Università degli studi di Milano-Bicocca

Text Mining and Search

Good or Bad? Ask Your Pet: Classificazione delle recensioni di "Amazon Review Dataset"

Authors: Alessandro Borroni[1], Andrea Corvaglia[1], Mirko Giugliano[1]*

Sommario
In questo lavoro vengono proposti una serie di modelli con il fine di classificare le recensioni di Amazon, più precisamente relative alla categoria "Prodotti per animali domestici" (dall'inglese Pet supplies), cercando di carpire il grado di soddisfazione degli utenti, il quale può essere positivo, in caso di alta soddisfazione, o negativo, in caso contrario. Tale scelta è giustificata dal fatto che le recensioni hanno una grande influenza sul comportamento d'acquisto dei consumatori. Le prime fasi si sono sviluppate attorno alla pulizia dei dati e al preprocessing dei testi analizzati. Al fine di testare i modelli si è optato per una suddivisione in training e validation set, essendo l'approccio di tipo supervisionato. In totale sono stati testati sei modelli di classificazione e si deciso di dar vita a un task non-supervisionato mediante una Sentiment Analysis (Opinion Mining), per vedere se la polarità delle recensioni corrispondesse al voto fornito tramite le stelle.

Keywords
TextMining — BinaryClassification — SentimentAnalysis — AmazonReview — Pet — DeepLearning — Reviews — Words — Python — OpinionMining

[1] *Data Science M.Sc., Department of Computer Science, Systems and Communication, University of Milano-Bicocca, Milan, Italy*
*__Corresponding author__: m.giugliano@campus.unimib.it

Indice

1 Introduzione **2**
1.1 Obiettivo 2
2 Dataset **2**
3 Approccio metodologico **3**
3.1 Pre-processing del testo 3
3.2 Rappresentazione del testo 4
3.3 Classificatori e Opinion Mining 5
4 Risultati e Conclusioni **6**
4.1 Risultati 6
4.2 Conclusioni 6
4.3 Sviluppi futuri 6
Riferimenti bibliografici **6**

1. Introduzione

La nascita di Amazon, una delle più grandi piattaforme di E-Commerce al mondo, ha rivoluzionato radicalmente le abitudini di consumo degli utenti. Nei giorni d'oggi, effettuare un acquisto non richiede più la presenza fisica in-store dei consumatori, bensì tale attività può essere svolta comodamente da casa, spesso a prezzi vantaggiosi, con una maggiore possibilità di scelta e confronto. Le tipologie di prodotti acquistabili sul sito sono innumerevoli: da utensili per la cucina, fino a strumenti di tipo ludico (videogiochi, film, ...), soddisfando praticamente tutte le fasce d'età e la maggior parte dei fabbisogni.

In questo sistema, di per sè molto solido, un contributo fondamentale è dato dagli stessi utenti. La strategia di Jeffrey Preston Bezos, fondatore e amministratore delegato di Amazon, consiste nell'instaurare un rapporto di fiducia con i clienti, permettendo loro di interagire e di basarsi sull'opinione altrui per lasciasri guidare nei loro acquisti. Differentemente dai negozi fisici, i consigli, in questo caso sotto forma di recensioni, sono svincolati da qualsiasi conflitto di interesse (escludendo i recenti casi di fake reviews). Il sistema è semplice: dopo ogni acquisto vi è la possibilità di lasciare una recensione pubblica, con tanto di valutazione (da 1 a 5 stelle), la quale esprime il proprio grado di soddisfazione.

1.1 Obiettivo

L'obiettivo di questo progetto è utilizzare il dataset Amazon Reviews, nello specifico analizzando la categoria Pets Supplies, al fine di formare un classificatore che sia in grado di eseguire un task di classificazione binaria. La classificazione del testo è il processo di assegnazione di tag o categorie al testo in base al suo contenuto. È uno dei compiti fondamentali nell'elaborazione del linguaggio naturale (NLP) con ampie applicazioni come l'analisi dei sentimenti, l'etichettatura degli argomenti, il rilevamento dello spam e il rilevamento degli intenti. Nel caso in esame, si intendono prevedere le due classi relative alla votazione degli utenti su un determinato articolo (rating "positivo" e "negativo"), e saranno mostrate più nel dettaglio nelle sezioni successive.

Tale task permetterebbe di comprendere se vi sia una correlazione diretta tra il voto fornito da ogni utente e la composizione testuale del commento lasciato sotto ogni articolo. Laddove vi fosse, sarebbe possibile evincere automaticamente la polarità della recensione senza considerare la valutazione.

2. Dataset

Amazon Reviews Dataset [1] è una vasta raccolta di recensioni provenienti dagli utenti per i prodotti venduti su Amazon. Il dataset include 233,1 milioni di recensioni per prodotti di diverse categorie, come Fashion, Books e Digital Music. Per questo progetto si è scelto di adoperare il sottoinsieme relativo alla categoria "Pets Supplies", come specificato precedentemente, per assolvere i task richiesti. Tale categoria raccoglie utensili volti alla cura e al benessere del proprio animale domestico: qui si possono trovare giochi, strutture per il comfort, cibo e tutta una serie di articoli annessi.

Le features di interesse in questo dataset sono "reviewText", il corpus testuale relativo alla recensione dell'utente, e "overall", e cioè il punteggio associato alla revisione, il quale è un valore intero compreso tra 1 e 5 stelle (rispettivamente "per niente soddisfatto" e "molto soddisfatto"). Dopo aver rimosso i valori mancanti dalla colonna reviewText e aver eliminato le recensioni non verificate, la dimensionalità del dataset è stata ridotta. In seguito a questa operazione, è stato possibile notare uno sbilanciamento evidente tra le classi.

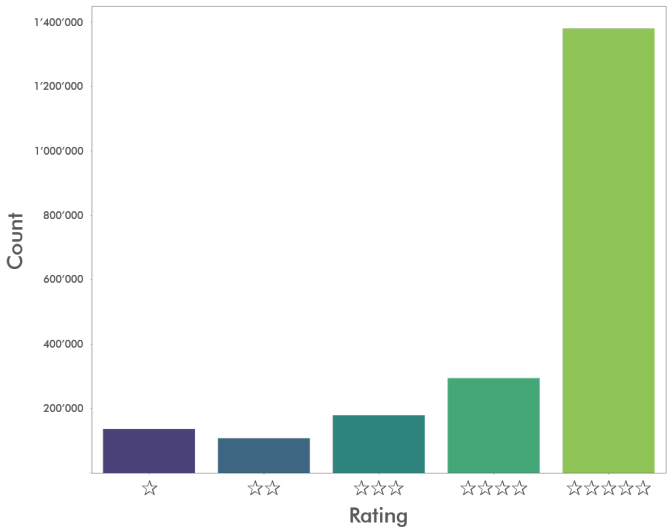

Figura 1. Bar chart rappresentante le classi disponibili in partenza, sbilanciate.

Ai fini di questo progetto, si è considerata che una recensione con punteggio strettamente superiore a 3 stelle abbia una polarità positiva, mentre tale sentimento risulta negativo nel caso in cui le stelle siano strettamente inferiori a questo valore soglia. Per ottenere un dataset bilanciato tenendo conto della condizione appena esplicitata, si è deciso di considerare 50'000 istanze per ogni classe di interesse. Questa scelta può essere contestata osservando che il sistema di filtraggio del sito di Amazon permette di scegliere tra "Solo recensioni positive", 4 e 5 stelle, e "Solo recensioni critiche", da 1 a 3 stelle. Tale convenzione è stata leggermente manipolata nel tentativo di considerare le recensioni veramente critiche, si è infatti optato per l'eliminazione di tutte le recensioni con 3 stelle, le quali sono state considerate più come neutrali. In questo modo si è ottenuto un dataset bilanciato, il quale presenta 100'000 istanze di rating positivo e 100'000 istanze di rating negativo, per un totale di 200'000 recensioni.

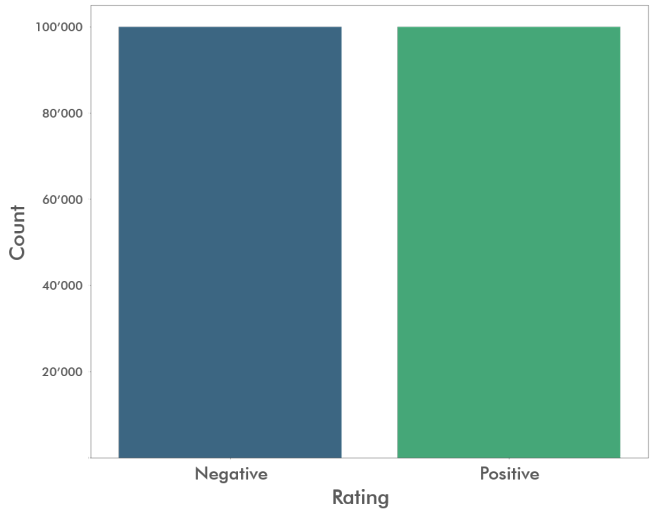

Figura 2. Bar chart rappresentante le classi finali (bilanciate).

3. Approccio metodologico

3.1 Pre-processing del testo

La pre-elaborazione del testo è tradizionalmente un passaggio importante per le attività di elaborazione del linguaggio naturale (NLP). Trasforma il testo in una forma più digeribile in modo che gli algoritmi di Machine Learning possano funzionare meglio.

Le recensioni sono state manipolate al fine di ottenere una migliore rappresentazione ed essere comprensibili per le macchine. In prima istanza si sono effettuate analisi esplorative al fine di evidenziare la distribuzione delle lunghezze delle review e delle parole più utilizzate, in forma di sostegno, sono state prodotte alcune visualizzazioni (quali bar chart e wordcloud) al fine di ottenere una più facile e immediata comprensione (Fig. 3).

Figura 3. Word cloud rappresentante le parole a più elevata frequenza provenienti da tutte le recensioni considerate (da 1 a 5 stelle, esclusa la classe non considerata).

Il primo gruppo di operazioni si è incentrato sulla pulizia delle recensioni, attraverso una serie di operazioni che verranno poi eplicitate, nonché su una fase di tokenizzazione, normalizzazione, POS tagging, lemmatizzazione e rimozione delle cosiddette "stop words". Questa prima parte è essenziale per conciliare alcune differenze nella rappresentazione del testo. Qui di seguito verranno esplicitate nel dettaglio le diverse operazioni prese in considerazione:

1. *Tokenizzazione*: Questo passaggio è necessario per suddividere ogni recensione, che viene archiviata come stringa, in più token costituiti da singole parole e caratteri di punteggiatura. Questa operazione è stata implementata usando la funzione word_tokenize all'interno della libreria nltk.

2. *Minuscole e maiuscole*: trasformare in minuscola (lower case) ogni lettera delle parole analizzate, altrimenti un computer tratterà due parole identiche come diverse solo perché presentano una stessa lettera nelle due versioni, minuscola e maiuscola (ex.: "pappagallo" diverso da "Pappagallo");

3. *Slang e abbreviazioni*: nelle recensioni analizzate sono presenti slang e abbreviazioni. Un sottoinsieme di essi è stato sostituito con il loro equivalente comune (ad es. "you" anziché "ya");

4. *Forma contratta*: le parole che presentano una forma contratta (ad es. "won't") vengono espanse nel loro modulo "standard" (ad es. "will not"). Questo viene fatto perché la parola chiave non ha un significato utile e le rappresentazioni come bi-grams possono beneficiare della sua presenza;

5. *Caratteri speciali e punteggiatura*: questi caratteri vengono rimossi, poiché non sono utili a carpire il grado di soddisfazione di una determinata recensione. Questa operazione aiuta anche a ridurre la dimensionalità del dataset, portando a un aumento dell'efficienza. I caratteri che potrebbero influire sulla tokenizzazione non vengono eliminati in questo passaggio.

6. *Lemmatizzazione e POS*: al fine di rimuovere il suffisso dalle parole e ricondurle alla loro forma base, si effettua la lemmatizzazione: in questo modo le forme singolari e plurali della stessa parola saranno sostituite con lo stesso termine. Le implementazioni degli algoritmi utilizzati in questa fase sono della libreria nltk di Python. Per rendere più efficace la lemmazizzazione si è effettuato part of speech tagging, per evitare che la libreria commetta errori sulla categoria lessicale e trasformi parole che non ne necessitano (es. "was" deve rimanere "was" e non diventare "wa").

7. *Stop words*: il passo successivo ha comportato la rimozione delle stop-words. Il vantaggio di questo compito è di ridurre ampiamente la dimensionalità del dataset, pur mantenendo intatta la sua informatività: parole come articoli e pronomi sono davvero frequenti nelle frasi e la loro rimozione contribuisce notevolmente a ridurre la dimensionalità del dataset senza alterarne il significato.

Una volta che il corpus delle recensioni è stato correttamente pre-elaborato, sono state applicate alcune tecniche di rappresentazione del testo (text representation).

3.2 Rappresentazione del testo

La rappresentazione del testo è uno dei problemi fondamentali nel mondo della Text Mining e dell'Information Retrieval (IR). Mira a rappresentare numericamente i documenti di testo non strutturati per renderli adatti al lavoro degli algoritmi.

La prima rappresentazione scelta è stata la bag-of-word. A causa dell'eterogeneità del corpus di recensioni e della sua elevata dimensionalità, il numero di parole uniche è davvero elevato; si è ritenuto pertanto essenziale identificare caratteristiche che aiutassero a discriminare le diverse recensioni. A questo scopo si è scelto di usare la TF-IDF, perchè tiene conto dell'informazione legata alla frequenza con cui compaiono i termini. quuesta frequenza è normalizzata rispetto alla lunghezza variabile delle recensioni e permette inoltre di dare meno peso alle parole più comuni nel corpus, le quali non hanno un buon potere di discriminazione. In particolare la funzione utilizzata per il calcolo della matrice permette di eliminare le parole troppo ricorrenti e troppo rare, ovvero quelle che si trovano ai lati della curva di Zipf's, fissando il cut-off inferiore e superiore. La matrice TF-IDF è stata costruita utilizzando solo il training set sulle recensioni: in un caso d'uso di produzione, è certo che a un certo punto emergeranno nuove parole, quindi questo scenario è stato simulato trovando le caratteristiche più generali associate a sentimenti positivi e negativi. Sia unigrammi che bigrammi sono stati presi in considerazione nella costruzione della matrice TF-IDF. La rappresentazione di bigram è stata scelta al fine di consentire ai modelli predittivi di avere una più ampia comprensione del significato delle recensioni: in questo modo parole come "non + aggettivo" saranno viste come caratteristiche uniche, il cui potere espressivo è maggiore di quello del due parole singole.

La matrice ottenuta è piuttosto grande e sparsa, di conseguenza si sono esplorati alcuni approcci per ottenere un embedding del corpus per facilitare la fase di classificazione. Avendo una rappresentazione numerica dei dati testuali grazie alla matrice TF-IDF è stato possibile adoperare la *Principal Component Analysis* in modo da ridurre la sparsità della rappresentazione in favore di un'altra rappresentazione con meno features, ma più significative. Nonostante queste nuove features non siano interpretabili, ma soltanto virtuali, sono ottime per una modellazione matematica e quindi per essere adoperate dagli algoritmi confrontati per la classificazione. Si è preferito scegliere la quantità di varianza da conservare piuttosto che il numero di componenti, ponendo la prima al 95% , ed assicurandosi così che la maggior parte dell'informazione venisse conservata. Il risultato è una matrice densa con una dimensione minore, che quindi velocizza il training dei modelli.

Si è anche esplorata come alternativa alla PCA

una rappresentazione tramite *Latent Dirichlet Allocation*, che solitamente è adoperata nell'ambito della *Topic Classification*, in quanto tenta di rappresentare un documento come un insieme di argomenti, ognuno dei quali è caratterizzato da una particolare distribuzione di termini. Si è ritenuto adatto questo approccio al caso trattato nel progetto in quanto si vanno a considerare dei topic fittizi, che nonostante siano in sè privi di significato, permettono di sintetizzare i documenti sulla base degli argomenti trattati, grazie alla distribuzione di probabilità sulle parole fornita dai topic. In conclusione, quindi, l'idea è quella di utilizzare topic fittizi per rappresentare le idee contenute nei documenti, in modo da ridurne le dimensioni e avere una matrice che invece delle componenti della PCA, ha per colonne le probabilità di appartenenza a ciascun "topic" del documento, il che risulta in una rappresentazione densa del corpus. Putroppo questo metodo, oltre ad essere piuttosto lento nel training, porta a risultati significativamente peggiori rispetto alla PCA, che ,dunque, è stata scelta come rappresentazione vincente per testare i vari classificatori nella sezione successiva.

3.3 Classificatori e Opinion Mining

Come specificato precedentemente, l'obiettivo del progetto è quello di costruire uno strumento predittivo che abbia il fine di associare una recensione alla sua etichetta più appropriata, che in questo caso emerge sotto forma di votazione in termini di stelle (o rating). Sono stati presi in esame diversi algoritmi, i quali a loro volta sono stati confrontati con il fine di valutarne l'efficienza e l'efficacia. Prima di affrontare la fase di implementazione vera e propria, si è stabilita una soglia di splitting tra Training e Test pari a 85%-15% al fine di valutare i modelli su un campione sufficientemente numeroso in test da un lato, e trainare le reti neurali su abbastanza esempi dall'altro.

Più precisamente, sono stati implementati i seguenti modelli:

- Support Vector Classifier (SVC);
- Random Forest (RF);
- K-Nearest Neighbors (KNN);
- Multinomial Naïve-Bayes (MNB);
- Fully Connected Neural Network (recensioni);
- Multi-input Fully Connected Neural Network (recensioni e sentiment);

Inoltre, al fine di tentare un approccio non supervisionato, si è deciso di implementare una Sentiment Analysis. Al fine di adempiere a questo task è stato utilizzato il dizionario Bing Liu, uno dei più adottati in materia di Opinion Mining [2]. L'idea è stata quella di provare a comprendere se dalla polarità delle recensioni potesse essere possibile discriminare le recensioni positive da quelle negative, associandole ai rispettivi score. Così la Multi-input Fully Connected Neural Network ha ricevuto in input sia le feature dalla PCA che gli score normalizzati della Sentiment Analysis, per provare ad implementare un modello che tenesse conto di tutta la informatività estratta. In particolare il modello appena citato consiste in una rete con due rami che si congiungono per dare un unico input. Il ramo che prende in input i risultati della Sentiment Analysis è molto più corto rispetto al ramo principale, così da non disperdere l'informazione associata, che, essendo già il risultato di una elaborazione, non necessita di ulteriori trasformazioni.

Va specificata anche la modalità di implementazione della SVC, per la quale si è deciso di usare un meta-classificatore ad *ensamble*, ovvero un modello che fa un fit di una serie di classificatori base, in questo caso SVC, su sotto-insiemi del dataset e restituisce una previsione unica basata su tutte le previsioni fornite dai singoli classificatori. In pratica seguendo il funzionamento del Random Forest. In questo modo è stato possibile eseguire il training in parallelo riducendo i tempi di computazione. Questa modalità di implementazione ha inoltre portato ad un miglioramento delle performance.

L'ultima precisazione va fatta sul Multinomial Naïve-Bayes (MNB), il quale è molto adatto alla classificazione con dataset di word count, anche sotto forma di conteggi frazionali come nel caso del tf-idf. La precisazione consiste nel fatto che questo classificatore non va usato con la PCA, perchè MNB applica il teorema di Bayes con l'assunzione di indipendenza tra le features ed inoltre non supporta eventuali valori negativi. Per questa ragione il modello è stato trainato sulla matrice TF-IDF senza alcun tipo di embedding, in ogni caso questo classificatore richiede un costo computazionale molto minore, permettendo di concludere il training in pochi secondi nonostante l'uso di una matrice così estesa.

4. Risultati e Conclusioni

4.1 Risultati

Model	Accuracy
KNN	0.62
Mi NN	**0.85**
MNB	0.81
NN	0.83
RF	0.78
Sentiment	0.68
SVC	0.82

Tabella 1. Tabella con i risultati di accuracy sul test set dei modelli di classificazione confrontati.

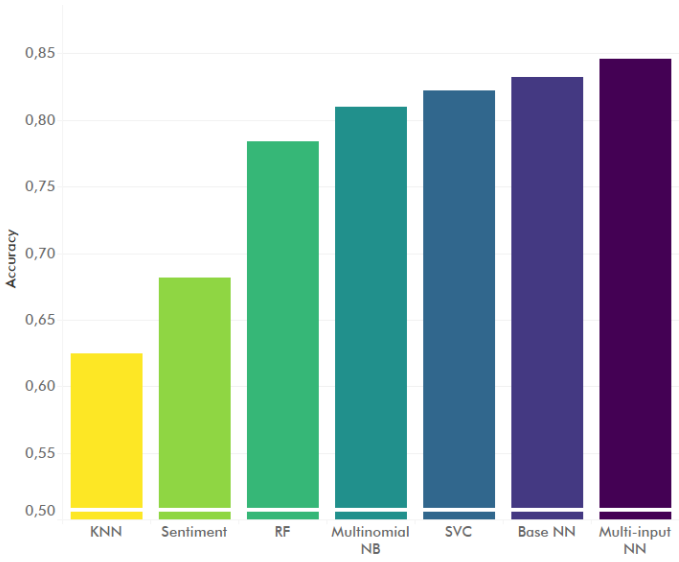

Figura 4. Grafico che confronta i valori di Accuracy ottenuti con i diversi modelli implementati.

4.2 Conclusioni

Come è possibile vedere dalla Tabella 1 e dalla Figura 4, il risultato migliore viene ottenuto con il modello Multi-input Fully Connected Neural Network, il quale però non ottiene risultati nettamente migliori rispetto alla Fully Connected Neural Network, pur sfruttando i risultati della Sentiment Analysis (oltre all'informazione fornita dalle sole recensioni).

Questo può essere spiegato dal fatto che la Sentiment non discrimini in maniera efficace. Pur utilizzando il dizionario Bing Liu, trainato ed estratto sulle stesse Amazon review, lo sbilanciamento delle parole verso il positivo non consente di comprendere a pieno la polarità delle recensioni, che nel nostro specifico caso impediscono alla rete multi input un significativo improvement rispetto alla rete baseline.

Gli altri classificatori ottengono risultati minori. Inoltre, dal momento che il training delle reti neurali non risulta eccessivamente lungo e dispendioso, le performance scarse dei modelli di stampo classico ci spingono a preferire senza dubbio le reti neurali per questo task.

4.3 Sviluppi futuri

Tra gli sviluppi futuri, si è primariamente pensato a una maggiore considerazione delle particelle pragmatiche (anche conosciute con il nome di emoji), che in ambito Sentiment Analysis e Irony Detenction apporterebbero un significativo contributo. L'ironia in questo tipo di testi è molto presente, soprattutto nelle review negative, quindi riuscire a coglierla sarebbe importante anche se decisamente impegnativo.

Inoltre si è pensato che un training più intensivo, considerando review anche di altri ambiti, non solo relative alla categoria Pet Supplies, potrebbe evitare il peggioramento dei risultati nel test set, poichè, avendo una matrice di TF-IDF più corposa, si diminuirebbero le probabilità di avere features mai viste nel test set.

Riferimenti bibliografici

[1] Jianmo Ni, Jiacheng Li, Julian McAuley (2019) *Empirical Methods in Natural Language Processing (EMNLP)*

[2] Minqing Hu, Bing Liu (2004) *Mining and summarizing customer reviews*, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery Data Mining, Seattle, Washington, USA