Intelligenza Artificiale
Relazione Elaborato

Mirko Del Moro 6368874

1. Introduzione

In questo elaborato si è fatto uso dell'algoritmo Naïve Bayes relativamente al problema dell'analisi del sentimento, un campo del Natural Language Processing che si occupa di interpretare e classificare opinioni e sentimenti all'interno di un testo attraverso alcune tecniche di analisi testuale computazionale. In particolare, si è analizzato il sentimento relativo a recensioni di farmaci presenti su un dataset reperibile sul repository UCI . Il dataset è costituito da alcuni attributi e per ogni riga di questo è presente il nome del farmaco utilizzato e una recensione del paziente sul prodotto, affiancata da un voto personale (fino a 10 stelle).

2. Naïve Bayes

Il classificatore Naïve Bayes utilizza il Teorema di Bayes, il quale fornisce un modo per calcolare la probabilità di un'ipotesi in base alla nostra precedente conoscenza. Questo algoritmo si basa sull'assunzione "naïve" che l'effetto dell'attributo su una data classe è indipendente condizionalmente dai valori degli altri attributi; quindi la presenza o l'assenza di una particolare feature non influenza la presenza o l'assenza di altre caratteristiche. Il problema della classificazione è quindi posto in termini probabilistici. Grazie a quest'assunzione è possibile considerare:

$$P(a_1, a_2, \dots, a_n | v_j) = \prod_i P(a_i | v_j)$$

dove a_1, … ,a_n sono gli attributi e v_j rappresenta la classe nota.

In questo progetto sono state utilizzate due diverse implementazioni di questo algoritmo: la versione multinomiale e quella di Bernoulli, entrambe disponibili nella libreria scikit-learn.

3. Sentiment Analysis

I modelli definiti dalla sentiment analysis sono in grado di identificare ed estrapolare da un testo qualsiasi (e.g. un commento, una recensione, un documento) un'opinione e quindi percepire una polarità generale all'interno di esso, riuscendo a classificarlo come positivo, neutrale o negativo. I modelli si basano su tecniche di classificazione ed estrazione del testo, tra cui quella di individuare keywords o di eliminare parole molto frequenti.

4. Procedimento

Inizialmente è stata utilizzata la libreria Pandas, disponibile per il linguaggio Python, per caricare il dataset, il quale era già suddiviso in un training-set e un test-set. Una volta caricati i sets sono state isolate le colonne di interesse per ciascuno dei due, ovvero la colonna di testo che corrisponde alle recensioni dei pazienti e quella dove è indicato il voto, che corrisponde al nostro target. Per quanto riguarda quest'ultima colonna ogni valutazione del paziente, che arriva fino ad un massimo di 10 stelle, è stata trasformata secondo tre diverse categorie: ad un rating ≤ 4 è stato associato il simbolo '-1', ad un rating ≥ 7 il simbolo '1' e il simbolo '0' a tutti gli altri voti (rating compreso tra 4 e 7 esclusi).
In merito al training-set circa il 66% dei voti ha un rating positivo ('1'), il 25% un rating negativo ('-1') e il restante 9% un rating considerato neutrale ('0').

A questo segue una fase di preprocessing che riguarda le colonne delle recensioni, che ha permesso di "pulire" e semplificare il testo. Sono stati eliminati dai testi i numeri, i caratteri speciali e la punteggiatura e ogni carattere alfabetico è stato trasformato in minuscolo. A questo punto si è suddiviso il testo in parti più piccole (tokens) e si è estratto il bagaglio di parole (bag of words). Per fare ciò è stata utilizzata la funzione *CountVectorizer ()*, presente in scikit-learn, la quale restituisce una rappresentazione matriciale dell'intero testo, costruendo un dizionario di tutte le parole presenti nel testo per ogni documento. Si è convertito quindi il testo di ogni recensione in una matrice sparsa di tokens.
Per rappresentare ogni recensione si è usato un approccio n-grams, ovvero non sono state usate solo le singole parole per rappresentare il testo, ma sono state considerate anche coppie e triple di parole adiacenti; questo è stato possibile grazie al parametro *ngram_range ()* presente in *CountVectorizer ()* che permette di definire il minimo e il massimo numero di parole considerate per la rappresentazione del testo.
Inoltre, per ridurre la dimensione del dizionario sono stati eliminati termini che avevano una frequenza maggiore di 0.9, ovvero quei termini che apparivano in più del 90% dei documenti, grazie al parametro *max_df ()* presente in *CountVectorizer ()*.
Per il classificatore multinomiale la matrice sparsa risultata dalla vettorizzazione contiene features che rappresentano effettivamente il numero di occorrenze di ogni token nel documento, mentre per quello di Bernoulli si è usato una vettorizzazione binaria (grazie al parametro *binary* di *CountVectorizer ()*) che restituisce una matrice sparsa di features sotto forma di valori binari: 1 se è presente almeno un'occorrenza di quel token nel documento, 0 altrimenti.
Infine, sono stati inizializzati i due diversi classificatori, utilizzando le implementazioni presenti entrambe in scikit-learn, sono stati addestrati sui dati del training-set precedentemente estratti e sono stati utilizzati per l'analisi e la predizione del test-set.
Entrambi i modelli sono stati valutati calcolando la matrice di confusione, l'accuratezza e la Kappa di Cohen.
Come aspetto di valutazione è stato considerato il rating generale (Overall Rating).

5. Risultati

Di seguito sono riportati i risultati ottenuti dai due diversi classificatori ed è stata prodotta la matrice di confusione per ciascuno modello. È possibile notare come i risultati siano migliori per quanto riguarda il modello multinomiale, soprattutto nella valutazione della Cohen's Kappa.

5.1 Multinomial Naïve Bayes

Aspect	Source	Accuracy	Cohen's Kappa
Overall Rating	Drugs.Com	0.86	0.67

Tabella 1: Valutazione modello multinomiale

$$\begin{bmatrix} 9990 & 0 & 3507 \\ 734 & 1218 & 2877 \\ 296 & 4 & 35140 \end{bmatrix}$$

Figura 1: Matrice di confusione per il modello multinomiale

5.2 Bernoulli Naïve Bayes

Aspect	Source	Accuracy	Cohen's Kappa
Overall Rating	Drugs.Com	0.77	0.42

Tabella 2: Valutazione modello di Bernoulli

$$\begin{bmatrix} 5967 & 0 & 7530 \\ 666 & 103 & 4060 \\ 58 & 4 & 35378 \end{bmatrix}$$

Figura 2: Matrice di confusione per il modello di Bernoulli

References

[1] Aspect-Based Sentiment Analysis of Drug Reviews Applying Cross-Domain and Cross-Data Learning

[2] Scikit-Learn: https://scikit-learn.org/stable/user_guide.html

[3] Pandas: https://pandas.pydata.org/docs/user_guide.html

[4] D. Barber. Bayesian Reasoning and Machine Learning. Cambridge University Press, 2012.