Modelli di Text Mining: uno studio comparativo.

GIOVANNI DE FEUDIS[1], **ADELE ZANFINO**[2], **AND NATALIYA ZAYEVA**[3]

[1] *820602 CdLM Data Science, Università degli Studi di Milano-Bicocca, g.defeudis1@campus.unimib.it*
[2] *867496 CdLM Data Science, Università degli Studi di Milano-Bicocca, a.zanfino@campus.unimib.it*
[3] *867981 CdLM Data Science, Università degli Studi di Milano-Bicocca, n.zayeva@campus.unimib.it*

Compiled April 6, 2022

In questo lavoro vengono esplorate le potenzialità del Text Mining attraverso tecniche di Classificazione, Clustering e Topic Modeling. Nelle prime fasi del lavoro si attuano delle tecniche di preprocessing e di pulizia dei dati, seguite da diverse rappresentazioni del testo. Lo scopo della classificazione è quello di carpire dalle recensioni testuali il grado di soddisfazione dei clienti, che può essere positivo o negativo. Analogamente il clustering suddivide le recensioni in cluster differenti. Mentre con la Topic Modeling proveremo ad estrapolare e comprendere le strutture semantiche che compongono i documenti. Infine verrà proposta una valutazione del modello di rappresentazione Doc2Vec attraverso una Topic Modeling.

CONTENTS

1. INTRODUZIONE

L'azienda Amazon fu fondata nel 1994 da Jeff Bezos con il nome di Cadabra.com ('Amazon', Wikipedia). Il sito, poi rinominato con il nome che oggi tutti conosciamo, era stato pensato come semplice libreria online, per poi differenziare l'offerta arrivando a vendere i prodotti più disparati, come elettrodomestici, supporti elettronici, vestiti e cibo.
Dalla fondazione a oggi, Amazon ha visto una forte crescita del valore di mercato delle proprie azioni e del fatturato, arrivando ad essere una delle aziende più importanti del pianeta, presente in moltissime nazioni.
Questo è avvenuto, oltre che per la differenziazione dell'offerta, grazie all'aggiunta di servizi come Amazon Prime Video o con il lancio dell'assistente personale intelligente Alexa e alla gestione dei dati dei consumatori, utilizzati per gestire meglio la pubblicità sul sito e i consigli di acquisto.
Riguardo a questo, una delle funzionalità del sito permette agli acquirenti di lasciare recensioni dei prodotti, così i venditori affidabili e i prodotti di buona qualità hanno il giusto riconoscimento, mentre chi vende prodotti scadenti viene individuato più facilmente.
Gli utenti Amazon, dopo aver completato l'acquisto, possono valutare il prodotto con un voto da 1 a 5 stelle, con la possibilità di aggiungere un commento personale.

1.1. Obiettivo

L'obiettivo di questo progetto era utilizzare il dataset delle recensioni dei prodotti del sito di e-commerce Amazon, al fine di trovare un classificatore che sia in grado di eseguire un task di classificazione binaria. Nel nostro caso, la finalità era riuscire a distinguere le recensioni in positive e negative, in modo tale da notare una corrispondenza tra il voto fornito al prodotto e la recensione testuale dell'utente.
Successivamente abbiamo confrontato vari metodi di rappresentazione all'interno di un task di clusterizzazione su due cluster. Quindi abbiamo provato ad estrapolare informazioni interne ai documenti attraverso tecniche di topic modeling ed infine abbiamo provato a valutare una rappresentazione Doc2Vec attraverso un metodo definibile euristico utilizzando una tecnica di topic modeling.

2. DATASET

Il dataset utilizzato nell'analisi consiste in 400.000 recensioni dei clienti Amazon e nel rating a stelle, ed è composto da due colonne:

- Label: composto da due classi **label 1** e **label 2**. Label 1 fa riferimento alle recensioni con 1 e 2 stelle, quindi considerate come recensioni negative. Invece label 2 fa riferimento a recensioni con 4 e 5 stelle, recensioni positive. Le recensioni con 3 stelle non vengono prese in considerazione fin dall'inizio;
- Text: le recensioni dei clienti.

In Figura 4 si osserva la distribuzione delle recensioni negative e positive nel dataset, che sono equilibrate tra le due classi.

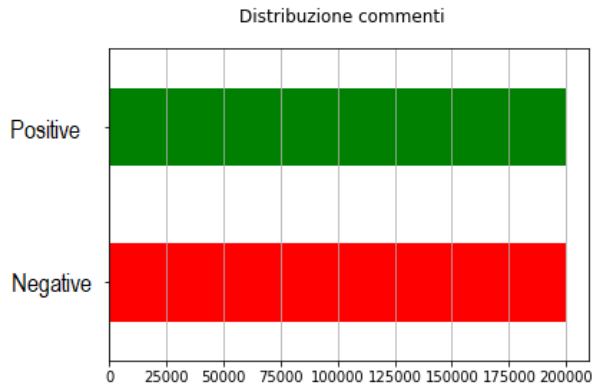

Fig. 1. Bar chart con la distribuzione delle recensioni

3. ASPETTI METODOLOGICI

3.1. Pre-processing

Uno dei passaggi preliminari più importanti nel contesto della classificazione testuale e, in generale, nell'elaborazione del linguaggio naturale (NLP) consiste nella pre-elaborazione del testo, utilizzata al fine di trasformare il testo in una forma più comprensibile per gli algoritmi di Machine Learning che verranno usati in seguito. La fase di pre-processing consiste, quindi, nella pulizia di termini molto utilizzati ma semanticamente inconsistenti, per esempio dalle congiunzioni e dagli avverbi e da parole generalmente considerate senza significato per l'analisi. Le prime operazioni che sono state effettuate sui dati si sono concentrate sulla suddetta pulizia del testo effettuando operazioni come la rimozione dei numeri, dei segni di punteggiatura e delle cosiddette "stopwords". Di seguito illustriamo le fasi successive del processo:

1. Normalizzazione: con questo termine si intende rendere le parole uniformi a livello di accenti, case folder, punteggiatura ed errori di battitura. Abbiamo rimosso caratteri speciali e punteggiatura, poiché non sono utili a carpire il grado di soddisfazione di una determinata recensione. Inoltre, tutto il testo è stato trasformato in minuscolo, per evitare che il computer tratti due parole identiche come diverse.

2. Stopword: l'espressione stopwords indica parole che, salvo in rari casi, non hanno un vero e proprio significato se isolate dal testo che le contengono. Il vantaggio di questo compito è di ridurre ampiamente la dimensionalità del dataset, pur mantenendo intatta la sua informatività.

3. Tokenization: un token è una sequenza di caratteri che formano una parola contenente significato dopo diverse operazioni preprocessing. L'obiettivo della tokenizzazione è quello di dividere un flusso di testo in unità dense di significato. Abbiamo effettuato questa procedura tramite la libreria nltk e la funzione word_tokenize;

4. Stemming: questa operazione viene definita come il processo di riduzione della forma flessa di una parola alla sua forma radice. Per questa fase ci siamo avvalsi della funzione Porter Stemmer contenuta anch'essa nella libreria nltk.

5. Lemmatization: questa operazione viene definita come il il processo di riduzione di una forma flessa di una parola alla sua forma canonica, detta lemma. Per questa fase è stata utilizzata la funzione Lancaster Stemmer contenuta nella libreria nltk.

Dopo questa fase preliminare, abbiamo applicato alcune tecniche di rappresentazione del testo

3.2. Rappresentazione del testo

La rappresentazione del testo è uno dei problemi fondamentali della Text Mining. Lo scopo è quello di rappresentare numericamente i documenti di testo al fine di adattarli al lavoro successivo compiuto dagli algoritmi.
È stata fatta una prima analisi esplicativa per evidenziare le parole più frequenti nelle recensioni, per ottenere una più facile e immediata comprensione (Figura 2).

Fig. 2. Wordcloud con le parole più frequenti

In questa fase, il primo passo è stato quello di decidere tra una serie di diversi modelli di rappresentazione quello che più si adattava al nostro tipo di analisi. Due modelli sono stati presi in considerazione.
La prima rappresentazione scelta è stata la *Bag-of-Word*. Esso rappresenta il modello più comune, ed è costituito da una tabella in cui vengono riportate le parole contenute nei testi ignorando l'ordine delle stesse. Il nome è stato scelto in quanto ogni documento viene considerato analogo ad una borsa contenete delle

parole, ciò consente una gestione delle parole basata sulle liste. Se la tabella riporta semplicemente la presenza/assenza delle parole essa è detta Binary term document matrix; nel caso in cui le parole vengono conteggiate all'interno di ogni testo si ha una term-document count matrix. A causa dell'eterogeneità del corpus di recensioni e della sua elevata dimensionalità, il numero di parole uniche è davvero elevato. Abbiamo quindi scelto di utilizzare un secondo modello, il modello *TF-IDF*, che tiene conto dell'informazione legata alla frequenza con cui compaiono i termini. In Figura 3 vengono rappresentate le 30 parole più frequenti usando *TF-IDF*.

Fig. 3. Top 30 tokens più frequenti nelle recensioni

In particolare, la funzione utilizzata per il calcolo della matrice permette di eliminare le parole troppo ricorrenti e troppo rare, ovvero quelle che si trovano ai lati della curva di Zipf.
Nei testi, infatti, la relazione tra la frequenza di una parola e il suo rango (ovvero l'ordine all'interno del testo) segue una legge di proporzionalità inversa: poche parole compaiono spesso, mentre la maggior parte dei lemmi appare più di rado. In pratica, in questo modo si esplicita il fatto che le "head words" sono molto ricorrenti ma non hanno significato, mentre le "tail words" prendono la porzione maggiore del vocabolario, ma occorrono raramente nei documenti.
Ne consegue che non tutte le parole in un testo descrivono il contenuto dello stesso con la medesima informatività. Questo viene analizzato da Luhn che ha notato che frequenza e posizione delle parole forniscono un'importante indicazione sul significato delle parole stesso. L'abilità delle parole nel discriminare il contenuto in un documento è massima nella posizione intermedia tra i due livelli di cut-off, che rappresentano da un lato (upper cut-off) le parole comuni, e dall'altro (lower cut-off) le parole rare.

Basandosi sull'analisi di Luhn sono stati proposti due fattori da identificare per assegnare dei pesi alle parole, corpus-wise e document-wise.
Per misurare questi fattori vengono utilizzate due euristiche: TF, che rappresenta il numero delle volte che il termine t occorre nel documento d e IDF, che rappresenta il numero di documenti d in cui appare il termine t.
Queste due euristiche possono essere combinate nella *TF-IDF*, utilizzato nel mostro modello di rappresentazione.
Di seguito riportiamo la formula.

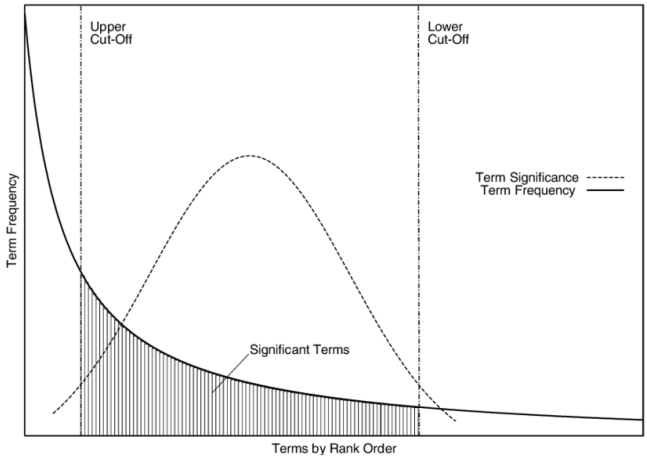

Fig. 4. Analisi di Luhn

$$w_t, d = \frac{tf_{t,d}}{max_{t_i \in d} tf_{t,d}} * log \frac{N}{df_t}.$$

TF-IDF cresce, quindi, sia quando aumenta il numero di occorrenze del temine nel documento sia aumenta la sua rarità.

3.3. Classificazione

La classificazione è l'attività volta alla predizione della classe di appartenenza dei dati in un set predefinito di classi.
Per quanto riguarda il Text Mining una definizione più formale del problema è data dalla seguente formula. Dato un set di documenti $D = \{D_1, D_2, ...\}$ pre-classificati in $C = \{C_1, C_2, ...C_n\}$ lo scopo è quello di ottenere una funzione: $h : D \rightarrow C$ che indica se un documento deve appartenere alla categoria C o non deve appartenere ad essa.
Per predire le recensioni con le etichette più appropriate, sono stati utilizzati diversi algoritmi di apprendimento supervisionato,i quali a loro volta sono stati confrontati con il fine di valutarne l'efficienza e l'efficacia. Prima di applicare i modelli il dataset è stato diviso in Training e Test set pari a 75%- 25%, per valutare l'accuratezza dei diversi modelli.
Sono stati implementati i seguenti modelli:

- Support Vector Classifier (SVC): sono modelli di apprendimento supervisionato il cui obiettivo è quello di trovare la retta di separazione delle classi che massimizza il margine tra le classi stesse. Il margine rappresenta la distanza minima dalla retta ai punti delle due classi.

- Logistic Regression (LR): è un modello di regressione non lineare utilizzato quando la variabile dipendente è binaria, è stata implementata la *Logistic Regression* della libreria Scikit-Learn.

- Decision Tree (DTs): un albero di decisione è un modello predittivo in cui ogni nodo interno rappresenta una variabile, ogni arco verso un nodo figlio rappresenta un possibile valore per quella proprietà, e ogni voglia il valore predetto partendo dai valori delle altre proprietà. Per classificare un record (nel nostro caso un testo) vengono effettuati una serie di test iniziando dal nodo radice e propagando via via il campione fino a terminare con un nodo foglia, che rappresenterà quindi l'etichetta assegnata allo specifico testo.

- Random Forest (RF): con questo metodo di apprendimento supervisionato vengono costituiti una moltitudine di alberi di decisione il cui output rappresenta l'etichetta più frequente emessa dai singoli alberi. In questo modo è possibile correggere la tendenza degli alberi di decisione a produrre un overfitting sul training set.

Gli algoritmi di classificazione necessitano di un input ben strutturato, per questo motivo i documenti devono essere convertiti in vettori in uno spazio vettoriale comune. Inoltre, considerando la dimensione della matrice td-idf, abbiamo deciso di utilizzare una feature selection (Feature Selection,sklearn) al fine di ridurre la dimesionalità del dataset.
Sui modelli è stata utilizzata la convalida incrociata (Cross Validation), per evitare i problemi di sovradattamento. Tale tecnica viene usata per alterare i dati sia del train che del test. Viene chiamata anche *K-fold cross validation* perché divide il dataset in k parti di uguale numerosità, usa $1/k$ come validation set, e la resta parte $k-1/k$ come training dataset. La procedura viene ripetuta per k volte, selezionando ogni volta un sottoinsieme diverso. È stato deciso di attribuire alle interazioni della Cross Validation *K=5*.

3.4. Clusterizzazione del Testo

Il processo di clusterizzazione permette attraverso tecniche di apprendimento non supervisionato la suddivisione di documenti in classi simili. La finalità è quindi quella di ottenere gruppi in cui gli oggetti siano simili tra loro e allo stesso tempo, oggetti appartenenti a gruppi differenti devono essere il più dissimili possibili. Il primo passo per procedere alla clusterizzazione è quindi quello di domandarsi come rappresentare i documenti che risiedono all'interno del dataset. Sono stati selezionati quattro tipologie differenti:

- Rappresentazione matriciale sparsa attraverso il conteggio dei token;
- Matrice TF-IDF;
- Truncated Singular value decomposition;
- Rappresentazione tramite Doc2vec.

Il passo successivo è stato quello di selezionare l'algoritmo di clustering. Si è optato per un algoritmo della famiglia Flat, cioè un *k-means* . L'algoritmo di K-means iterativamente cerca di minimizzare la varianza totale intra-gruppo ('K-means' (2018) Wikipedia), dove il numero dei gruppi, *k*, è un iperparamentro da selezionare. Avendo le etichette per ogni recensione abbiamo deciso di testare il modello e osservare come questo performasse su questo task. Quindi per ogni tipologia di rappresentazione è stata valutata la performance sulla clusterizzazione.

3.5. Topic Modeling

La Topic Modeling è una tecnica non supervisionata di machine learning che ha l'obiettivo di carpire da documenti e dalle parole che questi documenti contengono una serie di n topic ('Topic model' (2017) Wikipedia) . Questo consente di scoprire delle strutture semantiche all'interno del testo, e nel nostro caso ci permetterà di osservare quali gruppi di topic sono presenti all'interno dei documenti e valutare i risultati. Successivamente verrà testato il modello Doc2Vec cercando di estrapolare uno specifico argomento che poi verrà valutato attraverso una topic modeling.
Per realizzare la topic modeling verrà utilizzato *LDA* (Latent Dirichlet allocation), un modello statistico generativo. E' stato necessario costruire un dizionario, quindi a questo è stato applicato il modello LDA. Per valutare il numero ottimale di topic è stata utilizzata una metrica di *Intrinsic Evaluation Metrics*, la *Coherence*, che misura la *similarità semantica* tra le parole presenti all'interno di un topic. Attraverso una procedura di *Grid Search* è stato selezionato il numero di topic migliore su un subset dei dati disponibili. Il modello migliore è stato poi valutato utilizzando altre metriche:

- *Coherence c_v* che può variare tra 0 a 1;
- *Coherence u_mass* che può variare tra -14 a 14;
- *log perplexity*.

Successivamente i risultati sono stati valutati attraverso un metodo di *Eye Balling Models*, in cui i risultati vengono valutati direttamente dal ricercatore. Volendo infine valutare euristicamente il modello Doc2Vec, utilizzando la funzione di vettorizzazione presente al suo interno, si è deciso di provare ad estrapolare da una frase, gli eventuali topic ad essa associata e successivamente valutarne il risultato mediante la realizzazione di una *Topic Modeling*. Quindi è stata vettorizzata una frase contenente alcuni argomenti e successivamente sono state selezionate le 100 frasi più simili. Quindi è stato applicato un Topic Modeling e i risultati sono stati valutati mediante *Eye Balling Models*.

4. RISULTATI

4.1. Risultati Classificazione

I risultati che si sono ottenuti con la classificazione sono i seguenti:

	Mean Accuracy	**Standard Deviation**
SVC	0.896396	0.000614
LR	0.896899	0.000482
DTs	0.761322	0.001914
RF	0.861502	0.001104

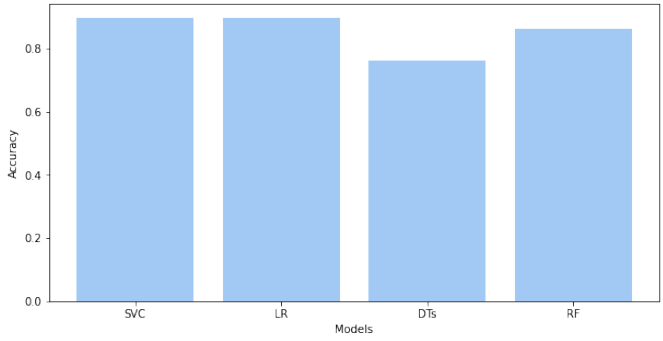

Fig. 5. Accuracy dei modelli di classificazione

Come si può osservare il modello migliore per la classificazione risultata essere il *Logistic Regression (LR)*

4.2. Risultati Clusterizzazione

Di seguito possiamo osservare i risultati ottenuti mediante la clusterizzazione:

	Score
CountVect	0.5001
Tfidf	0.5004
Tsdv	0.5004
Doc2Vec	0.5007

Le performance su due cluster sono veramente molto basse, sicuramente il modello non è in grado di estrapolare informazioni sfruttabili. Si potrebbero avere dei piccoli miglioramenti se si provassero a considerare più di due cluster, ma in questo progetto non verranno presi in considerazione.

4.3. Risultati Topic Modeling e Doc2Vec

La metrica utilizzata per valutare il miglior numero di Topic è stata la *coherence c_v*. Ricordando che la *c_v* può variare tra 0 e 1, dove 1 è il miglior risultato ottenibile. Di seguito possiamo osservare i risultati:

Num. Topic	Score
2	0.405
3	0.433
4	0.348
5	0.436
6	0.466
7	0.446
8	0.473
9	0.450

E' stato quindi selezionato il Modello con 8 Topic, di seguito possiamo osservare alcune metriche di valutazione dello stesso:

Evaluation	Score
Coherence c_m	0.503
Coherence u_mass	-2.486
Log_Perplexity	-8.017

Faremo ora uno zoom sui vari topic presenti e valuteremo i risultati[fig.6].
Ogni gruppo sembra poter descrivere un topic. Il *topic 0* sembra descrivere recensioni che parlano di musica. Il *Topic 1* sembra descrivere un topic più astratto, comunque inerenti ad aspetti contrastanti all'interno di alcune recensioni. Il *Topic 2* descrive argomenti vicini ai libri. Il *Topic 3* sembra descrivere un topic astratto, non facilmente comprensibile, dove però compare la parola libro come molto Importante. Il *Topic 4* sembrerebbe descrivere recensioni positive, riguardanti argomenti vicini a prodotti per bambini. Il *Topic 5* descrive l'argomento Film. Il *Topic 6* sembra descrivere delle recensioni positive. Infine il *Topic 7* non sembra avere caratteristiche definibili.
Per valutare il modello di rappresentazione Doc2Vec abbiamo

Fig. 6. Word Clouds di 10 Keywords in ogni topic

deciso di vettorizzare questa frase: "*really love dog animal food is not very good animals house*", quindi abbiamo selezionato i primi 100 documenti più simili, e successivamente è stato applicato un modello Topic Modeling. Dopo aver applicato un metodo di *Search Grid* sono stati selezionati 5 topic. Nella Figura 7 possiamo osservare quali sono stati gli argomenti estratti dal modello.

Fig. 7. Word Clouds di 10 Keywords, valutazione Doc2Vec

Il risultato ottenuto sicuramente si avvicina ai topic che potrebbero venir estratti dalla frase di partenza. Possiamo quindi ritenerci soddisfatti del risultato ottenuto tenendo conto che altre prove potrebbero venir realizzate.

5. CONCLUSIONI E SVILUPPI FUTURI

Lo scopo di questo lavoro era quello di evidenziare le potenzialità del text mining e delle tecniche di Machine Learning applicate ad esso. Nello specifico, selezionare in primo luogo un classificatore che riuscisse ad eseguire un task di classificazione binaria utilizzando il testo delle recensioni di un prodotto Amazon.
Come possiamo osservare dalla Figura 5, il modello migliore per la classificazione risulta essere il Logistic Regression. Mentre lo sviluppo dei modelli di classificazione è stato relativamente semplice, uno dei maggiori ostacoli riscontrati è stato gestire gli elevati costi computazionali rispetto alla dimensione del dataset da noi utilizzato.
In secondo luogo, è stato attuato un processo di clusterizzazione al fine di ottenere gruppi in cui le recensioni siano simili tra loro, e per attuare questo lavoro abbiamo scelto un algoritmo di K-mean. Successivamente, abbiamo applicato al suddetto dataset la Topic Modeling, anch'essa una tecnica di apprendimento non supervisionata con lo scopo di capire dai documenti e dalle parole una serie n di topic. Dopo aver creato un dizionario, e aver applicato l'LDA, attraverso una procedura di Grid Search è stato selezionato il numero di topic migliore su un sottoinsieme dei dati disponibili. Come è possibile osservare, i risultati delle performance su due cluster sono molto bassi, inducendoci a pensare che il modello non è in grado di dedurre informazioni sfruttabili.
Per quanto riguarda il Topic Modeling, è stato selezionato un modello con 8 Topic, che invece ha dato dei risultati promettenti.

Tra gli sviluppi futuri, potrebbero essere presi in considerazione modelli su cui testare questi obiettivi per quanto riguarda sia la classificazione che la clusterizzazione. Per quest'ultima si potrebbero avere dei miglioramenti considerando più di due cluster.

6. RIFERIMENTI BIBLIOGRAFICI

'K-means' (2018) Wikipedia. Available at: https://it.wikipedia.org/wiki/K-means. (Accessed: 10 February 2022).

'Topic model' (2017) Wikipedia. Available at: https://en.wikipedia.org/wiki/Topic_model. (Accessed: 10 February 2022).

'Feature Selection' Avaiable at: https://scikit-learn.org/stable/modules/feature_extraction.htmltext-feature-extraction
'Amazon' Avaiable at: https://it.wikipedia.org/wiki/Amazon.com