

---

# AMAZON FINE FOOD REVIEWS: TEXT MINING TECHNIQUES

## PROGETTO DEL CORSO TEXT MINING AND SEARCH

---

**Lorgna Lorenzo**

Corso di Laurea Magistrale in Data Science  
Università degli Studi di Milano-Bicocca  
l.lorgna@campus.unimib.it - 829776

**Marzorati Stefano**

Corso di Laurea Magistrale in Data Science  
Università degli Studi di Milano-Bicocca  
s.marzorati11@campus.unimib.it - 830272

*14 Febbraio 2022*

**ABSTRACT**

Il progetto si propone di analizzare attraverso tecniche di text mining un dataset costituito da recensioni di prodotti provenienti dal colosso dell'e-commerce Amazon. Si tratta di recensioni che rientrano principalmente nella categoria "fine foods", nonostante vi siano anche altre categorie di prodotti contemplate. A partire dal testo di queste recensioni in un primo momento è stata svolta un'analisi esplorativa per poi procedere con attività di data cleaning e preprocessing. A seguire sono stati applicati modelli di classificazione, binaria e multiclasse, di clustering e infine di topic modelling. I modelli di classificazione, Logistic Regression, SVM e Random Forest, hanno permesso di classificare una recensione a partire dal suo testo con un valore di score, 0/1 nel caso binario o da 1 a 5 nel caso multi classe. I modelli di clustering, K-means e Agglomerative Clustering, hanno invece cercato di raggruppare in cluster recensioni che avessero delle caratteristiche simili. Infine tramite tecniche di topic modelling, LDA, si è andati alla ricerca delle tematiche di maggior rilievo che emergessero dai testi delle recensioni a disposizione.

**Keywords** Exploration · Text Preprocessing · Text Representation · Text Classification · Text Clustering · Topic Modelling

## **Contents**

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Domande di ricerca</b>	<b>3</b>
<b>3</b>	<b>Data Cleaning</b>	<b>4</b>
<b>4</b>	<b>Data Exploration</b>	<b>4</b>
<b>5</b>	<b>Text Preprocessing</b>	<b>6</b>
<b>6</b>	<b>Classificazione Binaria</b>	<b>7</b>
6.1	Text Representation . . . . .	7
6.2	Implementazione algoritmi . . . . .	8
<b>7</b>	<b>Clustering</b>	<b>9</b>
<b>8</b>	<b>Single-Label Multi-Class Classification</b>	<b>10</b>
<b>9</b>	<b>Topic Modelling</b>	<b>11</b>
<b>10</b>	<b>Conclusioni</b>	<b>12</b>

## 1 Introduzione

Lo shopping online è ormai diventato una costante al giorno d'oggi, una comodità che permette di risparmiare sia dal punto di vista economico che dal punto di vista del tempo. Parte integrante dello shopping sono le recensioni che guidano l'utente tra la moltitudine di offerta e permettono una scelta più consapevole basata sull'esperienza di altri clienti. Solitamente una recensione si compone di un testo e uno score, il cui valore permette di riassumere il giudizio complessivo dell'utente. Questo permette una migliore organizzazione e presentazione non lasciando il cliente in situazioni di incertezza.

Il leader in questo settore è Amazon, che con la sua vastissima offerta di prodotti può fornire conseguentemente una moltitudine di recensioni. Un sottoinsieme di queste rappresentano il dataset e il punto di partenza del progetto. Nello specifico il dataset preso in considerazione è relativo alle recensioni di "Amazon Fine Foods".

Le recensioni contenute nel dataset, circa 500.000, spaziano in un periodo di più di 10 anni, da Ottobre 1999, a Ottobre 2012. Tali recensioni includono al loro interno informazioni relative sia al prodotto che all'utente, presentando anche il rating relativo e il testo completo della recensione. È bene precisare che le recensioni non sono solamente relative ad "Amazon Fine Foods", ma riguardano anche altre categorie presenti nello store online.

Andando più nello specifico si mostrano le caratteristiche principali del dataset preso in esame:

- Numero di recensioni: 568,454
- Numero di utenti: 256,059
- Numero di prodotti: 74,258
- Arco temporale: Ottobre 1999 - Ottobre 2012
- Numero di feature: 10

Prendendo invece in considerazione le features del dataset si identificano:

- **Id**: identificativo univoco della recensione
- **ProductId**: identificativo univoco del prodotto
- **UserId**: identificativo univoco dell'utente
- **ProfileName**: nome dell'utente
- **HelpfulnessNumerator**: numero di utenti che hanno valutato la recensione utile
- **HelpfulnessDenominator**: numero di utenti che hanno indicato se la recensione è stata utile o meno
- **Score**: score assegnato alla recensione che varia tra 1 e 5
- **Time**: data della recensione
- **Summary**: breve riassunto della recensione
- **Text**: testo della recensione

## 2 Domande di ricerca

Con l'intento di definire in maniera più precisa l'effettivo obiettivo del progetto sono state identificate le seguenti domande di ricerca:

- *A partire dal testo di una recensione è possibile classificare quest'ultima con un'etichetta positiva o negativa, cercando dunque di prevedere la valutazione dell'utente?*
- *Si possono individuare gruppi di recensioni con caratteristiche simili?*
- *A partire dal testo di una recensione è possibile classificare quest'ultima in una delle 5 modalità di "Score", cercando dunque di prevedere la valutazione dell'utente?*
- *È possibile estrarre le tematiche di maggior rilievo che emergono dalle recensioni e in caso positivo quali sono?*

### 3 Data Cleaning

Dopo aver effettuato una prima analisi esplorativa ad alto livello e avendo constatato la presenza di duplicati e inconsistenze nel dataset, si è proceduto ove possibile con procedure di data cleaning.

Il primo passo è stata la rimozione dei duplicati, in particolare basandosi sulle features *UserId*, *ProfileName*, *Time*, *Text* si è provveduto ad eliminare circa il 30% delle osservazioni presenti inizialmente nel dataset. La motivazione della scelta di tali features è dovuta dal fatto che non sia possibile per un utente scrivere più recensioni aventi lo stesso testo nel medesimo istante.

A seguire, focalizzando l'attenzione sulle features *HelpfulnessNumerator* e *HelpfulnessDenominator*, per ottenere una maggiore coerenza all'interno dei dati, sono stati rimossi 2 record il cui valore di *HelpfulnessNumerator* risultava maggiore di *HelpfulnessDenominator*. Questo non può accedere dal momento in cui il numero di recensioni utili non può essere maggiore della totalità delle recensioni prodotte. Viene riportato in Figura 1 il caso analizzato.

	<b><i>Id</i></b>	<b><i>ProductId</i></b>	<b><i>UserId</i></b>	<b><i>ProfileName</i></b>	<b><i>HelpfulnessNumerator</i></b>	<b><i>HelpfulnessDenominator</i></b>
	<b>44736</b>	44737	B001EQ55RW	A2V0I904FH7ABY	Ram	3
	<b>64421</b>	64422	B000MIDROQ	A161DK06JJMCYF	J. E. Stephens "Jeanne"	3

Figure 1: Inconsistenza Helpfulness

Infine prendendo in considerazione *Score*, *Summary*, *Text*, features rilevanti per le nostre analisi, sono stati rilevati 3 record presentanti missing value, in particolare per la feature *Summary*. Il problema è stato risolto introducendo una stringa vuota al posto del relativo valore mancante.

Conclusa la fase di data cleaning il numero di record è pari a 393.931.

### 4 Data Exploration

Dopo una prima fase di data cleaning descritta nella sezione precedente, si propone un'analisi esplorativa più dettagliata dei dati a disposizione.

Come primo passo il focus è stato posto sulle features principali dello studio, in particolare *Score* e *Text* analizzandone le loro caratteristiche.

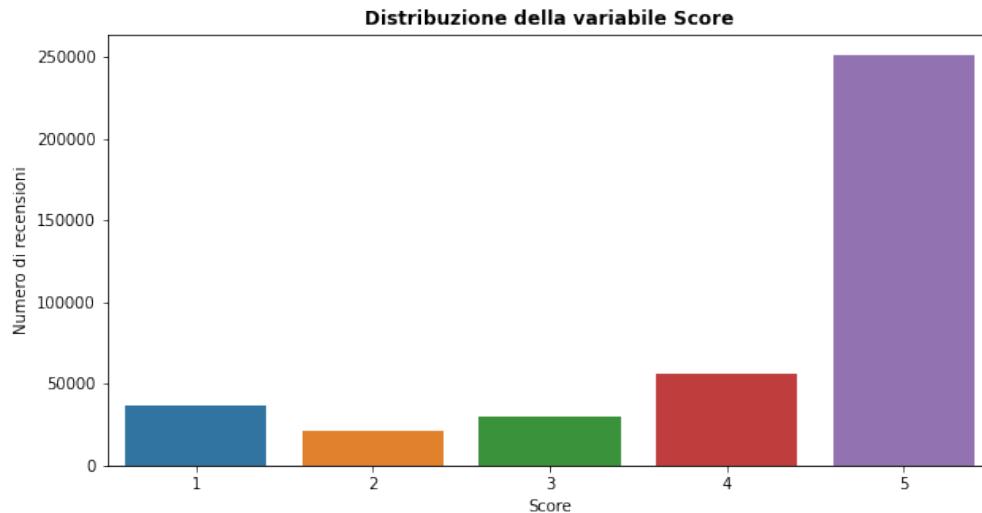


Figure 2: Distribuzione della feature *Score*

Osservando la distribuzione della variabile *Score* in Figura 2 è possibile notare che la maggior parte delle recensioni (250961) presentano un valore di *Score* pari a 5, mentre per le altre modalità si nota una maggiore uniformità nel numero di recensioni. Il valore di *Score* per cui sono presenti meno recensioni è 2 (20802).

Concentrandosi sulla feature *Text*, calcolando per ognuna delle recensioni il numero di parole presenti in essa, è stato realizzato il violin plot, che è possibile osservare in Figura 3. Quest'ultimo è rappresentativo delle recensioni che presentano meno di 500 parole nella feature *Text* per avere una visione più chiara sulla distribuzione, per ovviare alla presenza di recensioni anomale, tra le quali si identificano recensioni contenenti un numero di parole fino a 3500.

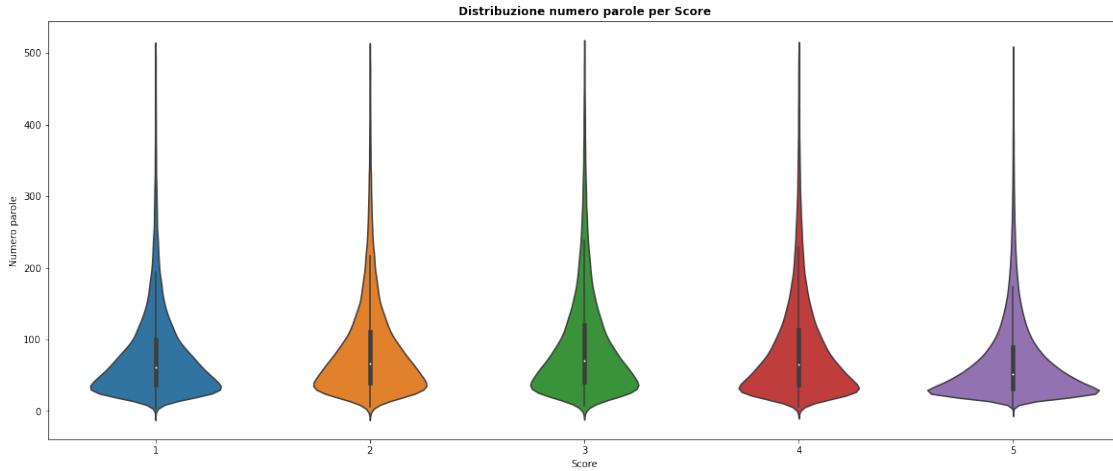


Figure 3: Distribuzione del numero di parole per *Score*

Analizzando il violin plot si nota come la distribuzione per lo *Score* 5 presenti una mediana inferiore alle altre, più precisamente pari a 52. Questo potrebbe derivare dal fatto che un utente soddisfatto dell'acquisto non abbia la necessità di dilungarsi in commenti negativi.

Pur non trattandosi di features centrali nelle analisi che verranno presentate di seguito, nei seguenti grafici si procede con un'esplorazione delle variabili *Time*, *HelpfulnessNumerator* e *HelpfulnessDenominator*.

Considerando la feature *Time*, osservando la Figura 4, è possibile notare che la maggior parte delle recensioni risalgono agli ultimi anni presenti nel dataset. Questa evidenza rispecchia l'adozione e l'evoluzione delle piattaforme e-commerce sempre più evidente degli ultimi anni.

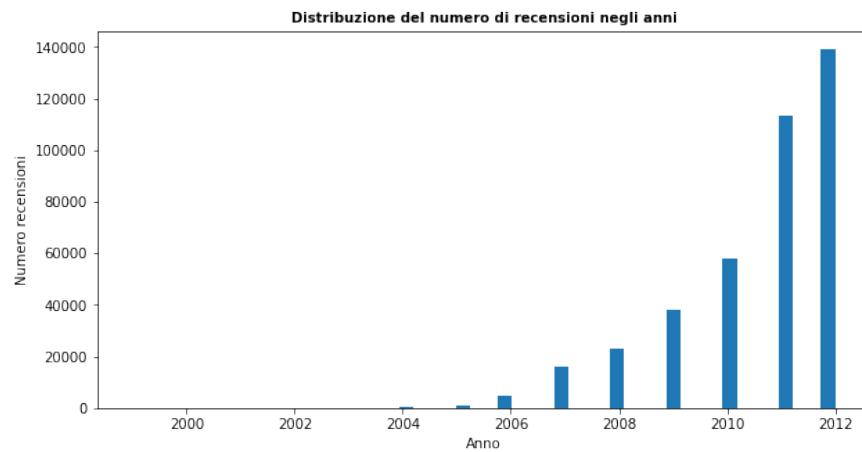


Figure 4: Distribuzione del numero recensioni nel tempo

Considerando le features *HelpfulnessNumerator* e *HelpfulnessDenominator* si è generata una nuova variabile, *PercentHelpful*, che permette di indicare per un prodotto quante delle recensioni realizzate sono state indicate come realmente utili.

$$PercentHelpful = \frac{HelpfulnessNumerator}{HelpfulnessDenominator} * 100 \quad (1)$$

Analizzando la Figura 5 si può notare che la maggior parte delle recensioni presentano un valore di *PercentHelpful* pari a 100%: questo può indicare la grande utilità delle recensioni come guida per l'utente nell'acquisto.

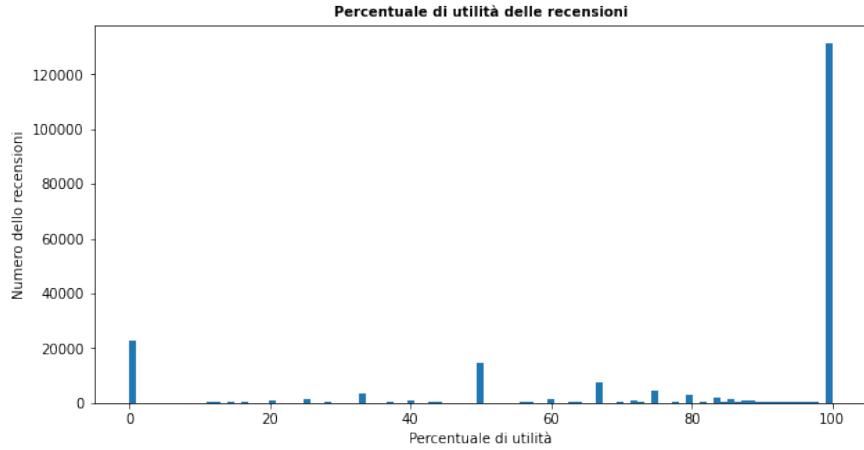


Figure 5: Distribuzione dell'utilità recensione

Considerando la totalità dei testi delle recensioni viene generato un word cloud, Figura 6, in modo da intuire ad alto livello le tematiche principali caratteristiche del dataset.

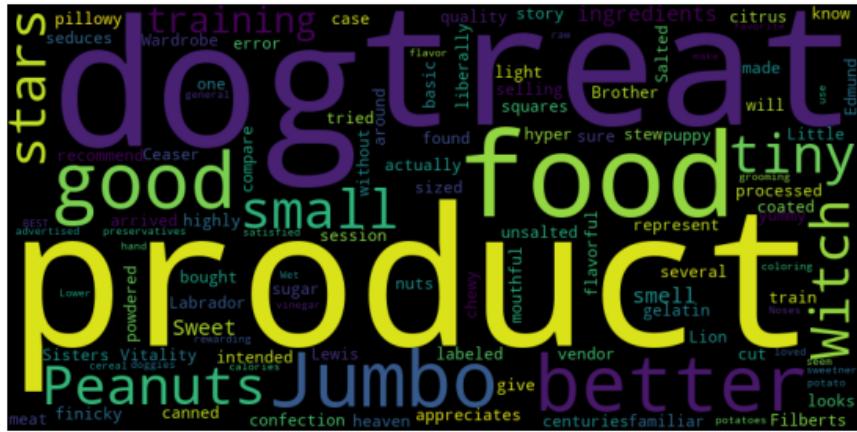


Figure 6: Word Cloud di *Text*

## 5 Text Preprocessing

Prima di procedere con l'esecuzione dei task per identificare delle risposte per le domande di ricerca inizialmente presentate, è necessario svolgere una fase di preprocessing sul testo.

Ciò permette di migliorare l'efficienza degli algoritmi utilizzati successivamente e al tempo stesso di uniformare e ripulire il testo.

**Normalization** È un'operazione eseguita sul testo e permette di eliminare le ambiguità presenti. Si compone di diverse fasi, in questo caso:

- *Conversione in caratteri minuscoli*: si è convertito tutto il testo in caratteri minuscoli, evitando potenziali duplicazioni delle parole e portando a una riduzione del feature set
- *Rimozione emoji*: vengono rimosse tutte le emoji presenti all'interno del testo
- *Rimozione di numeri*: vengono rimossi tutti i numeri presenti all'interno del testo
- *Espansione delle contrazioni*: dal momento in cui i testi sono in lingua inglese e sono presenti diverse contrazioni, per uniformare il tutto queste vengono riportate nella loro forma estesa
- *Rimozione HTML*: vengono rimossi tutti i tag HTML presenti all'interno del testo
- *Rimozione URL*: vengono rimossi tutti gli URL (www, http) presenti all'interno del testo
- *Rimozione di parole che presentano al loro interno 3 o più caratteri ripetuti*
- *Rimozione di punteggiatura e caratteri speciali*

**Tokenization** Dopo aver trattato il testo con normalization è stata effettuata la tokenization. Consiste nella suddivisione delle frasi in parole. Ciascuna parola è da considerarsi come un token separato. La tokenization risulta dunque essere un primo step fondamentale per poi eseguire ulteriori operazioni di preprocessamento che, altrimenti, risulterebbero complesse o inefficaci.

**Stop-words removal** Parole dallo scarso contenuto semantico come articoli, preposizioni, congiunzioni e avverbi sono esempi di stopwords. La loro presenza potrebbe interferire con il processamento del linguaggio, quindi parole come ‘a’, ‘an’, ‘the’, ‘is’, ‘are’, ‘which’, ‘at’ e ‘on’ sono state rimosse basandosi su un vocabolario presente nella libreria NLTK.

**Lemmatization** Si tratta di un’operazione che riduce le diverse forme di una parola nella loro forma base chiamata lemma. La lemmatization per ottenere il lemma, piuttosto che utilizzare regole predefinite o euristiche, si basa sull’analisi del contesto e di regole lessicali. Si è preferito utilizzare la lemmatization a scapito dello stemming, nonostante vengano richieste più risorse, visto che si tratta di un’operazione più accurata e in quanto lo stemming potrebbe alla generazione di parole non presenti nel vocabolario.

## 6 Classificazione Binaria

In questa sezione si cerca di dare risposta alla prima domanda di ricerca presentata precedentemente: *A partire dal testo di una recensione è possibile classificare quest’ultima con un’etichetta positiva o negativa, cercando dunque di prevedere la valutazione dell’utente?*

Per fare ciò si utilizzano tre diversi algoritmi di classificazione supervisionata, ovvero Logistic regression, Support-Vector Machines (SVM) e Random Forest.

I dati utilizzati per l’addestramento e la validazione di ciascuno dei modelli sono stati suddivisi grazie ad una procedura di *holdout*. Essa permette di suddividere la totalità dei dati in training set (67%) e test set (33%).

Dal momento in cui la feature *Score* presenta 5 modalità per svolgere una classificazione binaria vi è l’esigenza di una feature che indichi se una recensione risulta essere *negativa* (valore di *Score* 1-2) oppure *positiva* (valore di *Score* 4-5). Per quanto riguarda invece le recensioni il cui valore di *Score* è pari a 3 non sono state considerate in quanto ritenute neutrali.

Un ulteriore passaggio è stato quello di gestire lo sbilanciamento delle classi del training set, dal momento in cui è stato osservato che la classe *positiva* era parte predominante delle recensioni. In termini pratici è stato effettuato un undersampling della classe positiva ottenendo le classi perfettamente bilanciate come si può osservare in Figura 7.

### 6.1 Text Representation

Per l’applicazione dei modelli sopracitati è necessaria una rappresentazione formale del testo delle recensioni considerate. Tra le diverse tipologie di rappresentazione si è optato per Bag Of Words (BOW) e TF-IDF.

**BOW** Permette di rappresentare in forma vettoriale il testo delle recensioni. Il contenuto del vettore è il numero di occorrenze di ciascuna parola presente in esso.

Il numero di features ottenute con la rappresentazione BOW è pari a 49449.

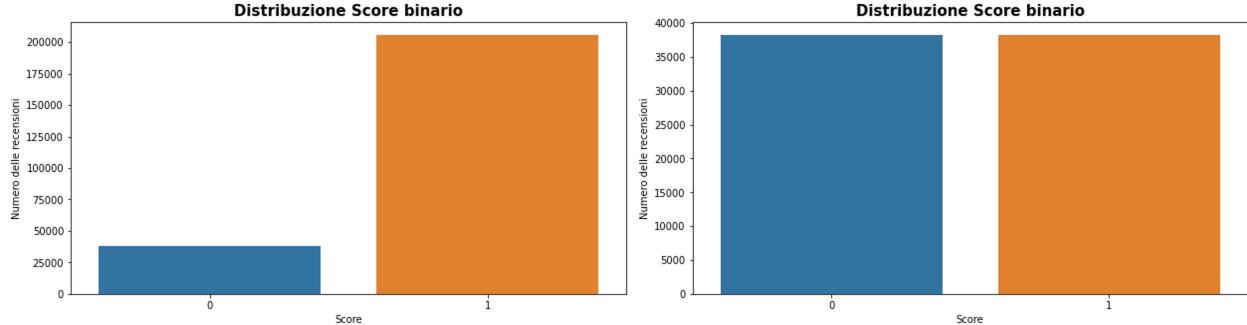


Figure 7: Distribuzione score binario

**TF-IDF** Con l'utilizzo di BOW ci si basa solamente sul numero di occorrenze delle parole. Per avere una rappresentazione più efficace si può utilizzare la rappresentazione TF-IDF che permette di andare a misurare quanto una parola è importante rispetto all'intero documento e all'intero corpus.

Il numero di features ottenute con la rappresentazione TF-IDF è pari a 14433.

**Dimensionality Reduction** Entrambe le rappresentazioni, BOW e TF-IDF, hanno portato a rappresentazioni composte da matrici molto grandi, dall'elevato numero di features. Per questo motivo nell'ottica di migliorare le performance in termini computazionali si è proceduto applicando una tecnica di dimensionality reduction, in particolare Singular value decomposition (SVD).

Il numero di componenti mantenuto è di 1.000 per la rappresentazione BOW, mentre 2.000 per quella TF-IDF. La varianza spiegata ottenuta in seguito alla riduzione della dimensionalità è pari a 0.792 per BOW e 0.747 per TF-IDF.

## 6.2 Implementazione algoritmi

I modelli considerati sono stati implementati considerando entrambe le rappresentazioni, BOW e TF-IDF. Inoltre tutti modelli appartengono alla libreria *sklearn*.

**Logistic Regression** Si tratta di un classificatore binario dal momento in cui calcola la probabilità che una determinata istanza appartenga a una specifica classe. In particolare se la probabilità stimata supera il 50% il modello prevede che l'istanza appartenga alla classe positiva, o negativa.

Di seguito vengono presentati per entrambe le rappresentazioni i risultati ottenuti:

16174	2672
12833	88495

	Precision	Recall	F1-Score
0	0.56	0.86	0.68
1	0.97	0.87	0.92
AVG	0.75	0.87	0.80
Accuracy		0.87	

Matrice di confusione e performance BOW

16585	2261
12188	89140

	Precision	Recall	F1-Score
0	0.58	0.88	0.70
1	0.98	0.88	0.93
AVG	0.78	0.88	0.81
Accuracy		0.88	

Matrice di confusione e performance TF-IDF

**SVM** Si tratta di un modello di classificazione il cui obiettivo è quello di trovare un iperpiano di separazione delle classi che massimizza il margine tra le classi stesse. Si procede con una versione approssimata (RBFSampler) di questo per ottenere migliori performance in termini di tempo. I risultati sono però meno soddisfacenti rispetto agli altri modelli. Di seguito vengono presentati per entrambe le rappresentazioni i risultati ottenuti:

7689	11157
41253	60075

	Precision	Recall	F1-Score
0	0.16	0.41	0.23
1	0.84	0.59	0.70
AVG	0.50	0.50	0.46
Accuracy	0.56		

Matrice di confusione e performance BOW

**Random forest** L'ultimo algoritmo utilizzato è un random forest con un numero di alberi pari a 100. L'indice per misurare la qualità degli split risulta essere quello di *Gini*. Di seguito vengono presentati per entrambe le rappresentazioni i risultati ottenuti:

15227	3619
32883	68445

	Precision	Recall	F1-Score
0	0.32	0.81	0.45
1	0.95	0.68	0.79
AVG	0.63	0.74	0.74
Accuracy		0.70	

Matrice di confusione e performance BOW

12470	6376
49464	51864

	Precision	Recall	F1-Score
0	0.20	0.66	0.31
1	0.89	0.51	0.65
AVG	0.55	0.59	0.48
Accuracy		0.54	

Matrice di confusione e performance TF-IDF

14984	3862
25298	76030

	Precision	Recall	F1-Score
0	0.37	0.80	0.51
1	0.95	0.75	0.84
AVG	0.66	0.77	0.67
Accuracy		0.76	

Matrice di confusione e performance TF-IDF

## 7 Clustering

In questa sezione si è cercato di rispondere alla seconda domanda di ricerca: *Si possono individuare gruppi di recensioni con caratteristiche simili?*

Per identificare una risposta si sono implementati due algoritmi di clustering:

- K-means
- Agglomerative clustering

Il numero di cluster è stato posto pari a 5, corrispondente al numero di modalità della feature *Score*.

Prima dell'esecuzione degli algoritmi di clustering si è provveduto a risolvere il problema della class imbalance procedendo in modo analogo come descritto per la classificazione e a rappresentare i dati i dati tramite TF-IDF. Inoltre è stata effettuata la riduzione della dimensionalità e infine una normalizzazione dei dati.

**K-means** Il risultato ottenuto dall'applicazione dell'algoritmo K-Means ha prodotto per V measure (misura che esprime l'omogeneità del cluster) e ARI (misura che esprime la similarità tra cluster) i rispettivi valore pari a 0.006884 e 0.006316.

Considerando i singoli cluster vengono riportati di seguito il numero di recensioni appartenenti a ciascuno di essi:

1	34428
2	5067
3	6208
4	4039
5	19943

Cluster K-means

Sulla base dei risultati ottenuti tramite la generazioni di Word cloud, Figura 8, dei cluster ottenuti si possono individuare le caratteristiche principali di ciascuno di essi, andando ad individuare caratteristiche specifiche.

**Agglomerative clustering** Nel caso dell'algoritmo Agglomerative clustering i valori ottenuti per V measure e ARI sono rispettivamente di 0.001802 e 0.000394.



Figure 8: Word Cloud cluster K-means

Considerando i singoli cluster vengono riportati di seguito il numero di recensioni appartenenti a ciascuno di essi:

1	16173
2	1806
3	822
4	926
5	273

Cluster Agglomerative clustering

Si riportano di seguito le Word Cloud, Figura 9, generate a partire dai cluster ottenuti.

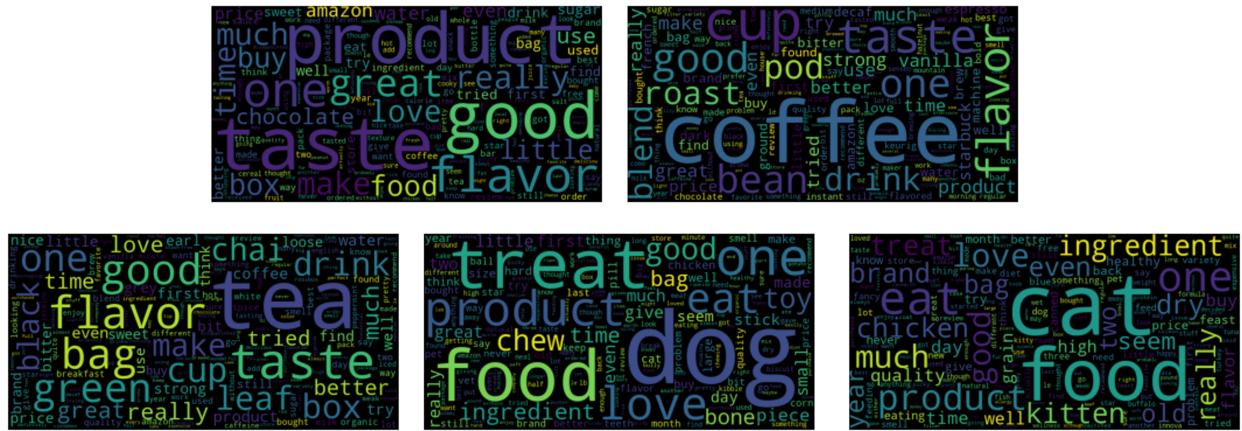


Figure 9: Word Cloud cluster Agglomerative clustering

## 8 Single-Label Multi-Class Classification

Considerando la terza domanda di ricerca, *A partire dal testo di una recensione è possibile classificare quest'ultima in una delle 5 modalità di "Score", cercando dunque di prevedere la valutazione dell'utente*, è stato implementato come algoritmo un modello Random Forest. Quest'ultimo è stato applicato sul testo rappresentato con TF-IDF, considerando in questo caso le 5 modalità che la feature Score può assumere.

Di seguito vengono riportati i risultati ottenuti.

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>1</b>	6207	2413	1354	946	1061
<b>2</b>	2132	1777	1347	910	699
<b>3</b>	1975	2195	2359	1928	1367
<b>4</b>	2262	2818	3547	5032	4852
<b>5</b>	9485	8617	9340	16612	38763

Matrice di confusione

	Precision	Recall	F1-Score
1	0.28	0.52	0.36
2	0.10	0.26	0.14
3	0.13	0.24	0.17
4	0.20	0.27	0.23
5	0.83	0.47	0.60
AVG	0.31	0.35	0.30
Accuracy		0.42	

Matrice di performance

Le performance ottenute, in questo caso, rispetto ai risultati ottenuti con la classificazione binaria, sono inferiori.

## 9 Topic Modelling

Per rispondere all'ultima domanda di ricerca, *È possibile estrarre le tematiche di maggior rilievo che emergono dalle recensioni e in caso positivo quali sono?*, si è deciso di procedere mediante l'approccio LDA (Latent Dirichlet Allocation). A differenza del clustering il cui obiettivo è raggruppare documenti, in questo caso recensioni, simili, tramite topic modelling si vanno ad identificare dei gruppi di parole simili. Si individuano perciò le tematiche di maggior rilievo che emergono dalla totalità delle recensioni a disposizione. Nel caso di LDA si assume che la distribuzione delle parole all'interno delle recensioni sia secondo la distribuzione Dirichlet. Il numero di topic in questo caso è stato scelto pari a 5.

Di seguito vengono riportati risultati ottenuti, in particolare per il *topic 3* e il *topic 4*.

Come si può osservare in Figura 10 nel *topic 3* si identificano parole relative anche ad animali, come dog e cat. Questo potrebbe indicare che le recensioni dei prodotti in questione riguardano il mondo animale.

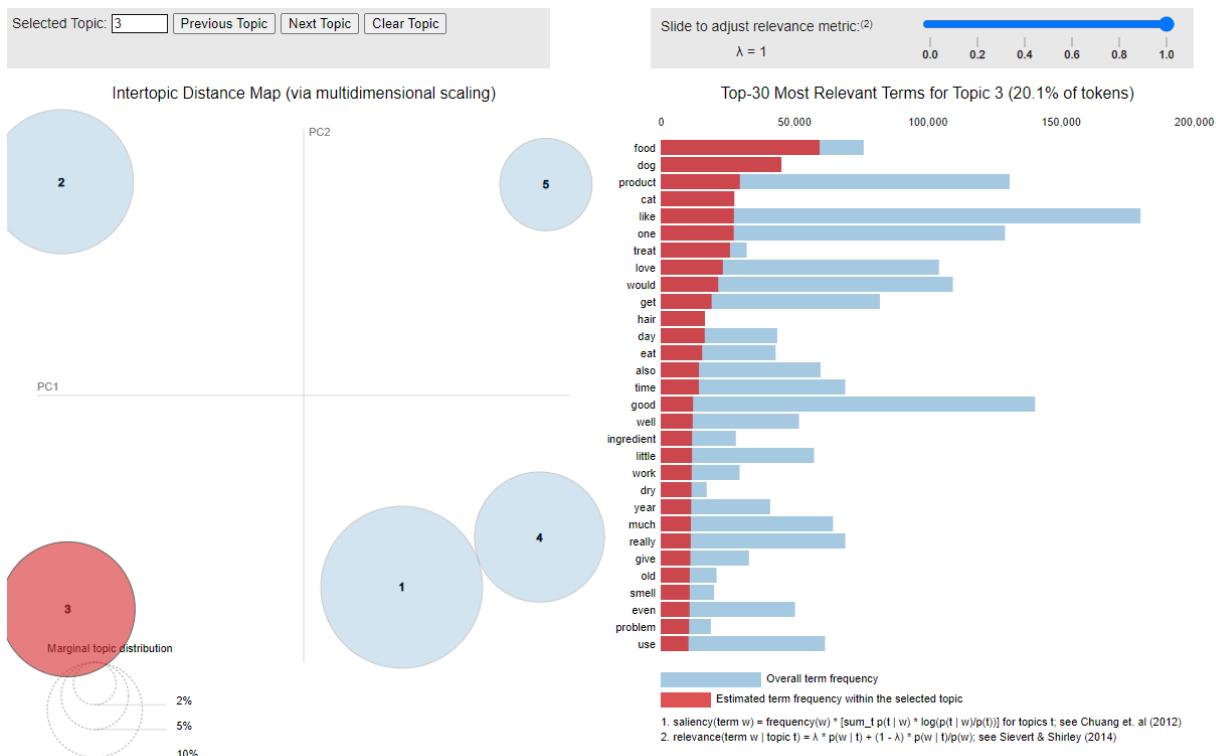


Figure 10: Topic Modelling: topic 3

Nel topic 4, come è possibile osservare in Figura 11, sono presenti come termini frequenti parole associabili a bevande e termini specifici relativi al gusto.

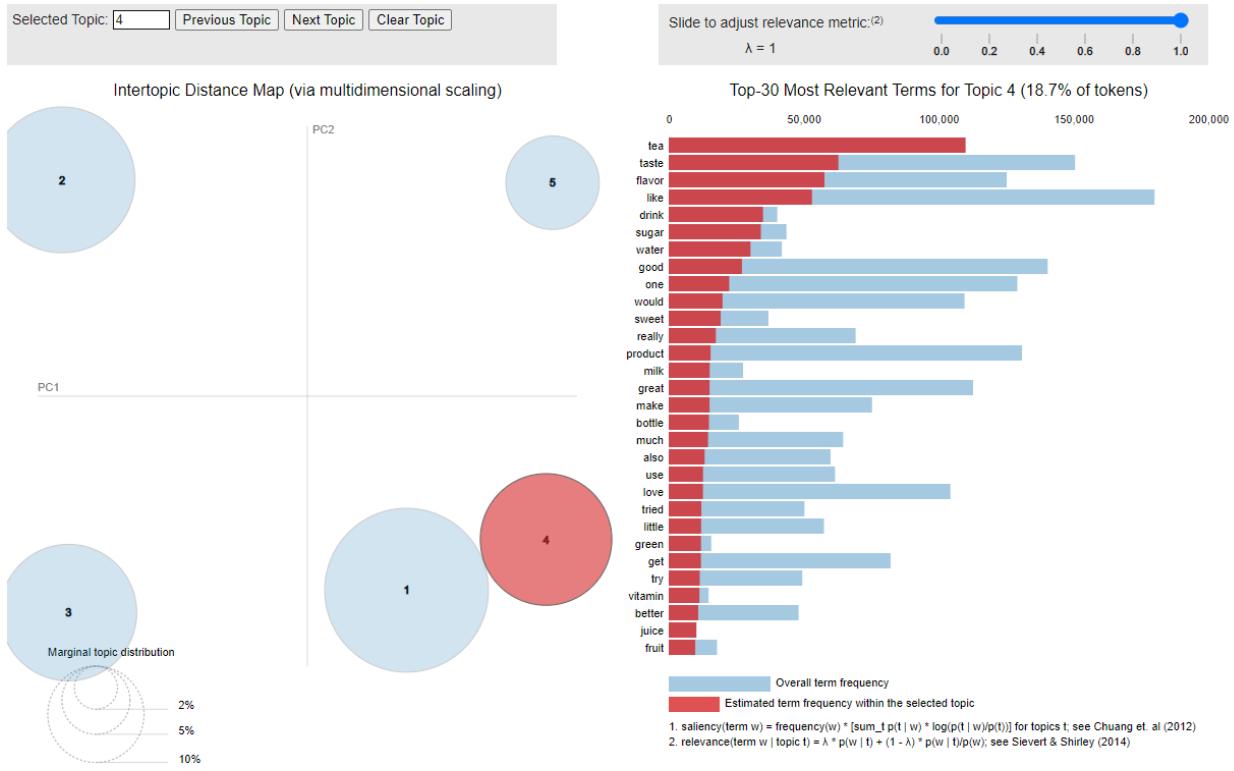


Figure 11: Topic Modelling: topic 4

## 10 Conclusioni

Si conclude cercando di dare una risposta alle domande formulate inizialmente:

- A partire dal testo di una recensione è possibile classificare quest'ultima con un'etichetta positiva o negativa, cercando dunque di prevedere la valutazione dell'utente?: è possibile rispondere a questa domanda tramite algoritmi di classificazione binaria. In questo caso il modello che in termini di accuracy ottiene il risultato migliore è la Logistic Regression con l'utilizzo di una rappresentazione TF-IDF che presenta un valore di accuracy pari a 0.88.
- Si possono individuare gruppi di recensioni con caratteristiche simili?: è possibile identificare dei cluster ognuno con recensioni simili in termini di contenuto e differente dagli altri.
- A partire dal testo di una recensione è possibile classificare quest'ultima in una delle 5 modalità di "Score", cercando dunque di prevedere la valutazione dell'utente?: risulta più complesso ottenere una classificazione multiclasse. Essendo molta soggettiva la valutazione in termini di Score, seppur esistendo una relazione tra il contenuto e il voto di una recensione, gli algoritmi non sono in grado di classificarla in modo soddisfacente. I risultati ottenuti sono inferiori rispetto a quelli ottenuti tramite classificazione binaria.
- È possibile estrarre le tematiche di maggior rilievo che emergono dalle recensioni e in caso positivo quali sono?: attraverso tecniche di topic modelling è possibile constatare che i topic sono direttamente legati alle categorie di prodotti venduti oppure temi riguardanti gli e-commerce.

## References

- [1] Stanford Network Analysis Project. *Amazon Fine Food Reviews*, <https://www.kaggle.com/snap/amazon-fine-food-reviews>, 2017.
- [2] J. McAuley and J. Leskovec. *From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews*, 2013.
- [3] G. Pasi and M. Viviani. *Text Mining and search course lecture notes and slides*, a.a. 2021/2022.