Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Laurea Magistrale in Informatica

Analisi del comportamento degli utenti Reddit: Tracciamento dei post nei subreddit per identificare gli Opinion Leader

Relatore:
Chiar.mo Prof.
Stefano Ferretti

Correlatore:
Chiar.mo Prof.
Marco Furini

Presentata da:
Giuseppe Massimo

Sessione I
2023/2024

Verso l'infinito
e oltre

Indice

Introduzione

Chi sono gli Opinion Leader e qual'è il loro ruolo? Come vengono diffuse le notizie su Reddit? Si possono prevedere o riconoscere degli eventi dalle interazioni tra gli utenti?
In questo mio studio di tesi ho effettuato un'analisi comportamentale degli utenti in relazione ai post pubblicati su Reddit negli ambiti Cryptocurrency e tecnologie connesse, al fine di determinare gli utenti più attivi ed influenti, denominati Opinion Leader, sulla piattaforma in un determinato periodo temporale.
Per rispondere alle domande che mi sono posto, è stato necessario raccogliere i dati necessari all'analisi, ovvero i post che gli utenti Reddit (redditors) hanno pubblicato nel periodo che va dal 31-12-2023 ad oggi nei migliori Subreddit che trattano argomenti relativi al mondo delle criptovalute.
Inizialmente, mi sono servito delle API messe a disposizione da Reddit per la raccolta dei dati, ma a causa delle limitazioni imposte dalla società agli sviluppatori senza accesso speciale, il numero di query era molto limitato. Per questo motivo, ho deciso di utilizzare dei dataset forniti da Academic Torrents, una piattaforma BitTorrent scalabile che permette ai ricercatori di condividere e avere a disposizione dati senza sostenere gli alti costi dei fornitori di hosting commerciali.
Attraverso un tool, sono riuscito ad ottenere tutti i post e i commenti pubblicati dagli utenti nei subreddit di mio interesse, totalizzando più di 150K post e quasi 9M di commenti. Dopo un'opportuna pulizia dei dati ottenuti, ho effettuato un pre-processing sul titolo e sul testo di ogni post attraverso spaCy, una libreria open source per l'elaborazione del linguaggio naturale, per poter riconoscere la natura del post e assegnarlo ad un opportuno cluster attraverso l'algoritmo K-means. Successivamente, ho estrapolato le parole chiave più utilizzate nei post per ogni cluster e su queste ho condotto un'analisi sulla loro frequenza di utilizzo nel tempo.
Ho condotto anche un'analisi sentimentale tramite TextBlob sui post pubblicati per estrapolare in maniera artificiale le opinioni riguardanti alcune tematiche di interesse.
Infine, ho determinato gli Opinion Leader utilizzando due metodi differenti per poi metterli a confronto. Il primo metodo per il calcolo dell'Opinion Leader Score è stato fornito dal mio correlatore che, in una sua ricerca scientifica, aveva già effettuato un'analisi simile per gli utenti di X. Ho quindi modificato il calcolo dell'engagement score per adattarlo alle interazioni presenti su Reddit.
Per il secondo metodo, invece, ho utilizzato PageRank, un algoritmo utilizzato da Google per attribuire un punteggio alle pagine web basandosi sulla qualità e quantità dei link in ingresso. L'algoritmo può essere utilizzato in tutti quei casi in cui ci sono oggetti connessi da riferimenti reciproci. Nel mio caso, ho costruito un grafo in cui i nodi (utenti) sono collegati tra loro solo se c'è stato uno scambio di commenti. Nelle sezioni successive verranno mostrate nel dettaglio tutte le procedure.
Come si può capire da questa breve introduzione, le motivazioni che mi hanno spinto ad analizzare il comportamento degli utenti su Reddit sono innanzitutto la voglia di scoprire

il funzionamento di un social non molto popolare in Italia e se, come per gli altri media, è possibile riconoscere un utente leader nell'informazione ed in particolare se è possibile riconoscere o prevedere eventi nell'ambito delle criptovalute.
er fare ciò, si è dovuta denaturalizzare la struttura fortemente decentralizzata di Reddit per poterla mettere al pari di altri social network come X ed ottenere una panoramica sull'attività degli utenti e verificare se questa viene influenzata da utenti specifici in relazione a particolari avvenimenti.

0.1 Reddit

0.1.1 Storia

Reddit prende vita il 23 giugno del 2005 da un'idea di Steve Huffman e Alexis Ohanian,due ventiduenni neolaureati all'Università della Virginia e compagni di stanza. Partito inizialmente come progetto start-up denominato "MyMobileMenu", riceve subito un finanziamento di 100 mila dollari dalla Y Combinator. Venne sviluppato il sito con lo scopo di aggregare notizie solo quando al team si unì il terzo co-fondatore, Aaron Swartz. Il nome Reddit deriva da "read it" (leggilo) e riflette l'obiettivo di creare un luogo in cui gli utenti potessero scoprire e discutere contenuti. Dopo la fusione con la Infogami di Swartz, avvenuta tra novembre 2005 e gennaio dell'anno successivo, il 31 ottobre 2006 la piattaforma viene acquistata da Condé Nast Publications (la società proprietaria di Wired) a una cifra compresa tra i 10 e i 20 milioni di dollari e ha spostato la propria sede a San Francisco. Nel 2008 Reddit diventa un progetto open source, con la pubblicazione del proprio codice sorgente su GitHub, popolare sito di hosting per progetti software. La piattaforma ha attirato molta attenzione su di sé quando, nel 2012, ha fatto una campagna contro la legge statunitense sulla pirateria online (Stop Online Piracy Act) e, insieme ad altri siti noti (tra cui Wikipedia) ha messo offline l'intero dominio con un blackout di protesta. La piattaforma ha affrontato negli anni sfide relative alla moderazione dei contenuti, cercando di bilanciare la libertà di espressione con la prevenzione di abusi e contenuti dannosi. Reddit oggi conta 430 milioni di utenti al mondo anche se la maggior parte, quasi il 70%, è concentrata negli Stati Uniti. Dal 2011 la società fa capo alla Advanced Publications, azienda statunitense che controlla Discovery Channel, Condé Nast Publications e Lycos. Nel 2024 Reddit debutta in borsa vantando una valutazione di 15 miliardi di dollari.

0.1.2 Come funziona?

Reddit si differenzia da altri social network sotto diversi punti di vista. Innanzitutto, si basa su gruppi di discussione e comunità tematiche, i subreddit,nei quali sono consentite discussioni solo su specifici argomenti, mentre altre realtà come Facebook, Instagram o X sono più orientate verso profili personali, connessioni dirette tra utenti e una homepage di contenuti personalizzata. La caratteristica distintiva di Reddit è la presenza di un sistema di votazione che permette di esprimere non solo giudizi positivi (upvote, freccia in alto) ma anche negativi (downvote, freccia in basso), contribuendo così a determinare la visibilità del contenuto e la sua posizione nella gerarchia del subreddit e dell'homepage. Gli utenti possono anche commentare i post, soggetti anch'essi a voti favorevoli o contrari. I karma point sono punteggi che contribuiscono alla reputazione del redditor e si accumulano attraverso gli upvote. La piattaforma è decentralizzata, senza un subreddit principale, e i nuovi account vengono iscritti automaticamente a diverse aree di interesse di default. L'homepage raccoglie una selezione di post dai subreddit a cui ogni utente è iscritto, basandosi su valori di rating come il tempo dalla pubblicazione, il rapporto upvote-downvote e il numero totale di voti. In breve, Reddit è una comunità online diversificata, decentralizzata e partecipativa, dove gli utenti possono esplorare e contribuire a una vasta gamma di discussioni e contenuti. Un'ultima caratteristica distintiva di Reddit è la libertà: ogni utente è al tempo stesso creatore, fruitore e curatore. Al di là delle policy di utilizzo, sono i moderatori dei vari subreddit a vigilare sulle proprie comunità.

0.1.3 Reddit & Crypto

Nel 2013 Reddit ha stipulato una partnership con Coinbase, piattaforma di compravendita di criptovalute, e ha iniziato ad accettare Bitcoin come corrispettivo della sottoscrizione del servizio Reddit Gold. Nel 2020, Reddit ha lanciato i Community Point ovvero dei token in formato ERC20 basati su Ethereum, punti che gli utenti potevano guadagnare in base alla reputazione che riuscivano a crearsi all'interno della community per l'acquisto di oggetti e badge escluisivi per i loro avatar. Il servizio venne dismesso per problemi di scalabilità nell'ottobre del 2023. Ad inizio 2024,in concomitanza con i preparativi per quotarsi in borsa, la società ha dovuto compilare un form necessario per l'offerta pubblica di azioni e successiva quotazione. Attraverso questo documento il gruppo ha confermato di possedere all'interno del proprio portafoglio Bitcoin ed Ethereum.

Capitolo 1

Tecnologie utilizzate e background

In questo capitolo verranno presentate brevemente tutte le tecnologie adottate più importanti per l'analisi e i riferimenti presenti in letteratura.

1.1 Python e librerie

L'analisi in questione è stata condotta utilizzando come linguaggio di programmazione Python e parte delle sue librerie.

1.1.1 Elabroazione del Linguaggio Naturale (NLP)

Per l'elaborazione del linguaggio naturale sono state utilizzate tre librerie differenti:

- **SpaCy** : SpaCy [1.5] è una libreria open-source di elaborazione del linguaggio naturale (NLP) in Python per la costruzione di applicazioni basate su dati testuali. In particolare SpaCy è stata utilizzata per individuare le parti di testo di ogni singolo post su Reddit prima della fase di clusterizzazione in modo da riconoscere in maniera più dettagliata il corpo dei post per l'assegnamento ad una categoria (cluster).

- **TfidfVectorizer** : TfidfVectorizer è una classe della libreria scikit-learn [2.5] in Python utilizzata per convertire una raccolta di documenti di testo in una matrice di caratteristiche TF-IDF. TF-IDF sta per Term Frequency-Inverse Document Frequency. Questa tecnica è ampiamente utilizzata nel campo del Natural Language Processing (NLP) per valutare quanto una parola sia importante rispetto al contesto di un documento o di una raccolta di documenti. In questo caso è stata utilizzata per convertire il testo processato da SpaCy in una matrice TF-IDF.TF-IDF pesa le parole in modo che le parole comuni (come "the", "and") abbiano un peso inferiore, mentre le parole più rare e significative per ciascun documento abbiano un peso maggiore e inoltre la rappresentazione vettoriale facilita il clustering.

- **TextBlob** : TextBlob [3.5] è una libreria di elaborazione del linguaggio naturale (NLP) per Python costruita su NLTK e Pattern per operazioni di NLP di base. In particolare è stata utilizzata per condurre una Sentiment Analysis sui post per cogliere il grado di accettazione da parte degli utenti sui post pubblicati.

1.1.2 Clustering e Riduzione della Dimensionalità

PCA

PCA è una tecnica di riduzione della dimensionalità utilizzata per trasformare un dataset con molte variabili correlate in un dataset con meno variabili non correlate, chiamate componenti principali.Nel particolare mi è stato utile a semplificare i dati in modo da avere una visualizzazione migliore dei cluster

K-means

K-Means è un algoritmo di clustering non supervisionato che suddivide un insieme di dati in k cluster, dove k è un parametro definito dall'utente. L'obiettivo è raggruppare i dati in modo che i punti all'interno di ciascun cluster siano il più simili possibile tra loro e il più dissimili possibile dai punti degli altri cluster. In particolare :

- Gli elementi del campione vengono assegnati casualmente ai k cluster definiti dall'utente.
- Per ogni elemento del campione, viene calcolata la distanza tra esso e tutti i centroidi di classe, che rappresentano i punti medi dei cluster.
- Ogni elemento del campione viene assegnato al cluster il cui centroide è più vicino.
- I centroidi vengono ricalcolati in base ai punti assegnati a ciascun cluster, rappresentando nuovi punti medi.
- Il processo di assegnazione e ricalcolo viene ripetuto fino a quando non si verifica la convergenza, ovvero fino a quando non ci sono ulteriori spostamenti di elementi dai cluster.

1.1.3 Visualizzazione dei Dati

Matplotlib

Matplotlib [4.5] è una libreria Python di plotting 2D molto popolare e pyplot è un modulo di Matplotlib. pyplot include funzioni per la creazione di figure, l'aggiunta di assi, grafici lineari, grafici a barre, istogrammi, scatter plot, e molto altro.

Seaborn

Seaborn [5.5] è una libreria di visualizzazione dei dati basata su Matplotlib che fornisce un'interfaccia di alto livello per la creazione di grafici statistici. Seaborn è progettato per funzionare bene con i dati strutturati come quelli contenuti in DataFrame di pandas.

Word Cloud

WordCloud [6.5] è una libreria Python per la generazione di word cloud, una rappresentazione visiva delle parole di un testo, dove la dimensione di ogni parola è proporzionale alla sua frequenza o importanza.

Plotly

Plotly [7.5] è una libreria di visualizzazione dei dati interattiva che supporta una vasta gamma di grafici e può essere utilizzata per creare visualizzazioni dinamiche che possono essere esplorate e manipolate direttamente dall'utente.

1.1.4 Analisi delle Reti

NetworkX

NetworkX [8.5] è una libreria Python per la creazione, manipolazione e studio della struttura di grafi complessi e reti. È ampiamente utilizzata per l'analisi di reti sociali, analisi dei dati di rete, bioinformatica, e molte altre applicazioni che coinvolgono la rappresentazione di dati come grafi.Nel mio studio è stata utilizzata per creare il grafo delle interazioni tra utenti per la valutazione del Pagerank

1.2 Stato dell'arte

In letteratura sono presenti diversi studi ed analisi riguardo al comportamento degli utenti nei vari social media ed in particolare su Reddit, ma nessuno di questi focalizza la propria attenzione sull'identificazione degli Opinion Leaders, ovvero quegli utenti che hanno interagito di più sulla piattaforma.
Sono riuscito a trovare vari studi di tesi con obiettivi simili al mio, ma nessuno di questi utilizzava Reddit come sensore passivo e, nonostante i diversi temi di ricerca, nessuno si è concentrato sull'andamento della community legata al mondo delle criptovalute. In queste tesi vengono presi in considerazione altri social media; ad esempio, è stata valutata la possibilità di prevedere un evento particolare (terremoto di Siena) confrontando i dati ottenuti da Instagram e Twitter [5].
In altri casi si è fatta un'analisi simile, ma per tematiche diverse, come le opinioni intorno alla didattica a distanza durante la pandemia di COVID-19 [6] oppure la ricerca di una strategia di marketing basandosi sui post pubblicati su Twitter [7].
Tutti questi studi hanno però in comune il fatto di voler ottenere un pattern dall'analisi dei dati e la ricerca dei temi di interesse attraverso l'individuazione di parole chiave o hashtag.
Alcuni articoli sono stati essenziali per comprendere meglio il concetto di Opinion Leader. Esistono in letteratura molti studi che riguardano la ricerca di personaggi che possano influenzare l'opinione pubblica su svariati temi: da quelli politici [8][9], alimentari [10], sociali [11].
Nonostante sia molto datato, uno studio interessante è stato realizzato da degli studenti cinesi che hanno provato ad identificare gli Opinion Leader su di un forum molto popolare in Cina (TianYa), creando un grafo di nodi utente collegati tra loro considerando il numero di risposte ottenute. Per certi versi, il funzionamento di un forum tende a ricordare il funzionamento di Reddit [12].
Bisogna precisare però che la mia influenza principale è dovuta a delle ricerche pubblicate dai miei relatori, che in passato hanno studiato il comportamento degli utenti su Twitter/X, focalizzando l'attenzione anche sulla ricerca degli utenti che potessero risultare più influenti di altri (Opinion Leader), a differenza degli articoli citati in precedenza che mi hanno unicamente aiutato ad approfondire il significato di Opinion Leader e come

portare avanti una social network analysis in maniera corretta.
In particolare, nel paper [13], è stato condotto uno studio approfondito per esplorare il potenziale di Twitter come sensore passivo nel prevedere il crollo della Stablecoin Unificata (USTC). L'obiettivo principale dello studio era esaminare la correlazione tra i sentimenti espressi nei tweet e il valore di mercato di USTC. In particolare, l'analisi ha rivelato una correlazione moderata con la polarità dei sentimenti.
Nel paper [14] è stata condotta un'analisi dettagliata per esplorare l'utilizzo di X (precedentemente noto come Twitter) come sensore passivo per identificare gli Opinion Leader. Gli Opinion Leader sono quegli utenti che hanno il potere di influenzare significativamente le opinioni e gli atteggiamenti della società. In questo articolo viene presentata una formula per calcolare l'Opinion Leader Score, discostandosi dalla visione tradizionale per cui il numero di followers sia l'unico parametro per identificare un Opinion Leader.
Infine, nella pubblicazione [15] si cerca di capire cosa sia successo sui social media, in particolare X, durante la pandemia di COVID-19 e chi fossero gli Opinion Leader in quell'occasione. In questo studio vengono mappati i tweet su di un grafo e sono stati estratti i leader attraverso l'algoritmo PageRank.
Il mio scopo in questa tesi è stato quello di prendere questi riferimenti ed applicare le metodologie già note su un social media differente e notare le differenze tra approcci diversi per il calcolo degli Opinion Leader.

Capitolo 2

Metodologia

In questa sezione verrano mostrati i dati utilizzati e i metodi ai fini dell'analisi.

2.1 Dataset

Sono stati utilizzati due dataset già processati e distribuiti attraverso Accademic Torrent,in formato JSONL, uno contenente i post degli utenti pubblicati su determinati subreddit e l'altro formato da tutti i commenti pubblicati sugli stessi.

I Subreddit presi in considerazione sono :

- AllCryptoBets
- avatartrading
- bitcoin
- bitcoinbeginners
- btc
- coinbase
- conspiracy
- cryptocurrency
- CryptoCurrencyClassic
- CryptoCurrencyICO
- CryptoCurrencyTrading
- CryptoGeneral
- cryptomarkets
- CryptoMoonShots
- decentraland
- elrondnetwork
- ethdev
- ethereum
- ethtrader
- futurology
- gadgets
- GreenEarthMetaverse
- Metaverse_Blockchain
- metaverse
- nft

La selezione dei Subreddit non è stata casuale, sono andato a ricercare i Subreddit che avessero più seguito per quanto riguarda gli argomenti relativi a Cryptovalute, Metaverso, NFT e tecnologie che potessero essere connesse a questi concetti (conspiracy,futurology,gadgets).

Successivamente per la formazione dei due dataset, ho utilizzato un web tool fornito da Accademic Torrent, che mi ha permesso di ricavare sia i post che i commenti di ogni singolo subreddit con la possibilità di selezionare anche un range di date.
Questo mi ha permesso di ridurre i tempi di download dei dati per poter ottenere solo quelli a me utili. Una volta ricavati i vari file in formato .jsonl ho effettuato un operazione di merge per ottenere due dataset distinti (dataset_posts e dataset_comments)

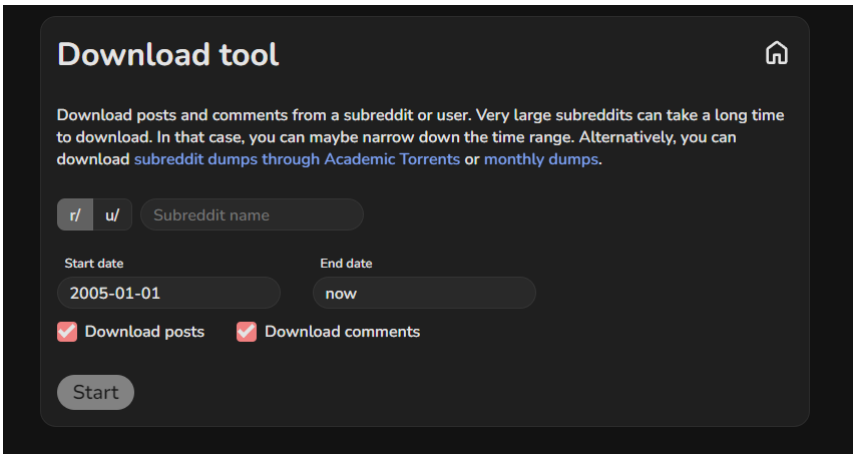

Figura 2.1: Web Tool

2.1.1 Struttura Dataset

Per semplificare la rappresentazione dei due dataset, qui sotto verrano mostrate due tabelle, una per il dataset_posts e l'altra per il dataset_comments, dove verranno indicati solo i campi dei file JSONL necessari all'analisi.

author	**title**	**selftext**	**subreddit**	**created_utc**	**score**	**num_comments**	**name**
username	titolo post	corpo del post	nome del subreddit	data di pubblicazione	punteggio	numero di commenti	post id

Tabella 2.1: Struttura dataset_posts

author	**score**	**parent id**
username	punteggio commento	id commento

Tabella 2.2: Struttura dataset_comments

Dataset Post

Le informazioni essenziali nel dataset per i post sono :

- **author** : Il nickname dell'utente Reddit (Redditors)
- **title** : Titolo del post
- **selftext** : Testo del post
- **subreddit** : Subreddit di appartenenza
- **created_utc** : Data di pubblicazione
- **score** : Punteggio ottenuto equivalente al numero di upvote
- **num_comments** : Numero di commenti ricevuti
- **name** : ID univoco che identifica il post (parent_id in dataset_comments)

Dataset Commenti

Le informazioni essenziali nel dataset per i commenti sono :

- **author** : Il nickname dell'utente Reddit (Redditors)
- **score** : Punteggio ottenuto equivalente al numero di upvote
- **parent_id** : ID univoco che identifica il commento
  (parent_id = name in dataset_posts)

2.1.2 Pulizia Dati

Trattandosi di dati già pre-processati,quindi con assenza di duplicati, non è stata richiesta una profonda pulizia.
Per quanto riguarda entrambi i dataset, l'unica accortezza è stata quella di rimuovere i post e gli autori che risultassero contrassegnati come [deleted] o [removed].

```
with open(input_file_path, 'w', encoding='utf-8') as output_file:
    # Lettura del file jsonl
    with open(input_file_path, 'r', encoding='utf-8') as input_file:
        for line in input_file:
            post = json.loads(line)
            selftext = post.get('selftext', '')
            if selftext and selftext not in ['[removed]', '[deleted]']:
                # Scrittura del post filtrato nel file di output
                output_file.write(json.dumps(post) + '\n')
```

Listing 2.1: Codice per lettura file JSONL e rimozione dei post rimossi o eliminati

Nel caso del dataset contenente i commenti,essendo utile solamente alla creazione del grafo per il calcolo del Pagerank, ho avuto l'accortezza,durante la creazione di nodi e archi, di rimuovere i commenti che avessero un autore contrassegnato come [deleted] per non influenzare in maniera negativa la costruzione del grafo.

```
if post_id in post_comments:
        for comment in post_comments[post_id]:
            comment_author = comment['author']
            if comment_author == '[deleted]':  # Ignora i commenti
               degli autori segnati come [deleted]
                continue
```

Listing 2.2: Rimozione Utenti bannati o non più esistenti durante la costruzione del grafo

2.2 Pre-processing Testo dei Post e Clustering

L'obiettivo di questa sezione è descrivere il processo di preprocessing e analisi dei testi dei post pubblicati su Reddit utilizzando strumenti di Natural Language Processing (NLP) come SpaCy e la tecnica di rappresentazione vettoriale TF-IDF (Term Frequency-Inverse Document Frequency). Questo processo è essenziale per preparare i dati testuali per ulteriori analisi, come il clustering, per identificare pattern e argomenti comuni nei post.

2.2.1 Pre-processing Spacy e TF-IDF dei Post

Caricamento del Modello Linguistico SpaCy

```
    nlp = spacy.load('en_core_web_sm')
```

Iniziamo caricando un modello pre-addestrato di SpaCy (`en_core_web_sm`), che include informazioni grammaticali e semantiche necessarie per analizzare il testo in lingua inglese.

Inizializzazione delle Liste per il Preprocessing

```
preprocessed_texts = []
original_posts = []
combined_texts = []
```

Creiamo tre liste vuote per memorizzare rispettivamente i testi preprocessati, i post originali e i testi combinati di titolo e corpo del post.

Lettura del File JSONL e Preprocessing del Testo

```
with open(cleaned_file_path, 'r', encoding='utf-8') as cleaned_file:
    for line in cleaned_file:
        post = json.loads(line)
        text = post['title'] + ' ' + post['selftext']
        doc = nlp(text)
        preprocessed_text = ' '.join([token.lemma_ for token in doc
            if not token.is_stop and not token.is_punct and not token.
               is_space and not token.like_num])
        preprocessed_texts.append(preprocessed_text)
        original_posts.append(post)
        combined_texts.append(text)
```

In questa fase, leggiamo i post da un file JSONL. Per ogni post:

1. Combiniamo il titolo(`title`) e il corpo del testo (`selftext`).
2. Utilizziamo SpaCy per analizzare il testo combinato.
3. Applichiamo tecniche di preprocessing come la lemmatizzazione (riduzione delle parole alla loro forma base) e la rimozione delle stop words (parole comuni e non significative), segni di punteggiatura, spazi e numeri.
4. Il testo preprocessato viene memorizzato in una lista (`preprocessed_texts`).
5. Conserviamo anche i post originali e i testi combinati per riferimento futuro.

Creazione della Matrice TF-IDF

```
vectorizer = TfidfVectorizer(max_features=5000)
tfidf_matrix = vectorizer.fit_transform(preprocessed_texts)
```

Infine, utilizziamo la tecnica TF-IDF per convertire i testi preprocessati in una rappresentazione vettoriale numerica. Il TF-IDF è utile per quantificare l'importanza di una parola in un documento rispetto a una raccolta di documenti. Usiamo `TfidfVectorizer` con un massimo di 5000 caratteristiche (parole) per rappresentare i testi.
Questo processo di preprocessing e rappresentazione vettoriale dei testi è un passaggio fondamentale nell'analisi dei dati testuali. Riduce il rumore nei dati, facilita l'estrazione di informazioni rilevanti e prepara i testi alla fase di clustering.

2.2.2 Clustering con K-means

Dopo aver preprocessato e rappresentato vettorialmente i testi dei post Reddit, ho utilizzato l'algoritmo di clustering K-Means per raggruppare i post in base a somiglianze testuali. Questa tecnica ha permesso di identificare gruppi di post con contenuti simili, facilitando l'analisi tematica.

Definizione e Applicazione dell'Algoritmo K-Means

```
k = 2
kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(tfidf_matrix)
```

Definiamo il numero di cluster (`k`) da identificare con K-Means, in questo caso ho optato per `k = 2` perchè dopo svariati tentativi è stato il valore migliore. Inizializziamo l'algoritmo con `KMeans(n_clusters=k, random_state=42)` e lo si applica alla matrice TF-IDF precedentemente calcolata utilizzando il metodo `fit()`.

Visualizzazione dei Cluster

```
cluster_post_counts = [0] * k
cluster_samples = [[] for _ in range(k)]

for idx, label in enumerate(kmeans.labels_):
    cluster_post_counts[label] += 1
    if len(cluster_samples[label]) < 5:
        cluster_samples[label].append(original_posts[idx])
```

L'algoritmo K-Means mi ha permesso di raggruppare i post Reddit in due cluster distinti, basati sulle caratteristiche testuali estratte con TF-IDF. Analizzando il numero di post in ciascun cluster e alcuni esempi rappresentativi, è possibile iniziare a comprendere le tematiche predominanti e la distribuzione dei contenuti all'interno del dataset.

2.3 Sentiment Analysis : Metodologia

La sentiment analysis è una tecnica di elaborazione del linguaggio naturale utilizzata per determinare l'atteggiamento o il tono emotivo espresso in un testo. In questa sezione

viene descritto come ho applicato la sentiment analysis ai post Reddit per valutare i sentimenti e la soggettività dei testi, utilizzando la libreria TextBlob.

Funzione per Analizzare il Sentiment di un Testo

```
def analyze_sentiment(text):
    blob = TextBlob(text)
    sentiment_score = blob.sentiment.polarity
    return sentiment_score
```

Questa funzione utilizza TextBlob per calcolare il punteggio di polarità di un testo, che varia da -1 (molto negativo) a 1 (molto positivo).

Funzione per Calcolare la Soggettività di un Testo

```
def analyze_subjectivity(text):
    blob = TextBlob(text)
    subjectivity_score = blob.sentiment.subjectivity
    return subjectivity_score
```

Similmente, questa funzione utilizza TextBlob per calcolare il punteggio di soggettività di un testo, che varia da 0 (molto oggettivo) a 1 (molto soggettivo).

Funzione per Valutare il Sentiment

```
def valutazione_sentimento(sentimento):
    if sentimento >= 0.6:
        return "Molto positivo"
    elif 0.2 <= sentimento < 0.6:
        return "Positivo"
    elif -0.2 <= sentimento < 0.2:
        return "Neutro"
    elif -0.6 <= sentimento < -0.2:
        return "Negativo"
    else:
        return "Molto negativo"
```

Questa funzione classifica il punteggio di polarità in cinque categorie: ”Molto positivo”, ”Positivo”, ”Neutro”, ”Negativo” e ”Molto negativo”, in base ai valori del sentiment.

Estrazione dei Campi Necessari

```
data['Testo_completo'] = data['title'] + ' ' + data['selftext']
```

Il titolo(`title`) e il corpo del post (`selftext`) sono stati uniti per formare un testo completo per ogni post. Così facendo è stato possiible ottenere una valutazione anche per quei post con corpo vuoto.

Applicazione della Funzione di Analisi del Sentiment

```
data['Sentiment'] = data['Testo_completo'].apply(analyze_sentiment)
data['Subjectivity'] = data['Testo_completo'].apply(
    analyze_subjectivity)
```

Viene poi applicata la funzione di analisi del sentiment e quella di soggettività a ciascun post nel dataset, aggiungendo i risultati come nuove colonne (`Sentiment` e `Subjectivity`) nel DataFrame.

Valutazione del Sentiment

```
data['Valutazione_sentimento'] = data['Sentiment'].apply(
    valutazione_sentimento)
```

Infine, classifichiamo ciascun valore di sentiment utilizzando la funzione di valutazione del sentiment, aggiungendo i risultati come una nuova colonna (`Valutazione_sentimento`) nel DataFrame.

L'analisi del sentiment mi ha permesso di comprendere meglio le emozioni e le opinioni espresse nei post su Reddit. Utilizzando TextBlob per calcolare i punteggi di polarità e soggettività, e una funzione di classificazione per valutare questi punteggi, è possibile identificare facilmente i post con sentimenti positivi, negativi o neutrali, migliorando così l'analisi complessiva dei dati testuali.

2.4 Calcolo Opinion Leader Score

In questa sezione, si descrive il processo per il calcolo dell'OLS (Opinion Leader Score) per ogni autore dei post su Reddit. Questi punteggi aiutano a quantificare l'engagement e l'influenza degli autori all'interno della comunità.
Ho strutturato l'algoritmo basandomi sulle formule fornite nel paper [6].

$$TES(K,i) = l_i + 5 \times r_{ti} + 15 \times s_{ri} + 20 \times lr_i + 25 \times sq_i + 30 \times lq_i \tag{2.1}$$

$$AXES(T,K) = \sum_{i=1}^{N} TES(K,i) \tag{2.2}$$

$$OLS(T,K) = \text{distance}(0,K) - \text{distance}(K,K) \tag{2.3}$$

$$OLS(T,K) = (\cos(\beta) - \sin(\beta)) \times \sqrt{N^2 K} + AXES(T,K)^2 \tag{2.4}$$

Funzione per il Calcolo del TES

```
def calculate_TES(post):
    upvote_weight = 0.5
    comments_weight = 0.5
    TES = (upvote_weight * post['score']) + (comments_weight * post['
        num_comments'])
    return TES
```

Questa funzione calcola il TES di un post, combinando il punteggio del post (`score`) e il numero di commenti (`num_comments`) con pesi uguali di 0.5.

Inizializzazione dei Dizionari e delle Liste per i Valori

```
autore_TES = {}
score_values = []
num_comments_values = []
aes_values = []
ols_values = []
```

Inizializziamo i dizionari e le liste per memorizzare i valori necessari per il calcolo del TES, AES (Account Engagement Score) e OLS.

Parsing del File JSONL

```
with open(dataset_post, 'r', encoding='utf-8') as cleaned_file:
    for line in cleaned_file:
        post = json.loads(line)
        autore = post['author']
        TES = calculate_TES(post)

        if autore in autore_TES:
            autore_TES[autore].append(TES)
        else:
            autore_TES[autore] = [TES]

        score_values.append(post['score'])
        num_comments_values.append(post['num_comments'])
        aes_values.append(TES)
```

Leggiamo i dati dei post Reddit da un file JSONL. Per ogni post:

1. Calcoliamo il TES utilizzando la funzione `calculate_TES`.
2. Aggiungiamo il TES al dizionario `autore_TES` per l'autore corrispondente.
3. Memorizziamo i valori di `score`, `num_comments` e TES nelle rispettive liste.

Calcolo dell'AES e dell'OLS per Ogni Autore

```
autore_AES = {}
autore_OLS = {}

for autore, TES_list in autore_TES.items():
    num_posts_autore = len(TES_list)
    AES_autore = sum(TES_list)
    autore_AES[autore] = AES_autore

    cos_beta = math.cos(math.radians(45))
    sin_beta = math.sin(math.radians(45))
    term1 = (cos_beta - sin_beta) * math.sqrt(num_posts_autore)
    term2 = AES_autore ** 2
    OLS_T_K = term1 + term2
    autore_OLS[autore] = OLS_T_K
```

Per ogni autore:

1. Calcoliamo il numero di post (`num_posts_autore`) e il totale del TES (`AES_autore`).
2. Memorizziamo l'AES nel dizionario `autore_AES`.
3. Memorizziamo l'OLS nel dizionario `autore_OLS`.

Il calcolo dell'OLS permette di identificare e quantificare l'influenza degli autori sui post Reddit. Utilizzando queste metriche, possiamo determinare quali autori hanno un maggiore impatto sulla comunità in termini di engagement e leadership di opinione.

2.5 Costruzione Grafo

In questa sezione, si descriveil processo di costruzione del grafo basato sulle interazioni tra autori di post e commenti su Reddit, e successivamente il calcolo del PageRank per determinare gli autori più influenti.
Rispetto ai metodi precedenti in questo caso viene utilizzatto anche il dataset contenente i commenti che per poter essere elaborato senza procurare crash alla macchina ho dovuto applicare dei parametri di filtro, in modo da portare a termine l'analisi.

Parametri di Filtro

```
min_post_count = 3  # Numero minimo di post per autore
min_total_score = 10  # Punteggio minimo totale per autore
```

Si definiscono inizialmente i criteri di filtro per selezionare gli autori:

- Un autore deve avere almeno 3 post.
- Il punteggio totale dei post di un autore deve essere superiore a 10.

Filtraggio Incrementale dei Commenti

```
def load_and_filter_comments(filename, eligible_users):
    filtered_comments = []
    with open(filename, 'r', encoding='utf-8') as file:
        for line in file:
            comment = json.loads(line)
            if comment['author'] in eligible_users:
                filtered_comments.append(comment)
    return filtered_comments
```

Funzione per caricare e filtrare i commenti, includendo solo quelli degli autori che soddisfano i criteri di filtro.

Filtraggio degli Utenti

```
eligible_users = defaultdict(lambda: {'num_posts': 0, 'total_score':
    0})
for post in posts:
    author = post['author']
    score = post['score']
    eligible_users[author]['num_posts'] += 1
    eligible_users[author]['total_score'] += score

eligible_users = {author: info for author, info in eligible_users.items
    () if info['num_posts'] >= min_post_count and info['total_score'] >
    min_total_score}
```

Si filtrano gli utenti che soddisfano i criteri, memorizzando il numero di post e il punteggio totale per ciascun autore.

Caricamento e Filtraggio Incrementale dei Commenti

```
filtered_comments = load_and_filter_comments(dataset_commenti,
    eligible_users)
```

Carichiamo e filtriamo i commenti utilizzando la funzione precedentemente definita.

Creazione di Dizionari di Mappatura

```
post_info = {post['name']: {'author': post['author'], 'score': post['
    score'], 'num_comments': 0} for post in posts if post['author'] in
    eligible_users}

post_comments = defaultdict(list)
for comment in filtered_comments:
    post_id = comment['parent_id']
    post_comments[post_id].append(comment)
```

Creiamo due dizionari:

- `post_info`: Mappa ogni post al suo autore e punteggio.
- `post_comments`: Mappa ogni post ai suoi commenti.

Creazione del Grafo

```
G = nx.DiGraph()

for post_id, info in post_info.items():
    author = info['author']
    if author == '[deleted]':
        continue
    score = info['score']
    num_comments = info['num_comments']
    weight = score + num_comments

    if post_id in post_comments:
        for comment in post_comments[post_id]:
            comment_author = comment['author']
            if comment_author == '[deleted]':
                continue
            if author != comment_author:
                if G.has_edge(author, comment_author):
                    G[author][comment_author]['weight'] += weight
                else:
                    G.add_edge(author, comment_author, weight=weight)
```

Creiamo un grafo diretto dove:

- I nodi rappresentano gli autori.
- Gli archi rappresentano le interazioni tra gli autori (post e commenti), con pesi basati sul punteggio del post e il numero di commenti.

Aggiunta di Nodi Isolati

```
for post_id, info in post_info.items():
    author = info['author']
    if author == '[deleted]':
        continue
    if post_id not in post_comments and author not in G.nodes:
        G.add_node(author)
```

Aggiungiamo i nodi degli autori che non sono direttamente collegati a nessun altro nodo.

2.5.1 Algoritmo Pagerank

```
top_pagerank = nx.pagerank(G, weight='weight')
top_authors = sorted(top_pagerank.items(), key=lambda x: x[1], reverse=
    True)[:50]
```

Calcoliamo il PageRank dei nodi nel grafo, ordinando gli autori in base al loro punteggio di PageRank e selezionando i top 50.

Costruendo un grafo delle interazioni tra autori di post e commenti, e calcolando il PageRank, è possibile identificare gli autori più influenti all'interno di Reddit inteso come comunità. Questo approccio ci permette di comprendere meglio le dinamiche di interazione e influenza tra gli utenti.

Capitolo 3

Analisi

In questa sezione verranno mostrati e commentati i risultati ottenuti durante la mia analisi.

3.1 Numero di Post

Nel grafico 3.1 e nella tabella 3.1 viene mostrato il numero di post collezionati per ogni Subreddit. Il numero totale di post è 152808.

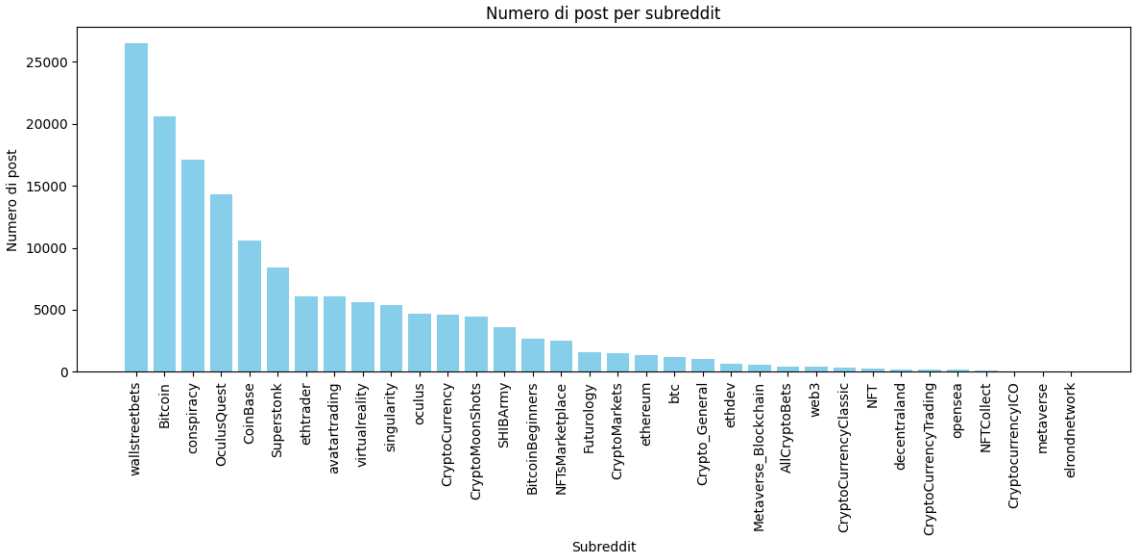

Figura 3.1: Istogramma numero di post per ogni Subreddit

Subreddit	**Numero Post**
Wallstreetbets	26485
Bitcoin	20672
Conspiracy	17128
OculusQuest	14303
CoinBase	10568

Tabella 3.1: Top 5 Subreddit per numero di post collezionati

È utile analizzare almeno i primi 5 Subreddit per quantitativo di post in modo da comprenderne la natura e il loro ambito :

- *Wallstreetbets*: Subreddit molto famoso per post divertenti ed esagerati su investimenti in borsa con attenzione ad opzioni ed investimenti rischiosi. Gli utenti condividono strategie di investimento (caso Gamestop), successi e fallimenti in maniera umoristica. Conta 16M di utenti.

- *Bitcoin* : è un Subreddit dedicato alla cryptovaluta più famosa al mondo, gli utenti discutono di notizie, previsioni di prezzo e sicurezza di Bitcoin. Conta circa 6.5M di utenti.

- *Conspiracy*: è un subreddit dedicato alla discussione di teorie del complotto che spaziano da eventi storici fino ad operazioni governative segrete. Nel contesto delle criptovalute gli utenti potrebbero discutere di presunti complotti rigurdante la manipolazione del mercato delle crypto e speculazioni. Conta circa 2M di utenti.

- *OculusQuest* : Subreddit dedicato agli utenti dei visori Oculus Quest per la realtà virtuale, gli utenti discutono di giochi, esperienze VR e in ambito criptovalute potrebbero esserci discussioni inerenti ai metodi di pagamento per alcuni acquisti in giochi VR oppure discussioni su progetti di sviluppo che integrano criptovalute o blockchain con la tegnologia VR (NFT).

- *Coinbase* : Subreddit dedicato ali utenti della piattaforma di scambio di criptovalute. Gi utenti parlano di questioni tecniche, problemi di sicurezza, strategie di trading ma è anche fonte di notizie sull'azienda e sulle criptovalute in generale. Conta più di 200k follower.

3.2 Clustering

In questa sezione viene mostrato il risultato ottenuto dopo il pre-processing del testo dei post tramite SpaCy e TF-IDF ai fini della clusterizzazione.
Nelle figure 3.2 e 3.3 è stata utilizzata una rappresentazione sottoforma di word cloud per dare un impatto visivo migliore alle parole più utilizzate all'interno dei post per entrambi i cluster, in questo modo è possibile categorizzare i documenti in maniera più semplice.
La stessa rappresentazione viene mostrata anche sottoforma di grafico a barre nelle figure 3.4 e 3.5
Nel grafico 3.6 è possibile visualizzare i singoli documenti e la loro appartenenza.

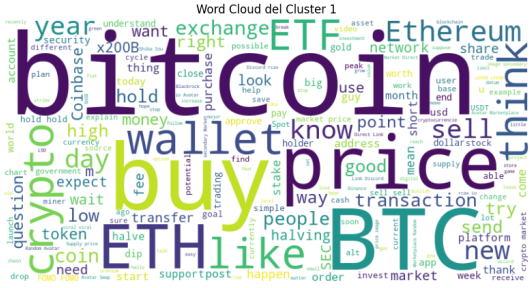

Figura 3.2: Word Cloud per Cluster 1

Figura 3.3: Word Cloud per Cluster 2

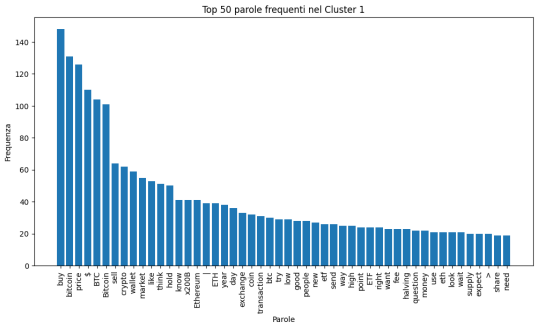

Figura 3.4: Top 50 parole nel Cluster 1

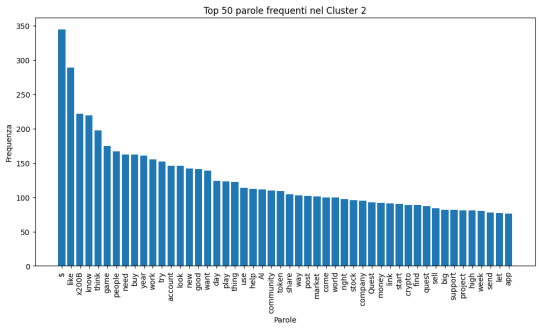

Figura 3.5: Top 50 parole nel Cluster 2

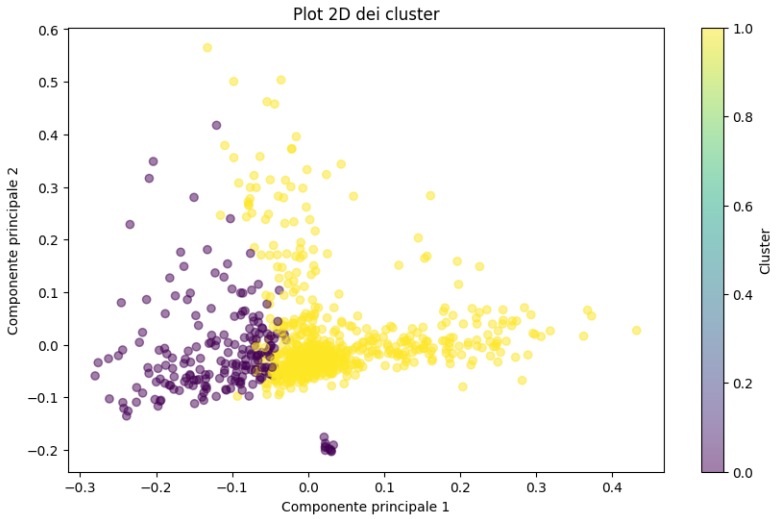

Figura 3.6: Plot 2D documenti nei cluster

Dai grafici si può notare come il pre-processing del testo abbia aiutato la fase di clustering perchè l'algoritmo K-mean è riuscito a dividere abbastanza bene i documenti. Infatti nel primo cluster sono molto frequenti le parole "bitcoin","wallet","eth" che fanno pensare che i post siano appartenenti ai subreddit relativi al mondo delle cryptovalute mentre nel secondo cluster ci sono parole come "Quest","game", "VR" relative più alle tecnologie connesse, in particolare al concetto di metaverso.

3.3 Analisi delle frequenze

In questa sezione verranno mostrati i risultati ottenuti nell'analisi delle frequenze all'interno dei post di quelle parole chiave che sono risultate le più frequenti durante la fase di pre-processing e clusterizzazione.
In particolare le parole utilizzate sono state salvate manualmente in un array denominato *keyword*.

```
keyword = [
 "","wallet","crypto","BTC","AI","bitcoin","Coinbase","transaction",
 "token","game","Quest","VR","Meta","ethereum"
]
```

Per ogni parola è stato tracciato un grafico lineare ed un grafico a barre che mostrano l'andamento del numero di post contenente quella parola chiave nel tempo.

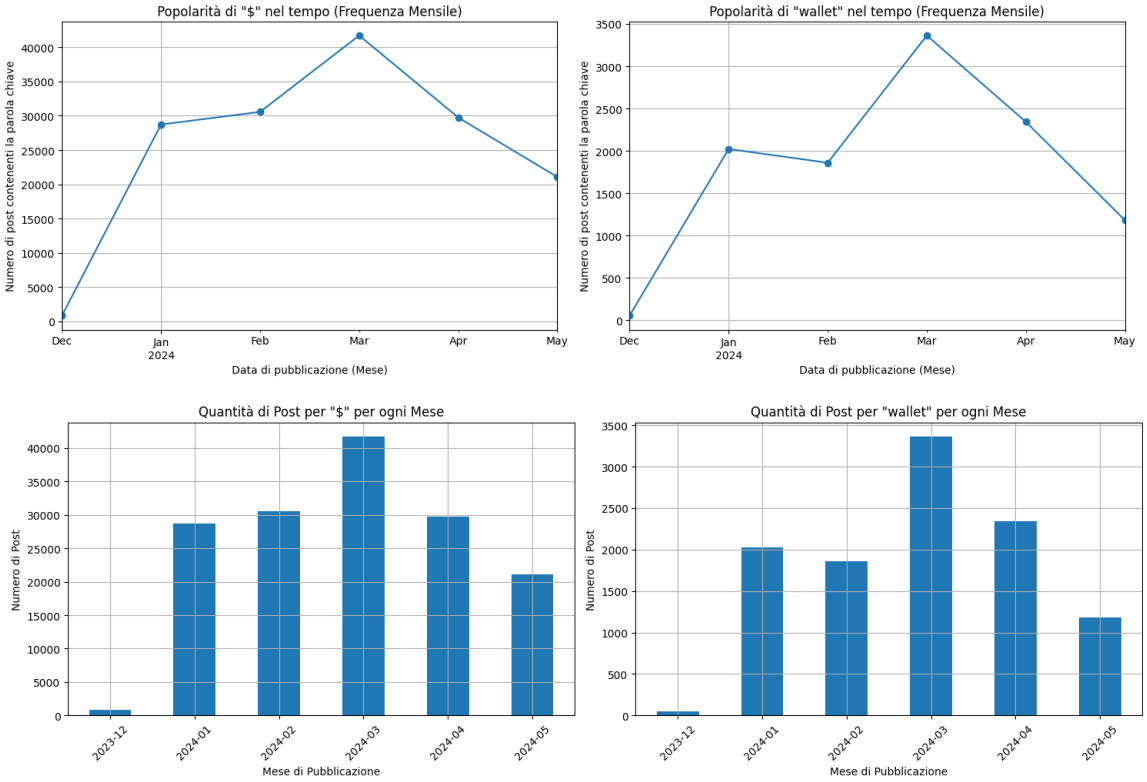

Figura 3.7: Grafici per $ e wallet

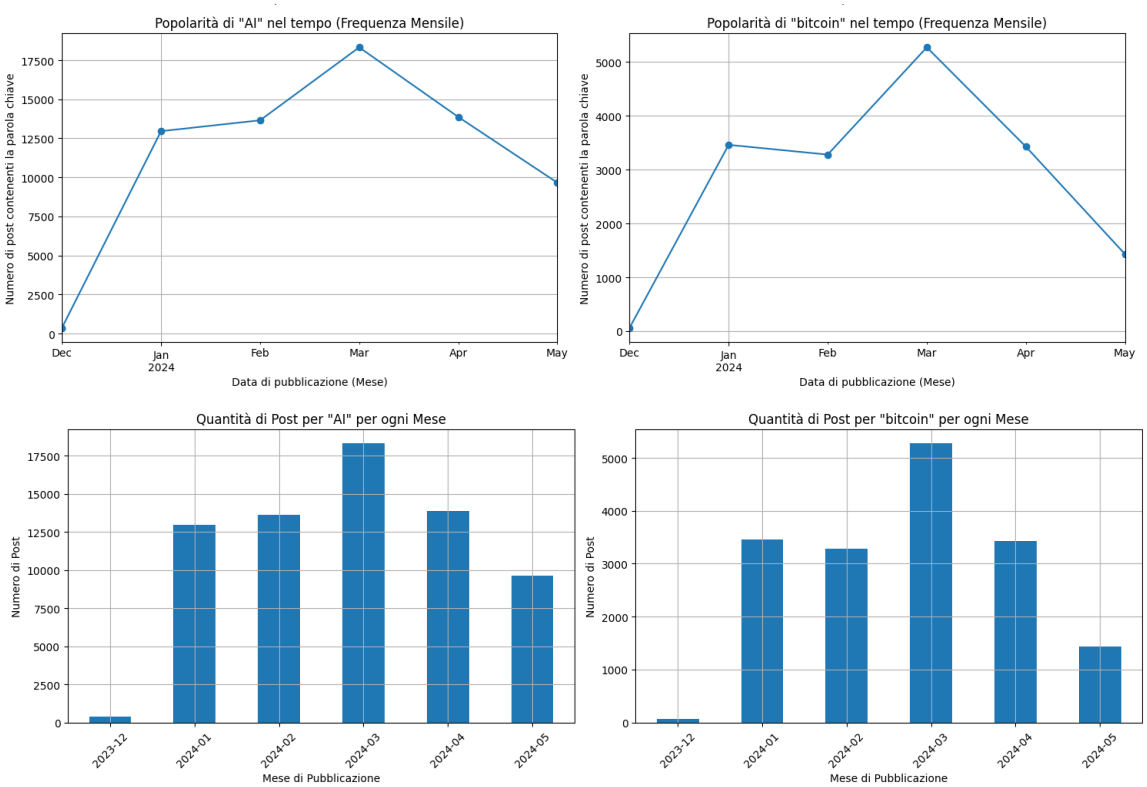

Figura 3.8: Grafici per Ai e Bitcoin

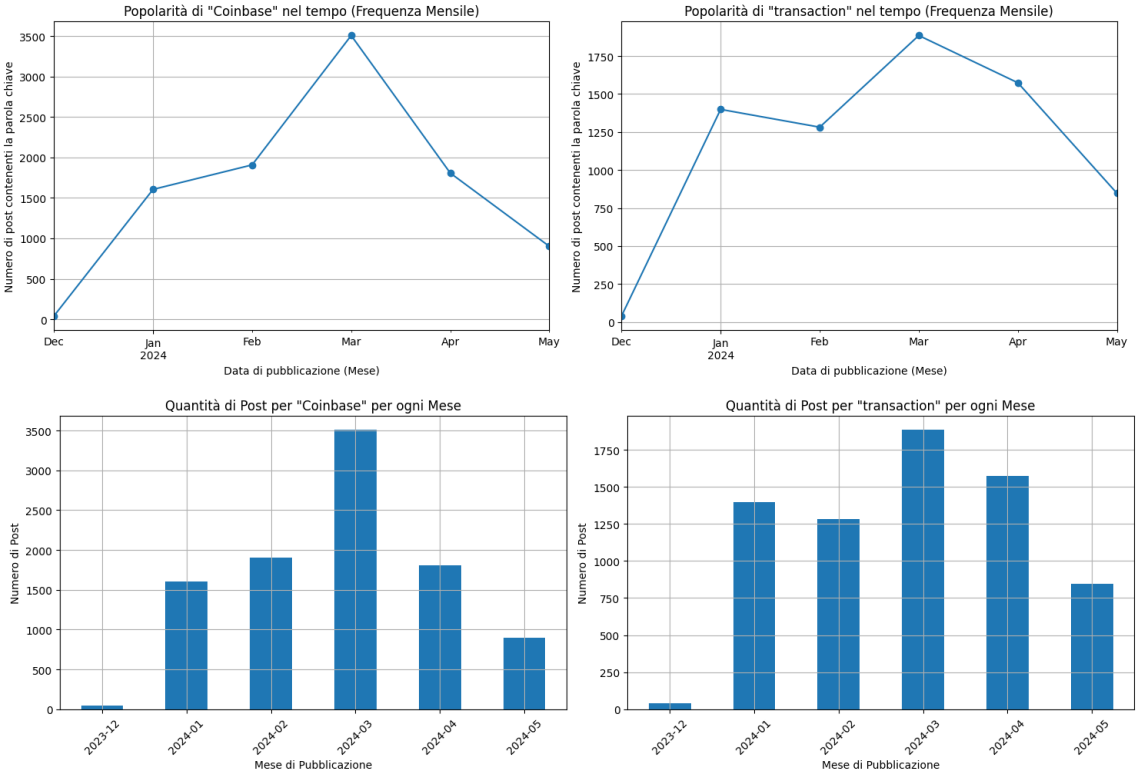

Figura 3.9: Grafici per Coinbase e transaction

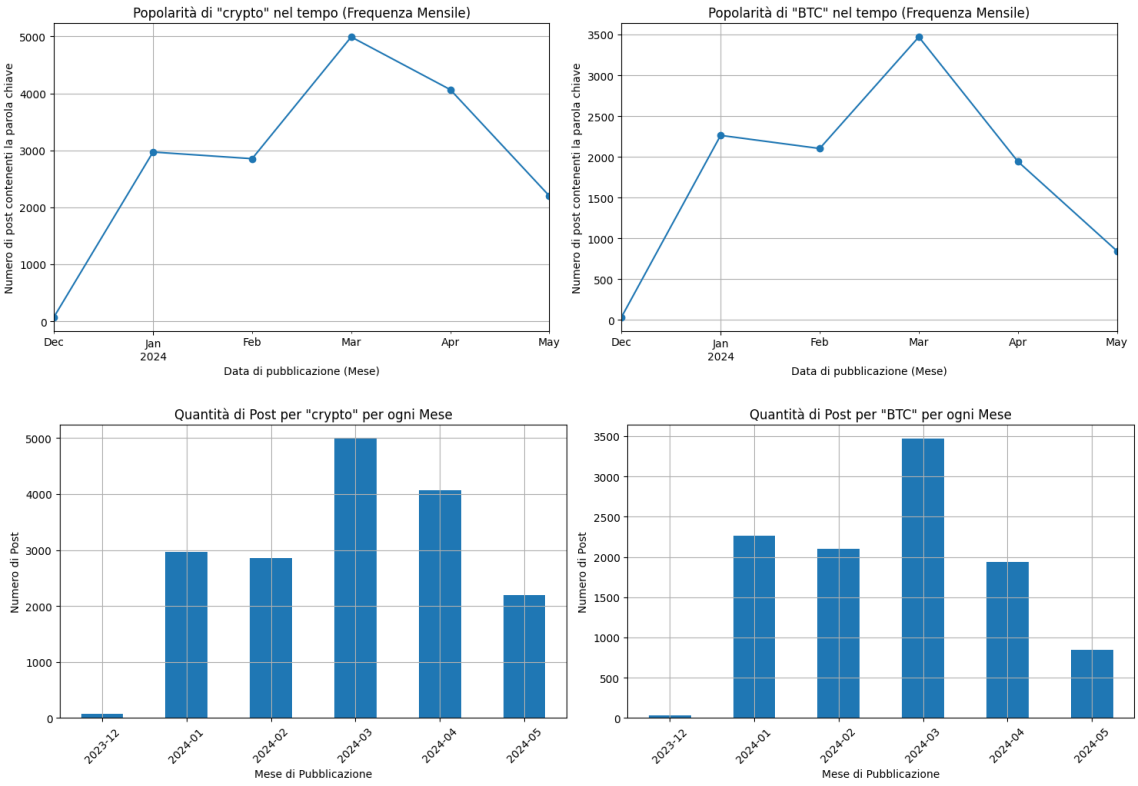

Figura 3.10: Grafici per crypto e BTC

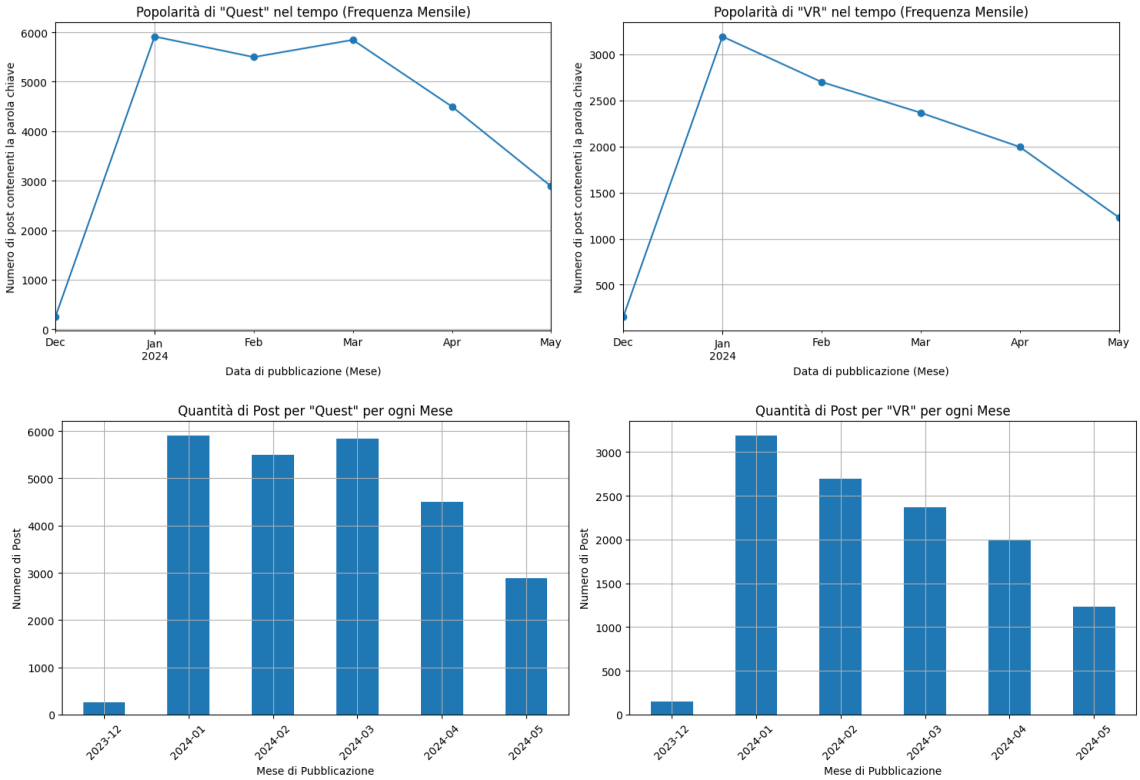

Figura 3.11: Grafici per quest e VR

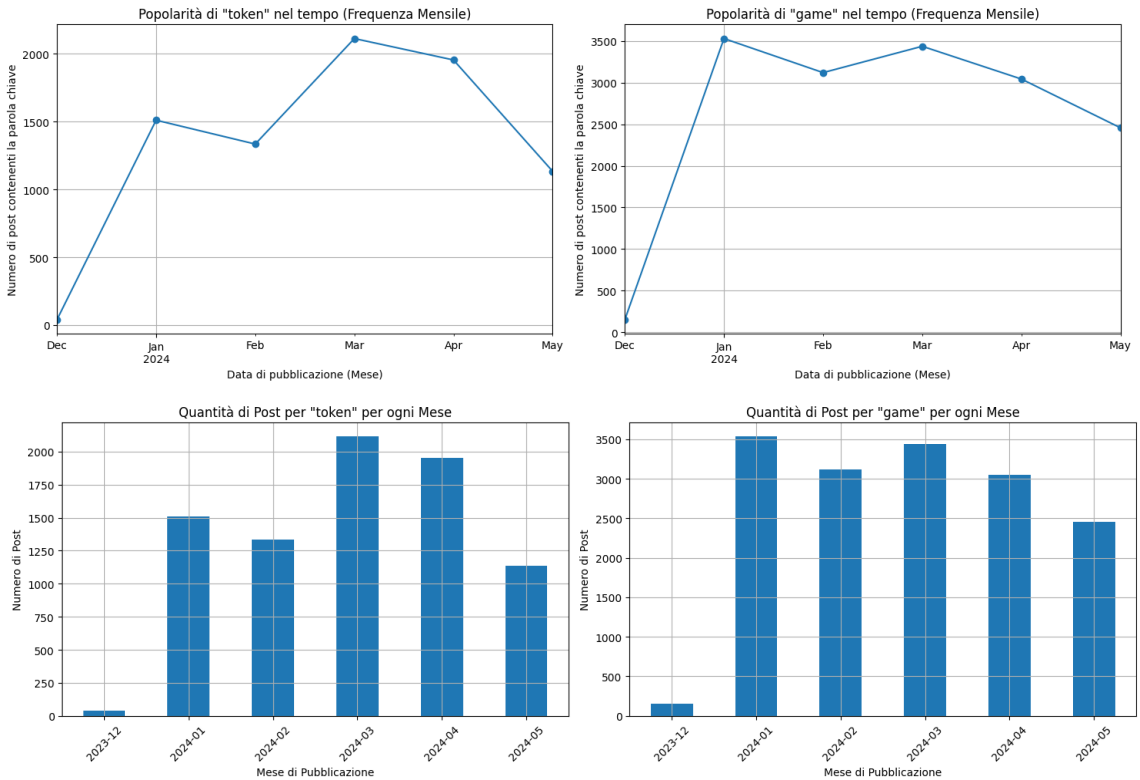

Figura 3.12: Grafici per token e game

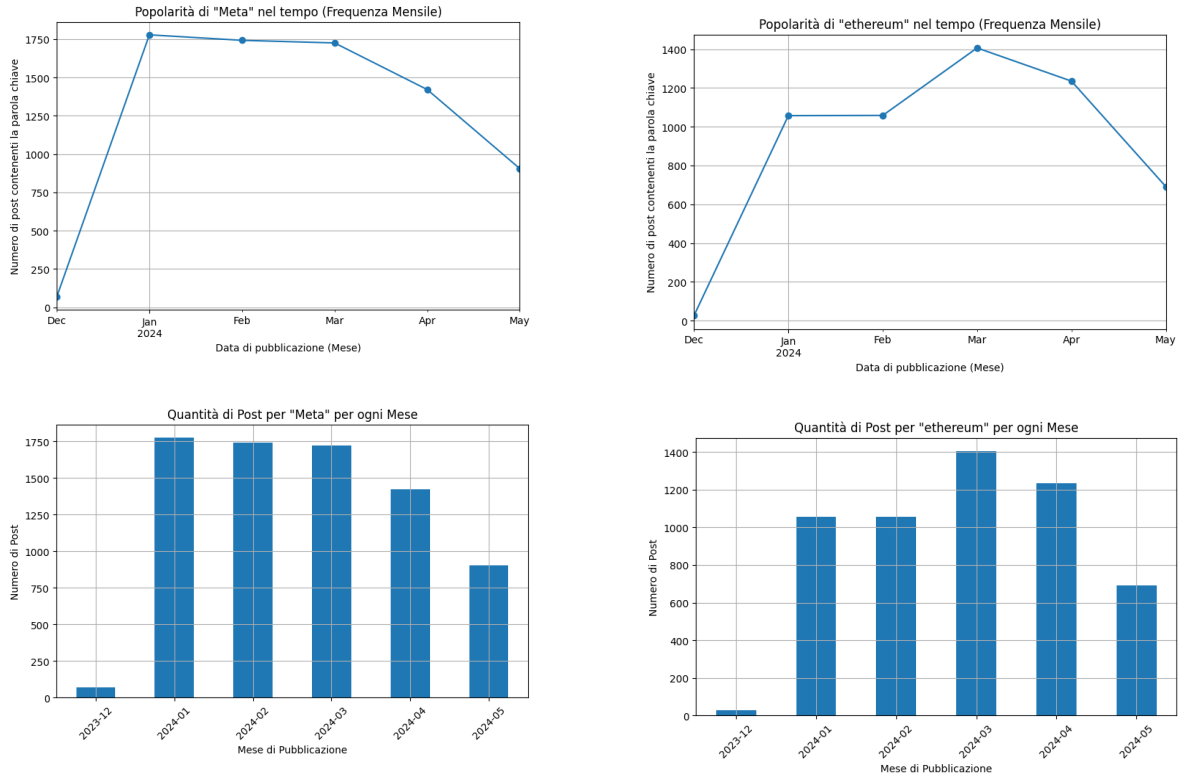

Figura 3.13: Grafici per Meta e Ethereum

Dai grafici ottenuti si può evidenziare come per quasi tutti i termini ci sia stata una crescita particolare nel periodo di Marzo 2024, soprattutto per quanto riguarda le parole: wallet 3.7, bitcoin 3.8, Coinbase 3.9, crypto, BTC 3.10 ed ethereum 3.14 che hanno registrato un incremento significativo di post, rispetto ai mesi precedenti, proprio in quel periodo.Essendo termini facenti parte dell'ambito criptovalute è ragionevole pensare che questo possa essere accaduto per via di alcuni eventi capitati tra i mesi di Marzo e Aprile in particolare l'Halving di Bitcoin e l'aggiornamento Dencun di Ethereum che potrebbero aver portato ad un maggior numero di discussioni in quel periodo.

Halving Bitcoin

L'*halving* di Bitcoin è un evento programmato nel protocollo Bitcoin che riduce del 50% la ricompensa per i miner che verificano le transazioni sulla blockchain di Bitcoin. Questo evento avviene approssimativamente ogni 210.000 blocchi, che corrispondono a circa ogni quattro anni. La diminuzione della ricompensa per i miner comporta una riduzione del tasso di creazione di nuovi Bitcoin. Questo contribuisce a rendere Bitcoin una risorsa deflazionistica, poiché la sua offerta totale è limitata a 21 milioni di unità. Il termine della fase di dimezzamento è avvenuto il giorno 20 Aprile 2024 ma va considerata una fase di pre-halving in cui gli utenti potrebbero aver discusso in preparazione dell'evento.

Dencun Ethereum

Il *Dencun* è un importante aggiornamento per la blockchain di Ethereum che mira a migliorare la scalabilità e la sostenibilità della rete. Questo aggiornamento fa parte del percorso evolutivo di Ethereum per passare da una blockchain di Proof of Work (PoW) a una di Proof of Stake (PoS) così da andare a migliorare la capacità della rete per gestire un numero maggiore di transazioni al secondo e inoltre ridurre il consumo energetico. Adottando PoS anche la sicurezza sarà migliorata. L'aggiornamento è avvenuto in data 13 Marzo.

Per verificare se effettivamente questi due tipi di eventi avessero influenzato la comunità Reddit, ho aggiunto alle keywords altri due termini : *halving* e *dencun*
Ho tracciato poi gli stessi grafici generati per le altre parole chiave.

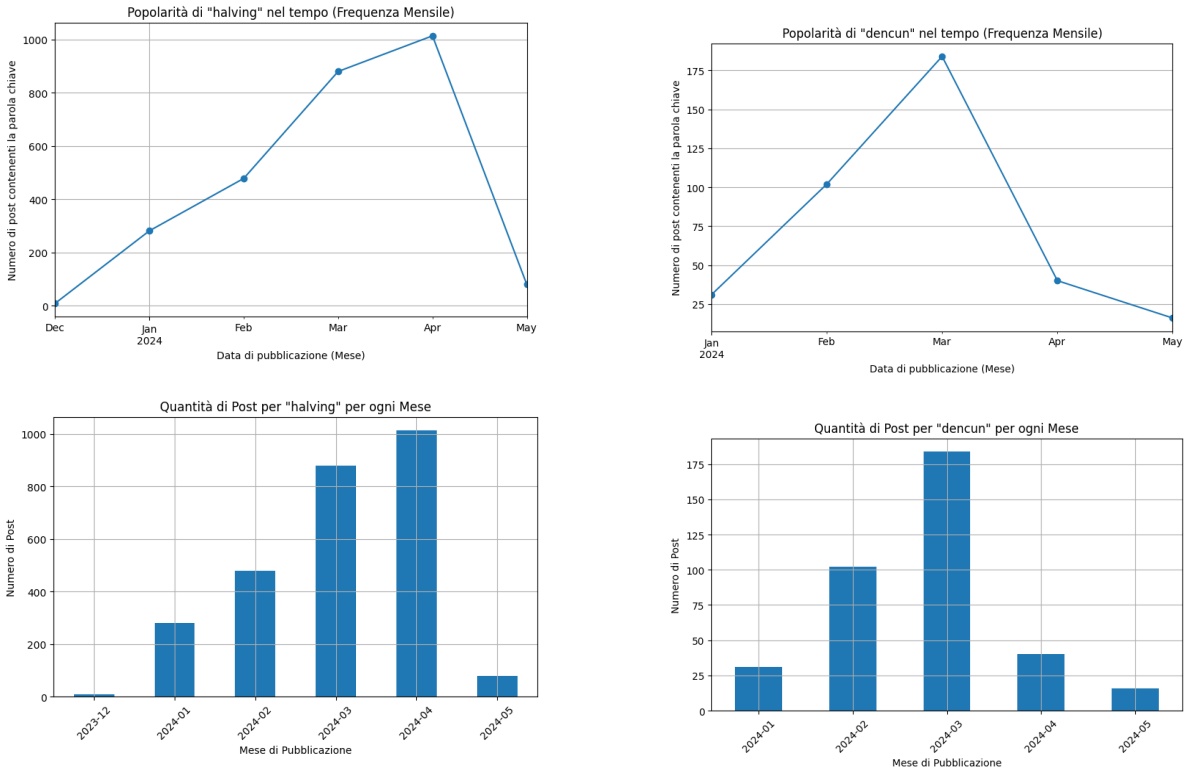

Figura 3.14: Grafici per halving e dencun

Effettivamente dai grafici riusciamo a capire che la parola *halving* è presente in molti post del mese di Aprile, mese in cui effettivamente è avvenuto il dimezzamento di Bitcoin. Invece la parola *dencun* è più frequente nei post di Marzo ovvero il mese in cui è stato rilasciato l'aggiornamento alla blockchain di ethereum.
Per confermare la mia ipotesi ho tracciato un grafico giornaliero che indicasse il numero di post pubblicati per le due parole chiave.
Per la parola *dencun* ho tracciato il grafico relativo al mese di Marzo 2024. Mentre per la parola *halving* è stato generato il graficp per il mese di Aprile 2024.

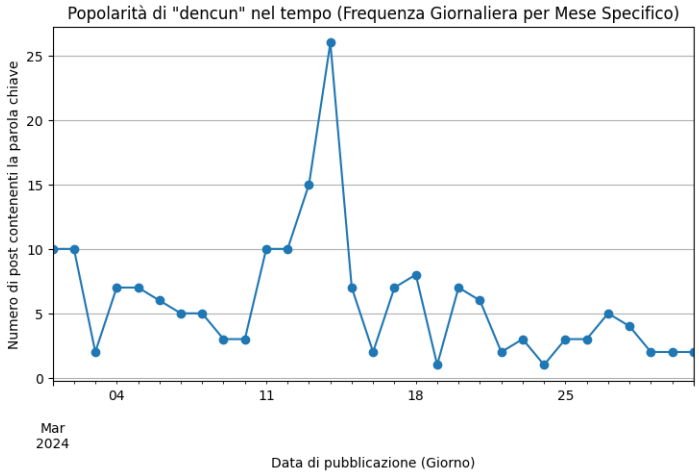

Figura 3.15: Andamento post contenenti la parola chiave *dencun*

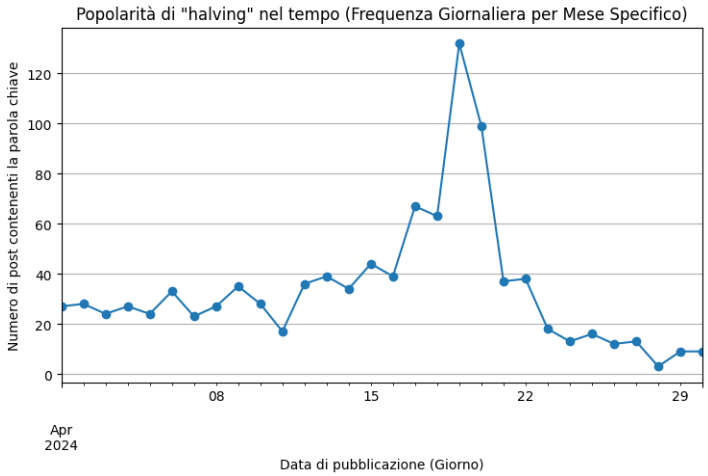

Figura 3.16: Andamento post contenenti la parola chiave *halving*

Nei grafici è evidente come nel giorno in cui si è verificato l'evento c'è un picco di post pubblicati, rispettivamente il giorno 13 Marzo 2024 per l'aggiornamento Dencun della blockchain di ehtereum 3.15 e il giorno 20 Aprile 2024, data di avvenuto dimezzamento del Bitcoin 3.16.
Questo dimostra come la comunità Reddit, nello specifico quegli utenti che sono interessati all'argomento criptovalute, genera molte più discussioni nei giorni in cui si verificano eventi signficativi.

Ho provato inoltre a tracciare la stessa tipologia di grafico anche per altre keywords che potessero essere collegate ai due eventi. In particolare il grafico di *ethereum*, *bitcoin* e *BTC*

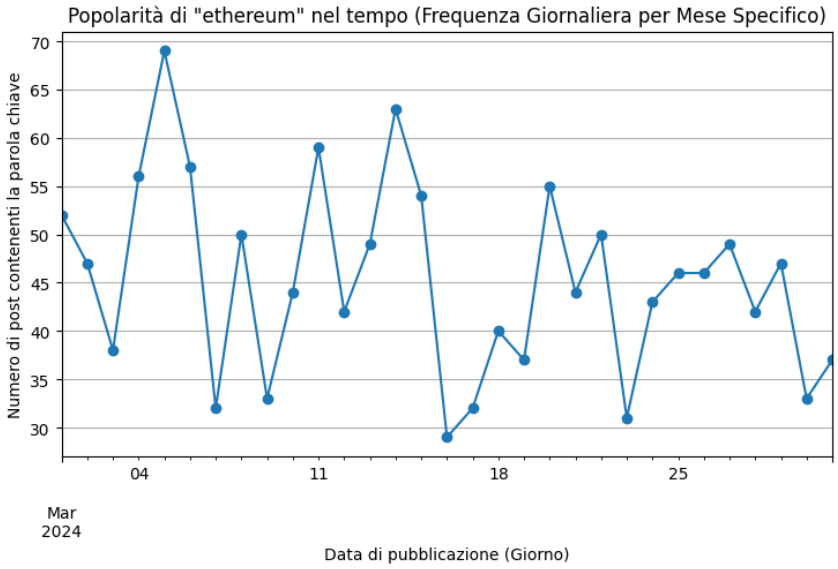

Figura 3.17: Andamento post contenenti la parola chiave *ethereum* (Marzo 2024)

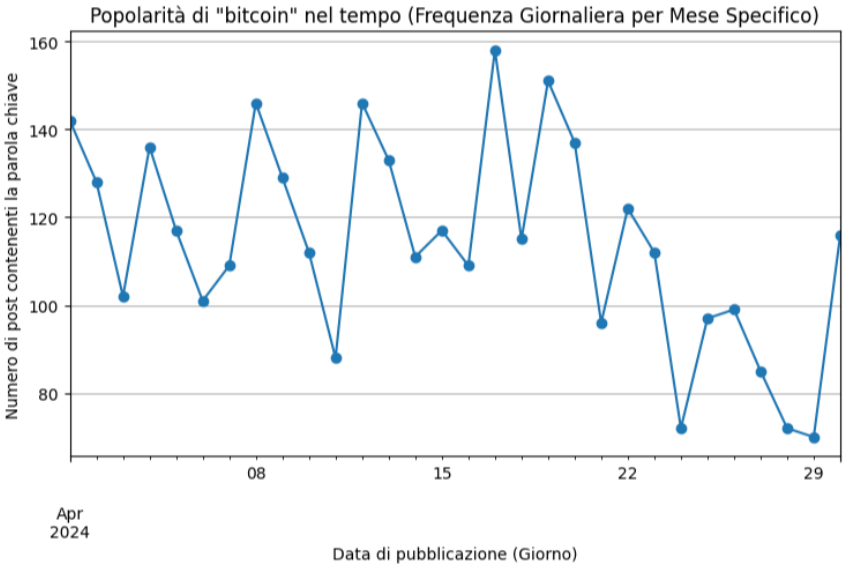

Figura 3.18: Andamento post contenenti la parola chiave *bitcoin* (Aprile 2024)

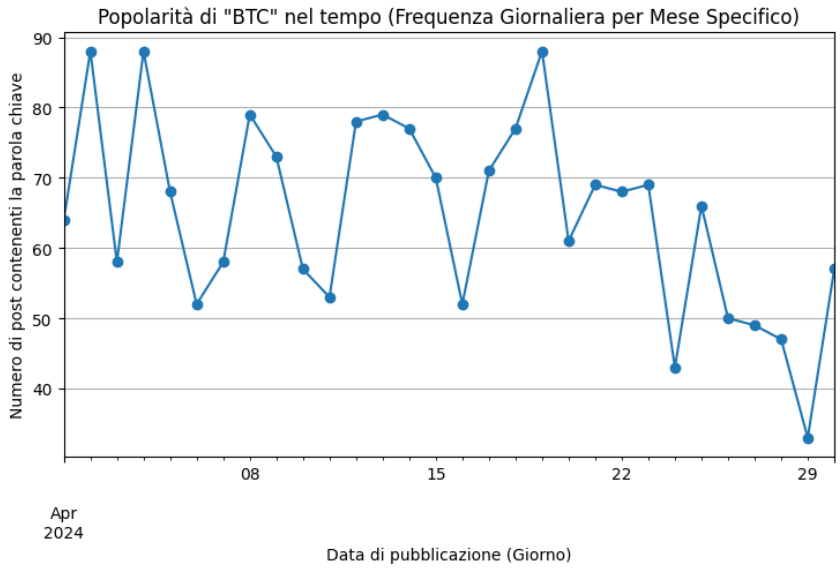

Figura 3.19: Andamento post contenenti la parola chiave *BTC* (Aprile 2024)

Nel caso del grafico 3.17 per la parola *ethereum* non sembra esserci nessun comportamento rilevante a parte una lieve crescita nel periodo che va dal 13 Marzo in poi.
Per quanto riguarda invece le parole *bitcoin* e *BTC* in entrambi i grafici 3.18 e 3.19 viene registrato un picco nel giorno dell'halving (20 Aprile 2024). Ciò fa pensare che nei post pubblicati in quel giorno è molto probabile che siano state usate queste tre parole chiave nello stesso momento in diversi post per parlare dell'evento.

3.4 Sentiment Analysis

Dopo aver analizzato la frequenza delle parole chiave all'interno dei post ho deciso di valutare la polarità e la soggettività dei post contenenti i termini all'interno dell'array *keyword* per cercare di capire le opinioni degli utenti su determinati argomenti utilizzando la consolidata libreria *TextBlob* di Python. I valori del sentimento sono stati divisi in Molto negativo, Negativo, Neutro, Positivo, Molto positivo.
Come è possibile notare dalla figura 3.20 nel campione di post presi in esame, alla maggior parte viene assegnata una valutazione neutra ma risultano essere molto di più i post positivi di quelli negativi. Nelle figura 3.21 viene mostrato l'andamento giornaliero della polarità per numero di post nel periodo di riferimento. Si può notare un certo equilbrio tra post positivi,neutri e negativi.

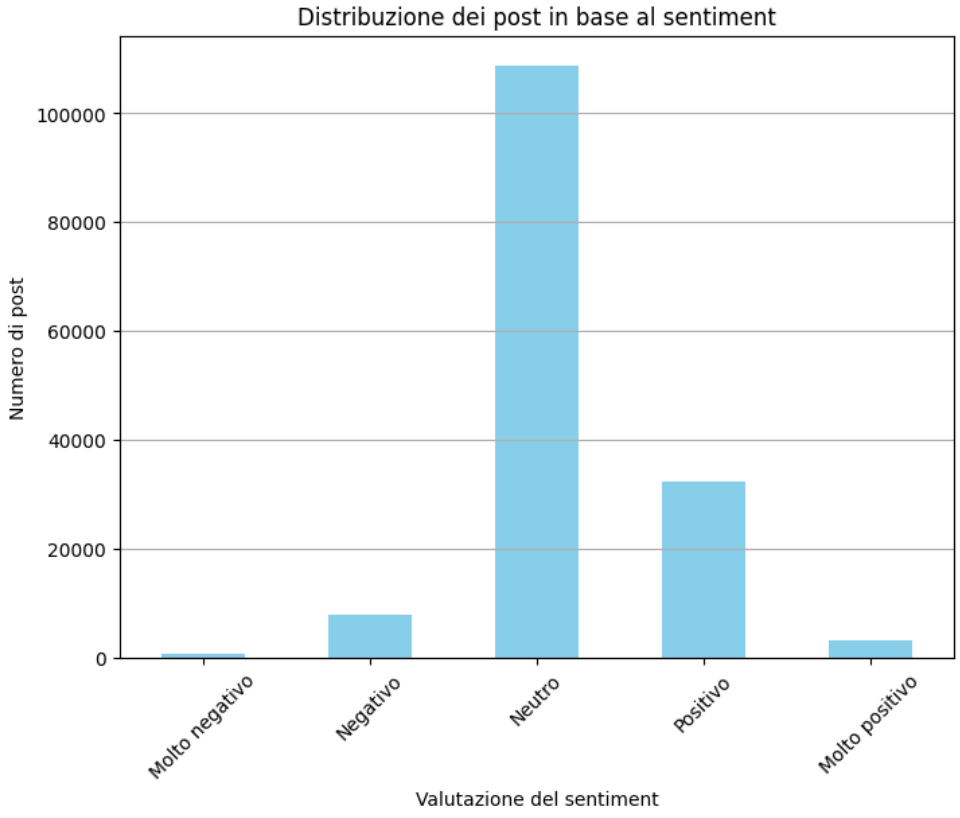

Figura 3.20: Distribuzione della polarità

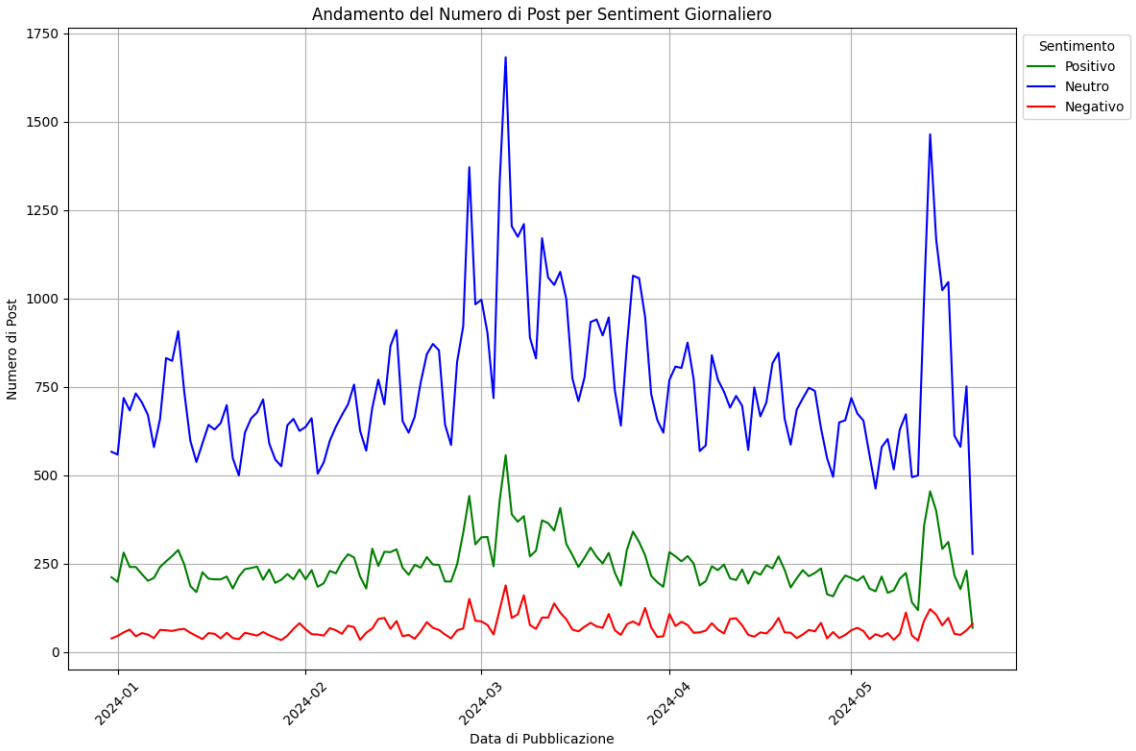

Figura 3.21: Andamento polarità giornaliero

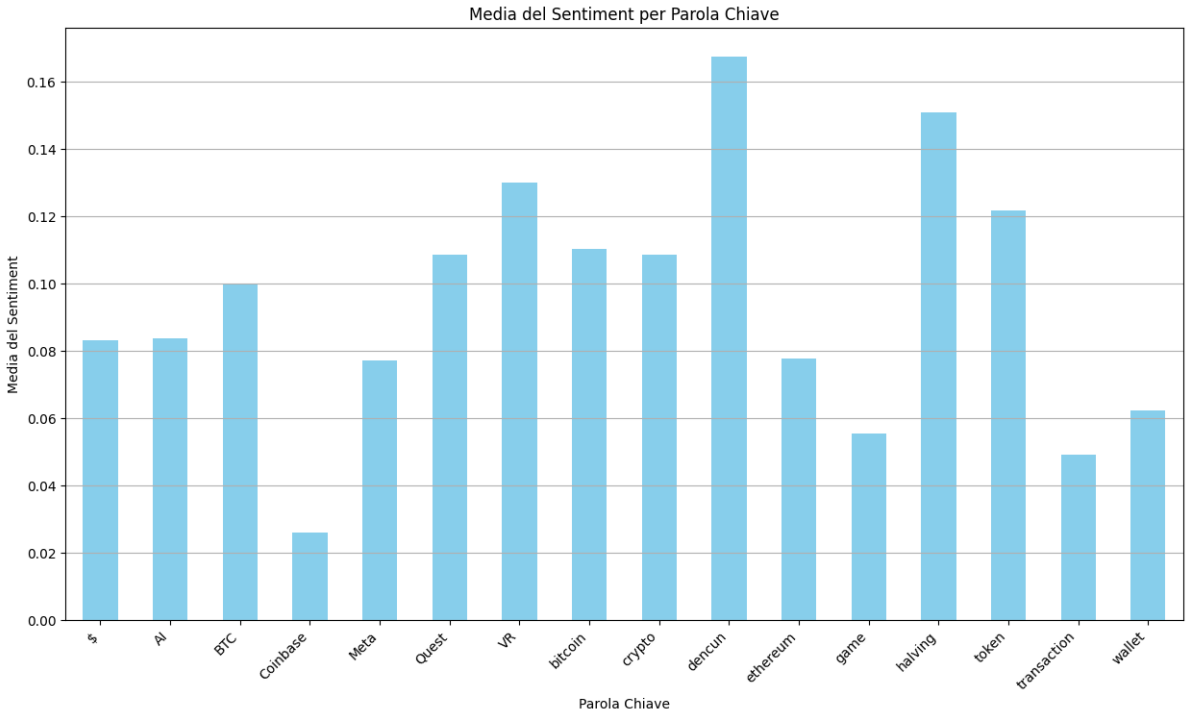

Figura 3.22: Valutazione media di ogni singola keyword

Nel grafico 3.22 invece è possibile visualizzare la valutazione del sentimento medio ottenuto dai post che contengono una delle parole chiave.
Salta all'occhio come le parole *dencun* e *halving* abbiano ottenuto un valore medio più alto rispetto agli altri termini,probabile indice del fatto che i due eventi siano stati presi positivamente da parte della community.

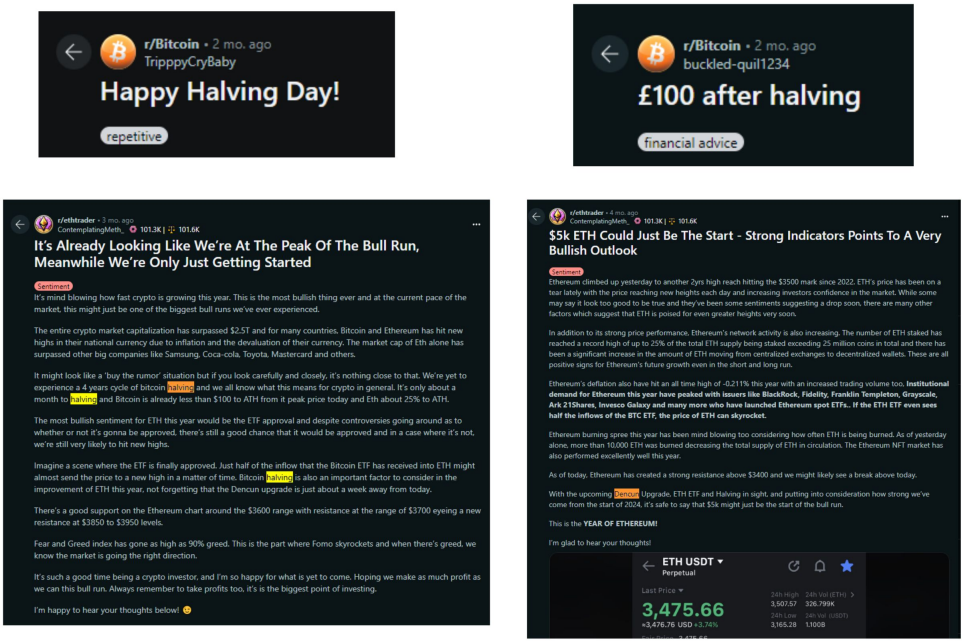

Figura 3.23: Alcuni post valutati positivamente riguardo l'aggiornamento Dencun e l'Halving di Bitcoin

Successivamente ho tracciato dei grafici che mostrassero l'andamento del sentimento nel tempo sia in maniera aggregata per tutte le parole chiave e sia per ogni singola keyword,in particolare le stesse per l'analisi delle frequenze(BTC,bitcoin,ethereum,dencun,halving).

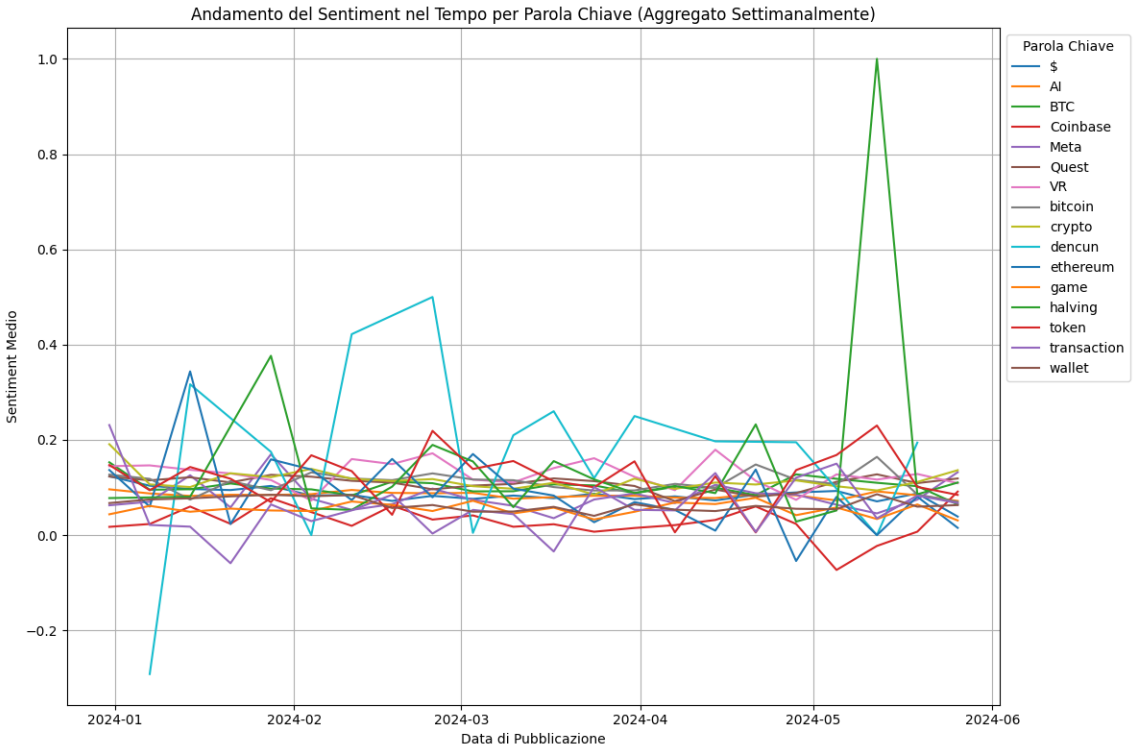

Figura 3.24: Andamento del sentimento aggregato settimanalmente per ogni parola chiave

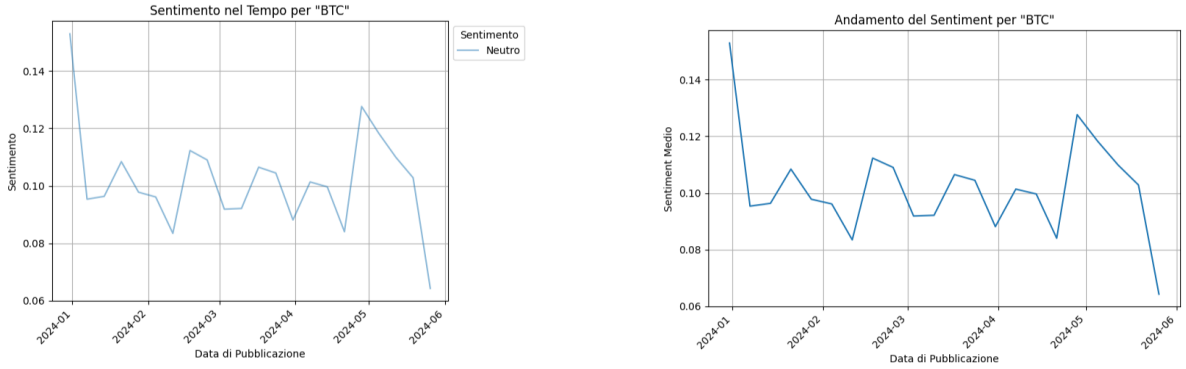

Figura 3.25: Valutazione e andamento sentiment medio per BTC

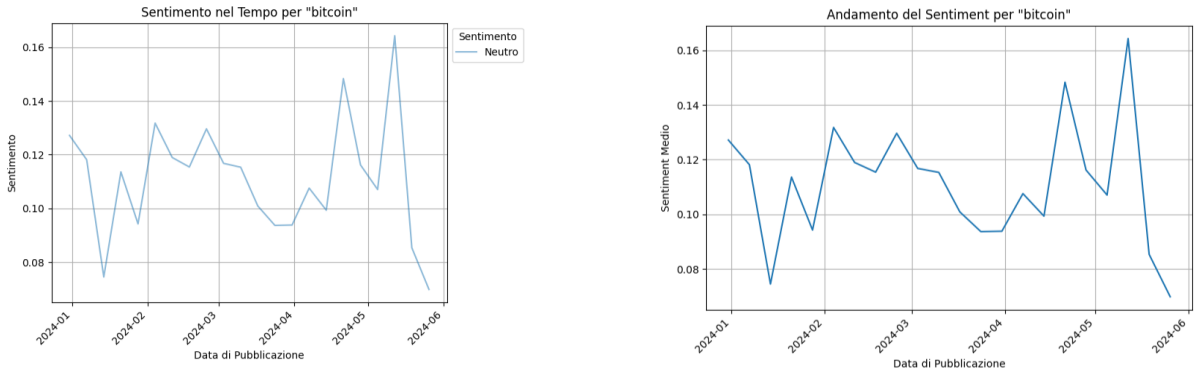

Figura 3.26: Valutazione e andamento sentiment medio per bitcoin

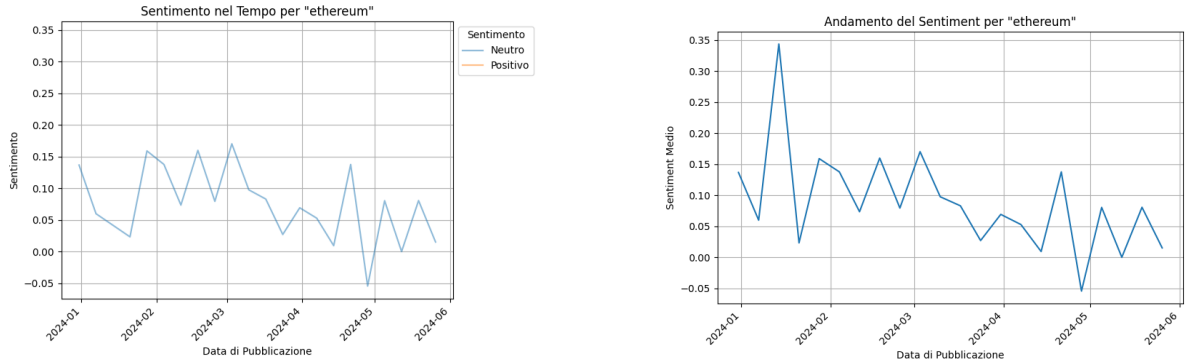

Figura 3.27: Valutazione e andamento sentiment medio per ethereum

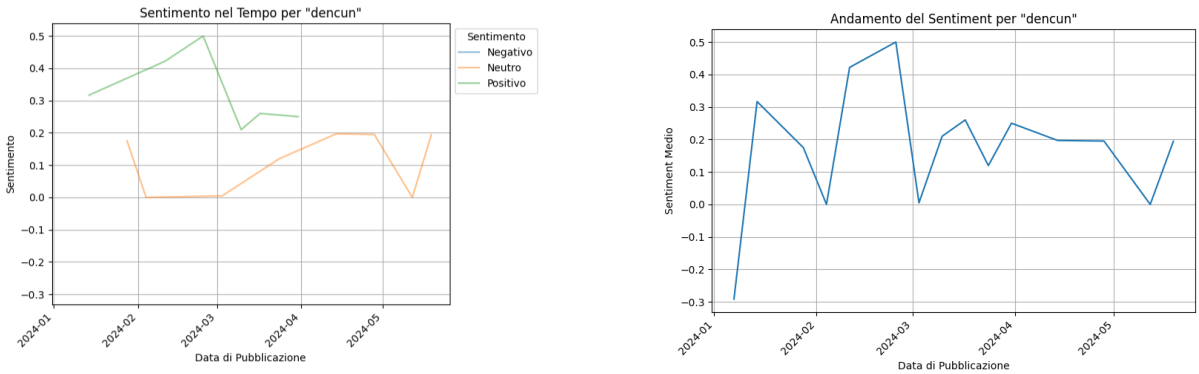

Figura 3.28: Valutazione e andamento sentiment medio per dencun

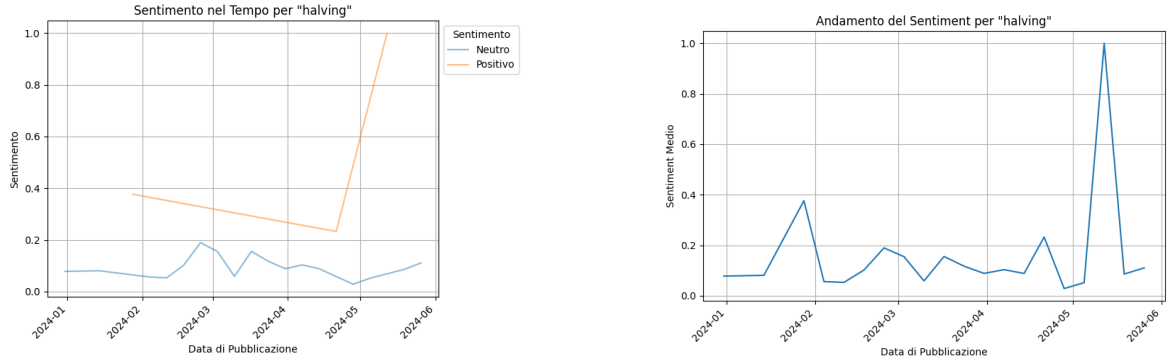

Figura 3.29: Valutazione e andamento sentiment medio per halving

In ogni figura viene mostrato sulla sinistra il grafico che evidenzia non solo la media del sentimento ma anche la valutazione che viene data in ogni singolo momento, in questo modo è possible vedere in che momenti i post sono stati contrassegnati come positivi,neutri, negativi.
Sulla destra invece viene semplicemente mostrato l'andamento medio del sentimento per i post contenenti la singola parola chiave.

Nelle figure 3.25 e 3.26 si può notare che nonostante i post vengano valutati come neutri in tutto il periodo di riferimento, dalla fine del mese di Aprile 2024 in poi si nota un aumento della valutazione dei post contenenti le parole *BTC* e *bitcoin*, sintomo dovuto ad una fase di attività ed eccitazione maggiore in un periodo considerabile come post-halving (20/04/2024).
In figura 3.29 dove viene raffigurato il grafico del sentiment per *halving* infatti si può notare come la polarità diventa positiva nel periodo sopra citato.
Nella figura 3.27 viene invece mostrato l'andamento del sentiment per la parola *ethereum*, in questo caso similarmente a quanto avvenuto per le parole chiave già citate, l'andamento mantiene una valutazione neutra con dei picchi nei mesi precedenti al giorno dell'aggiornamento Dencun (13/03/2024) sinonimo di una parte di utenza che probabilmente attendeva in maniera entusiasta l'arrivo dell'upgrade sulla blockchain di Ethereum. Analogalmente nel grafico 3.28 per la parola *dencun* si evidenzia una maggioranza di post positivi proprio nel periodo antecedente all'aggiornamento.

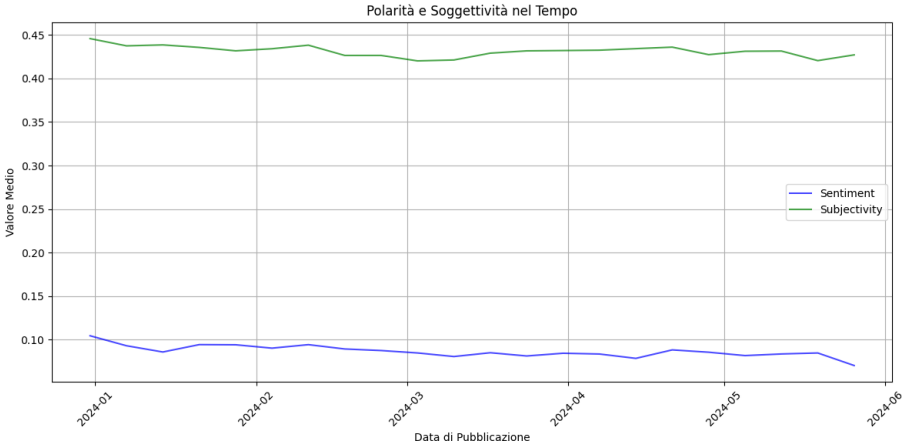

Figura 3.30: Confronto polarità e soggettività

Nella figura 3.30 vengono messe a confronto la polarità (Sentiment) e la soggettività (Subjectivity) medie per tutti i post presenti nel dataset. Si può notare come ci sia una certa distanza tra le due valutazioni soprattuto dovute al fatto che i post pubblicati, come volevasi dimostrare, contengono opinioni e pareri personali.

3.5 Opinion Leader Score

In questa sezione verrano mostrati e commentati i risultati ottenuti utilizzando i due diversi metodi per il calcolo degli Opinion Leader Score, ovvero quegli utenti che possano essere considerati più influenti di altri in una comunità.
Verrano quindi analizzati i risultati ottenuti mediante il calcolo descritto nella sezione 2.4 e l'adozione di un grafo che rappresenti le interazioni tra gli utenti Reddit per il calcolo dell'OLS mediante algortimo Pagerank come descritto nella sezione 2.5

3.5.1 Calcolo OLS

Nella tabella 3.2 vengono mostrati i top 10 utenti secondo il calcolo dell'Opinion Leader Score senza la creazione di una rete e considerando come peso per ogni post il numero di upvote sommato al numero di commenti ricevuti.

Autore	**OLS**
wsbapp	240838997762.25
OPINION_IS_UNPOPULAR	216028349732.25
AutoModerator	70022156689.0
welp007	5115468006.25
CryptoDaily	4909384489.0
Expensive-Two-8128	2680339984.0
iamwheat	1674078140.25
Parsnip	1372702500.0
Rydraelm	1023680025.0
hesapalmak	971350722.25

Tabella 3.2: Top 10 Utenti per OLS

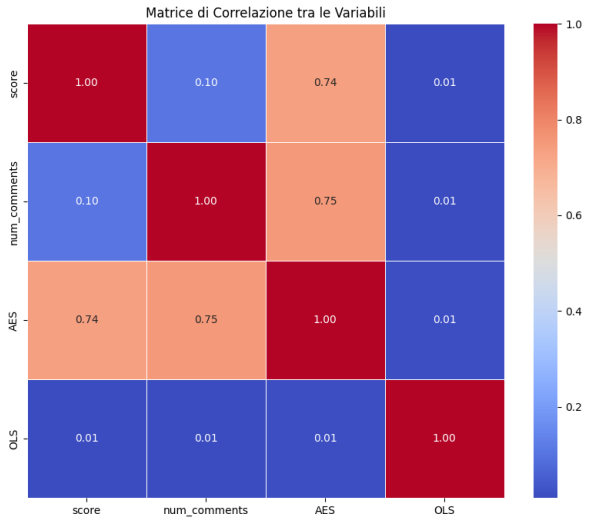

Figura 3.31: Matrice di correlazione per OLS

Dalla matrice di correlazione 3.31 è evidente come ci sia una forte correlazione tra il calcolo del *AES* (Account engagement score) e i valori *score* e *num_comments* per via del fatto che queste due variabili sono indicative per il calcolo del peso da attribuire ad ogni post. Si può notare anche una piccola correlazione tra *num_comments* e *score*, anche se non significativa potrebbe voler indicare che più è alto il numero di commenti ricevuti e più alto sarà il punteggio ottenuto dal post e viceversa.
Le analisi precedenti hanno evidenziato un significativo aumento di post da Marzo 2024 in poi, per questo ho deciso di calcolare l'OLS in due periodi distinti per verificare se i risultati della tabella 3.2 venissero confermati oppure se il maggior numero di post avesse cambiato le gerarchie tra gli utenti.

Autore	**OLS**
OPINION_IS_UNPOPULAR	170264342161.0
AutoModerator	10843473424.0
CryptoDaily	695218689.0
Kyrneh-1234	585664200.25
TortoiseAcquisition	283686649.0
Fausterion18	228176130.25
Grouchy_Letterhead53	196840900.0
Parsnip	176996416.0
McKoijion	120912016.0
akopley	116100625.0

Tabella 3.3: Top 10 Utenti per OLS (1° gennaio - 29 febbraio)

Autore	**OLS**
wsbapp	240838997762.25
AutoModerator	24607569424.0
welp007	3721061000.25
Expensive-Two-8128	2680236441.0
CryptoDaily	1797590404.0
OPINION_IS_UNPOPULAR	1521741090.25
iamwheat	1043386902.25
Rydraelm	1023680025.0
hesapalmak	890813562.25
marlennok	698544900.0

Tabella 3.4: Top 10 Utenti per OLS(1° marzo - oggi)

Dai risultati ottenuti si può constatare come ci sia effettivamente stato un cambiamento nelle gerarchie degli Opinion Leader nei due periodi.
Questo risultato può preannunciare innanzitutto che chi viene definito Opinion Leader non lo sarà per sempre e che c'è un'alta probabilità che gli utenti reagiscano ad eventi esterni.
Analizzando nello specifico i risultati ottenuti, nel primo periodo l'utente con l'OLS maggiore è risultato essere OPINION_IS_UNPOPULAR, un moderatore storico del subreddit *wallstreetbets*. Sotto di lui si posizionano due utenti che in realtà sono bot (*AutoModerator* e *CryptoDaily*). Su Reddit, i bot (moderatori automatici) vengono considerati

come utenti, con un flair posizionato vicino al nome. Essi possono pubblicare post e commentare, quindi devono essere considerati come utenti nell'analisi, ma con le dovute precisazioni.
Nella tabella 3.4, che si riferisce al periodo da marzo 2024 in poi, la prima posizione è ricoperta dall'utente *wsbapp*, un bot automatico che pubblica giornalmente post di aggiornamento sulle quotazioni delle crypto in tempo reale sul subreddit *wallstreetbets*.
Non è chiaro cosa possa aver scatenato questo cambio gerarchico, ma si può ipotizzare che le interazioni degli utenti siano cambiate nel periodo degli eventi sopra citati (Dencun e Halving Day), spingendoli ad aggiornarsi sul valore di mercato delle crypto.
L'unione dei due risultati riporta alla tabella 3.2 dove, dopo questa analisi, si può notare che il secondo periodo preso in esame abbia influenzato maggiormente il calcolo degli Opinion Leader per questo metodo.

3.5.2 Pagerank

Per calcolare il Pagerank è stato necessario costruire un grafo in cui ogni utente è rappresentato da un nodo e gli utenti vengono collegati tra loro solo nel caso in cui avvenga un'interazione, ovvero quando almeno uno dei due utenti ha commentato un post pubblicato dall'altro. Nella figura 3.32 è possibile visualizzare la composizione del grafo per i top 10 utenti secondo il Pagerank.

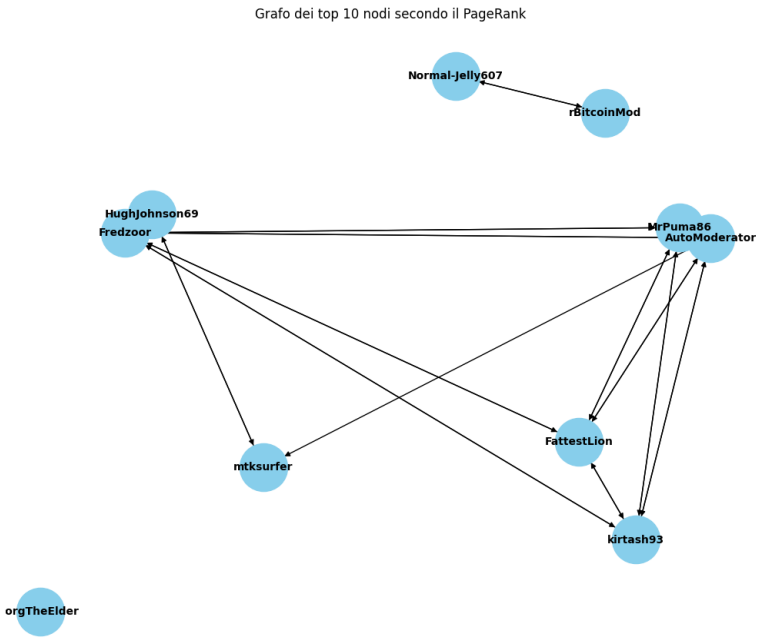

Figura 3.32: Grafo per i top 10 nodi per Pagerank

Autore	**PageRank**
AutoModerator	0.05375104333403372
rBitcoinMod	0.010938238216959129
mtksurfer	0.009097313969197392
MrPuma86	0.007192397162966668
Normal-Jelly607	0.005263441696784601
HughJohnson69	0.005205016256250552
JorgTheElder	0.0050474007883037
kirtash93	0.004837039028841061
FattestLion	0.004564424736939671
Fredzoor	0.004541677358931067

Tabella 3.5: Top 10 Utenti per Pagerank

Nella tabella 3.5 vengono mostrati i risultati ottenuti applicando l'algoritmo di Pagerank al grafo generato precedentemente. Come per il calcolo dell'OLS anche in questo caso ho deciso di calcolare il Pagerank sui due periodi differenti, in questo caso vengono creati due grafi, uno per ogni periodo e su ciascuno di essi viene applicato l'algoritmo di Pagerank. I risultati ottenuti sono visualizzabili nelle tabelle 3.6 e 3.7.

Autore	**PageRank**
AutoModerator	0.07687086639434183
MrPuma86	0.017891436060239706
kirtash93	0.015281827407377173
Friendly-Airline2426	0.012354943181046565
Fredzoor	0.011950476672133238
Buzzalu	0.011802854910361796
yester_philippines	0.01139862635876494
Every_Hunt_160	0.010299427732394856
Sky-876	0.010057402592161192
ContemplatingMeth_	0.009757744921284337

Tabella 3.6: Top 10 Utenti per Pagerank (1° gennaio - 29 febbraio)

Autore	PageRank
AutoModerator	0.05676644071915664
rBitcoinMod	0.017567176038042316
mtksurfer	0.012144966410254964
MrPuma86	0.008675599445677263
HughJohnson69	0.007476679628933253
Normal-Jelly607	0.007197054868594068
FattestLion	0.006085978318107579
joeker13	0.005645701211727907
Fredzoor	0.005440819615998723
ContemplatingMeth_	0.0054075134174552545

Tabella 3.7: Top 10 Utenti per Pagerank(1° marzo - oggi)

Come nel caso del calcolo dell'OLS avviene un cambio nelle gerarchie tra i due periodi presi in esame. In particolare nel secondo periodo, nella colonna degli autori, fa la comparsa un nuovo utente,anche in questo caso si tratto di un bot, rBitcoinMod. Si tratta del moderatore ufficiale del subreddit dedicato a *bitcoin*. Il fatto che abbia avuto una influenza maggiore nel secondo periodo potrebbe essere dovuta all'evento di dimezzamento del Bitcoin precedentemente citato che avrebbe portato il moderatore automatico a gestire una quantità superiore di post e quindi moderare con più frequenza rispetto al periodo precedente.

Nella figura 3.33 viene visualizzato l'andamento mensile del pagerank per i top 10 utenti, viene generato un grafo per ogni mese e calcolato il valore di pagerank per ogni utente. Come è possibile notare non esiste alcun utente che riesca a mantenere una inlfuenza superiore agli altri per un lungo periodo.

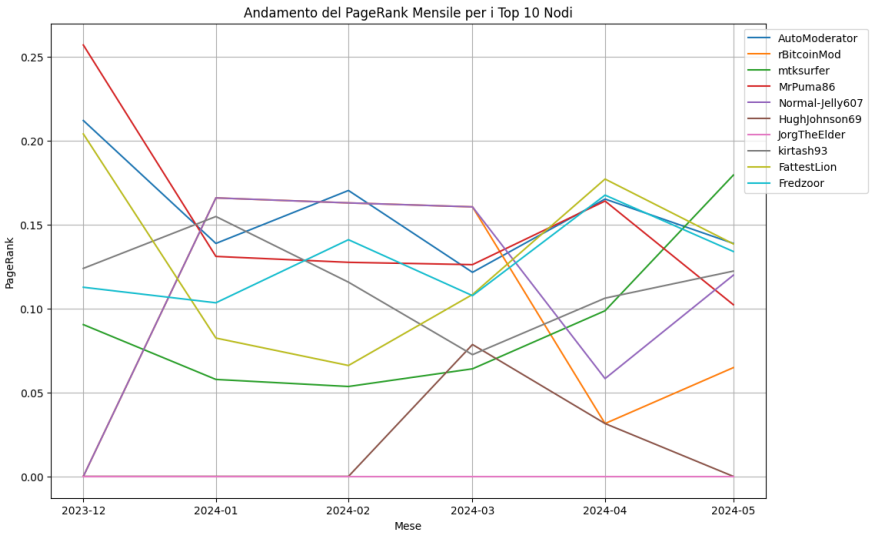

Figura 3.33: Andamento dei top 10 nodi per Pagerank

Conclusioni

L'analisi comportamentale degli utenti su Reddit in relazione ai post pubblicati nei subreddit dedicati alle criptovalute ha rivelato una serie di pattern significativi sia nelle frequenze dei termini utilizzati sia nell'analisi sentimentale.
Infatti, durante i mesi di marzo e aprile, si è verificato un incremento notevole di post, con ogni probabilità correlato a eventi significativi nel mondo delle criptovalute, come l'aggiornamento Dencun alla blockchain di Ethereum e l'halving di Bitcoin.
Come specificato nel paragrafo inerente all'analisi delle frequenze, verificando la presenza delle parole chiave "dencun" e "halving" nei post pubblicati, queste hanno mostrato una frequenza di utilizzo elevata durante questi periodi. Anche l'analisi sentimentale ha indicato che i post contenenti queste parole chiave hanno generalmente ottenuto valutazioni migliori rispetto alle altre parole analizzate, suggerendo che la comunità ha accolto favorevolmente questi eventi.
Per l'identificazione degli Opinion Leader sono stati utilizzati due metodi distinti: il metodo OLS offre una misura chiara e diretta dell'engagement basato sul contenuto creato da un utente, mentre il metodo basato su PageRank fornisce una valutazione più complessa e interconnessa dell'influenza all'interno della comunità. Sicuramente quest'ultimo metodo è stato il più convincente ed ha portato l'identificazione degli Opinion Leader ad uno step superiore, grazie al fatto di aver considerato Reddit come una rete simile a X, dove gli utenti, anche di subreddit diversi, sono connessi tra loro. I risultati ottenuti dai due metodi sono stati distinti, riflettendo la loro diversità di approccio. Tuttavia, in entrambi i casi hanno rivelato una presenza significativa di bot automatici tra gli utenti più attivi. In particolare, il moderatore automatico del subreddit Bitcoin ha mostrato un incremento significativo del suo PageRank nel periodo successivo a marzo, evidenziando l'influenza di questi bot in momenti cruciali per la comunità, come l'avvento dell'halving, che ha sicuramente portato il bot a generare più contenuti e di conseguenza più interazioni con gli utenti iscritti a quel subreddit e non solo.
Questi risultati evidenziano alcune considerazioni importanti. Gli eventi significativi nel mondo delle criptovalute hanno sicuramente un impatto diretto sull'attività della comunità su Reddit, dimostrando come l'interesse e le discussioni siano guidate da cambiamenti rilevanti. La presenza e l'attività dei bot automatici, che agiscono come utenti influenti, sollevano però delle domande sulla natura dell'influenza e su come questa venga percepita dagli altri utenti.

In sintesi, questa analisi fornisce una panoramica approfondita sull'attività degli utenti su Reddit nel contesto delle criptovalute, mostrando come gli eventi esterni influenzino le discussioni e come gli Opinion Leader, inclusi i bot automatici, giochino un ruolo cruciale nella formazione delle opinioni e delle interazioni. Dai risultati ottenuti si può riconoscere che nessun utente o bot è capace di restare leader per un lungo periodo, a dimostrazione del fatto che la rete è veloce, mutevole e soprattutto non prevedibile. Questo studio può contribuire a una migliore comprensione delle dinamiche delle comunità online e offrire spunti per ulteriori ricerche sull'influenza e l'interazione nei social media.

Sviluppi Futuri

Come in ogni studio di tesi, anche la mia analisi non è esente da possibili migliorie e sviluppi futuri.

Innanzitutto, si potrebbe estendere la stessa analisi a un dataset più ampio, che prenda come riferimento temporale un periodo più lungo, in modo da verificare se il medesimo comportamento si è verificato anche per altri eventi in passato.

Per quanto riguarda la fase di pre-processing del testo e la clusterizzazione, nel mio studio mi sono limitato a valutare 1000 campioni per velocizzare il processo di riconoscimento. Tuttavia, potrebbe essere utile aumentare questo numero per ottenere una precisione maggiore nel riconoscimento dei termini chiave e nella clusterizzazione.

Nel calcolo dell'OLS, oltre a una diversa distribuzione dei pesi, si potrebbero utilizzare altre metriche di engagement, come il numero di risposte ai commenti ricevuti, assegnando un peso maggiore ai commenti che generano discussioni lunghe.

Infine, si potrebbero applicare tecniche di machine learning per identificare e filtrare la presenza di bot, migliorando così la qualità dell'analisi.

Bibliografia

[1] Corriere.it *Reddit, cos'è e come funziona il social network che si quoterà in Borsa e vale 15 miliardi*(04/12/2023)
https://www.corriere.it/tecnologia/cards/reddit-cos-e-e-come-funziona/la-storia-di-reddit-chi-sono-i-proprietari.shtml#:~:text=Reddit%20nasce%20il%2023%20giugno,mila%20dollari%20dalla%20Y%20Combinator.

[2] cryptonomist.ch *Reddit lancia i propri token Community Points*(14/05/2020)
https://cryptonomist.ch/2020/05/14/reddit-lancia-token-community-points/

[3] cryptonomist.ch *Crypto news: Reddit chiude i "Community Points" basati su Blockchain*(18/10/2023)
https://cryptonomist.ch/2023/10/18/crypto-news-reddit-chiude-community-points/

[4] criptovaluta.it *Reddit compra Bitcoin e Ethereum — E ha altre crypto che...*(23/02/2024)
https://www.criptovaluta.it/83775/reddit-compra-bitcoin-e-ethereum-e-ha-altre-crypto-che

[5] Bertazzoni, Matilde *Rilevazione di eventi da Social Media: Twitter vs Instagram. Una valutazione comparativa. [Laurea], Università di Bologna, Corso di Studio in Informatica per il management [L-DM270]*(2023)
https://amslaurea.unibo.it/28004/

[6] Cinelli, Loli Piccolomini, E., & Morotti, E.*Analisi dei dati testuali da Twitter sulla Didattica a Distanza. Alma Mater Studiorum - Università di Bologna.*(2021)
https://amslaurea.unibo.it/22860/

[7] Giannini, Gallinucci, E., & Casadei, M. *Social Network Analysis: Architettura Streaming Big Data di Raccolta e Analisi Dati da Twitter. Alma Mater Studiorum - Università di Bologna.*(2022)
https://amslaurea.unibo.it/25378/

[8] Riquelme, Rivera, D., & Serrano, B. *Analyzing the far-right political action on Twitter: the Chilean constituent process. Social Network Analysis and Mining, 12(1), 161–161.*(2022)
https://doi.org/10.1007/s13278-022-00990-w

[9] Alieva, Moffitt, J. D., & Carley, K. M. *How disinformation operations against Russian opposition leader Alexei Navalny influence the international audience on Twitter. Social Network Analysis and Mining, 12(1), 80–80.*(2022)
https://doi.org/10.1007/s13278-022-00908-6

[10] Dos Santos, de Brito Silva, M. J., da Costa, M. F., & Batista, K. *Go vegan! digital influence and social media use in the purchase intention of vegan products in the cosmetics industry. Social Network Analysis and Mining, 13(1), 49–49.*(2023)
https://doi.org/10.1007/s13278-023-01034-7

[11] Britt, Hayes, J. L., Musaev, A., Sheinidashtegol, P., Parrott, S., & Albright, D. L. *Using targeted betweenness centrality to identify bridges to neglected users in the Twitter conversation on veteran suicide. Social Network Analysis and Mining, 11(1), 40.*(2021)
https://doi.org/10.1007/s13278-021-00747-x

[12] Zhang, He, H., & Cao, B. *Identifying and evaluating the internet opinion leader community based on k-clique clustering. Neural Computing & Applications, 25(3-4), 595–602*(2014)
https://doi.org/10.1007/s00521-013-1529-1

[13] Ferretti, S.; Furini, M *Cryptocurrency Turmoil: Unraveling the Collapse of a Unified Stablecoin (USTC) through Twitter as a Passive Sensor. Sensors 2024, 24, 1270*
https://doi.org/10.3390/s24041270

[14] Furini, M. *X as a Passive Sensor to Identify Opinion Leaders: A Novel Method for Balancing Visibility and Community Engagement. Sensors 2024, 24, 610.*
https://doi.org/10.3390/s24020610

[15] Marco Furini, Luca Mariotti, Riccardo Martoglia, and Manuela Montangero. *2022. On Designing a Time Sensitive Interaction Graph to Identify Twitter Opinion Leaders. In Proceedings of the 2022 ACM Conference on Information Technology for Social Good (GoodIT '22). Association for Computing Machinery, New York, NY, USA, 175–182.*
https://doi.org/10.1145/3524458.3547268

[1.5] SpaCy Documentation
https://spacy.io/api/doc

[2.5] Scikit-learn Documentation
https://scikit-learn.org/0.21/documentation.html

[3.5] TextBlob Documentation
https://textblob.readthedocs.io/en/dev/

[4.5] Matplotlib Documentation
https://matplotlib.org/stable/index.html

[5.5] Seaborn Documentation
https://seaborn.pydata.org/api.html

[6.5] WordCloud Documentation
https://amueller.github.io/word_cloud/

[7.5] Plotly Documentation
https://plotly.com/python-api-reference/

[8.5] NetworkX Documentation
https://networkx.org/documentation/stable/reference/index.html

Ringraziamenti

Dopo lunghi anni si conclude qui il mio percorso di studi, quello che non finirà sicuramente sarà la voglia di imparare e di mettersi in gioco con nuove esperienze.

Ringrazio tutti i professori per avermi donato i loro insegnamenti ed in particolare il mio relatore, il Prof. Stefano Ferreti e correlatore, il Prof. Marco Furini per aver sostenuto le mie idee ed avermi fatto da guida lungo il mio percorso di ricerca per questa tesi.

Grazie ai miei colleghi ed amici Daniele, Francesco e Stefano per aver trovato la giusta intesa e complicità che ci ha sostenuto nella realizzazione dei progetti durante il corso, siamo stati un ottimo team!

Grazie ad Alberto, il mio coinquilino e amico di una vita, nonché primo sostenitore. Abbiamo vissuto questo percorso fianco a fianco, crescendo insieme e condividendo momenti che rimarranno per sempre nei miei ricordi.

Ringrazio di cuore i miei amici, che sono stati una costante presenza nel mio percorso. I loro piccoli gesti, anche quelli più spontanei, mi hanno aiutato a superare i momenti difficili. Siete una seconda famiglia per me.

Un ringraziamento speciale va ai miei genitori, che hanno sempre desiderato il meglio per me facendo sacrifici enormi in questi anni. Senza il loro supporto, non sarei arrivato fin qui. Mi sento in debito con loro ed è giunto il momento di ricambiare per tutto ciò che hanno fatto.

Grazie a chi mi ha sostenuto da lassù, spero possiate essere fieri di me