AMAZON REVIEWS

University of Milano-Bicocca

Master's Degree in Data Science

Boschi Giulia - 804623

Ravazzi Lucia - 852646

Spandri Michelle - 819362

Sommario

1. SCOPO DEL PROGETTO

Lo scopo del progetto è quello di approfondire le metodologie per il processamento del linguaggio naturale e le tecniche di machine learning ad esso connesse. In particolar modo, abbiamo scelto di progettare e sviluppare diverse tecniche di rappresentazione del vocabolario in diversi task quali classificazione, clustering e topic modeling.

2. DESCRIZIONE DEL DATASET

Il dataset utilizzato è stato ottenuto da [1] ed è costituito da un corpus di recensioni di prodotti selezionati dalla piattaforma Amazon. Il dataset è stato già diviso in training e test set delle dimensioni di 3.600.000 e 400.000 documenti. Ogni documento presenta la seguente struttura:

__label__k titolo: testo

dove $\mathrm{k} \in \{1,2\}$ e se $k = 1$, le stelle associate al commento sono o una o due, altrimenti quattro o cinque. Si noti che questi commenti possono essere interpretati rispetto ad una polarità binaria: il prodotto è stato gradito o meno dal cliente. Inoltre, non è fornita alcuna indicazione rispetto allo stato neutro, ovvero le tre stelle. La maggior parte dei commenti sono scritti in lingua inglese e ci siamo limitate a considerare solo questi.

3. PREPROCESSING INIZIALE

Per poter lavorare con queste recensioni, è obbligatorio trasformare i dati testuali in un altro formato per poter essere manipolate dal calcolatore. Noi abbiamo scelto il modello relazionale per la sua compatibilità con molte funzioni implementate in Python. Tuttavia, per poter utilizzare tale formato è necessario possedere risorse computazionali non indifferenti dato che le tabelle che si ottengono per questi dati sono sparse e caratterizzate da molti campi. Di conseguenza, immaginare di poter implementare questo progetto con tutti i dati forniti, non è fattibile con le risorse a nostra disposizione. Per superare la saturazione immediata della RAM, abbiamo campionato sia il dataset di training sia quello di test. Poiché le etichette sono bilanciate in

entrambi i dataset, il campionamento è di tipo casuale. Si è scelta una dimensione del dataset di training gestibile per campionare il dataset di test con la medesima proporzione.

In seguito, abbiamo voluto trasformare la struttura contenente le recensioni. Infatti, fino a questo punto abbiamo memorizzato i documenti in semplici liste che però, non sono particolarmente efficienti. Di conseguenza, abbiamo considerato ogni commento, creando un dataframe caratterizzato da due colonne: una per il testo ed il titolo ed una seconda per l'etichetta. In questo modo, la libreria pandas ci offrirà una performance ed una struttura migliore rispetto al caso precedente.

A questo punto, implementiamo la pipeline per trasformare i documenti del corpus in liste token normalizzati. I passaggi implementati sono i seguenti:

- Tutte le parole sono state convertite in minuscolo.
- I numeri, la punteggiatura, i simboli \n, gli url, le emoji e gli spazi extra sono stati eliminati.
- Ogni documento è stato tokenizzato.
- Ogni termine è stato ricondotto al proprio lemma. Questa scelta è stata dettata dal fatto che ricondurre alla radice (stemming) ogni termine avrebbe potuto rendere meno interpretabili alcuni risultati per alcuni task. Inoltre, questa procedura fornisce l'opzione di poter considerare anche il POS delle parole. Tuttavia, si è scelto di non implementarla poiché richiede parecchio tempo.
- Tutti i token caratterizzati da parole molto brevi sono stati eliminati ed anche le stop words.

Adesso che i dati sono stati opportunamente processati, è arrivato il momento di discutere le rappresentazioni doc2vec scelte per i dati.

4. RAPPRESENTAZIONI

A questo punto, abbiamo creato un corpus di documenti tokenizzati e normalizzati. Vogliamo quindi costruire un vocabolario, ovvero un insieme di N parole chiave che catturino le caratteristiche principali dei documenti. Per capire meglio le performance dei nostri algoritmi nei vari tasks, abbiamo

pensato di costruire diversi tipi di vocabolari, ovvero caratterizzati da 1-gram, 1-gram e 2-gram ed infine, solo da 2-gram. Inoltre, sono state utilizzate in modi differenti le seguenti rappresentazioni:

- Bag of words con pesi binari.
- Bag of words con frequenze.
- Tf_idf

Queste sono tutte le possibili rappresentazioni utilizzate nel progetto, ma per ogni task verranno esplicitamente definite e descritte nei paragrafi seguenti. Passiamo ora alla descrizione dei tasks da noi implementati.

5. CLASSIFICAZIONE

Il primo task implementato è quello della classificazione. Infatti, ogni documento del corpus ha un'etichetta binaria che ci permette di sviluppare tali tecniche. Lo scopo di tale lavoro è quello di categorizzare una nuova recensione sulla base di quanto un cliente possa aver gradito il prodotto acquistato. Questo è particolarmente utile quando un cliente si dimentica di inserire il numero di stelle associate alla recensione data oppure, più in generale, per etichettare una recensione di un prodotto. Di conseguenza, questo lavoro può essere utilizzato anche per generiche recensioni di articoli simili a quelli proposti da Amazon.

Per implementare questo task, abbiamo voluto utilizzare più approcci: classificazione pura, sentiment supervised analysis e l'utilizzo delle reti neurali. Il primo si riferisce ad una tradizionale classificazione che, in generale, viene utilizzata per implementare la categorizzazione dei topic. Tuttavia, abbiamo voluto testare quanto queste tecniche potessero essere valide anche nel nostro contesto. Il secondo approccio utilizza delle tecniche di sentiment supervised analysis proposte più per i Social Media. Infine, l'ultimo metodo utilizza le librerie Bert e FastText.

5.1. CLASSIFICATIONE PURA

Prima di tutto è necessario puntualizzare che il numero di recensioni utilizzate per questo task è di 250.000 in totale ed il preprocessing sviluppato è quello descritto nel paragrafo 3.

Abbiamo utilizzato tutte e tre le rappresentazioni descritte nel paragrafo precedente ed inoltre, abbiamo provato a modificare il vocabolario utilizzando 1-gram, 1-gram e 2-gram e solo 2-gram. La rappresentazione forniva già di per sé un vocabolario dalla quale abbiamo eliminato le parole con un df molto basso (feature extraction). Inoltre, per diminuire la dimensione delle tabelle (feature selection), abbiamo calcolato il X^2 tra le feature e l'attributo target selezionando solo il top 10% della distribuzione, ovvero le feature che presentavano una maggior connessione con le etichette. Ora che abbiamo ricavato le nove diverse tabelle, possiamo capire qual è la migliore rappresentazione per i seguenti algoritmi:

- k-nearest neighbors (k=5)
- Support Vector Machine con kernel lineare
- Regressione logistica
- Linear classifier with stochastic gradient descent learning
- Adaboost
- Naïve Bayes (Multinomial e Bernullian)

La scelta di questi algoritmi è motivata dal fatto che possono supportare il formato *sparso* delle matrici il quale memorizza solo gli elementi non nulli ed i rispettivi indici. Ogni algoritmo è stato implementato attraverso una procedura di cross validation con 5 folds e le misure di performance scelte sono l'accuratezza, la F1, il tempo di training e di test. Ci siamo focalizzate sull'accuratezza perché le etichette sono bilanciate quindi tale misura è sufficiente per analizzare le performances. Per ognuna di tali misure abbiamo calcolato la media e l'intervallo di confidenza al 95%.
Analizzando tutti i possibili casi, si osserva che la rappresentazione migliore è la tf_idf con un vocabolario costituito da sia 1-gram e 2-gram con l'algoritmo Support Vector Machine con un kernel di tipo Lineare. Questa valutazione tiene in considerazione sia le misure di accuratezza e F1 ma anche la velocità e l'efficienza dell'algoritmo. Per evitare di affollare il report con tutte le misure di performance, riporto i tre migliori algoritmi e le caratteristiche della rappresentazione in Appendice.

5.2. SENTIMENT SUPERVISED ANALYSIS

Abbiamo notato che i risultati ottenuti dagli algoritmi precedenti sono molto buoni per il dataset fornito. Tuttavia, tale approccio è indicato più per una classificazione del topic piuttosto che un'analisi del sentimento che dovrebbe

essere investigata con altri algoritmi più specifici. A tal proposito, utilizzeremo delle tecniche note nel campo Social Media Analytics che potrebbero aiutarci nel capire la polarizzazione del cliente. Gli algoritmi utilizzati sono i seguenti:

- Afinn: offre un vocabolario in cui ogni termine ha un peso prefissato in [-5,5]. Per ogni recensione verrà quindi calcolato uno score dato dalla somma dei pesi delle parole che la costituiscono.
- Opinion lexicon: Tale algoritmo offre una lista di parole positive e negative dando un peso unitario a tutte queste. Di conseguenza, la polarità di ogni recensione sarà data dalla somma del numero totale di parole positive meno quelle negative.
- Vader: offre anch'esso un vocabolario che permette di calcolare un score tra [-1,1].

A questo punto, per ogni recensione abbiamo calcolato una polarità che però deve essere ricondotta a positivo/negativo per poter essere confrontata con l'etichetta fornita nel dataset. Prima di tutto, tali algoritmi considerano anche lo stato neutro che non è contemplato nelle nostre etichette. Di conseguenza, per ognuno dei tre algoritmi, abbiamo considerato una soglia ragionevole **k** per etichettare le recensioni a partire dalla polarità **p**:

$$\begin{cases} p < -k \quad negativo \\ -k \leq p \leq k \;\; neutro \\ p > k \;\; positivo \end{cases}$$

Tutte le recensioni che sono state etichettate come neutre sono state eliminate poiché non potevano essere confrontate con quelle fornite. Il numero di queste è una percentuale minima rispetto a tutte quelle considerate per tutti e tre i casi. Infine, abbiamo ottenuto una matrice di confusione che ci permettesse di confrontare le etichette così ottenute con quelle date. Possiamo affermare che i risultati dipendono dal valore **k** scelto anche se sembrano abbastanza robusti in seguito ad una sua modifica. Il risultato migliore è stato ottenuto con il secondo algoritmo: opinion lexicon. I risultati sono riportati in appendice.

5.3. RETI NEURALI PER NLP

Per esplorare il maggior numero di tecniche per la classificazione, abbiamo voluto implementare anche i noti algoritmi Bert e FastText.

BERT (*Bidirectional Transformers for Language Understanding*) è una rete neurale pre-addestrata che permette di modificare gli ultimi livelli della rete per adattarsi ai nostri dati. Le recensioni verranno processate opportunamente per poter essere successivamente utilizzate dalla rete.
FastText è stato introdotto da parte di Facebook nel 2015. Tale algoritmo implementa tecniche di word embedding. L'aspetto interessante di queste reti è che non utilizzano la tradizionale softmax per la classificazione ma la hierarchical softmax, ovvero una struttura gerarchica che diminuisce notevolmente il tempo di calcolo per la fase di training. Infatti, come mostrato da [3], tale algoritmo presenta le stesse misure di performance delle reti neurali tradizionali ma il tempo di computazione crolla. Abbiamo voluto semplicemente capire come funzionano questi algoritmi senza entrare troppo nel dettaglio. Le accuratezze per l'algoritmo BERT e FastText sono rispettivamente 95,7% e 90,5% con 250.000 documenti totali. Si potrebbe andare più a fondo ed introdurre anche un'incertezza di queste misure. Tuttavia, il nostro scopo era solo quello capire come migliorare le performance capendo l'importanza e l'utilità di algoritmi basati sulle reti neurali.

5.4 CONCLUSIONI

Tutti e tre gli approcci hanno dato ottimi risultati in termini del set di metriche utilizzate per la valutazione. Tuttavia, le tecniche di classificazione pure sono le migliori per questo task principalmente per i ragionevoli tempi di computazione richiesti.

6. CLUSTERING

Il secondo task che abbiamo voluto implementare è stato quello del clustering. I nostri obiettivi sono quelli di:

- Ricostruire le recensioni con label positive e negative già fornite dal dataset;
- Ricostruire la votazione in stelle (da 1 a 5) associata ad ogni recensione e non fornita nel dataset.

Per tali scopi si sono implementati diversi algoritmi: k-means con k ={2, 5}, DBSCAN e HDBSCAN.

6.1. PREPROCESSING

Per questo task la procedura di pulizia dei dati è stata modificata rispetto a quella descritta nella fase di preprocessing al paragrafo 3. La decisione di adottare le modifiche di seguito illustrate è stata guidata dalla scarsa qualità dei risultati che si ottenevano altrimenti. Innanzitutto, i modelli sono stati stimati su subset di 40.000 recensioni in quanto algoritmi come DBSCAN e, soprattutto, HDBSCAN richiedono tempi di computazione elevati.

Ciò che differenzia la pulizia delle recensioni in questo task rispetto al precedente è, principalmente, la decisione di mantenere solo i termini indicanti un sentimento escludendo ciò che riguardava il topic: infatti, i cluster avrebbero messo in risalto il topic anziché il sentimento. Questo è stato possibile grazie al dizionario di termini presente nella libreria opinion_lexicon in Python, descritta nel paragrafo di classificazione supervisionata. Inoltre, durante la rimozione delle stopwords, si è deciso di non rimuovere quelle indicanti una negazione ('not', 'don't', 'can't…), e di sfruttarle per indicare il caso in cui il sentimento sia negato. Sono state quindi costruite delle forme 'not_parola' nel caso in cui la parola in oggetto fosse preceduta da una negazione. Si è proceduto costruendo anche delle forme 'top_parola' qualora la parola fosse preceduta da un termine che ne amplificasse il significato come: 'really', 'very', 'many', 'too'… Le strutture così costruite verranno considerate come un unico gram anche se formate da due termini.

Per ripulire maggiormente da parole noise si è deciso di eliminare anche la forma 'as well as' per evitare che 'well' fosse mantenuto come sentimento, quando invece ha solo funzione comparativa.

Le recensioni così pulite sono state rappresentate sfruttando i modi già descritti (TF-IDF, BOW con pesi binari e BOW con frequenze) e con diverse combinazioni di n-gram (1-gram, 1 e 2-gram, 2-gram). La feature extraction, come nel caso della classificazione, ha permesso di escludere termini con valori troppo bassi di df. La feature selection, considerato il task di classificazione non supervisionata, non poteva basarsi sulla stima della χ^2 per cui si è proceduto escludendo le feature che non superavano una certa soglia di varianza **k** fissata. Tale soglia è stata modificata di volta in volta, in modo sperimentale, in base alla rappresentazione utilizzata. Infatti, la stessa soglia, su due rappresentazioni diverse, restituiva un numero di feature estremamente diverso. Le soglie usate sono state scelte al fine di ottenere un numero abbastanza stabile di feature in ogni rappresentazione.

6.2. K-MEANS

6.2.1. K = 2

L'applicazione dell'algoritmo delle k-medie con k = 2 è l'unico caso in cui è possibile applicare delle metriche di valutazione supervisionata grazie al confronto con le label contenute nel data frame di partenza. In generale, si osserva che l'algoritmo ottiene i migliori risultati sulla rappresentazione che contiene solo 2-gram. In particolare, la silhouette più elevata si ottiene sulla rappresentazione BOW con pesi binari, ma la rappresentazione BOW con frequenze ottiene un valore di silhouette poco inferiore e valori più alti di omogeneità, completezza, v-measure e Rand-index aggiustato.

Visualizzando in due dimensioni i risultati dell'algoritmo, sfruttando la PCA, applicato sulle due diverse rappresentazioni appena citate si ottiene:

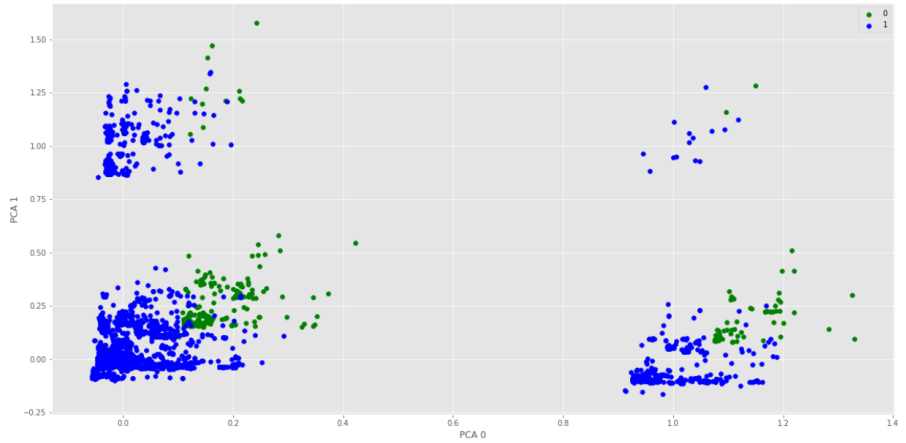

Figura 1 - Rappresentazione BOW con pesi binari, 2-gram

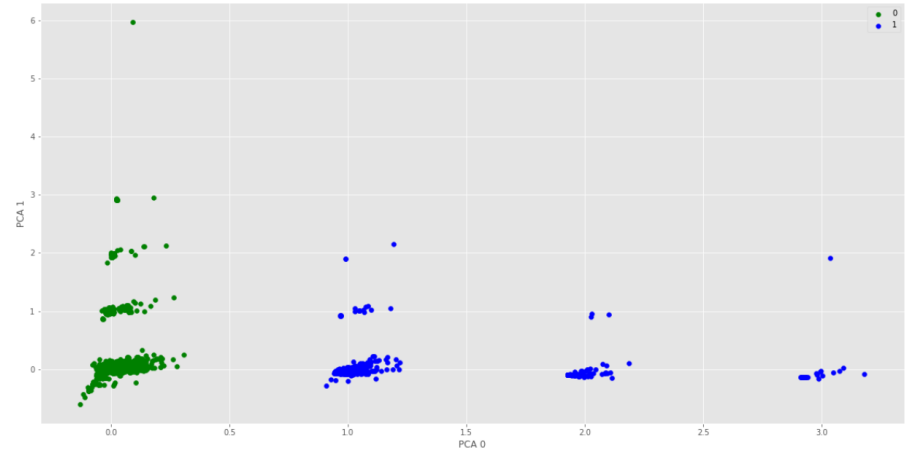

Figura 2 - Rappresentazione BOW con frequenze, 2-gram

Si approfondisce ora il risultato ottenuto sfruttando la rappresentazione BOW con frequenze e 2-gram. Osservando **le parole più frequenti all'interno delle recensioni** etichettate come cluster 0 o 1, è difficile stabilire in modo netto quale dei due contenga recensioni positive e quale negative, inoltre, in entrambi i cluster, si trovano stesse combinazioni di parole. Però, nel cluster 0 i più frequenti 2-gram sono 'excellent enjoy', 'complicated like' e 'creative like', mentre nel cluster 1 sono 'break not_recommend', 'cheap enough' e 'break useless'. Si noti che in realtà alcuni gram del vocabolario non sono propriamente 2 ma bensì 3: questo deriva dalla modifica iniziale spiegata in precedenza. Questa analisi fa propendere per una associazione del cluster 0 alle recensioni maggiormente positive e il cluster 1 a quelle principalmente negative.

6.2.2. K = 5

In questo caso non è più possibile sfruttare metriche di valutazione esterne, ma ci si può basare solo sul valore di silhouette. Anche in questo caso si ottengono risultati migliori sulle rappresentazioni che contengono strutture 2-gram e che quindi permettono di considerare maggiormente il contesto. Le rappresentazioni con 2-gram ottengono valori di silhouette superiori rispetto alle rappresentazioni con 1-gram o 1/2-gram, soprattutto utilizzando la rappresentazione BOW con pesi binari.

Si osserva inoltre che, in generale, la distribuzione delle recensioni all'interno dei cluster è piuttosto sbilanciata. In tutti i casi si crea un cluster estremamente numeroso, contenente più della metà delle recensioni, seguito da cluster via via più piccoli. Questa tendenza si accentua nel caso di rappresentazioni che usano 2-gram. Segue rappresentazione sfruttando le PCA:

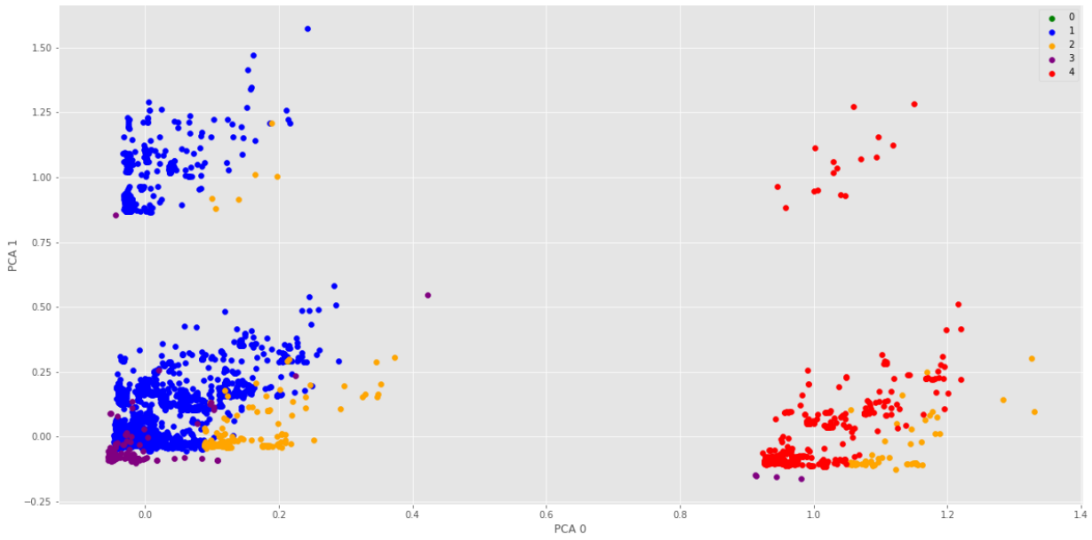

Figura 3 - Rappresentazione BOW con pesi binari, 2-gram

Analizzando le parole che più frequentemente ricorrono nelle recensioni incluse nei diversi cluster si denota lo stesso problema osservato nel caso di k = 2: alcune strutture 2-gram compaiono in più cluster ed è difficile stabilirne con esattezza il sentimento. Si è giunti alla seguente associazione delle valutazioni, si riportano i 2-gram più frequenti:

- Cluster 0 – 2 stelle: 'break terrible', 'break support', 'amazing happy';
- Cluster 1 – 5 stelle: 'clog clean', 'blame good', 'bad static';
- Cluster 2 – 1 stella: 'bad not_like', 'bad nice', 'beware free';
- Cluster 3 – 3 stelle: 'boring slow', 'boring not_much', 'boring repetitive';

- Cluster 4 – 4 stelle: ‘bad nice’, ‘bad static’, ‘classic funny’

6.3. DBSCAN e HDBSCAN

Per ogni diversa rappresentazione si sono individuati valori diversi per i parametri richiesti dagli algoritmi (**ε**, min_samples per DBSCAN e min_samples, min_cluster_size per HDBSCAN), in modo da ottenere la migliore possibile per il nostro obiettivo.

La potenza di questi algoritmi è quella di escludere quelle osservazioni che non sono associate ad alcun cluster (che vengono inserite nel cluster -1) e permettono quindi di ottenere cluster più puliti.

L’unica valutazione che si è ritenuta efficace in questa applicazione è quella manuale, grazie sia alla visualizzazione dei cluster (sfruttando la PCA) che all’analisi delle parole che più frequentemente compaiono nelle recensioni associate ad ogni cluster.

In generale si osserva che si verifica sempre più spesso il fenomeno osservato anche nel caso di k-medie, cioè la ripetizione degli stessi gram in cluster diversi. Si osserva ancora anche lo squilibrio della distribuzione delle recensioni nei diversi cluster: in queste applicazioni è il cluster delle recensioni outlier ad avere, generalmente, la numerosità maggiore.

6.3.1. DBSCAN

La miglior applicazione individuata è quella sulla matrice BOW di frequenze che considera sia 1-gram che 2-gram. Si individuano due cluster che contengono poche recensioni rispetto al bacino totale, ma la divisione delle parole all’interno di questi è l’unica che ci permette di discernere tra una classe più positiva ed una più negativa.

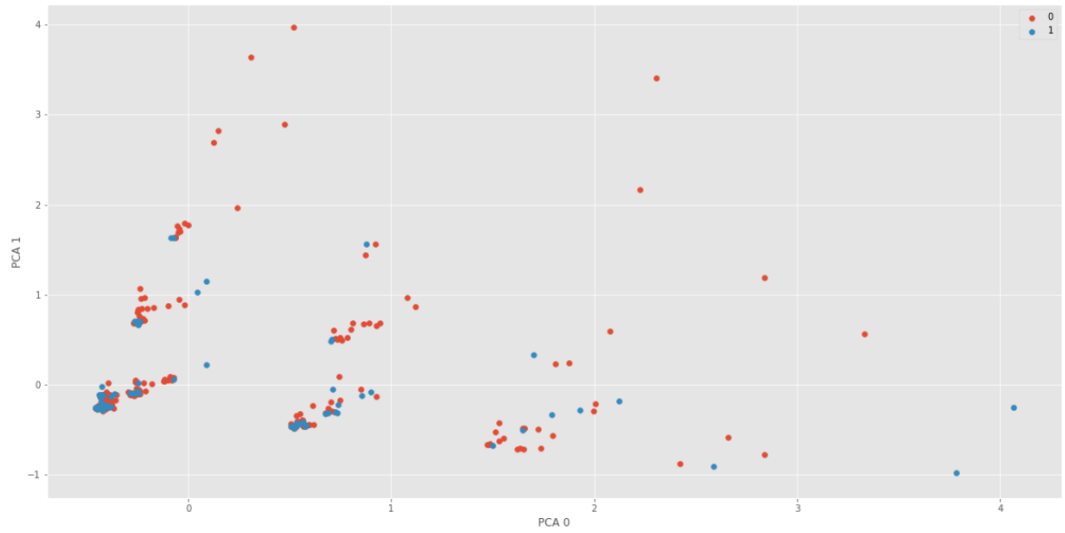

Figura 4 - Rappresentazione BOW con frequenze, 1-2gram, escludendo il cluster -1

I due cluster sono stati identificati come:

- Cluster 0 – maggiormente negativo: infatti contiene principalmente strutture come 'sensible', 'naughty', 'not_recieve', 'not_enjoyable';
- Cluster 1 – maggiormente positivo: che contiene 'good', 'top_saving', 'mediocrity ', 'top_awesome'.

6.3.2. HDBSCAN

Applicando questo algoritmo alle diverse rappresentazioni si osserva che è molto difficile ottenere 2 o 5 cluster, se non imponendo valori molto alti ai parametri. Inoltre, le prime parole più frequenti in ogni cluster sono sempre uguali ('good', 'like', 'well', 'work', 'love').

Al contrario delle applicazioni con DBSCAN, spesso il cluster contenente outlier non è il più numeroso e le recensioni sono distribuite in modo un po' più uniforme all'interno dei cluster.

L'applicazione che ottiene dei risultati che ci permettono di distinguere cluster di recensioni più o meno positive è quella sulla matrice TF-IDF ottenuta considerando solo 2-gram.

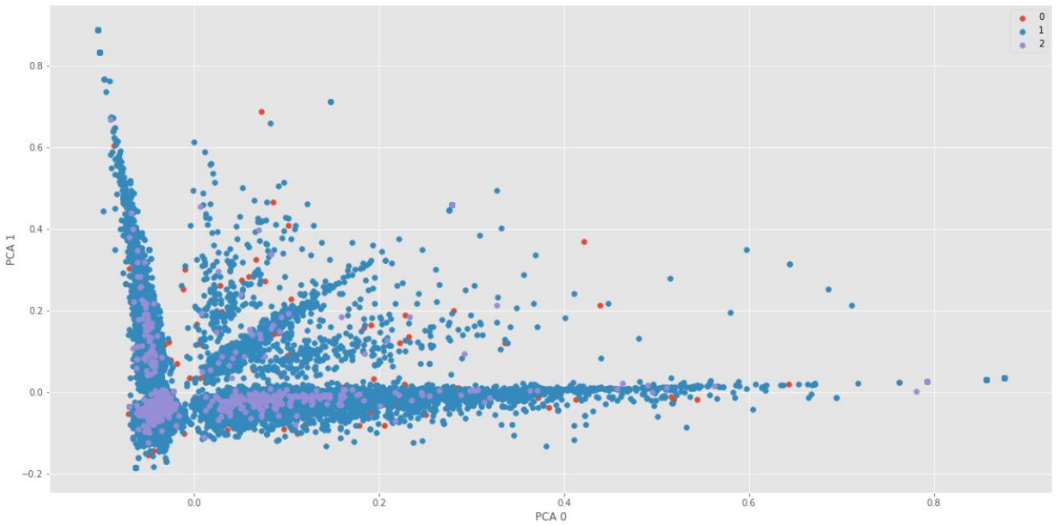

Figura 5 - Rappresentazione TF-IDF, 2-gram, escludendo il cluster -1

In questo caso i cluster 0 e 1 contengono recensioni in cui le parole più frequenti sono le stesse e tendenzialmente positive ('excelent' e 'enjoy') mentre il cluster 2 viene definito come negativo in quanto contiene frequentemente la parola 'problem'. Però, come indicato nel caso generale, anche in questo caso si presentano, tra le parole più frequenti di tutti e tre i cluster, le stesse parole: 'good', 'like', 'well', 'love', 'work', 'bad', 'recommend'.

6.4. CONCLUSIONI

In conclusione, l'applicazione che meglio ci permette di distinguere recensioni positive e negative è quella delle k-medie, con k = 2, sulla matrice BOW con frequenze considerando le strutture 2-gram. In generale questi algoritmi di clustering non hanno restituito risultati ottimi. In futuro si potrebbe provare a pulire in modo ancora diverso i documenti di partenza o testare diversi algoritmi.

7. TOPIC MODELING

Come terzo task si è scelto di implementare la tecnica di topic modeling, al fine di identificare le categorie di prodotti relative alle diverse recensioni. L'obiettivo di questo approccio è quello di ottenere un modello che consenta di individuare in maniera automatica la categoria di prodotti discussa in ciascuna recensione. Questo modello di topic modeling è utile se si è interessati a conoscere le categorie di prodotti alle quali si riferiscono le recensioni. Una possibile applicazione consiste nella realizzazione di un menù a tendina per Amazon sulla base delle categorie di prodotti identificate.

Si è scelto di implementare le due principali tecniche presenti nella letteratura per performare topic modeling, ovvero Latent Semantic Analysis (LSA) e Latent Dirichlet Allocation (LDA).

7.1. PREPROCESSING

Prima di poter eseguire il task di topic modeling è necessario preparare il corpus, in modo da poter applicare gli algoritmi necessari. Dopo l'applicazione del preprocessing descritto nel paragrafo 3, viene creato il dizionario di riferimento della collezione di documenti. Successivamente, per eseguire la feature selection, al dizionario viene applicato un filtro che consente di ammettere solo i termini che compaiono in almeno 5 documenti e non in più del 50% di quelli totali della repository di riferimento. Infine, Il dizionario viene utilizzato per convertire la lista di documenti (corpus) in una Document-Term matrix. In particolare sono state considerate due differenti tipi di rappresentazione dei documenti, ossia la rappresentazione Bag of Words con frequenze (BOW) e la rappresentazione Term Frequency–Inverse Document Frequency (TF-IDF).

Per verificare se il preprocessing sia avvenuto correttamente è stata realizzata una wordcloud per ottenere una rappresentazione visiva delle parole più comuni. È fondamentale, per comprendere i dati, garantire che si stia procedendo correttamente e se sia necessario un ulteriore preprocessing prima di addestrare i modelli di topic modeling.

Figura 6 - Wordcloud precedente all'eliminazione di parole non significative nel corpus

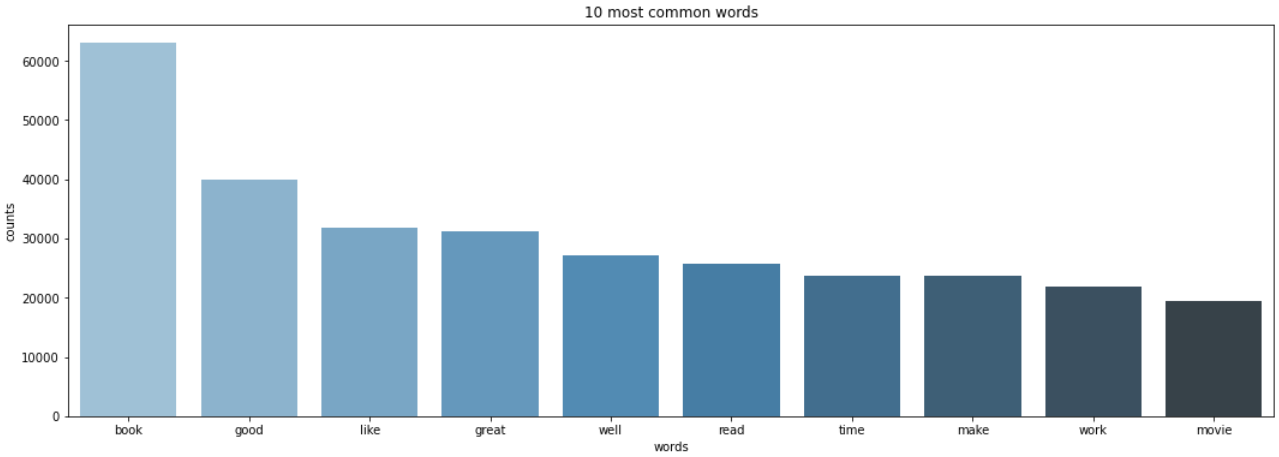

Figura 7 - Istogramma delle 10 parole più frequenti nel corpus precedente all'eliminazione di parole non significative

Come si può notare in Figura 6 e in Figura 7, nel corpus sono presenti parole poco utili per identificare le principali categorie discusse nelle recensioni (esempio: "well", "good", "even" o "love") **come è stato verificato dopo una prima applicazione dell'algoritmo sul corpus iniziale**. Queste parole risultano troppo generiche e quindi inutili per discriminare tra le diverse categorie di prodotti, di conseguenza sono state eliminate.

Figura 8 - Wordcloud dopo l'eliminazione di parole non significative nel corpus

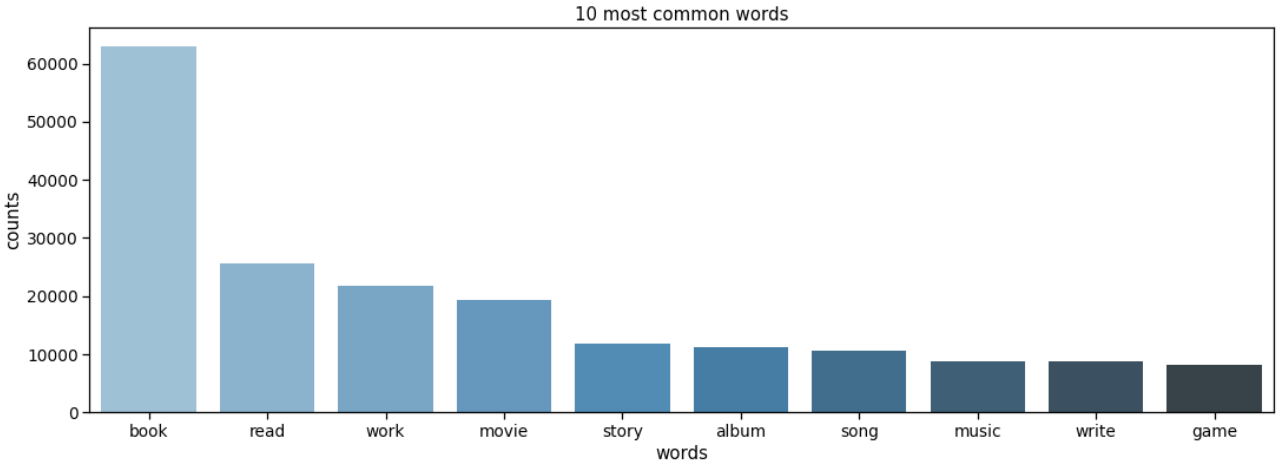

Figura 9 - Istogramma delle 10 parole più frequenti nel corpus dopo l'eliminazione di parole non significative

Dopo l'eliminazione dei termini non significativi, le parole più popolari sono tutte utili per discriminare tra le diverse categorie discusse nelle recensioni, come è possibile notare in Figura 8 in Figura 9.

7.2. LSA

Innanzitutto, per la realizzazione di un modello di topic modeling, è necessario identificare il numero opportuno di topic. In riferimento all'approccio LSA, il numero ottimale di topic può essere determinato attraverso la misura di coerenza, ovvero una misura della similarità semantica tra le principali parole del topic considerato. Di conseguenza è stato valutato il numero ottimale di topic per l'approccio LSA, sia in riferimento alla rappresentazione del corpus BOW sia in riferimento alla rappresentazione TF-IDF, considerando unigrammi di parole.

Per quanto riguarda la rappresentazione BOW, il numero ottimale di topic è 3. In particolare, con 3 topic si ottiene un valore di coerenza pari a 0,443. Tuttavia, è possibile notare in Tabella 1 che il risultato ottenuto tramite l'applicazione di un modello LSA con rappresentazione BOW non è particolarmente significativo; poiché il topic 2 potrebbe riferirsi sia alla categoria "Movies & TV Series" che alla categoria "Books" e anche il topic 3 potrebbe riferirsi sia alla categoria "Movies & TV Series" che alla categoria "Music".

Topic 1	Topic 2	Topic 3
Book: 0.900	Movie: 0.749	Movie: -0.533
Read: 0.301	Book: -0.201	Album: 0.489
Story: 0.092	Album: 0.194	Song: 0.366
Write: 0.082	Work: 0.181	Work: 0.308
Author: 0.068	Song: 0.176	Music: 0.193
Character: 0.065	Watch: 0.167	Sound: 0.163
Work: 0.60	Film: 0.150	Listen: 0.117
Movie: 0.052	Music: 0.119	Hear: 0.106
Life: 0.051	Story: 0.114	Game: 0.101
People: 0.049	Play: 0.100	Band: 0.097

Tabella 1 - Risultato del modello LSA con rappresentazione BoW del corpus (i valori esprimono il peso di appartenenza di ciascuna parola al topic)

Per quanto riguarda la rappresentazione TF-IDF, il numero ottimale di topic è 5. In particolare, con 5 topic si ottiene un valore di coerenza pari a 0.466. Anche in questo caso risulta difficile assegnare un'etichetta a ciascun topic, come mostrato in Tabella 2. Ad esempio, il topic 1 può riferirsi sia alla categoria "Movies & TV Series" sia alla categoria "Books".

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Book: 0.502	Movie: 0.669	Game: 0.856	Game: 0.856	Work: 0.497
Read: 0.284	Book: -0.496	Album: 0.424	Album: -0.288	Game: -0.303
Movie: 0.264	Read: -0.254	Song: 0.343	Song: -0.266	Album: -0.290
Story: 0.147	Watch: 0.194	Game: 0.239	Play: 0.189	Battery: 0.244
Work: 0.118	Album: 0.151	Book: -0.192	Music: -0.106	Song: -0.227
Write: 0.104	Song: 0.146	Music: 0.188	Graphic: 0.088	Book: -0.183
Album: 0.102	Film: 0.116	Sound: 0.147	Listen: -0.086	Movie: -0.126
Character: 0.102	Game: 0.103	Listen: 0.140	Band: -0.067	Phone: 0.121
Song: 0.101	Music: 0.101	Work: 0.133	Hear: -0.064	Charge: 0.108
Watch: 0.100	Play: 0.078	Play: 0.115	Work: 0.060	Read: -0.105

Tabella 2 - Risultato del modello LSA con rappresentazione TF-IDF del corpus (i valori esprimono il peso di appartenenza di ciascuna parola al topic)

Alcuni coefficienti sono negativi, ciò è dovuto al fatto che LSA si basa sulla decomposizione SVD (non si basa su un approccio probabilistico) e i coefficienti nella decomposizione SVD non sono facilmente interpretabili.

Coefficienti negativi o positivi non significano necessariamente un legame negativo o positivo con il topic corrispondente.

Si osserva che LSA presenta diversi svantaggi:

- Alcuni risultati possono non essere interpretabili dal punto di vista semantico.
- In alcuni casi non è possibile catturare il diverso significato delle parole in una collezione di documenti.
- Limitazioni del modello BOW (raccolta non ordinata delle parole).

A causa di questi svantaggi è stato proposto il modello alternativo LDA.

7.3. LDA

Anche per l'approccio LDA è necessario identificare il numero ottimale di topic, ma in questo caso le misure di valutazione considerate sono sia la coerenza che la perplexity, anche se il valore di coerenza ha un peso maggiore nelle valutazioni. Di conseguenza è stato valutato il numero ottimale di topic per l'approccio LDA, sia in riferimento alla rappresentazione del corpus BOW sia in riferimento alla rappresentazione TF-IDF, considerando inizialmente soltanto unigrammi e poi con anche i bigrammi.

7.3.1. UNIGRAMMI

Innanzitutto, vengono considerati gli unigrammi di parole, come è stato fatto anche per l'approccio LSA.

Per quanto riguarda la rappresentazione BOW, il numero ottimale di topic è 6. In particolare, con 6 topic si ottiene un valore di coerenza pari a 0,574 e un valore di perplexity pari a -7,995. Il valore di coerenza è superiore rispetto ai valori di coerenza ottenuti utilizzando l'approccio LSA.

Si ottiene un buon risultato di topic modeling (Tabella 3) in quanto è possibile identificare le etichette alle quali i topic si riferiscono.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
Movie: 0.051	Work: 0.046	Album: 0.030	Water: 0.008	Book: 0.051	Book: 0.082
Watch: 0.020	Battery: 0.008	Song: 0.029	Clean: 0.007	Help: 0.010	Read: 0.051
Film: 0.018	Picture: 0.007	Music: 0.024	Large: 0.006	Learn: 0.008	Story: 0.022
Show: 0.016	Camera: 0.006	Game: 0.022	Plastic: 0.006	Information: 0.008	Life: 0.014
Star: 0.013	Phone: 0.005	Play: 0.019	Baby: 0.006	Cover: 0.006	Write: 0.014
Video: 0.009	Support: 0.005	Sound: 0.017	Hand: 0.006	Understand: 0.006	Character: 0.013
Funny: 0.007	Call: 0.005	Hear: 0.015	Wear: 0.006	Example: 0.005	Author: 0.008
People: 0.007	Service: 0.005	Listen: 0.013	Side: 0.005	Author: 0.005	Enjoy: 0.008
Scene: 0.006	Case: 0.005	Band: 0.010	Fall: 0.005	Idea: 0.004	Child: 0.007
Season: 0.006	Company: 0.005	Track: 0.009	Daughter: 0.004	History: 0.004	Interesting: 0.007

Tabella 3 - Risultato del modello LDA con rappresentazione BoW del corpus, unigrammi (i valori esprimono il peso di appartenenza di ciascuna parola al topic)

Nello specifico le etichette identificate sono le seguenti:

Topic 1	Movies & TV Series
Topic 2	Electronics
Topic 3	Music
Topic 4	Family Needs
Topic 5	Textbooks
Topic 6	Novels

Tabella 4 - Etichette associate ai topic identificati con l'approccio LDA e rappresentazione BoW del corpus (unigrammi)

Per quanto riguarda la rappresentazione TF-IDF, il numero ottimale di topic è 5.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Album: 0.029	Battery: 0.017	Movie: 0.009	Plastic: 0.011	Book: 0.020
Song: 0.027	Camera: 0.013	Work: 0.007	Water: 0.009	Read: 0.005
Band: 0.012	Phone: 0.011	Read: 0.005	Hair: 0.008	Life: 0.005
Rock: 0.010	Service: 0.011	Game: 0.005	Coffee: 0.008	Author: 0.005
Record: 0.009	Replace: 0.010	Story: 0.004	Clean: 0.006	Write: 0.004
Track: 0.008	Charge: 0.009	Watch: 0.004	Filter: 0.006	Help: 0.004
Lyric: 0.005	Card: 0.009	Play: 0.004	Bottle: 0.006	Learn: 0.004
Sing: 0.005	Seller: 0.009	Book: 0.004	Blade: 0.006	Information: 0.004
Guitar: 0.005	Shipping: 0.008	Sound: 0.004	Headphone: 0.005	World: 0.003
Vocal: 0.005	Computer: 0.008	Show: 0.004	Knife: 0.005	Reader: 0.003

Tabella 5 - Risultato del modello LDA con rappresentazione TF-IDF del corpus, unigrammi (i valori esprimono il peso di appartenenza di ciascuna parola al topic)

In particolare, con 5 topic si ottiene un valore di coerenza pari a 0,510 e un valore di perplexity pari a -8,916. Sia il valore di coerenza che il valore di perplexity sono più bassi con il corpus TF-IDF che con il corpus BOW. Un minor valore di coerenza è un aspetto negativo e un minor valore di perplexity

è un aspetto positivo, ma poiché il valore di coerenza nelle valutazioni ha un peso maggiore, il risultato ottenuto utilizzando il corpus BOW (Tabella 3) risulta migliore rispetto a quello ottenuto utilizzando il corpus TF-IDF (Tabella 5).

Infatti, è possibile identificare i seguenti topic:

Topic 1	Music
Topic 2	Electronics
Topic 3	Movies & TV Series
Topic 4	Family Needs
Topic 5	Books

Tabella 6 - Etichette associate ai topic identificati con l'approccio LDA e rappresentazione TF-IDF del corpus (unigrammi)

Tuttavia, il risultato ottenuto con il corpus BOW risulta migliore perché consente, ad esempio, di distinguere tra "Textbooks" e "Novels" e, inoltre, il topic "Movies & TV Series" viene identificato meglio (come mostrato in Tabella 3).

Dato che il valore di coerenza risultava simile anche con 9 e 11 topic si è scelto di valutare anche questi modelli, ma senza ottenere risultati soddisfacenti. Infatti, sarebbe stato interessante capire se, all'aumentare del numero dei topic, sarebbe aumentata anche la precisione del topic definito, e.g. i vari tipi di film e non solo la categoria generale. In entrambi i casi, i topic "Movies & TV Series" e "Books" sono raggruppati insieme, ciò però non è desiderabile, in quanto sia il termine "Movie" che il termine "Book" sono tra le parole più frequenti all'interno del corpus (Figura 9).

7.3.2. UNIGRAMMI E BIGRAMMI

In riferimento all'approccio LDA non si considerano soltanto unigrammi, ma anche bigrammi di parole. Si è scelto di valutarlo soltanto per l'approccio LDA perché è la tecnica migliore in letteratura per performare topic modeling e perché è stato possibile ottenere risultati empirici migliori. In particolare, alla rappresentazione del testo mediante unigrammi, sono stati aggiunti bigrammi.

Per quanto riguarda la rappresentazione BOW, il numero ottimale di topic è 5. In particolare, con 5 topic si ottiene un valore di coerenza pari a 0,533 e un valore di perplexity pari a -8,009. Sia il valore di coerenza che il valore di

perplexity sono più bassi in riferimento al corpus BOW con unigrammi e bigrammi, che in riferimento al corpus BOW con solo unigrammi. Di conseguenza il risultato ottenuto utilizzando il corpus BOW con unigrammi (Tabella 3) risulta migliore rispetto a quello ottenuto utilizzando il corpus BOW con unigrammi e bigrammi (Tabella 7).

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Work: 0.028	Water: 0.010	Album: 0.036	Book: 0.068	Movie: 0.058
Battery: 0.005	Taste: 0.010	Song: 0.035	Read: 0.027	Game: 0.024
Picture: 0.004	Food: 0.009	Music: 0.029	Story: 0.010	Watch: 0.023
Camera: 0.004	Recipe: 0.007	Sound: 0.022	Write: 0.009	Film: 0.020
Design: 0.003	Coffee: 0.007	Hear: 0.016	Life: 0.007	Play: 0.016
Case: 0.003	Workout: 0.006	Listen: 0.015	Author: 0.006	Video: 0.012
Open: 0.003	Cook: 0.006	Band: 0.012	Character: 0.006	Show: 0.011
Replace: 0.003	Pump: 0.005	Track: 0.010	People: 0.005	Star: 0.008
Store: 0.003	Flavor: 0.004	Rock: 0.008	Page: 0.005	Scene: 0.007
Plastic: 0.003	Drink: 0.004	Record: 0.008	Child: 0.004	Funny: 0.007

Tabella 7 - Risultato del modello LDA con rappresentazione BoW del corpus (unigrammi + bigrammi)

Tuttavia, risultano evidenti le etichette che è possibile associare a ciascun topic:

Topic 1	Electronics
Topic 2	Kitchen tools
Topic 3	Music
Topic 4	Books
Topic 5	Movies

Tabella 8 - Etichette associate ai topic identificati con l'approccio LDA e rappresentazione BoW del corpus (unigrammi + bigrammi)

Osservato il valore di coerenza, si è considerato anche un numero di topic pari a 6 e un numero di topic pari a 7, ma senza ottenere risultati particolarmente significativi.

Per quanto riguarda la rappresentazione TF-IDF, il numero ottimale di topic è 9. In particolare, con 9 topic si ottiene un valore di coerenza pari a 0,567 e un valore di perplexity pari a -8,41. I valori di coerenza e perplexity sono soddisfacenti ma non si ottengono topic significativi per quanto riguarda la valutazione manuale.

7.4. CONCLUSIONE

In conclusione, i risultati più soddisfacenti sono stati ottenuti con il modello LDA e rappresentazione BoW del corpus (unigrammi).

FONTI

1. Amazon Reviews for Sentiment Analysis, a few million Amazon reviews in fastText format: https://www.kaggle.com/bittlingmayer/amazonreviews
2. fastText for Text Classification, https://towardsdatascience.com/fasttext-for-text-classification-a4b38cbff27c
3. Releasing fastText, https://fasttext.cc/blog/2016/08/18/blog-post.html

APPENDICE

Classificazione pura: i tre migliori

1. SVC CON KERNEL LINEARE, TF-IDF, 1-GRAM/2-GRAM		
Misura	Media	Standard Error
Accuratezza	0.894	1e-05
F1	0.894	1e-05
Tempo di training (s)	2.68	3e-04
Tempo di Test (s)	0.03	3e-06

2. SGD LINEARE, BOW FREQUENCIES, 1-GRAM/2-GRAM		
Misura	Media	Standard Error
Accuratezza	0.89	6e-06
F1	0.89	5e-06
Tempo di training (s)	1.29	5e-04
Tempo di Test (s)	0.03	3e-06

3. LOGISTIC REGRESSION, BOW FREQUENCIES, 1-GRAM/2-GRAM		
Misura	Media	Standard Error
Accuratezza	0.894	1e-05
F1	0.894	1e-05
Tempo di training (s)	16.7	9e-03
Tempo di Test (s)	0.03	3e-06

Si noti che I risultati del tempo di training e testing si riferiscono alla totale durata per 5 folds.

Sentiment supervised analysis

1. OPINION LEXICON	
Misura	Valore
Accuratezza	0.79
F1	0.8
Tempo di Test (s)	1.79

2. VADER	
Misura	Valore
Accuratezza	0.73
F1	0.71
Tempo di Test (s)	3 min 33 s

3. AFFIN	
Misura	Valore
Accuratezza	0.74
F1	0.72
Tempo di Test (s)	11 min 22 s