Università degli studi di
Milano-Bicocca

Text Mining and Search

Final Project

Amazon Reviews Classification

Authors:
Giulia Chiaretti - 800928 - g.chiaretti@campus.unimib.it
Federica Fiorentini - 807124 - f.fiorentini1@campus.unimib.it
Riccardo Maganza - 808053 - r.maganza@campus.unimib.it

Abstract

Utilizzare un sistema di reviews efficace potrebbe essere uno dei punti di forza di un sito e-commerce poiché permette agli utenti di avere un valor aggiunto rispetto ad un prodotto, oltre alla descrizione basica fornita dal sito. Può risultare importante, quindi, data una recensione di un cliente collegarla alla categoria di appartenenza del prodotto recensito. A tal proposito, è stato sviluppato un task di classificazione tramite l'utilizzo di modelli di Machine Learning che, a partire da un dataset di recensioni effettuate dagli utenti su Amazon appartenenti a 5 categorie, permette di identificarne l'area di appartenenza. Segue, quindi, la descrizione dell'analisi effettuata per svolgere il task, a partire dalla descrizione del dataset utilizzato, del pre-processing effettuato e dell'approccio metodologico utilizzato.

1 Introduction

Amazon.com è la più grande Internet company al mondo e, fondata nel 1994, è stata tra le prime grandi imprese a vendere merci su internet ed a rivoluzionare il mondo del commercio. Certamente uno degli argomenti più discussi negli ultimi anni è la crisi del commercio tradizionale e l'avanzata di nuovi modelli di business e troppo spesso si sente parlare della fine del commercio al dettaglio. Molte persone, però, si affidano ancora ai negozi per il semplice motivo che in questo modo possono vedere e giudicare fisicamente l'oggetto da acquistare. A tal proposito, Amazon si è basato su un sistema di collaborazione da parte degli utenti che, dopo ogni acquisto, possono lasciare una review in base all'esperienza condotta in modo che gli altri utenti possano ricevere una quantità di informazioni maggiore riguardo il prodotto.

L'obiettivo di questo progetto, quindi, è un task di *classification* per assegnare alle diverse review lasciate dagli utenti su Amazon la corretta categoria di appartenenza del prodotto di riferimento.

2 Datasets

Per svolgere il task è stato utilizzato un dataset, presente al link `http://jmcauley.ucsd.edu/data/amazon/` contenente diverse reviews in lingua inglese riguardanti le principali categorie di prodotti presenti su Amazon.

In questo caso, a causa di problemi computazionali, sono state utilizzate solamente 5 categorie di prodotti:

- Baby (160.792 reviews)
- Beauty (198.502 reviews)
- Grocery and gourmet food (151.254 reviews)
- Pet supplies (157.836 reviews)
- Sports and outdoors (296.337 reviews)

Per procedere ad uno sviluppo corretto dei modelli di classificazione, si è deciso di ricampionare adeguatamente il dataset, in quanto i dati a disposizione erano caratterizzati da uno sbilancialmento tra le diverse classi. Si è deciso, quindi, di considerare solamente un sample randomico di numerosità pari a 500.000 di reviews, 100.000 per ogni categoria considerata.

Il dati vengono messi a disposizione in formato *json* e, per questo motivo, è stata definita una funzione che permettesse di leggere i dati e trasformarli sottoforma di dataframe pandas. Inoltre, dopo aver importato i file riferiti ad ognuna delle categorie considerate, i diversi dataframe sono stati uniti in un unico dataset specificando la variabile *Category* che identifica la categoria di appartenenza della review.

Per ogni review, le variabili disponibili inizialmente sono *reviewesID, asin, reviewerTime, helpful, reviewText, overall, summary, unixReviewTime, reviewTime* e la variabile aggiunta *Category*. Ai fini del progetto, però, non erano necessarie tutte le informazioni e, a tal proposito, sono state conservate solamente le variabili ***reviewText e Category***.

3 Approccio metodologico

Per poter fare un'analisi ben precisa e approfondita, il testo dev'essere scritto in modo tale che il computer lo può facilmente leggere e analizzare. I documenti possono essere scritti in diversi modi, che possono essere a formato libero, a formato semi-strutturato, etc.

Per poter procedere con il task di classificazione delle reviews, quindi, è necessario effettuare un processo di modifica del testo in oggetto composto da diverse fasi.

3.1 Text processing

Il dataset creato nel punto precedente è stato sottoposto ad preprocessing del testo in modo tale da renderlo più interpretabile per i modelli, che consiste nel:

- **Tokenization**: In questa fase del processo, il documento viene suddiviso in unità più piccole, ognuna delle quali rappresenta un concetto. Ogni token, quindi, è una sequenza di caratteri con un significato semantico ben preciso. In questo caso i token sono costituiti tutti da parole singole;
- **Normalizzazione del testo**: Questa fase consiste nel mappare i termini tutti nella stessa forma, ovvero sequenze di caratteri con lo stesso significato devono essere scritte nello stesso modo. In particolare, sono state effettuate le seguenti attività:
  - Rimozione della punteggiatura presente nel testo;
  - Trasformazione del testo in caratteri tutti minuscoli.
- **Stop words removal**: le stop words sono le parole che compaiono molto frequentemente in un documento senza avere un significato sensato. Generalmente, quindi, vengono eliminate dal testo poiché ricoprono un ruolo poco importante nell'analisi. Questo processo è stato effettuato tramite la libreria *nltk - natural language toolkit* che mette a disposizione un dizionario di stopword inglesi;
- **Lemmatization**: questa tecnica consiste nel trasformare le parole nel loro lemma originario con l'obiettivo di eliminare le declinazioni delle varie parole che sono presenti nel testo ma tenendo sempre conto del contesto in cui si trovano i termini. Un'alternativa alla lemmatization era lo *stemming* ma, dopo aver effettuato diversi tentativi, i risultati raggiunti da quest'ultimo non portavano miglioramenti rispetto alla lemmatization e richiedevano sforzi computazionali maggiori.

3.2 Text representation e feature selection

Una volta ottenuto le reviews a cui è stato applicato tutto il processo di pre-processing, è stata costruita una matrice che rappresenti il testo avente

dimensioni pari al numero di osservazioni in riga e 10.000 feature in colonna, il tutto sottoforma di unigram. Ogni cella della matrice assume i cosiddetti pesi **tf-idf**, calcolati come il prodotto tra la frequenza del termine e la frequenza inversa del documento:. In questo modo, nella rappresentazione del testo si considerano due fattori, ovvero l'importanza della singola parola nel documento e l'importanza del documento all'interno dell'intera collezione. In particolare, il peso riferito alla parola aumenta all'aumentare della presenza del termine nel documento e all'aumentar edella rarità del termine nella collezione.

Di prassi è necessario diminuire il numero di feature presenti nella rappresentazione del testo per evitare il problema della *course of dimensionality*, secondo il quale un numero troppo elevato di feature non porta benefici all'analisi. A tal proposito, si è cercato di eliminare i cosiddetti termini sparsi, ovvero i termini con una frequenza inferiore allo 0.2 ma il numero di feature si riduceva in maniera esagerata. La soluzione utilizzata è stata di considerare solamente i primi 10.000 termini ordinati secondo la loro frequenza.

Una volta creata la matrice tf-idf di dimensioni 500.000x10000, si può procedere con il task di classificazione.

3.3 Classification task

Come primo step, il dataset è stato suddiviso in train e test set con una proporzione rispettivamente di 0.9 e 0.1. Per svolgere il task di classificazione sono stati implementati due algoritmi differenti:

- **Neural network**: è stato scelto di sviluppare una rete neurale Feed-Forward Fully Connected. In questo particolare tipo di rete, ogni layer necessita di un input, che coincide con l'output del layer precedente, e di una funzione di attivazione. Il modello sviluppato consiste in una rete avente 4 layer (composti da 1024, 512, 128 e 32 neuroni) per cui è stata utilizzata come funzione di attivazione la *relu* in quanto risultavano migliori le performance, eccetto nell'ultimo in cui è stata utilizzata la *softmax* poiché, solitamente, la relu viene utilizzata come funzione di attivazione nei layer intermedi mentre la softmax nei layer esterni, sempre nei problemi di classificazione. Ad ogni layer fully-connected è stato aggiunto un layer di Dropout per evitare il problema dell'overfitting poiché, inizialmente, la rete è strata trainata senza questo tipo di layer e dava problemi di overfitting sul validation set. In-

oltre, per l'ottimizzazione del modello, in cui si minimizza la funzione di loss, viene utilizzato Adam, un algoritmo di ottimizzazione basato sul gradienti del prim'ordine e su stime adattive dei momenti di ordine inferiore. Questo tipo di ottimizatore può essere utilizzato e può dare prestazioni buone in caso di problemi non stazionari con gradienti rumorosi o sparsi. Per quanto riguarda la loss function è stata utilizzata la categorical crossentropy, essendo appunto in presenza di un problema di classificazione multiclasse in cui solamente un risultato può essere corretto. Questa funzione di loss confronta la distribuzione dei valori previsti del modello con la distribuzione dei valori originali. Richiede, però, che i valori siano espressi tramite un one-hot encoding, L'obiettivo è quello di minimizzare la funzione di perdita durante la stima dei parametri W e b. Il modello è stato allenato sul training set utilizzando dei mini batch di dimensione 128, iterando per 50 epoche. E' stato implementato, inoltre, un criterio di EarlyStopping, secondo il quale il training del modello si arresta nel momento in cui non ottiene performance migliori in termini di loss calcolata sul validation set dopo 3 epoche di allenamento. Il modello è stato valutato sul validation set con l'obiettivo di misurare le performance tramite l'accuracy e la loss.

- **Random Forest**: questo modello fa parte dei cosiddetti *modelli euristici*, modelli che spesso sono in grado di ottenere approssimazioni ragionevoli senza richiedere sforzi computazionali eccessivi o ipotesi restrittive sui dati di input. Il random forest, in particolare, genera un certo numero di alberi decisionali e fornisce in output la moda delle previsioni dei singoli alberi.

  Prima di applicare il random forest ai dati di training, però, è stata effettuato un processo di feature selection per ulteriormente migliorare le performance del modello. Più precisamente, è stata applicata la cosiddetta *truncated SVD, Singular Value Decomposition*, che ha l'obiettivo di ridurre la dimensionalità della matrice in input. La singular value decomposition è una tecnica di feature synthesis che permette di fattorizzare una matrice tramite l'utilizzo di autovalori e autovettori. In particolare, la Truncated SVD è molto utilizzata in caso di presenza di matrici sparse come nel text mining. In questo caso preciso, il numero desiderato di feature da mantenere è stato scelto pari a 500. Una volta creata la matrice ridotta, quindi, è stato applicato il classificatore Random Forest scegliendo come numero di alberi per foresta 40. Anche

questo modello è stato trainato su una porzione di dataset pari al 0.9 e validato sul restante 0.1.

4 Results and Evaluation

I risultati ottenuti in fase di training sono stati molto soddisfacenti in termini di accuracy di entrambi i modelli:

- Neural Network:
  - Accuracy sul training set: 0.9821
  - Loss sul training set: 0.0606
  - Accuracy sul validation set: 0.9521
  - Loss sul validation set: 0.1842
- Random Forest:
  - Accuracy sul training set: 0.88

Nonostante le performance della NNet risultino più efficaci, ottenendo un'accuracy in fase di test pari a 0.94 , anche il random forest ottiene un buon livello di efficacia anche se leggermente inferiore alla NNet, con un'accuracy pari a 0.86.

Per quanto riguarda la rete neurale si può notare che non ci sono grossi miglioramenti durante le epoche e che si presenta un leggero overfitting poiché sia la loss che l'accuracy risultano rispettivamente più bassa e più alta in fase di training.

5 Conclusions

Valutando le metriche a disposizione e i modelli sviluppati, si evince che la rete neurale ottiene risultati più soddisfacenti ma entrambe riescono ad eseguire il task di classificazione con successo. L'obiettivo dell'analisi è stato raggiunto e ci si aspetta, quindi, che data una review lasciata da un utente su un qualsiasi prodotto, essa venga classificata nella corretta categoria di appartenenza del prodotto. Un possibile miglioramento consiste nell'utilizzare un numero maggiore di categorie prodotto ma questo richiede uno sforzo computazionale abbastanza pesante e potrebbe necessitare dello sviluppo di un modello differente.