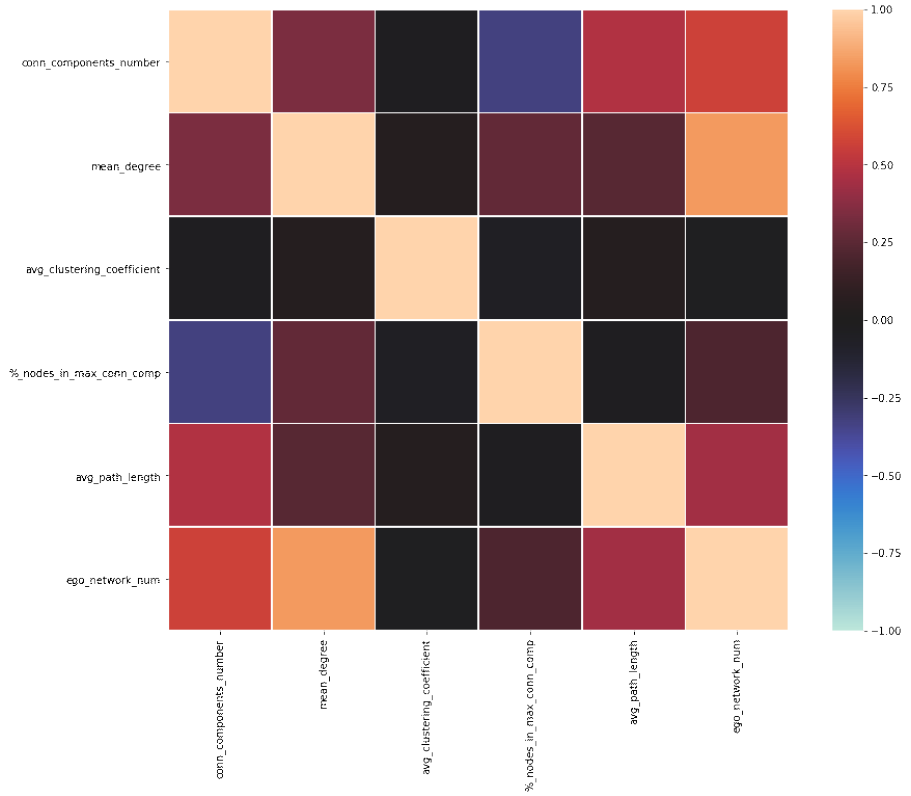

Figura 5.18: Matrice di correlazione ridotta. *Fonte: autoprodotta*

Le metriche selezionate, quindi, sono:

- numero di componenti connesse della rete, per intervallo;
- valore medio del grado dei nodi della rete, per intervallo;
- coefficiente di clustering medio della rete, per intervallo;
- percentuale di nodi contenuti nella massima componente connessa, rispetto al numero di nodi della rete, per intervallo;
- betweenness centrality;
- lunghezza media di tutti i percorsi della rete, per intervallo;
- numero di ego-network individuate nella rete, per intervallo.

6. Estrazione dei pattern

6.1 Analisi degli intervalli

In un'ottica di classificazione degli intervalli, e relativa descrizione qualitativa del momento di vita della challenge, è necessario sviluppare un processo che, a partire dai dati ottenuti analizzando le *feature* individuate al passo precedente, consenta di raggruppare in maniera automatica intervalli che, complessivamente, presentano dati simili.

L'idea, infatti, è quella di utilizzare un algoritmo di clustering per raggruppare in maniera automatica, ed in modo non-supervisionato, gli intervalli di tutte le challenge coinvolte nell'analisi che presentano caratteristiche statisticamente simili tra loro. La qualità dei parametri utilizzati per la generazione dei gruppi è di fondamentale importanza, poiché le sequenze di intervalli, elencate in Tabella 5.1, assumono una semantica in base al gruppo in cui gli intervalli che le compongono vengono inseriti. Una cattiva interpretazione dei dati, infatti, può portare a risultati completamente differenti, ed i risultati della ricerca possono essere invalidati.

Dall'analisi delle sequenze, contestualizzata sulla tipologia che caratterizza la challenge di riferimento, si possono individuare uno o più *pattern* che, con le opportune semplificazioni ed astrazioni, caratterizzano le challenge appartenenti ad una tipologia, e le distinguono da quelle relative alla tipologia opposta; le challenge positive possono essere, quindi, caratterizzate da un *pattern* univoco, mentre i trend negativi sono definiti da un secondo *pattern*, che permette di distinguerli dai primi.

Il dataset adottato in questa fase del progetto è lo stesso utilizzato per calcolare la matrice di correlazione mostrata in Figura 5.18; ad ogni intervallo di ogni singola challenge sono riportati tutti i valori assunti dalle *feature* elencate in precedenza.

6.1.1 Principal Component Analysis

I *cluster*, ovvero i raggruppamenti generati tramite un algoritmo di clustering, possono essere rappresentati in uno spazio n-dimensionale, ove n è il numero delle *feature* utilizzate per ottenere i raggruppamenti stessi. Rappresentando il problema con le sette metriche individuate, relative alla struttura delle reti che caratterizzano gli intervalli delle challenge, non è possibile visualizzare su un piano bidimensionale il risultato dell'algoritmo.

Per ovviare a questo problema, si è effettuata l'operazione di *PCA* (*Principal Component Analysis*); questo è un metodo matematico che permette di trovare le direzioni della massima varianza nei dati su n dimensioni, e di proiettarle su un sottospazio con dimensioni inferiori, o uguali, a quello originale. Utilizzando la proiezione matematica, il set di dati originale viene rappresentato da un ridotto numero di variabili, dette componenti principali; questa diminuzione di dimensione permette una maggior interpretabilità dei dati ed, in questo, caso comporta anche il benefcio di ottenere una corretta rappresentazione dei dati su un piano bidimensionale.

Con la PCA, la direzione degli assi di proiezione è indicata dagli autovettori $X_1, \ldots, X_n$ della matrice di correlazione, mentre i corrispondenti autovalori rappresentano l'ammontare della variabilità totale osservata sulle variabili originarie, espressa da ciascuna componente principale. Gli autovettori che presentano autovalori con minor valore portano meno informazioni sulla distribuzione dei dati; viceversa, quelli che presentano autovalori con valore maggiore rappresentano più fedelmente le caratteristiche dei dati originari. Per questo motivo, l'approccio comune è quello di classificare gli autovalori in ordine decrescente di valore, e scegliere i primi k autovettori corrispondenti. Nel caso specifico del progetto, le prime due componenti principali rappresentano il 72% circa dell'informazione totale sulla distribuzione dei dati, come si vede in Figura 6.1.

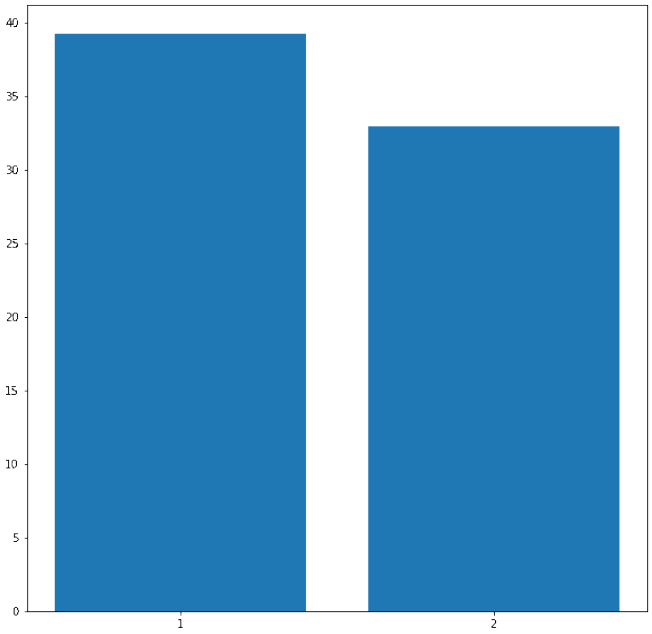

Figura 6.1: Contenuto informativo delle prime due componenti principali, post PCA. *Fonte: autoprodotta*

La figura mostra, infatti, che la varianza delle prime due componenti principali estratte descrive molto bene il dataset originale; non a caso, l'auspicio nell'applicazione della PCA è proprio quello che le varianze campionarie delle prime componenti principali siano di grande valore, mentre quelle delle altre componenti siano abbastanza ridotte da poter considerare trascurabili le corrispondenti componenti principali. Il restante 28% del contenuto informativo del dataset, infatti, è distribuito sulle altre cinque componenti principali, segno che possono essere omesse con relativa tranquillità.

In definitiva, quindi, il nuovo dataset presenta solo due *feature*; esse sono calcolate come combinazione lineare delle metriche originali, e sono rappresentabili come coppia di valori x ed y su un piano cartesiano. Lato pratico, per effettuare l'operazione di Principal Component Analysis, si è realizzato uno script Python che utilizza il metodo `fit_transform`, definito dall'oggetto `PCA`, distribuito dalla libreria `Scikit-Learn`.

6.1.2 Clustering

Come già anticipato, il clustering è una tipologia di algoritmo di apprendimento automatico ("*machine learning*"), che consente di raggruppare oggetti in modo non supervisionato, cioè senza l'utilizzo di esempi etichettati da utilizzare come base per l'apprendimento di un modello. Ogni cluster rappresenta una classe di appartenenza degli elementi. Non avendo a disposizione esempi da cui imparare, il clustering sfrutta delle similarità tra i dati che deve analizzare, i quali possono essere di varia natura ma che, in sostanza, definiscono una distanza euclidea tra i punti del dataset. Gli algoritmi di clustering si distinguono in tre famiglie:

- *Clustering partizionale*: gli algoritmi di clustering partizionale creano una suddivisione degli elementi, minimizzando una funzione di costo $\sum_{j=1}^{k} E(C_j)$, dove k è il numero dei cluster, C_j è il j-esimo cluster ed $E : C \to \mathbb{R}^+$ è la funzione di costo associata al singolo cluster.
- *Clustering gerarchico*: gli algoritmi di tipo gerarchico, invece, costruiscono una gerarchia di partizioni caratterizzate da un numero crescente di gruppi, visualizzabile mediante una rappresentazione ad albero.
- *Clustering density-based*: gli algoritmi di questa famiglia raggruppano gli elementi valutando la densità di punti nello spazio, in un intorno di raggio ε del punto considerato.

Durante lo svolgimento del progetto, si sono utilizzati algoritmi di clustering partizionale; in particolar modo *k-means* ed *EM* (*Expectation - Maximization*). k-means è uno degli algoritmi di clustering più diffuso e più "performante", nonché uno dei più semplici da implementare. Esso si basa sull'individuazione di punti, nello spazio delle *feature*, che mediano le distanze tra tutti i dati appartenenti al cluster associato a ciascuno di essi. Il suo funzionamento è riassumibile in tre fasi principali:

1. Inizialmente vengono scelti, in modo casuale, k punti non coincidenti di riferimento, che appartengono allo spazio delle *feature*.
2. Si calcola la distanza euclidea di ogni elemento del dataset rispetto ad ogni punto di cui al passo 1; l'elemento del dataset viene associato al cluster collegato al punto con minore distanza.
3. Si ricalcola la posizione di ogni punto calcolando la media delle posizioni di tutti gli elementi associati al cluster, e si itera fino a quando la classificazione degli elementi non varia più.

Come detto, k-means è un algoritmo molto semplice da utilizzare; si è adottato, come per l'applicazione della PCA, il metodo `fit_transform`; questa volta, però, questo metodo è definito dall'oggetto `KMeans`, fornito dalla libreria `Scikit-Learn`.

L'algoritmo Expectation - Maximization rappresenta, invece, un'evoluzione di k-means. Esso è un metodo iterativo utilizzato per trovare la massima verosimiglianza locale dei parametri nei modelli statistici, in cui il modello dipende da variabili latenti non osservate. L'iterazione EM alterna l'esecuzione di un passaggio di *Expectation* (E), che crea una funzione per l'aspettativa della verosimiglianza logaritmica, valutata utilizzando la stima corrente per i parametri, e un passaggio di *Maximization* (M), che calcola i parametri massimizzando la probabilità trovata al passaggio precedente. Queste stime dei parametri vengono, quindi, utilizzate per determinare la distribuzione delle variabili latenti nella fase di *Expectation* successiva. Anche in questo caso l'applicazione dell'algoritmo è immediata; si utilizza il metodo `predict` dell'oggetto `GaussianMixture` che implementa l'algoritmo EM, definito, anch'esso, nella libreria `Scikit-Learn`.

Entrambi gli algoritmi adottati richiedono, come parametro di funzionamento, il numero di raggruppamenti da generare, con cui gli oggetti nello spazio delle *feature* sono classificati. In generale, l'operazione di individuazione del numero di cluster, che rappresentano nel miglior modo possibile le entità non è semplice. Si possono, tuttavia, adottare dei metodi statistici, come, ad esempio, il metodo del gomito ("*elbow method*"), che aiutano ad individuare visivamente il giusto numero di cluster da creare. Il metodo itera l'esecuzione di k-means su diversi valori interi e sequenziali di k, ed ogni volta ne calcola la somma delle distanze al quadrato tra ogni media e gli

oggetti del corrispettivo cluster. Viene graficato, poi, il valore puntuale della somma delle distanze al quadrato per ogni valore di k. L'obiettivo è quello di individuare, all'interno della curva ottenuta, un angolo che ne interrompe la linearità, la cui ascissa rappresenta il valore ottimo di k (Figura 6.2).

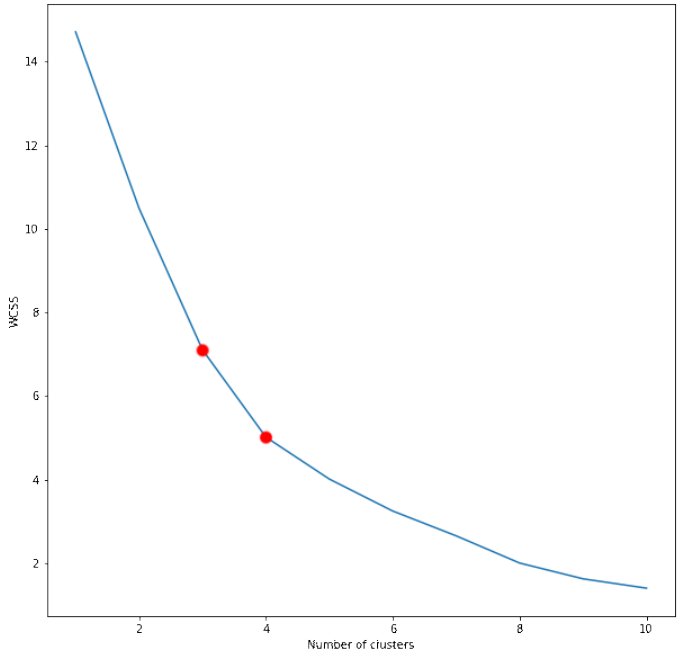

Figura 6.2: Metodo del gomito applicato sul dataset di riferimento. *Fonte: autoprodotta*

Nel caso del progetto, la curva ottenuta è piuttosto regolare e non ci sono punti in cui si presenta un gomito evidente. I punti di maggior interesse, ovvero quelli in cui la curva sembra piegare maggiormente, risultano essere quelli per k pari a 3 e k pari a 4. Tra il risultato ottenuto con k-means e quello con Expectation - Maximization, si è scelto il secondo, perché sembra restituire valori più coerenti, anche se i risultati ottenuti con i due metodi differiscono solo in minima parte. Per l'esecuzione dell'algoritmo si è scelto un numero di cluster pari a 4. I risultati della classificazione sono presentati in Tabella 6.1.

Tabella 6.1: *Tabella riassuntiva sui risultati del clustering.*

Challenge	**Intervallo**	**Cluster di appartenenza**
#bussitchallenge	1 -]0, 15]	***cluster_1*** 🔵
	2 -]15, 60]	***cluster_0*** 🔴
	3 -]60, 90]	***cluster_3*** 🟠
	4 -]90, 100]	***cluster_3*** 🟠
#copinesdancechallenge	1 -]0, 20]	***cluster_0*** 🔴
	2 -]20, 65]	***cluster_3*** 🟠
	3 -]65, 90]	***cluster_3*** 🟠
	4 -]90, 100]	***cluster_2*** 🟢
#emojichallenge	1 -]0, 10]	***cluster_3*** 🟠
	2 -]10, 35]	***cluster_3*** 🟠
	3 -]35, 65]	***cluster_3*** 🟠

	4 -]65, 100]	***cluster_0*** 🔴
#itookanap	1 -]0, 10]	***cluster_3*** 🟠
	2 -]10, 55]	***cluster_3*** 🟠
	3 -]55, 100]	***cluster_0*** 🔴
#colpiditesta	1 -]0, 15]	***cluster_2*** 🟢
	2 -]15, 45]	***cluster_2*** 🟢
	3 -]45, 95]	***cluster_1*** 🔵
	4 -]95, 100]	***cluster_0*** 🔴
#boredinthehouse	1 -]0, 15]	***cluster_0*** 🔴
	2 -]15, 60]	***cluster_3*** 🟠
	3 -]60, 90]	***cluster_3*** 🟠
	4 -]90, 100]	***cluster_3*** 🟠
#plankchallenge	1 -]0, 15]	***cluster_3*** 🟠
	2 -]15, 70]	***cluster_0*** 🔴
	3 -]70, 100]	***cluster_2*** 🟢
#silhouettechallenge	1 -]0, 20]	***cluster_2*** 🟢
	2 -]20, 55]	***cluster_3*** 🟠
	3 -]55, 100]	***cluster_1*** 🔵
#bugsbunnychallenge	1 -]0, 10]	***cluster_2*** 🟢
	2 -]10, 40]	***cluster_2*** 🟢
	3 -]40, 70]	***cluster_3*** 🟠
	4 -]70, 100]	***cluster_1*** 🔵
#strippatiktok	1 -]0, 5]	***cluster_2*** 🟢
	2 -]5, 25]	***cluster_2*** 🟢
	3 -]25, 100]	***cluster_1*** 🔵
#firewroks	1 -]0, 15]	***cluster_3*** 🟠
	2 -]15, 25]	***cluster_3*** 🟠
	3 -]25, 100]	***cluster_0*** 🔴
#fightchallenge	1 -]0, 10]	***cluster_3*** 🟠
	2 -]10, 60]	***cluster_3*** 🟠
	3 -]60, 75]	***cluster_3*** 🟠
	4 -]75, 100]	***cluster_0*** 🔴
#sugarbaby	1 -]0, 10]	***cluster_2*** 🟢
	2 -]10, 35]	***cluster_3*** 🟠
	3 -]35, 60]	***cluster_3*** 🟠
	4 -]60, 100]	***cluster_3*** 🟠
#updownchallenge	1 -]0, 10]	***cluster_2*** 🟢
	2 -]10, 30]	***cluster_2*** 🟢
	3 -]30, 100]	***cluster_0*** 🔴

La rappresentazione dei cluster nel piano cartesiano, invece, è riportata in Figura 6.3.

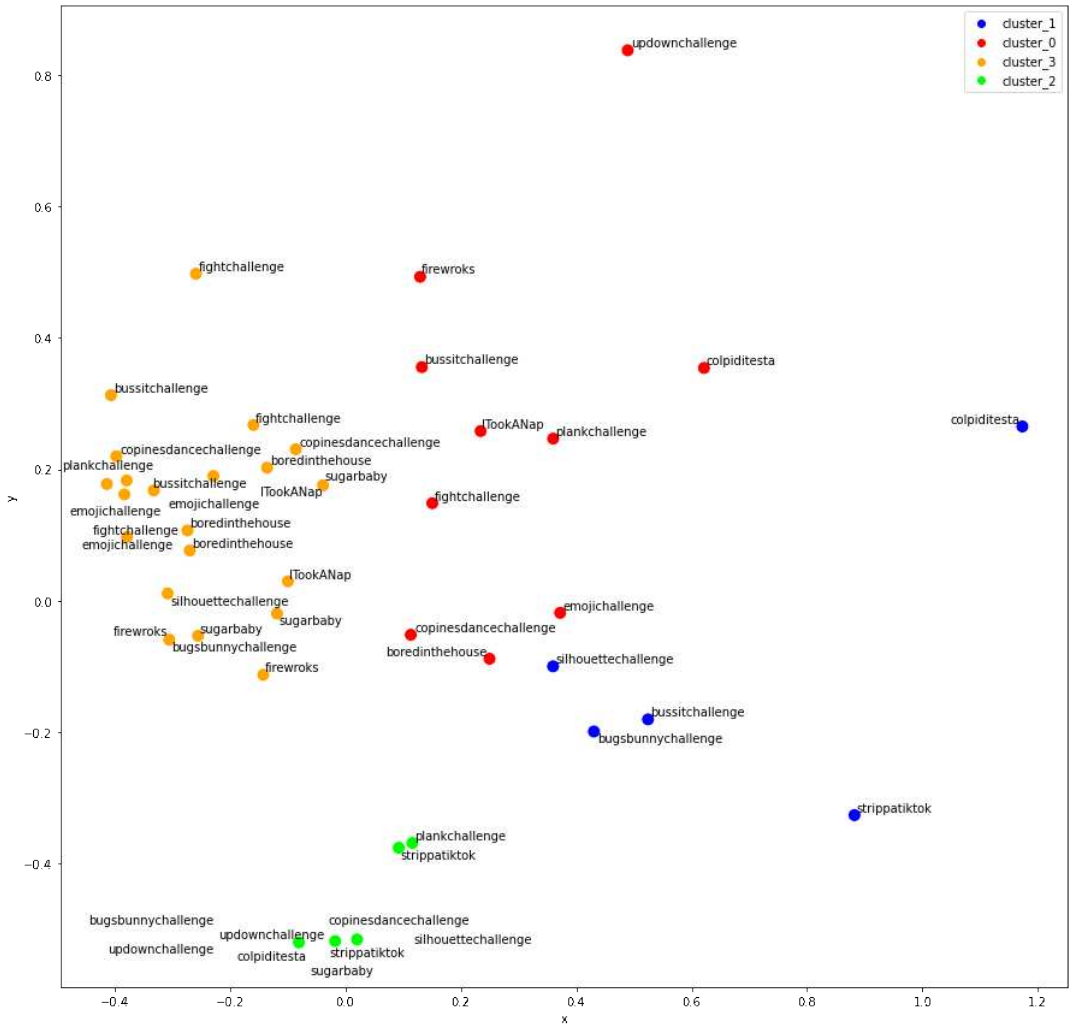

Figura 6.3: Rappresentazione dei cluster sul piano cartesiano. *Fonte: autoprodotta*

6.1.3 Caratterizzazione degli intervalli

Prima di procedere con l'analisi delle sequenze, si è effettuata un'analisi qualitativa relativamente al valore delle feature per ciascun cluster, al fine di categorizzare, dal punto di vista dei dati, le regioni individuate. I valori medi delle più interessanti *feature* analizzate sono riportati in Tabella 6.2.

Tabella 6.2: *Tabella riassuntiva sui valori delle principali feature.*

	cluster_0	***cluster_1***	***cluster_2***	***cluster_3***
Numero di intervalli	10	5	10	23
Numero medio di componenti connesse	35.4	21.4	1.7	10.9
Grado medio dei nodi	0.92	1.24	0.66	0.66
Percentuale media di nodi nella massima componente connessa	50.6%	83.7%	97.4%	35.1%
Numero medio di ego-network	30.6	65.4	1.7	4.1

Osservando i dati, si può definire la seguente classificazione:

- ***cluster_0***: gli intervalli compresi in questo raggruppamento sono caratterizzati da un numero di componenti connesse molto elevato con, tuttavia, un grado medio particolarmente grande. Ciò significa che, nelle comunità che si vengono a creare, vi sono dei nodi con molte connessioni. Infatti, le reti che caratterizzano gli intervalli raggruppati in questo cluster presentano un elevato numero di ego-network.
- ***cluster_1***: similmente al "*cluster_0*", le reti degli intervalli compresi in questo gruppo presentano tante componenti connesse costituite da nodi di grado molto elevato; valore riscontrato sul numero di ego-network è mediamente doppio rispetto a quelli del cluster precedente. La componente connessa di dimensione massima include un elevato numero di nodi, il che significa che le altre comunità sono costituite da pochi nodi, generalmente coppie di utenti. È il cluster che coinvolge il minor numero di intervalli, e quello che per caratteristiche si avvicina di più al "*cluster_0*".
- ***cluster_2***: questo terzo raggruppamento rappresenta intervalli le cui reti sono fortemente connesse; infatti, in media, gli intervalli ivi contenuti non includono più di due componenti. Inoltre, nei casi in cui le reti degli intervalli presentano più di una componente connessa, questa contiene quasi la totalità dei nodi della rete stessa.
- ***cluster_3***: questo cluster presenta caratteristiche intermedie rispetto ai due primi gruppi e al *cluster_2*. Le reti degli intervalli categorizzati con questo raggruppamento presentano diverse componenti connesse di dimensione più bilanciata. Infatti, mediamente, la più grande componente connessa contiene solo il 35% circa di nodi. È la categoria di cluster più diffusa, in quando rappresenta il maggior numero di intervalli.

6.2 Analisi delle sequenze

A questo punto, nota la relazione tra le sequenze degli intervalli delle challenge ed i raggruppamenti individuati, è necessario effettuare un'operazione di astrazione al fine di individuare dei *pattern* ad un livello più alto, ovvero sequenze di sequenze di intervalli, che si ripetono sulle due tipologie di challenge. Lo scopo di questa attività, infatti, è quello di ridurre la variazione delle sequenze, minimizzando ed unendo quegli intervalli che, in base al cluster di appartenenza, si presentano costantemente in coppia o tripletta; in questo modo è possibile diminuire il numero di intervalli per ogni trend, ed ottenere dei *pattern* chiari e distinguibili per ciascuna classe challenge.

L'analisi è effettuata separatamente per challenge positive e negative, poiché l'interesse è quello di individuare *pattern* distinti tra le due classi di trend; tuttavia, si sono effettuate contemporaneamente alcune operazioni preliminari; le sequenze composte da dittonghi di intervalli caratterizzati dallo stesso cluster, come, ad esempio, sequenze di tipologia "*cluster_2 - cluster_2*", sono state aggregate a formare un unico intervallo. Il motivo dietro questa scelta è che gli intervalli che compongono la sequenza possiedono caratteristiche simili, per cui ha senso unirli in un unico intervallo.

Il risultato dell'approssimazione è riassunto in Tabella 6.3. Si noti che, per semplicità, d'ora in avanti nella trattazione, i codici dei cluster vengono espressi solamente con l'identificativo sequenziale che rappresenta il raggruppamento, anziché con la codifica completa comprensiva del prefisso "*cluster_*".

Tabella 6.3: *Tabella riassuntiva sull'aggregazione delle sequenze.*

Challenge	**Aggregazione di intervalli, cluster di appartenenza**
#bussitchallenge	***1 - 0 - 3***

	🔵 - 🔴 - 🟠
#copinesdancechallenge	***0 - 3 - 2*** 🔴 - 🟠 - 🟢
#emojichallenge	***3 - 0*** 🟠 - 🔴
#itookanap	***3 - 0*** 🟠 - 🔴
#colpiditesta	***2 - 1 - 0*** 🟢 - 🔵 - 🔴
#boredinthehouse	***0 - 3*** 🔴 - 🟠
#plankchallenge	***3 - 0 - 2*** 🟠 - 🔴 - 🟢
#silhouettechallenge	***2 - 3 - 1*** 🟢 - 🟠 - 🔵
#bugsbunnychallenge	***2 - 3 - 1*** 🟢 - 🟠 - 🔵
#strippatiktok	***2 - 1*** 🟢 - 🔵
#firewroks	***3 - 0*** 🟠 - 🔴
#fightchallenge	***3 - 0*** 🟠 - 🔴
#sugarbaby	***2 - 3*** 🟢 - 🟠
#updownchallenge	***2 - 0*** 🟢 - 🔴

6.2.1 Ricerca di pattern, challenge positive

Inizialmente, ci si è concentrati sulla ricerca di *pattern* sulle challenge positive. Dalla Tabella 6.3, le cui prime sette righe rappresentano i trend di questa categoria, si notano che i due cluster "*0*" e "*3*" compaiono, ad eccezione della challenge ***#colpiditesta***, sempre contigui. La sequenza ordinata rappresenta un passaggio della challenge da un momento in cui gli utenti sono fortemente uniti, con presenza di un ampio numero di ego-network e poche componenti connesse, che caratterizzano un periodo in cui vi sono pochi di video che, però, assumono carattere virale, ad uno in cui i video virali hanno scemato di interesse, ma si sono formate diverse comunità di utenti che partecipano alla challenge. Viceversa, la sequenza con ordine inverso indica il passaggio da un'assenza di video virali ad un momento in cui alcuni di essi assumono una forte popolarità. A seguire si è avanzato

un'ipotesi:

- *Ipotesi A*: il cluster "*0*" può essere approssimato al cluster "*1*"; si definisce, quindi, un super-cluster "*A*" $\approx$ "*0*" $\approx$ "*1*".

Tale ipotesi è ripresa e valutata successivamente; se vera, permette di effettuare la sostituzione mostrata in Tabella 6.4. Si noti che, quando presente, si effettua l'aggregazione del pattern "*A - A*", che deriva dalla sequenza di intervalli che appartengono, nell'ordine, ai cluster "*1*" e "*0*".

Tabella 6.4: *Astrazione delle sequenze, challenge positive.*

Challenge	**Sequenza individuata**
#bussitchallenge	***A - 3*** 🟣 - 🟠
#copinesdancechallenge	***A - 3 - 2*** 🟣 - 🟠 - 🟢
#emojichallenge	***3 - A*** 🟠 - 🟣
#itookanap	***3 - A*** 🟠 - 🟣
#colpiditesta	***2 - A*** 🟢 - 🟣
#boredinthehouse	***A - 3*** 🟣 - 🟠
#plankchallenge	***3 - A - 2*** 🟠 - 🟣 - 🟢

Ponendo S_1 pari alla sequenza di cluster "*A - 3*", e ponendo S_2 pari, invece, alla sequenza "*3 - A*" si ottengono i risultati in Tabella 6.5.

Tabella 6.5: *Pattern, challenge positive.*

Challenge	**Sequenza individuata**
#bussitchallenge	S_1
#copinesdancechallenge	S_1 - *2*
#emojichallenge	S_2
#itookanap	S_2
#colpiditesta	*2 - A*
#boredinthehouse	S_1
#plankchallenge	S_2 - *2*

Si noti che il cluster "*2*", ove presente in sequenza, segue in un caso il *pattern* S_1, per

quanto riguarda la challenge ***#copinesdancechallenge***, mentre nel caso di ***#plankchallenge*** segue il *pattern* S_2. Osservando, invece, la sequenza che compone la challenge ***#colpiditesta***, costituita dai raggruppamenti "*2 - A*", si può concludere che tale coppia ordinata di cluster termini il ciclo di vita di una challenge positiva, e che quindi, acquisendo dati sui diversi trend in un momento futuro rispetto a quello della trattazione, tale sequenza appaia anche per tutte le altre challenge. In conclusione, definendo quindi S_{end} pari alla sequenza di coda "*2 - A*" di una challenge positiva, si possono distinguere due *pattern* S_1 - S_{end} ed S_2 - S_{end}, che caratterizzano le challenge positive.

6.2.2 Ricerca di pattern, challenge negative

La stessa operazione di ricerca di *pattern* è stata effettuata per le challenge negative. Considerando, ancora una volta, valida l'*ipotesi A* enunciata alla sezione precedente, si ottiene, per le challenge negative, l'astrazione riportata in Tabella 6.6.

Tabella 6.6: *Astrazione delle sequenze, challenge negative.*

Challenge	Sequenza individuata
#silhouettechallenge	***2 - 3 - A*** 🟢 - 🟠 - 🟣
#bugsbunnychallenge	***2 - 3 - A*** 🟢 - 🟠 - 🟣
#strippatiktok	***2 - A*** 🟢 - 🟣
#firewroks	***3 - A*** 🟠 - 🟣
#fightchallenge	***3 - A*** 🟠 - 🟣
#sugarbaby	***2 - 3*** 🟢 - 🟠
#updownchallenge	***2 - A*** 🟢 - 🟣

In questo caso, l'individuazione di *pattern* è più immediata; ad eccezione delle challenge ***#strippatiktok*** e ***#updownchallenge***, il super-cluster "***A***" appare preceduto immediatamente dal raggruppamento "*3*", mentre in ogni caso il cluster "*2*" precede il super-cluster "*A*", e "*3*". Per questo motivo, si può considerare il dittongo "*3 - A*" come sequenza finale del ciclo di vita di una challenge negativa, mentre un intervallo classificato come cluster "*2*" può essere visto come la sua fase iniziale; nel caso delle due challenge che fanno eccezione, è opportuno specificare che gli intervalli sono calcolati mediante operazioni che, di volta in volta, approssimano i risultati ottenuti. Un intervallo di breve durata può essere, infatti, eliminato da tale somma di semplificazioni, per cui la presenza di eccezioni, che siano, tuttavia, nel complesso, coerenti con i risultati ed in quantità contenuta, non è motivo di preoccupazione.

Per concludere, si definisce S_{neg} la sequenza di cluster "*2 - 3 - A*", che caratterizza tutte le challenge considerate negative, o pericolose.

6.3 Considerazioni e dimostrazione

Tutte le astrazioni elaborate finora sono funzione dell'*ipotesi A*, la cui non veridicità invalida il risultato raggiunto a questo punto. Inizialmente, si è formulato questa ipotesi sulla base di quanto riportato in Tabella 6.2, la quale suggerisce, di per sè, una somiglianza tra i due cluster "*0*" e "*1*", e mostra, per il secondo gruppo, una cardinalità molto bassa. Inoltre, effettuando diversi tentativi di clustering, mantenendo sempre il numero di cluster pari a 4, si è notato che, mentre i gruppi "*2*" e "*3*" sono molto stabili ad ogni esecuzione, ed in ogni caso categorizzano gli stessi intervalli, in alcuni casi gli intevalli inizialmente contenuti nel cluster "*1*" vengono parzialmente inglobati nel raggruppamento "*0*". A questo punto, si è effettuata l'operazione di clustering riducendo a tre il numero di cluster attesi. Il risultato del'operazione è mostrato in Figura 6.4.

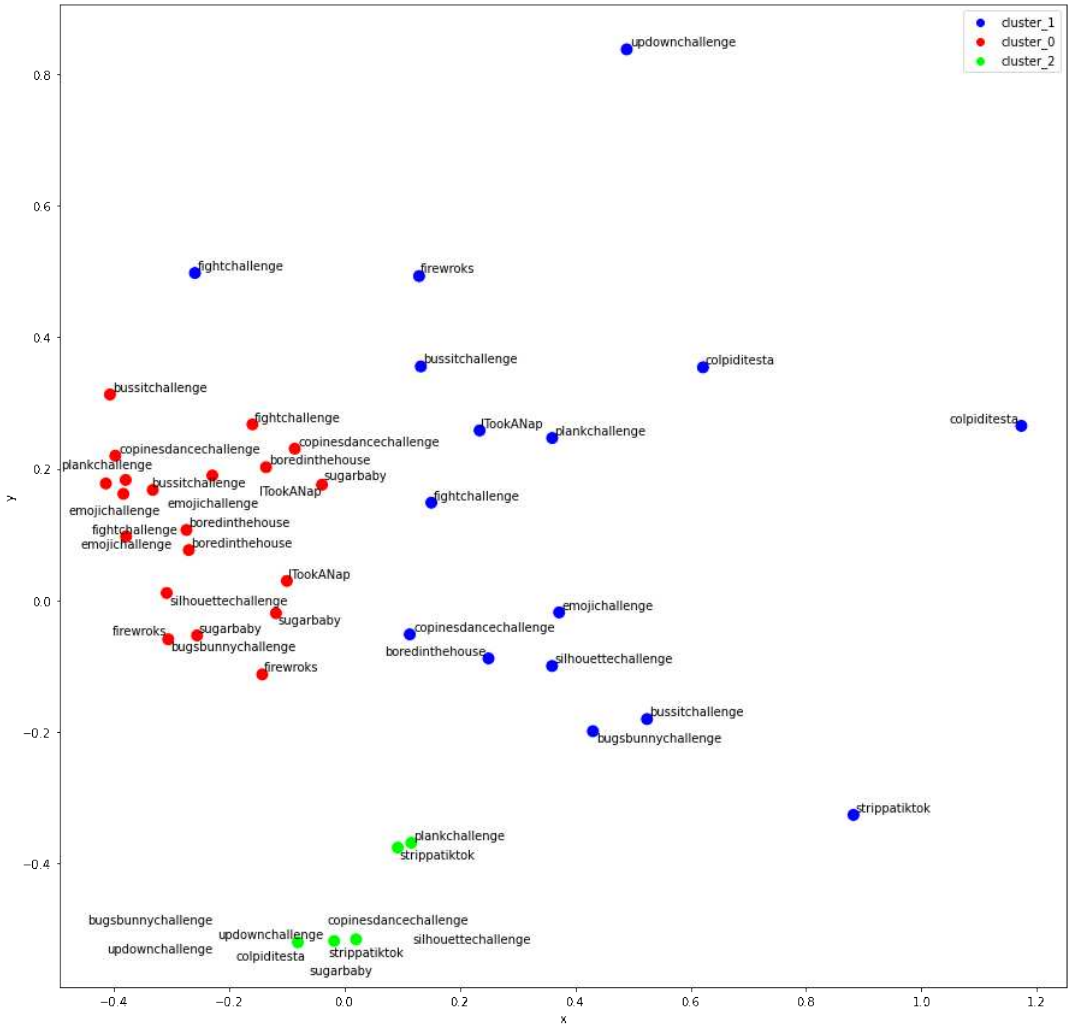

Figura 6.4: Rappresentazione dei cluster sul piano cartesiano, $k = 3$. *Fonte: autoprodotta*

Come si nota, anche in questo caso i cluster "*2*" e "*3*" (ora rappresentati da "*cluster_2*" e "*cluster_0*", rispettivamente) rimangono invariati, sia in cardinalità che in intervalli contenuti. Il qui rappresentato "*cluster_1*" corrisponde, invece, al'unione dei cluster "*0*" e "*1*", confermando la loro somiglianza, e consentendo di aggregare i due cluster in un unico super-raggruppamento. Ciò conferma l'*ipotesi A*, alla quale si può legare il seguente corollario:

Il super-cluster "*A*", definito come in *ipotesi A*, può essere approssimato pari al "*cluster_1*", ottenuto in seguito all'esecuzione dell'algoritmo di clustering, con k pari a 3.

7. Confronto con approcci correlati

7.1 Stato dell'arte

TikTok è il social network più diffuso, almeno tra i giovanissimi. Per tale motivo, nel corso del tempo, sono stati pubblicati diversi lavori che si focalizzano su tale social network. Così, in letteratura, esistono numerosi *paper* che si occupano di questa piattaforma sociale, analizzandola da più punti di vista.

Alcuni di essi, come il *paper "Research on the Causes of the "TikTok" App Becoming Popular and the Existing Problems"* di Xu Li, Yan Xiaohui e Zhang Zhengwu, analizzano i motivi per cui TikTok è diventato così popolare. All'interno di questo lavoro, vengono analizzati i vantaggi ed i pericoli di un'utilizzo intensivo dell'applicazione. Viene, inoltre, fornita un'analisi relativa alle strategie utilizzate, come quelle di promozione e di *advertising*, all'uso dell'Intelligenza Artificiale ed alla soddisfazione di utilizzo da parte dell'utente.

I lati negativi dell'utilizzo di TikTok, come, ad esempio, la forte presenza di fake news, e la crescente dipendenza degli utenti verso la piattaforma, sono tematiche molto dibattute e su cui sono stati effettuati diversi studi.

Il *paper "The paradox of TikTok anti-pro-anorexia videos: how social media can promote non-suicidal self injury and anorexia"*, di Giuseppe Lo grieco, Maria Rosaria Marchili, Mario Roversi e Alberto Villani, esamina la manipolazione del pensiero adolescenziale in TikTok. Essa è dovuta al fatto che alcuni video presenti sulla piattaforma possono spingere verso comportamenti depressivi, i quali possono sfociare, come da titolo, in anoressia o, addirittura, nel suicidio.

Ancora, la ricerca *"Cyberbullying in the world of teenagers and social media: A literature review"*, curata da Sophia Alim, tratta il tema del cyber-bullismo, che, negli ultimi anni, è diventato un problema sempre più sentito tra gli adolescenti, a causa del loro ingente utilizzo di social media, come TikTok. Infatti, gli studi su questo tema hanno evidenziato un crescendo di criticità, come l'elevato volume di incidenti di cyber-bullismo a scuola, la maggiore divulgazione di informazioni personali sui social media, il condizionamento da parte degli *influencer* e la sicurezza dell'ambiente scolastico, sia per il bullo che per la vittima.

La piattaforma è stata analizzata anche da un punto di vista politico; il *paper "Spreading hate on TikTok"*, di Gabriel Weimann e Natalie Masri, analizza in modo descrittivo il contenuto di alcuni

video caricati su TikTok nel primo trimestre del 2020. I risultati rivelano una crescente presenza in TikTok di gruppi estremisti, legati ad ideologie radicate, di estrema destra, che propagandano le loro idee pubblicando video, commenti, e promuovendo simbologie ed icone storicamente legate a tale ala politica.

Sempre da un punto di vista politico, il *paper "Social Media and Fake News in the 2016 Election"* di Hunt Allcott e Matthew Gentzkow, tratta alcune fake news circolate in merito alle elezioni presidenziali americane del 2016. Questa ricerca approfondisce la costruzione di notizie false, e presenta nuovi dati sulla loro pubblicazione prima delle elezioni. Infatti, risulta che circa il 14% dei cittadini americani considera i social media la loro fonte d'informazione principale. In secondo luogo, il *paper* individua alcune delle notizie false maggiormente virali apparse nei tre mesi precedenti alle elezioni, e dimostra come quelle a favore di Donald Trump, il candidato vincitore, sono state condivise complessivamente circa 30 milioni di volte sui social media, mentre quelle a favore della sua rivale Hillary Clinton sono state condivise circa 8 milioni di volte.

La tematica delle notizie false e della disinformazione sono riprese, anche, nella pubblicazione *"Trends in the diffusion of misinformation on social media"* di Hunt Allcott, Matthew Gentzkow e Chuan Yu. Negli ultimi anni, si è diffusa la preoccupazione che la disinformazione sui social media possa danneggiare le società e le istituzioni democratiche; in risposta, le piattaforme hanno annunciato azioni per limitare la diffusione di contenuti falsi.

Analogamente, il *paper* degli autori Juan Carlos Medina Serrano, Orestis Papakyriakopoulos, e Simon Hegelich *"Dancing to the Partisan Beat: A First Analysis of Political Communication on TikTok"*, effettua una valutazione primaria della comunicazione politica su TikTok. Nello studio sono raccolti una serie di video di autori americani, sia di fazione repubblicana che democratica, caricati su TikTok. L'obiettivo è quello di indagare su come tali utenti comunicano tra di loro. Con l'aiuto di tecniche che si servono di intelligenza artificiale, viene dimostrato che la comunicazione politica su tale social network è molto più interattiva rispetto ad altre piattaforme, con gli utenti che fanno uso di molteplici canali di informazione per diffondere le loro idee. In termini di dati demografici, emerge gli utenti repubblicani hanno generato un maggior numero di contenuti, ed i loro video hanno ricevuto più interazioni; gli utenti democratici, al contrario, si sono impegnati molto di più in discussioni trasversali.

Come detto in precedenza, gli ambiti di ricerca su TikTok, ed in generale sulle diverse piattaforme di social network, sono molteplici. Ad esempio, il *paper "Communicating COVID-19 information on TikTok: a content analysis of TikTok videos from official accounts featured in the COVID-19 information hub"*, di Li Yachao, Guan Mengfei, Hammond Paige e Berrey Lane, mostra come le caratteristiche ed il contenuto dei video di TikTok, relativi alla pandemia mondiale di COVID-19, sono correlati agli indicatori quantitativi del coinvolgimento degli utenti, inclusi il numero di visualizzazioni, di like, di commenti e di condivisioni. Dai risultati emerge come i video che trasmettevano emozioni di allarme, preoccupazione, suscettibilità e gravità, relativamente al virus COVID-19, hanno avuto un enorme coinvolgimento da parte degli utenti. Questo studio può aiutare gli enti di sanità pubblica ad essere maggiormente consapevoli dell'opportunità che rappresenta TikTok, nonché gli altri social network, nella comunicazione sanitaria. Tale consapevolezza può contribuire a creare una comunicazione del rischio incentrata sul pubblico, per coinvolgere e informare i membri della comunità.

Conclusioni

Conclusioni

In questo elaborato è stato presentato un approccio innovativo per la caratterizzazione delle diverse fasi di sviluppo di una challenge di TikTok, sfruttando le metodologie che gli studi sulla Social Network Analysis, nel tempo, hanno messo a disposizione.

Inizialmente, sono stati presentati l'ambito del lavoro e gli strumenti teorici applicati nel corso del progetto. Successivamente, si è definita la tecnica di estrazione delle informazioni, punto di partenza del progetto e dell'approfondimento dei capitoli successivi. A seguire, si è dapprima definito il modello a grafo con cui si sono rappresentati i dati ottenuti, e, successivamente, la metodologia di individuazione degli intervalli temporali che segnano i diversi momenti di vita di una challenge, legandovi caratteristiche quantitative ereditate dalla teoria dei grafi. In conclusione, si è definita una metodologia per astrarre ed aggregare intervalli comuni a più challenge, con il fine di studiare *pattern* caratteristici delle due classi di eventi oggetto di studio, ovvero le challenge pericolose e quelle non pericolose.

Sviluppi futuri

Nel futuro si può pensare di approfondire il processo di astrazione di *pattern*, dimostrando, da un punto di vista quantitativo, l'*ipotesi A*, su cui è basata la definizione delle sequenze comuni.

Successivamente sarebbe interessante analizzare la fattibilità della realizzazione di uno strumento che, basato sul processo descritto nel lavoro, sia capace di analizzare i video che vengono caricati sulla piattaforma TikTok, associandoli allo storico della challenge a cui si riferisce, e classificandoli automaticamente come pericolosi o non pericolosi.

Bibliografia

Riferimenti bibliografici

- Lorenzo Giuliani, Silvia Cecchini, codebase for tiktoknet, `https://github.com/lor95/tiktoknet`, 2021
- O'Reilly Tim, What is Web 2.0, Design Patterns and Business Models for the Next Generation of Software. 2005.
- Solima Ludovico, Social network: verso un nuovo paradigma per la valorizzazione della domanda culturale, Sinergie, n. 82/10.
- Johannes Ahlse, Felix Nilsson, Nina Sandström, It's time to TikTok. 2020.
- Ben Cost, Asia Grace, Marisa Dellatto and Eric Hegedus. The 21 craziest TikTok challenges so far. `nypost.com/article/craziest-tiktok-challenges-so-far/`. 2021.
- John Scott, "Social Network Analysis". Sage Publications, 1991.
- Lawrence Page, Sergey Brin, Rajeev Motwani and Terry Winograd, 1998. The PageRank Citation Ranking: Bringing Order to the Web. Stanford InfoLab.
- Stack Overflow Developer Survey 2021 - Integrated Development Environment. Stack Overflow Insights. Stack Exchange (2021).
- David Teather, TikTok-Api: Unofficial TikTokApi, `github.com/davidteather/TikTok-Api`. 2021.
- Juan Carlos Medina Serrano, Orestis Papakyriakopoulos, and Simon Hegelich. 2020. Dancing to the Partisan Beat: A First Analysis of Political Communication on TikTok. In 12th ACM Conference on Web Science (WebSci '20). Association for Computing Machinery, New York, NY, USA, 257–266.
- Alim, S. (2017). Cyberbullying in the world of teenagers and social media: A literature review. In Information Resources Management Association, Gaming and technology addiction: Breakthroughs in research and practice (pp. 520–552). Information Science Reference/IGI Global.
- Allcott, Hunt, and Matthew Gentzkow. 2017. "Social Media and Fake News in the 2016 Election." Journal of Economic Perspectives, 31 (2): 211-36.
- Li Y, Guan M, Hammond P, Berrey LE. Communicating COVID-19 information on TikTok:

a content analysis of TikTok videos from official accounts featured in the COVID-19 information hub [published online ahead of print, 2021 Mar 1]. Health Educ Res. 2021.

- Gabriel Weimann & Natalie Masri (2020) Research Note: Spreading Hate on TikTok, Studies in Conflict & Terrorism, DOI: 10.1080/1057610X.2020.1780027.
- Omar, B. & Dequan, W. (2020). Watch, Share or Create: The Influence of Personality Traits and User Motivation on TikTok Mobile Video Usage. International Association of Online Engineering. Retrieved June 27, 2021 from https://www.learntechlib.org/p/216454/.
- Basch, Corey H., Grace C. Hillyer, and Christie Jaime. "COVID-19 on TikTok: harnessing an emerging social media platform to convey important public health messages." International journal of adolescent medicine and health 1.ahead-of-print (2020).
- Hayes, Clare, et al. "“Making Every Second Count”: Utilizing TikTok and Systems Thinking to Facilitate Scientific Public Engagement and Contextualization of Chemistry at Home." (2020): 3858-3866.
- Masciantonio, Alexandra, et al. "Don't put all social network sites in one basket: Facebook, Instagram, Twitter, TikTok, and their relations with well-being during the COVID-19 pandemic." PloS one 16.3 (2021): e0248384.
- Herrick, Shannon SC, Laura Hallward, and Lindsay R. Duncan. "This is just how I cope: An inductive thematic analysis of eating disorder recovery content created and shared on TikTok using EDrecovery." International Journal of Eating Disorders (2020).
- Zhao, Zhengwei. "Analysis on the “Douyin (Tiktok) Mania” Phenomenon Based on Recommendation Algorithms." E3S Web of Conferences. Vol. 235. EDP Sciences, 2021.
- Ravikumar, Vaishali, et al. "Is TikTok the New Instagram? Analysis of Plastic Surgeons on Social Media." Plastic and Reconstructive Surgery 147.5 (2021): 920e-922e.
- Zhu, Yumei. "The Expectation of TikTok in International Media: A Critical Discourse Analysis." Open Journal of Social Sciences 8.12 (2020): 136-148.
- Olivares-Garcia, Francisco J., and Maria Ines Mendez Majuelos. "Analysis of the main trends published on TikTok during the quarantine period by COVID-19." Revista Espanola De Comunicacion En Salud (2020): S243-S252.
- Aisa, Aufia, and Mega Kirana Dewi. "Z Genereations Perspective: Analysis of Islamic Learning through Tiktok Social Media." SCHOOLAR: Social and Literature Study in Education 1.1 (2021): 22-25.
- Gray, Joanne. "The geopolitics of 'platforms': the TikTok challenge." Internet Policy Review (2021).
- Klug, Daniel. " It took me almost 30 minutes to practice this". Performance and Production Practices in Dance Challenge Videos on TikTok." arXiv preprint arXiv:2008.13040 (2020).
- Barbotti, Ilaria. TikTok Marketing: Video virali e hashtag challenge: come fare business con la Generazione Z. Hoepli Editore, 2020.
- Atherton, Rachel Rose. "The 'Nutmeg Challenge': a dangerous social media trend." Archives of disease in childhood 106.5 (2021): 517-518.
- Ahlse, Johannes, Felix Nilsson, and Nina Sandström. "It's time to TikTok: Exploring Generation Z's motivations to participate in Challenges." (2020).
- Henneman, Todd. "Beyond Lip-Synching: Experimenting with TikTok Storytelling." Teaching Journalism & Mass Communication 10.2 (2020): 1-14.
- Knowledge, I. "The TikTok Strategy: Using AI Platforms to take over the world." Retrieved from Insead Knowledge: https://knowledge.insead.edu/entrepreneurship/the-tiktok-strategy-using-ai-platforms-to-take-over-theworld-11776 (2019).

RINGRAZIAMENTI

Voglio dedicare questa tesi ai miei genitori, alla mia ragazza, ai miei amici, e a tutti coloro che mi hanno sostenuto ed affiancato durante questi lunghi anni di studio.

Ringrazio il mio relatore Prof. Domenico Ursino per l'ineccepibile lavoro svolto nel gestire il gruppo di progetto durante la ricerca, e per la pazienza impiegata nel guidarmi durante la stesura di questo elaborato. Ringrazio i dottori Enrico Corradini, Gianluca Bonifazi e Luca Virgili per il grande contributo dato durante tutte le fasi di ricerca. Infine, ringrazio la mia collega Silvia Cecchini, che ha condiviso con me tutta l'esperienza di tirocinio e di scrittura della tesi.

Ancona, 22 ottobre 2021