ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

DIPARTIMENTO DI SCIENZE STATISTICHE

"PAOLO FORTUNATI"

Corso di Laurea in Scienze Statistiche

VALUTAZIONE DELL'EFFICACIA DELLA TWITTER SENTIMENT ANALYSIS COME STRUMENTO DI PREVISIONE DEL MERCATO AZIONARIO

(Utilizzo Statistico di Banche Dati Economiche Online)

Presentata da:

Davide Giardini

matricola: 924276

Relatore:

Chiar.mo Prof. Ignazio Drudi

APPELLO I

ANNO ACCADEMICO 2021/2022

INDICE

1 Introduzione

1.1 La Sentiment Analysis

1.2 Ricerche passate

2 Fase Operativa

2.1 Download

2.1.1 Scelta del Campione

2.1.2 Scrittura del codice

2.1.3 Automatizzazione del download

2.1.4 Statistiche descrittive del Dataset

2.2 Analisi

2.2.1 Pulizia e lemmatizzazione

2.2.2 Naive Bayes

2.2.3 Syuzhet

2.2.4 Udpipe

2.2.5 Costruzione dei Datasets

3 Risultati

3.1 Apple

3.2 Google

3.3 Nike

3.4 Nestlé

3.5 Beyond Meat

3.6 Bayer

3.7 NovaVax

3.8 Conclusioni

1. INTRODUZIONE

1.1 La Sentiment Analysis

L'analisi dei sentimenti (in inglese "sentiment analysis" o anche "opinion mining") è un campo dell'analisi testuale che si occupa della costruzione di sistemi in grado di estrarre opinioni, polarità, sentimenti, emozioni e valutazioni da testi. Questa sua caratteristica si dimostra particolarmente utile non tanto in singoli testi brevi, laddove l'analisi può essere affrontata con più accuratezza dall'uomo, ma piuttosto in insiemi molto grandi di testi, nei quali la macchina si rivela largamente più veloce. Per questo motivo le doti e le possibilità di questo tipo di analisi vengono risaltate quando ci si trova in territori come Internet, dove milioni di utenti scrivono miliardi di testi sugli argomenti più disparati.
La possibilità di determinare l'opinione che il pubblico detiene su un determinato argomento rende l'analisi dei sentimenti sul web una tecnica sempre più utilizzata in diversi ambiti. In politica viene impiegata per valutare il pubblico consenso di un candidato. Nei mercati azionari, ambito di riferimento di questa ricerca, viene sfruttata per conoscere l'apprezzamento della massa o degli altri investitori per una determinata azienda. Infine, campi come il marketing o la valutazione delle preferenze del consumatore se ne avvalgono, tra le altre cose, per esaminare l'alterazione dell'opinione pubblica riguardo al proprio prodotto a seguito di una campagna pubblicitaria.

L'obbiettivo di questa ricerca è quello di valutare l'efficacia della Twitter Sentiment Analysis come strumento di previsione del mercato azionario.
Twitter è un social network, creato nel 2006 in California, che si occupa per lo più di notizie e microblogging, ovvero di una pubblicazione costante di piccoli contenuti in rete sotto forma di brevi messaggi di testo. Questa sua caratteristica lo rende il social network perfetto sul quale applicare l'analisi dei sentimenti allo scopo di determinare l'opinione che il pubblico ha su determinati argomenti.
Ciò che questa tesi vuole fare, quindi, è determinare la polarità dei tweet relativi a un campione di aziende e utilizzarla per valutare la capacità previsiva che questa ha sul movimento del valore delle aziende stesse sul mercato azionario. Ciò che vogliamo capire è se l'opinione che il pubblico esprime su Twitter sul prodotto, sulla marca, e sulla quota azionaria dell'azienda, siano correlate con la serie storica della valutazione dell'azienda sul mercato azionario.

Un riscontro positivo di questa ipotesi potrebbe non solo significare una forte utilità di questo tipo di dati per gli azionisti, ma anche suggerire una possibile implementazione di questa tecnica statistica in Sistemi di Trading Automatico (STA o "trading robot").
I trading robot sono sistemi informatici che creano in automatico ordini di compravendita da inviare ad un mercato borsistico. Le regole di acquisto e vendita possono essere basate su condizioni semplici come una media mobile o su regole più complicate. Se l'analisi delle polarità dei tweet si rivelasse efficace, i trading bot se ne potrebbero avvalere per effettuare decisioni più accurate, mantenendo il grande vantaggio che già ora li contraddistingue dall'uomo a tal punto da essere gli emittenti del 75% delle negoziazioni sui mercati degli Stati Uniti: la velocità di analisi, di decisione e di azione.

1.2 Ricerche passate

Un quesito fondamentale con cui ci dobbiamo scontrare qualvolta si parli di predire il mercato azionario è se il suo andamento possa realmente essere predetto.
L'ipotesi dei Mercati Efficienti (EMH) afferma che i prezzi delle attività riflettono tutte le informazioni disponibili, e che quindi siano mossi per lo più da nuova informazione, più che dai prezzi presenti e passati. Sempre secondo questa teoria, dato che le novità sono imprevedibili, il mercato azionario seguirà un modello Random Walk, per cui non sarà possibile predirlo con accuratezza maggiore del 50 per cento.
Il dibattito che segue l'ipotesi dei mercati efficienti è, come si può immaginare, molto vivace, e sono numerose le ricerche dell'ultimo secolo che hanno fornito evidenze empiriche per sostenere sia una parte della discussione che l'altra. Lavori come quello di Walczak (2001) [1] convalidano l'impossibilità di una predizione con accuratezza maggiore del 50%. D'altra parte, altre ricerche dimostrano come il mercato azionario non segua un modello Random Walk, e che quindi esso possa essere predetto fino a un certo grado (Bollen et al., 2011) [2].
Indipendentemente dalla parte in cui ci si schieri, si potrebbe essere d'accordo che le notizie siano sì imprevedibili, ma che alcuni dei loro primi indicatori possano essere estratti dai social media. Insomma, nessuno può predire la nuova informazione, ma probabilmente ci sono metodi in grado di coglierla, e aggiustare di conseguenza le proprie decisioni, più veloci di altri. Pubblicazioni passate, ad esempio, utilizzano chat pubbliche online per predire le vendite di libri [3], o i sentimenti all'interno di blog per predire gli incassi di film [4].

Inoltre, sappiamo dalla ricerca psicologica che le emozioni, in addizione alle informazioni, giocano un ruolo importante nel processo di decisione umano. La branca della Finanza Comportamentale ("Behavioral finance"), ad esempio, si occupa dello studio dell'impatto che le emozioni e l'umore hanno sulle decisioni finanziarie. Studi come quelli di Johan Bollen, Huina Mao e Xiao-Jun Zeng [2], già citato sopra, hanno ad esempio trovato correlazione tra alcune misure dell'umore collettivo derivato dall'analisi dei testi di Twitter e l'indice Dow Jones Industrial Average (DJIA).

In conclusione, le pubblicazioni passate forniscono a questa indagine tre fondamenta. Per prima cosa, alcune di esse dimostrano una possibilità, entro una certa misura, di predizione del mercato azionario, in opposizione all'ipotesi dei Mercati Efficienti. Per secondo, ci mostrano la velocità con cui gli strumenti statistici e tecnologici che stiamo utilizzando sono in grado di cogliere le nuove informazioni. Per ultimo, provano un legame tra i sentimenti degli esseri umani e il loro processo decisionale.

2. FASE OPERATIVA

2.1 Download

2.1.1 Scelta del Campione

Ciò che distingue questa ricerca da quella di Bollen, Mao e Zeng [2] è che, mentre quest'ultima indaga una relazione tra l'umore generale e collettivo della società con l'andamento complessivo del mercato azionario (usando l'indice Dow Jones come indicatore di quest'ultimo), il nostro obbiettivo è quello di confrontare le opinioni che gli individui hanno su una marca o sul prodotto che questa marca produce con i movimenti del valore di questa stessa azienda sul mercato azionario.

La prima cosa fatta, dunque, è stata scegliere quali marche sarebbero state oggetto dello studio. Volendo essere sicuri che le conclusioni a cui saremmo arrivati non fossero applicabili solo ad un determinato settore, ma al mercato in generale, le aziende all'interno del campione sono state scelte da settori disparati:

- Informatica
    - **Apple (AAPL)**
      Multinazionale statunitense che produce sistemi operativi, smartphone e computer.
    - **Alphabet (GOOGL)**
      Tra la grande quantità di servizi offerti troviamo il motore di ricerca Google, il sistema operativo Android e servizi quali YouTube, Gmail e Google Maps.
- Abbigliamento
    - **Nike (NKE)**
      Multinazionale statunitense che produce calzature, abbigliamento e accessori sportivi.
- Agricoltura/Alimentare
    - **Nestlé (NESN)**
      Multinazionale attiva nel settore alimentare, produce e distribuisce articoli come acqua, omogeneizzati, surgelati e latticini.
    - **Beyond Meat (BYND)**
      Nata nel 2009, produce sostituti per la carne e prodotti caseari a base di vegetali.

- Farmaceutica
    - **Bayer (BAYN)**

      Multinazionale farmaceutica con sede a Leverkusen, in Germania.
    - **Novavax (NVAX)**

      Società di biotecnologie statunitense che si occupa di scoperta, sviluppo e commercializzazione di vaccini e adiuvanti per gravi malattie infettive.

A questo punto costruiamo sette stringhe di ricerca, una per marca, che saranno poi utilizzate per fornire all'API di Twitter i termini da cercare fra i post. All'interno di ogni stringa inseriamo il nome della marca, ancora il nome, questa volta preceduto da un hashtag (#), e il codice dell'azienda. In questo modo non solo ci saranno restituiti tutti i tweets che contengono il nome dell'azienda, ma anche quelli che hanno trattano dell'azienda (cioè che contengono il suo hashtag) e quelli che parlano delle quote dell'azienda sul mercato azionario (cioè che contengono il codice).
Per marche come Apple, inoltre, sono stati inseriti anche i nomi di alcuni dei suoi prodotti. In questo modo abbiamo potuto immagazzinare anche tweet che recensiscano le sue merci, o che valutino lanci di nuove linee di telefoni, computer o tablet.

Dopo un primo download di prova, abbiamo controllato i tweet scaricati per cercare eventuali risultati non consoni. Ci siamo quindi accorti di alcuni problemi interni alle stringhe di ricerca che le portavano a recuperare tweet che trattavano di argomenti non correlati con una valutazione o una notizia dell'azienda, o addirittura non legati in nessun modo alla marca:

- AAPL
    - ***"apple.news"***

      Apple offre un servizio di giornalistica ai possessori dei propri smartphone e computer. Tali notizie possono poi essere condivise su Twitter. Ci sono molti tweet, quindi, che parlano della notizia in questione, e che contengono al loro interno il link all'articolo. Dato che il link a "Apple.news" contiene la stringa "Apple" tutti i tweet inerenti alle notizie più disparate venivano scaricati solo per il fatto che queste erano state condivise tramite Apple-News.
    - ***"apple.com"***

      In modo simile a quanto avveniva con Apple-News, molti tweet recensivano podcast, canzoni e applicazioni presenti all'interno dei servizi che Apple

fornisce ai propri utenti: “podcast.apple.com”, “music.apple.com”, “itunes.apple.com”.

- BAYN
  - ***“Peter Bayer”*** è il segretario della FIA (“Federazione Internazionale dell’Automobile”). Tutti gli articoli a lui legati venivano scaricati nonostante non abbia nessun legame con l’azienda Bayer.
  - ***“Bayer Leverkunsen”*** è una squadra di calcio tedesca.
  - ***“Tor_BaYeR”*** è un account di spam molto attivo non legato all’azienda.
- GOOGL
  - ***“google.com”***

    Similmente a quanto successo con Apple, togliendo dalle stringhe di ricerca “google.com” evitiamo di scaricare tweet che contengono link a: questionari creati tramite google (“docs.google.com”), posizioni condivise tramite l’applicazione di mappe (“maps.google.com”), documenti e immagini caricate sul cloud di google (“drive.google.com”), applicazioni scaricati tramite il servizio che google fornisce ad Android (“play.google.com”)
  - ***“search”* e *“googled”***

    Dato che l’azienda fornisce anche il famosissimo motore di ricerca, erano numerosi i tweet di persone che si riferissero a articoli trovati su google o ricerche effettuate tramite il famosissimo sito.
- NESN
  - ***“NESN”*** si riferisce anche a “New England Sports Network”. Per fortuna Twitter offre l’opzione di categorizzare i tweet che parlano di finanza utilizzando il simbolo del dollaro ($) come se fosse un hashtag, prima del codice dell’azienda. Essendo questo un metodo comodo e sicuro per trovare solo i risultati desiderati, apportiamo l’aggiunta del dollaro prima del codice a tutte le marche.
- NKE
  - Essendo Nike una famosissima azienda di moda sono numerosissimi i così detti “reseller” (rivenditori), cioè persone che per lavoro comprano da Nike a prezzi vantaggiosi e poi rivendono su internet. Per evitare di scaricare tweet di questo tipo eliminiamo dalla ricerca qualsiasi termine che possa essere collegato all’atto

della vendita: ***"giveaway"***, ***"sale"***, ***"#AD"***, ***"NBA"***, ***"NFL"***, ***"buy here"***, ***"now available"***.

Dopo queste modifiche, le stringhe di ricerca si presentano in questo modo:

Script 1: Stringhe di ricerca e vettore "marche"

```
str_AAPL <- "Apple OR $AAPL OR #Apple OR iPhone OR OR iMac OR iPad -apple.com -apple.news"
str_BAYN <- "Bayer OR $BAYN OR #Bayer -Peter -FIA -ToR"
str_BYND <- "Beyond Meat OR $BYND OR #BeyondMeat"
str_GOOGL <- "Google OR $GOOGL OR #Google -google.com -search -googled"
str_NESN <- "Nestlé OR Nestle OR $NESN OR #Nestle OR #Nestlé"
str_NKE <- "Nike OR $NKE OR #Nike -giveaway -#AD -NBA -NFL -Sale -here"
str_NVAX <- "Novavax OR $NVAX OR #Novavax "

marche <- c("AAPL", "GOOGL", "NKE", "NESN", "NVAX", "BAYN", "BYND")
```

2.1.2 Scrittura del codice

Per costruire il dataset da utilizzare come base per la ricerca abbiamo fatto affidamento all'API di Twitter. Con API, ovvero l' "Interfaccia di Programmazione di una Applicazione", si indica un insieme di procedure atte a risolvere uno specifico problema di comunicazione tra diversi computer o software. Le API, quindi, permettono a prodotti o servizi di comunicare con altri prodotti o servizi: nel nostro caso ci permettono di comunicare con il servizio di Twitter per il download dei dati a cui siamo interessati.
L'API di Twitter[5] ha tre particolarità cui dobbiamo tenere conto nella fase di download. La prima è che permette il download di 18 mila tweets ogni quarto d'ora. La seconda è che è impossibile specificare l'arco di tempo da cui scaricare i tweet: verrà effettuato i download dei più recenti fino a che non si raggiungono i 18 mila. La terza regola è che, anche se questo limite non viene superato, non possono essere scaricati tweet più vecchi di una settimana.
Tutto ciò fa sì che, se vogliamo costruirci una serie storica della polarità dei tweet, dobbiamo impegnarci noi stessi a scaricarli giorno per giorno.

Qui di seguito, dunque, il codice utilizzato su R ogni giorno per scaricare i tweet di tutte le marche:

Script 2: loop di ricerca

```
1       for (i in marche) {
2         # Ricostruisce il nome del file salvato ieri e il percorso in cui è stato salvato
3         name_t <- paste(i, Sys.Date()-1, "csv", sep=".")
4         address_t <- paste("./DataFrames/Tweets", i, name_t, sep= "/")
5         # Apre il file di ieri e legge l'id dell'ultimo tweet scaricato
6         tab <- read.csv(address_t)
7         id <- sub('.','',tab[1,2])
8
9         # Costruisce il nome del vettore contenente le parole da cercare
10        terms <- paste("str", i, sep="_")
11        # Prende l'oggetto con il nome appena creato e lo duplica all'interno di "terms"
12        terms <- eval(parse(text=terms))
13
14        # Ricerca
15        tweets <- search_tweets(terms, n=6000, lang="en", include_rts = F, since_id = id)
16
17        # Costruisce il nome del file da salvare e il percorso in cui salvarlo
18        name_t <- paste(i, Sys.Date(), "csv", sep=".")
19        address_t <- paste("./DataFrames/Tweets", i, name_t, sep= "/")
20        # Scrittura del file nella cartella desiderata
21        write_as_csv(tweets, address_t)
22
23        # Se la marca è Nike aspetta 15 minuti
24        if(i=="NKE"){
25          Sys.sleep(60*15)
26        }
27      }
```

Il comando "for" (riga 1) ci permette di costruire un loop. Ciò significa che tutto il codice all'interno delle parentesi graffe che lo segue verrà ripetuto 7 volte, una per marca, e in ognuna di queste l'oggetto "i" assumerà uno dei valori del vettore "marche", descritto nello *script 1*. Alla riga 10 costruiamo un oggetto che contenga una stringa di testo identica al nome dell'oggetto in cui abbiamo precedentemente salvato la stringa di ricerca. Successivamente, alla riga 12, utilizziamo le funzioni "eval()" e "parse()" per trasformare lo stesso oggetto nella stringa di ricerca il cui nome era uguale alla stringa all'interno dell'oggetto. Ora, quindi, l'oggetto "terms" contiene al suo interno la stringa di ricerca della marca: di Apple se siamo alla prima iterazione del loop, di Google se siamo alla seconda, e così via. Nella riga 15 avviene il vero e proprio download dei tweet, grazie alla funzione "search_tweets()" della librearia "rtweet"[6]. Al suo interno specifichiamo la stringa di ricerca ("terms"), il numero di tweet da scaricare (6000), la lingua dei tweet da scaricare (inglese) e infine se scaricare anche i retweet (no).

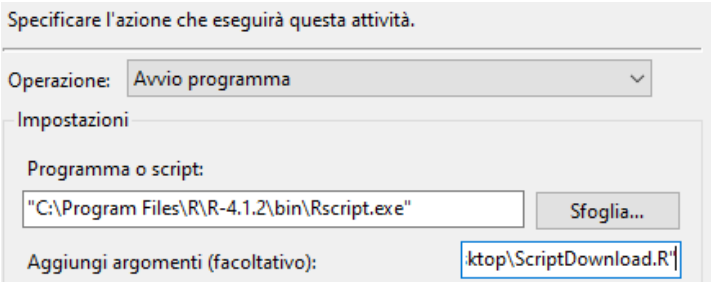
Per ultimo, costruiamo il nome con cui deve essere salvato il dataset scaricato oggi (riga 18). Lo facciamo affiancando al nome della marca la data del giorno, separati da un punto. Costruiamo il percorso dove vogliamo salvare il dataset, in modo da archiviare i dati delle diverse aziende ognuno nella propria cartella (riga 19). E infine utilizziamo la funzione "write_as_csv()" per salvare i dati all'indirizzo specificato, con il nome specificato, in formato csv (riga 21).

Inoltre, dato che Apple, Google e Nike, essendo le più famose, raggiungono sempre i 6 mila tweet giornalieri, è necessario aspettare 15 minuti prima di passare all'iterazione del loop successiva. Questo perché, come abbiamo detto, possiamo scaricare solo 18 mila tweet ogni 15 minuti. Dato che tutte le altre aziende hanno un traffico di dati notevolmente minore non è necessario ripetere questa procedura per altre iterazioni.
Un codice di questo tipo, però, porterebbe ad una sovrapposizione di dati. Rischieremmo cioè di scaricare, oltre ai dati del giorno stesso, quelli che abbiamo scaricato ieri, ieri l'altro, e così via finché non si raggiunge il limite dei 6000 tweet che abbiamo impostato oppure quello della settimana imposto da Twitter. Non essendoci alcun modo per impostare un limite a livello di data e ora all'interno della funzione search_tweet oltre al quale smettere di scaricare i dati, l'unico modo per evitare la sovrapposizione è quello di ricostruire il nome del dataset del giorno prima, utilizzando la data di ieri (righe 3 e 4), leggere il csv (riga 6), e salvare all'interno dell'oggetto "id" l'id dell'ultimo tweet che è stato scaricato (riga 7). In questo modo possiamo specificare all'interno della funzione search_tweet l'id del tweet al quale terminare la propria ricerca (since_id = id).

2.1.3 Automatizzazione del Download

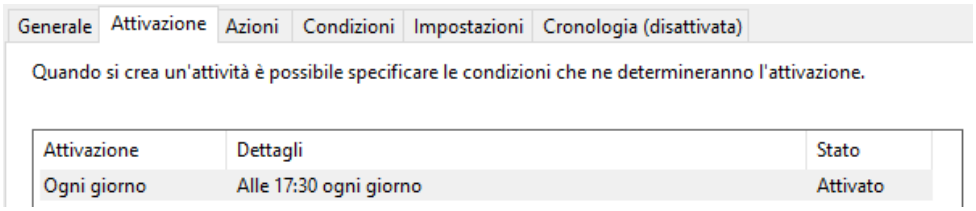
Una volta scritto il codice che permettesse di scaricare i dati necessari alla ricerca siamo passati alla fase di automatizzazione dello script, in modo che il download dei dati avvenisse automaticamente ogni giorno alla stessa ora.
Per fare ciò è stato utilizzato "Windows Task Scheduler", in italiano "utilità di pianificazione di Windows". L'utilità di pianificazione è una componente integrata nel sistema operativo di Microsoft Windows che fornisce la possibilità di pianificare il lancio dei programmi o di script in periodi predefiniti o dopo intervalli di tempo specificati.
È stata creata quindi un'attività che attivasse "Rscript.exe". In "aggiungi argomenti" si è specificato il percorso e il nome del file che Rscript.exe dovrà eseguire (il loop di ricerca descritto in *Script 2)*(*Immagine 1*).

Immagine 1: azione eseguita dall'attività di Windows Task Scheduler



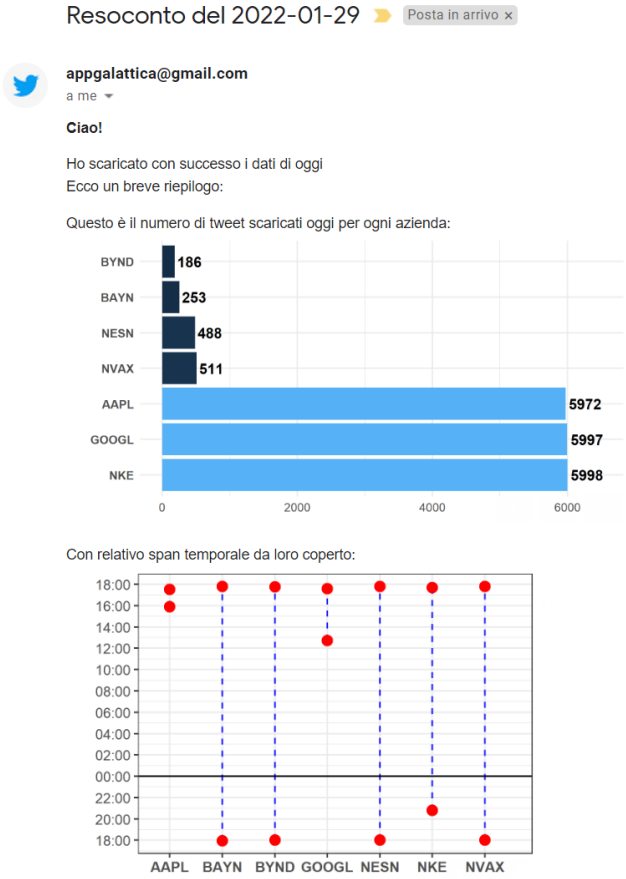
Come rivedremo poi più in dettaglio, aziende come Google ed Apple, essendo più famose, sono oggetto di molti più di 6 mila tweet al giorno. Ciò significa che il nostro script può scaricare solo i tweet di un determinato periodo di tempo: spesso poco più di un paio d'ore. Per questo motivo, dobbiamo effettuare una scelta ragionata sull'esatto momento della giornata nel quale avviare l'attività del Task Scheduler. Ci è sembrato più intelligente avviare il programma alle 17.30 di ogni giorno. Così facendo, i tweet di Google e Apple coprono uno span che va all'incirca dalle 15.30 alle 17.30, cioè le prime ore di apertura della borsa americana (9:30 EST, cioè 15:30 CET), pensando che queste potessero essere le più significative per la previsione del valore al tempo di chiusura (*Immagine 2)*.

Immagine 2: Condizioni di attivazione dell'attività di Windows Task Scheduler



A questo punto l'incarico del download giornaliero dei tweet è stato affidato ad un secondo computer. Il fatto che lo script di download fosse salvato su OneDrive (il servizio cloud di Microsoft) ha fatto sì che ogni tipo di modifica al codice potesse essere fatta da remoto. Allo stesso modo, dato che il percorso sul quale lo script salvava i dati scaricati ogni giorno era su OneDrive, i dataset erano disponibili in qualsiasi momento da qualsiasi computer. Inoltre, per rendere il tutto più automatizzato, è stata anche implementata nello script una mail di notifica che rassicurasse il diretto interessato dell'avvenuto download e fornisse alcune statistiche descrittive per assicurarsi che tutto stesse procedendo correttamente (*Immagine 3)*.

Immagine 3: esempio di mail di notifica del 29 gennaio



2.1.4 Statistiche Descrittive del Dataset

Il dataset generato comprende 77 giorni di dati scaricati: dal 21 Gennaio 2022 al 12 Aprile 2022.

In totale sono stati scaricati 1,62 GB di dati, per un complesso di 1.644.237 tweet. Nei grafici seguenti (*Immagine 4* e *5*) sono rappresentati rispettivamente il numero di tweet scaricati per marca e il peso dei dati scaricati per marca:

Immagine 4: Numero di Tweet scaricati per marca

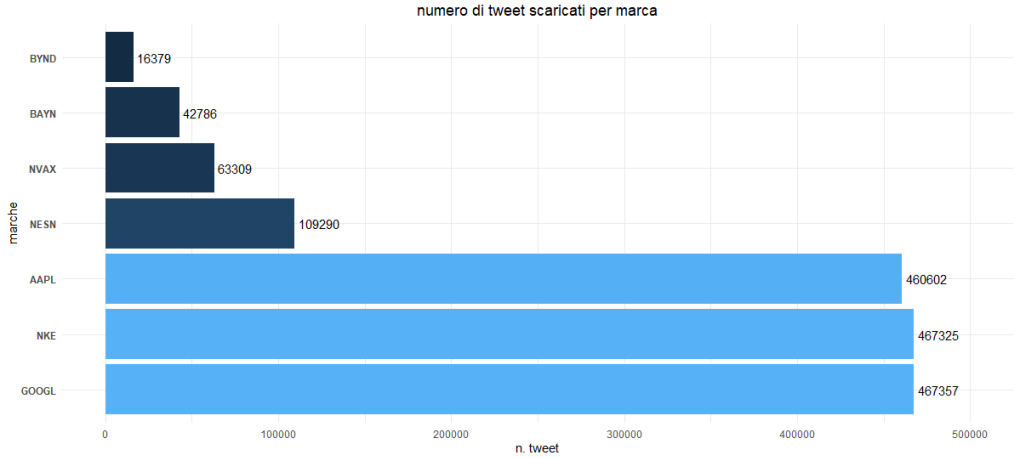
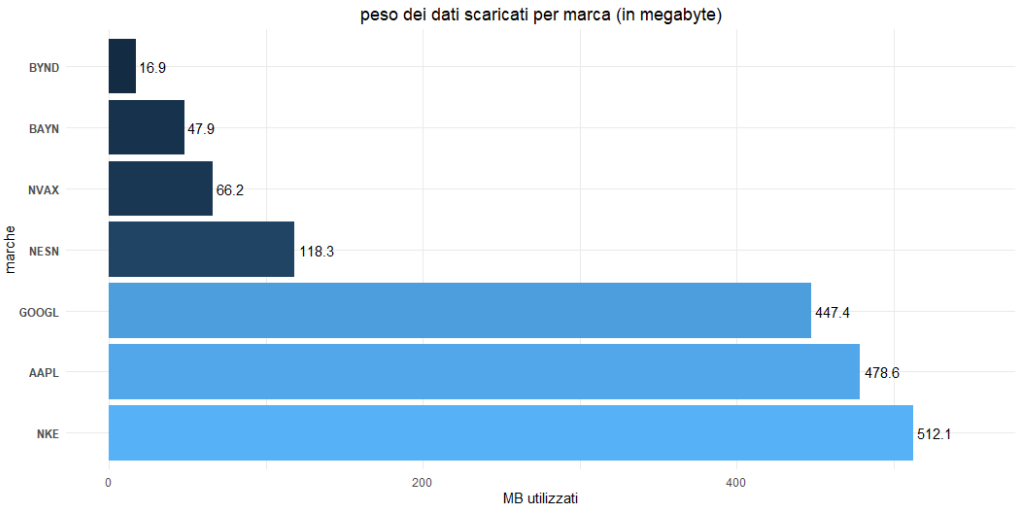
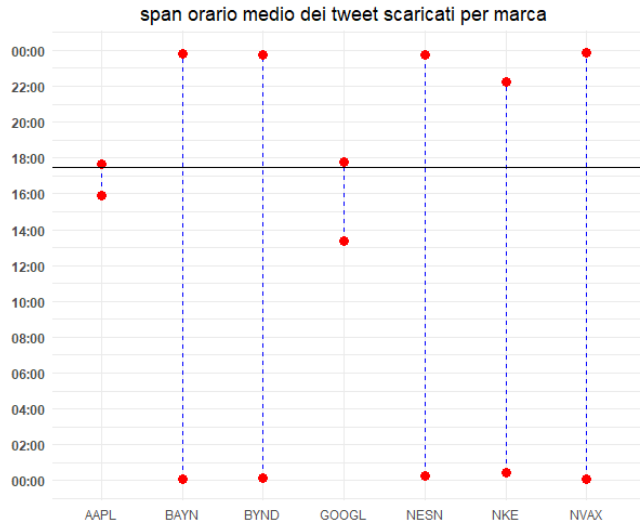


Immagine 5: Peso dei dati scaricati per marca (in megabyte)



Infine, rappresentiamo in un ultimo grafico lo span orario medio coperto giornalmente dai tweet scaricati (*Immagine 6*):

Immagine 6: Span orario medio dei tweet scaricati per marca



Grazie all'informazione mostrata da questi tre grafici è possibile suddividere le sette marche in tre gruppi:

1. **Marche a traffico elevato**

   Apple e Google sono marche che giornalmente sono menzionate in più di 6 mila tweet. Ciò fa sì che tutti i tweet che abbiamo scaricato in questo periodo di 77 giorni provengano da uno span orario di circa due/quattro ore (tra le 17:30 e le 15:30/13:30). È importante specificare per queste marche, quindi, che il download dei tweet non è avvenuto con campionamento casuale all'interno di tutto l'arco giornaliero, ma all'interno di un paio d'ore: le prime ore di attività del mercato azionario statunitense.

2. **Marche a medio traffico**

   All'interno del gruppo a medio traffico rientra solo Nike. L'azienda in questione è menzionata giornalmente circa 6 mila volte. In questo modo, ad ogni download vengono scaricati tutti i tweet delle scorse 24 ore.

3. **Marche a basso traffico**

   Le restanti aziende (Bayer, Novavax, Beyond Meat e Nestlé) vengono menzionate meno di 6 mila volte. Ciò ci assicura di aver scaricato tutti i tweet inglesi che le menzionassero negli ultimi 77 giorni.

   Queste marche sono ben al di sotto della soglia di 6 mila impostata nel nostro script, con aziende come Beyond Meat che contano addirittura una media di 228 tweet giornalieri.

2.2 Analisi

2.2.1 Pulizia e lemmatizzazione

Prima di passare alla vera e propria Sentiment Analysis bisogna fare delle opportune modifiche e semplificazioni al testo, in modo che possa essere poi correttamente analizzato. Dopo aver unito tutti i dati dei diversi giorni in un unico documento per marca, codifichiamo il testo in UTF-8 e applichiamo la funzione "CleanTesto".
Quest'ultima funzione, scritta dal professore Fabrizio Alboni dell'Università di Bologna nell'ambito del corso di "Utilizzo Statistico di Banche Dati Economiche Online", compie la "pulizia" del testo. In particolare, è stata da noi utilizzata per:

- Rimuovere i link
- Rimuovere la punteggiatura
- Rimuovere i caratteri di controllo (ad esempio "\n")
- Rimuovere i caratteri che non sono grafici
- Rimuovere le menzioni
- Rimuovere i numeri
- Rimuovere tabulazioni e spazi nel testo
- Trasformare l'intero testo in minuscolo

Dopo la pulizia del testo si passa alla fase di lemmatizzazione. Questo processo consiste nella riduzione di una forma flessa di una parola (ad esempio "corriamo") alla sua forma canonica, detta "lemma" (ad esempio "correre").
Il lemma è la coppia di informazioni (vocabolo, categoria grammaticale) con cui una parola è presente nel dizionario della lingua. In realtà, lemmatizzare un testo significa anche effettuare:

- La *tokenizzazione*, ovvero l'individuazione delle unità minime morfologiche
- Il *tagging grammaticale*, ovvero l'identificazione della categoria grammaticale delle parole

La lemmatizzazione è stata ottenuta utilizzando la funzione "udpipe_annotate", della libreria udpipe[7], assieme al treebank "english ewt ud 2.5". La lemmatizzazione con udpipe è racchiusa all'interno della funzione "lemmaUDP", sempre del professor Fabrizio Alboni, assieme all'individuazione delle così dette "stopwords". Le stopwords sono delle parole accessorie

rispetto alle principali parole del vocabolario, come: articoli, congiunzioni e preposizioni. Come elenco di stopwords è stato utilizzato:

- Un elenco specifico per ogni marca (stw_AAPL, stw_GOOGL, ecc.) che contenesse il nome e il codice della marca stessa. Questo perché, essendo questi termini all'interno delle stringhe di ricerca, ci aspettiamo che siano all'interno di tutti i tweet, e che quindi non costituiscano alcuna particolarità.
- Un elenco generale delle stopwords più comuni in inglese, ottenuto tramite la libreria tm[8] e opportunamente modificato in modo da mantenere tutti quei termini (come negazioni, intensificatori e diminutori) che saranno poi necessari per l'analisi dei sentimenti: "*non* bello" è sicuramente diverso da "bello", così come "*molto* bello" o "*abbastanza* bello".

La lemmatizzazione è un processo relativamente lungo: il computer ha impiegato almeno due ore per ciascuna delle aziende più grosse come Apple, Google e Nike.

2.2.2 Naive Bayes

I classificatori Naive Bayes sono una famiglia di classificatori probabilistici basati sull'applicazione del teorema di Bayes. Il teorema di Bayes viene impiegato per calcolare la probabilità di una causa che ha provocato l'evento verificato, tramite la formula:

$$P(A_i|E) = \frac{P(E|A_i)P(A_i)}{P(E)}$$

La classificazione viene fatta utilizzando il "MPQA Subjectivity Lexicon" di Janyce Wiebe[9], cioè un dizionario inglese al cui interno ogni termine è associato alla propria polarità e grado di soggettività.

Lo script per l'analisi dei sentimenti delle marche si presenta quindi in questo modo (*Script 3*):

Script 3: Analisi dei sentimenti con metodo Naive Bayes

```
1       for (i in marche) {
2         # Apriamo i documenti lemmatizzati
3         name <- paste("lemm", i, sep = "_")
4         name <- paste(name, "RData", sep = ".")
5         addr <- paste("./Dataframes/1. Lemmatizzazione", name, sep = "/")
6         load(addr)
7         # Calcolo della Sentiment
8         polG = myClassPolarity(textColumns = tw$txtLL,
9                                algorithm = "bayes",
10                               lexicon = "./subjectivity_en.csv")
11        # Aggiungo una colonna con il giorno
12        polG$documenti$day <- as.Date(tw$created_at)
13        # Aggiungo una colonna con i testi
14        polG$documenti$text <- tw$text
15        # Creazione della colonna "best_fit_num"
16        polG$documenti %>% add_column(best_fit_num = NA)
17        polG$documenti$best_fit_num[polG$documenti$best_fit == "positive"] <- 1
18        polG$documenti$best_fit_num[polG$documenti$best_fit == "neutral"] <- 0
19        polG$documenti$best_fit_num[polG$documenti$best_fit == "negative"] <- -1
20        # Costruzione della tabella Sentiment
21        tabpolG <- polG$documenti %>%
22          group_by(day) %>%
23          summarise(n.tweet = n(), score_best_fit=sum(best_fit_num))
24        # Salvataggio
25        name_s <- paste("sent", i, sep = "_")
26        name_s <- paste(name_s, "RData", sep = ".")
27        addr_s <- paste("./Dataframes/2. Sentiment", name_s, sep = "/")
28        save(polG, tabpolG, file = addr_s)
29      }
```

Alla riga 1 utilizziamo il comando for per costruire un loop che ripeta i comandi all'interno delle parentesi graffe per tutte le marche sotto analisi. Dalla riga 3 alla 6 costruiamo il percorso in cui sono salvati i testi puliti e lemmatizzati e lo utilizziamo per importarli grazie alla funzione load().

Dalla riga 8 alla 10 utilizziamo la funzione myClassPolarity, scritta dal Professor Alboni, per applicare il metodo di classificazione Naive Bayes.

Alla riga 16 aggiungiamo una nuova colonna al dataframe appena creato, i cui valori saranno: "1" se il tweet è stato classificato come positivo, "0" se neutro e "-1" se negativo. In questo modo possiamo, alle righe 21-23, costruire una tabella ("tabpolG") che abbia come colonna la differenza tra il numero di tweet positivi e quelli negativi per ogni giorno d'analisi. Infine, dalla riga 25 alla 28, costruiamo il nome e il percorso del file in cui salvare i dati, e procediamo all'archiviazione.

È importante sottolineare che questo è di gran lunga la metodologia meno efficiente a livello di tempistiche e di utilizzo delle risorse dell'elaboratore. I tempi di attesa per l'analisi dei sentimenti sono infatti arrivati a toccare le 14 ore per le marche più pesanti come Google, Apple e Nike.

2.2.3 Syuzhet

L'utilizzo della libreria Syuzhet[10] è probabilmente il più semplice dei tre metodi utilizzati per l'analisi dei sentimenti dei dati raccolti. Tutto ciò che bisogno fare, infatti, è importare la libreria "syuzhet" e applicare ai dati la funzione "get_sentiment()". Il dizionario di default, chiamato a sua volta "syuzhet" e impiegato da questa ricerca, è stato sviluppato dal Nebraska Literary Lab estraendo 165 mila frasi da romanzi contemporanei.

L'output restituito dalla funzione in questione differisce da quello del metodo precedente in quanto, invece di assegnare il tweet alla classe "positiva", "negativa" o "neutra", si limita a dare un punteggio basato sulla polarità delle parole da cui è composta la frase. Al contrario del metodo Naive Bayes, quindi, la libreria Syuzhet restituisce un output che non solo distingue tra tweet positivi e negativi, ma permette anche di determinare, ad esempio, quale fra due tweet positivi è più positivo.

Similmente a quanto fatto nel metodo precedente, costruiamo, partendo dall'output fornito, una tabella che abbia, per ogni riga, la somma dei punteggi di tutti i tweet scritti in quel particolare giorno.

2.2.4 Udpipe

L'ultimo metodo con cui l'analisi dei sentimenti è stata affrontata è quello dell'impiego della libreria "udpipe"[7]. La particolarità di questa libreria è che tiene conto sia delle negazioni che degli intensificatori (positivi e negativi). Così facendo è in grado di distinguere, ad esempio, tra: "bene", "non bene", "molto bene", "abbastanza bene".
La prima cosa che è stata fatta, dunque, è stata costruire i vettori di negazioni e intensificatori (Script 4):

Script 4: polarityShifter, intensifier eweakener utilizzati per metodo udpipe

```
> polarityShifter_EN
[1] "isn't"    "aren't"   "wasn't"   "weren't"  "hasn't"   "haven't"  "hadn't"
[8] "doesn't"  "don't"    "didn't"   "won't"    "wouldn't" "couldn't" "mustn't"
[15] "against"  "no"       "nor"      "not"

> intensifier_EN
[1] "further"       "all"           "more"          "most"          "too"
[6] "very"          "much"          "many"          "far"           "truly"
[11] "really"        "extremely"     "highly"        "specially"     "particulary"
[16] "especially"    "mostly"        "great"         "large"         "big"
[21] "loud"          "strong"        "high"          "exceptional"   "awesome"
[26] "outstanding"   "incredible"    "incredibly"    "unbelieavable" "remarkable"
[31] "remarkably"    "considerable"  "considerably"

> weakener_EN
[1] "few"       "only"      "just"      "minimal"   "minimally" "little"    "short"
[8] "shortly"   "fewer"     "partial"   "partially" "least"     "almost"    "nearly"
[15] "about"
```

Come dizionario di polarità è stato utilizzato "subjectivity_en", cioè lo stesso utilizzato per il metodo "Naive Bayes". Ciò significa che tutte le differenze tra il primo e terzo metodo sono da attribuire solamente alle diverse modalità di calcolo dei punteggi e alla capacità di udpipe di considerare anche quelle parole che possono modificare il significato della frase.
Per il calcolo della polarità è stata usata la funzione, sempre della libreria udpipe, "txt_sentiment()", per cui lo script utilizzato si presenta in questo modo (*Script 5*):

Script 5: sentiment analysis con metodo udpipe

```
for (i in marche) {
# Apriamo i documenti lemmatizzati
  addr <- paste("lemm", i, sep = "_")
  addr <- paste(addr, "RData", sep = ".")
  addr <- paste("./Dataframes/1. Lemmatizzazione", addr, sep = "/")
  load(addr)

  # calcolo sentiment con txt_sentiment
  sentUOp <- txt_sentiment(lemmCV,term = "lemma",
                       polarity_terms = subjectivity_en,
                       polarity_negators = polarityShifter_EN,
                       polarity_amplifiers = intensifier_EN,
                       polarity_deamplifiers = weakener_EN)

  # Costruzione polG
  polG <- as.data.frame(sentUOp$overall$sentiment_polarity)
  colnames(polG) <- "sentiment"
  polG$text <- tw$text
  polG$created_at <- tw$created_at

  # Costruzione tabpolG
  tabpolG <- polG %>%
    group_by(as.Date(created_at)) %>%
    summarise(n.tweet = n(), score=sum(sentiment))

  # Salvataggio
  addr_s <- paste("sent", i, sep = "_")
  addr_s <- paste(addr_s, "RData", sep = ".")
  addr_s <- paste("./Dataframes/8. Sentiment UdPipe", addr_s, sep = "/")
  save(polG, tabpolG, file = addr_s)
}
```

2.2.5 Costruzione dei datasets

Finita la fase di sentiment analysis, alcuni piccoli aggiustamenti erano necessari prima di passare all'analisi delle serie storiche così create.

Prima di tutto, abbiamo messo tutti i dati ricavati dalle diverse sentiment analysis nella stessa tabella tabpolG, al cui interno quindi, per ogni giorno, è riportato il punteggio ricavato con ognuno dei tre metodi.

Un'altra aggiunta che è stata portata è quella di selezionare solo i tweet che riportassero solo il nome della marca e il suo codice nel mercato azionario, oppure solamente il codice. In questo modo, quindi, per ognuno dei tre metodi, è stata aggiunta alla tabella di ogni marca anche il punteggio calcolato escludendo i tweet che parlassero dei prodotti (nella colonna denominata "name"), e quello calcolato utilizzando solo i tweet che contenessero il codice della marca (nella

colonna denominata “stock”). Così facendo saremo in grado di capire quali tweet sono più utili per la previsione:

1 Quelli che danno valutazioni dei prodotti della marca, che parlano della marca in generale e del suo andamento in borsa.
2 Solo quelli che parlano della marca o del suo andamento in borsa.
3 Esclusivamente quelli che fanno riferimento all’andamento della marca nel mercato azionario.

Per quanto riguarda i dataset con i valori azionari, sono stati scaricati dal sito “finance.yahoo.com”. L’intervallo temporale va, come per i tweet scaricati, dal 21 gennaio al 12 Aprile, con frequenza giornaliera. Per tutte le marche c’è un dato mancante il 21 febbraio: festività statunitense nata per commemorare il compleanno di George Washington che comporta la chiusura della borsa americana. Il dato mancante è stato corretto utilizzando una media tra il valore del giorno precedente e quello del giorno successivo.

Abbiamo poi provveduto a eliminare tutti i punteggi di polarità calcolati nel fine settimana, in quanto la borsa è chiusa il sabato e la domenica. I 7 dataset risultati sono quindi composti da 58 osservazioni (giorni) di 15 variabili. Le variabili sono:

1 ***Date***: la data
2 ***Open***: prezzo dell’azione all’apertura della borsa americana
3 ***Close***: prezzo dell’azione alla chiusura della borsa americana.
Variabile dipendente.
4 ***NB.bestfit***: differenza tra i tweet classificati come positivi e quelli classificati come negativi dal metodo Naive Bayes nel giorno x.
5 ***NB.perc***: rapporto tra *NB.bestfit* e il numero di tweet scaricati nel giorno x dell’azienda.
6 ***NB.name***: differenza tra i tweet classificati come positivi e quelli classificati come negativi dal metodo Naive Bayes nel giorno x, computato escludendo i tweet che facessero riferimento ai prodotti della marca. (disponibile solo per Apple)
7 ***NB.stock***: differenza tra i tweet classificati come positivi e quelli classificati come negativi dal metodo Naive Bayes nel giorno x, computato utilizzando solamente i tweet contenenti il codice dell’azienda.
8 ***SY.score***: punteggio calcolato sommando tutti i punteggi dei tweet scaricati nel giorno x e ricavati tramite l’impiego del dizionario Syuzhet.
9 ***SY.perc***: rapporto tra *SY.score* e il numero di tweet scaricati nel giorno x dell’azienda.

10 ***SY.name***: punteggio calcolato sommando tutti i punteggi dei tweet scaricati nel giorno x (escludendo quelli che facessero riferimento ai prodotti della marca) e ricavati tramite l'impiego del dizionario Syuzhet. (disponibile solo per Apple)

11 ***SY.stock***: punteggio calcolato sommando tutti i punteggi dei tweet scaricati nel giorno x e ricavati tramite l'impiego del dizionario Syuzhet, computato utilizzando solamente i tweet contenenti il codice dell'azienda.

12 ***UD.score***: punteggio calcolato sommando tutti i punteggi dei tweet scaricati nel giorno x e ricavati tramite il metodo udpipe.

13 ***UD.perc***: rapporto tra *UD.score* e il numero di tweet scaricati nel giorno x dell'azienda.

14 ***UD.name***: punteggio calcolato sommando tutti i punteggi dei tweet scaricati nel giorno x (escludendo quelli che facessero riferimento ai prodotti della marca) e ricavati tramite il metodo udpipe. (disponibile solo per Apple)

15 ***UD.stock***: punteggio calcolato sommando tutti i punteggi dei tweet scaricati nel giorno x e ricavati tramite il metodo udpipe, computato utilizzando solamente i tweet contenenti il codice dell'azienda.

3. RISULTATI

Per ogni marca del campione ci siamo, per prima cosa, concentrati sulla comparazione dei vari metodi da noi utilizzati per l'analisi dei sentimenti, costruendo grafici temporali nel quale fosse possibile paragonare i loro movimenti nel tempo. Abbiamo fatto questa analisi per tutti i dataset utilizzati:

- "**score**", contenente tutti i tweet scaricati dal 21 Gennaio al 12 Aprile 2022
- "**name**", contenente tutti i tweet scaricati, tranne quelli che facessero riferimenti ai prodotti della marca, mantenendo quindi solo quelli contenenti il nome dell'azienda e il suo codice. Questo dataset è presente solo per Apple.
- "**stock**", contenente solo i tweet contenenti il codice dell'azienda ($AAPL, $GOOGL, $NKE, $NVAX). Questo dataset non è presente per Bayer, Beyond Meat e Nestlé poiché possedevano vari giorni in cui nessun tweet conteneva il codice dell'azienda.
- "**perc**". Per le marche per cui non è disponibile il dataset "stock" (Bayer, Beyond Meat e Nestlé) abbiamo deciso di aggiungere i dati "perc", semplicemente ottenuti dividendo il punteggio di polarità per il numero di tweet pubblicati nel giorno stesso. Questo perché queste marche sono tutte appartenenti alla categoria "a basso traffico": in questo modo il dataset "perc" tiene conto esclusivamente della polarità, mentre il dataset "score" sia della polarità che del numero di tweet prodotti nella giornata.

Abbiamo poi analizzato le differenze nei punteggi ricavati dall'utilizzo dei diversi dataset, plottandoli tutti nello stesso grafico.
Per concludere l'analisi delle differenze e somiglianze tra metodi di sentiment analysis abbiamo prodotto la matrice di correlazione.

Per quanto riguarda invece l'analisi delle correlazioni tra polarità dei tweet e serie storica dei prezzi azionari abbiamo:

- Plottato le due serie in un grafico, per consentire un'analisi visiva. Per semplificare l'interpretazione del grafico abbiamo aggiunto anche, per ogni serie, il suo lisciamento, ottenuto con la funzione "geom_smooth()" di ggplot, con metodo "loess". Per evitare un grafico troppo confuso abbiamo sempre utilizzato, per la serie di polarità, solo quella ottenuta con metodo udpipe.
- Costruito la matrice di correlazione tra le combinazioni di metodi di analisi e dataset utilizzato con le serie di apertura (Open) e chiusura (Close). Abbiamo voluto inserire

la serie di apertura perché dato che, come è già stato detto, il download dei tweet è avvenuto all'orario di apertura della borsa americana, si voleva indagare una ulteriore correlazione con tale prezzo. Nonostante ciò, la variabile da prevedere rimane il valore di chiusura.

A questo punto, valutiamo la capacità previsiva delle serie storiche da noi costruite. Inizialmente avevamo deciso di utilizzare il test di causalità di Granger. La causalità di Granger è un concetto espresso nel 1969 da Clive Granger e ampliato successivamente da Christopher Sims mirante a determinare una causalità tra le variabili espresse in un modello. Una serie storica $\{x_t\}_t$ causa (nel senso di Granger) una seconda serie storica $\{y_t\}_t$ se, condizionando rispetto ai valori passati di x_t, l'errore quadratico medio di previsione della $y + t$ risulta ridotto rispetto al caso in cui l'informazione dei valori passati di x_t sia ignorata. Formalmente:

$$E[y_t - E(y_t|\cdot) \mid y_{t-1}, y_{t-2}, \dots; x_{t-1}, x_{t-2}, \dots]^2 \leq E[y_t - E(y_t|\cdot) \mid y_{t-1}, y_{t-2}, \dots]^2$$

Semplicemente guardando la formula, però, è semplice accorgersi che non è esattamente ciò che vogliamo, e che alcune modifiche devono essere apportate per essere adeguato allo scopo della nostra ricerca.

Prima di tutto non vogliamo che il primo errore quadratico sia condizionato solo ai valori passati di x_t, vogliamo che lo sia anche a quello presente. Le informazioni che noi ricaviamo tramite il download dei tweet sarebbero infatti disponibili già dalle 17.30 (orario di download), mentre la chiusura della borsa sarà solo alle 22. Ciò che vogliamo scoprire è se queste informazioni, ricavate 4 ore e mezza prima, sono utili per predire il prezzo di chiusura. Per seconda cosa, vogliamo che i dati forniti dalla sentiment analysis siano utili considerando tutti gli altri dati già disponibili alle 17.30: il più importante di tutti è il prezzo di apertura. In conclusione, secondo questa variante del test di Granger appositamente costruita, la nostra serie storica $\{s_t\}_t$ causa la serie storica dei prezzi di chiusura $\{c_t\}_t$ se, condizionando rispetto ai valori presenti e passati di s_t e della serie storica dei prezzi di apertura $\{o_t\}_t$, l'errore quadratico medio di previsione della $c + t$ risulta ridotto rispetto al caso in cui l'informazione dei valori presenti e passati di s_t sia ignorata. Formalmente:

$$E[y_t - E(y_t|\cdot) \mid y_{t-1}, y_{t-2}, \dots; x_t, x_{t-1}, \dots; o_t, o_{t-1}, \dots]^2 \leq E[y_t - E(y_t|\cdot) \mid y_{t-1}, y_{t-2}, \dots; o_t, o_{t-1}, \dots]^2$$

La metodologia di calcolo di questo nuovo test, a cui mi riferirò per comodità come "test Close", è identica a quella del Granger Test. Il test G. infatti è semplicemente un test F sulla

significatività dei valori ritardati della variabile x_t nel modello completo. Allo stesso modo, il test Close costruisce un modello completo di ritardi della variabile dipendente, valori presenti e ritardati della variabile s_t e valori presenti e passati della variabile o_t, e calcola il test F sulla significatività congiunta della variabile s_t e i suoi ritardi.

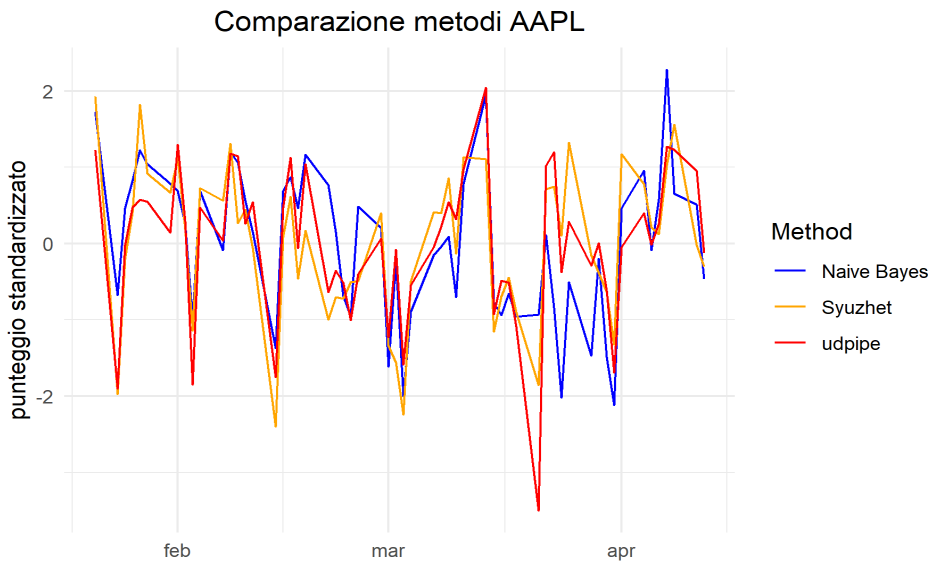
La funzione closeTest() da noi costruita prende così in input le serie da utilizzare come "s", "c" e "o", e il numero di ritardi da applicare. Costruisce quindi il modello completo del numero di ritardi scelto, e utilizza la funzione "linearHypothesis()" della libreria "car"[11] per valutare la significatività congiunta della variabile "s" e dei suoi ritardi. Riporta quindi in output il p-value risultante dal test F.

È importante sottolineare, a questo punto, che tutti i dati di tutte le marche sono categorizzati per giorno di download. I dati relativi al giorno *x* comprendono i tweet scaricati nel giorno *x,* non quelli scritti e pubblicati in dato giorno. In questo modo, anche per le marche i cui tweet coprivano l'intera giornata fino al download del giorno precedente, il test Close lavora sempre sui dati disponibili già dalle ore 17:30, cioè ben prima dell'orario di chiusura della borsa americana.

3.1 Apple

Nel grafico seguente (*immagine 7*) è riportato l'andamento temporale della polarità dei giudizi che gli utenti hanno espresso su Apple, sui suoi prodotti e sul suo andamento in borsa, ricavato attraverso i tre metodi precedentemente descritti.

Immagine 7: Comparazione dei tre metodi attraverso i quali è stata ottenuta la serie storica della polarità dei giudizi su Apple



È facile notare, semplicemente guardando il grafico, come ci sia un sostanziale accordo tra i tre metodi di sentiment analysis. Ricordiamo che le differenze di Syuzhet dagli altri metodi sono dovute sia ad una diversa modalità di calcolo del punteggio che a un diverso dizionario. Le differenze invece tra udpipe e Naive Bayes sono esclusivamente dovute alla modalità di calcolo e alla possibilità di utilizzo, da parte del primo, di intensificatori e negatori, in quanto il dizionario utilizzato è il medesimo.

Le stesse similitudini si ritrovano comparando sia le serie storiche delle polarità espresse nei confronti di Apple (escludendo i suoi prodotti) (*immagine 8*), che quelle esclusivamente riferite al codice di Apple sul mercato azionario ($AAPL) (*immagine 9*).

Immagine 8: Comparazione dei tre metodi attraverso i quali è stata ottenuta la serie storica della polarità dei giudizi su Apple, escludendo i tweet che parlassero dei prodotti della marca

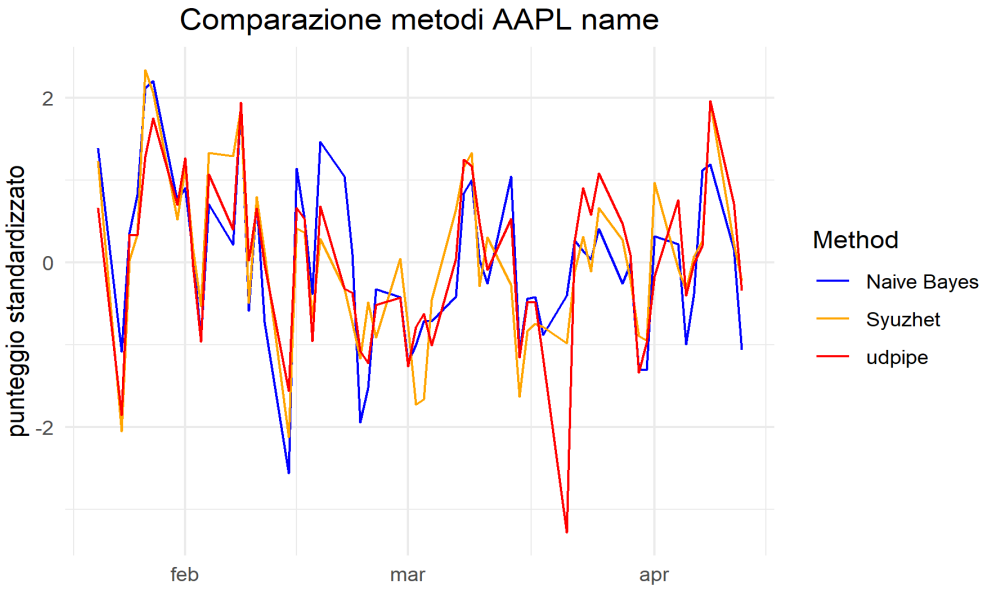
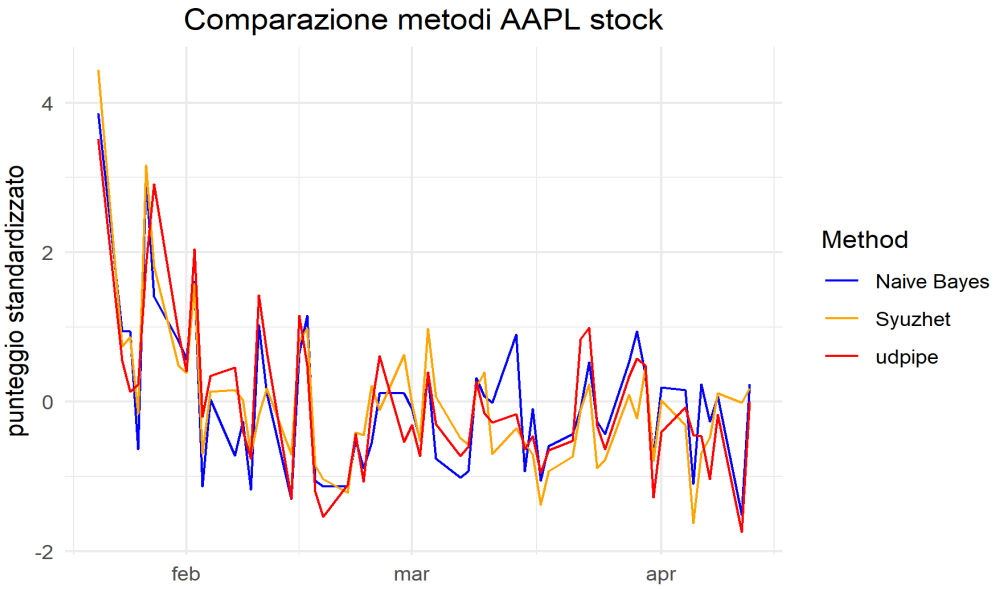
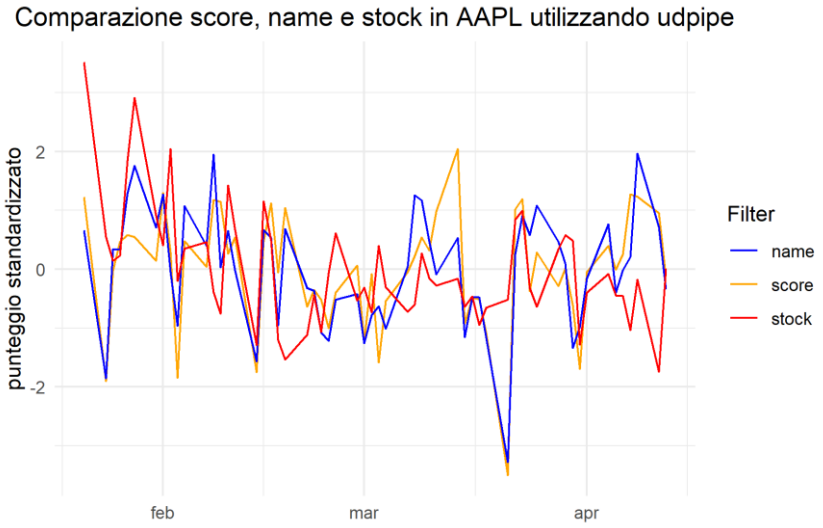


Immagine 9: Comparazione dei tre metodi attraverso i quali è stata ottenuta la serie storica della polarità dei giudizi su Apple, utilizzando solo i tweet riferiti al codice $AAPL



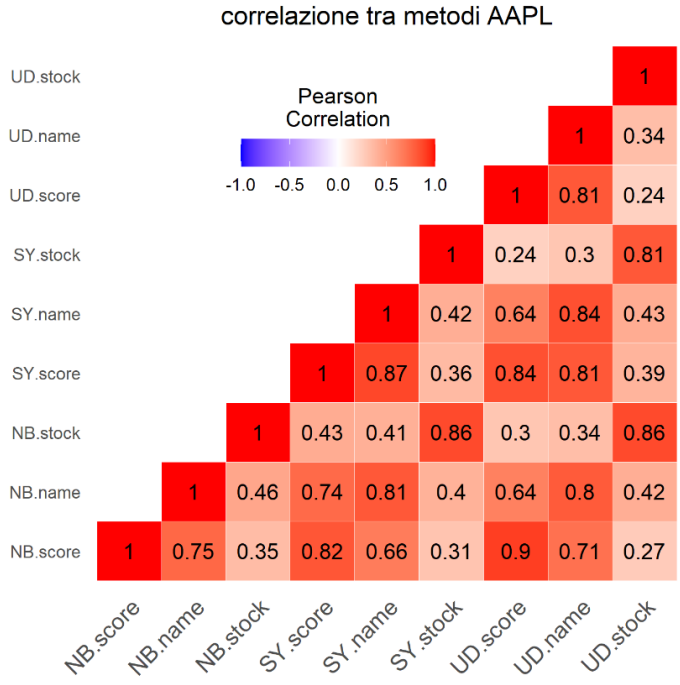
Utilizzando udpipe per confrontare la serie storica riferita al dataset completo ("score") con quelle riferite ai dataset filtrati per nome e codice ("name") o solo per codice ("stock"), notiamo invece delle sostanziali differenze (*immagine 10*):

Immagine 10: Comparazione della serie "score", "name" e "stock", utilizzando udpipe, per Apple



La serie "stock", in particolare, si differenzia dalle altre due in periodi di picchi negativi o positivi come, rispettivamente, quelli nella seconda metà di marzo e nella prima metà di aprile. Il distacco è forte anche nelle prime osservazioni, dove l'utilizzo dei soli tweet contenenti il codice "$AAPL" porta a un punteggio molto più positivo della polarità percepita. Queste osservazioni sono provate dal calcolo delle correlazioni tra serie storiche (*Immagine 11*).

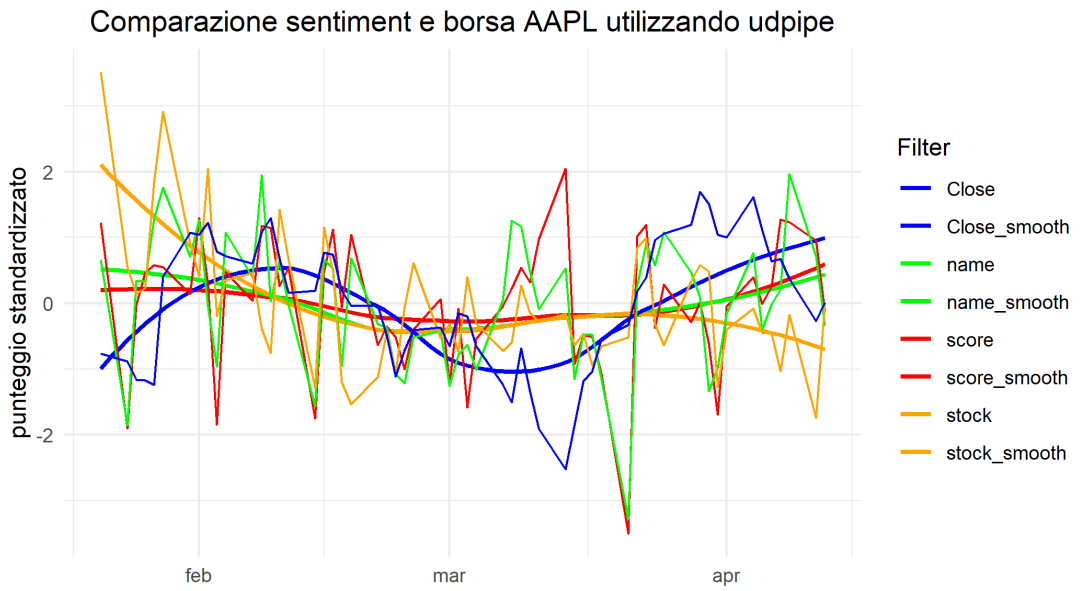
Immagine 11: matrice di correlazione tra i diversi metodi utilizzati per analizzare la polarità di Apple



La heatmap risultante mostra una forte correlazione tra i diversi metodi con cui è stata affrontata l'analisi. Lo notiamo dal fatto che, a parità di dataset utilizzato ("score", "name" o "stock"), le correlazioni tra i diversi metodi ("NB": Naive Bayes, "SY": Syuzhet e "UD": udpipe) sono molto elevate. Lo stesso si può dire tra serie analizzate attraverso lo stesso metodo e facenti riferimento ai dataset "score" e "name". Come però avevamo già intuito dal grafico, le serie ricavate dal dataset "stock" differiscono dalle altre. Ciò si vede perché, a parità di metodo utilizzato per l'analisi (NB, SY o UD), le correlazioni tra le serie "stock" e quelle "score" o "name" sono tutte molto basse.

A questo punto, però, la cosa più importante è verificare la relazione che queste serie hanno con quella che vogliamo prevedere, cioè quella dei prezzi giornalieri di AAPL alla chiusura della borsa americana ("Close").
Sempre tenendo come metodo di riferimento udpipe, andiamo a mappare sullo stesso grafico tutte le serie che vogliamo analizzare (*immagine 12*).

Immagine 12: comparazione serie storiche di polarità e dei prezzi di AAPL

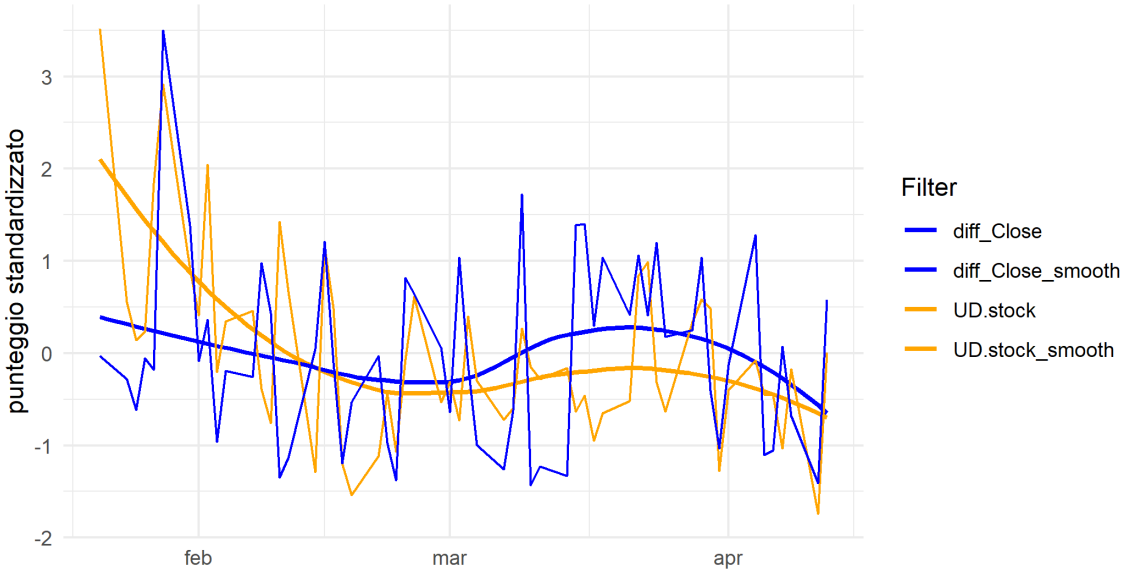


Le tre serie "score", "name" e "stock" sembrano essere a prima vista del tutto incorrelate con i prezzi delle azioni di Apple. Mentre ciò può essere effettivamente vero per le prime due, abbiamo notato come invece la terza, soprattutto nella sua prima metà, sembri comportarsi come una derivata della serie Close. Seguendo la serie lisciata dall'inizio vediamo infatti come, per valori positivi di questa, la serie lisciata di Close cresca. Continuando, all'avvicinarsi della prima allo zero la seconda rallenta la sua crescita, e all'intersezione della prima con l'asse delle

ascisse la seconda raggiunge un massimo relativo. Dopodiché, a valori negativi della prima corrisponde una decrescita della seconda.

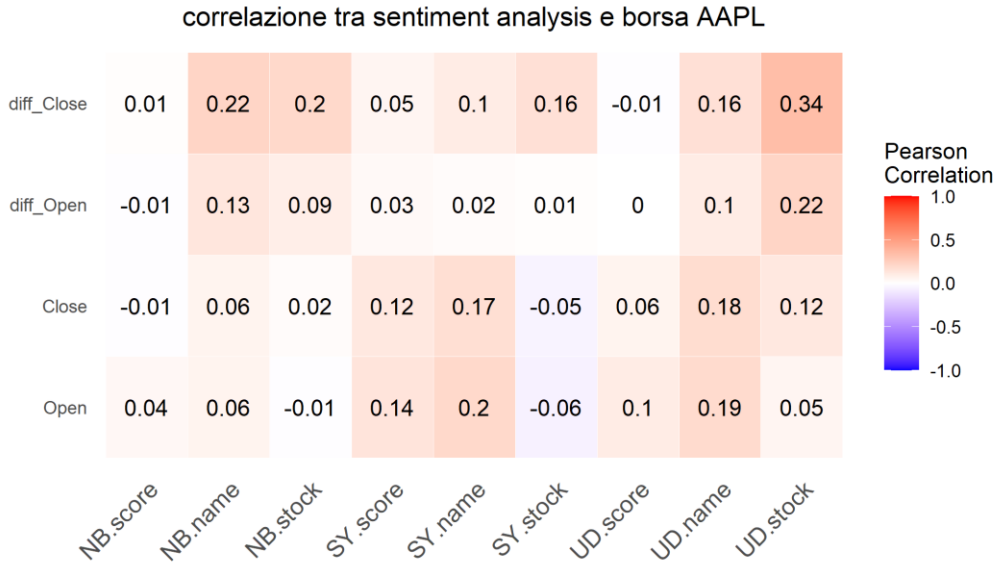
Analizzando in maniera campionaria i tweet troviamo anche una spiegazione logica a questo comportamento: un aumento del valore delle azioni di Apple comporta un aumento del valore del "portfolio" degli azionisti che hanno investito nell'azienda, e il contrario si può dire per un calo. Ciò si tradurrà in emozioni e valutazioni positive del marchio quando questo aumenterà di valore, e negative quando diminuirà. Questa naturale reazione degli azionisti a un crollo o un aumento dell'azione potrebbe essere il motivo per cui l'analisi dei sentimenti non rispecchia il prezzo al variare del tempo, ma le sue variazioni. Andando a mappare "stock" e la serie ottenuta tramite le differenze di "Close" (che chiamiamo "diff_Close") possiamo notare visivamente una certa relazione tra le due, che si palesa ancora di più nel lisciamento delle due serie (*Immagine 13*).

Immagine 13: comparazione "stock" e "diff_Close"



La correlazione tra la serie "stock", analizzata tramite metodo udpipe, e la serie di differenze di "Close" si rivela la più forte nel caso di Apple (0,34) (*Immagine 14*). Tutti gli altri dati rivelano correlazioni molto basse tra gli altri metodi di analisi e le variabili dipendenti.

Immagine 14: Correlazione tra i metodi di sentiment e i valori di AAPL in borsa



Per quanto riguarda Apple, il test Close trova una relazione di causalità tra NB.stock e SY.stock sulla serie differenziata dei prezzi di chiusura e ritardo pari a 1, con p-value rispettivamente uguali al 0,089 e 0,091. Ciò sembra indicare che, almeno nel caso di Apple, la sentiment analysis fornisca un'ulteriore informazione che permette di migliorare la previsione della quota azionaria dell'azienda.

3.2 GOOGLE

La similitudine tra i diversi metodi di analisi si ritrova anche in Google e, come vedremo, un po' in tutte le marche analizzate. Allo stesso modo, viene confermata sia graficamente che dalla matrice di correlazione anche la sostanziale differenza dei risultati ricavati dall'utilizzo del dataset "score" e "stock" (*immagine 15 e 16*).

Immagine 15: comparazione dei metodi utilizzati per l'analisi sia tramite il dataset "score" (in alto) che quello "stock" (in basso) per Google

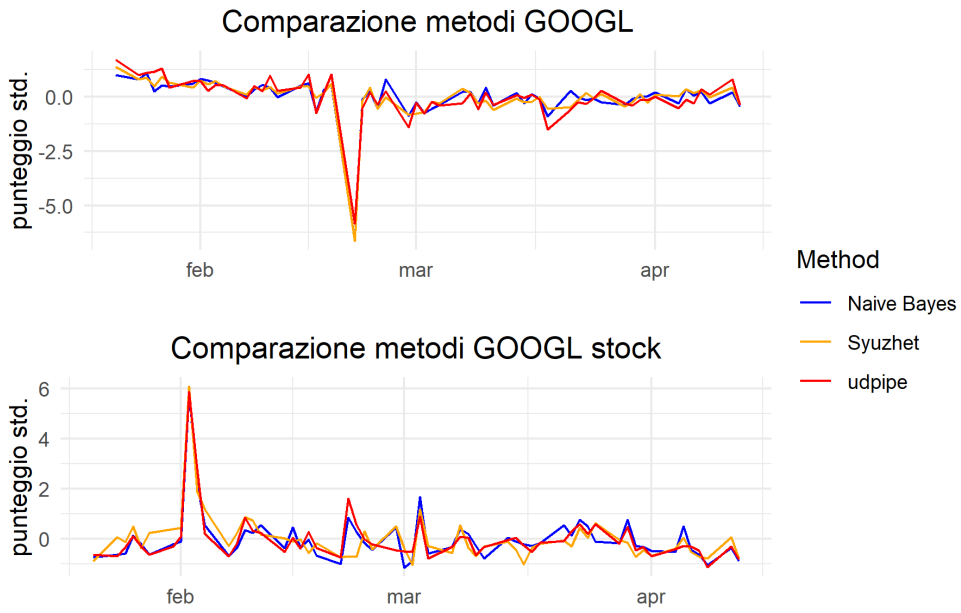
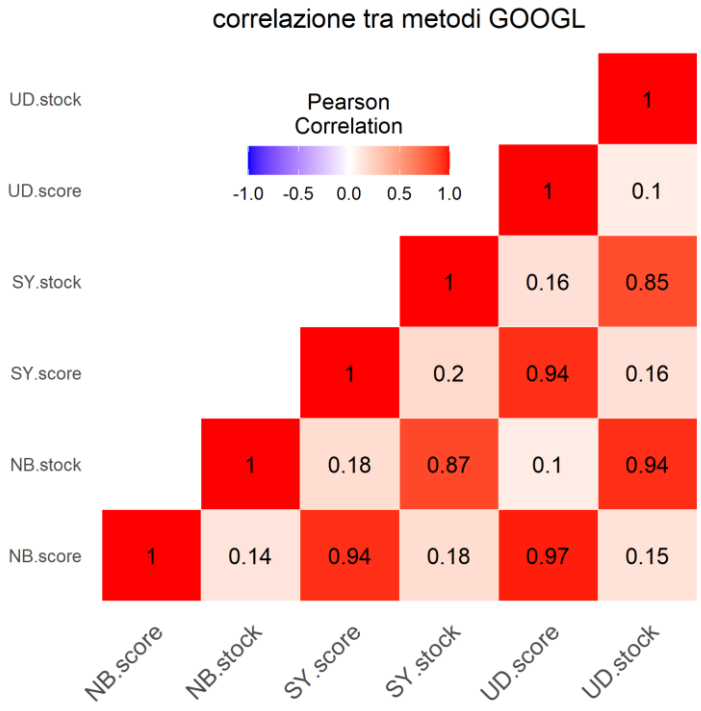


Immagine 16: heatmap delle correlazioni tra i diversi metodi di analisi per Google



La somiglianza tra analisi con uguale dataset e diverso metodo e la differenza tra analisi con uguale metodo ma diverso dataset è, per Google, ancora più accentuata. Le correlazioni fra le prime, infatti, non scendono sotto lo 0,85 (registrato tra UD.stock e SY.stock); le correlazioni tra le seconde, invece, non salgono sopra lo 0,2 (registrato tra SY.score e SY.stock).

Per quanto riguarda la relazione con la borsa, invece, l'analisi dei sentimenti di Google dimostra graficamente una maggiore correlazione, rispetto ad Apple, sia per la serie dei prezzi di chiusura (*immagine 17*), sia per quella delle differenze dei prezzi (*immagine 18*) utilizzando il dataset stock.

Immagine 17: comparazione sentiment e borsa GOOGL

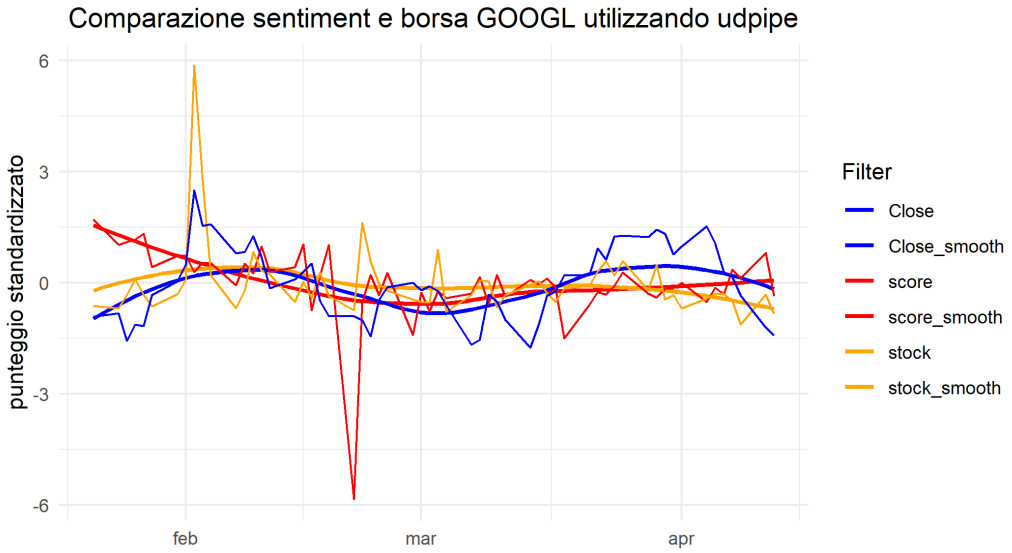
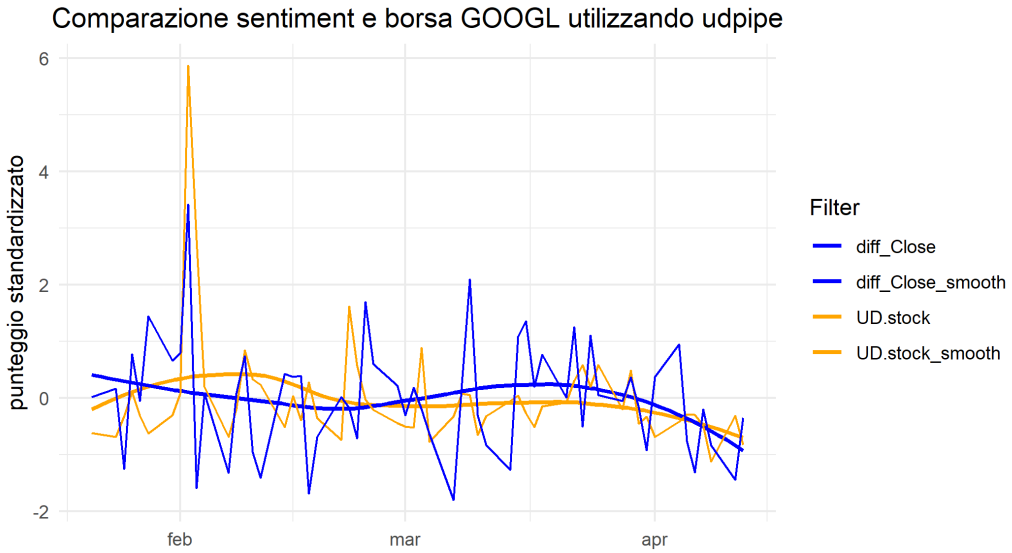
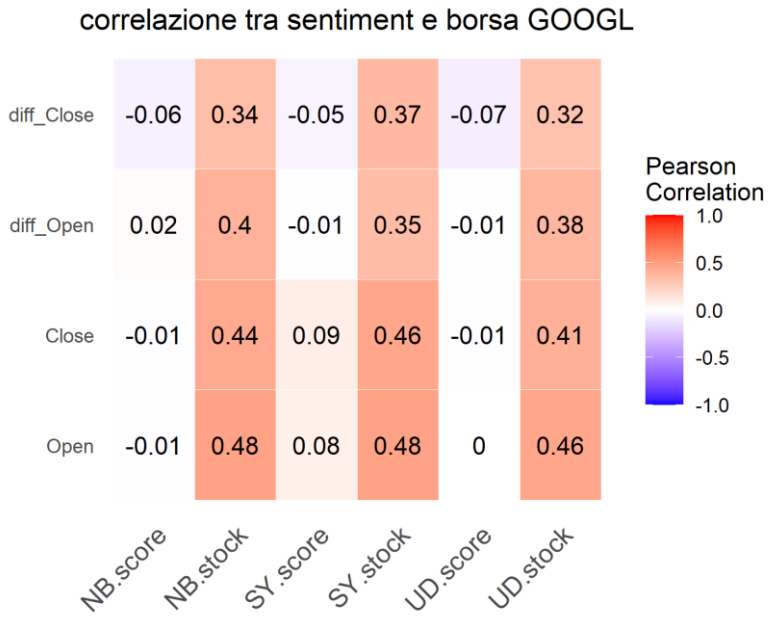


Immagine 18: comparazione sentiment e serie delle differenze della borsa GOOGL



La mappa di correlazione (*Immagine 19*) conferma la forte correlazione del dataset stock sia con la serie originale Close che, anche se meno, con quella nelle differenze diff_Close. La correlazione più forte con le serie di chiusura la troviamo fra il dizionario Syuzhet e la serie dei prezzi originale (0,46). Il dataset completo, invece, non sembra essere per nulla correlato con nessuna serie dei prezzi.

Immagine 19: correlazione tra i tre metodi di sentiment analysis sui due dataset e le serie dei prezzi di GOOGL



Questa forte correlazione si traduce in risultati, da parte del test Close, molto soddisfacenti. Al ritardo 1, Naive Bayes, con dataset "stock", registra un p-value di 0,068, mentre Syuzhet e udpipe sono entrambi significativi al 5%: rispettivamente 0,032 e 0,022. Viene così confermato che, per Google, possedere all'apertura della borsa americana informazioni riguardanti la polarità dei tweet con codice "$GOOGL" delle ultime ore ci permette di costruire un modello previsivo più accurato del modello contenente solo il prezzo di chiusura del giorno precedente e il prezzo di apertura odierno.

Il test Close ci fa anche notare una causalità da parte di tutti e tre i metodi con dataset completo sulle serie Close e diff_Close a ritardi più elevati. Nella *tabella 1* riportiamo le più interessanti:

Tabella 1: test Close con risultati più significativi per ritardi elevati di Google

modello	ritardo	p-value	
diff_Close ~ NB.score	4	0,0121	**
diff_Close ~ SY.score	4	0,0111	**
Close ~ SY.score	4	0,0370	**
diff_Close ~ NB.score	5	0,0256	**
Close ~ NB.score	5	0,0487	**
diff_Close ~ SY.score	5	0,0240	**
Close ~ SY.score	5	0,0397	**

3.3 NIKE

Ancora una volta, l'analisi di Nike dimostra, in maniera ancora più accentuata, le similarità tra i metodi e le differenze tra l'utilizzo del dataset completo "score" e quello filtrato per codice $NKE "stock" (Immagini 20 e 21). Queste ultime portano addirittura, come dimostrato dalla matrice di correlazione, ad una totale incorrelazione (o addirittura leggera correlazione negativa) tra l'analisi dei sentimenti ricavata dal dataset intero e quello filtrato.

Immagine 20: comparazione dei metodi utilizzati per l'analisi sia tramite il dataset "score" (in alto) che quello "stock" (in basso) per Nike

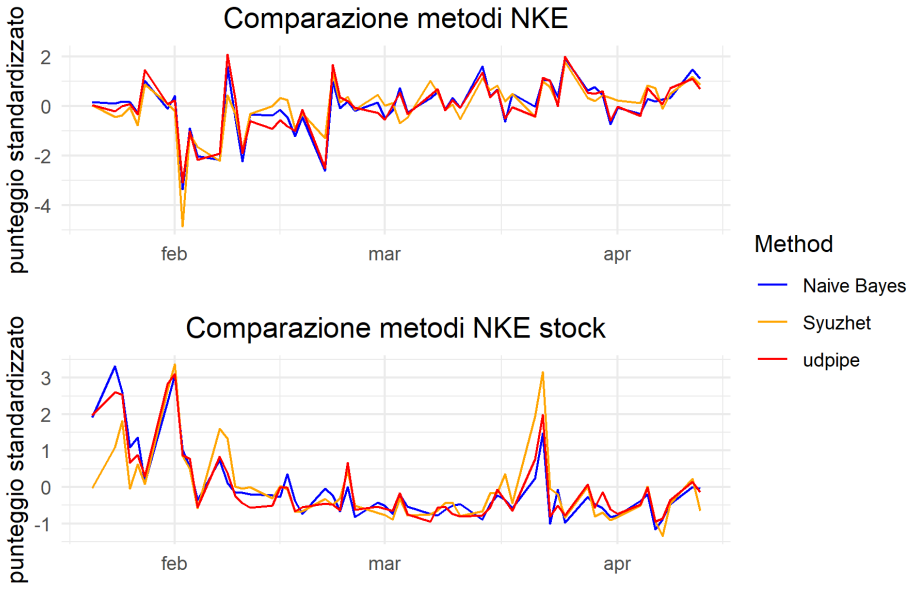
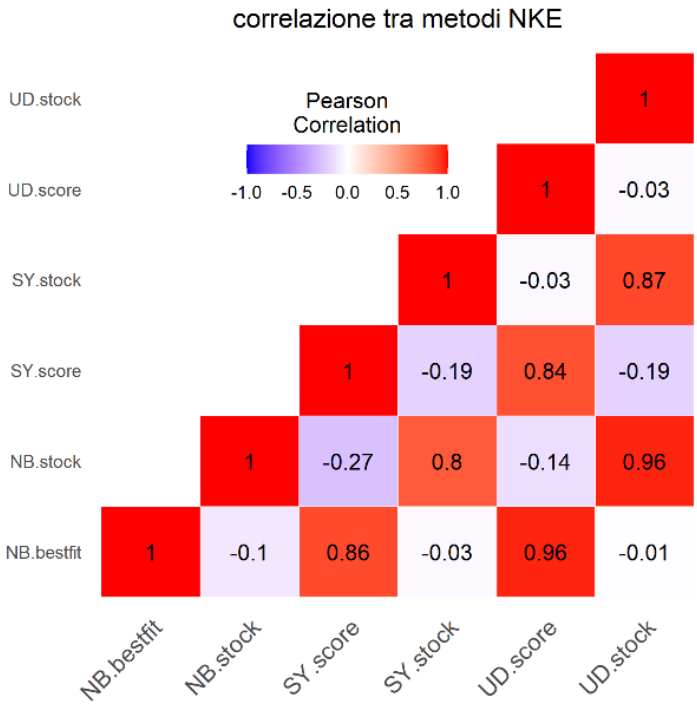


Immagine 21: heatmap delle correlazioni tra i diversi metodi di analisi per Nike



Come nelle marche precedenti, fra le due, l'analisi ricavata dal dataset "stock" si rivela sia graficamente che statisticamente più correlata con la nostra variabile dipendente, rispetto all'analisi ricavata dal dataset completo (*Immagini 22, 23 e 24*).

Immagine 22: comparazione sentiment e borsa NKE

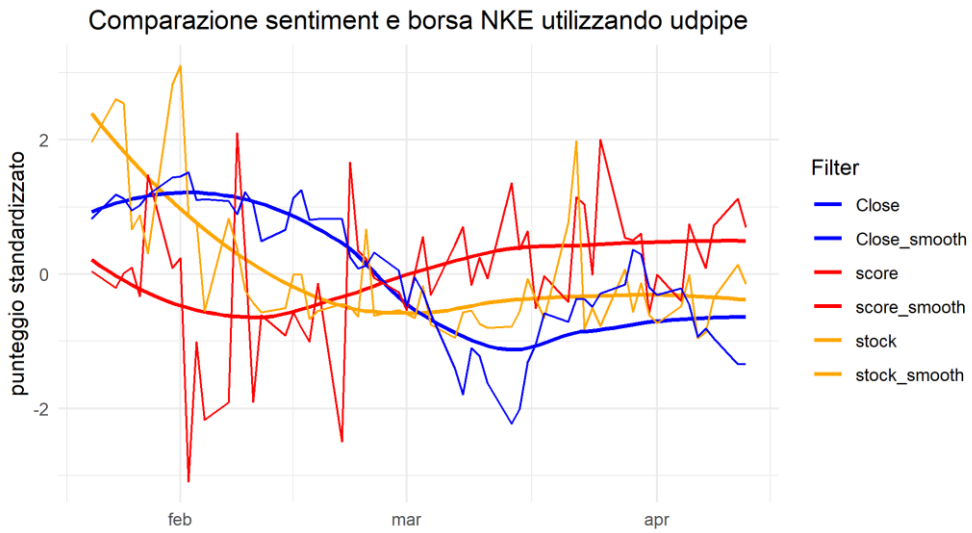


Immagine 23: comparazione sentiment e serie delle differenze della borsa NKE

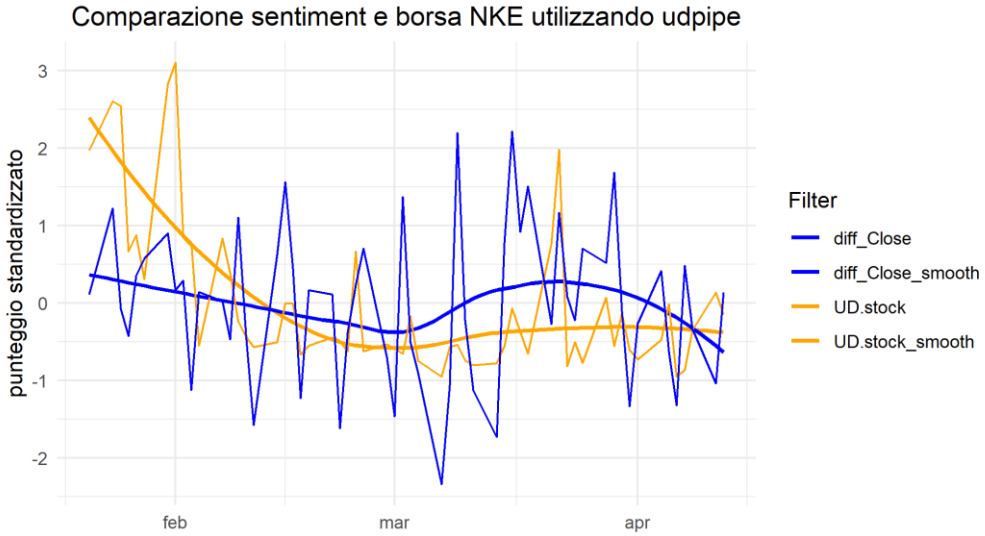
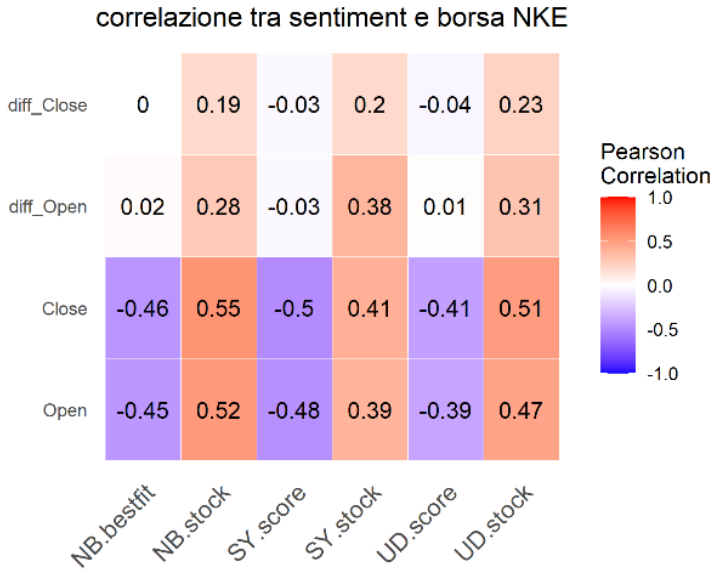


Immagine 24: correlazione tra i tre metodi di sentiment analysis sui due dataset e le serie dei prezzi di NKE



Anche qui possiamo ritrovare lo stesso pattern di Google: le serie maggiormente correlate sono quelle ricavate dai dataset "stock" con la serie dei prezzi di chiusura originale. Per la prima volta, però, appaiono forti correlazioni negative, tra le serie ricavate dai dataset completi e i prezzi di apertura e chiusura.

Il test Close rileva, al ritardo 1, capacità previsiva dei metodi Naive Bayes e udpipe calcolati su dataset "stock" sulla serie Close, rispettivamente con p-value uguale a 0,066 (*) e 0,042 (**). Già dal ritardo 2, però, è il dataset "score" che raggiunge i risultati migliori, come mostrato nella *tabella 2*.

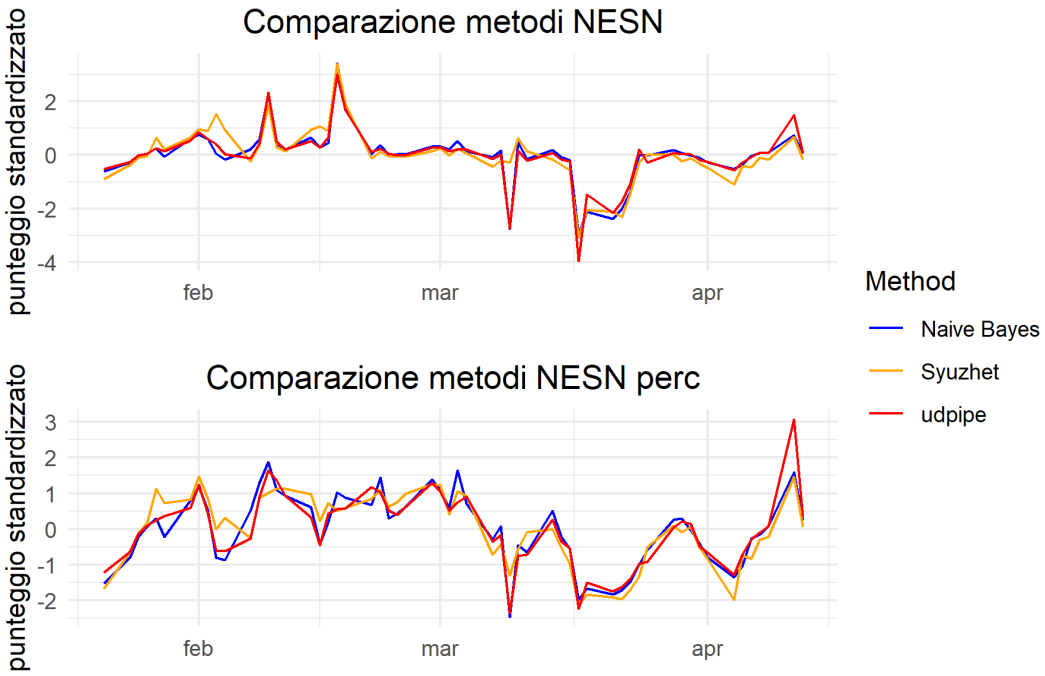
Tabella 2: test Close con risultati più significativi per ritardi di Nike

Modello	Ritardo	p-value	
diff_Close ~ NB.score	2	0,0807	*
diff_Close ~ SY.score	2	0,0752	*
diff_Close ~ UD.score	2	0,0702	*
diff_Close ~ SY.score	4	0,0868	*
diff_Close ~ UD.score	4	0,0853	*
diff_Close ~ NB.score	5	0,097	*
diff_Close ~ UD.score	5	0,0791	*

3.4 NESTLÉ

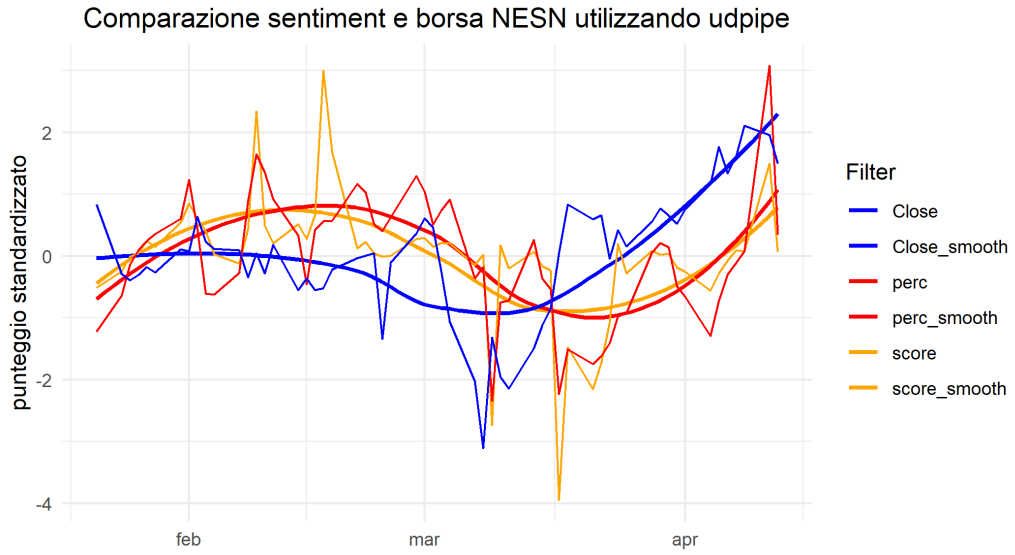
Nestlé è la prima marca che analizziamo in cui non è presente il dataset “stock”, e in cui quindi utilizziamo il dataset “perc”. Come si può immaginare, i dataset saranno più correlati di quanto lo fossero “score” e “stock”, lo si può notare dal grafico dell’*immagine 25*.

Immagine 25: comparazione dei metodi utilizzati per l’analisi sia tramite il dataset “score” (in alto) che quello “stock” (in basso) per Nestlé



Comparando le nostre serie di polarità con la serie dei prezzi di chiusura, possiamo notare, in particolare dalle serie lisciate, come sembrerebbe che siano le prime a seguire la seconda, dopo un certo intervallo di tempo (*Immagine 26*).

Immagine 26: comparazione sentiment e borsa NESN



In effetti, Nestlé è l'unica marca per cui il test Close non rileva nessuna capacità previsiva delle serie di sentiment sulle serie Close e diff_Close. Applicando però un granger test inverso, andando così a valutare un rapporto di causa effetto delle variazioni dei prezzi in borsa sulle polarità dei giudizi espressi su Twitter, troviamo per i ritardi 3, 4 e 5, rapporti tutti significativi allo 0,05 (**) o 0,01 (***). Si può quindi affermare che, almeno per Nestlé, le polarità dei giudizi degli utenti di Twitter non influiscono sul prezzo di chiusura della marca, ma che è piuttosto vero il contrario: i movimenti del prezzo modifica le polarità degli utenti a distanza di qualche giorno. Il granger test per Nestlé è stato effettuato grazie alla funzione grangertest() della libreria "lmtest"[12].

3.5 BEYOND MEAT

Beyond Meat è probabilmente la marca meno conosciuta del nostro campione, ne è prova il fatto che è quella con meno tweet scaricati (*Immagine 4*). L'*immagine 27* ci mostra le serie storiche sentiment da noi costruite, mentre l'*immagine 28* le pone a confronto con la serie storica dei prezzi di chiusura.

Immagine 27: comparazione dei metodi utilizzati per l'analisi sia tramite il dataset "score" (in alto) che quello "stock" (in basso) per Beyond Meat

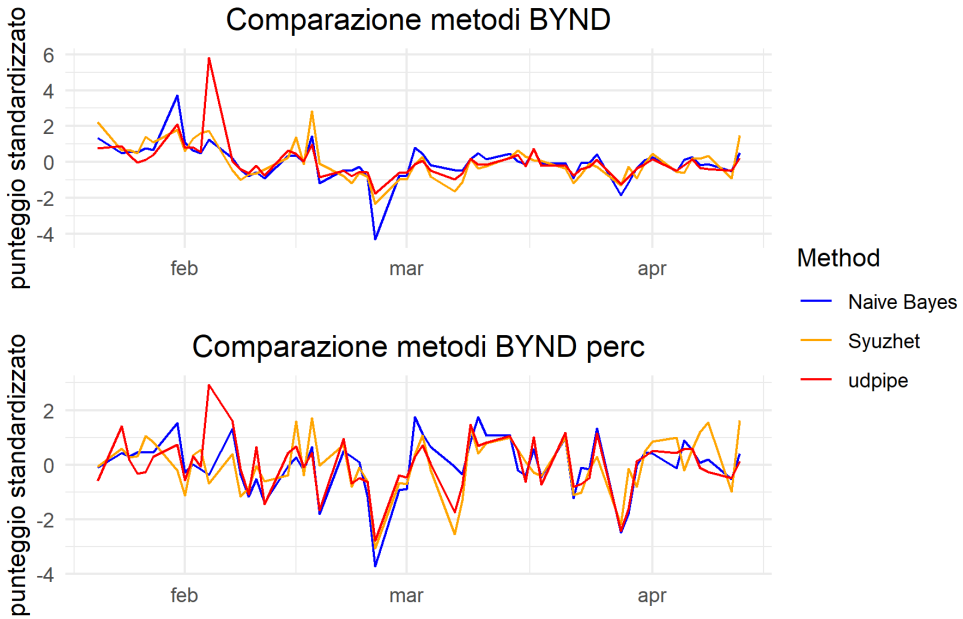
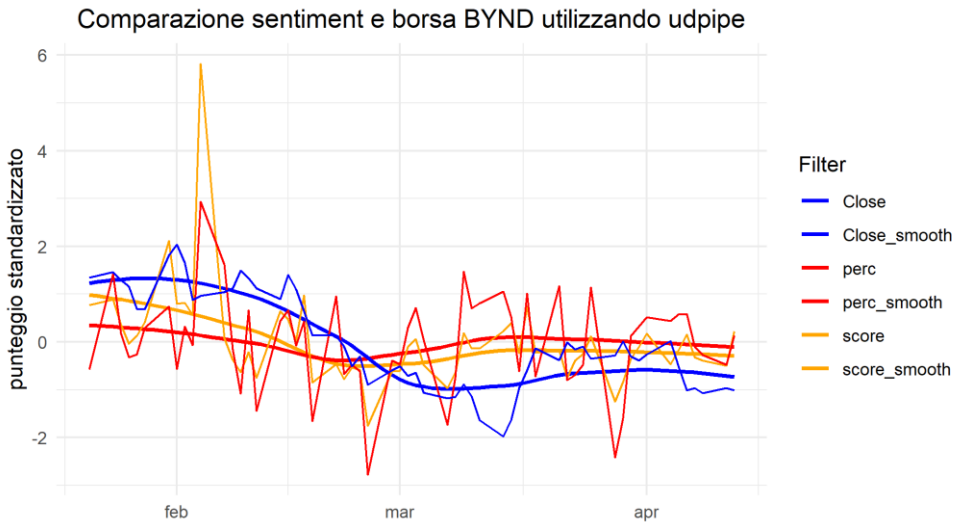


Immagine 28: comparazione sentiment e borsa BYND



Il Close test rileva, per Beyond Meat, una capacità previsiva della serie ottenuta tramite metodo Naive Bayes, nella sua variante percentuale, sulla serie Close in tutti i ritardi. Altre relazioni di causa effetto significative la ottiene la stessa serie su diff_Close a ritardo 4 e la serie ottenuta tramite metodo Syuzhet, sempre nella sua variante percentuale, sulla serie Close a ritardo 1. Tutte queste relazioni sono sintetizzate nella *Tabella 3*.

Tabella 3: test Close con risultati più significativi per Beyond Meat

Modello	Ritardo	P-value	
Close ~ NB.perc	1	0,0323	**
Close ~ SY.perc	1	0,0442	**
Close ~ NB.perc	2	0,0228	**
Close ~ NB.perc	3	0,0507	*
diff_Close ~ NB.perc	4	0,0353	**
Close ~ NB.perc	4	0,0458	**
Close ~ NB.perc	5	0,0551	*

3.6 BAYER

Passando ora all'analisi delle aziende farmaceutiche, Bayer dimostra ancora un sostanziale accordo tra diversi metodi di sentiment analysis (*immagine 29*). L'*Immagine 30*, invece, mostra per questa marca il confronto tra il metodo udpipe e la serie dei prezzi Close.

Immagine 29: comparazione dei metodi utilizzati per l'analisi sia tramite il dataset "score" (in alto) che quello "stock" (in basso) per Bayer

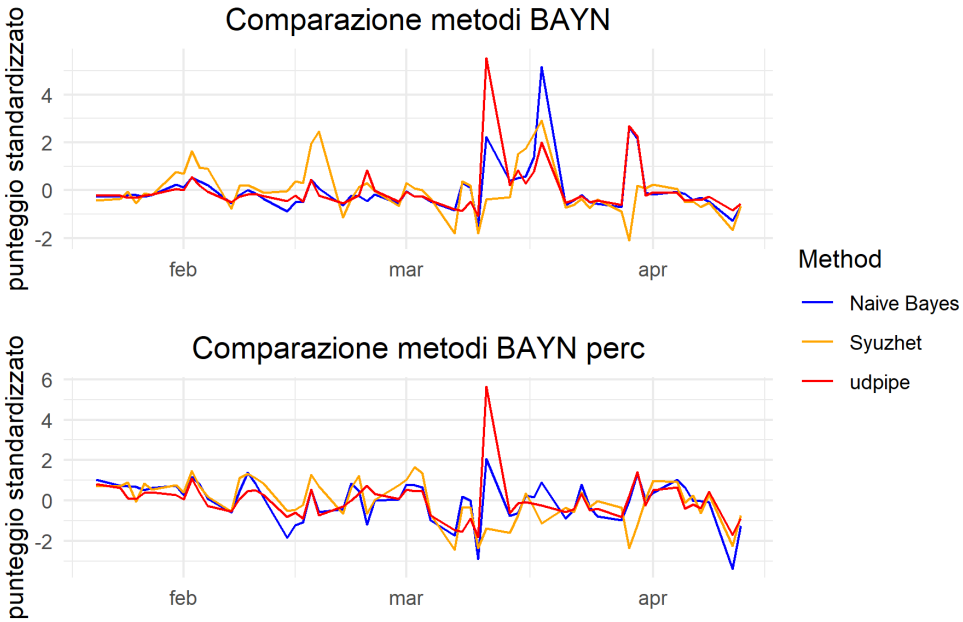
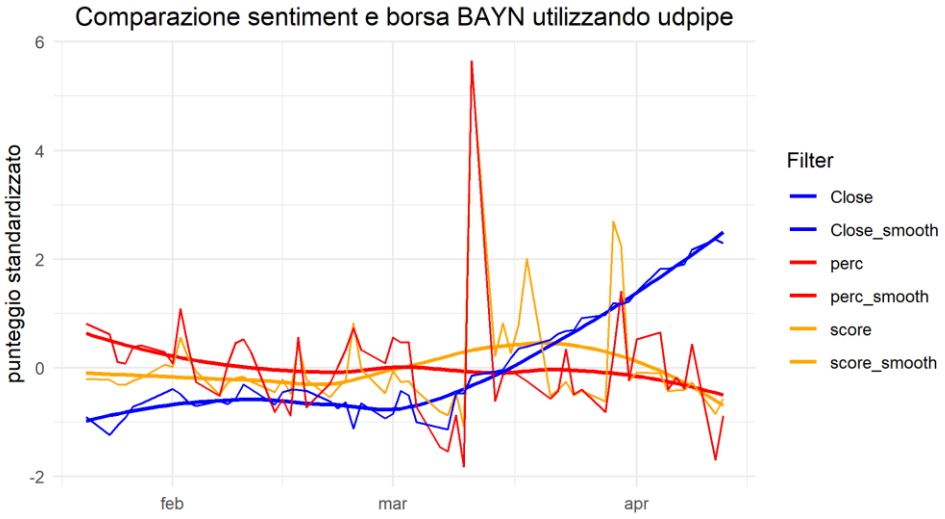


Immagine 30: comparazione sentiment e borsa BAYN



Nonostante il grafico non dimostri, visivamente, un forte legame tra le due serie, il test Close rileva molte relazioni di causa effetto tra le due serie. I metodi Syuzhet e udpipe, soprattutto nella loro variante percentuale, dimostrano infatti per i ritardi dal 2 al 4 capacità previsive significative allo 0,10 (*) sia con la serie Close che diff_Close. Al ritardo 5, invece, il test Close rileva per udpipe relazioni causali su entrambe le serie dei prezzi di chiusura con significatività al 5% o all'1% (*tabella 4*).

Tabella 4: test Close con risultati più significativi per ritardi elevati di Bayer

modello	ritardo	p-value	
Close ~ UD.score	5	0,0156	**
diff_Close ~ UD.perc	5	0,0087	***
Close ~ UD.perc	5	0,0105	**

3.7 NovaVax

Nelle *Immagini 31 e 32*, possiamo rispettivamente confrontare i diversi metodi utilizzati per l'analisi dei tweet dell'ultima azienda del campione, e paragonare il metodo udpipe con la serie storica Close.

Immagine 31: comparazione dei metodi utilizzati per l'analisi sia tramite il dataset "score" (in alto) che quello "stock" (in basso) per NovaVax

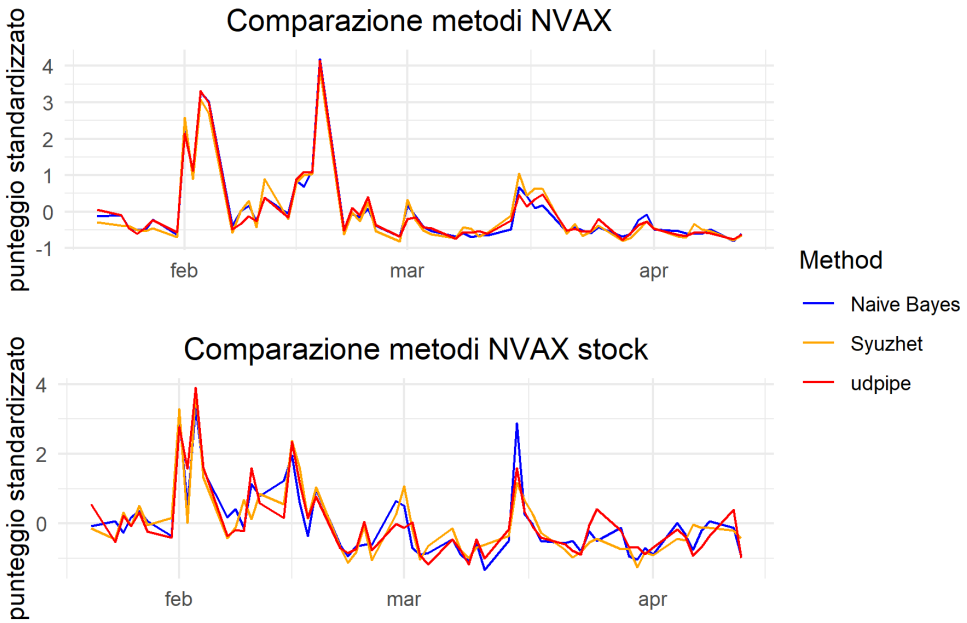
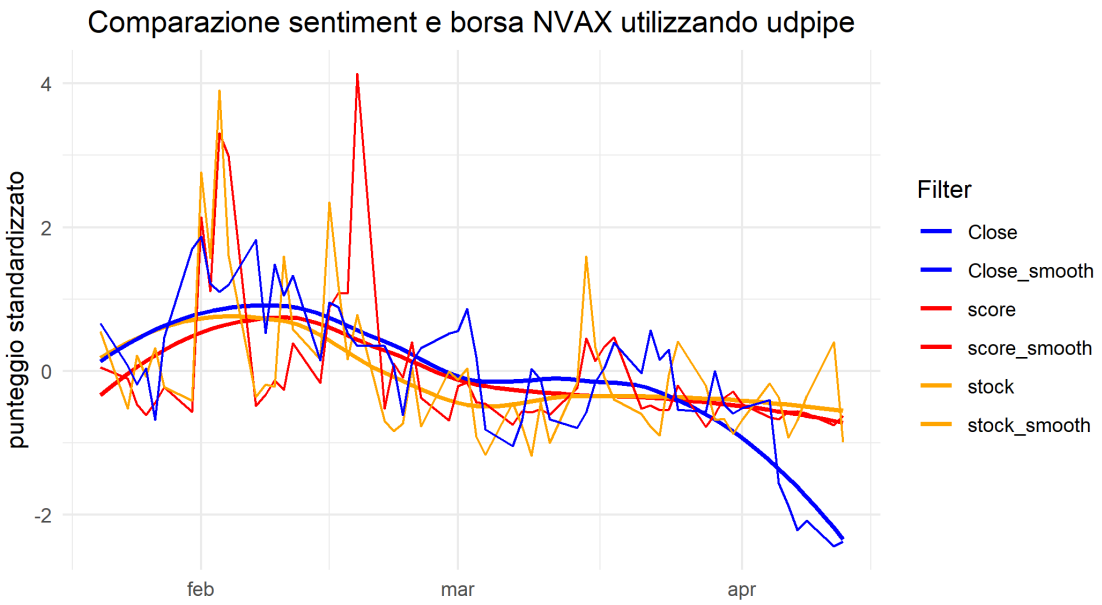


Immagine 32: comparazione sentiment e borsa NVAX



Dal grafico dell'immagine 32 è possibile già intravedere una buona similitudine tra le serie di sentiment e quella dei prezzi, soprattutto se ci concentriamo sulle serie lisciate. Il test Close conferma la presenza di una capacità previsiva da parte di tutti e tre i metodi, calcolati su dataset "score", su entrambe le serie Close e diff_Close (*tabella 5*), tutti al ritardo 5.

Tabella 5: test Close con risultati più significativi per NovaVax

modello	ritardo	p-value	
diff_Close ~ NB.score	5	0,0613	*
Close ~ NB.score	5	0,0442	**
Close ~ SY.score	5	0,0936	*
diff_Close ~ UD.score	5	0,0929	*
Close ~ UD.score	5	0,0586	*

3.8 CONCLUSIONI

Per quanto riguarda i metodi con cui è stata affrontata l'analisi dei sentimenti (Naive Bayes, Syuzhet e udpipe), tutti e tre hanno portato, lungo tutta la ricerca, a serie storiche molto simili e correlate. Ciò ha fatto sì che tutti e tre i metodi si rivelassero ugualmente efficaci a predire la variabile dipendente: le relazioni di causalità, rilevate dal test Close, in cui Naive Bayes è il metodo usato per costruire la variabile indipendente sono 17, 19 per Syuzhet e 19 per Udpipe. I tre metodi appaiono cioè nello stesso numero di relazioni causali individuate dal test Close, senza che nessuno di essi si riveli più utile degli altri nella previsione del prezzo di chiusura.

Per quello che riguarda i dataset utilizzati, invece, un'analisi più attenta dei risultati ottenuti rivela un pattern interessante nelle marche per cui il dataset "stock" è disponibile: Apple, Google, Nike e NovaVax. Tale pattern vede il dataset "stock" più utile a prevedere i movimenti nel breve periodo (ritardo 1), mentre quello "score" più efficace nel lungo periodo (ritardi più elevati). Sintetizziamo le informazioni nella *tabella 6*:

Tabella 6: confronto tra le capacità previsive del dataset score e stock, nelle marche in cui quest'ultimo è disponibile, al variare dei ritardi

marca	*dataset*	*ritardo*	
AAPL	stock	1	2 relazioni *
GOOGL	stock	1 e 2	2 relazioni * | 2 relazioni **
NKE	stock	1	1 relazione * | 1 relazione **
GOOGL	score	4 e 5	3 relazioni * | 7 relazioni **
NKE	score	2, 4 e 5	7 relazioni *
NVAX	score	5	3 relazioni * | 1 relazione **

Sembra perciò, secondo il test Close, che le polarità dei giudizi che gli utenti di Twitter esprimono direttamente sul valore dell'azienda in borsa (dataset "stock") causino, in parte, il prezzo di chiusura del giorno stesso, mentre le valutazioni che gli utenti esprimono sull'azienda in generale causino movimenti con ritardi di 4 o 5 giorni.

Infine, con sola eccezione di Nestlé, tutte le analisi dei tweet delle marche del campione analizzato hanno rivelato una capacità previsiva (per alcune più debole, per altre più forte) sul prezzo delle azioni della marca stessa alla chiusura della borsa americana. Questo è un segnale positivo, che suggerisce una possibile efficacia dell'implemento della sentiment analysis nei Sistemi di Trading Automatico.

BIBLIOGRAFIA

[1] Walczak,S. (2001); An empirical analysis of data requirements for financial forecasting with neural networks; Journal of Management Information Systems, 17(4), 203–222.

[2] Bollen J., Mao H., & Zeng X. (2011); Twitter mood predicts the stock market; Journal of Computational Science, 2(1), 1–8.

[3] Gruhl, D, Guha, R, Kumar, R, Novak, J, & Tomkins, A. (2005) The predictive power of online chatter. (ACM, New York, NY, USA), pp. 78–87

[4] Mishne, G & Glance, N. (2006) Predicting Movie Sales from Blogger Sentiment. AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs

[5] Twitter API - https://developer.twitter.com/en/docs/twitter-api

[6] Package “rtweet” - https://cran.r-project.org/web/packages/rtweet/rtweet.pdf

[7] Package “udpipe” - https://cran.r-project.org/web/packages/udpipe/udpipe.pdf

[8] Package “tm” - https://cran.r-project.org/web/packages/tm/tm.pdf

[9] T. Wilson, J. Wiebe, P. Homann (2005); Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis – MPQA subj lexicon: https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

[10] Package “Syuzhet”- https://cran.r-project.org/web/packages/syuzhet/syuzhet.pdf

[11] Package “car” - https://cran.r-project.org/web/packages/car/car.pdf

[12] Package “lmtest” - https://cran.r-project.org/web/packages/lmtest/lmtest.pdf