Università degli studi di
Milano-Bicocca

Text Mining and Search

Final Project

Text Classification on 20newsgroup Dataset

Authors:
Louis Fabrice Tshimanga - 847529-
l.tshimanga@campus.unimib.it
September 2, 2020

Abstract

Il 20newsgroup dataset originale è una collezione di circa 20 000 documenti ripartiti pressoché equamente in 20 classi in base al newsgroup originario, raccolta nel 1995. Un newsgroup è uno spazio virtuale adibito a gruppo di discussione per un determinato argomento (topic). Il dataset è diventato uno standard per sviluppare algoritmi di Text Mining, Classification e Clustering.

1 Introduzione

Una delle applicazioni più frequenti ed efficaci del Machine Learning, nell'ambito dell'elaborazione del linguaggio naturale in forma testuale e del mining di informazioni, è la Text Classification. La classificazione è un problema in cui ad un input testuale va associato un output qualificante la classe del testo. Nell'ambito di questo progetto si è scelto di risolvere il problema di classificazione fra classi multiple (20) con unica classe di output, in un set di etichette predeterminate e disponibili per l'algoritmo di apprendimento, ossia un problema di Supervised Learing.

Il 20newsgroup dataset è una collezione di documenti ripartiti pressoché equamente in 20 classi in base al newsgroup originario. Un newsgroup è uno spazio virtuale adibito a gruppo di discussione per un determinato argomento (topic). I documenti sono quindi post, messaggi scambiati in comunità di utenti, che possono rivolgersi a membri specifici ed esulare anche dall'argomento di discussione ("andare off-topic"), o citare messaggi precedenti. In questo progetto si è tenuto conto di tali caratteristiche nell'implementare e nel discutere i risultati dei modelli di classificazione, che non può e non deve essere accurata al 100% affinché si possa confidare che il modello stia acquisendo informazione rilevante, dove la rilevanza è un concetto solo approssimabile ma centrale.

2 Dataset

Il dataset è disponibile tramite il pacchetto python scikit-learn (https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html). La funzione adibita al recupero dei dati permette di filtrare le categorie testuali, nonché i singoli file

rimuovendo "headers", "footers", "quotes" dei messaggi, cioè sezioni del messaggio che possono contenere titoli inerenti lo specifico newsgroup, gli utenti coinvolti, le loro firme ed in generale informazioni che permettono ai modelli di classificazione di andare in overfitting a prescindere dal contenuto semantico dei messaggi di testo di rilevanza per il lettore umano. La funzione di scikit-learn permette inoltre di caricare separatamente un sottoinsieme di training (11314 istanze) e uno di testing (7532 elementi).

3 Metodi e Risultati

Una volta scelto come "core task" quello della classificazione, bisogna selezionare un modello ed un formato dei dati adeguato all'input che il modello richiede. A questa procedura logica ne corrisponde una operativa in direzione opposta, con il preprocessing dei dati ed il passaggio degli stessi al modello. Infine, un lavoro realistico generalmente prevede feedback e riprese dei singoli passaggi schematizzati.
In questo progetto sono stati utilizzati due classificatori implementabili tramite scikit-learn, SGDClassifier e MultinomialNB, che possiamo rispettivamente far afferire alle Support Vector Machines (SVM, classificatori lineari, al netto delle funzioni kernel) e ai classificatori probabilistici Bayesiani "naif" (Naive Bayes, NB).
I due classificatori ricevono come input dei vettori di features. Per questo motivo i documenti sono stati rappresentati, secondo un procedimento ormai canonico, con vettori dei valori TF-IDF delle parole contenute nel corpus. Il valore TF-IDF è calcolato come prodotto di Term Frequency (conteggio del termine nel documento, sull'estensione di termini del documento) e Inverse Document Frequency (numero di documenti, su numero di documenti contenenti il termine), e pesa l'importanza di un termine come "feature" caratterizzante. I vettori così generati sono sparsi, con valori non nulli in media 3 ordini di grandezza inferiori rispetto al numero di features.
Inizialmente si è utilizzata la SVM praticando solamente la vettorizzazione in valori TF-IDF, ottenendo un'accuracy del 68.39%. Dalla matrice di confusione e dall'heatmap ricavata si può vedere come il sistema tenda a confondere topic simili o indicare la classe "misc[ellaneous]". Rimuovere le stop-words, normalizzare gli accenti e diminuire le features utilizzate a 20 000 (su 101 631, ispirandosi ad un principio di Pareto, non dissimile da quello di Zipf) ha reso l'accuracy leggermente peggiore, 67.93%. A questo punto si sono implemen-

tati tokenization (libreria nltk, funzione WhitespaceTokenizer) e stemming (funzione SnowballStemmer) separatamente rispetto alla funzione che crea i vettori calcolando TF-IDF, così da poter segmentare gli effetti. L'accuracy è calata ancora leggermente a 67.68%, rendendo come principale processo sospetto quello di stemming, in cui le parole vengono riportate ad una radice che può di fatto non appartenere al vocabolario e privare il classificatore di sfumature e variazioni sostanziali.
Utilizzando il classificatore bayesiano si è notato lo stesso calo di performance, sebbene su una baseline maggiore rispetto all'SVM, con accuracy di 69.77% per il modello a features ridotte e normalizzato, 69.70% per quello con stemming.
A questo punto si è deciso di optare per la lemmatization, che al contrario dello stemming riporta le parole al vocabolo da cui derivano, e non ad una più astratta radice. Operando la medesima pipeline con lemmatization in luogo dello stemming, l'accuracy raggiunta è di 70.03%.
Essendo i risultati comunque piuttosto vicini tra loro, si è anche ridotto il problema alla classificazione per un sottoinsieme di label, scelte utilizzando tutti i prefissi identificativi e mantenendo la prima classe più numerosa tra quelle con lo stesso prefisso, riducendo il problema da 20 a sole 7 classi. Il classificatore bayesiano è stato in questo caso in grado di raggiungere l'84.51% di accuracy. La matrice di confusione mostra in questo caso soprattutto un "leakege" dalla categoria "alt.atheism" a quella "soc.religion.christian".

4 Discussione e Conclusione

I risultati ottenuti dai due modelli di classificazione mostrano come il preprocessing dei testi sia fondamentale per migliorare le performance, ma non possa avvenire utilizzando una somma indiscriminata delle tecniche disponibili, che sono invece da scegliersi in base alle specifiche del dataset, del problema e della soluzione che si scelgono di affrontare. Le SVM sono strumenti spesso efficaci anche per dati che non siano linearmente separabili nello spazio di partenza, ma l'approccio probabilistico bayesiano si è rivelato ancora più versatile ed in generale le tecniche derivate dal calcolo delle probabilità sono uno strumento particolarmente potente in applicazioni testuali in cui distribuzione e argomento delle parole siano centrali (es. Topic Modelling). Le performance relativamente modeste si possono attribuire alla semplicità e velocità dei modelli implementati, così come alle specifiche del dataset e ai bias

nell'identificazione o persino produzione delle categorie. Ad esempio si ha un newsgroup di oggetti in vendita che possono dare sovrapposizioni verbali con i newsgroup di elettronica di consumo o veicoli a motore, oppure newsgroup di ateismo che si confonde con quello di sociologia del cristianesimo. Nei forum e gruppi di discussione l'apertura di topic è in parte arbitraria, così come la categorizzazione, e a sua volta sensibile a messaggi off-topic. Inoltre ai modelli non sono state fornite informazioni di intitolazioni e autori dei documenti, limitando la possibilità di intuire il contesto a prescindere dal contenuto. Per tale ragione e dati gli specifici errori, i risultati ottenuti riducendo il problema a 7 categorie sono soddisfacenti. Il passo successivo sarebbe quello di ricategorizzare i testi facendo clustering o topic modelling (Unsupervised Learning), ed eventualmente applicare classificatori con apprendimento supervisionato sulle nuove etichette generate internamente.