Mi consigli un film?

Analisi recensioni cinematografiche da IMDb

Andrea Marinoni 799690, Andrea Cattaneo 815585

Text-Mining and Search, Febbraio 2022

Introduzione 3

1.Raccolta Dati 4

2.Preprocessing visualizzazioni e esplorazione 4

2.1 Uniformare i testi, eliminando i caratteri maiuscoli 5

2.2 Rimuovere emoji 5

2.3 Rimuovere segni di punteggiatura e White Space 5

2.4 Tokenizzazione 5

2.5 Lemmatization 6

2.6 Rimozione delle stopwords 6

2.7 Wordcloud 2 7

3. Text Representation e Classification 8

3.1 tf - idf 8

3.2 binary tf - idf (unsupervised feature weighting) 9

3.3 bigram tf - idf (unsupervised feature weighting, rappresenta importanza di due parole per un documento) 9

3.4 trigram tf - idf (unsupervised feature weighting, rappresenta importanza di tre parole per un documento): 9

3.5 Bag of Words: 10

3.6 Bag of Words with N-grams: 10

3.7 Risultati classification: 10

4. Topic modelling 14

5.Conclusioni 16

6.Sviluppi Futuri 16

Introduzione

La Text-Mining Analysis è una disciplina che si occupa di trasformare informazioni, che sono esplicite nei testi, in conoscenza.

La necessità è quindi quella di rendere quantificabili aspetti qualitativi, cercando di comprendere il più possibile le qualità semantiche del testo, in modo da poter condurre delle analisi.

Obiettivo del progetto è realizzare un modello di text classification capace di riconoscere la sentiment positiva o negativa di una recensione di un film scritta da un utente su IMDb.

Per fare ciò si è ricorso a diversi modelli di classificazione addestrati su dati pre processati e modellati attraverso numerose tecniche di text representation.

Si vuole trovare la migliore combinazione, che garantisca performance elevate.

Realizzata la text classification si è provato ad effettuare il topic modelling delle recensioni. Esso è una tecnica di machine learning non supervisionata che si occupa di individuare argomenti astratti intrinseci in documenti di testi utilizzando modelli probabilistici.

L'obiettivo è quindi quello di comprendere meglio le recensioni, anche in questo caso il modello di topic modelling verrà valutato per capirne le performance.

1.Raccolta Dati

Il lavoro compiuto riguarda l'analisi testuale di recensioni di film, contenute nel database IMDb.

IMDb (Internet Movie Database), è un database multimediale fornito da Amazon, che fornisce informazioni su milioni di film e programmi televisivi.

I dati importati per l'analisi sono stati resi disponibili sul sito di analyticsindiamag [1] .

Il dataset si compone di 50.000 reviews, recensioni degli utenti, di film. E' stato fornito sia in forma di raccolta di 50.000 recensioni miste, che in forma di dataset singoli con solo recensioni positive o solo recensioni negative.

Il dataset è perfettamente bilanciato nella distribuzione della variabile target. Si compone, infatti, di 25.000 recensioni fortemente positive e 25.000 recensioni fortemente negative.

Il primo passo compiuto è stata l'unione in dataset separati per train, test e sentimento negativo e positivo delle review, e la creazione di un dataset unico misto.

Al di fuori dei testi delle recensioni e della loro sentiment negativa o positiva non sono stati forniti altri dati relativi ai film.

Partendo dai dataset di sole recensioni positive e sole recensioni negative si è costruito un unico dataframe presentante due colonne, una, review, con i testi delle recensioni e una, sentiment, corrispondente ad una variabile booleana che assume i valori "True" e "False" a seconda che la recensione sia positiva o negativa.

2.Preprocessing visualizzazioni e esplorazione

Il preprocessing dei dati è il processo di trasformazione dei dati grezzi in un formato comprensibile. È un passo importante nell'elaborazione dei testi poiché senza di esso le analisi desiderate risulterebbero falsate e non coerenti con la realtà.

Non esiste un preprocessing standard da seguire, ma la qualità dei testi deve essere adattata alla tipologia di analisi e al contesto d'interesse.

La Natural Language Processing (NLP) è una branca della scienza dei dati che si occupa di dati in formato testuale.

I passaggi di pulizia dei dati utilizzati nel progetto sono:

- Uniformare i testi, eliminando i caratteri maiuscoli
- Rimuovere emoji
- Rimuovere segni di punteggiatura
- Rimuovere White Space
- Tokenizzazione
- Lemmatizzazione
- Rimuovere le Stop Words

2.1 Uniformare i testi, eliminando i caratteri maiuscoli

Si è utilizzata la funzione .lower() che permette di convertire tutti i caratteri maiuscoli di una stringa in caratteri minuscoli rendendo uguali tra loro parole differenziate solo dalla presenza di lettere maiuscole.

2.2 Rimuovere emoji

Presumibilmente nelle recensioni dei film su IMDB non sono utilizzate emoji, tuttavia non avendone la certezza, si è scelto di adottare una linea prudenziale implementando comunque una procedura per rimuovere quelle eventualmente presenti.

2.3 Rimuovere segni di punteggiatura e White Space

La rimozione della punteggiatura è un passaggio di preprocessing comune in molte attività di analisi dei testi.
Se si sta creando un modello di classificazione del testo, la punteggiatura non è di alcuna utilità e rischia di rendere il processo più lungo e scorretto.
Per quanto riguarda i White Space, essi possono essere inseriti erroneamente in più punti soprattutto in testi scritti superficialmente su social, o rewiev come nel caso di analisi.
Si è deciso di rimuoverli e di sostituirli con un solo White Space per parola.

2.4 Tokenizzazione

La tokenizzazione separa le parole contenute in un testo formando dei token, delle parole o gruppi di parole, potenzialmente significativi. Esistono diversi tipi di tokenizzazione che

differiscono per le regole utilizzate nell'individuare i token. In questo caso si è scelto di utilizzare il word tokenizer fornito dalla libreria nltk. Questo tokenizer, divide le stringhe in base allo spazio bianco e alla punteggiatura, la sua qualità principale è il riuscire a riconoscere verbi inglesi in forma abbreviata separati da apostrofi dal loro soggetto, e di dividerli in tokens differenti.

Figura 1. Prime 5 review tokenizzate con successo

```
[i, admit, the, great, majority, of, films, re...
[take, a, low, budget, inexperienced, actors, ...
[everybody, has, seen, back, to, the, future, ...
[doris, day, was, an, icon, of, beauty, in, si...
[after, a, series, of, silly, fun, loving, mov...
                        ...
```

2.5 Lemmatization

La lemmatization è una tecnica che raggruppa le parole in forma flessa e le riduce al rispettivo lemma.

Questa tecnica permette di ridurre la dimensionalità considerando identiche parole scritte in modo diverso, ma con lo stesso significato (es: stesso verbo coniugato con persona diversa), bisogna però prestare attenzione perché in alcuni casi, può portare ad una riduzione dell'informazione .

```
[i, admit, the, great, majority, of, films, re...
[take, a, low, budget, inexperienced, actors, ...
[everybody, has, seen, back, to, the, future, ...
[doris, day, was, an, icon, of, beauty, in, si...
[after, a, series, of, silly, fun, loving, mov...
```

Figura 2. review tokenizzate

```
[i, admit, the, great, majority, of, film, rel...
[take, a, low, budget, inexperienced, actor, d...
[everybody, ha, seen, back, to, the, future, r...
[doris, day, wa, an, icon, of, beauty, in, sin...
[after, a, series, of, silly, fun, loving, mov...
```

Figura 3. review tokenizzate e lemmatizzate

2.6 Rimozione delle stopwords

All'interno di un testo ci sono parole dallo scarso potere informativo che si ripetono spesso, è il caso di congiunzioni, avverbi e preposizioni. Queste parole appesantiscono l'analisi e la disturbano provocando errori non voluti, si è quindi scelto di rimuoverle. Alcune stopwords

però potrebbero portare informazioni in alcuni contesti (es: congiunzioni avversative), per questo la rimozione delle stopwords potrebbe introdurre un bias nel modello.

2.7 Wordcloud [2]

E' una tecnica di visualizzazione di testi, la si è utilizzata per visualizzare i token chiave delle review.

I token che descrivono le review sono raggruppati insieme per formare una nuvola di parole. Ogni parola in questa nuvola ha una dimensione del carattere e una tonalità di colore variabili. Pertanto, questa rappresentazione aiuta a determinare le parole di rilievo. Una dimensione del carattere più grande di una parola ritrae la sua importanza rispetto alle altre.

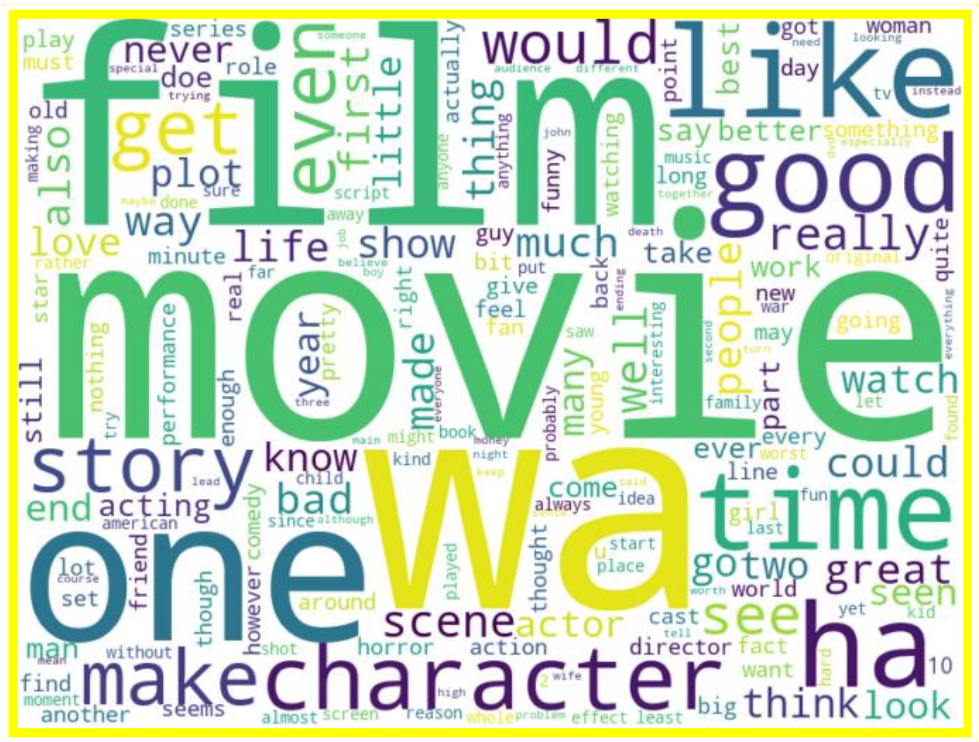

Figura 4. Wordcloud preprocessing

3. Text Representation e Classification

Lo scopo del progetto è quello di affrontare un problema di classificazione binaria, comprendere la sentiment positiva o negativa delle review. Per farlo sono stati addestrati e testati tre diversi modelli di classificazione, Multinomial Naive Bayes, Decision Tree Classifier e Logistic Regression. Il training e il test set sono di uguali dimensioni, 25.000 osservazioni per ciascun set, equamente divise in recensioni positive e negative. Per addestrare un modello di text classification occorre effettuare una text representation che permetta di quantificare il testo trasformando le parole o i gruppi di parole in features che il modello utilizzerà per classificare. Esistono diversi tipi di text representation, si è scelto di addestrare i tre modelli utilizzando tecniche di rappresentazione diverse al fine di trovare la combinazione di text representation e modello di classificazione con le migliori performance predittive. Nello specifico sono state testate le seguenti tecniche di text representation: tf-idf, binary tf - idf, tf - idf bigram, tf - idf trigram, Bag of Words e Bag of Words with N-grams.

Ci si aspetta che le rappresentazioni di tipo Bag of Words si traducano in performance predittive più povere da parte dei modelli. Nel caso di analisi di sentiment come questa, infatti, le Bag of Words representations tendono a non fornire sufficiente informazione. Per questo motivo si è scelto di utilizzare delle text representation che fornissero più informazione al modello. Queste rappresentazioni differenziano l'importanza delle singole parole pesandole con la tf-idf. Nonostante ciò, sono comunque stati provati anche i Bag of Words per verificare empiricamente se, anche con solo due classi di sentiment, non riuscissero a rappresentare informazione sufficiente ad ottenere buoni risultati.

3.1 tf - idf

Attribuisce dei pesi alle parole, dando valore ai termini più rari e caratteristici
tf (term frequency)⇒ Il termine è la misura della frequenza di una parola in un documento. È uguale al numero di istanze della parola nel documento diviso per il numero totale di parole nel documento. La frequenza dei termini serve come metrica per determinare l'occorrenza di una parola in un documento rispetto al numero totale di parole in esso.
idf (Frequenza inversa del documento) ⇒ questo parametro fornisce un valore numerico dell'importanza di una parola. La frequenza inversa è definita come il logaritmo in base 10

del rapporto tra il numero totale di documenti in un corpus testuale e il numero di documenti contenenti la parola d'interesse.

tf-idf ⇒ é il peso attribuito ad una parola. Corrisponde al prodotto tra tf e idf. Solitamente è una delle migliori metriche per determinare se un termine è significativo per un testo. Rappresenta l'importanza di una parola in un particolare documento.

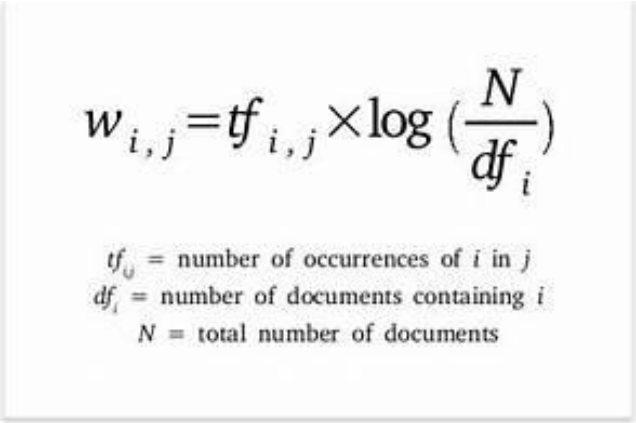

Figura 5. Formula Tf-idf

3.2 binary tf - idf (unsupervised feature weighting)

Tutti i conteggi dei termini diversi da zero sono impostati su 1. Ciò non significa che gli output avranno solo valori 0/1, ma che il termine tf in tf-idf sarà binario.

3.3 bigram tf - idf (unsupervised feature weighting, rappresenta importanza di due parole per un documento)

Un bigram è una sequenza di due elementi adiacenti in una stringa di token , che sono tipicamente parole. Un bigram è un n -gram per n = 2. Accresce la significatività semantica delle features, ma anche la dimensionalità k della rappresentazione risultante e quindi la capacità computazionale richiesta.

3.4 trigram tf - idf (unsupervised feature weighting, rappresenta importanza di tre parole per un documento):

Un trigram è un n -gram per n = 3. In termini di vantaggi e svantaggi è molto simile al bigram tf-idf.

3.5 Bag of Words:

Non considera l'ordine delle parole, costruisce una matrice che ha una colonna per ogni parola presente almeno una volta nell'insieme di documenti e una riga per ciascun documento. Se in un documento una parola viene rappresentata almeno una volta, allora le viene associato il valore 1, altrimenti le viene associato il valore 0.

3.6 Bag of Words with N-grams:

Ciascuna feature/colonna della matrice è costituita da una sequenza di N tokens contenuti in una parte di testo. Conserva l'informazione posizionale e riesce a cogliere la dipendenza funzionale tra le parole, tuttavia fornisce una visione statistica, accresce la dimensione del vocabolario risultando in una matrice più sparsa. Sulle righe si trovano sempre i documenti, quando in un documento è presente una sequenza, allora in quella colonna sarà presente il valore 1, altrimenti si troverà il valore 0.

3.7 Risultati classification:

Per tutte le tecniche di text representation si è deciso di applicare modelli differenti di classificazione, in modo da confrontarli e trovare il migliore a livello di performance.
I classificatori usati sono:

- `MultinomialNB()`: Il modello multinomiale Naive Bayes apprende dalle occorrenze tra funzionalità come il conteggio delle parole e le classi discrete. Il vettore di input deve contenere valori positivi.
- `DecisionTreeClassifier()`: gli alberi decisionali sono un metodo di apprendimento supervisionato non parametrico utilizzato per la classificazione. L'obiettivo è creare un modello che preveda il valore di una variabile target apprendendo semplici regole decisionali dedotte dalle caratteristiche dei dati. Un albero può essere visto come un'approssimazione costante a tratti.
- `LogisticRegression()`: La regressione logistica è un processo di modellazione della probabilità di un risultato discreto. Il modello di regressione logistica utilizza un risultato binario. La regressione logistica è un metodo di analisi utile per i problemi di classificazione.

Con tf - idf representation:

MultinomialNB				DecisionTreeClassifier				Logistic Regression			
Train Accuracy	Test Accuracy	Test Precision	Recall	Train Accuracy	Test Accuracy	Test Precision	Recall	Train Accuracy	Test Accuracy	Test Precision	Recall
0.8397	0.8348	0.8316	0.8395	1.0000	0.7049	0.7069	0.7024	0.8736	0.8605	0.8544	0.8691

Il modello di classificazione più efficace in questo caso sembra essere la Logistic Regression, che ha le migliori performance su tutte le metriche considerate e si adatta bene sia ai dati del training set che a quelli del test set. I suoi buoni valori di Recall e Precision, inoltre, suggeriscono che riesca a contenere sia il numero di falsi positivi che il numero di falsi negativi.
Si è notato, inoltre, che la Multinomial NB ha un buon comportamento, mentre il Decision Tree è in chiaro overfitting in quanto ha un'accuratezza del 100% sul training che crolla al 70% sul test. Ciò potrebbe essere causato dall'inadeguatezza del modello al caso o da un setting non ottimale dei parametri del modello.

Con binary tf - idf representation:

MultinomialNB				DecisionTreeClassifier				Logistic Regression			
Train Accuracy	Test Accuracy	Test Precision	Recall	Train Accuracy	Test Accuracy	Test Precision	Recall	Train Accuracy	Test Accuracy	Test Precision	Recall
0.8429	0.8370	0.8323	0.8442	1.0000	0.6988	0.6983	0.7002	0.8743	0.8610	0.8543	0.8703

Il modello migliore è ancora la Logistic Regression, che migliora in tutte le metriche ad eccezione della precision, che registra un peggioramento infinitesimale. La Multinomial NB registra anch'essa un miglioramento in tutte le metriche, mentre il decision tree continua ad overfittare e la performance peggiora.

Con bigram tf - idf representation:

MultinomialNB				DecisionTreeClassifier				Logistic Regression			
Train Accuracy	Test Accuracy	Test Precision	Recall	Train Accuracy	Test Accuracy	Test Precision	Recall	Train Accuracy	Test Accuracy	Test Precision	Recall
0.7684	0.7509	0.7520	0.7487	0.9850	0.6730	0.6747	0.6681	0.7798	0.7554	0.7460	0.7746

Il modello migliore continua ad essere la Logistic Regression per tutte le metriche considerate, le performance, però, risultano nettamente peggiori rispetto a quelle ottenute con le text representation precedenti.

Con trigram tf - idf representation:

MultinomialNB				DecisionTreeClassifier				Logistic Regression			
Train Accuracy	Test Accuracy	Test Precision	Recall	Train Accuracy	Test Accuracy	Test Precision	Recall	Train Accuracy	Test Accuracy	Test Precision	Recall
0.6396	0.6049	0.7110	0.3534	0.7104	0.6165	0.5810	0.8356	0.6594	0.6298	0.5888	0.8605

Tutti i modelli registrano un ulteriore peggioramento delle performance su training e test. Il modello migliore risulta essere ancora la Logistic Regression.

Con Bag of Words representation:

MultinomialNB				DecisionTreeClassifier				Logistic Regression			
Train Accuracy	Test Accuracy	Test Precision	Recall	Train Accuracy	Test Accuracy	Test Precision	Recall	Train Accuracy	Test Accuracy	Test Precision	Recall
0.8342	0.8312	0.8321	0.8272	1.0000	0.7031	0.7066	0.6946	0.8746	0.8572	0.8503	0.8670

Il modello migliore rimane la Logistic Regression, registra buone performance, ma inferiori a quelle rilevate con una representation di tipo binary tf - idf. Il Decision Tree è ancora in overfitting.

Con Bag of Words with N-grams representation:

MultinomialNB				DecisionTreeClassifier				Logistic Regression			
Train Accuracy	Test Accuracy	Test Precision	Recall	Train Accuracy	Test Accuracy	Test Precision	Recall	Train Accuracy	Test Accuracy	Test Precision	Recall
0.8410	0.8366	0.8316	0.8352	1.0000	0.7012	0.7032	0.6958	0.8768	0.8560	0.8493	0.8658

Nuovamente le performance migliori si registrano con la Logistic Regression su tutte le misure, la Multinomial NB ha buone prestazioni, ma rimane inferiore alla Logistic, mentre il Decision Tree risulta ancora in overfitting e con performance abbastanza povere.

Complessivamente la combinazione migliore sembra essere una Logistic Regression addestrata usando dati rappresentati in forma di binary tf - idf (test accuracy 0,8610). Notiamo che i due modelli costruiti su una rappresentazione Bag of Words e Bag of Words with N-grams mostrano un maggiore adattamento ai dati di training (training accuracy 0,8746 e 0,8768), ma performance peggiori sul test rispetto alla binary tf-idf. Nonostante lo scetticismo iniziale, dunque, le rappresentazioni di tipo Bag of Words non sembrano così inadeguate a questo tipo di classificazione, forse anche per via della forte polarità delle recensioni analizzate, che riduce le sfumature da cogliere nel testo richiedendo meno informazione per classificare correttamente.
In questo studio non è stato effettuato un tuning ottimale dei parametri. La soglia di features massime selezionata nella text representation è piuttosto alta e non sono stati esclusi i termini potenzialmente troppo frequenti o troppo poco frequenti, inoltre i modelli sono stati applicati con i parametri standard. Costruire una funzione che setti gli iperparametri cercando di massimizzare le performance potrebbe portare a migliorare le prestazioni dei modelli, separando meglio il segnale dal rumore e minimizzando distorsione e varianza per ridurre gli errori di classificazione sul test.
Avendo a disposizione maggiore capacità computazionale si potrebbe provare ad addestrare un modello random forest, mentre per una rete neurale potrebbero essere necessari più dati in fase di training.

4. Topic modelling

Il topic modelling si occupa di individuare argomenti astratti intrinseci in documenti di testo utilizzando modelli probabilistici.

Nel caso di analisi si è optato per l'utilizzo dell modello LDA (Latent Dirichlet Allocation).

LDA è una popolare tecnica di modellazione di topic utilizzata per estrarre argomenti da un insieme di testi. Questa tecnica tratta i documenti come bag of words e assume che siano prodotti da un mix di argomenti, categorizzando i documenti per argomento con un modello probabilistico generativo. Partendo dall'ipotesi che i documenti siano una distribuzione di topics e i topics siano una miscela di parole, l'assunto fondamentale di questa tecnica riguarda la distribuzione dei topic in ciascun documento e delle parole in ciascun topic.

Entrambe queste distribuzioni vengono considerate equivalenti a distribuzioni di Dirichlet. Questo assunto comporta che i topics correlati agli argomenti siano pochi e la disambiguazione tra le parole sia migliore. Nel LDA il parametro di concentrazione della distribuzione è sempre considerato sparso (<1) sia per i topic che per le parole.

Il modello di Dirichlet descrive lo schema delle parole che si ripetono insieme, che ricorrono frequentemente, e raggruppa queste parole considerandole simili tra loro e potenzialmente relative allo stesso topic.

L'iperparametro principale in questo modello è il numero di topics che si vogliono identificare. Nel caso di analisi si è deciso di impostare il numero di topics da individuare a 8 e si sono escluse le parole troppo rare, con meno di 5 ripetizioni nei documenti, e quelle troppo comuni, presenti in più della metà dei documenti.

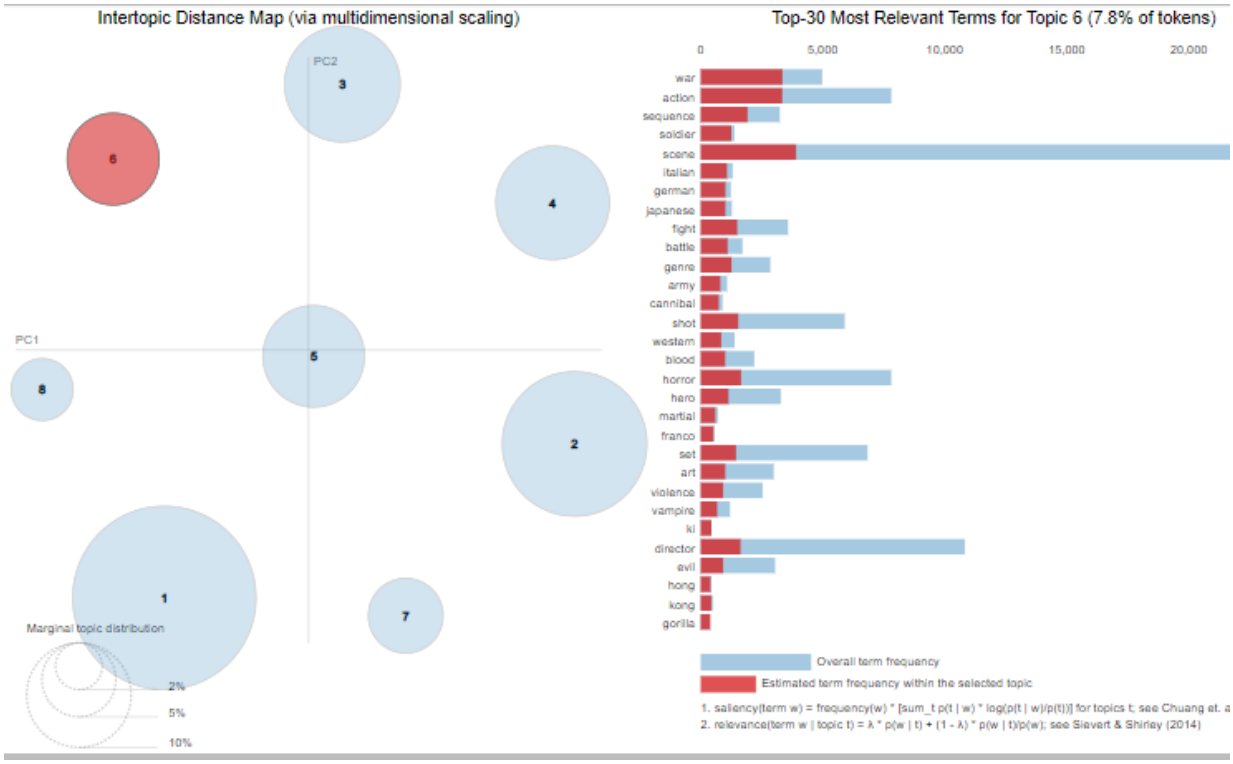

Figura 6. Output Topic Modeling

Il topic modelling è una tecnica di machine learning non supervisionata, per valutarne l'efficacia si sono utilizzate sia metriche intrinseche, che misurano la coerenza all'interno dei topic, sia una valutazione umana. La sola valutazione statistica della coerenza dei topics e della vicinanza dei loro termini, infatti, potrebbe far apparire significativi dei cluster ininterpretabili.

Da una prima analisi euristica i topic creati risultano sensati, si possono chiaramente distinguere argomenti quali, film d'amore, di guerra, gialli, commedie, musical. Appurata l'interpretabilità dei cluster individuati si è passati alla valutazione tramite metriche intrinseche utilizzando la Perplexity e il Coherence score.

- Perplexity, è una misura di incertezza, una metrica statistica che valuta quanto bene un modello probabilistico predice un campione. Punta a catturare quanto un modello sia sorpreso da nuovi dati. Più è bassa migliore è il modello. Nel caso di analisi è risultata -8.4965, il modello quindi sembra performare adeguatamente.
- Coherence Score (misurato con la c_v), misura la similarità semantica tra le top words del topic. Maggiore è la coerenza, migliori sono le performance del modello. Assume valori nel range [0,1]. Nel caso in analisi risulta 0.3131, segno che il modello non è perfetto e probabilmente migliorabile. Per farlo si potrebbe adottare una funzione di

parameter tuning volta a trovare il numero di topic ideale, qui trascurata per carenza di potenza di calcolo.

5.Conclusioni

Come dimostrato nel report Il classificatore migliore risulta essere la Logistic Regression, per tutte le text representation, Multinomial NB ha delle buone performance, ma inferiori, mentre il Decision Tree purtroppo scivola nell'overfitting.
Le text representation bigram e trigram in tf-idf non ottengono buoni risultati e sono quindi da escludersi.
I due bag of words e i tf-idf standard e binario, sono le text representation migliori da usare e si attestano tutte su valori molto simili di performance.
Per quanto riguarda il topic modelling è stata impostata la ricerca di 8 topics. A livello di perplessità il modello pare performare adeguatamente, mentre a livello di coerenza si potrebbe migliorare. Nonostante questo, la buona interpretabilità dei cluster creati permette di guardare con ottimismo ai risultati ottenuti ritenendoli più che accettabili.

6.Sviluppi Futuri

Possibili sviluppi futuri a livello di classificazione possono sicuramente riguardare l'implementazione dell'analisi con altri modelli che non sono stati utilizzati quali random forest, svm (mancanza di capacità computazionale), neural network (probabilmente pochi dati).
Si potrebbe anche provare a ragionare in termini di Luhn's analysis per evitare che parole poco influenti interferiscano con le analisi, e impostare attraverso un parameter tuning il numero di features che ottimizza la modellazione.
Per quanto riguarda il topic modelling, un sicuro sviluppo futuro di interesse potrebbe essere quello di utilizzare anche qui funzioni di parameter tuning per trovare il perfetto numero di topics.

Sitografia:

1. collegamento ipertestuale al sito analyticsindiamag per estrazione dei dati https://analyticsindiamag.com/10-open-source-datasets-for-text-classification/
2. approfondimento wordcloud https://www.geeksforgeeks.org/generating-word-cloud-python/