Stock Market Sentimental Analysis

Domenico Antonio Izzo, Ciro Maccarone, Adelio Antonini

Settembre 5, 2023

1 Introduction

Il mercato azionario è una componente del sistema finanziario globale, rappresenta il luogo in cui gli investitori acquistano e vendono azioni di società a quotazione pubblica e riflette le condizioni generali dell'economia di un paese o anche globali. È di particolare interesse studiare il mercato azionario per differenti motivi:

- **Investimenti**: Individui e istituzioni possono investono nelle azioni di società quotate per detenere parte della società e condividere parte del rischio, in cambio di potenziali profitti futuri.
- **Indicatori economici**:L'andamento del mercato azionario è spesso utilizzato come indicatore anticipato delle condizioni economiche, ad esempio di crescita di un settore, influenzando le decisioni di investimento e le strategie aziendali.
- **Impatto globale**: Il mercato azionario non è confinato alle frontiere nazionali; le sue fluttuazioni hanno un impatto su scala globale, influenzando le economie di diversi paesi.

La Sentimental Analysis è una delle pratiche utilizzate nella comprensione delle dinamiche dei trend di mercato, la nostra analisi nello specifico riguarda il mercato azionario, in particolare il mercato azionario statunitense ed europeo.
Nel 2022 a seguito di una correzione dei mercati dovuta all'alzamento dei tassi di interesse (sia della BCE che dalla FED), sui social media in particolare twitter (ora X), uno dei tanti hashtag utilizzati è stato "#stockmarketcrash".
Tramite questo hashtag è stato possibile prendere i Tweet ad esso associati per addestrare un modello BERT ovvero un modello di rete neurale basato sui Trasformers per il processamento del linguaggio naturale (NLP). Tale modello seppur utilizzato a fini puramente didattici potrebbe essere raffinato e utilizzato in futuro per scopi come:

- Prevedere le fluttuazioni di mercato basate sulle emozioni e le opinioni degli investitori.
- Comprendere come gli investitori percepiscono il mercato, influenzando le strategie di investimento.
- Gestione del rischio, identificando situazioni di rischio reali da situazioni di rischio percepite.

Nel nostro caso di studio tuttavia ci limiteremo ad effettuare una classificazione ternaria tra i Tweet, discriminando tra quelli che hanno un "sentimento" positivo, negativo o neutro.

2 Descrizione del Dataset

Il dataset è stato ottenuto dalla piattaforma Kaggle all'URL :

https://www.kaggle.com/datasets/tejasurya/huge-stock-market-crash-2022 .

Il dataset contiene i dati in formato .csv relativi ai Tweet del 2022 per un totale di 33946 Tweet che sono stati estratti ricercando hashtag #stockmarketcrush, i dati che sembravano di maggior rilevanza sono ovviamente quelli relativi al "text_sentiment", ma anche alla colonna "likecount", "replycount" e "retweetcount", tuttavia tali dati non sono bilanciati e per tal motivo sono stati tralasciati nella nostra analisi ad eccezione del dato relativo al "text_sentiment".

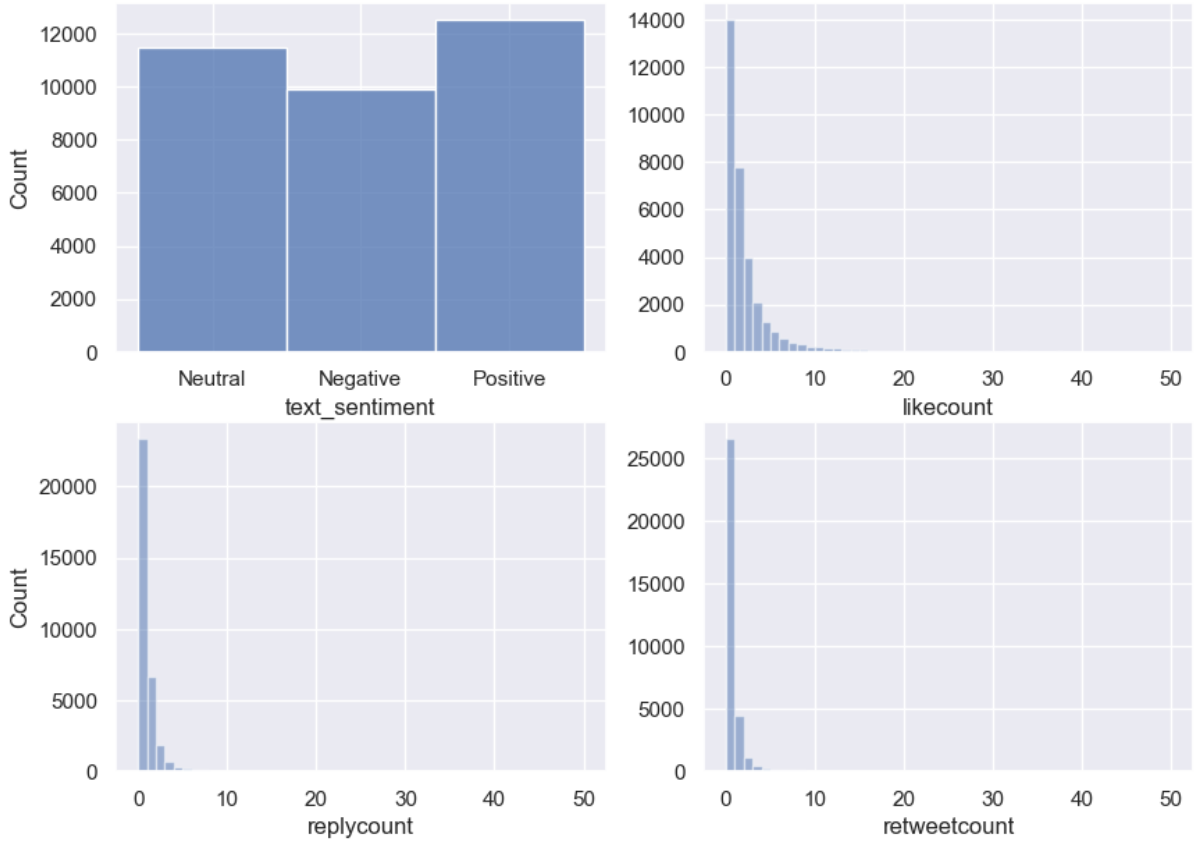


Figure 1: Distribuzione dei dati del dataset relativo alle 4 colonne sopracitate.

Di seguito riportiamo un esempio dei dati pre-processamento:

	text	text_sentiment	likecount	replycount	retweetcount
0	When will the #NYSE #stockmarketcrash happen?	Neutral	1	0	0
1	Aaj ka gyan: If a company isn't a quality c...	Negative	8	0	1
2	The stock market needs to crash hard to make i...	Negative	0	0	0

La colonna hashtags del dataframe invece ci da una ulteriore vista su quelli che possono essere gli hashtag collegati all'hashtag "stockmarketcrush", abbiamo quindi visto quali sono i 10 hashtags più utilizzati:

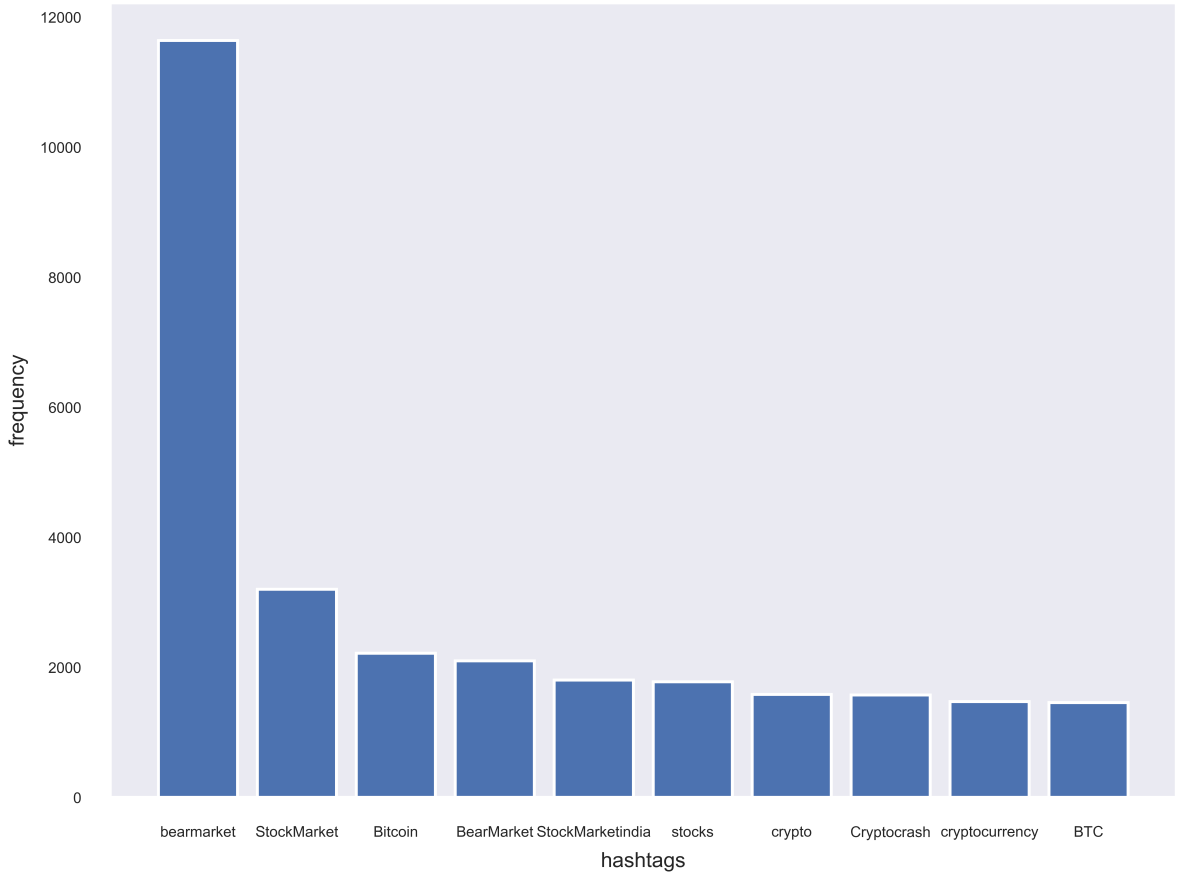


Figure 2: I 10 hashtag più frequenti ad eccezione di #stockmarketcrash.

3 ETL (Extract, Transform, Load)

Prima di passare al training, è stata eseguita una prima pulizia del testo della colonna "text", in particolare sono state eseguite le seguenti operazioni:

- Conversione in minuscolo: tramite la funzione lower() per assicurare maggior coerenza nella formattazione del testo.
- Rimozione di caratteri speciali: La riga re.sub(r"[â-zA-Z?.!,¿]+", " ", text) tutti i caratteri speciali e simboli di punteggiatura sono stati sosituiti con spazi vuoti.
- Rimozione di URL: Sono stati rimossi gli indirizzi URL presenti nel testo, sostituendole con una stringa vuota.
- Rimozione di tag HTML: sono stati rimossi eventuali tag HTML presenti nel testo e sostituiti con una stringa vuota.
- Rimozione di punteggiatura.
- Rimozione delle stopwords: ovvero connettivi, articoli, etc.
- Rimozione delle emoji

In seguito per lavorare con dati numerici, è stato necessario trasformare i labels "neutrale", "positivo", "negativo" in label numerici per cui si è utilizzata la seguente codifica applicata tramite lambda functions:

Neutrale	0
Positivo	1
Negativo	2

Inoltre è stata eseguita la Tokenizzazione attraverso la classe BertTokenizer della liberia trasformers di python. Alla fine di questa fase ad esempio per il testo: "never trade opinions prorsitip stockmarketcrash" può essere associata la tabella :

Tokenized	**Token IDs**
never	2196
trade	3119
opinions	10740
pro	4013
##rs	2869
##iti	25090
##p	2361
stock	4518
##market	20285
##cr	26775
##ash	11823

Inoltre essendo la lunghezza massima delle frasi analizzate uguale a 160, il parametro è stato "max_length" è impostato a 160 per evitare padding.
Infine i token_ids, labels e maschere di attenzione ottenuti dalla tokenizzazione e che verranno effettivamente utilizzati durante l'addestramento e il testing vengono convertiti in tensori PyTorch.

4 Training & Testing

Il nostro dataset è stato suddiviso in tre parti: da prima è stato effettuato uno split 80/20 tra set di training e set di testing. Successivamente è stato effettuato questa un ulteriore split del set di training sempre rispettando il rapporto 80/20, in modo da ottenere il set di validazione.
Questo è lo split dei dati iniziale:

Table 1: Totale del set di addestramento (validazione compreso): 27157

Etichetta di Sentimento	**Conteggio**
0	9,110
1	9,957
2	8,090

Table 2: Totale del set di Testing: 6790

Etichetta di Sentimento	**Conteggio**
0	2,388
1	2,586
2	1,816

Per migliorare le performance di accuratezza è stato eseguito un bilanciamento nel set di training, sono stati quindi tagliati una parte dei dati con label 0 e 1 (Neutrale e Positivo) in modo da ottenere 24270 righe totali (8090 per label), il dataset di testing è stato lasciato invariato.
Infine è stato inizializzato il modello della classe "BertForSequenceClassification" della liberira trasformers con i seguenti parametri:

Table 3: Parametri Principali del Modello BERT

Parametro	
Epoche	4
Algoritmo di Ottimizzazione	AdamW
Tasso di Apprendimento	2e-5
Epsilon per la Stabilità Numerica	1e-8
Dimensione del Batch	16
Numero di Labels di Output	3
Architettura del Modello	bert-base-uncased
Warm Up Steps	0

Per eseguire il training è stato prima effettuata una prova su CPU, stimato il tempo di oltre 4 ore su processore M1, si è passato al training su GPU T4 attraverso Colab che ha permesso di completare il training in 39 minuti e 2 secondi. Di seguito sono riportate le performance del training: Come si può vedere sia dai dati delle performance che dal grafico della loss, dopo la seconda epoca il modello va in overfitting, ciò nonostante già alla seconda epoca si riesce ad ottenere una buona accuratezza superiore al 85%.

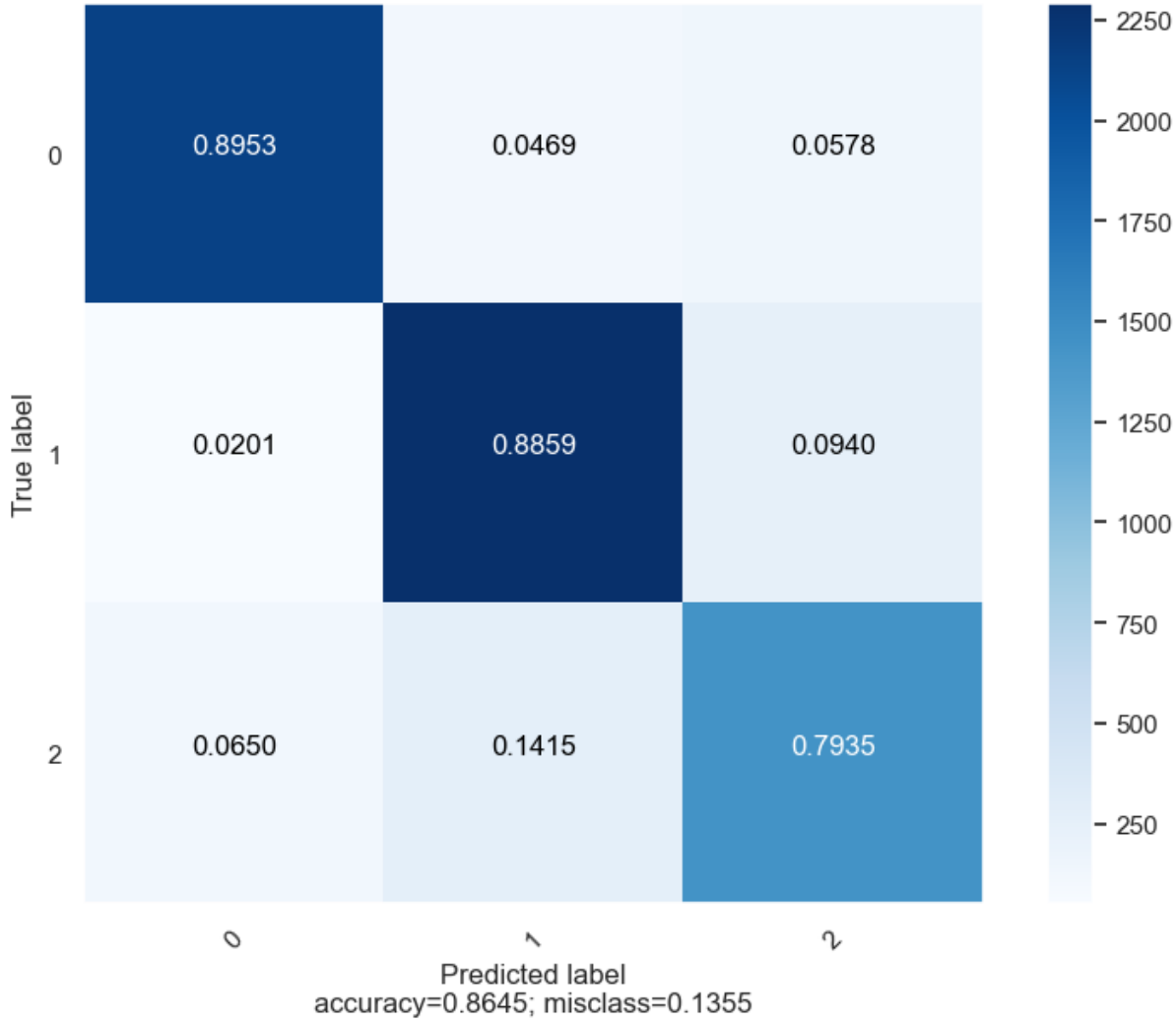
Table 4: Risultati dell'Addestramento per Epoche

Epoche	Training Loss	Valid. Loss	Valid. Accur.	Training Time	Validation Time
1	0.6221	0.4638	0.8366	0:08:14	0:00:42
2	0.3791	0.4289	0.8686	0:08:17	0:00:41
3	0.2738	0.4440	0.8680	0:08:13	0:00:41
4	0.2086	0.5174	0.8686	0:08:12	0:00:41

Figure 3: La loss durante la validation diverge rispetto alla loss del training.

Come si può vedere dalla matrice di confusione, inaspettatamente il maggiore overlap è tra le classi 1 e 2, ovvero tra le classi che predicono un sentimento positivo e negativo, mentre l'overlap con la classe "neutrale" è minimo.

Figure 4: Matrice di confusione (normalizzata).



5 Conclusioni

In questo progetto didattico, abbiamo addestrato un modello di elaborazione del linguaggio naturale basato su BERT. I risultati hanno mostrato un'accuratezza superiore all'85%, ma con evidenti segni di overfitting dopo la seconda epoca di addestramento. Le matrici di confusione hanno rivelato un'inaspettata sovrapposizione tra le classi "positivo" e "negativo". Per migliorare il modello e le analisi future, potrebbe essere apportati una serie di miglioramenti, come: Ottimizzazione del modello: effettuare un iperparametrizzazione attraverso tecniche come la grid search. Feature aggiuntive: Esplorare nuove feature e aumentare il numero di dati da elaborare. Analisi dell'overfitting: Approfondire l'analisi delle cause dell'overfitting. Test in tempo reale: Valutare il modello su dati in tempo reale per scopi di previsione, eventualmente implementare una pipeline, per estrarre, processare i dati e riallenare il modello sui dati aggiornati. Maggiori ricerche e test sono necessari per sviluppare un modello più robusto e affidabile che possa essere utilizzato all'interno di applicazioni reali.