TEXT MINING AND SEARCH
PROGETTO FINALE

MINING IMDB REVIEWS

Autori:
Casula Filippo Maria (860894)
Mura Giulia (860910)
Pisani Caterina (853058)

Febbraio 2021

MINING IMDB REVIEWS

INTRODUZIONE

Il text mining si pone l'obiettivo di studiare metodi e algoritmi per estrarre automaticamente conoscenza dal testo per classificare (o raggruppare) documenti in base ai contenuti.

In questo caso sono state applicate diverse tecniche di text mining su un dataset composto da una grande quantità di recensioni di film tratte dalla piattaforma IMDB.

Sito web di proprietà di Amazon.com, IMDb (Internet Movie Database) è un archivio di informazioni su tutto ciò che ruota intorno ai film e alle serie televisive: attori, registi, sceneggiatori, nonché le recensioni da parte degli utenti.

Il progetto è stato strutturato su diverse fasi:

- Analisi esplorativa, preprocessing e rappresentazione del testo delle recensioni
- Implementazione di cinque modelli per la classificazione delle label
- Applicazione di sentiment analysis attraverso tre differenti approcci
- Text clustering con algoritmo k-means
- Text modeling con LSA e LDA

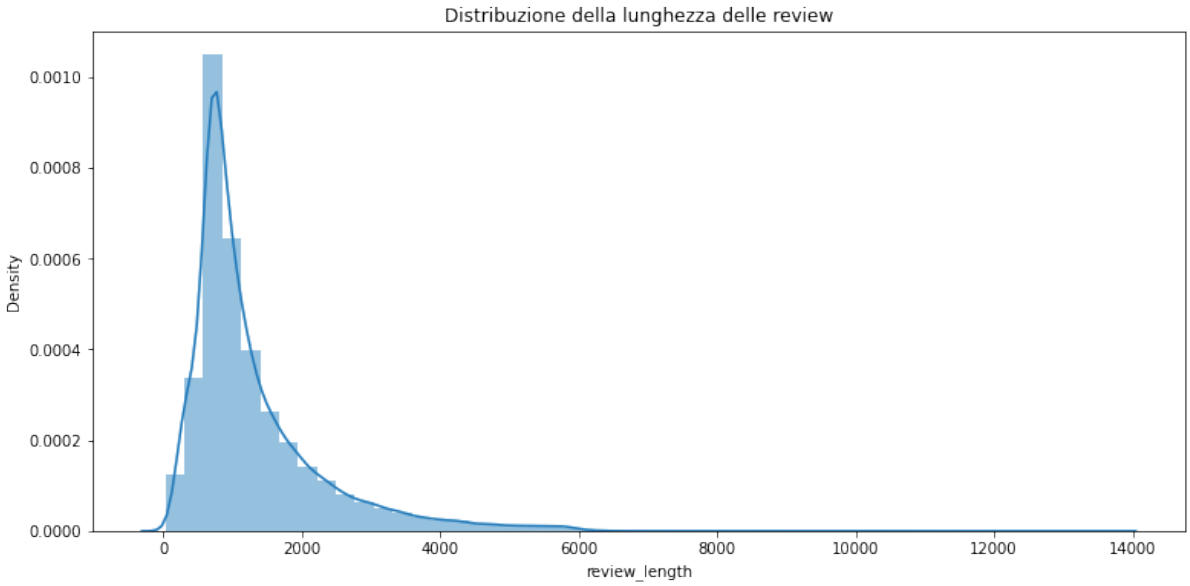
DATASET

Il dataset utilizzato contiene in totale 25000 recensioni positive e 25000 recensioni negative pre-etichettate. Si presenta con una cartella di train e una di test, all'interno delle quali si trova un'ulteriore suddivisione in cartelle "pos" e "neg" perfettamente bilanciate. Le review sono sotto forma di file txt, portando ad avere 25000 file nella cartella train e 25000 in quella di test.

Inoltre, le review sono etichettate come positive se il voto ricevuto dal film è compreso tra 7 e 10 o come negative se il voto sta tra 1 e 4. I voti pari a 5 e 6 sono considerati neutri e per questo non vengono inclusi in nessuna classificazione.

Per rendere il dataset di più facile manipolazione, vengono riportati tutti i txt sotto forma di un unico file csv chiamato composto da 50000 righe e due colonne: "review" dove viene riportato il testo della recensione e "sentiment" che rappresenta l'etichetta positiva o negativa delle recensioni (0 se negativa, 1 se positiva).

Di seguito viene riportata la distribuzione del numero di caratteri presenti nelle recensioni e si può notare come la maggiorparte di queste abbiano meno di 2000 caratteri.



TEXT PREPROCESSING

Il preprocessing del testo è un passaggio cruciale per l'elaborazione del linguaggio naturale (NLP), in quanto ha lo scopo di rendere il testo più comprensibile per le macchine e gli algoritmi di Machine Learning.

La prima fase affrontata è stata quella di normalizzazione del testo, la quale è costituita da diverse operazioni:

- Espansione delle forme contratte nel loro modulo "standard" (es: won't diventa will not)
- Trasformazione di ogni carattere in minuscolo (lower case)
- Rimozione di \n, \t e
- Rimozione di urls
- Rimozione della punteggiatura
- Rimozione degli spazi bianchi all'inizio e alla fine delle stringhe
- Rimozione dei numeri
- Rimozione delle emoji
- Rimozioni degli spazi bianchi di troppo in mezzo alle stringhe

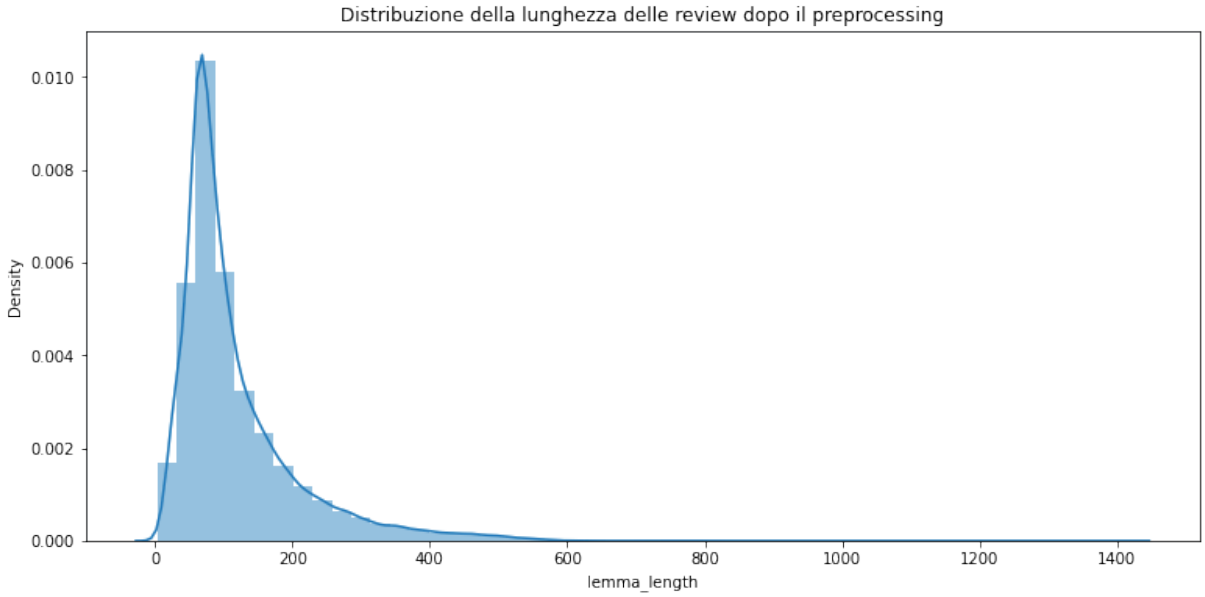
Si è deciso di applicare prima di tutto l'espansione delle forme contratte e solo dopo la trasformazione in lower case perché si è notato che in caso contrario quest'ultima funzione non era in grado di riconoscere alcune lettere comprese nelle forme contratte (es: in "I'm" il pronome "I" rimaneva maiuscolo), il che avrebbe comportato problemi nelle operazioni successive.

La seconda fase è consistita nella rimozione delle stop-words. Tra queste ricadono parole come pronomi e articoli che compongono una buona parte del dataset ma non contribui-

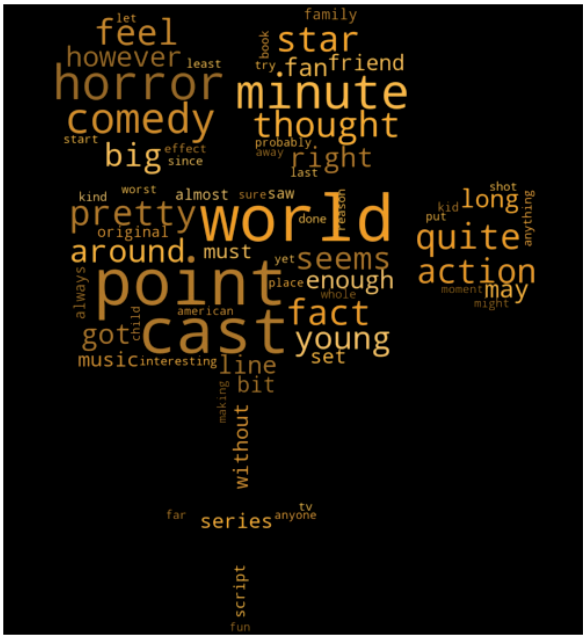
scono alla sua informatività. Per questo, la loro eliminazione ha il vantaggio di diminuire le dimensioni del dataset senza ridurne la qualità in termini di informazioni che possono essere ricavate.

Infine, il testo è stato lemmatizzato e tokenizzato. La lemmatizzazione riconduce le parole alla loro forma base, mentre la tokenizzazione suddivide il testo in unità più piccole dette "token".

Nell'immagine si può vedere com'è cambiata la distribuzione del testo dopo la fase di preprocessing. In particolare, si osserva come la maggior parte della recensioni siano ora comprese tra i 100 e i 200 caratteri, invece dei precedenti 1000/2000.



Viene inoltre mostrata una word cloud che riporta i token a seguito della lemmatizzazione in un range di frequenza compreso tra 20 e 70, per poter visualizzare solo i termini più rilevanti.



TEXT REPRESENTATION

Per la rappresentazione del testo si è optato per 3 diverse tecniche, che saranno portate avanti parallelamente in ogni step di analisi:

- Bag of words: rappresentazione vettoriale in cui viene associato ad ogni termine la sua frequenza all'interno di un documento
- Bag of words con rappresentazione binaria: rappresentazione vettoriale in cui viene associato ad ogni termine un valore pari a 0 o a 1, rispettivamente se è assente o presente nel documento
- Tf-Idf: rappresentazione vettoriale in cui viene associato ad ogni termine il valore Tf-Idf. Quest'ultimo è definito come il rapporto tra la Term Frequency e la Inverse Document Frequency. La Term Frequency è la frequenza del termine all'interno del documento. La Inverse Document Frequency è una misura di quanto il termine occorre all'interno dell'intero corpus. Questa rappresentazione ha il vantaggio di enfatizzare i termini più rilevanti.

Il risultato di queste rappresentazioni sono matrici sparse, ovvero caratterizzate da un'ampia presenza di zeri. Di conseguenza, si è proceduto ad una riduzione della dimensionalità per ottenere un embedding del corpus di più facile manipolazione per la fase di classificazione.

In particolare, la tecnica adottata è stata la Singular Value Decomposition (SVD), che ha consentito di ridurre la sparsità delle features in favore di una maggiore capacità computazionale.

TEXT CLASSIFICATION

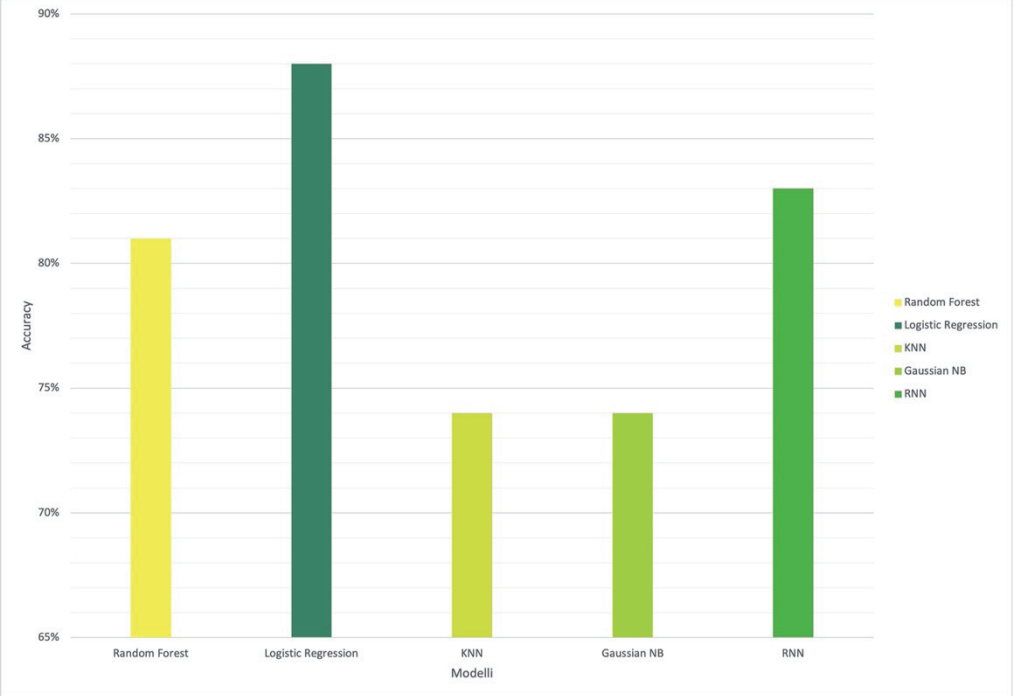
I risultati ottenuti in seguito all'applicazione dei metodi Bag of Words, Binary e Tf-Idf e alla rispettiva Dimensionality Reduction, sono infine stati splittati in training set e test set rispettivamente pari all'80% e al 20%.

La classificazione in questione è una classificazione binaria, in quanto il dataset è rappresentato dalle due sole labels, 0 e 1.

I modelli di classificazione utilizzati in questa analisi sono:

- Random Forest
- Logistic Regression
- KNN
- GaussianNB
- Recurrent Neural Network

Si è constatato che per i primi quattro modelli di classificazione si è ottenuta una performance migliore con l'utilizzo del metodo Tf-Idf. In questo caso il modello che ha performato meglio è stato Logistic Regression con un'accuracy pari all'88%.



Per la Recurrent Neural Network è stato utilizzato un approccio diverso rispetto agli altri modelli.

Come primo step è stato creato un vocabolario di dimensione pari a 30.000 riportando i token in sequenze ed impostando un massimo di lunghezza del vettore pari a 250. Si è proceduto alla partizione impostando, come negli altri modelli, il training set all'80% e il test set al 20%.

La rete è stata impostata con le seguenti caratteristiche:

- Input layer: Embedding layer il quale riporta tre parametri. Il primo parametro pari a 30.000 indica la dimensione del vocabolario, il secondo riporta la dimensione dell'embedding impostata pari a 100 ed infine il terzo parametro indica la lunghezza di ogni sequenza pari a 250;
- LSTM layer: impostato con 100 neuroni. Le funzioni di attivazione e attivazione ricorrente sono state mantenute come di default dal pacchetto Keras.
- Output layer: indicato da un Dense layer costituito da 2 neuroni (ricordiamo che si hanno solo due classi) e in questo caso come funzione di attivazione si è impostato sigmoid

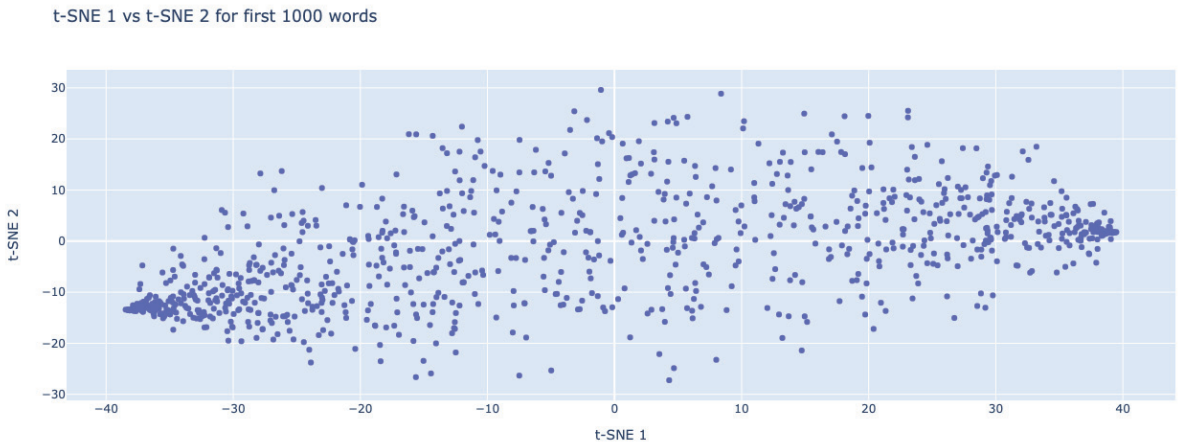
La compilazione della rete è stata fatta con ottimizzatore "adam", "loss binary crossentropy" e come metrica "accuracy". Per l'addestramento della rete sono state impostate 3 epoche ed un batch size pari a 2.048.

In conclusione, la rete ha performato sul test set con un accuracy pari all'83%. Di seguito viene riportato il corrispondente classification report.

	precision	recall	f1-score	support
Negative_Reviews	0.83	0.85	0.84	5007
Positive_Reviews	0.84	0.82	0.83	4993
accuracy			0.83	10000
macro avg	0.83	0.83	0.83	10000
weighted avg	0.83	0.83	0.83	10000

Word embedding visualization

Si è deciso di visualizzare i pesi del word embedding ottenuto dalla rete riducendo la dimensione dello stesso da 100 a 2 dimensioni utilizzando t-SNE.



SENTIMENT ANALYSIS

La Sentiment Analysis è un'analisi procedurale di calcolo dei sentimenti e delle opinioni espresse nei testi su un prodotto o un servizio, su un'azienda, un brand o un evento.

Si tratta di un'applicazione di data mining ai social network/e-commerce/blog/etc con lo scopo di comprendere l'opinione o il giudizio di un utente etichettando i testi con differenti aggettivi.

Si sono sviluppate 3 differenti Sentiment Analysis:

- **Vader**: è un modello utilizzato per l'analisi del sentiment testuale, disponibile nel pacchetto NLTK, si basa su un dizionario che mappa le caratteristiche lessicali con le intensità delle emozioni, fornendo una percentuale di positività, di neutralità e di negatività del testo analizzato.
- **TextBlob**: è un pacchetto per attività di elaborazione del linguaggio naturale (NLP), è basato su NLTK e può essere utilizzato per eseguire una varietà di attività tra cui l'analizzatore sentimentale
- **Afinn**: diversamente dai precedenti, la libreria Afinn si basa su elenchi di parole per l'analisi del sentiment.
  Viene assegnato un punteggio alla singola parola all'interno della frase e la loro somma corrisponde al *sentimental score* della frase.

Una volta applicate le 3 sentiment analysis, il loro punteggio è stato normalizzato nell'intervallo 0-1, in cui 0 rappresenta una recensione molto negativa e 1, invece, molto positiva. Successivamente, è stata assegnata un'etichetta 0 (negativa) alle recensioni con un punteggio

compreso tra 0 e 0.50 e un'etichetta 1 (positiva) a quelle con valori compresi tra 0.51 e 1. I risultati ottenuti sono stati confrontati con le etichette (positivo, negativo) di cui il dataset era già provvisto per comprendere l'accuracy dei classificatori. Il metodo più performante è risultato essere TextBlob con un'accuracy quasi del 70% seguito da Vader con 66.6% e infine Afinn con 61.24%.

	TextBlob	Vader	Afinn
FALSE	15094	16706	19938
TRUE	34906	33294	30062

TEXT CLUSTERING

Nella fase di text clustering si è utilizzato l'algoritmo k-means per individuare due cluster all'interno del testo, considerando ogni diverso tipo di rappresentazione (bag of words, binary e tfidf).

L'obiettivo è quello di stabilire se l'algoritmo sia in grado di individuare gruppi che condividono informazioni con la suddivisione tra positivo e negativo delle labels a disposizione.

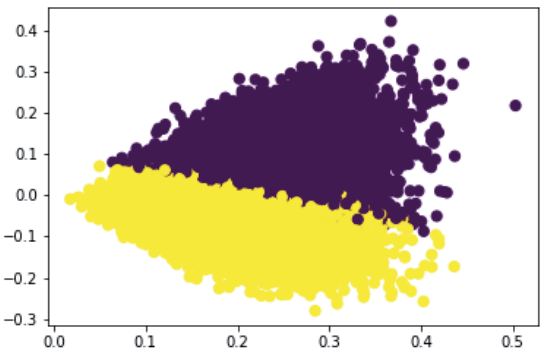
La metrica grazie alla quale è possibile stabilirlo è la Mutual Information, che misura quanto l'informazione riguardo ad una variabile casuale X o Y riduca l'incertezza dell'altra, in altre parole misura l'informazione condivisa dalle due variabili.

I cluster ottenuti sono diversi per le 3 rappresentazioni dei testi e l'indice di Mutual Information normalizzato è risultato molto basso in ogni caso, ad indicare che non c'è corrispondenza tra i cluster trovati dall'algoritmo e la suddivisione dei labels.

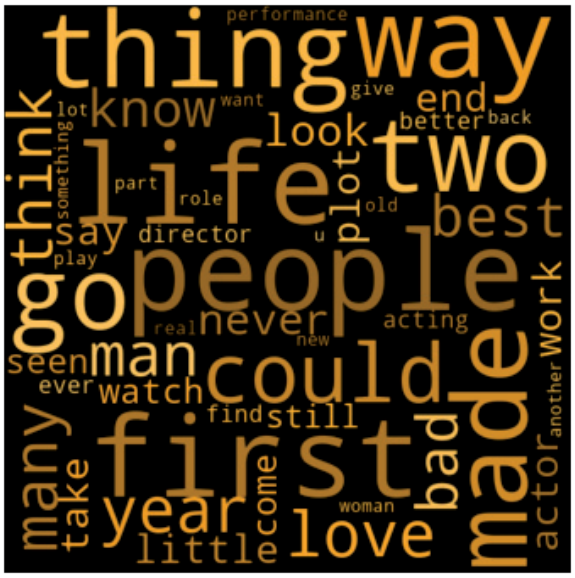
Di seguito vengono riportati i valori dell'indice:

	Bag of words	Binary	Tf-Idf
Normalized Mutual Information	9.56e-05	0.00016	0.01734

Nell'immagine viene riportata la rappresentazione dei due cluster individuati nel caso di rappresentazione tfidf. I testi sembrano essere più o meno equamente spartiti tra i due gruppi e sono piuttosto distinguibili.



A riconferma del fatto che i cluster non corrispondano ad una suddivisione tra review positive e negative, vengono riportate anche le word cloud rappresentanti i due cluster dalle quali è difficile estrapolare una tematica particolare.



TOPIC MODELING

Nello svolgimento di questo ultimo task si è ridotta ulteriormente la dimensione della matrice (utilizzando SVD come in precedenza) e l'analisi è stata svolta sulle recensioni lemmatizzate.

Sono state applicate le tecniche LSA e LDA.

Nell'applicazione della tecnica LSA è stata decomposta la matrice ottenendo 5 differenti topic rappresentati ognuno da 7 parole.

Topic 0	Topic 1	Topic 2	Topic 3	Topic 4
Like	Bad	Great	Character	Series
Good	Worst	Funny	Good	Episode
Time	Acting	Good	Actor	Book
Character	Really	Episode	Story	Tv
Story	Like	Series	Performance	Season
Really	Good	Watch	Acting	Original
Make	Watch	Comedy	Scene	Character

Prendendo come esempio il topic 2, si potrebbe identificare più o meno facilmente una tematica legata al genere comico.

Nell'applicazione di LDA è stato utilizzato il pacchetto gensim per la creazione di un dizionario utilizzando le recensioni lemmatizzate. Tale dizionario è stato costruito impostando dei filtri quali: tenere le parole presenti in meno di 5 documenti e presenti in più del 50% dei documenti totali ed infine il mantenimento delle parole più frequenti pari alle prime 100.000.

Topic 0	0.007 * 'great' + 0.006 * 'best' + 0.006 * 'role' + 0.005 * 'play' + 0.005 * 'also' + 0.005 * 'year' + 0.005 * 'comedy' + 0.005 * 'performance' + 0.005 * 'cast' + 0.004 * 'well'
Topic 1	0.008 * 'story' + 0.006 * 'life' + 0.005 * 'world' + 0.005 * 'time' + 0.005 * 'character' + 0.005 * 'war' + 0.004 * 'u' + 0.004 * 'people' + 0.004 * 'many' + 0.004 * 'well'
Topic 2	0.007 * 'woman' + 0.007 * 'man' + 0.006 * 'character' + 0.005 * 'life' + 0.004 * 'performance' + 0.004 * 'scene' + 0.004 * 'wife' + 0.004 * 'young' + 0.004 * 'story' + 0.004 * 'two'
Topic 3	0.014 * 'like' + 0.011 * 'good' + 0.010 * 'really' + 0.009 * 'time' + 0.008 * 'would' + 0.008 * 'see' + 0.008 * 'even' + 0.008 * 'character' + 0.007 * 'make' + 0.007 * 'bad'
Topic 4	0.010 * 'horror' + 0.006 * 'scene' + 0.005 * 'get' + 0.005 * 'like' + 0.005 * 'effect' + 0.004 * 'action' + 0.004 * 'look' + 0.004 * 'bad' + 0.003 * 'even' + 0.003 * 'kill'

Con l'implementazione di LDA possiamo notare come in particolar modo il topic 2 e il topic 4 possano identificare rispettivamente una tematica legata alla vita di coppia e l'altro ad un genere thriller-horror.

Nonostante ciò, lo score della Coherence Measures applicato sul modello LDA risulta essere piuttosto basso, pari a 0.33.

CONCLUSIONI

In conclusione, il progetto si è occupato di svolgere diversi tipi di analisi sul dataset di 50000 reviews di IMDB.

Dopo essere stato normalizzato e lemmatizzato, il testo è stato rappresentato tramite Bag of Words, Bag of words Binary e Tf-Idf.

Il task di classificazione ha evidenziato come rappresentazione più idonea il tf-idf, il quale restituisce generalmente valori di accuracy più elevati ed in particolare si hanno i risultati migliori con il classificatore di Logistic Regression. La seconda classificazione migliore è restituita dalla Recurrent Neural Network, allenata sul testo lemmatizzato e su cui è stato applicato word embedding.

Un altro genere di classificazione è stato affrontato performando la Sentiment Analysis con tre diversi approcci supervisionati, il migliore dei quali è stato il Text Blob, seppur rimanendo inferiore alla "pura" classificazione ottenuta precedentemente.

Il terzo task è consistito nel text clustering con il quale si è voluto verificare se una suddivisione in due cluster con k-means potesse corrispondere alla distinzione espressa dai labels

già a disposizione. Guardando alla metrica di Mutual Information, si è constatato che tale assunzione non era verificata.

Infine, con il Topic Modeling si sono considerati un numero di topic pari a 5 per verificare se si riuscisse ad identificare il genere dei film oggetto di recensione. Tuttavia, sia LDA che LSA sembrano restituire topic di definizione più generica. Anche la misura di coherence calcolata sul modello LDA conferma questa tendenza.