MACHINE LEARNING

Autori: Simone Trinca 0298986, Fabio Fontana 0298720

ANNO ACCADEMICO: 2021-2022

1. Introduzione

Il seguente progetto, realizzato e presentato relativamente al corso di ML 2021/2022, presents come fine ultimo l'applicazione di metodologie di apprendimento supervisionato a task di *Topic classification*. L'elaborato prevede una prima introduzione e breve descrizione dei modelli utilizzati seguita dalla descrizione del Dataset utilizzato contenente provenienza e costituzione di questo. In seguito viene descritta la pipeline definita per l'implementazione ultima dei modelli, descrittiva delle varie operazioni intermedie realizzate, come lowercasing, tokenization, per citarne alcuni. Si passa poi all'analisi dei risultati in cui vengono riportati i risultati ottenuti in termini di accuracy seguita dalla fase di analisi degli errori, in cui si vanno a valutare gli errori commessi dai modelli cercando di capirne la causa e soprattutto di definirne i confini. Conclude poi l'elaborato un accenno a lavori futuri che possono essere applicati a tale dataset basati sostanzialmente sull'adozione di tecniche che attualmente rappresentano lo stato dell'arte, come LSTM e Transformer.

2. Modelli

l Machine Learning è una branca dell'Intelligenza Artificiale che studia gli algoritmi in grado di apprendere e conseguentemente migliorare le prestazioni, grazie all'esperienza.

"Si dice che un programma apprende dall'esperienza E con riferimento ad alcuni task T e con misurazione delle performance P, se le sue performance nel compito T, misurate mediante P, migliorano con l'esperienza E

Le tecniche di Machine Learning propongono una serie di algoritmi, strategie e tecniche per la produzione di soluzioni sub-ottime, ma efficaci. Nel processo di apprendimento (learning) i dati suggeriscono l'ipotesi risulutiva per la funzione di mapping. l pattern recognition avviene mediante un approccio induttivo, ossia apprendere elementi utili da un insieme di dati, partendo da un set di esempi. Siamo dunque nell'ambito del Supervised Learning.

2.1. Multinomial Naive Bayes

Un classificatore Naive Bayes è un modello probabilistico di machine learning usato per task di classificazione. Tale modello si basa sul teorema di Bayes:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

dove, sostanzialmente riusicamo a caratterizzare la probabilità che l'evento A si verifichi dato che B si è verificato. Un'assunzione fatto da tali modelli è l'assunzione *Naive* di indipendenza delle features. Nell'ambito di applicazione a topic di classification, possiamo di fatto vedere il teorema nella seguente forma:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

dove, $P(y)$ sono detto *prior* e vengono rilevate dalla distribuzione di probabilità delle etichette (label) all'interno del dataset. $P(X)$ essendo costante, non influisce nell'attribuzione

dell'etichetta che avviene seguendo l'approccio:

$$y = \underset{y}{\operatorname{argmax}}\, P(y) \prod_{i=1}^{n} P(x_i|y)$$

. Nel caso di classificazione multi-classe si utilizza il modello Multinomiale, mentra nel caso di classificazioni binaria il modello bivariato.

2.2. Decision Tree

Nell'analisi delle decisioni, un albero decisionale può essere utilizzato per rappresentare visivamente ed esplicitamente le decisioni e il processo decisionale. Come dice il nome, utilizza un modello ad albero delle decisioni. Per quanto riguarda l'apprendimento sceglie una features e un valore soglia rispetto al quale fare la partizione e divide. Dopodiché sceglie nuovamente un'altra features e ricerca un'altra soglia e divide ancora. Ogni nodo dell'albero corrisponde ad una partizione (più o meno grande). Quando una regione, cioè un nodo, ha tutti punti della stessa classe allora questa viene dichiarata foglia. Se ci sono punti di classe diversa posso decidere se partizionarla o non farlo e per esempio assegnare quella foglia alla classe più presente. L'albero è finito quando smesso di partizionare. Una volta finito l'apprendimento è facilissimo classificare perché basta far scorrere gli elementi nell'albero.

2.3. Random Forest

Il modello *Random Forest* consiste nell'applicazione del bagging ad un insieme di alberi decisionali (Decision trees). La classificazione è ottenuta poi mediante voting tra gli alberi. Il bagging consiste proprio nell'addestrare L modelli differenti e ottenere la predizione finale come media della predizioni restituite da ogni modello, per il dataset a disposizione. Sono metodi che rientrano nell'ambito dell'*ensable methods*, modelli per cui la performance è migliorata dal fatto che vari modelli vengono, in qualche modo, combinati tra loro. Mostrano pertanto una performance migliore rispetto all'uso di un singolo modello. Le random forest consentono dunque di limitare il problema tipico del Decision Tree, ossia, la sua dipendenza dal dataset di applicazione (alta varianza). Infatti, per dataset che differiscono leggermente, il decision tree fornisce risultati differenti. Nel caso di classificazione la predizione restituita dal modello *RM* è quella maggioritaria, ossia quella restituita dal maggior numero di alberi (voting), mentre nel caso di regressione sarà la media delle predizioni degli alberi stessi.

2.4. Support Vector Machine

Le *Support Vector Machines* sono dei classificatori lineari che dato un input x (vettore delle istanze da classificare) mediante w (vettore dei pesi) e b (termine noto o bias), distinguono nel piano, due semipiani a seconda della dimensione, in modo da classificare gli elementi delvettore x. Computazionalmente sono semplici. Un classificaore lineare lo possiamo vedere come:

$$h(x) = g(w^T \phi(x_i) + w_0)$$

. Per ogni elemento del training set, viene definito il margine funzionale dell'elemento i-esimo come

$$\bar{\gamma_i} = t_i(w^T \phi(x_i) + w_0).$$

E' importante definire inoltre il margine geometrico, che corrisponde sostanzialmente, al prodotto tra t_i e la distanza dell'istanza x_i dal piano, che di fatto corrisponde alla lunghezza del segmento congiungente x_i e la sua proiezione sul piano di confine. L'obiettivo delle SVM è quello di determinare l'iperpiano di maggior margine. Si ottiene il margine andando a risolvere un problema di ottimizzazione quadratica vincolate, generalemente utilizzando la Lagrangiana e passando per il suo duale. Vengono selezionati, tra tutti gli elementi del Training Set, solo quelli che contribuiscono in modo attivo caratterizzati da $\alpha_i > 0$ (coefficiente di Lagrange), Tali coefficienti assumono valore maggiore di 0 solo per per elementi che giacciono su frontiera.

2.5. Multy-layer Perceptron

Il percettrone multistrato (MLP) è un algoritmo di apprendimento supervisionato che apprende una funzione $f(.) : R^m \rightarrow R^n$, attraverso l'addestramento su un set di dati, dove m è il numero di dimensioni per l'input e n è il numero di dimensioni per l'output. Dato un insieme di caratteristiche $X = (x_1, x_2, \ldots, x_m)$ e un target y, può apprendere una funzione non lineare per la classificazione o la regressione. Si differenzia dalla regressione logistica in quanto tra lo strato di ingresso e quello di uscita possono esserci uno o più strati non lineari, detti strati nascosti. Lo strato più a sinistra, noto come strato di ingresso, è costituito da

un insieme di neuroni che rappresentano le caratteristiche in ingresso. Ogni neurone dello strato nascosto trasforma i valori dello strato precedente con una somma lineare ponderata seguita da una funzione di attivazione non ineare. Lo strato di uscita riceve i valori dall'ultimo strato nascosto e li trasforma in valori di uscita. L'addestramento è reso possibile mediante l'utilizzo di algoritmi di ottimizzazione, come lo la discesa del gradiente, applicati ad una funzione di Loss. Tali algoritmi sono applicabili in quanto, mediante **Backpropagation** le derivate parziali della funzione di Loss rispetto ai parametri sono note.

3. Topic Classification

Il task scelto è il **Topic Classification**, ossia l'attribuzione di un'etichetta inerente il topic trattato dalla frase oggetto di predizione. Si è in ambito di Classificazione multi-classe poichè si hanno le $n = 5$ seguenti classi: "Hystory", "Geography", "Science", "Music", "Literature" da attribure alle sentences del test set, grazie ai modelli di predizione ottenuti basando l'addestramento sul training set. Il codice è stato sviluppato in linguaggio Python, utilizzando l'editor di notebook Jupyter. Sono state utilizzate varie librerie inserite in Python tra cui Pandas, Sckit-learn, Seaborn, Matplotlib.

3.1. Dataset

Il Datset utilizzato è stato preso da una competizione universitaria, interna, somministrata dal IST (Istituto Superior Tecnico) di Lisbona. Il training set è composto da 9500 elementi, mentre il test set da 500. La singola istanza è costituita da una sentence, in lingua inglese, e l'etichetta corrispondente relativa al topic. Un pezzo di dataset è visibile in figura 1, come come viene fornito in formato txt.

```
LITERATURE      "This book by Virginia Woolf inspired Michael Cunningham's novel
GEOGRAPHY       "The ""amiable"" former name of the Tongan archipelago" the Frie
GEOGRAPHY       "The Rhine Valley occupies one-third of this 62-square-mile coun
MUSIC   "PBS fans know that ""Evening at Pops"" refers to this city's Pops"
LITERATURE      "In 1996 he simultaneously published ""The Regulators"" as Richa
HISTORY "In 1843 Congress allocated $30,000 to string one between Baltimore & Wa
SCIENCE According to Chuck Jones, whenever possible, this force of nature was to
MUSIC   This 1940 Disney film featured the music of Bach, Beethoven, Stravinsky,
SCIENCE The Babylonians kept abreast of the times using a form of this instrumen
HISTORY Dying in 2009 at age 113, British WWI vet Henry Allingham was the last o
SCIENCE -273 Celsius    absolute zero
```

Figure 1: Training set

.

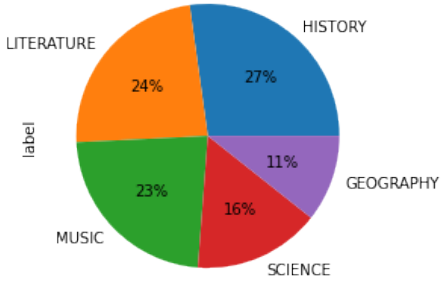

Figure 2: Esempi per ogni etichetta

3.2. Implementazione modellii

Per poter applicare i modelli descritti in precedenza e forniti dalla libreria Scikit-learn è stata necessaria un prima frase di importazione, formattazione e pre-processing del Dataset. Dataset che mostra la ripartiziane di esempi per categoria di etichetta come in Figura 2.
Tra le operazioni di pre-processing implementate, troviamo:

1. Lowercasing
2. Tokenizzation
3. Lemmatization
4. POS tagging

Successivamente, parte fondamentale per consentire l'utilizzo di elementi testuali, è la rappresentazione delle sentences in forma vettoriale numerica. Per tale obiettivo, si è scelta la rappresentazione mediante vettorizzazione *Term frequency - Inverse Docuemnt Frequency (Tf-idf)* per quanto riguarda la sentences, mentre per le etichette sono state queste trasformate in valori numerici, associando ad ognuna un univoco valore numerico reale. Tale processo è stato realizzato allo stesso modo per il test set, in modo da renderlo processabile. Dopodichè, si è passati all'implementazione dei modelli descritti, passando a questi come input per il training, il testo e le etichette rappresentate vettorialmente come descritto.

4. Risultati

Come Basaline può essere considerata la scelta dell'etichetta più frequente. In tale ottica, scegliendo sempre *Hystory* si otterrebbe un'accuracy pari a 27.00% (Vedi Figura 2). La scelta randomica di un'etichetta, essendo 5, avrebbe pertanto una probabilità minore e pari a 25.00% nel caso di equidistribuzione delle etichette, pertanto viene preso come baseline il val-

ore 27.00%. I risultati ottenuti con i vari modelli utilizzati sono in seguiti riportati. MultiNB ha ottenuto un'accuracy pari a 84.39%. Decision Tree (DT) ha ottenuto 72.20%, mentre Random Forest (RF) 81.00%. Il modello SVM 87.00%, mentre MLP 86.00%. Tali valori sono riassunti in Tabella 1. Si evince che, con tale configurazione, il modello maggiormente performante sono le Support Vector Machine, che utilizzano un kernel lineare.

	Classifier	Accuracy
Baseline	Hystory	27,00%
Model1	NB	84.39%
Model3	DT	72,20%
Model4	RF	81,00%
Model2	SVM	87,00%
Model5	MLP	86,00%

Table 1: Confronto dei Risultati

5. Analisi degli errori

L'analisi degli errori è un passo importanto verso l'ottica di miglioramento continuo del modello. Possiamo pertanto avere un'idea del perchè alcune predizione delle frasi sono sbagliate. Per capire cosa non va, sono state costruite $n = 5$ matrici di confusioni, che ci mostrano quante volta la classificazione è sbagliata e quale etichetta viene assegnata a scapito di quella corretta. Sono di seguito riportate le matrici, una per ogni modello.

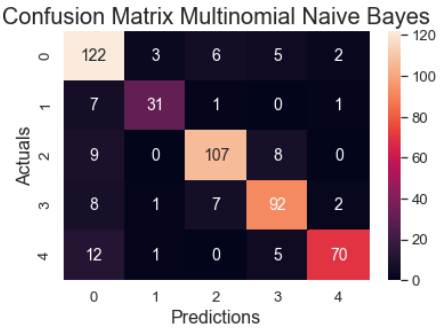

Figure 3: Matrice di Confusione NB

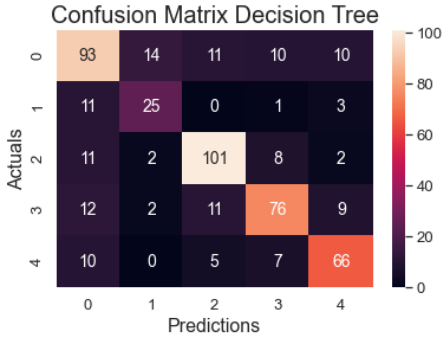

Figure 4: Matrice di Confusione DT

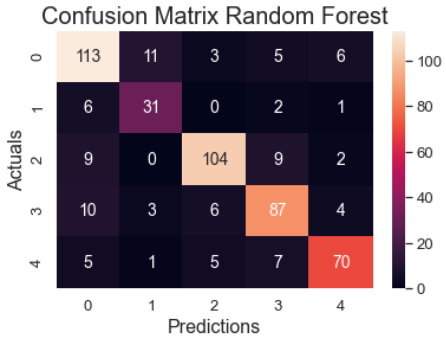

Figure 5: Matrice di Confusione RF

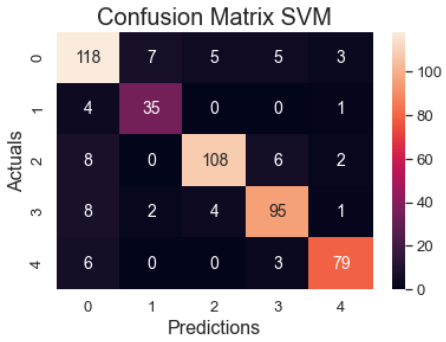

Figure 6: Matrice di Confusione SVM

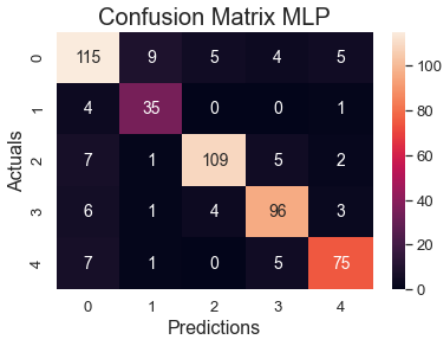

Figure 7: Matrice di Confusione MLP

E' stata inoltre valutata la coerenze di risposte esatte restituite in modo concorde dai modelli. Per

quanto riguarda l'intero pacchetto di modelli si evince che il 60% dei documenti viene classificato bene da tutti i modelli, come mostra in Figura 8.

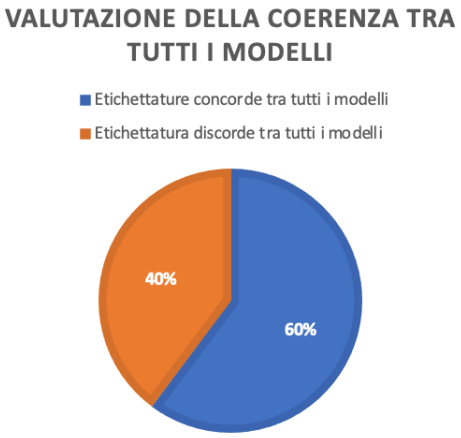

Figure 8: Coerenza modelli

Per quanto riguarda invece la coerenza dei modelli più performanti, SVM e MLP, risulta che questi sono concordi e classificano in modo esatto l'84% delle volte, come riportato in Figura 9.

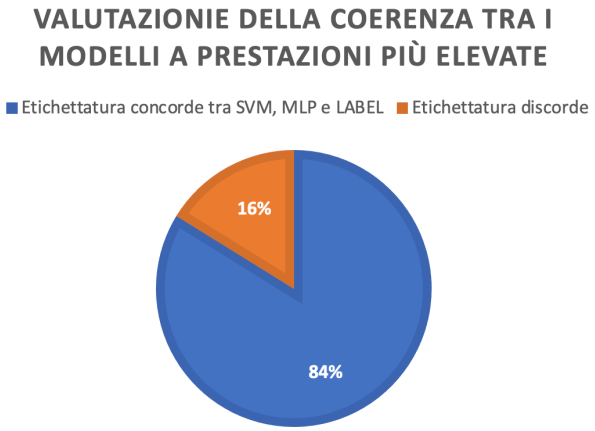

Figure 9: Coerenza MLP & SVM

Da alcune valutazioni effettuate emerge che:

- 45 volte su 500 (9%) risulta che SVM e MLP sono concordi ma in realtà sbagliano la predizione.
- 16 volte su 500 (3,2%) risulta che SVM fa una predizione corretta e MLP no.
- 11 volte su 500 (2,2%) risulta che MLP fa una valutazione corretta e SVM no.

6. Future work

Lo stato dell'arte ora, per quanto riguarda teniche in ambito di *Natural Language Processing* vede l'utilizzo di Reti Neurali Ricorrenti e Transformer. Questi si prestano molto bene a task linguistici in quanto dotati di memoria (RNN) e meccanismi di attenzione (Transformer) che consentono loro di ottenere e utilizzare un gran numero di informazioni estrapolate dai testi. Tra le reti neurali ricorrenti, sono molto utilizzate le architetture relative alle *Long Short Term Memory*, ossia delle reti neurali ricorrenti in grado di tenere in memoria e utiizzare informazioni precedentemente processate. Risolvono il probelma del vanishing/exploding gradient tipico delle semplici RNN attraverso dei *gates* che vengono utilizzati per il passaggio e lo storage delle informazioni ritenute rilevanti tenute in memoria grazie alla cell state. I Transformer rappresentano di fatto lo stato dell'arte attuale, e sostituiscono quanto fatto dalla LSTM con dei meccanismi che consentono di attenzionare a quale parte del processo si è interessanti per il conseguimento del task oggetto di analisi. Un'applicazione futura potrebbe proprio essere quelle di applicare strutture quali LSTM e Tansformer a tale task di Topic Classification, nell'idea di poter ottenere risultati molto più importanti rispetto a quelli ottenuti utilizzando dei modelli semplici e resi utilizzabili easily da Scikit-learn.

7. Bibliografia

[1] A Guide to Text-CLassification(NLP) using SVM and Naive Bayes with Python