Giugno 2022

IMDb Reviews Analysis

Text Mining & Search project

Marco Donzella 829358, Rebecca Picarelli 834286

Indice

1.Introduzione

Grazie a diverse tecniche di text mining, ossia un insieme di processi per analizzare ed estrarre informazioni utili da diversi formati di testo non strutturato, è possibile convertire la complessità e le caratteristiche qualitative dei testi in rappresentazioni più strutturate per facilitarne l'analisi e la generazione di conoscenza. Nel caso di questo progetto, ideato per il dataset IMDb contenente recensioni testuali su film e programmi televisivi lasciate dagli utenti, è stato deciso di sfruttare la text classification congiuntamente alla text representation per catalogare correttamente la sentiment positiva o negativa di ogni singola recensione.

Inoltre, su queste ultime è stata implementata anche la tecnica di apprendimento automatico non supervisionato del topic modelling con il fine di risalire ad alcuni argomenti astratti presenti nei documenti basandosi su strategie di clustering e sul concetto probabilistico di occorrenza delle parole.

1.1 Descrizione dei dati

Il dataset utilizzato per questo lavoro, scaricato dal sito *analyticsindiamag.com* , contiene 50.000 recensioni estratte dall'Internet Movie Databade (IMDb) suddivise a metà in train e test e, sempre a metà, in recensioni polarizzate in positivo (>=7 su 10) e in negativo (<=4). Inoltre, è presente anche una label di tipo binario indicante se la recensione fornita dall'utente è di tipo positivo o negativo. In aggiunta, vengono fornite ulteriori 50.000 recensioni che possono avere un punteggio qualsiasi tra 0 e 10. Infine, è precisato che non sono state inserite più di 30 recensioni per un unico film per evitare problemi di correlazione nel rating.

2. Analisi preliminari e preprocessing dei dati

La fase di preprocessing e di pulizia del dataset risulta fondamentale nell'ambito della text mining, in quanto i dati sono non strutturati e, quindi, difficili da poter analizzare. È necessario adottare una serie di strategie che permettano di rappresentare i dati testuali, recensioni in questo caso, in matrici e vettori numerici.

Prima di fare questo, però, sono stati preparati i dataset per le analisi di classificazione e topic modelling. Le recensioni (rappresentate in diversi elementi testuali *.txt*) sono state raccolte in 5 diversi dataset, a seconda del futuro utilizzo:

- Dataset di **training** contenente i commenti **positivi** (12500 record)
- Dataset di **training** contenente i commenti **negativi** (12500 record)
- Dataset di **test** contenente i commenti **positivi** (12500 record)
- Dataset di **test** contenente i commenti **negativi** (12500 record)
- Dataset di commenti **unsupervised**, senza label di riferimento (50000 record)

Dopo aver creato i dataset, sono state eseguite se le seguenti fasi di preprocessing:

1) Uniformazione dei testi in caratteri minuscoli
2) Rimozione di elementi superflui (emoji, punteggiatura, spazi bianchi)
3) Tokenizzazione
4) Lemmatizzazione
5) Rimozione delle stopwords

Una volta ottenuti i dati preprocessati, sono state generate 5 diverse word cloud (una per ogni dataset), così da rappresentare graficamente le parole più frequenti all'interno delle varie recensioni. Inoltre, è possibile verificare se sia presente una coerenza, a livello lessicale almeno, tra i cinque dataset.

2.1 Case folding

Come prima fase di preprocessing si è deciso di uniformare i dati in caratteri esclusivamente minuscoli, andando così ad eliminare l'eventuale presenza di lettere maiuscole. Questo è stato fatto utilizzando la funzione *.lower()*.

2.2 Emoji, punctuations & white spaces removal

Dopo aver uniformato il testo, si è proseguito con la rimozione di alcuni elementi superflui, quali emoji (emoticons ed altri simboli particolari), segni di punteggiatura e spazi bianchi non opportuni.

Nonostante nelle recensioni dei film non sia solito l'utilizzo di emoji, si è deciso di applicare comunque questa fase di preprocessing per prudenza, così da essere sicuri di ottenere dati puliti.

La rimozione della punteggiatura è, invece, una fase molto importante da applicare, soprattutto in caso di classificazione, in quanto semplificano l'analisi del testo e permettono una migliore implementazione di modelli ed algoritmi.

La rimozione di spazi bianchi è stata eseguita, in quanto può succedere di commettere un errore e digitare qualche spazio in eccesso. Sono stati rimossi ed è stato inserito un solo spazio per parola.

Tutte e tre le procedure sono state applicate definendo delle funzioni apposite.

2.3 Tokenizzazione

La tokenizzazione è una delle fasi di preprocessing più importanti: consiste nel separare la frase in token, cioè parole o gruppi di parole potenzialmente significativi. Esistono diversi metodi di individuazione dei token e, di conseguenza, altrettante tecniche di tokenizzazione.

In questo caso di studio è stato applicato il word tokenizer fornito dalla libreria nltk (*nltk.word_tokenize()*). Questa funzione distingue i token in base agli spazi bianchi ed alla punteggiatura. Inoltre, è particolarmente consigliato l'uso di questo tokenizer per la lingua inglese,

in quanto l'algoritmo riconosce anche i verbi in forma abbreviata (separati da apostrofi) e li divide in token differenti.

2.4 Lemmatizzazione

Questa tecnica di preprocessing consiste nel convertire le parole, che sono in forma flessa nel testo, nel loro lemma. Il lemma di una parola rappresenta la sua radice, il suo esponente.

In questo modo è possibile ridurre la dimensionalità del dataset, così da riunire in un'unica parola le parole caratterizzate dallo stesso lemma. Questo è stato fatto implementando una funzione che utilizzasse la funzione *WordNetLemmatizer()* offerta dal pacchetto nltk.

In alcuni casi è consigliabile applicare la tecnica di stemming (che riconduce allo stemma della parola, invece che al lemma), dato che comporta una minore perdita dell'informazione.

2.5 Stop words removal

Infine, sono state eliminate tutte le parole considerate non importanti, perché di uso comune, come ad esempio articoli, congiunzioni, preposizioni, ecc.

È possibile procedere tranquillamente alla rimozione di tali parole, in quanto non hanno un potere informativo di interesse per l'analisi. Sono state eliminate le stop words della lingua inglese, contenute nel pacchetto nltk.

È consigliabile fare attenzione in questa fase di pulizia dei dati, in quanto la rimozione di alcune parole (come, ad esempio, la negazione 'not') potrebbe portare ad una perdita di informazione, che sarebbe possibile evitare.

In seguito a tutte e 5 le fasi di preprocessing i dati sono stati trasformati nel seguente modo (vengono mostrati i primi record del dataset training positivi):

0	If you want mindless action, hot chicks and a ...	0	[want, mindless, action, hot, chick, post, apo...
1	Director Kinka Usher stays true to his own cre...	1	[director, kinka, usher, stay, true, credo, pl...
2	The novel is easily superior and the best part...	2	[novel, easily, superior, best, part, film, di...
3	Silly movie is really, really funny. Yes, it's...	3	[silly, movie, really, really, funny, yes, got...
4	Jesse and Celine (Ethan Hawke and Julie Delpy)...	4	[jesse, celine, ethan, hawke, julie, delpy, tw...

Figura 2: dati prima e dopo preprocessing

2.6 Word Cloud

Una volta puliti i dataset, è stata utilizzata la tecnica Word Cloud per avere una visualizzazione più chiara dei principali token presenti nelle recensioni. Questa tecnica è stata eseguita su tutti e 5 i dataset (training positivi, test positivi, training negativi, test negativi, unsupervised) in modo tale da verificare se fosse presente o no un qualche tipo di coerenza tra i diversi dati, almeno a livello lessicale.

In questo tipo di rappresentazione le parole sono caratterizzate da colori differenti e le dimensioni del carattere ne rispecchiano l'importanza.

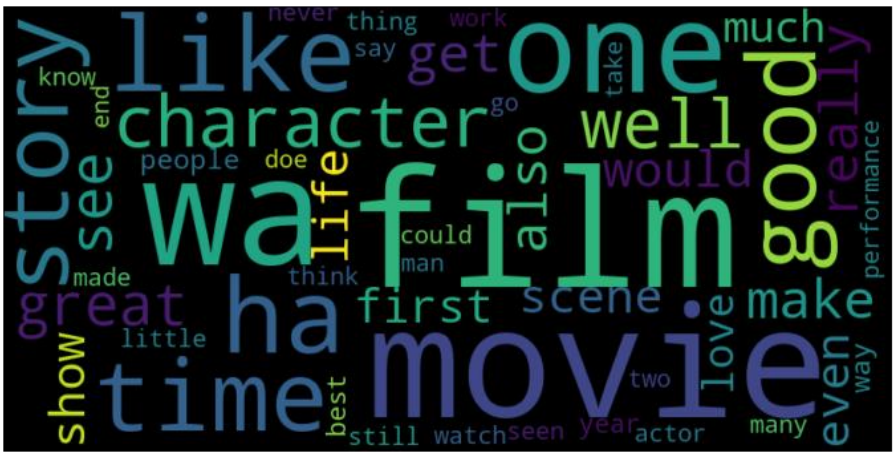


Figura 3: Word Cloud del dataset training positive

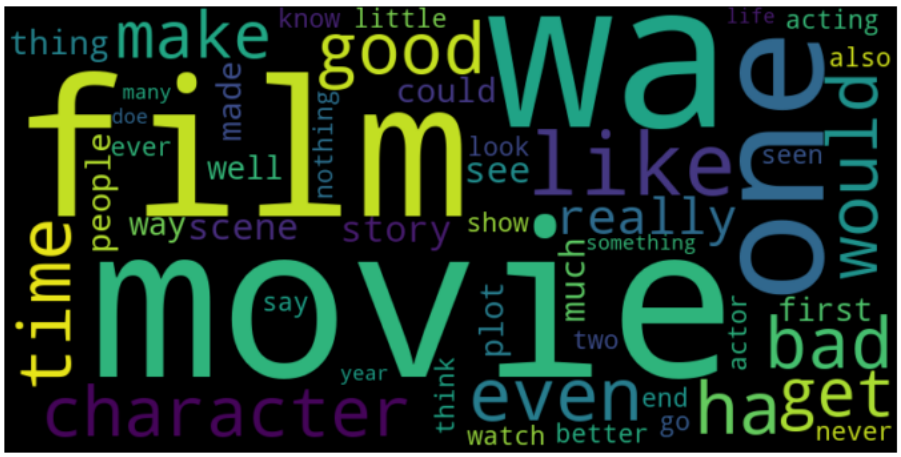


Figura 4: Word Cloud del dataset training negative

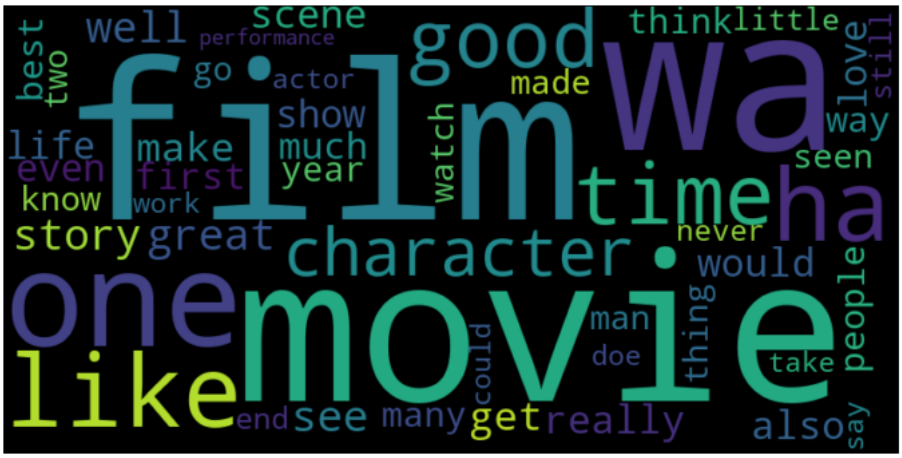


Figura 5: Word Cloud del dataset test positive

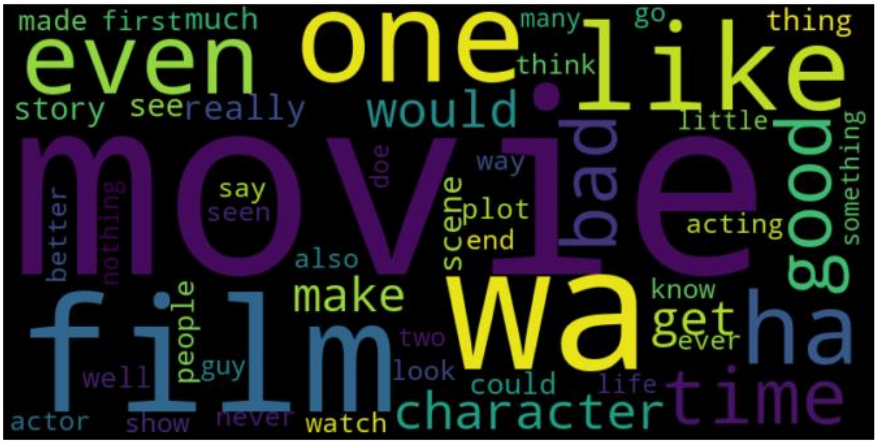


Figura 6: Word Cloud del dataset test negative

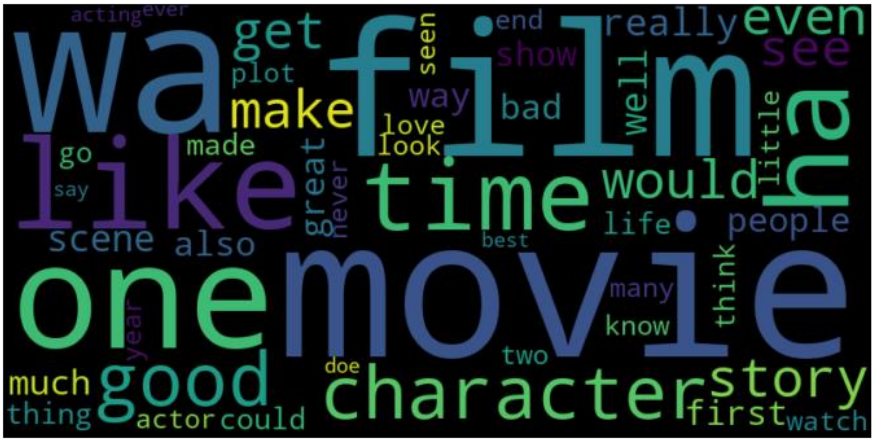


Figura 7: Word Cloud del dataset unsupervised

Dalle immagini sopra riportate, è possibile evincere una coerenza tra le parole dei 5 dataset, in quanto molte parole come 'movie', 'film', 'like', 'good', 'one', ecc. sono ripetute. È dubbia la presenza della parola 'wa', ricorrente in tutte le word cloud; tale termine potrebbe essere un acronimo.

3.Text Representation e Text Classification

Lo scopo principale del progetto consiste nell'affrontare un problema supervisionato di classificazione binaria, considerando come variabile target la sentiment (positiva o negativa) legata alle recensioni dei vari film. In questo modo risulta facile estrarre l'informazione delle diverse recensioni, potendo poi sfruttare tale informazione in diversi modi.

Per rispondere a questa domanda di ricerca, sono state eseguite diverse tecniche di rappresentazione testuale (in quanto risulta complicato trattare i dati in forma puramente testuale) e sono stati addestrati e confrontati più modelli di classificazione.

In particolare, le tecniche di text representation usate sono:

- Bag of Words binaria
- Bag of Words
- Bag of Words Bigram
- TF-IDF
- TF-IDF Bigram

La Bag of Words è una tecnica di rappresentazione testuale fondamentale nel NLP. Essa permette di trasformare un array di stringhe in un vettore numerico, trasformando così il dataset di interesse in una matrice, adatta da essere lavorata dai vari algoritmi di machine learning. I documenti rappresentano le righe di tale matrice, i token le colonne.

Esistono molte varianti di tale metodologia, in questo caso particolare ne sono state utilizzate tre.

La prima, Bag of Words binaria, costruisce dei vettori caratterizzati solo dal valore 0 in caso di assenza del token e dal valore 1 in caso di presenza.

La seconda, Bag of Words, costruisce dei vettori allo stesso modo, con la differenza che i valori non assumeranno unicamente valore 1 in caso di presenza del token d'interesse, ma assumeranno un valore pari alle occorrenze della parola all'interno del documento.

La terza, Bag of Words Bigram, sfrutta la stessa logica citata sopra, ma non considera un token alla volta, ma coppie di token, in quanto è stato definito un parametro aggiuntivo, ovvero l'N-gram range. In questo modo parole molto connesse tra loro, verranno considerate insieme.

Queste tre tecniche sono state implementate utilizzando la funzione *CountVectorizer()*.

Il termine TF-IDF sta per term frequency-inverse document frequency. Tale metodo di text representation attribuisce dei pesi alle parole dei documenti, dando un valore maggiore ai termini più rari e caratteristici all'interno della collezione. È una tecnica di text representation più sviluppata rispetto alla Bag of Words. Il peso viene ottenuto come prodotto tra la term frequency tf (la frequenza del termine nel documento) e la inverse document frequency idf (un valore che identifica l'importanza del termine nel documento). In questo modo, il valore del peso sarà più elevato per i termini rari e caratteristici del documento.

In genere è una delle migliori metriche da usare per la text representation.

In questo caso di studio, questa tecnica è stata implementata per mezzo della funzione *TfidfVectorizer()*.

Come nel caso della Bag of Words, anche qui è stata adoperata una variante considerando il parametro degli N-gram, imponendo un range da uno a due (considerando quindi sia unigrammi che bigrammi).

3.1 Modelli

Per rispondere al task di classificazione sono stati utilizzati i seguenti modelli di classificazione:

- Naive Bayes Multinomiale
- Support Vector Machine Lineare
- Regressione Logistica
- Random Forest
- Classificatore Ada Boost

Sono stati scelti modelli con proprietà diverse (linear, ensemble, boost), così da studiarne il comportamento con i dati testuali e le diverse metodologie di rappresentazione testuale adottate.

Ogni modello è stato addestrato sui dati di training ed è stato valutato sui dati di test, calcolandone statistiche come Accuracy, Precision, Recall, F1, F2.

Sono stati ottenuti i seguenti risultati:

1) Classificazione – Bag of Words binaria

	Train Accuracy	Test Accuracy	Precision	Recall	F1
MultinomialNB	0.84096	0.83660	0.83518	0.83872	0.83695
Linear SVM	0.87644	0.85576	0.84710	0.86824	0.85754
Logistic Regression	0.87680	0.85600	0.84929	0.86560	0.85737
Random Forest	1.00000	0.82660	0.82942	0.82232	0.82585
Ada Boost Classifier	0.80364	0.80092	0.78229	0.83392	0.80728

2) Classificazione – Bag of Words

	Train Accuracy	Test Accuracy	Precision	Recall	F1
MultinomialNB	0.83424	0.83016	0.83016	0.82720	0.82966
Linear SVM	0.87384	0.85680	0.84855	0.86864	0.85848
Logistic Regression	0.87468	0.85712	0.85028	0.86688	0.85850
Random Forest	1.00000	0.82836	0.83034	0.82536	0.82784
Ada Boost Classifier	0.80476	0.80012	0.78367	0.82912	0.80575

3) Classificazione – Bag of Words bigram

	Train Accuracy	Test Accuracy	Precision	Recall	F1
MultinomialNB	0.83136	0.82604	0.82487	0.82784	0.82635
Linear SVM	0.87336	0.85600	0.84738	0.86840	0.85776
Logistic Regression	0.87496	0.85584	0.84903	0.86560	0.85723
Random Forest	1.00000	0.82720	0.82678	0.82784	0.82731
Ada Boost Classifier	0.80432	0.80004	0.78432	0.82768	0.80542

4) Classificazione – TF-IDF

	Train Accuracy	Test Accuracy	Precision	Recall	F1
MultinomialNB	0.83968	0.83476	0.83160	0.83952	0.83554
Linear SVM	0.87500	0.85880	0.85365	0.86608	0.85982
Logistic Regression	0.87360	0.86048	0.85436	0.86912	0.86168
Random Forest	1.00000	0.82920	0.83047	0.82728	0.82887
Ada Boost Classifier	0.80472	0.80068	0.78257	0.83272	0.80687

5) Classificazione – TF-IDF bigram

	Train Accuracy	Test Accuracy	Precision	Recall	F1
MultinomialNB	0.83676	0.83236	0.82439	0.84464	0.83439
Linear SVM	0.87592	0.85816	0.85269	0.86592	0.85925
Logistic Regression	0.87272	0.85976	0.85337	0.86880	0.86102
Random Forest	1.00000	0.83172	0.83191	0.83144	0.83167
Ada Boost Classifier	0.80576	0.80164	0.78048	0.83936	0.80885

Osservando le tabelle riassuntive dei vari classificatori, è possibile affermare che complessivamente la combinazione migliore tra modello e text representation è la regressione logistica addestrata usando i dati rappresentati in forma tf-idf, con una Accuracy sul test set pari circa a 0.86. Tuttavia, questa differenza è minima rispetto agli altri modelli con le diverse rappresentazioni dati, che comunque portano a risultati soddisfacenti (attorno all'80% di accuratezza).

Il modello Random Forest è caratterizzato in tutti e 5 i casi da un problema di overfitting, in quanto il valore di Accuracy sui dati di training si discosta notevolmente da quello sui dati di test. Questo problema potrebbe essere risolto procedendo con la stima dei migliori parametri per tale classificatore, attraverso un procedimento di tuning (non effettuato in questo studio per nessun modello). È lecito pensare che questo potrebbe portare la Random Forest ad avere performance migliori, forse pure della regressione logistica.

Il classificatore Ada Boost, invece, sembrerebbe essere il peggiore tra tutti i modelli.

Contrariamente a quanto ci si potesse aspettare, le rappresentazioni Bag of Words portano a risultati tanto soddisfacenti quanto quelle tf-idf. Il parametro N-gram non sembra apportare alcun cambiamento significativo ai modelli di classificazione, in quanto le performance non variano.

4. Topic Modelling

Come già accennato in precedenza, con il topic modelling si arriva ad ottenere un numero (arbitrario) di topics astratti dal testo che dovrebbero rappresentare un insieme sensato di parole, o, meglio, una qualche struttura semantica/pattern presente nelle parole e nelle frasi.

Un modello molto comune, scelto anche per questo progetto grazie alla libreria open-source Gensim di Python, è quello della Latent Dirichlet Allocation (LDA) secondo il quale ogni documento, generato da un processo statistico generativo, è inteso come un insieme di topic e ognuno di questi un insieme di parole. Inoltre, i documenti sono rappresentabili in termini di Bag of Words e si ipotizza che la distribuzione dei topic nei vari documenti e quella delle parole in ogni topic assuma l'andamento della distribuzione di Dirichlet. In questo modo, quindi, le parole che tendono a co-occorrere spesso vengono raggruppate perché ritenute simili e ipoteticamente appartenenti ad un topic comune. Questo modello, inoltre, si caratterizza per due parametri di concentrazione, cioè alfa relativo alla document-topic density e beta per la topic-word density, che se mantenuti inferiori a 1 dovrebbero riuscire a fornire risultati più realistici. Tuttavia, esistono anche alcuni iperparametri di cui il principale è quello per selezionare a priori il numero di topic da ottenere nell'output del modello, che inizialmente è stato posto pari a 8, mentre sono stati tenuti esclusi sia i termini troppo frequenti che quelli troppo rari (che compaiono in più del 70% dei documenti e in meno del 30% rispettivamente).

Si precisa, in aggiunta, che l'analisi è stata portata avanti in parallelo per le recensioni positive che quelle negative, unendo inizialmente i dataset di training e test, oltre che sui dati forniti senza labels. Sono state implementate rapidamente anche alcune tecniche di preprocessing, come la rimozione di spazi bianchi, di punteggiatura e di numeri in modo da sistemare definitivamente il formato dei dati di input per il topic modelling. In tutti e 3 i casi si è potuto osservare che l'output grafico (Intertopic Distance Map), mostrato nelle pagine successive, risultava più chiaro (bolle grandi e distanziate) mantenendo un numero ridotto di topic pari a 5, a scapito, però, dell'interpretabilità che risultava eccessivamente confusionale. Per tale ragione la configurazione di riferimento selezionata è rimasta quella con 8 topic: graficamente i cerchi risultavano comunque ben distanziati e non troppo piccoli e,per di più, anche i valori della metrica di valutazione intrinseca della Coherence (c_v), che riflette la similarità semantica delle top word di un topic, sono rimasti piuttosto simili. Inoltre, sono stati valutati anche i punteggi di questa

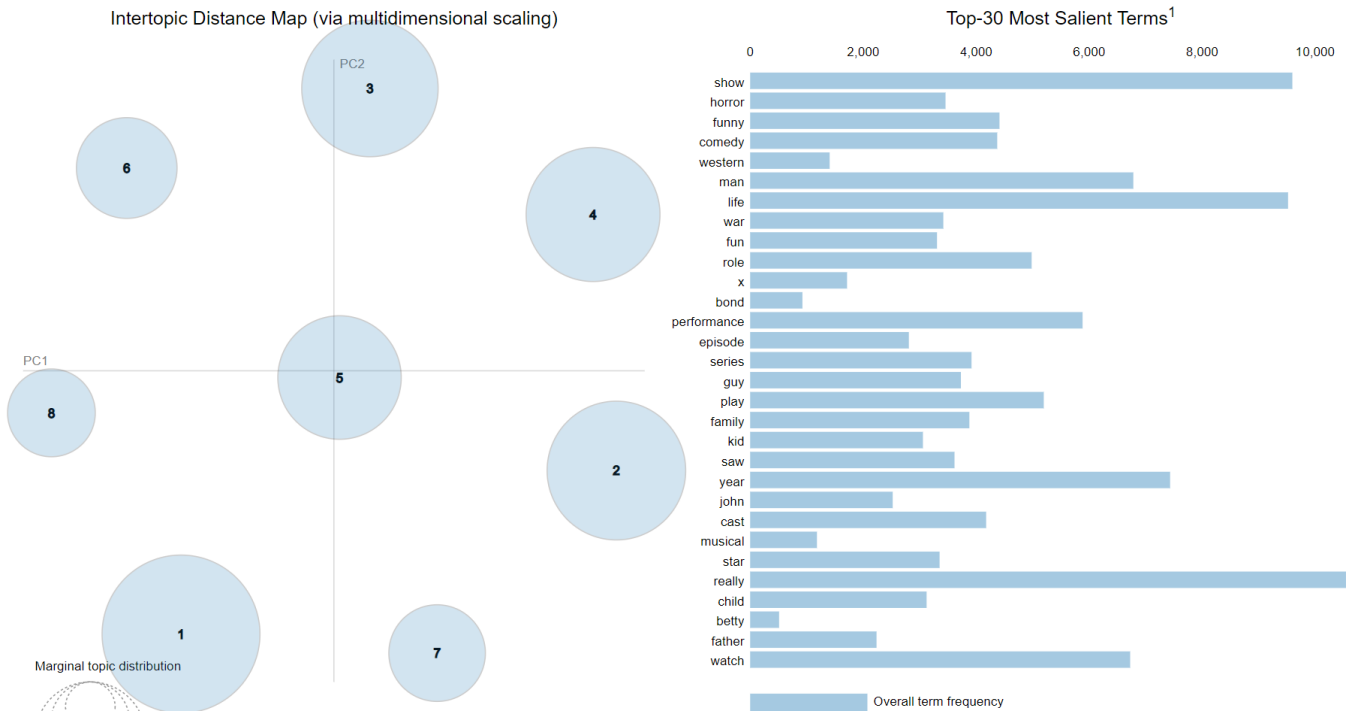
metrica provando a inserire come numero di topic 10,15 e 19. Per la precisione, si segnala che il punteggio più alto del Coherence Score, per tutti i 3 dataset utilizzati, si ottiene con 19 topic, che però pare eccessivo considerando il tipo di dati analizzati. Infine, è stata calcolata anche l'altra misura intrinseca della Perplexity, i cui valori, essendo negativi, hanno confermato che i modelli sono in grado di prevedere abbastanza bene nuovi dati.

Nella tabella seguente si riportano i valori ottenuti per le metriche sopracitate:

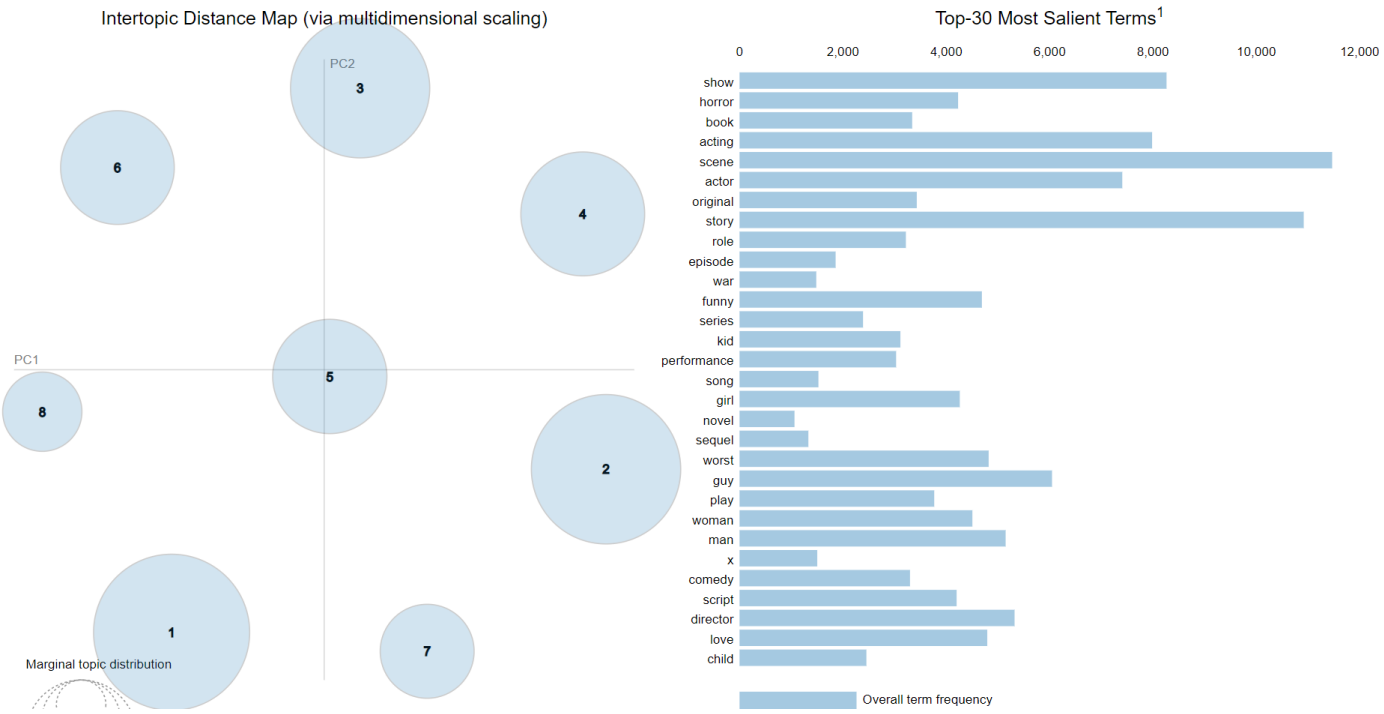
	Positive reviews	**Negative reviews**	**Mixed reviews**
Perplexity(8topics)	-8,393	-8,321	-8,410
Coherence (5 topics)	0,293	0,283	0,293
Coherence (8 topics)	0,282	0,281	0,286
Coherence (10 topics)	0,290	0,282	0,318
Coherence (15 topics)	0,294	0,281	0,339
Coherence (19 topics)	0,303	0,284	0,337

Tuttavia, per quanto concerne una valutazione umana dei risultati del modello, come già precedentemente accennato, si può affermare che non sempre è stato possibile distinguere con facilità gli argomenti dei topic osservando le parole che li costituiscono. In tutti e tre i casi, comunque, le parole emerse nei topic sono chiaramente riferite al mondo dello spettacolo, con dei possibili riferimenti al cast e al tipo di personaggi (o spettatori?), al genere della rappresentazione (commedie, romantici, horror, serie tv) e alle opinioni degli utenti.

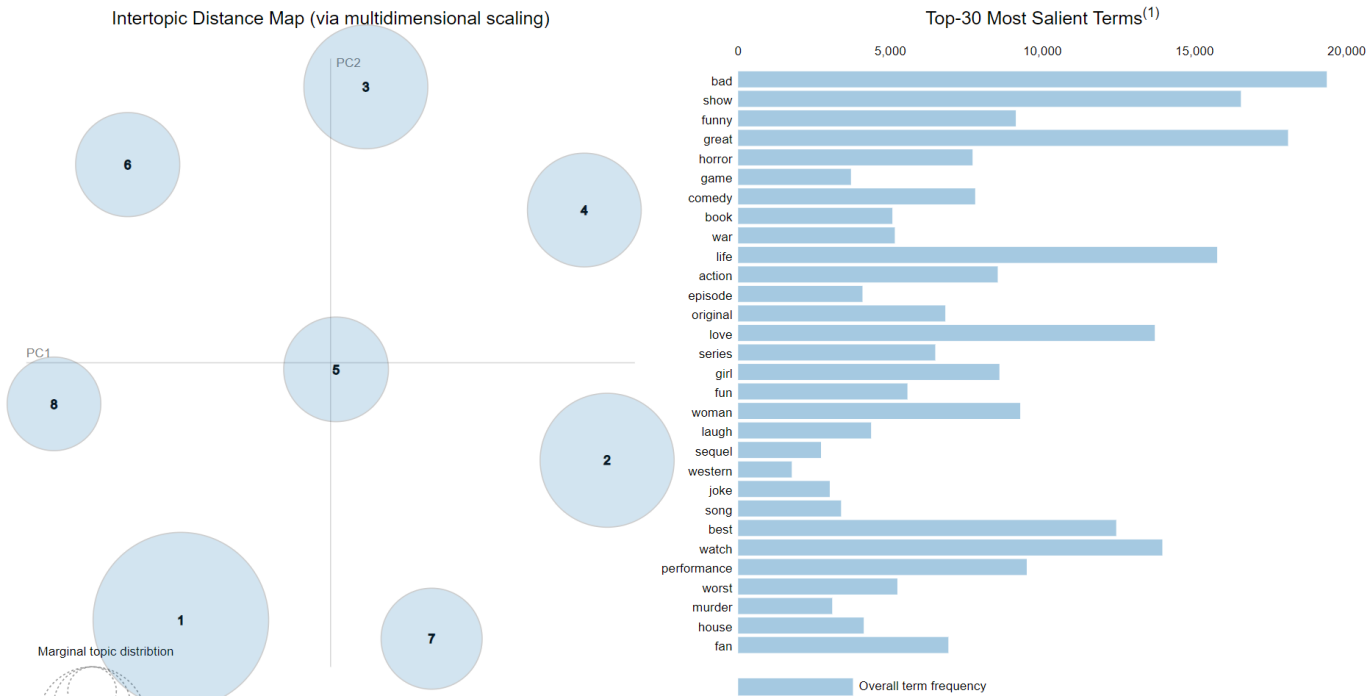
Infine, per curiosità, la scelta di creare due grandi dataset separati unendo training e test dei dataset con recensioni polarizzate è stata fatta per cercare di confrontare i risultati osservando eventualmente se il sentiment trasparisse dalle parole usate per generare i topic. Tuttavia, ciò non è risultato particolarmente chiaro, salvo per l'individuazione di parole come "worst" nella top 30 dei termini salienti del modello basato su recensioni negative, o di "loved" e "funny" per quello con recensioni positive, anche se presenti pure nell'altra classifica.



Output LDA con recensioni positive



Output LDA con recensioni negative



Output LDA con recensioni miste

5. Conclusioni

In conclusione, sebbene tendenzialmente quasi tutte le combinazioni di text representation e modelli generassero risultati più che discreti, l'abbinamento vincente è quello della regressione logistica implementata sui dati rappresentati in forma tf-idf. Si è anche osservato che con le rappresentazioni Bag of Words non vi è alcun peggioramento nei risultati, che rimangono sostanzialmente invariati anche considerando il parametro N-gram.

Per quanto concerne il topic modelling, in sintesi, si è quindi optato per selezionare il modello con 8 topics considerando sia i valori delle metriche di Coherence, e in misura più limitata la Perplexity, sia l'interpretabilità umana dei cluster ottenuti, che in entrambi i casi avrebbero un notevole margine di miglioramento. Inoltre, un'identificazione chiara del tipo di recensioni polarizzate utilizzate non è stata possibile.

6. Sviluppi futuri

Per quanto riguarda il task di classificazione, nonostante i risultati siano soddisfacenti, con alcuni accorgimenti sarebbe quasi sicuramente possibile giungere a prestazioni migliori.

Ulteriori miglioramenti potrebbero essere portati semplicemente dall'utilizzo di altri classificatori ed altre tecniche di rappresentazione testuale.

Avendo a disposizione dei processori potenti (GPU) è possibile ricorrere a tecniche di deep learning sia per il task di rappresentazione testuale (embedding) sia per la classificazione (usando reti neurali preaddestrate). Anche il classificatore Support Vector Machine in forma non lineare potrebbe portare a risultati soddisfacenti. Per farlo lavorare al meglio però, sarebbe necessaria innanzitutto la stima dei suoi migliori parametri (computazionalmente onerosa, vista la quantità di dati a disposizione).

Per il topic modelling, invece, si potrebbe effettuare un preprocessing più accurato, specialmente per la rimozione di alcune parole che non contribuiscono a risalire ad un argomento vero e proprio (alcuni verbi, verbi modali, preposizioni…), oltre che tentare di implementare un modello che lavori con bigrammi o trigrammi. Inoltre, sarebbe opportuno soffermarsi maggiormente sul tuning di alcuni parametri delle funzioni utilizzate.