Relazione per l'esame di Intelligenza Artificiale

Arman Lagevardi 6226749

6 settembre 2021

1 Introduzione

In questo elaborato ho messo a confronto le curve di apprendimento di due classificatori: il **Bernoullian Naive Bayes** e il **Perceptron**, applicandoli a due Dataset di documenti testuali quali **20News Group** e **Reuters21578**.

2 Classificatori

Generalmente, i classificatori, hanno bisogno di un dataset, ovvero di un insieme finito di valori aventi un "etichetta" (o classe), per poter fare delle predizioni sulla classe di appartenenza di un nuovo documento su base statistica o su modelli matematici più o meno complicati.

- **Bernoullian Naive Bayes:** è un algoritmo di apprendimento supervisionato molto semplice e veloce che, tramite forti assunzioni sui dati (nel nostro caso documenti testuali), riesce ad attribuire ai documenti le giuste classi di appartenenza (sport/politica/...) discretamente bene. La sua unicità sta nell'assumere che una particolare*feature*, data la classe della variabile, è indipendente da qualsiasi altra all'interno del documento.
- **Perceptron:** è un classificatore binario ed è iterativo, cioè ad ogni iterazione (o epoca) aggiorna la sua funzione affinché riesca a trovare un iperpiano migliore per separare i dati (vettori) su un piano cartesiano, e non termina se non per un limite preimpostato di epoche o perchè ha trovato il numero minimo di errori accettabili). La funzione di questo algoritmo non è detto che converga in un numero finito di passi, se il suo training set non è linearmente separabile.

2.1 Datasets

I dataset utilizzati in questo esperimento sono:

- **20News Group**
  Disponibile qui
- **Reuters 21578**
  Disponibile qui

Queste raccolte sono costituite da numerose raccolte di documenti testuali forniti di TAG come ad esempio $< TOPIC > earn < /TOPIC >$ per l'argomento del testo . Il primo contiene circa 10'000 documenti divisi in 20 macro argomenti, il secondo, invece, conta circa 7'000 documenti, presentando solo le 10 categorie più ricorrenti.

3 Librerie

Il progetto è scritto interamente in Python 3. Ho utilizzato la seguenti librerie esterne nel mio progetto:

- **Scikit-learn** per le implementazioni dei due classificatori, per caricare il dataset 20news Group, per vettorizzare gli input e per calcolare lo *score*.
- **Nltk - Natual Language Tool Kit** per l'estrazione della *bag of words*.
- **matplot** per mettere su grafico in modo chiaro i miei risultati.

4 Bag of Words

I dataset vengono presentati ai classificatori sottoforma di task binari, cioè solo **una** categoria per volta viene distinta dalle altre, così che i due algoritmi non introducano rumore e disperdano le loro energie quando sono presenti categorie con un basso numero di esemplari. Per estrarre la *Bag of Words* di ciascun dataset ricorro alla funzione CountVectorizer (della libreria scikit) che accetta un tokenizer della libreria NLTK, grazie a loro posso ottenere una rappresentazione matriciale dell'intero dataset costruendo un dizionario di tutte le parole presenti nel dataset per ogni documento i salvando la feature di ciascuna parola w in *X[i, j]* dove j è l'indice della parola w nel dizionario. Ho scelto, inoltre, di eliminare tutte le parole che sono presenti nella lista stop-words='english', oltre quelle di lunghezza inferiore alle 3 lettere (su base empirica). Inoltre utilizzo la stessa funzione, ma senza i parametri per le stop-words e per le parole di dimensione minima, per convertire la lista dei target (contenente le classi degli attributi dei vari documenti) in un vettore.

5 Classificatori e Cross Validation

Successivamente alla creazione della Bag of Words ho inizializato una coppia di classificatori per ogni dataset, tramite le implementazioni esistenti in sklearn.naive_bayes e sklearn.linear_model rispettivamente per il Bernoullian naive Bayes e per il Perceptron. Per calcolare infine l'andamento dell'accuratezza col numero di esempi di ciascun dataset è stato utilizzato un cross-validator sulle 100 frazioni successive dell'intero dataset. Ogni subset ottenuto in modo random (80%) viene utilizzato per allenare entrambi i classificatori. Infine si effettua una predizione sul subset restante (20%) che rappresenta il testset. Viene calcolata infine la media degli score ottenuti su ciascuna iterazione del cross valdation.

6 Risultati

Nel seguente paragrafo vengono riportati gli score ottenuti dai due classificatori al variare del numero di esempi. Possiamo notare che il classificatore Perceptron, in entrambi i casi, ottiene uno score quasi sempre superiore rispetto al Bernoullian naive Bayes che, in particolare nel dataset *Reuters*, ottiene uno score molto vicino al 99%, ottenendo sin da subito risultati eccezionali.

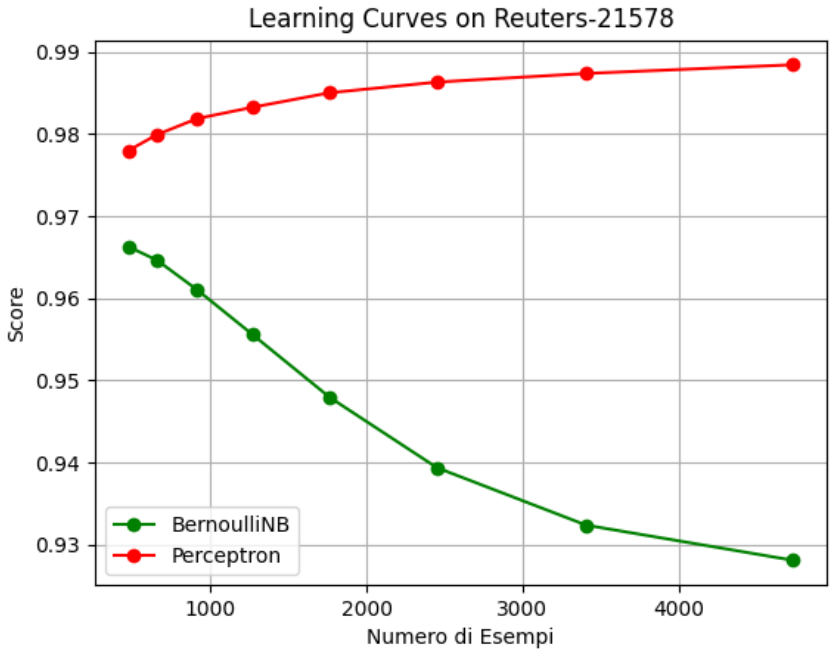

Figura 1: Confronto dei due algoritmi sul dataset "Reuters"

Notiamo anche come il classificatore Naive si trova spesso in difficoltà a riconoscere correttamente le classi (presentate secondo task binari, in accordo alla sezione "Bag of Words") quando gli esempi di *test* si fanno numerosi, pur mantenendo un buon punteggio (sempre sopra il 91%).

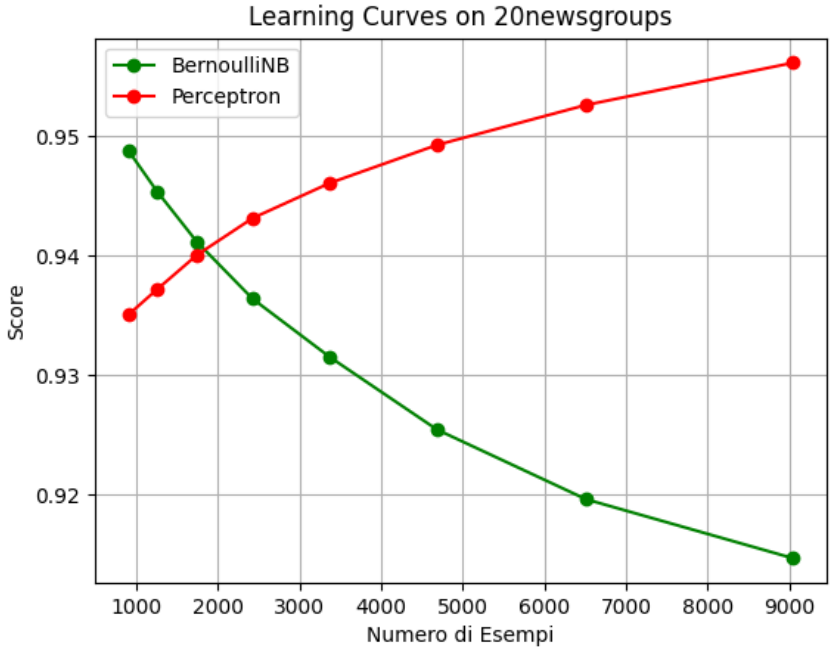

Figura 2: Confronto dei due algoritmi sul dataset "20NewsGroup"

7 Conclusioni

In conclusione, i miei esperimenti dimostrano che, se presentiamo dei task binari ai due classificatori possiamo ottenere *ottimi* risultati, in particolare il Perceptron continua a migliorare i suoi score con l'aumentare del numero di file di esempio, viceversa il Bernoullian Naive Bayes, scarseggia nelle suddette condizioni.