



Politecnico  
di Torino

Dipartimento di Scienze  
Matematiche "G. L. Lagrange"



## Progetto di analisi di dati su dataset di recensioni Amazon

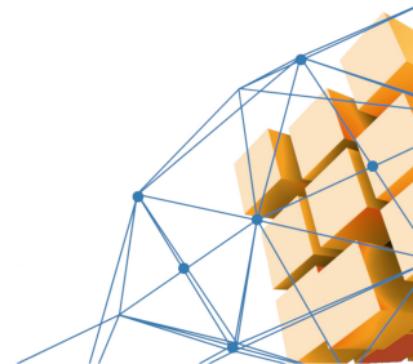
Candidati:

Simone Chiominto  
Giuseppe Intilla

Professori:

Prof. T. Cerquitelli  
Prof. P. Bethaz

Giugno 2022



# Piano della presentazione

- 1 Analisi del dataset
- 2 Preprocessing dei dati testuali
- 3 Presentazione dell'applicazione business
- 4 Algoritmi di clustering
- 5 Implementazione dell'applicazione business

## Presentazione del dataset

Il dataset da analizzare è un file .csv contenente 4000 reviews effettuate da utenti Amazon. Per ogni record abbiamo 10 attributi:

- reviewerID: codice identificativo dell'utente
- asin: codice identificativo del prodotto recensito
- reviewerName: nome dell'utente
- helpful: valutazione sull'utilità della recensione
- reviewText: testo della recensione
- overall: valutazione sul prodotto
- summary: riassunto della recensione
- unixReviewTime, reviewTime: data della recensione
- label: categoria a cui il prodotto appartiene

Le categorie dei prodotti sono: *sports & outdoors, film & TV, cellulari e accessori, salute e cura della persona.*

## Presentazione del dataset

Il dataset da analizzare è un file .csv contenente 4000 reviews effettuate da utenti Amazon. Per ogni record abbiamo 10 attributi:

- reviewerID: codice identificativo dell'utente
- asin: codice identificativo del prodotto recensito
- reviewerName: nome dell'utente
- helpful: valutazione sull'utilità della recensione
- reviewText: testo della recensione
- overall: valutazione sul prodotto
- summary: riassunto della recensione
- unixReviewTime, reviewTime: data della recensione
- label: categoria a cui il prodotto appartiene

Le categorie dei prodotti sono: *sports & outdoors, film & TV, cellulari e accessori, salute e cura della persona.*

## Analisi temporale

Le recensioni sono dateate tra l'8 Aprile del 1999 al 22 Luglio 2014 distribuite in maniera non uniforme negli anni, si può vedere infatti un notevole trend di crescita.

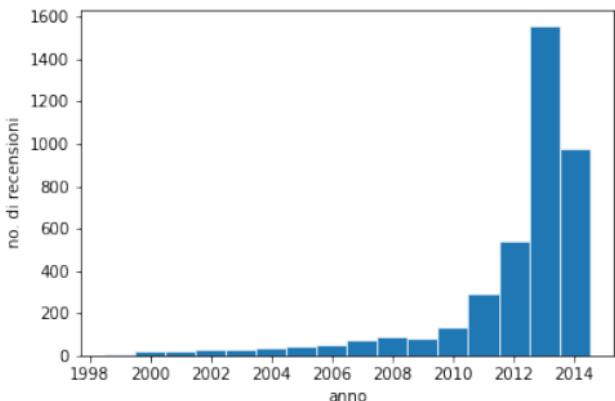


Figura: Istogramma del numero delle recensioni per ogni anno

Le recensioni sembrano però essere distribuite quasi uniformemente tra i mesi (con una diminuzione in autunno e un leggero aumento a dicembre e gennaio ) e tra i giorni della settimana.

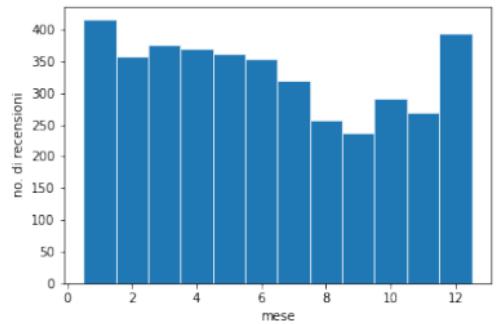


Figura: Istogramma del numero delle recensioni per ogni mese

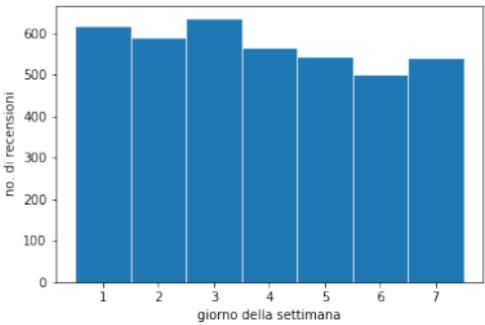


Figura: Istogramma del numero delle recensioni per ogni giorno della settimana

## Distribuzione dei prodotti e degli utenti

Notiamo poi che la maggior parte dei prodotti sono stati recensiti una sola volta e che quasi ogni utente ha recensito un solo prodotto.

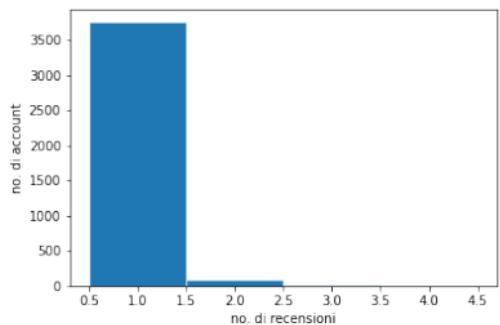


Figura: Istogramma del numero di account per numero di recensioni fatte

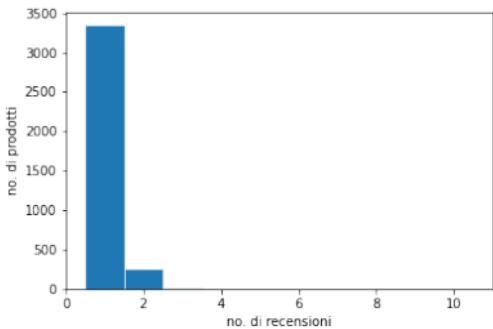


Figura: Istogramma del numero di prodotti per numero di recensioni

## Distribuzione dei voti e dell'utilità

Notiamo che gran parte delle recensioni danno una valutazione positiva dei prodotti e non sono state ritenute utili da alcun utente. Solo una piccola minoranza di recensioni è considerata utile a molti utenti.

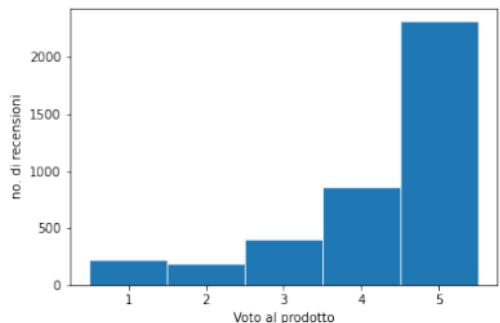


Figura: Istogramma del numero di recensioni per voto

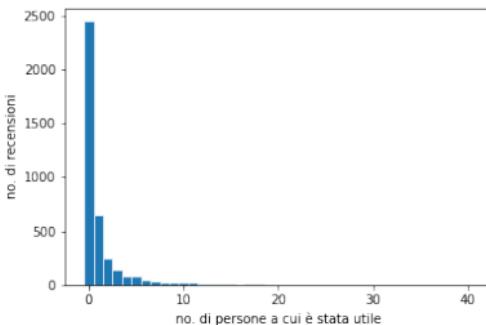


Figura: Distribuzioni del numero di persone a cui la recensione è stata utile

## Preprocessing dei dati testuali

Prima di fare un'analisi degli attributi **summary** e **TextReview** è necessario processare i dati in modo da essere esprimibili in forma quantitativa. I passi necessari sono stati:

- rendere minuscolo il testo
- rimuovere la punteggiatura
- rimuovere le stop words
- tokenizzare
- applicare stemming
- vettorializzare tramite TF-IDF

## Preprocessing dei dati testuali

Prima di fare un'analisi degli attributi **summary** e **TextReview** è necessario processare i dati in modo da essere esprimibili in forma quantitativa. I passi necessari sono stati:

- rendere minuscolo il testo
- rimuovere la punteggiatura
- rimuovere le stop words
- tokenizzare
- applicare stemming
- vettorializzare tramite TF-IDF

## Preprocessing dei dati testuali

Prima di fare un'analisi degli attributi **summary** e **TextReview** è necessario processare i dati in modo da essere esprimibili in forma quantitativa. I passi necessari sono stati:

- rendere minuscolo il testo
- rimuovere la punteggiatura
- rimuovere le stop words
- tokenizzare
- applicare stemming
- vettorializzare tramite TF-IDF

## Preprocessing dei dati testuali

Prima di fare un'analisi degli attributi **summary** e **TextReview** è necessario processare i dati in modo da essere esprimibili in forma quantitativa. I passi necessari sono stati:

- rendere minuscolo il testo
- rimuovere la punteggiatura
- rimuovere le stop words
- tokenizzare
- applicare stemming
- vettorializzare tramite TF-IDF

## Preprocessing dei dati testuali

Prima di fare un'analisi degli attributi **summary** e **TextReview** è necessario processare i dati in modo da essere esprimibili in forma quantitativa. I passi necessari sono stati:

- rendere minuscolo il testo
- rimuovere la punteggiatura
- rimuovere le stop words
- tokenizzare
- applicare stemming
- vettorializzare tramite TF-IDF

## Preprocessing dei dati testuali

Prima di fare un'analisi degli attributi **summary** e **TextReview** è necessario processare i dati in modo da essere esprimibili in forma quantitativa. I passi necessari sono stati:

- rendere minuscolo il testo
- rimuovere la punteggiatura
- rimuovere le stop words
- tokenizzare
- applicare stemming
- vettorializzare tramite TF-IDF

## TF-IDF

### Definizione

Sia  $D$  una collezione di documenti con cardinalità  $m$ . La tf-idf di un termine  $t$  in un documento  $d \in D$  è definita come

$$\text{tf-idf}(t) := \text{freq}(t, d) \cdot \log \left( \frac{m}{\text{freq}(t, D)} \right)$$

Questa metrica ci permette di misurare quanto un termine è comune in un documento ma poco nell'intera collezione e quindi di individuare le parole che caratterizzano meglio un preciso documento.



## Word cloud dell'intera collezione

Visualizziamo quindi le parole più importanti dell'intera collezione di documenti separatamente per gli attributi `summary` e `TextReview` tramite l'uso di word cloud. Notiamo che, come atteso, soprattutto nel caso dell'attributo `summary` le parole che rappresentano un giudizio sul prodotto sono le più importanti.



**Figura:** Word cloud per l'attributo textReview sull'intero dataset



**Figura:** Word cloud per l'attributo Summary sull'intero dataset



## Word cloud delle categorie

Visualizziamo ora le word cloud, separatamente per categoria, dell'attributo [textReview](#). Per rendere la visualizzazione più informativa abbiamo rimosso le parole che rappresentano un giudizio sul prodotto.



Figura: Word cloud per la categoria sports & outodoors



Figura: Word cloud per la categoria film & TV



## Word cloud delle categorie, continua...



Figura: Word cloud per la categoria cellulari e accessori



Figura: Word cloud per la categoria salute e cura della persona

## Principal component analysis

Adesso visualizziamo la porzione di varianza spiegata in funzione del numero di componenti principali separatamente per gli attributi `textReview` e `Summary`.

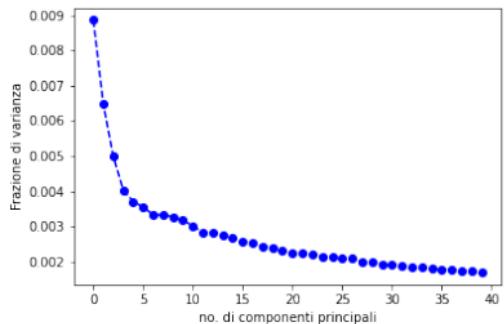


Figura: Frazione di varianza spiegata per componente principale per `textReview`

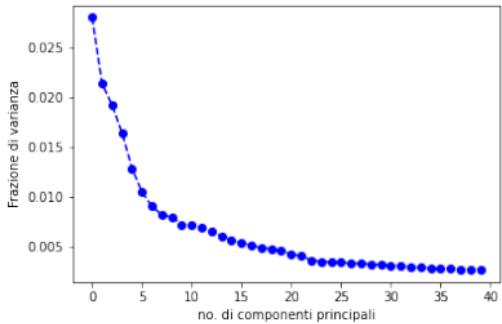


Figura: Frazione di varianza spiegata per componente principale per `Summary`

## Gestione dei valori mancanti

Nel dataset sono presenti in totale 41 valori mancanti, di cui 38 nell'attributo `reviewerName`, 2 nell'attributo `textReview` e 1 in `summary`. Per gestire questi dati mancanti:

- abbiamo chiamato i dati mancanti di `reviewerName` con il nome generico *Amazon customer*, già presente nel dataset per vari account.
- per gli attributi mancanti di `textReview` abbiamo copiato il testo del sommario nella recensione
- per quanto riguarda l'unico valore di `Summary` abbiamo letto la recensione corrispondente e fatto un riassunto.

## Gestione dei valori mancanti

Nel dataset sono presenti in totale 41 valori mancanti, di cui 38 nell'attributo `reviewerName`, 2 nell'attributo `textReview` e 1 in `summary`. Per gestire questi dati mancanti:

- abbiamo chiamato i dati mancanti di `reviewerName` con il nome generico *Amazon customer*, già presente nel dataset per vari account.
- per gli attributi mancanti di `textReview` abbiamo copiato il testo del sommario nella recensione
- per quanto riguarda l'unico valore di `Summary` abbiamo letto la recensione corrispondente e fatto un riassunto.

## Gestione dei valori mancanti

Nel dataset sono presenti in totale 41 valori mancanti, di cui 38 nell'attributo `reviewerName`, 2 nell'attributo `textReview` e 1 in `summary`. Per gestire questi dati mancanti:

- abbiamo chiamato i dati mancanti di `reviewerName` con il nome generico *Amazon customer*, già presente nel dataset per vari account.
- per gli attributi mancanti di `textReview` abbiamo copiato il testo del sommario nella recensione
- per quanto riguarda l'unico valore di `Summary` abbiamo letto la recensione corrispondente e fatto un riassunto.

## Applicazione business

Il nostro obiettivo è estrarre dal dataset delle informazioni utili per poter guidare il cliente nell'acquisto. La nostra idea è di permettere ad un utente di filtrare in maniera sequenziale il dataset per arrivare alla classe di prodotti specifica a cui è più interessato. I filtri sono delle parole chiave che descrivono in maniera efficace porzioni del dataset. L'ispirazione viene dalla possibilità nelle recensioni di Google di filtrare le recensioni di una singola attività per parole chiave.

Argomenti citati spesso dagli utenti



Figura: Esempio dell'idea di Google

## Applicazione business, continua ...

La nostra applicazioni però avrà le seguenti differenze:

- Verrà applicata a classi di recensioni e non alle recensioni di un singolo prodotto
- La scelta delle opzioni è esclusiva
- La parola chiave rappresenta un cluster di recensioni, ma non è detto che quella parola sia presente in tutte le recensioni del cluster

Per implementare questa idea abbiamo a disposizione:

clustering

algoritmi di clustering

Ci concentreremo sugli algoritmi di clustering perché le regole di associazione non riescono ad identificare porzioni di dataset quanto più delle caratteristiche generali del dataset.

## Applicazione business, continua ...

La nostra applicazioni però avrà le seguenti differenze:

- Verrà applicata a classi di recensioni e non alle recensioni di un singolo prodotto
- La scelta delle opzioni è esclusiva
- La parola chiave rappresenta un cluster di recensioni, ma non è detto che quella parola sia presente in tutte le recensioni del cluster

Per implementare questa idea abbiamo a disposizione:

• Algoritmi di clustering

• Algoritmi di associazione

Ci concentreremo sugli algoritmi di clustering perché le regole di associazione non riescono ad identificare porzioni di dataset quanto più delle caratteristiche generali del dataset.

## Applicazione business, continua ...

La nostra applicazioni però avrà le seguenti differenze:

- Verrà applicata a classi di recensioni e non alle recensioni di un singolo prodotto
- La scelta delle opzioni è esclusiva
- La parola chiave rappresenta un cluster di recensioni, ma non è detto che quella parola sia presente in tutte le recensioni del cluster

Per implementare questa idea abbiamo a disposizione:

Algoritmi di clustering

Algoritmi di associazione

Ci concentreremo sugli algoritmi di clustering perché le regole di associazione non riescono ad identificare porzioni di dataset quanto più delle caratteristiche generali del dataset.

## Applicazione business, continua ...

La nostra applicazioni però avrà le seguenti differenze:

- Verrà applicata a classi di recensioni e non alle recensioni di un singolo prodotto
- La scelta delle opzioni è esclusiva
- La parola chiave rappresenta un cluster di recensioni, ma non è detto che quella parola sia presente in tutte le recensioni del cluster

Per implementare questa idea abbiamo a disposizione:

Algoritmi di clustering

Regole di associazione

Ci concentreremo sugli algoritmi di clustering perchè le regole di associazione non riescono ad identificare porzioni di dataset quanto più delle caratteristiche generali del dataset.

## Applicazione business, continua ...

La nostra applicazioni però avrà le seguenti differenze:

- Verrà applicata a classi di recensioni e non alle recensioni di un singolo prodotto
- La scelta delle opzioni è esclusiva
- La parola chiave rappresenta un cluster di recensioni, ma non è detto che quella parola sia presente in tutte le recensioni del cluster

Per implementare questa idea abbiamo a disposizione:

- Algoritmi di clustering
- Regole di associazione

Ci concentreremo sugli algoritmi di clustering perché le regole di associazione non riescono ad identificare porzioni di dataset quanto più delle caratteristiche generali del dataset.

## Applicazione business, continua ...

La nostra applicazioni però avrà le seguenti differenze:

- Verrà applicata a classi di recensioni e non alle recensioni di un singolo prodotto
- La scelta delle opzioni è esclusiva
- La parola chiave rappresenta un cluster di recensioni, ma non è detto che quella parola sia presente in tutte le recensioni del cluster

Per implementare questa idea abbiamo a disposizione:

- Algoritmi di clustering
- Regole di associazione

Ci concentreremo sugli algoritmi di clustering perchè le regole di associazione non riescono ad identificare porzioni di dataset quanto più delle caratteristiche generali del dataset.

## Applicazione business, continua ...

La nostra applicazioni però avrà le seguenti differenze:

- Verrà applicata a classi di recensioni e non alle recensioni di un singolo prodotto
- La scelta delle opzioni è esclusiva
- La parola chiave rappresenta un cluster di recensioni, ma non è detto che quella parola sia presente in tutte le recensioni del cluster

Per implementare questa idea abbiamo a disposizione:

- Algoritmi di clustering
- Regole di associazione

Ci concentreremo sugli algoritmi di clustering perchè le regole di associazione non riescono ad identificare porzioni di dataset quanto più delle caratteristiche generali del dataset.

## Applicazione business, continua ...

La nostra applicazioni però avrà le seguenti differenze:

- Verrà applicata a classi di recensioni e non alle recensioni di un singolo prodotto
- La scelta delle opzioni è esclusiva
- La parola chiave rappresenta un cluster di recensioni, ma non è detto che quella parola sia presente in tutte le recensioni del cluster

Per implementare questa idea abbiamo a disposizione:

- Algoritmi di clustering
- Regole di associazione

Ci concentreremo sugli algoritmi di clustering perchè le regole di associazione non riescono ad identificare porzioni di dataset quanto più delle caratteristiche generali del dataset.

## K-means

Osserviamo ora il comportamento dei vari algoritmi di clustering sui dati dell'attributo [reviewText](#). Iniziamo con k-means. Per migliorarne la performance ripetiamo l'algoritmo 10 volte e scegliamo il risultato con SSE migliore. La distanza scelta è la *cosine similarity*.

Testando k-means con  $k = 4$  vorremmo vedere se riesce ad identificare le 4 categorie. I risultati sono i seguenti:

il cluster 0 (il più numeroso) copre quasi completamente le categorie *sports & outdoors e salute e cura della persona*

il cluster 1 ha la maggioranza degli elementi nella categoria *cellulari e accessori*

il cluster 2 ha la maggioranza degli elementi nella categoria *modelli di abbigliamento*

il cluster 3 ha la maggioranza degli elementi nella categoria *cosmetici e prodotti per la casa*

## K-means

Osserviamo ora il comportamento dei vari algoritmi di clustering sui dati dell'attributo `reviewText`. Iniziamo con k-means. Per migliorarne la performance ripetiamo l'algoritmo 10 volte e scegliamo il risultato con SSE migliore. La distanza scelta è la *cosine similarity*.

Testando k-means con  $k = 4$  vorremmo vedere se riesce ad identificare le 4 categorie. I risultati sono i seguenti:

- il cluster 0 (il più numeroso) copre quasi completamente le categorie *sports & outodoors* e *salute e cura della persona*
- il cluster 1 ha la maggioranza degli elementi nella categoria *cellulari e accessori*
- il cluster 2 ha la maggioranza degli elementi nella categoria *film & TV*
- il cluster 3 non è abbastanza grande per identificare chiaramente una categoria

## K-means

Osserviamo ora il comportamento dei vari algoritmi di clustering sui dati dell'attributo `reviewText`. Iniziamo con k-means. Per migliorarne la performance ripetiamo l'algoritmo 10 volte e scegliamo il risultato con SSE migliore. La distanza scelta è la *cosine similarity*.

Testando k-means con  $k = 4$  vorremmo vedere se riesce ad identificare le 4 categorie. I risultati sono i seguenti:

- il cluster 0 (il più numeroso) copre quasi completamente le categorie *sports & outodoors* e *salute e cura della persona*
- il cluster 1 ha la maggioranza degli elementi nella categoria *cellulari e accessori*
- il cluster 2 ha la maggioranza degli elementi nella categoria *film & TV*
- il cluster 3 non è abbastanza grande per identificare chiaramente una categoria

## K-means

Osserviamo ora il comportamento dei vari algoritmi di clustering sui dati dell'attributo [reviewText](#). Iniziamo con k-means. Per migliorarne la performance ripetiamo l'algoritmo 10 volte e scegliamo il risultato con SSE migliore. La distanza scelta è la *cosine similarity*.

Testando k-means con  $k = 4$  vorremmo vedere se riesce ad identificare le 4 categorie. I risultati sono i seguenti:

- il cluster 0 (il più numeroso) copre quasi completamente le categorie *sports & outodoors* e *salute e cura della persona*
- il cluster 1 ha la maggioranza degli elementi nella categoria *cellulari e accessori*
- il cluster 2 ha la maggioranza degli elementi nella categoria *film & TV*
- il cluster 3 non è abbastanza grande per identificare chiaramente una categoria

## K-means

Osserviamo ora il comportamento dei vari algoritmi di clustering sui dati dell'attributo [reviewText](#). Iniziamo con k-means. Per migliorarne la performance ripetiamo l'algoritmo 10 volte e scegliamo il risultato con SSE migliore. La distanza scelta è la *cosine similarity*.

Testando k-means con  $k = 4$  vorremmo vedere se riesce ad identificare le 4 categorie. I risultati sono i seguenti:

- il cluster 0 (il più numeroso) copre quasi completamente le categorie *sports & outodoors* e *salute e cura della persona*
- il cluster 1 ha la maggioranza degli elementi nella categoria *cellulari e accessori*
- il cluster 2 ha la maggioranza degli elementi nella categoria *film & TV*
- il cluster 3 non è abbastanza grande per identificare chiaramente una categoria

## K-means, continua...

Le parole chiave dei cluster sono

- cluster 0: product,great,work,use;
- cluster 1: look,protect,phone,case;
- cluster 2: one,watch,film,movi;
- cluster 3: phone,charger,charg,batteri

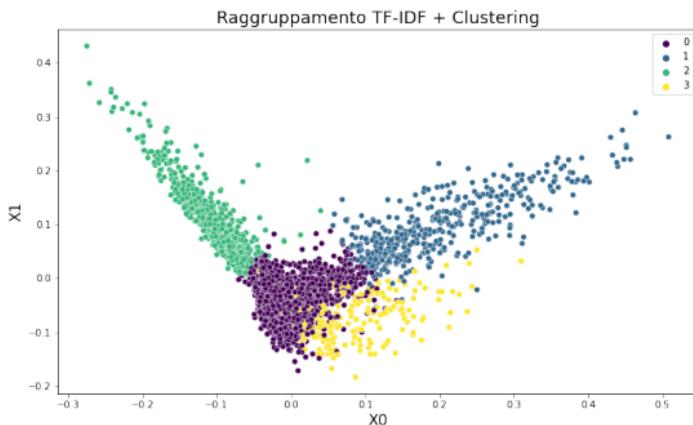


Figura: Clustering tramite k-means visualizzato con PCA

## Bisecting K-means

Per migliorare la robustezza di KMeans abbiamo testato Bisecting K-Means che notiamo però darci risultati peggiori in termini di SSE (3858.60 di K-Means e 3865.83 di Bisecting K-Means), con i quattro cluster che dividono meno chiaramente nelle 4 categorie.

Le parole chiave dei cluster sono

- cluster 0:  
one,watch,film,movi;
- cluster 1: get,one,work,use;
- cluster 2:  
use,price,great,product;
- cluster 3:  
screen,protect,phone,case;

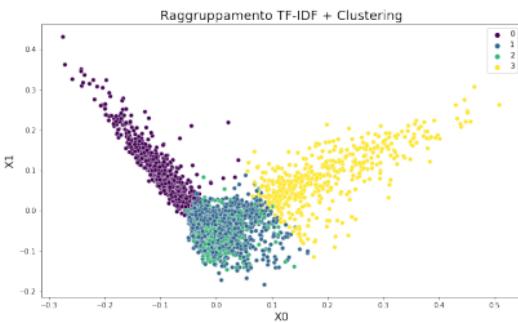


Figura: Clustering tramite bisecting k-means visualizzato con PCA

## Clustering gerarchico

Abbiamo confrontato poi con algoritmi di clustering gerarchico di tipo agglomerativo. I risultati sono stati abbastanza insoddisfacenti con diverse definizioni della inter-cluster similarity. L'unico caso in cui non si è creato un cluster con quasi tutti i nodi è stato con la *Ward's similarity*

Le parole chiave dei cluster sono

- cluster 0:  
product,great,work,use;
- cluster 1:  
look,protect,phone,case;
- cluster 2:  
one,watch,film,movi;
- cluster 3:  
phone,charger,charg,batteri;

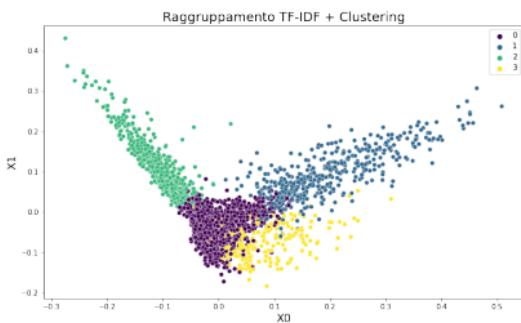


Figura: Clustering tramite algoritmo gerarchico agglomerativo visualizzato con PCA

## DBSCAN

Abbiamo testato l'algoritmo DBSCAN. La ricerca dei parametri è stata fatta tramite una grid search selezionando quelli con *silhouette* migliore. La distanza scelta è la *cosine similarity*.

I cluster identificati sono 5:

- il cluster -1 è quello che identifica il rumore
- il cluster 0 copre parte delle categorie *sports & outdoors* e *salute e cura della persona*
- il cluster 1 non è abbastanza grande per identificare chiaramente una categoria
- il cluster 2 ha la maggioranza degli elementi nella categoria *cellulari e accessori*
- il cluster 3 ha la maggioranza degli elementi nella categoria *film & TV*

## DBSCAN

Abbiamo testato l'algoritmo DBSCAN. La ricerca dei parametri è stata fatta tramite una grid search selezionando quelli con *silhouette* migliore. La distanza scelta è la *cosine similarity*.

I cluster identificati sono 5:

- il cluster -1 è quello che identifica il rumore
- il cluster 0 copre parte delle categorie *sports & outdoors* e *salute e cura della persona*
- il cluster 1 non è abbastanza grande per identificare chiaramente una categoria
- il cluster 2 ha la maggioranza degli elementi nella categoria *cellulari e accessori*
- il cluster 3 ha la maggioranza degli elementi nella categoria *film & TV*

## DBSCAN

Abbiamo testato l'algoritmo DBSCAN. La ricerca dei parametri è stata fatta tramite una grid search selezionando quelli con *silhouette* migliore. La distanza scelta è la *cosine similarity*.

I cluster identificati sono 5:

- il cluster -1 è quello che identifica il rumore
- il cluster 0 copre parte delle categorie *sports & outdoors* e *salute e cura della persona*
- il cluster 1 non è abbastanza grande per identificare chiaramente una categoria
- il cluster 2 ha la maggioranza degli elementi nella categoria *cellulari e accessori*
- il cluster 3 ha la maggioranza degli elementi nella categoria *film & TV*

## DBSCAN

Abbiamo testato l'algoritmo DBSCAN. La ricerca dei parametri è stata fatta tramite una grid search selezionando quelli con *silhouette* migliore. La distanza scelta è la *cosine similarity*.

I cluster identificati sono 5:

- il cluster -1 è quello che identifica il rumore
- il cluster 0 copre parte delle categorie *sports & outdoors* e *salute e cura della persona*
- il cluster 1 non è abbastanza grande per identificare chiaramente una categoria
- il cluster 2 ha la maggioranza degli elementi nella categoria *cellulari e accessori*
- il cluster 3 ha la maggioranza degli elementi nella categoria *film & TV*

## DBSCAN

Abbiamo testato l'algoritmo DBSCAN. La ricerca dei parametri è stata fatta tramite una grid search selezionando quelli con *silhouette* migliore. La distanza scelta è la *cosine similarity*.

I cluster identificati sono 5:

- il cluster -1 è quello che identifica il rumore
- il cluster 0 copre parte delle categorie *sports & outdoors* e *salute e cura della persona*
- il cluster 1 non è abbastanza grande per identificare chiaramente una categoria
- il cluster 2 ha la maggioranza degli elementi nella categoria *cellulari e accessori*
- il cluster 3 ha la maggioranza degli elementi nella categoria *film & TV*

## DBSCAN, continua...

Le parole chiave dei cluster sono:

- cluster 0: product,great,work,use;
- cluster 1: phone,charger,charg,batteri;
- cluster 2: screen,protect,phone,case;
- cluster 3: one,watch,film,movi.

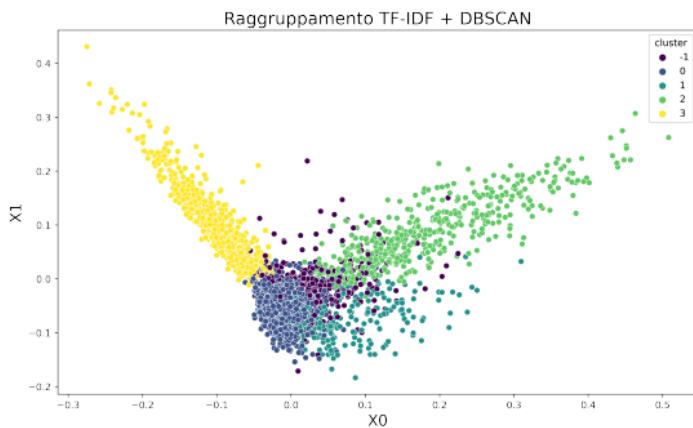


Figura: Clustering tramite DBSCAN visualizzato con PCA

## Pseudocodice per applicazione business

```
1: procedure get_suggestions(TextFiles)
2:   filter_words ← empty list
3:   while True do
4:     X ← preprocessing(TextFiles,filter_words)
5:     clusters ← cluster_algo(X)
6:     top_words ← get_keyword(clusters,X)
7:     cluster ← input()
8:     if cluster = -1 then
9:       visualizza(TextFiles)
10:      break
11:    end if
12:    filter_words ← insert (filter_words,top_words[cluster])
13:    TextFiles ← TextFiles[cluster]
14:   end while
15: end procedure
```

## Pseudocodice per applicazione business, continua...

Nello pseudocodice non abbiamo specificato il significato di:

- preprocessing()**: Nel nostro caso è la funzione che dai file testo ci dà un array con i valori del tf-idf, eliminando le parole scelte come identificative del cluster;
- cluster\_algo()**: Abbiamo lasciato il nome vago per generalità, nel nostro caso il miglior compromesso è k-means;
- get\_keywords()**: La funzione che da un sottoinsieme di testi ci dà la parola con il più alto tf-idf medio;
- visualizza()**: La funzione che ci permette di visualizzare i prodotti con le recensioni nel cluster individuato, contiene all'interno un algoritmo di ordinamento.

## Ordinamento

Per quanto riguarda la visualizzazione possono essere utilizzati due approcci:

- Visualizzo per primi gli oggetti con recensioni più vicine al centroide (pertinenza);
- Visualizzo per primo gli oggetti con recensioni considerate più utili (utilità).

Relativamente al secondo approccio, dato che la maggior parte delle recensioni hanno utilità 0 facciamo una regressione lineare per stimarne la loro utilità. Per poter utilizzare una regressione lineare in grado di darci dei risultati soddisfacenti riduciamo prima il dataset tramite PCA. In particolare scegliamo empiricamente il numero di componenti principali in modo da massimizzare l'indice  $R^2$  calcolato tra le previsioni stimate e i valori originali della colonna *helpful*.

## Un esempio

Facendo un esempio sulla categoria *salute e cura della persona* otteniamo i seguenti risultati:

- use, shave, work, product alla prima iterazione, scegliendo product otteniamo
- good, vitamin, tast, qualiti, take, pain, great alla seconda iterazione, scegliendo pain otteniamo
- garden, knee, hand, joint, black, tablet, doctor, red, scegliendo infine knee infine ci restituisce le seguenti recensioni

'I have osteo arthritis that affects almost all the joints in my body The Osteo BiFlex took the knee pain away with in a week of starting it It hasnt done much for the other areas of arthritis pain but its great not having sharp knee pain when I walk I used it previously for a couple of months and when I stopped taking it my knee pain returned I started taking it again no knee pain I will continue to take it from here on out Amazon subscribe save has the best price I have a bottle shipped to my house every 2 months'

Figura: Recensione 3091

## Un esempio, continua...

'Aches muscle fatigue bruising all over pain This is the product for muscle and subcutaneous stuff However it does not address nerve issues I have been dealing with a knee injury and the Traumeel didnt help However Topricin did the trick While the contain says its for feet its actually for any area on the body that has nerve pain I used the Topricin and my knee pain in conjunction with physical therapy went away I will save the Traumeel for regular aches and pains'

Figura: Recensione 3190

'Good Price and seems like a sturdy product However I am still not sure if they do anything for me I use them when i play football and my knees are still very sore after playing'

Figura: Recensione 3944

## Miglioramenti

Possibili miglioramenti:

- possibilità di ricercare una parola specifica e individuare un cluster coerente (si potrebbero usare regole di associazione). Ad esempio, cercare "regalo" ed essere indirizzati ad oggetti adeguati per un regalo;
- trovare altre possibili forme di ordinamento;
- modi per velocizzare l'algoritmo su grandi moli di dati.

## Miglioramenti

Possibili miglioramenti:

- possibilità di ricercare una parola specifica e individuare un cluster coerente (si potrebbero usare regole di associazione). Ad esempio, cercare "regalo" ed essere indirizzati ad oggetti adeguati per un regalo;
- trovare altre possibili forme di ordinamento;
- modi per velocizzare l'algoritmo su grandi moli di dati.

## Miglioramenti

Possibili miglioramenti:

- possibilità di ricercare una parola specifica e individuare un cluster coerente (si potrebbero usare regole di associazione). Ad esempio, cercare "regalo" ed essere indirizzati ad oggetti adeguati per un regalo;
- trovare altre possibili forme di ordinamento;
- modi per velocizzare l'algoritmo su grandi moli di dati.



Grazie per l'attenzione

