# Druglib reviews: tecniche di text mining per lo studio degli effetti collaterali dei farmaci

---

Anna Mattiroli, matricola 826381, email: a.mattiroli@campus.unimib.it
Beatrice Barzaghi, matricola 831829, email: b.barzaghi3@campus.unimib.it
Guglielmo Muoio, matricola 826029, email: g.muoio@campus.unimib.it

Anno Accademico 2021/2022

## Indice

**1 Introduzione** . . . . . **1**
**2 Analisi esplorativa** . . . . . **2**
**3 Pre-processing** . . . . . **3**
**4 Text Representation** . . . . . **4**
**5 Text Classification** . . . . . **4**
**6 Text Clustering** . . . . . **7**
**7 Conclusioni** . . . . . **10**

## 1 Introduzione

Le recensioni dei farmaci giocano un ruolo importante nel fornire informazioni cliniche sia per i medici sia per i pazienti. I consumatori utilizzano i siti di recensioni online per dare voce alle proprie opinioni ed esprimere pareri sui farmaci utilizzati. Tipicamente le recensioni online di farmaci sono formate da due elementi principali: un commento testuale ed una valutazione. Quest'ultima indica la valutazione complessiva dell'utente utilizzando una scala numerica mentre i commenti testuali offrono maggiori dettagli sull'efficacia e sugli effetti collaterali dei farmaci. L'obiettivo principale del progetto è analizzare un dataset proveniente da DrugLib, uno tra i più grandi e più visitati siti tra le risorse informative farmaceutiche, attraverso tecniche di text mining. A seguito di un'opportuna analisi esplorativa, sono state impiegate delle tecniche di preparazione dei dati, come preprocessing e text representation, focalizzando l'attenzione sui commenti relativi agli effetti collaterali dei pazienti. Successivamente sono stati sviluppati 2 task: uno di classificazione testuale ed uno di clustering.

### Domande di ricerca

**Task 1**: è possibile classificare la gravità degli effetti collaterali di un farmaco basandosi sulle opinioni espresse dagli utenti online?

**Task 2**: è possibile dividere in gruppi i farmaci in base agli effetti collaterali che gli utenti riportano nelle loro recensioni?

## 2 Analisi esplorativa

Il dataset utilizzato per sviluppare i due task è composto da un totale di 4143 osservazioni e 8 attributi tra cui:

- urlDrugName (categorical): nome del farmaco (540 valori unici)
- condition (categorical): condizione clinica per cui il farmaco è stato assunto
- benefitsReview (text): review sui benefici ottenuti dal paziente
- sideEffectsReview (text): review sugli effetti collaterali avuti dal paziente
- commentsReview (text): commento generale del paziente
- rating (numerical): valutazione da 1 a 10 del paziente
- sideEffects (categorical): valutazione degli effetti collaterali su una scala di 5 classi
- effectiveness (categorical): valutazione dell'efficacia su una scala di 5 classi

I dati considerati, inizialmente suddivisi in training e test, vengono riportati ad un unico dataset. Successivamente si procede con l'eliminazione delle osservazioni duplicate, le quali risultano essere in totale 50. Inoltre, si eliminano le righe contenenti alcune osservazioni nulle: 1 nella colonna `condition`, 2 nella colonna `sideEffectsReview` e 8 nella colonna `commentsReview`. La variabile `rating` viene ricodificata in modo tale da passare da una scala 1-10 ad una scala in 3 classi nel seguente modo: i valori da 1 a 4 vengono codificati come "low", quelli da 5 a 8 come "medium" e infine quelli da 9 a 10 come "high". In questo modo, come si evince dalla Figura 1, la distribuzione della variabile risulta essere più simmetrica.
Anche per la colonna `sideEffects` viene effettuata una ricodifica del tipo:

1. No Side Effects → No Side Effects
2. Mild Side Effects → Mild/Moderate Side Effects
3. Moderate Side Effects → Mild/Moderate Side Effects
4. Severe Side Effects → Severe Side Effects
5. Extremely Severe Side Effects → Severe Side Effects

In questo modo la colonna `sideEffects` passa da avere 5 classi ad averne 3. Si riporta il grafico della distribuzione della variabile `sideEffects` a seguito della ricodifica in Figura 2.

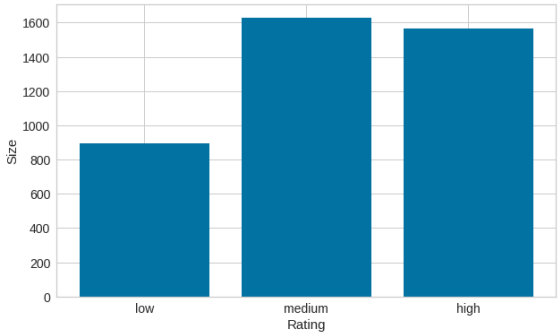


Figura 1: Distribuzione variabile rating a seguito della ricodifica

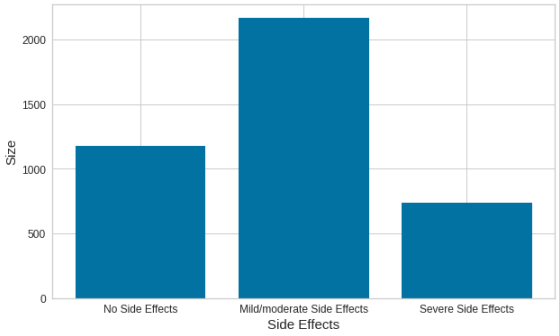


Figura 2: Distribuzione variabile sideEffects a seguito della ricodifica

# 3 Pre-processing

La variabile testuale che viene presa in considerazione per l'intera analisi è l'attributo `SideEffectsReview`. Prima di procedere con lo svolgimento dei tasks, è stato necessario effettuare delle operazioni di text processing al fine di uniformare ed estrarre parole dal testo. Le attività di processing effettuate, attraverso utilizzo di espressioni regolari, sono:

- Conversione di tutte le parole in caratteri minuscoli sfruttando la funzione `lower()`.
- Rimozione di spazi in eccesso, numeri e punteggiatura.
- Le contrazioni della lingua inglese vengono rimosse e sostituite con le rispettive scritture estese, come ad esempio *don't* viene sostituito con *do not*.
- Rimozione di URLs.
- Tokenization, ovvero suddivisione dei testi in differenti unità.
- Rimozione di *Stopwords* al fine di pulire il dataset da parole che non contribuiscono a definire il commento.
- *Stemming* e *Lemmatization* attraverso l'utilizzo della libreria `nltk`. Queste due operazioni sono tra loro esclusive. La *Lemmatization* riduce termini flessibili ad una forma base. Lo *Stemming* riduce invece le parole alle loro radici. Entrambe hanno il vantaggio di ridurre la dimensione del dizionario analizzato. La scelta di implementare una di queste due tecniche verrà ripresa nelle sezioni successive.

# 4 Text Representation

La text representation è la fase finale di preparazione dei dati che risulta esLsere cruciale per ottenere delle migliori performance dei vari modelli sviluppati in seguito. Al fine di scegliere la migliore rappresentazione dei commenti sono state testate tre differenti implementazioni:

1. **Binary**: a ciascun documento è associato un vettore binario in cui ogni elemento corrisponde alla presenza o meno della parola nel testo.

2. **Bag of Words**: ciascun documento è identificato attraverso un vettore in cui è presente il numero di occorrenze di ciascuna parola.

3. **Tf-Idf**: il numero di occorrenze di ciascuna parola, ovvero la *term frequency*, è pesato rispetto all'inverso della presenza della parola nel corpus, ovvero *l'inverse document frequency*, in modo da ottenere una maggior caratterizzazione del documento. Questa misura cresce sia con il numero di occorrenze all'interno di un documento sia con la rarità del termine nella collezione.

Tutte e tre le rappresentazioni sono state implementate considerando sia un dizionario composto esclusivamente dalle singole parole (uni-grammi) sia un dizionario composto anche dalle coppie di parole (bi-grammi). Queste ultime sono state poi messe da parte poiché prive di valore aggiunto nello svolgimento dei tasks. Come si approfondirà nella sezione successiva, tra le differenti rappresentazioni si è scelto di utilizzare la rappresentazione **binary** per il primo task di classificazione e la **Tf-Idf** per il task di clustering.

# 5 Text Classification

Avendo a disposizione differenti text representations, è necessario individuare quella più adatta al task di classificazione. Per tutte e 6 sono inizialmente state prese in considerazione un numero massimo di features pari alle 1500 più rilevanti, per evitare un sovraccarico computazionale. La valutazione della classificazione dell'attributo `sideEffect` per ciascun tipo di rappresentazione testuale, è stata effettuata tramite l'implementazione di quattro differenti modelli della libreria Scikit-Learn: *SVC, K-Nearest Neighbor, Random Forest e Multi-Layer Perceptron.*

- **SVC**: questi tentano di mappare i dati in uno spazio più ampio, con l'obiettivo di trovare l'iper-piano che meglio separi le osservazioni in base alla variabile di interesse. È stata integrata una Support Vector Machine (SVM) utilizzando il classificatore `LinearSVC`.

- **K-Nearest Neighbor**: è un algoritmo di riconoscimento dei pattern basato sulla vicinanza dei dati. Nella classificazione un oggetto è classificato da un voto di pluralità dei suoi vicini, con l'oggetto assegnato alla classe più comune tra i suoi k Neighbors più vicini.

- **Random Forest**: sono molto flessibili e di facile comprensione in quanto è possibile misurare l'effetto delle diverse variabili nella classificazione grazie ai coefficienti assegnati dal modello. Nell'analisi è stato implementato il classificatore `LogisticRegression`.

- **MLP**: sono modelli black-box che sfruttano una rete di neuroni artificiali per stimolare funzioni complesse. Tuttavia, proprio per la loro struttura, sono di difficile interpretazione. Tra questi è stato implementato il classificatore `MLP`.

Per identificare il migliore modello, in termini di accuracy, si procede con un approccio Cross-Validation con 5 folds.

| | **SVM** | **RF** | **KNN** | **MLP** |
|---|---|---|---|---|
| **Bin stem** | 0.69 | 0.73 | 0.59 | 0.71 |
| **Bin lemm** | 0.70 | 0.73 | 0.59 | 0.71 |
| **BoW stem** | 0.68 | 0.73 | 0.61 | 0.70 |
| **BoW lemm** | 0.69 | 0.73 | 0.61 | 0.70 |
| **Tf-Idf stem** | 0.72 | 0.73 | 0.56 | 0.69 |
| **Tf-Idf lemm** | 0.72 | 0.73 | 0.56 | 0.68 |

Tabella 1: Valori di accuracy medi sul Cross-Validation.

Sulla base dei risultati ottenuti dopo questa prima implementazione, pur essendo tra loro abbastanza simili, si è deciso di optare per la rappresentazione binary lemmatized in quanto più rappresentativa per recensioni molto brevi.

Riducendo l'analisi a quest'unica rappresentazione si è deciso di prendere in considerazione l'intero dizionario lemmatized composto da 7666 uni-grammi e di implementare però una riduzione della dimensionalità attraverso la combinazione di due differenti tecniche. Il primo step consiste in una fase di *feature selection* subito seguita da una *feature synthetization* che ha permesso di ridurre ulteriormente il numero delle variabili. Gli strumenti utilizzati per ridurre la dimensionalità sono:

- **Test Chi-quadro**: si utilizza per determinare la relazione tra le variabili indipendenti e la classe. Adoperando questo test per la *feature selection*, siamo in grado di selezionare le features che dipendono fortemente dalla risposta.
- **Principal Component Analysis (PCA)**: l'analisi delle componenti principali è una tecnica per la semplificazione dei dati utilizzata nell'ambito della statistica multivariata. Consiste in una trasformazione ortogonale per convertire un insieme di variabili, potenzialmente correlate, in un insieme di variabili linearmente non correlate, chiamate componenti principali.

Quindi, per prima cosa, è stata applicata una *feature selection* mantenendo solamente l'80% delle variabili migliori secondo il test Chi-quadro, la quale ha permesso di ridurre inizialmente il numero di variabili da 7666 a 6132. Successivamente è stata implementata una PCA in grado di abbassare la dimensionalità del dataset a 1000 features, tali da riuscire a spiegare comunque più del 90% della varianza totale.

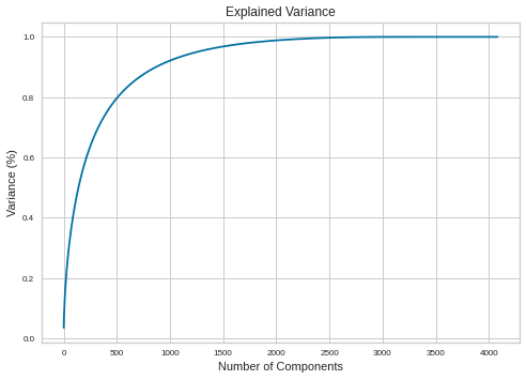


Figura 3: Varianza spiegata per numero di componenti PCA

Anche in questo caso sono stati nuovamente testati i modelli SVC, RF e MLP, escludendo il modello KNN che, come già visto, porta a risultati peggiori. Inoltre il dataset viene suddiviso in un set di training ed in uno di test seguendo una proporzione 90/10, ovvero il 90% del dataset viene utilizzato per la fase di addestramento del modello, mentre il restante 10% viene sfruttato per la fase di test. Con questo metodo si ottengono due subsets che risultano avere rispettivamente 3673 e 409 osservazioni.

| | **Accuracy CV** | **Accuracy Test** | **CV Execution Time** |
|---|---|---|---|
| **SVC** | 0.71 | 0.76 | 36 sec |
| **RF** | 0.68 | 0.67 | 44 sec |
| **MLP** | 0.71 | 0.74 | 84 sec |

Tabella 2: Metriche per il confronto dei modelli.

Si considera il modello SVC come il più adeguato per risolvere il problema di text classification, con un valore di accuracy sul test set di 0.76, è anche il più efficiente in termini di tempi computazionali. Si presenta in Figura 4 la matrice di confusione della classificazione del modello SVC per avere una rappresentazione più efficace dell'accuratezza della classificazione. Il modello performa meglio nella classicazione delle recensioni appartenenti alle classi mild/moderate side effects e severe side effects, rispetto a quelle che non riportano alcun effetto collaterale.

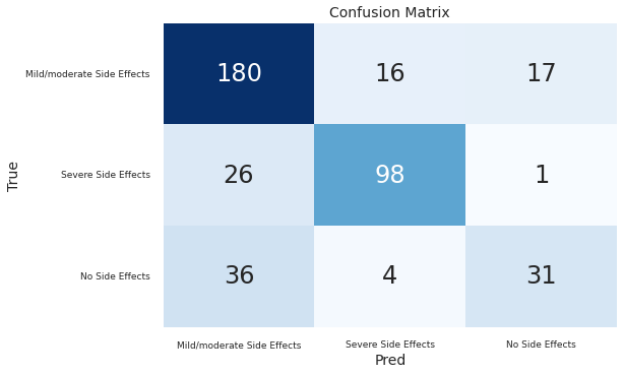


Figura 4: Matrice di confusione modello SVC sul test set.

# 6 Text Clustering

Oltre al task di text classification è stato sviluppato anche un task di text clustering. Si tratta di un metodo non supervisionato di raggruppamento di documenti in classi di oggetti simili tra loro. Le tecniche di clustering prese in considerazione sono:

- **K-Means**: appartiene agli algoritmi definiti di *flat clustering*. Crea un insieme piatto di clusters, ovvero senza alcun elemento strutturale esplicito che metta in relazione i gruppi tra loro. Tende a creare clusters di forma sferica.
- **Gerarchico agglomerativo**: a differenza del primo permette di creare una gerarchia di clusters seguendo un approccio bottom-up.
- **DBSCAN**: fa parte degli algoritmi di clustering *density-based*. Non richiede il numero di clusters da identificare come parametro iniziale bensì è in grado di trovare il giusto numero in modo autonomo ed identificare clusters con forme diverse.

Anche in questo task sono state testate diverse forme di text representation e si è deciso di utilizzare la Tf-Idf stemmed, a differenza di quanto fatto per il task di classificazione. Questa scelta è stata presa a seguito di una valutazione logica, data dall'interpretazione dei clusters ottenuti, che verranno illustrati in seguito.

Per prima cosa sono stati calcolati i valori di *silhouette* e distorsione sul k-means per differenti k, di cui si riportano le visualizzazioni in Figura 5.

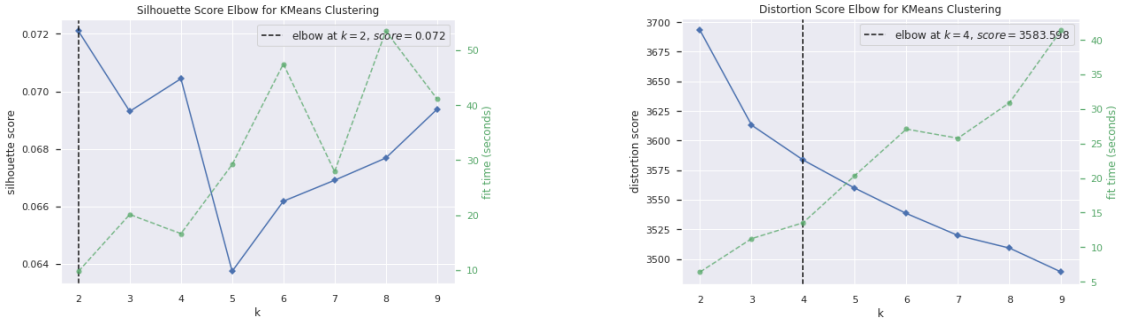


Figura 5: Grafici per silhouette (sx) e distorsione (dx), con k-means

Fatta questa prima selezione per il k-means, in cui il k ottimale risulta essere pari a 4, sono state sperimentate tutte e tre le tecniche di clustering presentate in precedenza. Per ciascuna di esse si è valutato il numero di clusters ideale e si è fatta una prima visualizzazione grafica per mezzo di t-SNE, un algoritmo di riduzione della dimensionalità largamente adoperato in ambito text mining. Il t-SNE ha permesso di rappresentare in 2 dimensioni i clusters ottenuti con i diversi modelli, i quali sono stati messi a paragone con le classi della variabile `sideEffects` utilizzata per il task di classificazione. Come si evince dalla Figura 6 sottostante, si può notare che alcuni clusters sembrano avvicinarsi alla separazione delle classi dell'attributo `sideEffects`.

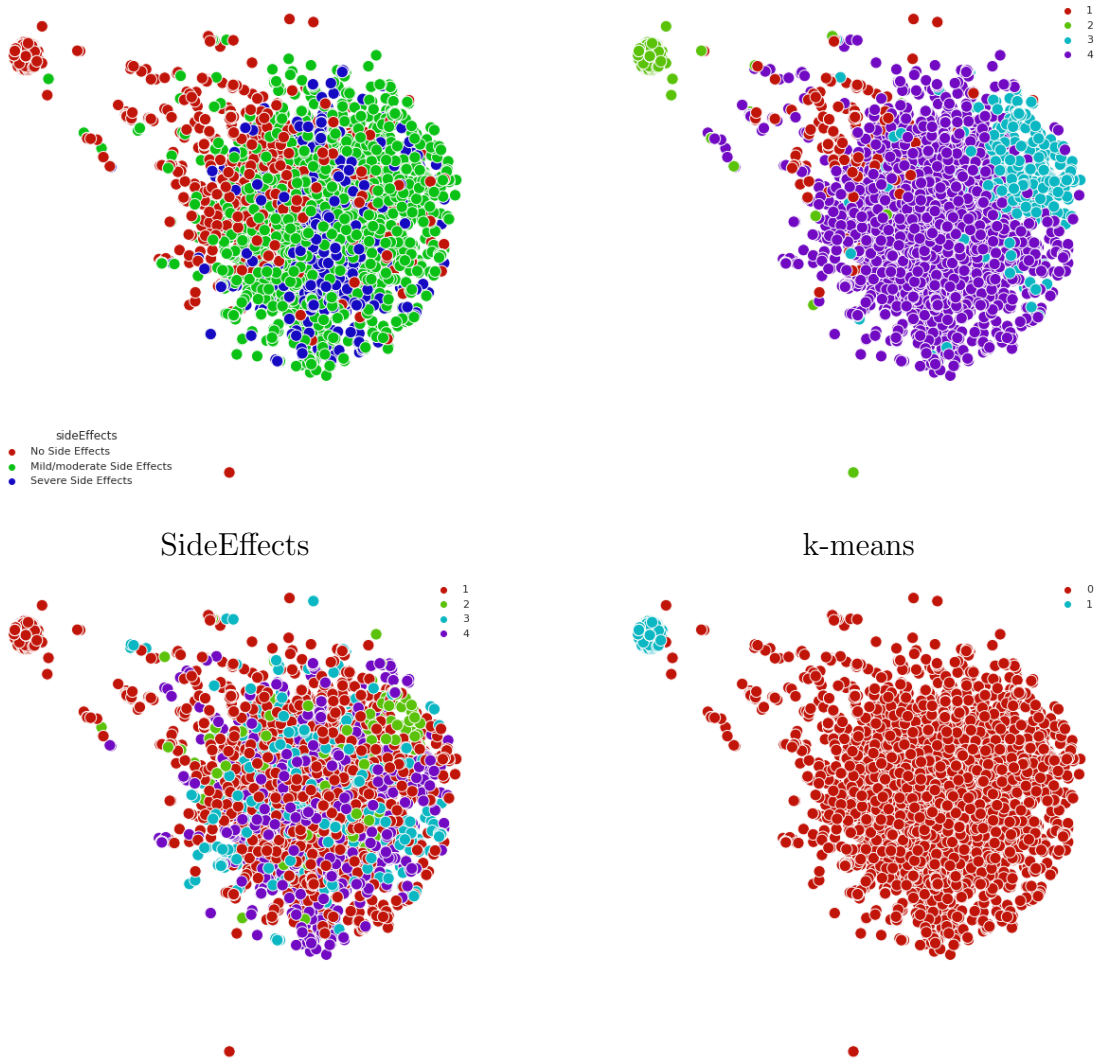


Figura 6: Visualizzazione bidimensionale dei clusters ottenuti

Successivamente si è deciso di visualizzare le parole più utilizzate per ciascun cluster e la tecnica che ha dato i risultati più apprezzabili è stata k-means. Dalla Figura 7, si evince che il secondo gruppo può essere associato senza alcun dubbio all'assenza di effetti collaterali, poiché la parola *None* spicca sulle altre. Il primo gruppo, con una scarsa eterogeneità di parole, può essere associato alla debole comparsa di effetti collaterali mentre i gruppi rimanenti, il terzo e il quarto, possono essere associati ad effetti collaterali evidenti e/o gravi. Il terzo sembra essere associato ad effetti collaterali di tipo cutaneo come secchezza e arrossamento della pelle, mentre il quarto, ad occhio, non lo si può ricondurre ad alcun tipo di effetto particolare. Si ipotizza anche che il numero di parole presenti all'interno delle *wordclouds* possa essere direttamente proporzionale alla manifestazione di più effetti collaterali.

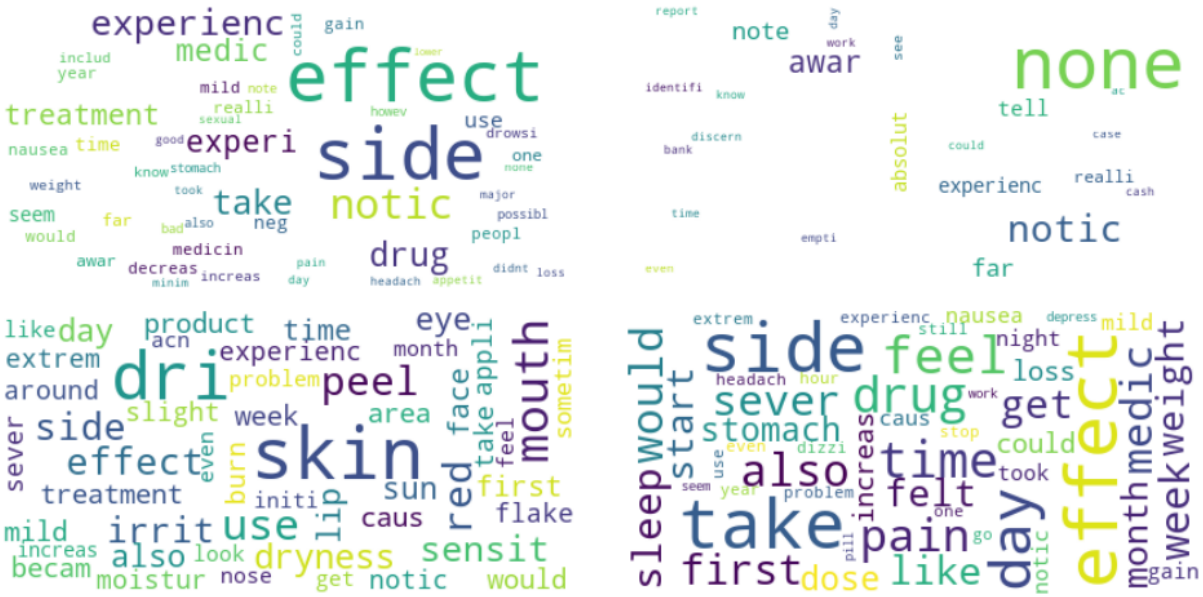


Figura 7: Wordcloud k-means, k=4

È stata poi svolta un'analisi incrociata tramite l'istogramma in pila, riportato in Figura 8, per verificare se fosse davvero presente una relazione tra i clusters appena trovati e la gravità degli effetti collaterali. Si è fatto ricorso anche alla metrica della *Normalized Mutual Information*, ottenendo un valore di 0.2. Essa può essere interpretata come la correlazione tra l'appartenenza ad un gruppo e la variabile risposta `sideEffects`. Tuttavia, questa ipotesi non sembra trovare piena conferma né dal grafico né dal valore di *Normalized Mutual Information* ottenuto.

Infine, si è cercato di capire se ci fosse qualche evidenza per quanto riguarda la relazione tra i medicinali, le loro recensioni, e i clusters individuati. Si sono quindi identificati i farmaci più recensiti e si è rappresentato tramite istogramma in pila la percentuale di appartenenza ai 4 diversi clusters (Figura 9). Le osservazioni sul cluster 3 riguardanti effetti collaterali di natura cutanea trovano conferma nel fatto che alcuni farmaci utilizzati per il trattamento di acne, tra cui Retin-a, Accutane e Sotret, abbiano una forte appartenenza al terzo cluster.

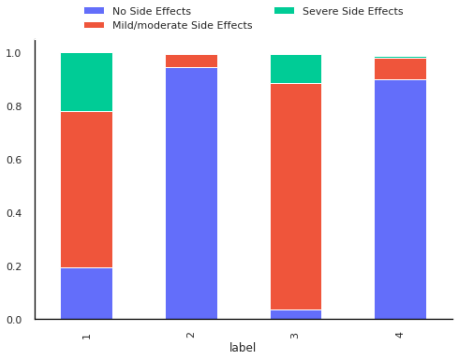


Figura 8: Istogramma in pila dei cluster sulla variabile SideEffects

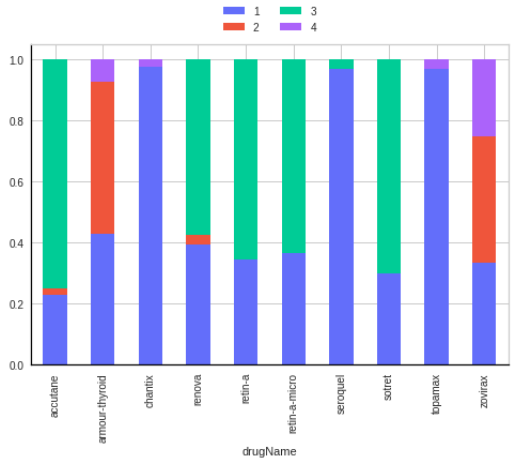


Figura 9: Istogramma in pila dei cluster sulle top-drugs

# 7 Conclusioni

Recuperando le domande di ricerca, dopo un'analisi dei dati, è possibile affermare che:

1. *È possibile classificare la gravità degli effetti collaterali di un farmaco basandosi sulle opinioni espresse dagli utenti online?*: Nel task di classificazione si può dire di aver ottenuto dei risultati abbastanza soddisfacenti. La rappresentazione del testo che mediamente ottiene il valore più elevato in termini di accuracy è la rappresentazione Binary lemmatizzata, che pare quindi più adatta a questo tipo di problema. È ragionevole pensare però che prevedere una variabile multiclasse possa essere complicato in quanto la classe viene assegnata sulla base di una scala soggettiva. Nel caso in cui la variabile target fosse stata binomiale, probabilmente la classificazione sarebbe stata più precisa.

2. *È possibile dividere in gruppi i farmaci in base agli effetti collaterali che gli utenti riportano nelle loro recensioni?*: I metodi gerarchici e density-based non si sono rivelati adatti ad affrontare questo tipo di problema, a differenza di quanto accaduto per il metodo delle k-means, che ha permesso di identificare gruppi discretamente significativi in un'ottica di interpretabilità.