

Università degli Studi di Verona

DIPARTIMENTO DI INFORMATICA

Corso di Laurea in Matematica Applicata

TESI DI LAUREA

**USO DEL TEXT MINING APPLICATO A TWITTER PER
PREDIRE L'ANDAMENTO DI BITCOIN**

Candidato:

Eleonora Fiorentino

Matricola VR443555

Relatore:

Diego Lubian

Indice

Introduzione	1
1 Sentiment Analysis e Text Mining: di cosa si tratta	3
1.1 Dati non strutturati	3
1.2 Analisi dati	3
1.3 Analisi del sentimento	5
1.4 Intelligenza artificiale per il Text Mining	5
1.5 Text Mining	7
1.6 Natural Language Processing (NLP)	8
2 Bitcoin, Blockchain e criptovalute	11
2.1 Blockchain	11
2.2 Criptovalute	12
2.3 Bitcoin	13
3 Applicazione pratica del text mining	15
3.1 Tecniche pratiche	15
3.2 Alcuni studi	19
4 Lavoro svolto e risultati	21
4.1 Breve introduzione a TextBlob	21
4.2 Altre librerie utilizzate	22
4.3 Il programma	23
4.4 Risultati (confronti)	26
4.5 Analisi di regressione	30
5 Conclusioni	35
A Codice main	37
B File Tabella	41
C Codice R	43

Introduzione

Oggiorno i siti di social networking e in generale i canali di aggregazione digitali sono in piena espansione, quindi su di essi viene generata ogni giorno una vastissima quantità di dati. Milioni di persone condividono costantemente opinioni, consigli, abitudini e molte altre parti della loro quotidianità su siti e applicazioni, promuovendo un modello di comunicazione basato su frasi sempre più brevi e concise. In questa trattazione in particolare indagheremo la presenza di una correlazione fra i sentimenti espressi dagli utenti di un famoso servizio di micro blogging, Twitter, riguardo la criptovaluta Bitcoin e l'andamento del prezzo di mercato della stessa. Sappiamo che ci sono quasi 111 siti di microblogging. I siti Web di microblogging non sono altro che social media su cui l'utente pubblica post brevi e frequenti: Twitter è uno dei più famosi servizi di microblogging, infatti la lunghezza massima di un “tweet” (un messaggio) dev'essere necessariamente di 280 caratteri. Ai fini della ricerca useremo proprio questi tweet come dati grezzi.

Per farlo, ci avvarremo di alcune tecniche di data mining, come l'uso dell'algoritmo di analisi del sentimento, utilizzando algoritmi di apprendimento automatico. Grazie all'implementazione di un programma in grado di utilizzare questi algoritmi ai nostri fini e di estrarre i tweets a cui siamo interessati riusciremo a classificare automaticamente i sentimenti dei Tweet presi dal set di dati di Twitter e a compierne uno studio statistico. Useremo infatti una specifica libreria di Python, TextBlob, che fornisce una funzione in grado di estrarre la polarità di tali messaggi: essi verranno classificati come positivi, negativi o neutri rispetto a un termine della query (“domanda”). Questo metodo è in generale molto utile per le aziende che vogliono conoscere il feedback sui marchi dei loro prodotti o per i clienti che vogliono cercare l'opinione di altri consumatori su un prodotto prima dell'acquisto.

Negli ultimi anni l'analisi del sentimento è diventata una delle aree di ricerca più popolari in ambito computazionale grazie all'esplosione di informazioni sul sentimento provenienti da siti web, social (ad esempio Instagram, Twitter e Facebook), forum online e blog. Per quanto riguarda Bitcoin, sappiamo che il suo valore non è determinato da agenti esterni al mercato (come succede ad esempio per alcuni beni alimentare, la cui produzione può dipendere da condizioni ambientali e altri fattori esterni), bensì dipende da speculazioni sul prezzo e da aspettative, che spesso sono determinate da dichiarazioni

fatte da entità con grande peso economico, politico o sociale. Infatti, le criptovalute non possono essere comprese usando i fattori macroeconomici che di solito influenzano i mercati azionari, valutari e delle materie prime. Proprio per questo motivo ha senso aspettarsi che le due variabili, prezzo effettivo e aspettativa dei compratori, siano in relazione diretta, e che la seconda sia positivamente influenzata dalla prima. Negli anni passati la ricerca si è concentrata sull'analisi fondamentale e tecnica per prevedere il prezzo delle criptovalute. In questa trattazione invece, verificheremo se il valore di Bitcoin può essere soddisfacentemente determinato dal sentimento e ne analizzeremo il grado di correlazione.

Capitolo 1

Sentiment Analysis e Text Mining: di cosa si tratta

1.1 Dati non strutturati

La maggior parte delle informazioni create ai giorni nostri sono dati non strutturati, ovvero informazioni digitali in forma grezza, dati che non rientrano in una struttura ben definita. Proprio per questa loro caratteristica, che li differenzia dai dati strutturati, le informazioni che contengono non possono essere mappate in schemi di database relazionali e in generale risulta quasi impossibile per un computer accedere direttamente a un contenuto informativo specifico. La maggior parte di questi dati è sotto forma di testo: post sui social media, e-mail, recensioni online, rapporti aziendali, ecc., ma esistono anche in forma di audio e dati video.

Questi dati possono avere origini molto diverse, che spaziano dall'estrazione da un linguaggio umano con tecniche di NLP (Natural Language Processing), ai social media fino all'acquisizione tramite sistemi di misura elettronici: questa caratteristica ne rende difficile la comprensione e ambigua la collocazione. Inoltre questi dati tendono a raggiungere dimensioni elevate e occupano volumi di gran lunga superiori rispetto ai dati strutturati, arrivando anche a scale del Petabyte (l'equivalente di un milione di Gigabyte).

Tuttavia, questi dati sono tutt'altro che inutili: contengono infatti enormi quantità di informazioni, che possono diventare un mezzo molto potente per le aziende per arrivare a comprendere meglio il mercato e i suoi consumatori. Qui è dove entra in gioco l'analisi dei dati.

1.2 Analisi dati

Con “analisi dei dati” si intendono tutti quei processi tramite i quali si ricavano informazioni da dati in forma grezza che vengono estratti, trasformati e messi in relazione fra loro per scoprire e interpretare eventuali schemi na-

4CAPITOLO 1. SENTIMENT ANALYSIS E TEXT MINING: DI COSA SI TRATTA

scosti, relazioni, tendenze, correlazioni e anomalie, oppure per convalidare una teoria o un'ipotesi.

Queste informazioni vengono poi analizzate per prendere decisioni in tempo reale, individuare trend emergenti e svelare condizioni che non sarebbero altrimenti evidenti utilizzando processi di gestione dei dati obsoleti.

Negli ultimi anni i dati hanno assunto importanza via via crescente nell'organizzazione delle attività di produzione e di scambio, a tal punto da poter essere considerati a tutti gli effetti come un'importante risorsa economica in moltissimi settori.

Ma da dove provengono questi dati?

In realtà qualsiasi byte di informazione prodotto da qualunque individuo, su qualsiasi piattaforma online può diventare un dato più o meno importante: riporto in Figura 1.1 un grafico, che può fornire un'idea generale in merito alla quantità di informazioni che vengono scambiate su internet ogni minuto (studio risalente all'anno 2021, riferimenti in nota).

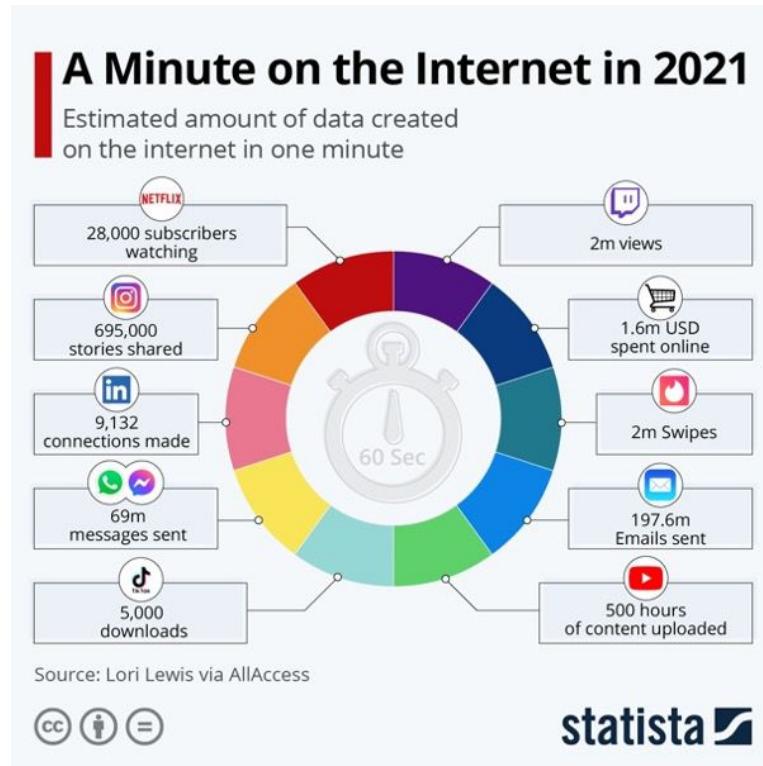


Figura 1.1: Statista, “A minute on the Internet in 2021” by Claire Jenik, Jul 30, 2021

È chiaro, dunque, il motivo per cui moltissime imprese, aziende, centri di ricerca e persino enti pubblici vogliono garantirsi il possesso di questi dati: in essi si celano infatti i pensieri, le abitudini, le preferenze e i gusti di

moltissime persone, fra le quali si trovano naturalmente anche i ‘target’ delle sopraccitate organizzazioni.

Tuttavia esistono anche altri tipi di dati, ovvero quei dati relativi all’azienda in sé: anche queste informazioni hanno un’importanza ormai vitale, e vengono utilizzate prevalentemente per aumentare l’efficienza di processi e costi, guidare la strategia e il cambiamento e monitorare e migliorare i risultati finanziari.

Per quanto anche questa categoria sia estremamente utile e centrale ormai in molte imprese, ciò di cui ci occuperemo in questa trattazione sarà un altro tipo di analisi, ovvero la ‘Sentiment Analysis’.

1.3 Analisi del sentimento

Col termine ‘Sentiment Analysis’ si intende l’analisi dei sentimenti e delle opinioni dei consumatori, al fine di comprendere qual è il grado di apprezzamento ottenuto da un soggetto, un servizio, un bene o un’azienda che si espone sui social media. Ci concentreremo, nello specifico, sull’analisi di fonti di tipo testuale, attinenti allo scopo di questa ricerca.

Se pensiamo a quanto detto precedentemente però, ci rendiamo subito conto di quanto sia fondamentale l’impiego di programmi e tools informatici specifici, che riescano non solo a trattare quantità di dati tanto elevate, ma anche a portare a termine il compito più delicato: riuscire ad interpretare una particolare disposizione d’animo solo tramite l’analisi di un testo e delle parole che contiene.

Naturalmente, se un compito simile fosse eseguito da un essere umano, risulterebbe naturale ritenere che egli sia in grado di interpretare il sentimento generale di un messaggio, sebbene talvolta persino gli uomini non riescano a cogliere alcune vene di sarcasmo o ironia velate, e potrebbero quindi essere portati a formulare una valutazione errata.

Immaginiamo ora di dover programmare un calcolatore, affinché sia in grado di fare la medesima minuziosa attività di “leggere” parola per parola e attribuire un sentimento generale al messaggio: ci si rende facilmente conto di quanto un simile compito sia complicato, e quale possa essere il margine di errore anche del programma che riproduca quanto più verosimilmente possibile la mente umana.

1.4 Intelligenza artificiale per il Text Mining

Come è possibile aspettarsi, a svolgere una simile impresa non è però un semplice programma, bensì un’intelligenza artificiale (AI), che viene addestrata a leggere i messaggi e ad interpretare il grado di positività, neutralità o negatività che essi esprimono.

6CAPITOLO 1. SENTIMENT ANALYSIS E TEXT MINING: DI COSA SI TRATTA

Senza addentrarci troppo in specifiche tecniche in questo capitolo, che vuole essere una semplice introduzione su cos'è il ‘Sentiment Analysis’ e in cosa consiste l'attività di ‘Text mining’ (o ‘Text Analysis’, termini che pur se diversi hanno significati simili), vorrei spiegare brevemente come avviene questo particolare addestramento: in un primo momento vengono sottoposti a quella che diverrà la nostra AI numerosi esempi di testi, ciascuno presentato insieme ad una valutazione in merito alla polarità (sentimento) del messaggio stesso: dopo molti di questi esempi l'AI “impara” (proprio come farebbe una mente umana) a individuare subito certe parole ricorrenti e a classificare (ma mai senza errori) il testo che le viene sottoposto.

Riporto in Figura 1.2 una rappresentazione esemplificativa di tale processo.

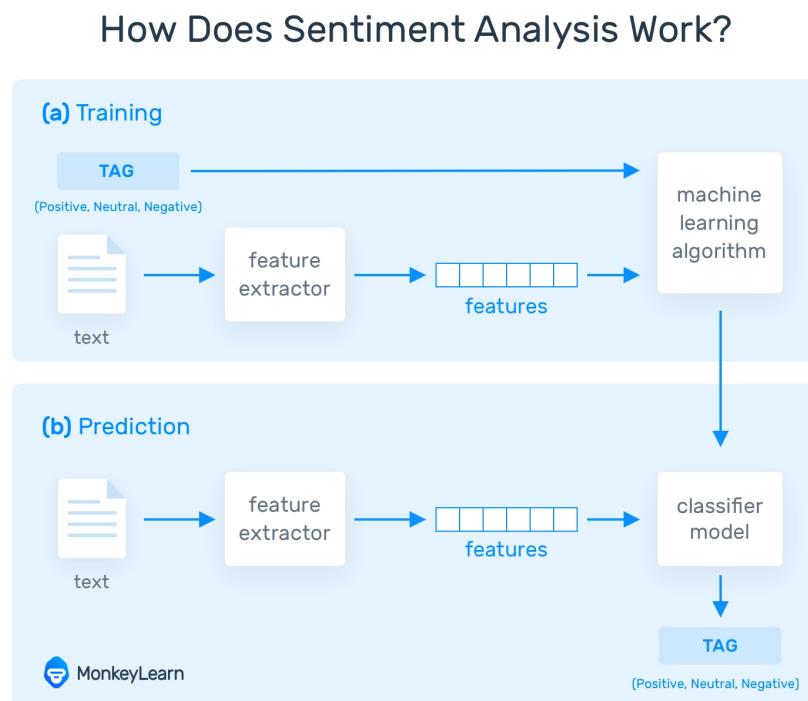


Figura 1.2: Fonte: MonkeyLearn, 2022

Il risultato finale dell'analisi di un particolare testo è un numero reale compreso fra -1 e 1 che ne indica la polarità. Nello specifico:

- -1 indica un sentimento totalmente negativo
- 0 indica neutralità (ma talvolta uno 0 pieno potrebbe anche indicare che l'AI non è riuscita ad analizzare il messaggio che le è stato sottoposto)

- 1 indica un sentimento totalmente positivo

Come già detto, questo indice restituisce numeri reali e può pertanto assumere infiniti valori (anche se a causa delle limitatezze della macchina non si potrebbe propriamente parlare di “infiniti”).

1.5 Text Mining

Come già anticipato, il Text Mining è una tecnica di Intelligenza Artificiale (AI) che utilizza l’elaborazione del linguaggio naturale (NLP) per trasformare un testo libero (non strutturato), in dati strutturati e normalizzati.

Lo scopo finale di questa pratica è appunto quello di attribuire una polarità ad un certo testo attraverso la creazione di un indice di gradimento che rispecchi il sentimento che un soggetto assume rispetto al tema.

Grazie a questa trasposizione dal mondo generalmente non classificabile dell’emotività e del pensiero ad un mondo prettamente numerico, questi dati possono poi essere espressi attraverso opportune rappresentazioni tabellari, grafici, mappe mentali ecc. oppure possono essere integrati in database, data warehouse o dashboard di business intelligence e utilizzati per analisi descrittive, predittive e prescrittive.

Gli scopi principali del Text Mining sono:

- classificare i documenti in categorie predefinite
- scoprire legami e associazioni nascoste
- individuare i temi principali di un testo e suddividerli per gruppi
- estrarre informazioni specifiche

Un processo di Text Mining si struttura generalmente in tre fasi: l’indicizzazione, il mining, la valutazione.

Nella fase di indicizzazione viene effettuata la parte di analisi linguistica e di “pulizia” del testo, eliminando le parole non necessarie all’analisi (articoli, congiunzioni ecc.) e la punteggiatura, attuando una eventuale riduzione di dimensionalità per poter arrivare ad una rappresentazione vettoriale del documento.

Ai documenti così trasformati, nella fase di "mining" viene applicato un algoritmo di Data Mining (“Text”, nel nostro caso), specifico per l’obiettivo da raggiungere. Infine, la fase di valutazione consiste nel calcolo di misure di efficacia e/o nell’interpretazione dei risultati ottenuti.

Una particolare applicazione di Text Mining è quella di cui si è parlato nel paragrafo precedente, ovvero la Sentiment Analysis: in quel caso la fase di “mining”, tramite l’ausilio di algoritmi di linguistica computazionale, consente di identificare ed estrarre le informazioni soggettive dalle diverse fonti

testuali, ovvero le opinioni e le emozioni in esse descritte. L'AI che guida la Sentiment Analysis è in grado di comprendere ed elaborare (anche se ricordiamo ancora che essa è tutt'altro che infallibile) i feedback emotivi che vengono forniti in modo spontaneo attraverso la scrittura di testi su web, post/commenti social quali Instagram, Facebook, Twitter, ecc. ma anche recensioni scritte su blog o siti aggregatori. Si tratta di informazioni preziosissime perché per loro natura rispecchiano le reali aspettative e gli stati d'animo di chi scrive.

1.6 Natural Language Processing (NLP)

Col termine Natural Language Processing, o elaborazione del linguaggio naturale, si intendono algoritmi di intelligenza artificiale in grado di analizzare, rappresentare e quindi comprendere il linguaggio naturale. Le possibili applicazioni spaziano dalla comprensione del contenuto, alla traduzione, fino alla produzione di testo in modo autonomo a partire da dati o documenti forniti in input.

Tra i sistemi di Intelligenza Artificiale, il Natural Language Processing (NLP) rientra tra le soluzioni software che negli ultimi anni hanno registrato maggiori progressi. Alcune fra le applicazioni più note sono ad esempio i correttori ortografici e i sistemi di traduzione automatici, che sono diventati ormai parte integrante della nostra vita quotidiana.

Grazie soprattutto al contributo di tecniche di AI sempre più avanzate, come Machine Learning e Deep Learning, l'NLP trova numerosi ambiti applicativi, anche se non siamo ancora giunti ad una completa ottimizzazione: basti pensare ad esempio alle difficoltà nel cercare di analizzare una lingua complessa come l'italiano, caratterizzata da modi di dire, espressioni gergali e influenzata da numerosi dialetti.

L'NLP si occupa principalmente di testi, ovvero sequenze di parole che esprimono uno o più messaggi (e provenienti ad esempio da pagine web, post, tweet, informazioni aziendali), mentre l'elaborazione del parlato (riconoscimento vocale) è considerato un ambito a sé e, come anticipato in precedenza, non è oggetto di questa ricerca.

Alcuni dei compiti fondamentali dell'AI che si occupa di NLP sono anche compiti generalmente basilari, quali ad esempio il riconoscimento della lingua o la scomposizione della frase in unità elementari, ma non solo: essi infatti comprendono anche l'analisi semantica e l'analisi del sentimento.

Più nello specifico, con le tecniche di NLP siamo in grado di performare le seguenti task:

- **Text Analysis:** analisi di un testo e, se necessario, individuazione di elementi chiave;

- **Text Classification:** interpretazione di un testo per classificarlo in una categoria predefinita;
- **Sentiment Analysis:** rilevamento dell'umore all'interno di un testo (di cui abbiamo visto un esempio sopra col calcolo della polarità);
- **Intent Monitoring:** comprensione del testo per prevedere comportamenti futuri;
- **Smart search:** ricerca, all'interno di archivi, dei documenti che meglio corrispondono ad una richiesta formulata in linguaggio naturale;
- **Text Generation:** generazione automatica di un testo;
- **Automatic Summarization:** produzione di una versione sintetica di uno o più documenti testuali;
- **Language Translation:** traduzione di testi scegliendo, volta per volta, il significato migliore a seconda del contesto.

In conclusione, l'unione di tutte le tecniche e delle tecnologie elencate in questo primo capitolo ci porta ad avere una solida base da cui poter iniziare la trattazione.

Capitolo 2

Bitcoin, Blockchain e criptovalute

2.1 Blockchain

Dal momento che lo scopo di questa tesi è di scoprire e indagare una possibile correlazione fra il sentimento degli utenti di Bitcoin riguardo allo stesso e l'andamento sul mercato della criptovaluta, è doveroso premettere una breve digressione su cosa sia Bitcoin e, più in generale, sulla tecnologia Blockchain. Il termine "Blockchain" significa letteralmente "catena di blocchi", ed è un registro decentralizzato e distribuito che archivia record di dati solitamente denotati "transazioni". Tutti i dati archiviati su una blockchain non possono essere modificati, rendendo la tecnologia un'ottima soluzione per attività come i pagamenti, dove la sicurezza informatica e la trasparenza dei processi sono essenziali.

Infatti, una volta scritti i dati in un blocco, questi non possono essere modificati o eliminati in un secondo momento senza che vengano alterati anche tutti i blocchi successivi ad esso, cosa che comunque necessiterebbe del consenso della maggioranza della rete. Inoltre, anche l'aggiunta di un nuovo blocco è regolata da un protocollo condiviso, e deve essere autorizzata da ciascun nodo. Ogni nodo aggiornerà poi la propria copia privata. La solidità del processo e la sua oggettività sono dunque conseguenza della natura stessa della struttura dati.

L'utilizzo di questa tecnologia consente anche di superare il problema dell'infinita riproducibilità di un bene digitale (un altro utilizzo importante della Blockchain riguarda proprio il mondo degli NFT, ovvero i non-fungible tokens) e della spesa doppia, senza che sia necessario ricorrere ad un server centrale o ad un'autorità. Infatti uno dei pilastri principali delle criptovalute in tutto il mondo è proprio che si tratta di sistemi trustless, ovvero che non necessitano di terze parti, come una banca centrale o un governo, per dimostrare l'identità di un soggetto della rete. Per fare ciò si utilizza una

funzione hash, un'intelligente crittografia che converte qualsiasi dato in un lungo codice composto sia da lettere che da numeri per nascondere il dato originale. Questa funzione di hash è particolarmente interessante perché produce sempre lo stesso codice per un determinato blocco di dati, ma se dovesse venire apportata una benché minima modifica nella fonte originale, il codice cambia completamente.

Pertanto, le funzioni di hash possono essere utilizzate per determinare chi possiede un dato senza che venga rivelato quale sia questo dato, come se si trattasse di una sorta di chiave digitale. La blockchain è quindi rappresentabile come una lista di blocchi, a ciascuno dei quali può essere associate una o più transazioni. Inoltre, ciascun blocco è collegato a quello precedente tramite un puntatore hash, e possiede una propria marca temporale, ovvero contiene l'esatta data della transazione a cui è associato.

Se i blocchi in una blockchain non fossero collegati, sarebbe facile inserirne uno falso, ecco perché ogni blocco è collegato a quello precedente. Questo collegamento, come già anticipato, avviene tramite un puntatore hash. In questo modo chiunque può verificare che le transazioni nel blocco siano successive a quelle precedenti, e ciò consente effettivamente di garantire che l'intera catena sia legittima e che non siano stati aggiunti eventuali altri blocchi da utenti non autorizzati. Nel caso specifico di Bitcoin, ciò significa anche che è possibile ricostruire la storia di ogni singola moneta da quando è stata estratta.

Un esempio di piattaforma blockchain è la piattaforma software Ethereum, che, come si evince dal nome, ospita la criptovaluta Etherium.

2.2 Criptovalute

Le criptovalute sono sistemi di pagamento decentralizzati in cui la proprietà viene dimostrata crittograficamente e tracciata in ogni passaggio tramite la tecnologia blockchain. Il loro numero è aumentato incredibilmente negli ultimi anni, e riportiamo qui solo i nomi di alcune di esse, le più conosciute e acquistate sul mercato al momento (secondo Forbes): Bitcoin, Ethereum, Tether, U.S. Dollar Coin, Binance Coin, XRP ecc. Al giorno d'oggi, le nuove criptovalute sono più comunemente create come applicazioni su un'altra criptovaluta già presente sul mercato. Tali criptovalute sono chiamate Token. Creare una nuova criptovaluta è facile, ma il suo valore dipende dalla disponibilità degli utenti a pagare per le sue unità: pertanto se una criptovaluta perde il suo bacino d'utenza, diventa inutile.

Come anticipato, questa trattazione prende in esame una specifica criptovaluta, ovvero Bitcoin. Ci limiteremo dunque a parlare approfonditamente solo di quella.

2.3 Bitcoin

Bitcoin è una criptovaluta, ovvero una valuta digitale utilizzata come forma di pagamento alternativa e creata utilizzando algoritmi di crittografia. L'uso di tecnologie di crittografia significa che le criptovalute funzionano sia come valuta che come sistema di contabilità virtuale. Per utilizzarle è necessario possedere di un portafoglio di criptovalute: si tratta di software basati su cloud o archiviati su un computer o su un dispositivo mobile personale. I "wallet" sono lo strumento su cui sono memorizzate tutte le chiavi di crittografia che confermano l'identità del possessore e si collegano alla sua criptovaluta.

Ma quali sono i rischi nell'usare una criptovaluta come Bitcoin? Purtroppo esse sono ancora relativamente nuove, e il loro mercato è molto volatile. Infatti, dal momento che le criptovalute non hanno bisogno di banche o terze parti per essere regolate, tendono a non essere assicurate e sono difficili da convertire in una valuta tangibile (come dollari USA o euro). Inoltre, poiché sono risorse basate sulla tecnologia, possono essere violate come qualsiasi altra risorsa tecnologica immateriale (per esempio "rubando" le chiavi d'accesso al wallet virtuale). Infine, dal momento che esse sono memorizzate in un portafoglio digitale, la perdita del portafoglio (o l'accesso ad esso o ad uno dei suoi backup) implica necessariamente la perdita dell'intero investimento nella criptovaluta (cosa che è effettivamente accaduta a James Howells, un 36enne di Newport, nel sud del Galles, che ha cestinato per errore un disco rigido nel quale erano archiviate le chiavi private di un portafoglio digitale con circa 8000 Bitcoin, l'equivalente di 179 milioni di euro).

Sebbene le criptovalute siano un argomento piuttosto in voga solo da un paio d'anni, la nascita di Bitcoin risale a ben 13 anni fa: infatti, dopo l'introduzione della prima Blockchain avvenuta nel 2008 per opera di Satoshi Nakamoto (si tratta tuttavia solo di uno pseudonimo, la persona o il gruppo che vi si celano dietro è tuttora oggetto di speculazioni), la piattaforma venne implementata nel 2009 affinché fungesse da registro per le transazioni della nascente criptovaluta Bitcoin: proprio in quell'anno Bitcoin venne utilizzata per la prima volta per l'acquisto di una pizza.

Bitcoin, a causa della sua esistenza decentralizzata e del processo elettronico, come già detto non è regolamentato o controllato da alcun governo o banca centrale. L'obiettivo principale di questa criptovaluta è promuovere le transazioni di beni e servizi. Bitcoin si è evoluto immensamente ed è riuscito ad attrarre un vasto numero di utenti, guadagnando un'immensa popolarità grazie alla sua regolare menzione e diffusione sui media. Proprio a causa della sua popolarità risulta interessante creare un modello in grado di prevederne il prezzo, che fluttua costantemente in tempo reale come una borsa valori, utilizzando i dati presi dai social media (Twitter nel nostro caso). Infatti, se potesse essere previsto con un ragionevole grado di precisione, potrebbe risultare una conoscenza molto vantaggiosa per tutti quegli investitori, uomini

d'affari, banche, organizzazioni, ecc. che lo utilizzano per le loro transazioni. Concludiamo con un dato interessante riguardo l'estrazione di Bitcoin: l'estrazione delle criptovalute è un processo ad intensità energetica estremamente alta, che minaccia la capacità dei governi di tutto il mondo di ridurre la nostra dipendenza dai combustibili fossili che riscaldano il clima.

In una ricerca di Earthjustice, una delle principali organizzazioni no-profit di diritto ambientale di interesse pubblico¹, si stima infatti che l'estrazione di criptovalute alimentata da combustibili fossili abbia aumentato l'inquinamento dell'aria, dell'acqua e del rumore locale, in generale innalzando l'inquinamento climatico in un momento in cui dovremmo fare tutto ciò che è in nostro potere per muoverci nella direzione opposta e per riuscire a contrastare gli effetti della crisi climatica.

¹Presentato dal programma per l'energia pulita di Earthjustice in collaborazione con il programma di diritto ambientale del Sierra Club nell'anno 2022. Tutti i riferimenti si trovano nella sezione "Fonti e Bibliografia"

Capitolo 3

Come utilizzare il Text Mining per predire l'andamento di Bitcoin

3.1 Tecniche pratiche

In questo capitolo discuteremo di come poter utilizzare l'analisi testuale e del sentimento per poter predire l'andamento di Bitcoin, quali strategie vengono generalmente adottate e quali sono i pregi e i difetti di ciascun approccio.

Come già anticipato, il text mining è un processo automatico che utilizza l'elaborazione del linguaggio naturale per estrarre informazioni da testo non strutturato. Trasformando i dati grezzi in informazioni quantitative che le macchine possono comprendere, il text mining automatizza il processo di classificazione dei testi per sentimento, argomento e intento.

In generale, il text mining utilizza quattro diversi metodi:

1. **Metodo basato sui termini:** questo metodo consiste nell'analizzare un documento in base a uno o più termini che esso contiene. Il termine può avere un qualche valore o significato particolare nel contesto da analizzare. Ogni termine è associato a un valore, noto come "peso". Questo metodo, tuttavia, presenta due problemi:
 - (a) Polisemia (un termine potrebbe avere più significati possibili)
 - (b) Sinonimia (parole diverse potrebbero avere lo stesso significato).
2. **Metodo basato su frasi:** come indica il nome, questo metodo analizza un documento basandosi sulle frasi: esse contengono più informazioni di un singolo termine, perché sono una raccolta di termini semantici, inoltre sono più descrittive e meno ambigue. Tuttavia neanche questo metodo è privo di problemi, infatti alcune problematiche

piuttosto comuni sono la bassa frequenza di occorrenze di alcuni termini e la presenza di frasi ridondanti o “noisy”, termine utilizzato per indicare quelle frasi archiviate elettronicamente che non possono essere classificate correttamente da un programma di estrazione di testo.

3. **Metodo basato sui concetti:** nel metodo basato su concetti i termini sono previsti o indovinati a livello di frase o documento. Piuttosto che un’analisi di un singolo termine, questo modello cerca infatti di analizzare un termine alla luce dell’intero documento o frase, trovando in modo appropriato un termine significativo corrispondente. Questo modello avviene in tre fasi:
 - Esame della costruzione semantica delle frasi.
 - Costruzione di un grafo ontologico concettuale (COG) per descrivere la struttura semantica trovata.
 - Estrazione dei concetti principali in base alle prime due fasi per creare vettori di funzionalità tramite l’implementazione di un modello di spazio vettoriale standard.
4. **Metodo della tassonomia del modello:** in questo metodo, un documento viene analizzato in base a un modello, ovvero una relazione tra termini, per formare una tassonomia, ovvero una struttura ad albero. L’approccio basato su modelli può migliorare l’accuratezza del sistema per l’assegnazione dei pesi dei singoli termini, perché i modelli scoperti sono in grado di cogliere meglio il significato generale del testo analizzato. I modelli possono essere scoperti utilizzando tecniche di data mining come il mining di pattern chiuso, il mining di pattern sequenziale, il mining di set di elementi frequente e il mining di regole di associazione; tuttavia in questa trattazione non entreremo nel dettaglio di queste tecniche e ci limiteremo ad enunciarle. La tecnica basata su modelli utilizza due processi di distribuzione ed evoluzione del modello, perfezionando i modelli scoperti nel documento analizzato. Tutto il processo di estrazione di testo segue specifici passaggi:
 - Raccolta di informazioni: i dati testuali provenienti da una o più fonti e che sono in un formato semi-strutturato o non strutturato vengono raccolti per eseguire il text mining.
 - Conversione in dati strutturati: la preelaborazione comporta la pulizia dei dati raccolti (di questo parleremo meglio in seguito).
 - Identificazione del modello: in questa fase vengono applicate varie tecniche per estrarre informazioni significative.
 - Analisi del modello: i dati ottenuti vengono analizzati per estrarne conoscenza e significato.

- Analisi avanzata: ottenute le conoscenze richieste, queste possono quindi essere utilizzate per ulteriori analisi

Nella fase di identificazione del modello vengono eseguite diverse tecniche di estrazione di testo. Esse consistono generalmente in una serie di passaggi standard, che elencheremo qua sotto, ma possono avere finalità diverse, che vedremo in seguito. Le parti fondamentali sono dunque le seguenti:

- Raggruppamento
- Purificazione del testo
- Analisi fattoriale (insieme di tecniche statistiche utilizzate per ricerare l'esistenza di variabili latenti a partire una serie di variabili osservate)
- Classificazione del testo
- Riassunto del testo
- Ricerca di documenti simili
- Ricerca di un'associazione tra i termini
- Ricerca dei termini che ricorrono più comunemente.

Tecniche di estrazione di testo popolari:

1. **Recupero delle informazioni (IR):** Il recupero delle informazioni (IR) è la tecnica più famosa utilizzata nel text mining e si riferisce alla ricerca e alla raccolta di informazioni rilevanti da una varietà di risorse, solitamente documentate in un formato non strutturato. È un insieme di metodi o approcci atti a recuperare il materiale ricercato dall'utente (tramite una query, una richiesta) da una raccolta di database. IR aiuta a estrarre modelli pertinenti e associati in base a un determinato insieme di parole o frasi. In questa tecnica di estrazione di testo, i sistemi IR utilizzano diversi algoritmi per tracciare e monitorare i comportamenti degli utenti e scoprire di conseguenza i dati rilevanti. I motori di ricerca Google e Yahoo sono i due sistemi IR più famosi.
2. **Estrazione di informazioni (IE):** L'estrazione di informazioni (IE) è una tecnica utilizzata per estrarre automaticamente un'informazione definita e strutturata da dati non strutturati o semistrutturati sotto forma di testo utilizzando l'elaborazione del linguaggio naturale. Viene utilizzato per estrarre entità dal testo, come nomi di persone, organizzazioni, posizione e relazioni tra entità, attributi, eventi e caratteristiche specifiche. Le informazioni estratte sono ben organizzate (strutturate) e archiviate in un database per un ulteriore utilizzo. Sintetizzando, possiamo dire che IE estraie attributi ed entità specifici dal

documento e ne stabilisce la relazione. Il processo utilizzato per verificare e valutare la pertinenza dei risultati è chiamato "Precisione e richiamo".

3. **Categorizzazione del testo:** Questa tecnica serve a individuare certe categorie predefinite in documenti a testo libero di varia natura. Lo scopo della classificazione/categorizzazione del testo è di filtrare i documenti nel modo migliore possibile e categorizzarli nel modo corretto. Un'applicazione molto diffusa della categorizzazione del testo è il filtro antispam, in cui i messaggi di posta elettronica vengono classificati rispettivamente nelle due categorie di spam e non spam. Per portare un ulteriore esempio, anche i referti dei pazienti nelle organizzazioni sanitarie sono spesso indicizzati da molteplici aspetti, utilizzando tassonomie di categorie di malattie, tipi di procedure chirurgiche, codici di rimborso assicurativo e così via. Si tratta dunque di una forma di apprendimento "supervisionato", in cui i normali testi in lingua sono assegnati a un insieme predefinito di argomenti a seconda del loro contenuto. Pertanto, la categorizzazione o piuttosto il Natural Language Processing (NLP) è un processo di raccolta di documenti di testo, elaborazione e analisi per scoprire gli argomenti o le categorie giuste per ciascun documento.
4. **Raggruppamento di documenti ("clustering"):** Questa tecnica viene utilizzata per trovare gruppi di documenti con contenuto simile. Fa uso di descrittori e di estrazione di descrittori che sono essenzialmente delle "etichette", insiemi di parole che descrivono i contenuti all'interno del cluster (raggruppamento). È un processo non supervisionato responsabile della classificazione degli oggetti in gruppi chiamati cluster, ciascuno dei quali contiene più documenti. Eventuali etichette associate agli oggetti sono ottenute esclusivamente dai dati. Il vantaggio di questa tecnica è che garantisce che nessun documento venga perso nei risultati di ricerca poiché i documenti possono emergere in numerosi sotto-argomenti. Ad esempio, se il clustering viene eseguito su una raccolta di articoli di notizie, può assicurarsi che documenti simili siano tenuti più vicini tra loro o si trovino nello stesso cluster.
5. **Visualizzazione del testo (summarisation):** La visualizzazione del testo è una tecnica che rappresenta grandi informazioni testuali in una mappa visiva, espediente che fornisce funzionalità di navigazione avanzate insieme a una semplice ricerca. Nel text mining, i metodi di visualizzazione aiutano a migliorare e semplificare la scoperta di informazioni rilevanti. I flag di testo vengono utilizzati per mostrare la categoria del documento, per rappresentare singoli documenti o gruppi di documenti e i colori vengono utilizzati per mostrare la densità. L'estrazione di testo visivo inserisce grandi fonti testuali in una

gerarchia visiva appropriata, che aiuta l'utente a interagire con il documento ridimensionando e zoomando e in generale, garantendo una visione d'insieme dei dati.

Technique	Characteristics	Tools
Retrieval	Retrieves valuable information from unstructured text	Intelligent Miner, Text Analyst
Extraction	Extract information from structured database	Text Finder, Clear Forest Text
Summarization	Reduce length by keeping its main points and overall meaning as it is	Tropic Tracking Tool, Sentence Ext Tool
Categorization	Document based categorization	Intelligent Miner
Cluster	Cluster collection of documents, Clustering, classification and analysis of text document	Carrot, Rapid Miner

Figura 3.1: Fonte: upGrad, scritto dall'analista dati Abhinav Rai.s

3.2 Alcuni studi

Due economisti dell'Università di Yale: Aleh Tsvyinski e Yukun Liu, in uno studio intitolato "Rischi e rendimenti delle criptovalute", spiegano come il mercato volatile di Bitcoin (e in generale delle criptovalute) possa essere interpretato e anticipato utilizzando l'effetto "attenzione degli investitori", che si basa su intuizioni simili a quelle sviluppate in questa trattazione. Esaminando le variazioni di Google Search per le parole chiave Bitcoin, Ripple e Ethereum e confrontando l'esito settimanale con i rispettivi dati di prezzo, i due economisti hanno scoperto che l'aumento medio delle query di ricerca per ciascuna valuta indicherebbe che il suo prezzo aumenterà nelle prossime settimane, come in parte già suggeriva uno studio del software Semrush.

Allo stesso modo, l'attenzione negativa degli investitori, associata a parole chiave come "Bitcoin hack", prevederebbe invece una diminuzione del prezzo. Anche su Twitter è riscontrabile un simile meccanismo: secondo Tsvyinski e Liu l'aumento del numero di post su bitcoin su Twitter è stato quasi sempre il seguito da un rialzo nelle successive settimane: "Un aumento di una deviazione standard del conteggio delle post su Twitter per la parola "Bitcoin" produce un aumento del 2,50 percento nei rendimenti Bitcoin di 1 settimana prima". "I nostri risultati – afferma lo studio - mettono in dubbio le spiegazioni diffuse che il comportamento delle criptovalute dipenda dal loro legame con la tecnologia blockchain, come capiterebbe con azioni, valute, o come depositi di valore come materie prime e metalli preziosi", tuttavia prosegue

affermando che esistono altre strade per prevedere l'andamento dei prezzi, come ad esempio l'effetto "attenzione degli investitori".

Anche il sopraccitato studio di Semrush arriva a conclusioni simili: citando le parole del suo direttore strategico, Eugene Levin: "Il volume di ricerca di Google è un buon indicatore di attenzione generale e popolarità, mostra quante persone sono interessate a saperne di più su determinati argomenti", e aggiunge che: "a volte l'attenzione e la popolarità sono in correlazione con la domanda e i prezzi di valute e azioni".

Capitolo 4

Lavoro svolto e risultati

4.1 Breve introduzione a TextBlob

Esaurita la parte introduttiva di questa trattazione, entriamo ora nello specifico del programma, e in particolare iniziamo da TextBlob, una libreria di Python che ho utilizzato per valutare la polarità. Come riportato nella documentazione di Python: “TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more”. (TextBlob è una libreria Python (2 e 3) per l’elaborazione di dati testuali. Fornisce una semplice API per approfondire le comuni attività di elaborazione del linguaggio naturale (NLP) come la codifica di parti del discorso, l’estrazione di frasi nominali, l’analisi del sentimento, la classificazione, la traduzione e altro ancora).

In particolare, la funzione di questa libreria che ci interessa utilizzare è la polarità:

```
TextBlob().sentiment.polarity
```

Quando si calcola un sentimento per una singola parola, TextBlob utilizza la tecnica della "media", che viene applicata ai valori di polarità per calcolare un punteggio di polarità per una singola parola, e quindi una procedura simile si applica a ogni singola parola, risultando in una polarità combinata per testi più grandi.

TextBlob comprende anche le negazioni e la polarità è raddoppiata di -0,5. TextBlob ha una caratteristica intrigante in quanto gestisce i modificatori, noti anche come intensificatori, che intensificano il significato del testo in base al suo schema. TextBlob ignora la polarità e la soggettività quando viene inclusa una parola modificatore, invece di fare affidamento esclusivamente sull’intensità per calcolare il sentimento del testo.

4.2 Altre librerie utilizzate

Oltre textblob il programma fa uso di altre librerie, tutte importate nelle prime righe di codice:

```
import tweepy
from textblob import TextBlob
import pandas as pd
import re
import matplotlib.pyplot as plt
import datetime
plt.style.use('fivethirtyeight')
import json
```

Brevemente, introduciamo ciascuna libreria e spieghiamo per cosa viene utilizzata:

- **Tweepy:** Tweepy è una libreria open source che permette di accedere all'API di Twitter tramite Python. Tramite l'API di Twitter è possibile accedere a quasi tutte le funzionalità di Twitter come like, follower, tweet, retweet, ecc. Le specifiche e le limitazioni di questa libreria sono stati determinanti nella scrittura del codice e in alcune scelte di tipo pratico. Approfondiremo in seguito questo argomento.
- **Pandas:** pandas è una libreria utilizzata per la manipolazione e l'analisi dei dati. In particolare, offre strutture dati e operazioni di manipolazione di tabelle numeriche e serie temporali. Il nome deriva dal termine "panel data", un termine econometrico usato per indicare set di dati che includono osservazioni fatte sullo stesso campione in periodi di tempo diversi.
- **Re:** Supporto per le espressioni regolari (RE).
Un'espressione regolare (o RE) specifica un insieme di stringhe che le corrispondono; le funzioni in questo modulo consentono di verificare se una determinata stringa corrisponde a una determinata espressione regolare (o viceversa).
- **Matplotlib.pyplot:** matplotlib.pyplot è un'interfaccia basata sullo stato (state-based) per matplotlib. “Interfaccia basata sullo stato” significa che le funzioni di pyplot agiranno su uno stato attualmente definito: questo approccio è fondamentalmente diverso da uno orientato agli oggetti. In particolare, in matplotlib.pyplot vengono preservati vari stati attraverso le chiamate di funzione, in modo che si tenga traccia della figura corrente e dell'area di tracciamento e che le funzioni di tracciamento siano dirette agli assi correnti. Questa libreria fornisce un modo di tracciare implicito, simile a MATLAB, inoltre apre

le figure sullo schermo e funge da gestore della GUI delle figure. Pyplot è principalmente destinato a grafici interattivi e semplici casi di generazione di grafici programmatici. L'API esplicita (orientata agli oggetti) è consigliata per grafici complessi, sebbene pyplot sia ancora solitamente utilizzato per creare figure e spesso gli assi nei grafici.

- **Datetime:** Il modulo datetime fornisce classi per manipolare date e orari.

Sebbene sia supportata l'aritmetica di data e ora, l'obiettivo dell'implementazione è l'estrazione efficiente degli attributi per la formattazione e la manipolazione dell'output.

- **Json:** pacchetto utilizzato per poter lavorare con file con estensione .json.

4.3 Il programma

L'idea centrale di questo lavoro è di poter accedere ai tweets pubblicati giornalmente riguardo Bitcoin, con lo scopo di analizzarne il sentimento e ricavare una media giornaliera, che verrà poi usata come ordinata in un grafico che contenga in ascissa i giorni i cui questo lavoro è stato eseguito.

L'arco temporale della ricerca è stato di circa due mesi, da fine settembre a fine novembre 2022, e per ciascun giorno sono stati raccolti 10000 tweets.

L'estrazione dei tweets non è così intuitiva come potrebbe sembrare, anzi è necessario studiare attentamente l'API di Twitter e creare un account da sviluppatore: questo permette di sbloccare funzionalità, come l'estrazione di messaggi, non consentite ad utenti standard.

Sono inoltre disponibili diversi tipi di account da sviluppatore, ciascuno con differenti livelli di accesso ai dati, si veda la Figura 4.1.

Per ottenere un account con funzionalità maggiori rispetto a quelle di base da sviluppatore (“Essential account”), è necessario presentare una richiesta formale a Twitter specificando il motivo per cui si richiede un accesso più elevato e rispondere ad alcune domande. In questo programma è stato utilizzato un account di tipo “Elevated”.

Il numero massimo di richieste per ciascun progetto (ovvero di tweets che possono essere estratti) per questo tipo di account è abbastanza alto, 2 milioni di messaggi ogni mese, tuttavia è necessario tenere in considerazione che, se per estrarre 10000 messaggi un normale computer non impiegherebbe più di qualche minuto, ogni 2000 richieste (circa, in realtà il numero può variare di alcune centinaia in base ad altri fattori) è necessario aspettare circa 15 minuti prima di poter ricominciare ad estrarre altri messaggi. Risulta dunque chiaro quanto in realtà sia oneroso estrarre un numero consistente di tweets, soprattutto considerando che lo schermo del device utilizzato

	Essential	Elevated	Elevated+ (coming soon)	Academic Research
Getting access	Sign up	Apply for additional access within the developer portal	Need more? Sign up for our waitlist	Apply for additional access
Price	Free	Free		Free
Access to Twitter API v2	✓	✓		✓
Access to standard v1.1	✓ (Limited access - only media endpoints)	✓		✓
Access to premium v1.1	✗	✓		✓
Access to enterprise	✗	✓		✓
Project limits	1 Project	1 Project		1 Project
App limits	1 App per Project	3 Apps per Project		1 App per Project
Tweet caps	Retrieve up to 500k Tweets per month	Retrieve up to 2 million Tweets per month		Retrieve up to 10 million Tweets per month
Filtered stream rule limit	5 rules	25 rules		1000 rules
Filtered stream rule length	512 characters	512 characters		1024 characters
Filtered stream POST rules rate limit	25 requests per 15 minutes	50 requests per 15 minutes		100 requests per 15 minutes
Search Tweets query length	512 characters	512 characters		1024 characters
Access to full-archive search Tweets	✗	✗		✓
Access to full-archive Tweet counts	✗	✗		✓
Access to advanced filter operators	✗	✗		✓

Figura 4.1: Livelli di accesso per i diversi account sviluppatore.

per l'estrazione deve restare acceso durante tutto il procedimento e che anche la connessione internet dev'essere stabile per evitare che la raccolta si interrompa bruscamente.

Il programma in sé si articola in realtà in più file diversi, ciascuno utilizzato per una specifica esigenza: il codice “principale” è contenuto nel Main, che è dove avviene l'intero processo di raccolta e analisi dei dati. È dunque anche il file più lungo e quello su cui vengono utilizzate le varie librerie elencate in precedenza.

Un altro file è quello che contiene le credenziali segrete per accedere all'account da sviluppatore di Twitter: generalmente esse possono essere inserite anche nel Main se questo non viene divulgato; tuttavia, è sempre raccomandato creare un file a parte in cui salvarle e da cui richiamarle nel momento del bisogno. In particolare, si tratta di un file con estensione .json, non .py, dove json (“JavaScript Object Notation”) è un formato standard basato su testo per la rappresentazione di dati strutturati basati sulla sintassi dell'og-

getto JavaScript. Viene comunemente utilizzato per la trasmissione di dati nelle applicazioni Web o per archiviare dati temporanei. Inoltre, i dati temporanei possono essere dati generati dall'utente, come un modulo inviato su un sito web. Infine, JSON può anche essere utilizzato come formato dati per qualsiasi linguaggio di programmazione per fornire un elevato livello di interoperabilità.

Un altro file, sempre con l'estensione .json, è quello che raccoglie il risultato del Main, ovvero la media giornaliera del sentimento. La struttura in cui ciascun sentimento viene associato ad una specifica data è identica ad un dizionario in Python, ovvero un insieme di coppie in cui ad una “key” (la data) viene associato un “value” (il sentimento medio).

L'ultimo step di questo processo viene eseguito dal file “tabella.py”, che costruisce un grafico estraendo i dati dal file data.json in cui sono salvati. Anche questo programma fa uso di librerie, in particolare:

```
import matplotlib.pyplot as plt
import numpy as np
import json
```

Dopo questa introduzione generale, entriamo più nello specifico del Main: dopo aver importato le varie librerie, le successive 30 righe circa di codice servono per importare le credenziali Twitter e accedere all'API e sono abbastanza standard. In particolare, Twitter richiede 4 chiavi di accesso:

- accessToken
- accessTokenSecret
- consumerKey
- consumerSecrets

In seguito vengono impostati i parametri della ricerca, ovvero il numero di tweets e il giorno in cui cercarli, e vengono inizializzati alcuni oggetti che verranno utilizzati in seguito per conservare i messaggi estratti e i relativi dati (numero di “like” ricevuti e ora in cui sono stati creati). È importante specificare che come data di ricerca è necessario scegliere la data successiva a quella a cui siamo effettivamente interessati: infatti Twitter restituirà esclusivamente messaggi creati a partire dalle 23.59.59 del giorno precedente a quello inserito, andando indietro di massimo 24 ore.

La raccolta dei tweets avviene poco dopo e la funzione utilizzata permette di specificare alcuni parametri della ricerca: il termine su cui vogliamo effettuare la ricerca (“q”), la lingua in cui vogliamo che siano scritti i tweets (“lang”), la lunghezza del testo restituito (“tweet_mode”), che nel nostro caso restituisce il testo intero senza troncamenti, e infine il numero di tweets a cui siamo interessati.

```
for i in tweepy.Cursor(api.search_tweets, q="bitcoin", lang='en',
    tweet_mode="extended").items(number_of_tweets):
```

Dopo averli raccolti, puliamo il testo dei messaggi affinché il lavoro di analisi degli stessi non risulti inutilmente oneroso al programma, rimuovendo alcuni termini che non contribuiscono all'analisi del sentimento (links, hashtags, menzioni ecc).

A questo punto creiamo un dataframe con 4 colonne: una che contenga i tweets ripuliti, una che contenga il numero di likes, una per memorizzare la data e una per l'ora. Queste ultime in particolare provano che il programma sta funzionando a dovere e sta effettivamente raccogliendo i dati del giorno a cui siamo interessati. Anche l'ora è interessante per capire il range temporale in cui vengono prodotti 10000 messaggi riguardo Bitcoin nel mondo.

Successivamente si trovano alcune definizioni di funzioni utilizzate per implementare nel programma l'analisi del sentimento: utilizziamo quindi TextBlob per ottenere la polarità dei tweets e la salviamo in un'ulteriore colonna, che aggiungeremo al nostro Dataframe.

Tuttavia, è intuitivo immaginare che tweets visti e apprezzati da più persone debbano avere un peso maggiore rispetto a tweets che abbiano avuto un minore successo sulla piattaforma: per valutare questo “apprezzamento” utilizziamo i likes.

In particolare, pesiamo ciascun messaggio in base al numero di “mi piace” ricevuti. Per capire il motivo di una simile scelta basta paragonare un tweet pubblicato da un utente qualsiasi ad un tweet di Elon Musk, entrambi su Bitcoin: chiaramente il secondo verrà visto, e dunque influenzerà, molte più persone del primo, ed è quindi sensato che ai fini della nostra analisi abbia un peso maggiore.

Infine si calcola la media dei sentimenti di tutti i tweets e la si salva come un dato nel file .json che raccoglie la polarità giornaliera.

4.4 Risultati (confronti)

Dopo aver raccolto un numero consistente di dati per circa due mesi, che conclusioni possiamo trarre da questa ricerca?

Per capire se il sentimento può essere un buon indicatore per predire l'andamento di Bitcoin è possibile innanzitutto fare un primo confronto paragonando il grafico ottenuto utilizzando le medie giornaliere del sentimento (in Figura 4.2) col grafico effettivo di Bitcoin (in Figura 4.3).

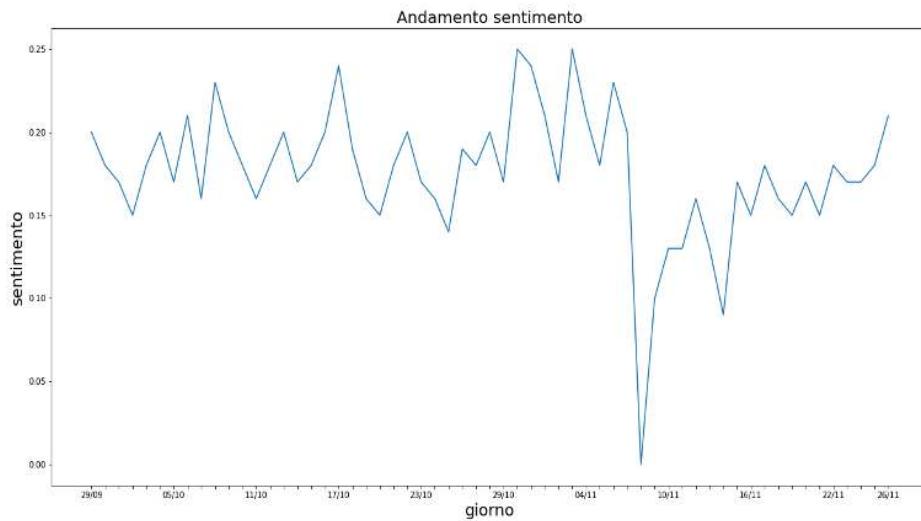


Figura 4.2: Grafico ottenuto tramite il programma.



Figura 4.3: Grafico effettivo.

Ciò che otteniamo sovrapponendo le due immagini è riportato in Figura 4.4.



Figura 4.4: Confronto dei due grafici precedenti.

Già ad un’analisi superficiale appare chiaramente una forte somiglianza fra i due grafici, ma naturalmente ciò non è sufficiente a mostrare l’effettiva correlazione fra il sentimento medio e l’andamento del grafico reale.

Un’altra osservazione da fare consiste nel notare la presenza di un break strutturale, ovvero uno shift nella serie storica, subito dopo il 06 novembre: infatti da quel momento in poi il grafico subisce un “crollo” e sarebbe necessario aggiungere un’ulteriore variabile dummy per predire correttamente l’andamento da tale punto in avanti. Nel nostro studio ci limiteremo dunque a studiare i dati in nostro possesso fino a quel momento.

Infine, proporremo due regressioni, una sul prezzo di chiusura di Bitcoin e l’altra sul rendimento. Entrambe verranno confrontate col sentimento del giorno stesso e col sentimento del giorno precedente, per individuare eventuali correlazioni e/o capacità “predittive” dei sentimenti. Per studiare tale correlazione utilizzeremo alcuni strumenti statistici avvalendoci, per comodità, del software R.

Innanzitutto importiamo da un file .csv lo storico prezzi di Bitcoin del periodo a cui siamo interessati in forma di dataset e salviamo in alcune variabili la data, il prezzo di chiusura e i rendimenti di ciascuna giornata. In un’altra variabile salviamo invece un array coi valori di sentimento salvati nel file data.json. Infine costruiamo i dataframe dai quali raccoglieremo poi i dati per le varie operazioni.

Iniziamo analizzando la correlazione fra prezzi e sentimenti:

	sentimento_medio	close
sentimento_medio	1.0000000	0.4457767
close	0.4457767	1.0000000

Notiamo che c'è una correlazione positiva abbastanza forte fra le due variabili (circa 0.45).

In generale, se vogliamo etichettare la forza della correlazione per valori assoluti di r , 0-0.19 è considerato molto debole, 0.2-0.39 debole, 0.40-0.59 moderato, 0.6-0.79 forte e 0.8-1 molto forte. Tuttavia questi sono limiti piuttosto arbitrari e bisogna sempre tenere in considerazione anche altri fattori. Da questa analisi ci aspettiamo quindi che la correlazione possa essere approssimata moderatamente bene con una retta.

Valutiamo ora la correlazione fra i prezzi al tempo t e i sentimenti al tempo $t - 1$:

	sentimento_medio_t_1	close
sentimento_medio_t_1	1.0000000	0.3372279
close	0.3372279	1.0000000

Riportiamo anche, per completezza, un'osservazione: valutando la medesima correlazione in un arco temporale maggiore, e dunque includendo anche i dati post-break, otteniamo un indice di correlazione molto maggiore (nello specifico, pari a circa 0.51). Questo potrebbe indicare due cose: la correlazione maggiore potrebbe infatti essere casuale e legata all'imprecisione dei dati, oppure può anche darsi che il numero, derivato da una maggiore quantità di dati, sia più preciso.

Studiamo ora la correlazione fra i rendimenti e il sentimento. Ricordiamo che, in generale, il rendimento al tempo t è dato da:

$$\frac{\text{prezzo}(t) - \text{prezzo}(t - 1)}{\text{prezzo}(t - 1)}$$

In questo caso studiamo la correlazione fra rendimenti al tempo t e sentimenti al tempo t , $t-1$, $t-2$:

	rendimenti1	sentimento_medio_t
rendimenti1	1.0000000	0.0530146
sentimento_medio_t	0.0530146	1.0000000
	rendimenti2	sentimento_medio_t_1
rendimenti2	1.0000000	-0.1764223
sentimento_medio_t_1	-0.1764223	1.0000000
	rendimenti3	sentimento_medio_t_2
rendimenti3	1.0000000	-0.4259444
sentimento_medio_t_2	-0.4259444	1.0000000

Come possiamo notare, non solo non vi è una correlazione significativa fra le due variabili, ma anzi pare esserci addirittura correlazione negativa quando si tenta di utilizzare il sentimento per effettuare la predizione.

Per provare i risultati ottenuti di scarsa correlazione fra rendimenti e sentimento, mostriamo i risultati della regressione lineare su queste due variabili: come possiamo notare in questo caso il sentimento medio non è una variabile significativa.

```
Call:
lm(formula = rendimenti ~ sentimento_medio, data = dati_rend)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.4656 -1.2270 -0.5058  0.8376  4.3794 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.4128     1.9377  -0.213   0.832    
sentimento_medio 3.2597    10.0941   0.323   0.749    
                                                        
Residual standard error: 1.744 on 37 degrees of freedom
Multiple R-squared:  0.002811, Adjusted R-squared:  -0.02414 
F-statistic: 0.1043 on 1 and 37 DF,  p-value: 0.7486
```

4.5 Analisi di regressione

Per stimare la relazione funzionale esistente tra la variabile dipendente (*close*) e quella indipendente (*sentimento_medio*) studiamo la regressione lineare operata su queste due variabili:

```
Call:
lm(formula = close ~ sentimento_medio, data = dati_close)

Residuals:
    Min      1Q  Median      3Q     Max 
-728.6 -311.8 -220.6  336.3 1271.0 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 18510.1     536.1  34.530 < 2e-16 ***
sentimento_medio 8459.0    2792.5   3.029  0.00445 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Residual standard error: 482.5 on 37 degrees of freedom
 Multiple R-squared: 0.1987, Adjusted R-squared: 0.1771
 F-statistic: 9.176 on 1 and 37 DF, p-value: 0.004453

Innanzitutto, per valutare la significatività della variabile, eseguo un test t sul coefficiente della variabile indipendente “sentimento_medio”:

$$\frac{\beta_1}{SE} : \frac{8459.0}{2792.5} = 3.03$$

e lo confronto con il quantile della normale standard con livello di significatività pari al 5% (che vale 1.96). Poiché $3.03 > 1.96$ rifiuto H_0 , quindi il coefficiente è significativamente diverso da 0 (ed è pertanto significativo). Per arrivare alla medesima conclusione si poteva anche far riferimento al valore del p-value della variabile il quale, essendo approssimativamente nullo, ci porta a concludere che la variabile è significativa.

Riportiamo dunque in Figura 4.5 il grafico ottenuto mettendo in relazione le due variabili oggetto del nostro studio.

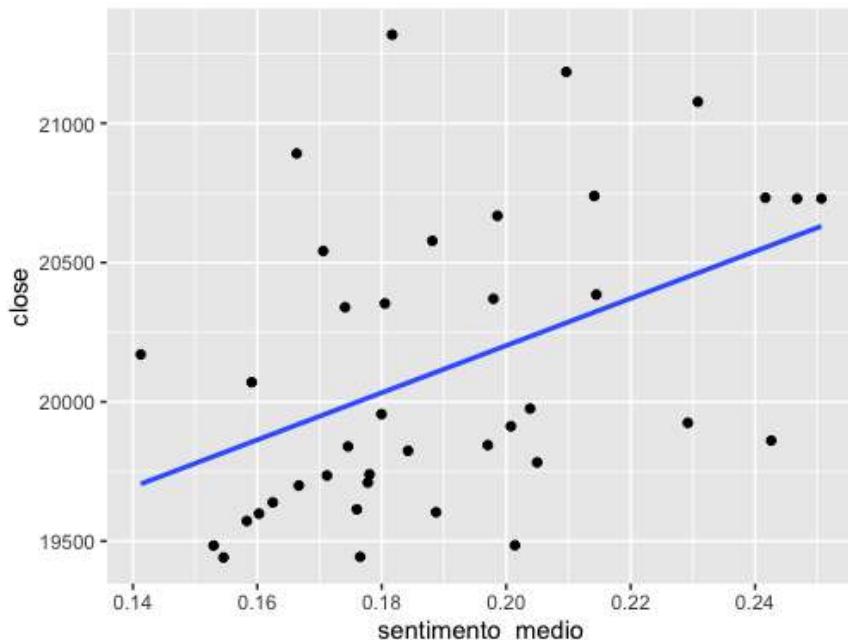


Figura 4.5: Regressione del prezzo di chiusura sul sentimento utilizzando solo i dati prima del break strutturale.

Possiamo confrontare questo grafico con quello ottenuto considerando tutti i dati, anche quelli successivi al break. Ciò che troviamo è il grafico mostrato in figura 4.6.

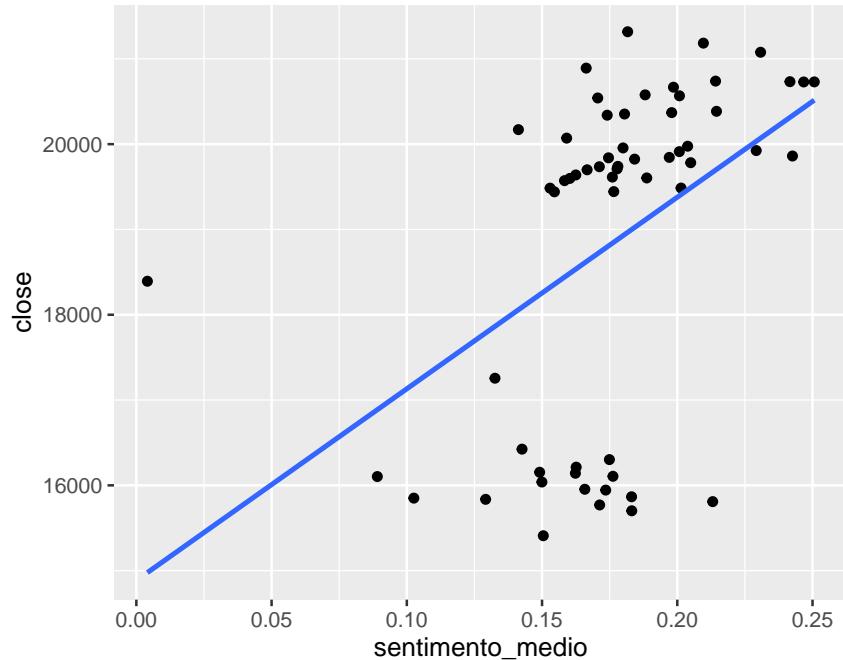


Figura 4.6: Regressione del prezzo di chiusura sul sentimento utilizzando tutti i dati raccolti.

Appare dunque netta la scissione fra i punti prima del break (in alto a destra), e quelli successivi (nella parte centrale, tutti al di sotto della retta di regressione).

In questa regressione ho messo in correlazione il sentimento medio col prezzo di chiusura del medesimo giorno, arrivando a concludere che c'è una correlazione temporalmente molto stretta fra le due variabili. Tuttavia vorremmo anche analizzare la capacità del sentimento di predire, seppur di un solo giorno, l'andamento del prezzo di Bitcoin.

Per fare ciò utilizziamo il medesimo programma su R, ma paragonando a ciascun prezzo il sentimento di tutta la settimana precedente. Ciò che ne emerge sono i seguenti risultati:

Residuals:

Min	1Q	Median	3Q	Max
-993.1	-356.5	-159.6	301.9	1362.9

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	12152	1637	7.424	2.84e-08	***
sentimento_medio7	1788	4710	0.380	0.706868	
sentimento_medio6	-3890	4589	-0.848	0.403322	

```

sentimento_medio5      3885      3090   1.257 0.218428
sentimento_medio4      4656      3055   1.524 0.137917
sentimento_medio3      5717      3063   1.866 0.071793 .
sentimento_medio2      5776      3193   1.809 0.080505 .
sentimento_medio1      11926     3148   3.789 0.000680 ***
sentimento_medio0      11644     2920   3.988 0.000395 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

Residual standard error: 711 on 30 degrees of freedom
 Multiple R-squared: 0.7647, Adjusted R-squared: 0.7019
 F-statistic: 12.19 on 8 and 30 DF, p-value: 1.453e-07

(NB: “sentimento_medio7” indica il sentimento al tempo $t-7$, quindi una settimana prima) Notiamo che regredendo nella linea temporale i coefficienti perdono significatività, e che quelli più significativi sono solamente quelli che corrispondono al tempo t e $t-1$. Osserviamo inoltre che questi 2 coefficienti si rivelano molto buoni per predire l’andamento del grafico effettivo e che la perdita di significatività aumenta quanto più ci allontaniamo dal giorno che ci interessa, suggerendo che seppur giusta l’ipotesi di una capacità predittiva del sentimento, essa è tuttavia limitata ad un arco temporale molto ristretto.

Capitolo 5

Conclusioni

Questa trattazione ha indagato la possibilità di poter prevedere le fluttuazioni dei prezzi della criptovaluta Bitcoin tramite i tweets estrapolati da Twitter. L'ipotesi alla base di questo lavoro è che le opinioni espresse nei social media possano funzionare come utili predittori di tali fluttuazioni, soprattutto nella misura in cui incorporano caratteristiche come il sentimento e l'opinione.

Naturalmente bisogna tenere in considerazione che queste stime non sono completamente ottimizzate: si potrebbero utilizzare più metodi combinati fra loro per cercare di aumentare il grado di precisione della previsione, inoltre potrebbero anche essere implementate alcune funzioni prettamente "strutturali": ad esempio, l'AI che si occupa di riconoscere il sentimento dei messaggi potrebbe essere migliorata, oppure potrebbe essere ridotto lo "sleeping time" che costringe ad attendere 15 minuti ogni 2000 tweets raccolti (circa), aumentando la quantità di tweets analizzabili.

Nonostante ciò, sembra fortemente plausibile l'idea che il prezzo dipenda dal sentimento degli utenti: ciò trova conferma anche in alcuni studi fatti in precedenza, i quali affermavano che l'analisi del sentimento migliorasse il grado di predizione di più del 60% (articolo del Financial Innovation intitolato "Bitcoin price change and trend prediction through twitter sentiment and data volume" del 05.05.2022, del quale lascio il link in nota¹). Dunque, che considerazioni possiamo trarre dai risultati di questa ricerca? In questa trattazione è stato riscontrato un buon grado di correlazione fra le due variabili e una grande somiglianza nei grafici e, in definitiva, i risultati ottenuti confermano la relazione esistente fra l'analisi del sentimento dei tweets e il prezzo di Bitcoin e suggeriscono la possibilità di predire con un buon grado di accuratezza l'andamento del prezzo di mercato nel giorno immediatamente successivo a quello corrente.

¹<https://jfin-swufe.springeropen.com/articles/10.1186/s40854-022-00352-7>

Appendice A

Codice main

```
import tweepy
from textblob import TextBlob
import pandas as pd
import re
import matplotlib.pyplot as plt
import datetime
plt.style.use('fivethirtyeight')
import json

with open("data.json", "r") as d:
    jsonData = json.load(d)

#allego link per tweepy documentation
# https://docs.tweepy.org/en/stable/api.html

#twitter API credentials
with open("credentials.json", "r") as d:
    credentials = json.load(d)
# dumps the json object into an element
json_str = json.dumps(credentials)
# load the json to a string
resp = json.loads(json_str)
# extract an element in the response

accessToken = resp["accessToken"]
accessTokenSecret= resp["accessTokenSecret"]
consumerKey = resp["consumerKey"]
consumerSecret = resp["consumerSecret"]

#create the authentication object
```

```

auth=tweepy.OAuthHandler(consumerKey, consumerSecret)

# set the access token and access token secret
auth.set_access_token(accessToken, accessTokenSecret)

#create the API object while passing in the auth information
api=tweepy.API(auth, wait_on_rate_limit = True)

number_of_tweets=10000
tweets=[]
likes=[]
time=[]
user=[]

giorno = '19/11/2022'
giornoDatetime = datetime.datetime.strptime(giorno, '%d/%m/%Y')

def tweetPerData():
    for i in tweepy.Cursor(api.search_tweets, q="bitcoin", lang='en',
                           tweet_mode="extended").items(number_of_tweets):
        tweets.append(i.full_text)
        likes.append(i.favorite_count)
        time.append(i.created_at)

    tweetPerData()

#create a function to clean the tweets
def cleanTwt(twt):
    twt=re.sub('RT',' ',twt) #remove 'RT' from the tweets
    twt=re.sub('#[A-Za-z0-9]+',' ',twt) #remove the '#' from the tweets
    twt=re.sub('\n',' ',twt) #remove '\n' from the tweets
    twt=re.sub('https?:\/\/\S+',' ',twt) #remove hyperlinks from the tweets
    twt=re.sub('@[\S]*',' ',twt) #remove @mentions from the tweets
    twt=re.sub('^\s+|\s+$',' ',twt) #remove leading and trailing whitespaces from the tweets

    return twt

#list with cleaned text
cleanedTweets = []
for i in range (len(tweets)):
    newTweet = cleanTwt(tweets[i])
    cleanedTweets.append(newTweet)

#create a list with days in datetime format for the dataframe

```

```

date= []
for i in range (len(time)):
    data = time[i].strftime("%d/%m/%Y")
    date.append(data)

#create a new dataframe
df=pd.DataFrame({'Cleaned_Tweets': cleanedTweets,
                 'likes':likes, 'time':date, 'ora':time})
#remove any duplicate rows
df.drop_duplicates(subset=['Cleaned_Tweets'], inplace=True)
idx=list(range(0,len(df)))
df=df.set_index(pd.Index(idx))
#create a function to get the polarity
def getPolarity(twt):
    return TextBlob(twt).sentiment.polarity
#create one new column for and polarity
df['Polarity']=df['Cleaned_Tweets'].apply(getPolarity)

def trovaMediaGiornoConLike(giorno):
    nElementi = 0
    media= 0
    for i in range (len(df['Cleaned_Tweets'])):
        if (datetime.datetime.strptime(df['time'][i],"%d/%m/%Y").day == giorno.day \
            and getPolarity(df['Cleaned_Tweets'][i]) !=0:
            nElementi = nElementi+df['likes'][i]+1
            media+= getPolarity(df['Cleaned_Tweets'][i])*(df['likes'][i]+1)
    media = media / nElementi
    return media

giornoInizialeString = giornoDatetime.replace(day=giornoDatetime.day-1)
giornoInizialeString = giornoInizialeString.strftime("%d/%m/%Y")

#updating the json file with the new value
jsonData[giornoInizialeString] = \
    trovaMediaGiornoConLike(giornoDatetime.replace(day=giornoDatetime.day-1))
with open("data.json", "w") as d:
    json.dump(jsonData, d)

df

```


Appendice B

File Tabella

```
import matplotlib.pyplot as plt
import numpy as np
import json

with open("data.json", "r") as d:
    data = json.load(d)
    # # dumps the json object into an element
    json_str = json.dumps(data)
    # # load the json to a string
    resp = json.loads(json_str)
    # # extract an element in the response

testo = json_str.split(",")

listaDate= []
listaValori= []
for i in range (len(testo)):

    testo2 = testo[i].split(":")

    testo3 = testo2[0].split("\\")

    testo2[0] = testo3[1]
    testo4 = testo2[0].split("/")
    testo2[0] = testo4[0]+"/"+testo4[1]
```

```
testo5 =testo2[1].split("}")  
testo2[1] = testo5[0]  
testo2[1] = round(np.double(testo2[1]),2)  
  
listaDate.append(testo2[0])  
listaValori.append(testo2[1])  
  
xpoints = np.array(listaDate)  
ypoints = np.array(listaValori)  
  
plt.figure(figsize=(20, 10))  
plt.plot(xpoints, ypoints)  
plt.title('Andamento sentimento', fontsize=20)  
plt.xlabel('giorno', fontsize=20)  
plt.ylabel('sentimento', fontsize=20)  
plt.show()
```

Appendice C

Codice R

```
library("tidyverse")
source("inference.r")

date=BTC.EUR8$date
high=BTC.EUR8$High
low=BTC.EUR8$Low
close=BTC.EUR8$Close

closegiornodopo=Data$closegiornodopo
close7giornidopo=Data$close7giornodopo

rendimenti=Data$rendimenti
rendimentigiornodopo=Data$rendimentigiornodopo
rendimenti2giornidopo=Data$rendimenti2giornidopo

sentimento_medio7=Data$sentimento_medio7
sentimento_medio6=Data$sentimento_medio6
sentimento_medio5=Data$sentimento_medio5
sentimento_medio4=Data$sentimento_medio4
sentimento_medio3=Data$sentimento_medio3
sentimento_medio2=Data$sentimento_medio2
sentimento_medio1=Data$sentimento_medio1
sentimento_medio0=Data$sentimento_medio0
sentimento_medio=Data$sentimento_medio

var_sentimento=Data$var_sentimento

#dataframe per analisi e correlazione coi RENDIMENTI
```

```

data_per_cor=data.frame(rendimenti,sentimento_medio)
data_per_cor1=data.frame(rendimentigiornodopo,sentimento_medio)
data_per_cor2=data.frame(rendimenti2giornidopo,sentimento_medio)
dati_rend=data.frame(date,rendimenti,var_sentimento, sentimento_medio,
                      rendimentigiornodopo)

cor(data_per_cor)
cor(data_per_cor1)
cor(data_per_cor2)

#dataframe per analisi e correlazione coi PREZZI DI CHIUSURA
data_per_cor_close=data.frame(sentimento_medio,close)
data_per_cor_close1=data.frame(sentimento_medio, closegiornodopo)
dati_close=data.frame(date,close,close7giornidopo,sentimento_medio0,
                      sentimento_medio1,sentimento_medio2,sentimento_medio3,
                      sentimento_medio4,sentimento_medio5,sentimento_medio6,
                      sentimento_medio7,closegiornodopo)

cor(data_per_cor_close)
cor(data_per_cor_close1)

#regressione sul prezzo pred 7 gg prima
m1 <- lm(close7giornidopo ~ sentimento_medio7+sentimento_medio6+
          sentimento_medio5+sentimento_medio4+sentimento_medio3+
          sentimento_medio2+sentimento_medio1+sentimento_medio0,
          data = dati_close)
summary(m1)
ggplot(data=dati_close,aes(x = sentimento_medio,
                            y = close7giornidopo)) +
  geom_point() +
  stat_smooth(method = "lm", se = FALSE)

#regressione sul prezzo
m1 <- lm(close ~ sentimento_medio, data = dati_close)
summary(m1)
ggplot(data=dati_close,aes(x = sentimento_medio, y = close)) +
  geom_point() +
  stat_smooth(method = "lm", se = FALSE)

#regressione sui rendimenti
m1 <- lm(rendimenti ~ sentimento_medio, data = dati_rend)
summary(m1)

```

```
ggplot(data=dati_rend,aes(x = sentimento_medio, y = rendimenti)) +  
  geom_point() +  
  stat_smooth(method = "lm", se = FALSE)  
  
#regressione sui rendimenti pred  
m1 <- lm(rendimentigiornodopo ~ sentimento_medio, data = dati_rend)  
summary(m1)  
ggplot(data=dati_rend,aes(x = sentimento_medio, y = rendimentigiornodopo)) +  
  geom_point() +  
  stat_smooth(method = "lm", se = FALSE)
```


Fonti e Bibliografia

Cap.1

- MonkeyLearn, “Sentiment Analysis: A Definitive Guide”, 2022
<https://monkeylearn.com/sentiment-analysis/>
- “A survey on sentiment analysis methods, applications, and challenges”,
07.02.2022
<https://link.springer.com/article/10.1007/s10462-022-10144-1>

Cap.2

- EarthJustice, “The Environmental Impacts of Cryptomining”, 09.2022
<https://earthjustice.org/features/cryptocurrency-mining-environmental-impacts>

Cap.3

- upGrad, “What is Text Mining: Techniques and Applications”, 06.10.2022
<https://www.upgrad.com/blog/what-is-text-mining-techniques-and-applications/>
- Wired, “Bitcoin, c’è un rapporto tra il prezzo e quanto lo cerchiamo su Google?”, 14.03.2018
<https://www.wired.it/economia/finanza/2018/03/14/prezzo-bitcoin-google/>
- Wired, “2 modi per prevedere l’andamento di bitcoin e criptovalute”,
04.09.2018
<https://www.wired.it/economia/finanza/2018/09/04/previsioni-bitcoin-criptovalute-tecnica/>

Cap.4

- AIM, “How to Obtain a Sentiment Score for a Sentence Using Text-Blob?”, 28.10.2021

<https://analyticsindiamag.com/how-to-obtain-a-sentiment-score-for-a-sentence-using-textblob/>

- Twitter API Documentation

<https://developer.twitter.com/en/docs/twitter-api>

- Python3 Documentation

<https://docs.python.org/3/>