VIDEOGAME’S REVIEWS CLUSTERING

Ricerca di tendenze e comportamenti di prodotti della categoria Video Games su Amazon attraverso tecniche di Text Clustering

Tornatora Raffaele

Mat. 786357

1 Febbraio 2019

Sommario

INTRODUZIONE 1

IL PROBLEMA 2

IL DATASET 3

IL PROCESSO 4

ESTRAZIONE 5

OPERAZIONI PRELIMINARI 6

PREPROCESSING 7

TOKENIZATION 7

STEMMING 8

RIMOZIONE STOP-WORDS 8

RAPPRESENTAZIONE 8

CLUSTERING 9

MODELLI 9

FITTING SUL CORPUS E VALUTAZIONE DELLE PERFORMANCE 11

VISUALIZZAZIONE E ANALISI DEI CLUSTERS 11

RISULTATI 12

PERFORMANCE DEI MODELLI 12

VISUALIZZAZIONE DEI CLUSTERS 13

ANALISI CLUSTERS 14

NUMERO DI PAROLE CHIAVE CONTENUTE NEI CLUSTERS 15

MEDIA DELLE VALUTAZIONI 17

NUMERO DI RECENSIONI PER PRODOTTO 18

CONCLUSIONI E SVILUPPI FUTURI 21

BIBBLIOGRAFIA 22

# INTRODUZIONE

Un’analisi di Text Clustering è un processo di Clustering applicato a testi scritti in linguaggio naturale.

Un processo di clustering permette di raggruppare un insieme di oggetti in classi di oggetti considerati simili. E’ la più comune forma di apprendimento non supervisionato.

In Information Retrival il Text Clustering ha diverse applicazioni:

* Clustering per i risultati di ricerca (es. da un motore di ricerca);
* Scatter-gather: metodi alternativi (al testo) di ricerca;
* Clustering di collezione;
* Modellazione di linguaggio;
* Recupero di informazioni basate su cluster.

Lo studio presentato utilizza tecniche di Text Clustering per ricercare tendenze e comportamenti comuni di alcune recensioni di prodotti nella categoria Video Games del celebre sito di e-commerce Amazon [1] venduti fino al luglio del 2014.

Il testo delle recensioni è in lingua inglese.

L’analisi è eseguita attraverso notebook Jupyter [2] con linguaggio di programmazione Python [3], mentre alcuni grafici di analisi sono eseguiti attraverso Tableau Public [4].

L’ambiente di programmazione è composto da un notebook Apple MacBook Pro 15” Retina del 2015 con distribuzione Anaconda composta da Python, R, Jupyter, JupyterLab, QTConsole, RStudio, Spyder e Visual Studio Code. Nella macchina è inoltre installato Tableau Public 2018.3.

Nei prossimi due sottocapitoli vengono esposti il problema e il dataset; nel prossimo capitolo viene esposto tutto il processo d’analisi in dettaglio; nel capitolo successivo vengono esposti alcuni risultati principali tra cui le performance dei modelli, la rappresentazione dei modelli e i risultati di alcune analisi notevoli.

Nel capitolo finale vengono esposte le conclusioni e alcune proposte di lavori futuri.

## IL PROBLEMA

L’analisi di Text Clustering presentata è di tipo hard, in cui ogni documento appartiene ad uno di determinati clusters. Il numero di cluster può essere determinato dall’utente, oppure determinato automaticamente dal modello (algoritmo di clustering). Nel nostro caso è determinato da me e ho scelto di suddividere il testo prima in 5 clusters poi in 3; questo perché molte delle informazioni possedute, come il nome del prodotto, valutazione da 0 a 5 data dall’utente, etc. possono essere logicamente divise in 5 o 3 clusters (es. in base ai nomi dei prodotti si possono distinguere i videogiochi in un cluster, le console in un altro e i servizi online legati al gaming in un altro ancora. Oppure in base alle valutazioni si possono raggruppare le valutazioni negative, con valori compresi tra 0 e 2, in un cluster, le valutazioni positive, con valori tra 3 e 5, in un altro, e valutazioni indifferenti, con valori tra 2 e 3, in un altro ancora).

Il testo è rappresentato in uni-grammi pesati attraverso la tecnica Tf-Idf (term frequency-inverse document frequency), la miglior tecnica di rappresentazione del linguaggio naturale utilizzata in Information Retrieval.

I modelli scelti sono due, entrambi di tipo piatto: il primo è il

K-Means, l’algoritmo di clustering più comune e utilizzato, mentre il secondo è il Gaussian EM. Le performance dei modelli sono misurate attraverso il coefficiente di Calinski-Harabaz [5].

Infine i clusters estratti dai modelli con suddivisione del testo in 3 clusters e 5 clusters con le più alte performance vengono visualizzati e analizzati per la ricerca di tendenze e comportamenti comuni.

## IL DATASET

Il dataset è fornito da [6] ed è stato utilizzato nei seguenti studi [7][8].

E’ un dataset contenente recensioni e metadati di alcuni prodotti Amazon nella categoria Video Games venduti fino a Luglio 2014. E’ un estratto dal dataset che contiene le recensioni, i metadati e i link di circa 143 milioni di prodotti Amazon con data di produzione dal maggio 1966 a luglio 2014.

Il dataset si compone di due archivi compressi in formato json.gz: uno che contiene le recensioni e uno che contiene i metadati dei prodotti. Assieme agli archivi vengono forniti dei codici sviluppati nei più comuni linguaggi di programmazione che permettono di importare facilmente i dati in formato tabellare. Nello studio presentato, siccome ho utilizzato Python come linguaggio di programmazione, ho utilizzato quello per Python che ha richiesto le librerie gzip e pandas.

Il dataset contenente le recensioni si compone di 231780 osservazioni e 9 attributi:

* reviewerID: l’id dell’utente;
* asin: l’id univoco dato da Amazon identificativo del prodotto;
* reviewerName: il nickname dell’utente;
* helpful: il grado di affidabilità della recensione dato da Amazon;
* reviewText: il testo della recensione;
* overall: la valutazione data dall’utente in un range da 0 a 5 con passo di 0.5;
* summary: un riassunto della recensione;
* unixReviewTime: la data di pubblicazione della recensione espresso in formato UNIX;
* reviewTime: la data di pubblicazione della recensione espresso il formato Giorno (attraverso due cifre) Mese (attraverso una cifra), Anno (attraverso quattro cifre).

Mentre il dataset contenente i metadati si compone di 50953 osservazioni e 9 attributi:

* asin: l’identificativo univoco dato da Amazon relativo al prodotto;
* description: una descrizione del prodotto;
* price: il prezzo del prodotto in dollari americani;
* imURL: il link alle immagini relative al prodotto;
* related: il codice asin di prodotti correlati, ovvero prodotti visti dagli stessi utenti che hanno visto il prodotto in oggetto;
* salesRank: il numero di prodotti venduti;
* categories: le categorie del prodotto;
* title: il nome del prodotto;
* brand: la marca del prodotto.

Il dataset contenente le recensioni non contiene valori nulli o mancanti; mentre quello contenente i metadati contiene molti valori mancanti, soprattutto negli attributi relativi al nome del prodotto e del brand.

# IL PROCESSO

Il processo è composto da varie fasi illustrate nella figura alla pagina successiva.

Dopo la figura, nei prossimi sotto capitoli vengono presentate tutte le fasi in modo dettagliato.

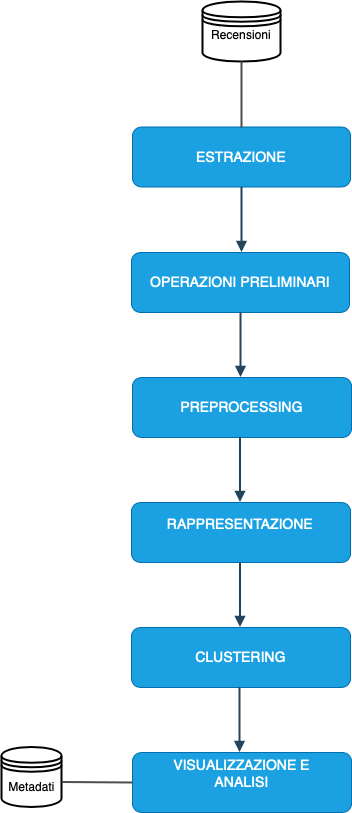


Figura 1: Schema delle fasi dell'analisi

## ESTRAZIONE

La prima fase è l’estrazione del dataset. Come descritto nel capitolo introduttivo di presentazione del dataset, è stato estratto attraverso un codice fornito assieme al dataset stesso. Il codice è presentato nella pagina successiva.

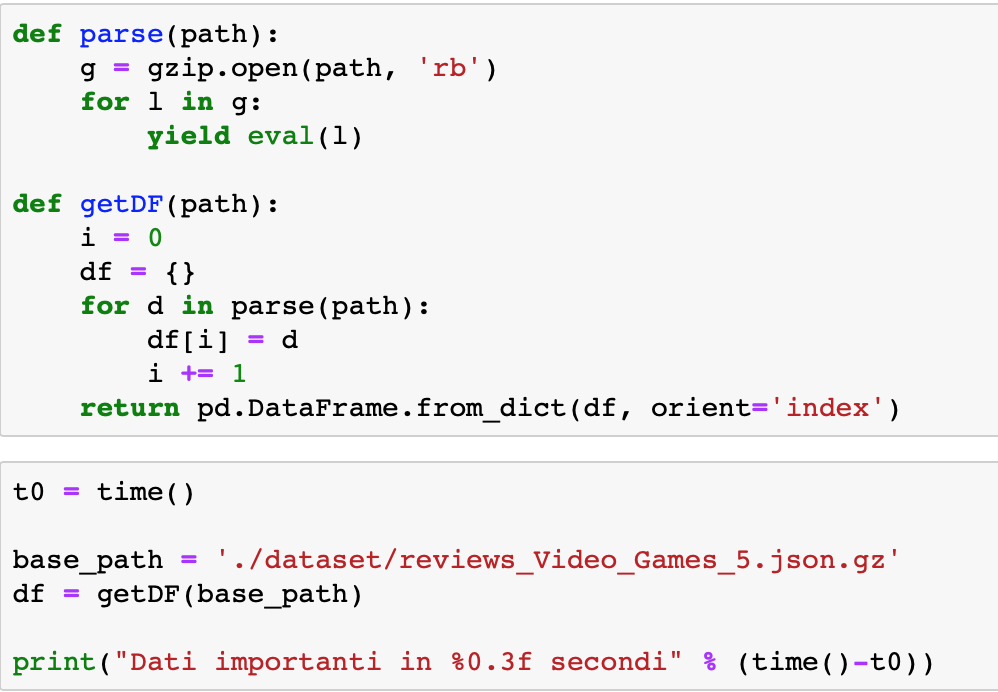


Figura 2: Codice fornito

## OPERAZIONI PRELIMINARI

La fase successiva all’estrazione del dataset sono le operazioni preliminari. Queste operazioni consentono di estrarre e preparare i dati per l’analisi di Text Clustering.

Per prima cosa dal dataset si estrae il corpus, ovvero l’insieme di documenti su cui verrà effettuata l’analisi, nel nostro caso l’insieme delle recensioni; queste vengono estratte estraendo le osservazioni dell’attributo reviewText del dataset.

Sul corpus estratto vengono poi applicate delle operazioni di normalizzazione del testo, più specificatamente:

* Espansione delle forme contratte: alcune forme contratte, come quelle verbali, vengono espanse (es. isn’t -> is not). Questo permette di ottenere più tokens e più facilmente comprensibili;
* Rimpiazzamento dei segni diacritici: lettere con accenti e simboli particolari sono portate alla lettera senza accento. Questo permette di avere tokens di più facile comprensione;
* Conversione in lower case: tutte le lettere maiuscole vengono portate alla forma minuscola;
* Rimozione dei segni di punteggiatura, numeri e caratteri speciali;
* Rimozione degli spazi extra.

## PREPROCESSING

Dopo le operazioni preliminari inizia il processo di analisi vero e proprio; la prima fase è il preprocessing del testo ed è composto da tre sotto fasi: tokenization, stemming e rimozione delle stop-words. Verranno illustrate meglio nei prossimi sotto capitoli.

Nello studio ho effettuato queste operazioni con la libreria nltk, una delle più conosciute librerie per il processing di testo in linguaggio naturale per Python.

### TOKENIZATION

La prima sotto fase è quella di tokenization. Questa operazione consente di estrarre dal testo parti che lo compongono; parti che vengono dette tokens e possono essere caratteri, parole, o più parole.

Quando il testo viene rappresentato attraverso singole parole che lo compongono si dice che il testo è rappresentato in uni-grammi; quando viene rappresentato a gruppi di due parole che lo compongono si dice che è rappresentato in bi-grammi, e così via…

Nello studio presentato il testo è rappresentato in uni-grammi; è stato scelto questo tipo di rappresentazione perché il testo di ogni recensione è relativamente corto.

### STEMMING

La seconda sotto fase è quella di Stemming. Attraverso questa operazione ogni token viene ridotto alla propria radice (es. automate, automatic, automation -> automat).

Nello studio questa operazione viene effettata attraverso uno Stemmer, un algoritmo di stemming definito, nel nostro caso lo Snowball [9].

### RIMOZIONE STOP-WORDS

L’ultima sotto fase è l’eliminazione delle stop-words. Le stop-words sono delle parole contenute all’interno di un particolare elenco, detto appunto elenco di stop-words, che contiene parole molto comuni ma di poco rappresentative per il testo come gli avverbi e gli articoli. C’è un elenco per ogni lingua, ma nel pacchetto nltk è disponibile sono quello della lingua inglese.

## RAPPRESENTAZIONE

Dopo il preprocessing viene la fase di rappresentazione del testo.

In questa fase viene calcolata la significatività di ogni token estratto da ogni documento del corpus e viene creata una matrice, il cui nome varia in base al metodo utilizzato per determinare questa significatività, espressa come peso del token nel documento. Infine da questa matrice vengono eliminati tutti i token che nei vari documenti hanno un peso minore di una certa soglia, detta valore di cut-off. Nello studio questo valore è stato scelto per ottenere il più alto numero di tokens possibili mantenendo la sparsità della matrice sotto l’80%. I tokens rimanenti costituiscono l’insieme di features che verranno utilizzate nell’analisi.

Nello studio il metodo scelto per determinare la significatività dei tokens è il metodo Tf-Idf: questo metodo è composto da due parti moltiplicate tra loro; la prima è Tf (term frequency) che si calcola come la frequenza del termine normalizzato in base alla frequenza del termine stesso nei documenti. La seconda è Idf (Inverse Document Frequency) e si ottiene effettuando il logaritmo in base 10 del rapporto tra il numero il numero di occorrenze del termine all’interno del documento e il numero di documenti che contengono il termine stesso.

Dopo vari tentativi è stato ottenuto un buon compromesso tra numero di tokens e sparsità della matrice con il valore di cut-off 0.12, attraverso il quale è stato possibile estrarre 69 features con una sparsità della matrice del 78.92%.

## CLUSTERING

La fase di Clustering è la fase principale dell’analisi. In questa fase si esegue il Text Clustering vero e proprio.

Come prima sotto fase si definiscono i modelli, poi si esegue l’operazione di fitting sulle features estratte nella fase precedente e infine si calcolano e valutano le performance ottenute dai modelli.

### MODELLI

Come già detto nella prima parte del documento, i due modelli presentati sono di tipo piatto. Vediamoli più in dettaglio.

#### K-Means

Come già detto, il modello K-Means è l’algoritmo di clustering più utilizzato. E’ anche utilizzato per problemi di classificazione.

L’algoritmo ha l’obiettivo di minimizzare la varianza totale intra-cluster. Ogni cluster viene identificato mediante un centroide o un punto medio. Segue una procedura interattiva: inizialmente crea K partizioni e assegna ad ogni partizione i punti d’ingresso o casualmente o usando alcune informazioni euristiche.

Quindi calcola il centroide di ogni gruppo. Costruisce quindi una nuova partizione associando ogni punto d’ingresso al cluster il cui centroide è più vicino ad esso.

Quindi vengono ricalcolati i centroidi per i nuovi cluster e così via, finchè l’algoritmo non converge.

Più informazioni sull’algoritmo si possono trovare qui [10].

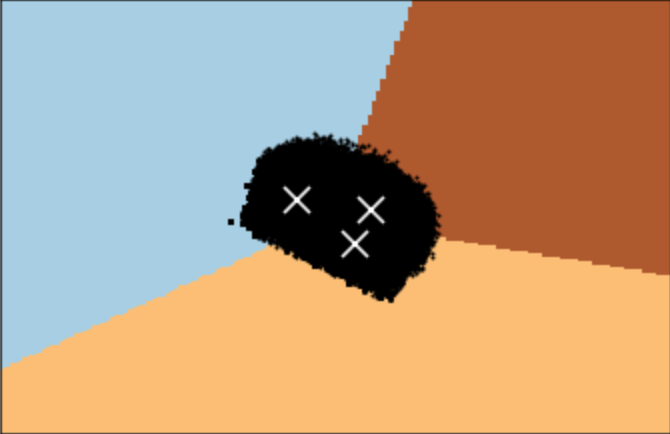


Figura 3: Rappresentazione K-Means con 3 clusters (i dati sono stati ridotti attraverso una SVD troncata con 2 componenti)

#### Gaussian EM

Questo modello di clustering si basa su un modello di distribuzione di probabilità Gaussian mixture.

Nello studio viene utilizzo con quattro setup di covarianza:

* Full: ogni componente ha la sua matrice di covarianza;
* Tied: tutti i componenti condividono la stessa matrice di covarianza;
* Diag: ogni componente ha la sua matrice di covarianza diagonale;
* Spherical: ogni componente ha la sua varianza.

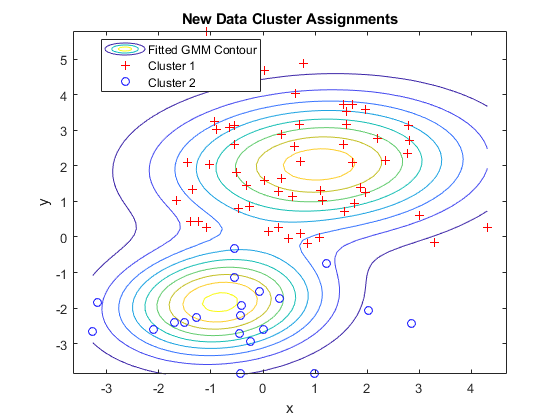


Figura 4: Rappresentazione modello Gaussian EM con 2 clusters

### FITTING SUL CORPUS E VALUTAZIONE DELLE PERFORMANCE

Dopo aver definito e caricato i modelli, viene eseguita l’operazione di fitting di ogni modello sulla matrice Tf-Idf. Questa operazione esegue il clustering attraverso l’algoritmo dividendo i documenti in un numero di classi dichiarato nella definizione del modello.

Dopo aver eseguito questa operazione viene calcolato l’indice di Calinski-Harabaz di ogni modello.

Infine vengono estratti i clusters creati dai modelli con suddivisione del testo in 3 e 5 clusters che hanno ottenuto il più alto indice di Calinski-Harabaz per l’ultima fase del processo.

## VISUALIZZAZIONE E ANALISI DEI CLUSTERS

La fase finale del processo è quella di visualizzazione e analisi dei clusters estratti nella fase precedente.

La visualizzazione dei clusters è stata effettuata, dopo aver estratto 15 osservazioni random dal dataset, attraverso un bubble chart in cui all’interno di ogni bubble è contenuto il nome del prodotto e, a capo, il nome del recensore. La bubble chart è stata creata con Tableau Public.

Dopo la visualizzazione sono state eseguite le analisi, di cui riportiamo quelle con i risultati più significativi nel prossimo capitolo. Le analisi performate sono state:

* Numero delle parole chiave (features) contenute nei vari clusters: attraverso una word cloud;
* Numero di recensioni per cluster;
* Media delle valutazioni nei vari clusters;
* Numero di recensioni per prodotto nei vari clusters;
* Numero di recensioni per recensore nei vari clusters;
* Numero di recensioni per anno di pubblicazione nei vari clusters.

# RISULTATI

In questo capitolo vengono illustrati i principali risultati ottenuti durante il processo.

## PERFORMANCE DEI MODELLI

Nella pagina successiva viene illustrata una tabella contenente il coefficiente di Calinski-Harabaz dei vari modelli.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | K Means | Gaussian EM  Full | Gaussian EM  Tied | Gaussian EM  Diag | Gaussian EM  Spherical |
| 3 clusters | 5718 | 2031 | 4600 | 1972 | 4937 |
| 5 clusters | 7056 | 3011 | 4131 | 2948 | 6213 |

Figura 5: Tabella degli indici di Calinski-Harabaz dei vari modelli

Come si vede dalla tabella, il modello con le performance più elevate, sia con suddivisione del testo in 3 clusters che in 5 clusters, è il K-Means. I clusters analizzati saranno quindi quelli creati da questi due modelli.

## VISUALIZZAZIONE DEI CLUSTERS

Sotto i due bubble chart rappresentativi dei clusters.

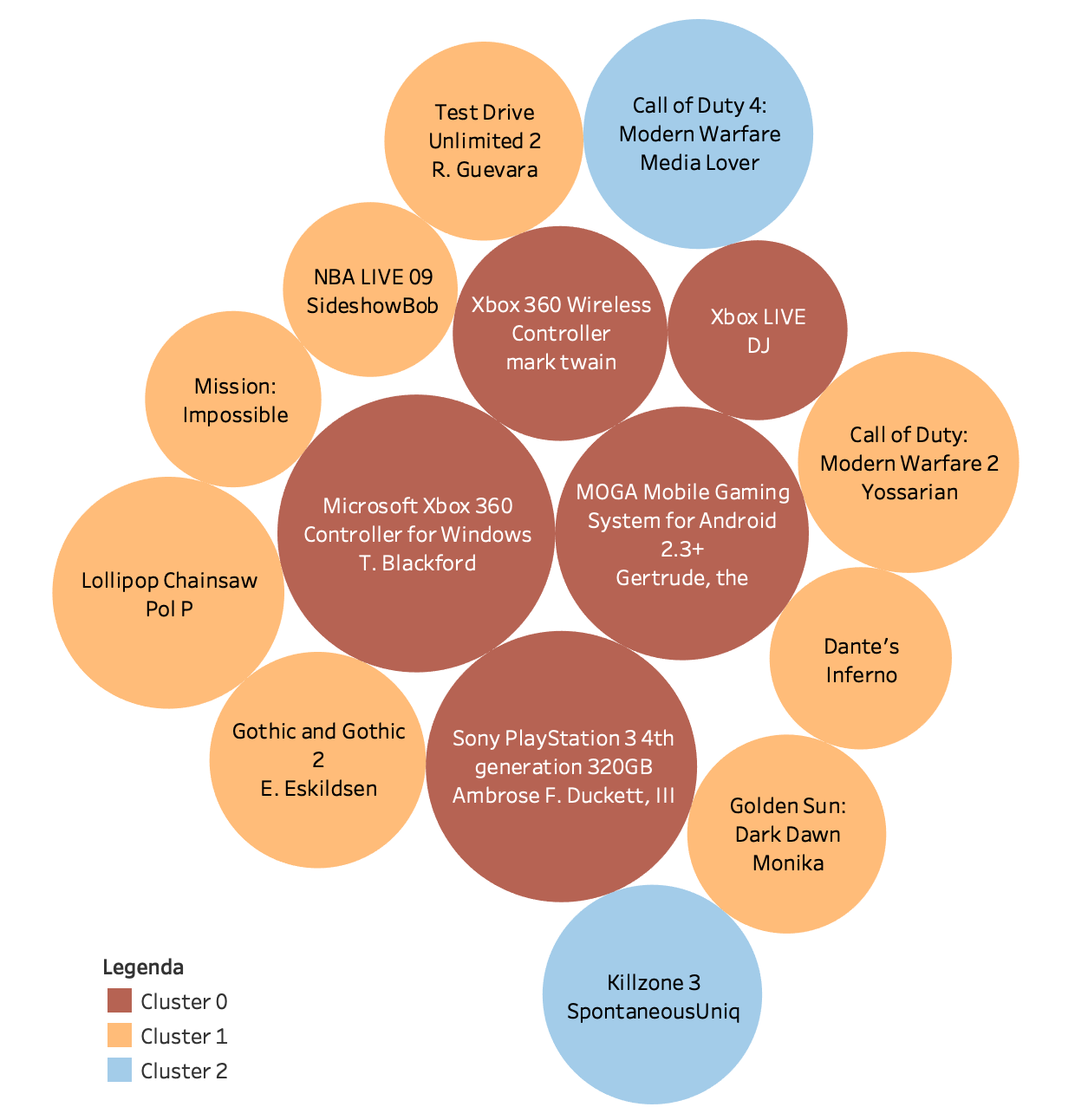


Figura 6: Rappresentazione dei clusters del modello K-Means con suddivisione in 3 clusters



Figura 7: Rappresentazione dei clusters del modello K-Means con suddivisione in 5 clusters

## ANALISI CLUSTERS

Sotto vengono illustrate alcune analisi effettuate, quelle con i risultati più significativi, tutte le altre analisi si possono trovare all’interno dello Script.

### NUMERO DI PAROLE CHIAVE CONTENUTE NEI CLUSTERS

#### Clusters modello K-Means con suddivisione in 3 clusters



Figura 8: Wordcloud rappresentante il numero di parole chiave contenute nel Cluster 0



Figura 9: Wordcloud rappresentante il numero di parole chiave contenute nel Cluster 2

Come si può vedere, il senso semantico delle key words più presenti nel Cluster 0 è di sicurezza; con termini come control e use.

Mentre il senso semantico delle key words più presenti nel Cluster 2 è l’amore; con termini come love e great.

#### Clusters modello K-Means con suddivisione in 5 clusters



Figura 10: Wordcloud rappresentante il numero di parole chiave contenute nel Cluster 0



Figura 11: Wordcloud rappresentante il numero di prole chiave contenute nel Cluster 3

In questo caso il significato semantico dei termini più utilizzati nei primi tre clusters, di cui riporto la rappresentazione del Cluster 0, è di amore con termini come like, love e great. Mentre quello dei termini più utilizzati negli ultimi due clusters, di cui riporto la rappresentazione del Cluster 3, è di sicurezza con termini come control e use.

### MEDIA DELLE VALUTAZIONI

#### Clusters modello K-Means con suddivisione in 3 clusters

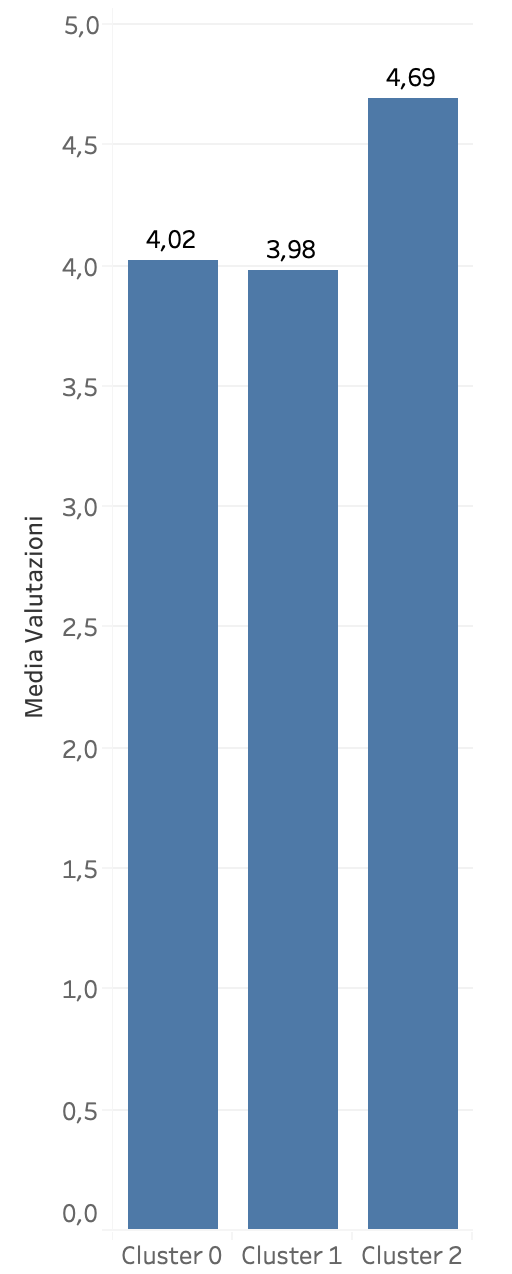


Figura 12: Bar chart rappresentante la media delle valutazioni contenute nei clusters

Come si vede nel bar chart, l’ultimo cluster ha la media di valutazioni più alta mentre i primi due hanno una media molto simile sulla valutazione di 4/5.

### NUMERO DI RECENSIONI PER PRODOTTO

#### Clusters modello K-Means con rappresentazione in 5 clusters

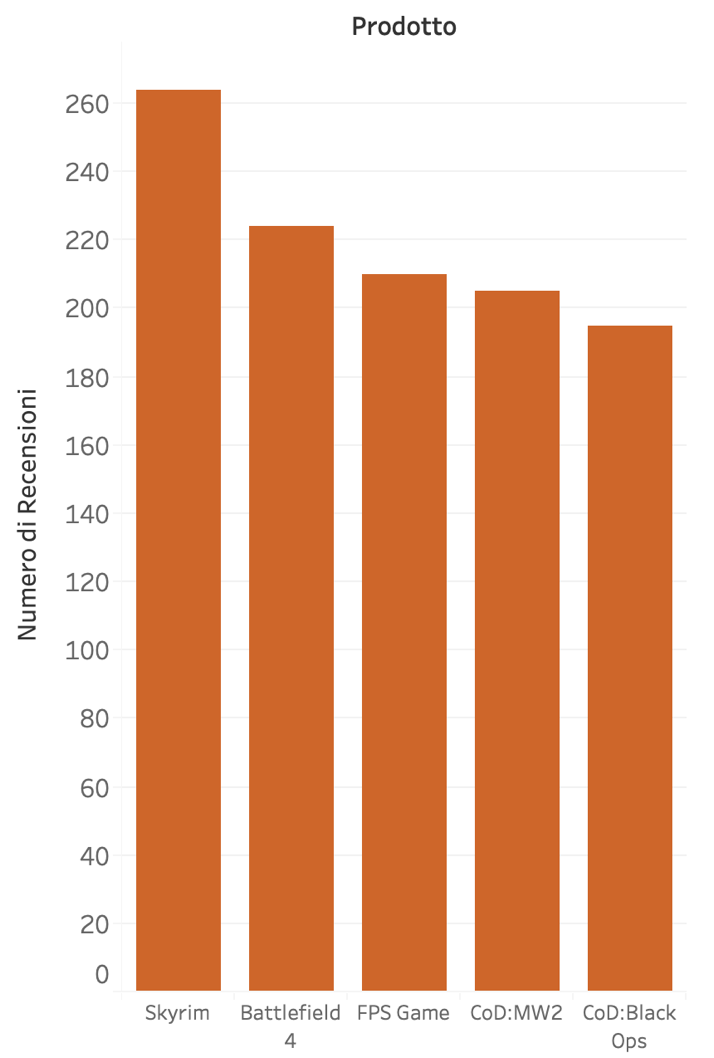


Figura 13: Bar chart rappresentante i primi cinque prodotti con il più alto numero di recensioni nel Cluster 1

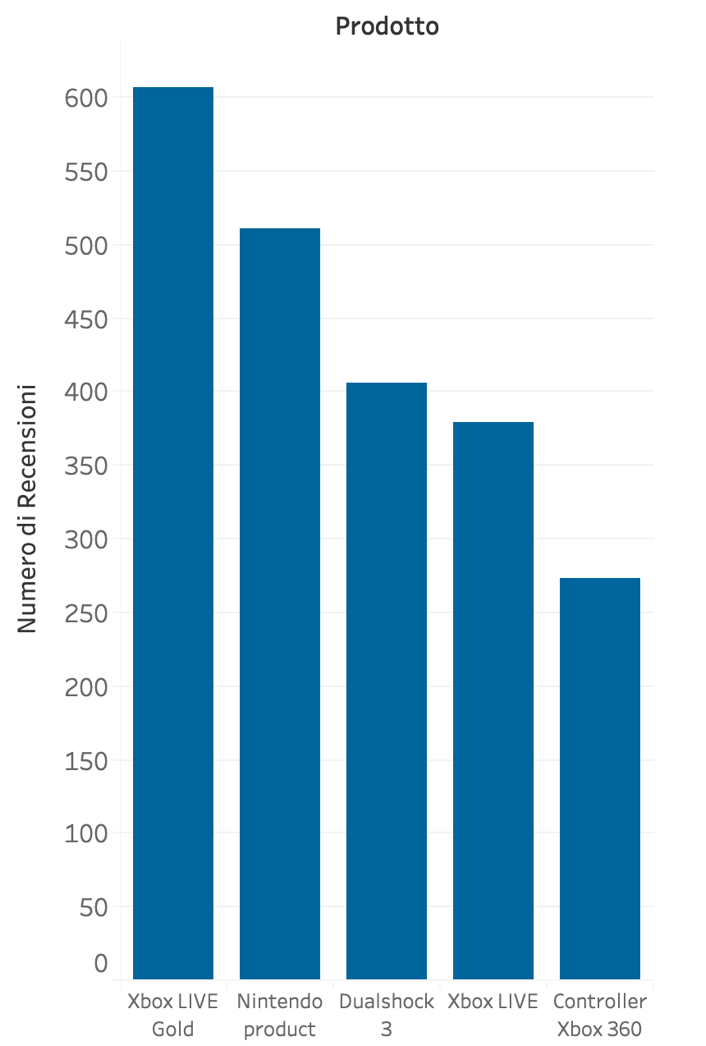


Figura 14: Bar chart rappresentante i primi cinque prodotti con il più alto numero di recensioni nel Cluster 3

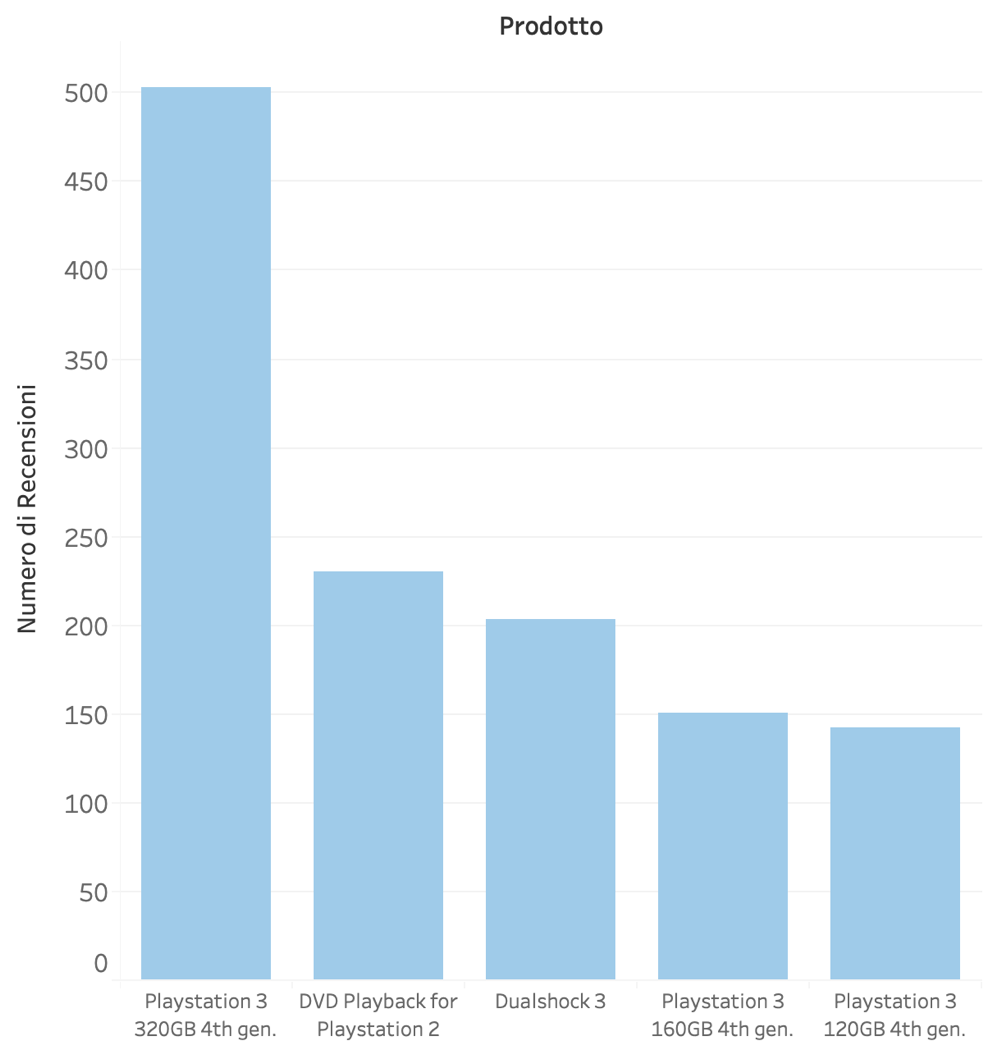


Figura 15: Bar chart rappresentante i primi cinque prodotti con il più alto numero di recensioni nel Cluster 4

Guardando per ogni cluster i cinque prodotti con il più alto numero di recensioni nel cluster si può vedere come le recensioni contenute nel Cluster 1 sono più riferite a Videogiochi, di tipo maggiormente FPS (Sparatutto in prima persona). Mentre le recensioni contenute nel Cluster 3 sono maggiormente riferite ad accessori per Console. Infine le recensioni contenute nel Cluster 4 sono maggiormente riferite a Console ed accessori PlayStation.

# CONCLUSIONI E SVILUPPI FUTURI

L’analisi ha creato dei clusters abbastanza differenti per alcune caratteristiche.

In particolare, i 3 clusters ricavati dal rispettivo modello K-Means sono lievemente caratterizzati dalla media delle valutazioni: costante nei primi due e più elevata nell’ultimo, da cui si suppone che le recensioni nei primi due cluster siano mediamente indifferenti, mentre quelle nell’ultimo siano positive.

Mentre alcuni dei 5 clusters ricavati dal rispettivo modello K-Means differiscono per il tipo di prodotti recensiti: il Cluster 1 contiene maggiormente Videogiochi, il Cluster 3 contiene maggiormente accessori per Console e il Cluster 4 contiene maggiormente prodotti PlayStation.

Infine si può affermare che il significato semantico dei termini più presenti sia in alcuni clusters di amore mentre in altri di sicurezza.

L’analisi può essere migliorata attraverso:

* L’utilizzo di un algoritmo gerarchico come il GAAC, di cui è presente nello script il codice dimostrativo, ma che non è stato possibile fittare nello studio a causa delle limitate prestazioni della macchina a disposizione;
* La definizione di modelli più specializzati, creati attraverso un tuning più fine;
* La rappresentazione del testo in maniera più complessa, ad esempio in bi-grammi.

Per concludere posso ritenermi abbastanza soddisfatto dell’analisi performata.

# BIBBLIOGRAFIA

[1] <https://www.amazon.com>

[2] <https://jupyter.org>

[3] <https://www.python.org>

[4] <https://public.tableau.com/s/>

[5] Caliński, T., & Harabasz, J. (1974). “A dendrite method for cluster analysis”. Communications in Statistics-theory and Methods 3: 1-27. doi:10.1080/03610926.2011.560741.

[6] <http://jmcauley.ucsd.edu/data/amazon/>

[7] Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering

R. He, J. McAuley

WWW, 2016

[8] Image-based recommendations on styles and substitutes

J. McAuley, C. Targett, J. Shi, A. van den Hengel

SIGIR, 2015

[9] Porter, M. “An algorithm for suffix stripping.” Program 14.3 (1980): 130-137.

[10] <https://it.wikipedia.org/wiki/K-means>