FRONTESPIZIO

[0. Introduzione 2](#_Toc138780815)

[1. Capitolo 1: Analisi dei dati 3](#_Toc138780816)

[1.1. Descrizione generale dei dati 3](#_Toc138780817)

[1.2. Analisi del target 5](#_Toc138780818)

[1.3. Analisi delle feature contestuali 6](#_Toc138780819)

[1.4. Preparazione dei dati 7](#_Toc138780820)

1. Introduzione

Per questo progetto di Data Analytics affronterò il problema espresso nella traccia 3, ovvero la rilevazione del sarcasmo. Questo problema è uno dei più interessanti nell’ambito di NLP (Natural Language Processing), per la sua complessità; causata della sua natura soggettiva e delle sfumature linguistiche coinvolte; infatti, spesso non è semplicissimo neanche per un essere umano. In particolare, il sistema dovrà esaminare un testo, e aiutandosi con degli elementi esterni (il contesto), dovrà capire se esso è sarcastico o no.

Nella ricerca sono state adottate diverse strategie per questo task, divisibili in due categorie:

* Basati sulle caratteristiche linguistiche: in cui si analizzano i testi alla ricerca di pattern ligustici legati al sarcasmo; come l’uso di più parole contradditorie nella stessa frase, o rispetto al contesto, l’uso di contrasti tra parole con connotazione positiva e negativa oppure l’uso di esagerazioni.
* Basati su modelli di Machine Learning o Deep Learning: in cui si sfruttano tecniche di apprendimento supervisionato (o semi) per addestrare modelli statistici con dati etichettati per rilevare i testi sarcastici; generalmente i modelli posso essere più “semplici” come SVM, Random Forest Naive Bayes, o piccole reti neurali, fino ad arrivare ai modelli di Deep Learning, come RNN (Recurrent Neural Network) o i Transfomer che possono raggiungere e superare i milioni di parametri.

In questo progetto, la strategia risolutiva adottata si baserà sul training supervisionato con i Transformer; in quanto sono un modello che sta rivoluzionando il settore, e sarà interessante applicarli in primo piano (seppur in una scala ridotta) a questo difficile task.

La relazione sarà divisa in tre capitoli principali: nel primo descriverò l’analisi dei dati effettuata, da cui si baseranno le scelte adottate; nel secondo tratterò il modello usato e il relativo addestramento; nel terzo discuterò i risultati prodotti.

1. Capitolo 1: Analisi dei dati

In questo capitolo descriverò l’analisi dei dati di addestramento, esaminando gli elementi del dataset, il target della classificazione, i testi e gli elementi del contesto. Con cui poter trarre delle conclusioni per poter processare i dati e addestrare il modello con una maggior accortezza.

* 1. Descrizione generale dei dati
     1. Introduzione ai dataset e Splitting

Alla traccia, sono associati due dataset, uno con i dati di addestramento (train) e uno per il testing. Essi sono due insiemi di commenti di Reddit dal 2009 al 2016; in particolare, per ogni istanza abbiamo:

* Text: Testo del commento da classificare;
* Sarcastic: Label booleana che indica se il commento è sarcastico (True) o no (False);
* Author: Nome dell’autore del commento;
* Subreddit: Nome del subreddit in cui è stato pubblicato il commento;
* Date: Data di pubblicazione del commento;
* Parent: Testo della discussione a cui il commento si riferisce;

Il dataset di training originale, con un milione di righe (precisamente 1010822), sarà diviso casualmente in due parti: quello di training definitivo (con il 95%), su cui verranno effettuate tutte le analisi e assunzioni; quello di validation (con il restante 5%), usato durante il processo di training per misurare le performance del modello.

Il dataset di training è molto completo, perché il numero di righe duplicate e non valide è relativamente irrilevante (rispettivamente 200 e 53); dunque non è necessario attuare alcuna strategia particolare per recuperare i dati mancanti o nulli (come l’imputazione), e si può procedere con la loro rimozione. Inoltre, sia la codifica dei testi che delle date rispetta gli standard e non sono dunque presenti errori di conversione.

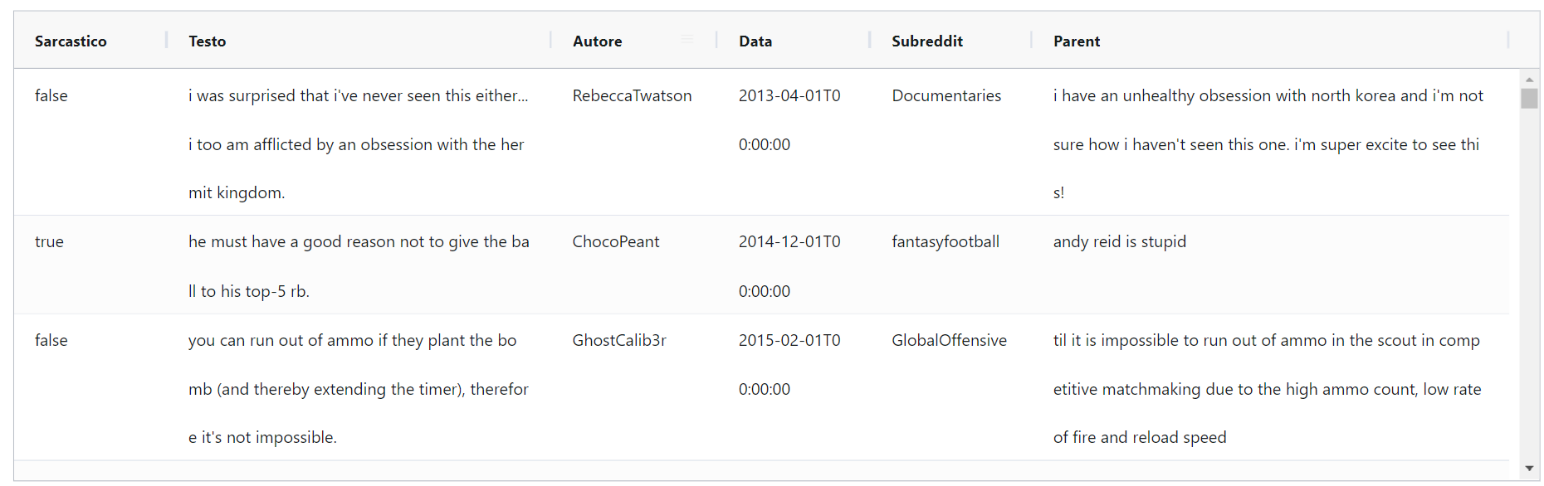


Figura .1: Alcune righe di training di esempio

* + 1. Descrizione della metodologia di analisi

Ogni riga è composta da una label e dalla relativa istanza, i cui attributi possono essere divisi nel testo fondamentale da classificare (l’attributo Text), e una serie di attributi contestuali che aggiungono informazioni per il testo (i restanti attributi).

Vista questa divisione ho strutturato il processo di analisi in due fasi:

* Nella prima fase esaminerò gli attributi del contesto, calcolando per ogni elemento unico di ogni attributo qual è il rateo con cui compare in testi sarcastici e non (detto Rateo Informativo), in modo da individuare gli elementi (e quindi i relativi attributi) più rilevanti ad un modello.
* Nella seconda fase esaminerò i diversi modi processare il testo, calcolando quale sistema produce il testo con un maggior contento di rateo informativo.

Nel dettaglio il rateo informativo di un elemento unico ( verrà calcolato come:

Dove: e sono rispettivamente il numero di commenti sarcastici e non sarcastici associati ad e; mentre, è la probabilità della classe più probabile.

* 1. Analisi del target

Il target, ovvero l’attributo Sarcastic, è la Label che identifica la natura del commento, e sarà dunque l’obbiettivo del modello di NLP.

Per la buona riuscita del training è importante che la sua distribuzione sia bilanciata, ovvero che il numero di istanze sarcastiche e non sia approssimativamente simile. Come sostenuto nell’articolo “A Large Self-Annotated Corpus for Sarcasm” (allegato alla traccia), e dopo una fase di verifica, posso affermare che il dataset presenta questa caratteristica e quindi non è necessario applicare tecniche per migliorare il bilanciamento.

Immagine che contiene testo, schermata, diagramma

Descrizione generata automaticamente

Figura 1.: Distribuzione del target ottenuta dopo la fase di verifica, come si può notare il bilanciamento è quasi perfetto

Inoltre, questa analisi mi è servita per calcolare con precisione .

* 1. Analisi delle feature contestuali

Come affermato precedentemente si definiscono tali: Parent, Date, Subreddit e Author. Esse potrebbero essere utili per la predizione aggiungendo delle informazioni di contesto al commento. Nel dettaglio:

* Il Parent: codificata come una stringa a più parole, rappresenta il testo della discussione a cui il commento è stato lasciato; potrebbe essere molto utile per rilevare significati contrastanti con il testo (caratteristica principale dell’ironia), oppure per individuare discussioni polarizzate sarcastiche o no.
* Data: codificata come una stringa “anno-mese”, potrebbe avere rilevanza nel trovare date altamente sarcastiche (o no).
* Subreddit: codificata come una stringa ad una parola, rappresenta il nome del subreddit in cui è stato postato il commento; potrebbe essere utile per trovare subreddit tendenzialmente più o meno sarcastici.
* Author: codificata come una stringa ad una parola, rappresenta il nome dell’autore del commento; potrebbe essere utile per trovare autori più o meno sarcastici.

A partire da queste premesse ho effettuato un’analisi per avere un reale riscontro sui dati.

* + 1. Analisi del contesto rispetto al numero di commenti

In questa fase ho analizzato per ogni feature quanti sono gli elementi che si ripetono (prendendoli una sola volta) e ad ognuno quanti commenti sono associati.

Immagine che contiene testo, schermata, diagramma, linea

Descrizione generata automaticamente

Figura 1.: Distribuzione, tramite boxplot, degli elementi unici del contesto rispetto al numero di commenti associati

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Figura 1.: Tabella mostrante il numero di elementi unici (e che si ripetono quindi in più commenti), e il numero di testi associati

Da Figura 1.2 e Figura 1.3, si nota particolarmente che:

* Quasi ogni commento si riferisce ad un Parent unico; infatti, nel boxplot la distribuzione è “schiacciata” verso l’1 (il 1°, 2° e 3° quartili sono a 1), e nella tabella il 97.3% dei Parent si ripete almeno una volta, di cui il 98.4% si ripete esattamente una volta.
* Le date, come atteso essendo solo 96, si ripetono con un’alta frequenza nei commenti; infatti, nel boxplot i valori si distribuiscono principalmente in un range dai 668 a 16000, con mediana a 3108.
* La feature autore potrebbe essere rilevante; in quanto il numero di autori unici rispetto ai commenti è del 26.7% di cui solo il 7% ha un solo commento, come mostrato nella tabella; il boxplot conferma la tesi, infatti, la distribuzione è posta nel range 1, 4
* La feature subreddit, con una distribuzione nel range 1, 9 e un numero di elementi unici a 1.5% (14545 rispetto ai 960000), risulta ancora più interessante; vale la pena notare però che il 40.6% di essi ha un solo commento.
  + 1. Analisi del rateo informativo del contesto singolarmente

Dopo aver analizzato come si distribuiscono gli elementi unici del contesto, procederò con l’analisi del contenuto informativo di quegli elementi che hanno almeno più di un commento associato. Approfondendo ogni feature separatamente, tramite degli istogrammi che presentano, per un range di commenti associati, la media del rateo.

Immagine che contiene testo, schermata, linea, Rettangolo

Descrizione generata automaticamente

Figura 1.: Rateo informativo per gli elementi di Autore.

Immagine che contiene schermata, diagramma, Diagramma, testo

Descrizione generata automaticamente

Figura 1.: Rateo informativo per gli elementi di Autore. Con y scalata

Come si può notare in Figura 1.4 e Figura 1.5 il rateo informativo di autore è molto basso; infatti, è necessario scalare l’asse y tra 0 e 1.8 per poter distinguere gli istogrammi. Ne consegue che non possiamo distinguere molti autori a maggioranza sarcastica o no, e quindi che questa feature potrebbe essere irrilevante.

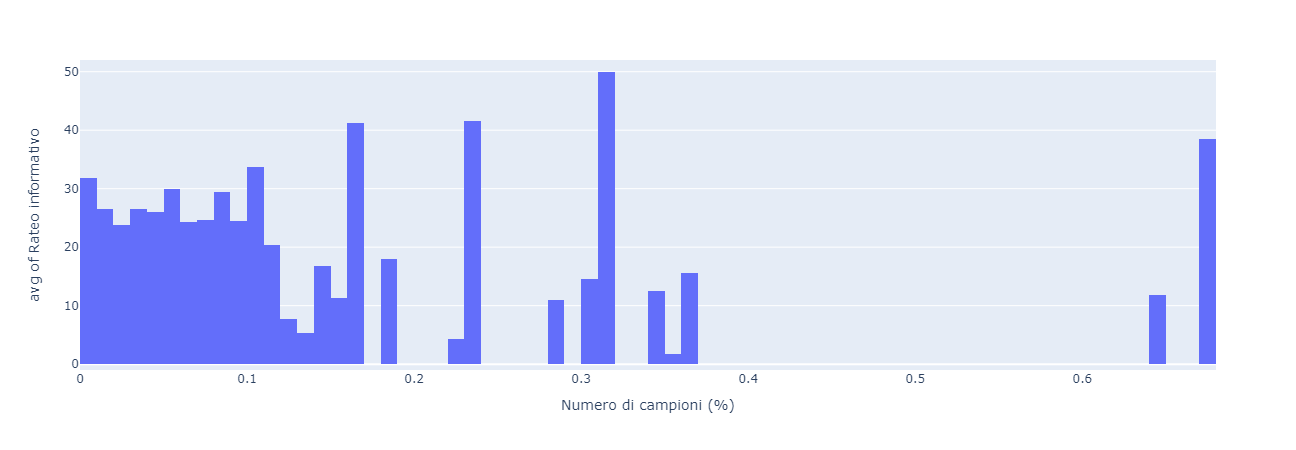


Figura 1.: Rateo informativo per gli elementi di Parent

Dalla Figura 1.6: Notiamo che il rateo informativo per gli elementi di Parent è alto; tuttavia, come si vede nell’asse x (e come conferma la Figura 1.3) il numero di commenti associato è basso.

Immagine che contiene testo, schermata, linea, Diagramma

Descrizione generata automaticamente

Figura 1.: Rateo informativo per gli elementi di date

Come si può notare dalla Figura 1.7 il rateo informativo è abbastanza scarso, anche se il numero di campioni per elemento è abbastanza alto.

Immagine che contiene schermata, linea, Diagramma, diagramma

Descrizione generata automaticamente

Figura 1.: Rateo informativo per gli elementi di Subreddit

Dalla Figura 1.8 possiamo notare che il rateo informativo per Subreddit è decisamente migliore rispetto a quello di date, raggiungendo e superando anche il 10% in molti casi; come atteso, notiamo però, che il rateo cala all’aumentare del numero di commenti.

* + 1. Analisi aggregata del rateo informativo

Dopo aver visto le distribuzioni singolarmente del rateo informativo, è utile aggregare i risultati in modo da poterli confrontare in modo efficiente. Per farlo ho seguito due approcci:

* Threshold: in cui ho calcolato la percentuale degli elementi unici che supera delle soglie (10%, 20%, 30%, 40%) di rateo informativo.
* Media: in cui ho calcolato la media e deviazione standard pesate sul numero di commenti associati.

Immagine che contiene schermata, Policromia, diagramma, Diagramma

Descrizione generata automaticamente

Figura 1.: Threshold: Percentuale di elementi unici che supera determinate soglie, per ogni feature del contesto

Figura 1.: Media e STD pesate del rateo informativo per i tipi di contesto

Immagine che contiene schermata, diagramma, testo, Diagramma

Descrizione generata automaticamente

Possiamo notare da Figura 1.9 e Figura 1.10:

* Data e Autore hanno un basso rateo informativo medio, e pochi superano la soglia del 10%.
* Parent ha il rateo informativo medio più alto, e quasi tutti quelli che superano la soglia del 10%, superano anche quella del 40%; questo, significa che i pochi elementi ripetuti sono molto significativi.
* Subreddit ha un rateo informativo accettabile, con ben il 64% degli elementi che supera il 10% e il 22% che supera la soglia del 40%.
  + 1. Conclusioni delle analisi sul contesto

Unendo le analisi fatte precedentemente possiamo giungere a delle conclusioni utili per il modello:

* Autore presenta non troppi elementi unici (associati quindi ad abbastanza commenti), tuttavia essi non hanno abbastanza rateo informativo; perciò, introdurrebbero troppo rumore nell’addestramento del modello.
* Data come autore, ha davvero pochi elementi unici ma questi non hanno un rateo informativo significate; perciò, introdurrebbe anche lui troppo rumore, e sarebbe totalmente inutile con commenti in un periodo temporale successivo.
* Parent al contrario ha un alto rateo informativo, ma il numero di elementi unici è pari quasi al numero di commenti; quindi, l’uso dell’intera frase in modo statistico non avrebbe alcun senso, però potrebbero essere utili le associazioni delle parole fra di loro o con il testo del commento.
* Subreddit presenta un buon rateo informativo e un numero adeguato di elementi unici; perciò, potrebbe essere una feature utile al modello.

In conclusione, la feature Subreddit verrà usata dal modello, Data e Autore non verranno usate e Parent verrà usata solo in legame con il testo.

* 1. Analisi del testo

Per l’analisi del testo del commento ho adottato una logica simile a quella per il contesto, ma al posto di paragonare le diverse feature, ho confrontato i vari modi di elaborare il testo.

Ho fatto tale analisi, con quattro grafici prendendo singolarmente ogni tipo di testo:

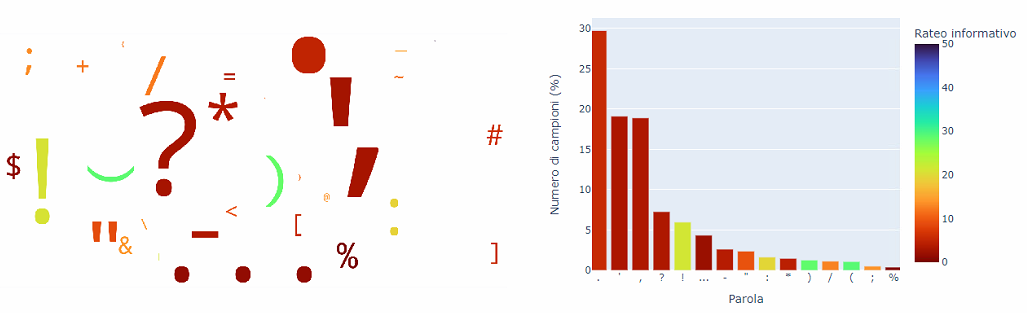
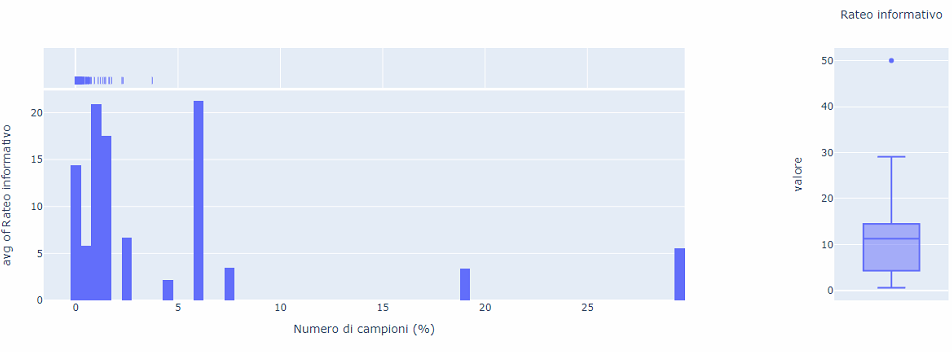
* Istogramma: come quello usato nell’analisi del contesto, mostra il rateo informativo associato ad un range di frequenze dei token nei commenti.
* Boxplot: mostra in sintesi la distribuzione del rateo informativo.
* Wordcloud: Grafico che mostra i 200 token più frequenti; la grandezza indica la frequenza, mentre il colore indica il rateo informativo.
* Barplot: mostra in ordine i token più frequenti e il loro rateo informativo.

I vari tipi di testo, elaborati ed analizzati, sono:

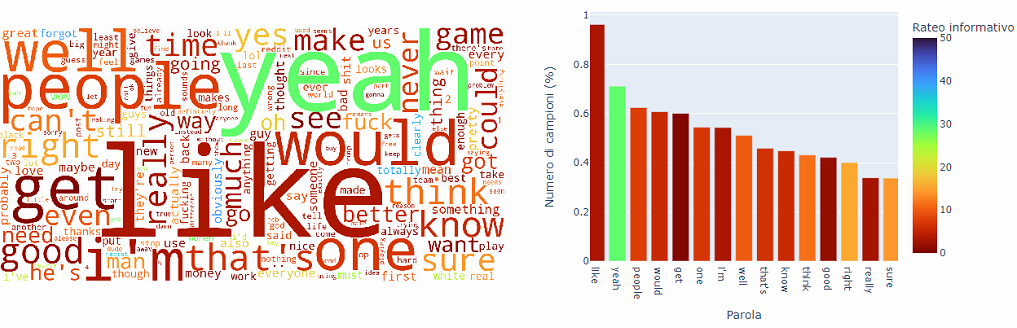
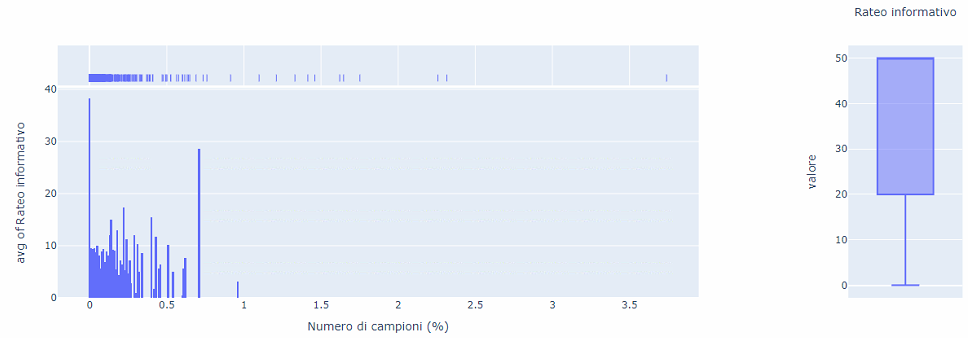
* Normale
* Senza stopwords: a cui sono state rimosse le parole non chiave; come articoli, congiunzioni, ecc...
* Con stemming: in cui le parole derivate e flesse vengono ridotte alla loro forma radice.
* Senza stopwords e con stemming: in cui si applicano simultaneamente i due approcci precedenti.

Però prima dell’analisi è necessario tokenizzare il testo, ovvero convertirlo da una stringa unica in una lista di parole (token); per poi rimuovere la punteggiatura, in quanto spesso non è significativa e comporta solo del rumore. Tuttavia, al posto di rimuovere tutta la punteggiatura ho effettuato l’analisi del rateo informativo dei punti per mantenere quelli con un rateo significativo, in modo da non avere eccessive perdite di informazione.

Figura 1.: Distribuzione del rateo per i simboli punteggiatura



Come si può osservare dalla Figura 1.11 l’unico segno di punteggiatura significativo è il “!”; perché presenta un rateo di 21 (infatti le frasi con almeno un “!” sono sarcastiche il 71% delle volte), con una frequenza circa del 6% (corrisponde alla barra più alta nell’istogramma). Dunque, ho deciso di mantenere questo punto e rimuovere gli altri, per poi procedere all’analisi singola dei tipi di testo.



* 1. Preparazione dei dati