Classificazione articoli ANSA via topic modelling

Alessandro Stefani[1] e Cristi Gutu[2]

[1] Corso di laurea in Statistica per le tecnologie e le scienze, nome d'arte Aleste matricola 1148387
alessandro.stefani.6@studenti.unipd.it
[2] Corso di laurea in Statistica per le tecnologie e le scienze, matricola 1147351
gheorghecristi.gutu@studenti.unipd.it

Sommario Questo progetto tratta la realizzazione di un classificatore di notizie in macro categorie provenienti dall'agenzia ANSA. Come scopo primario e' stato fissato la massimizzazione dell'accuratezza del classificatore, cioe' il valore definito dall'espressione $= (\frac{Previsioni\_corrette}{Totale\_tentativi\_previsione})$, in quanto, avendo un numero di osservazioni bilanciato per classe e con classi di equa importanza, non si rischia di incorrere nel paradosso dell'accuratezza. Sono state sfruttate tecniche di analisi dei testi, tra cui: Latent Dirchlet Allocation[1], Alberi di decisione[2], raggruppamento in n-grammi[3] dei termini.

Keywords: *Classification · Text mining · Text classification · n-gram · LDA · Python · sklearn*

1 Introduzione

In questo progetto vogliamo confrontare l'efficacia di alcuni modelli di rappresentazione dei documenti nel problema di classificazione, in particolare ci concentriamo su: LDA, Term Frequency e Term Frequency basato su n-grammi.

2 Dataset

I dati sono stati reperiti dall'agenzia ANSA, nota in Italia. Il campionamento degli articoli e' stato fatto ottenendo i link traminte il motore di ricerca DuckDuckGo. Si e' deciso di includere 6 macro categorie: (Economia, Politica, Cultura, Sport, Tecnologia, Cronaca), escludendo la categoria Mondo perche' si confonde con categorie come Economia e Politica, per ogni categoria sono stati reperiti 400 articoli sfruttando la ricerca mirata solo al sito ansa.it via le espressioni:

- site:ansa.it/sito/notizie/economia
- site:ansa.it/sito/notizie/politica
- site:ansa.it/sito/notizie/cultura
- site:ansa.it/sito/notizie/sport
- site:ansa.it/sito/notizie/tecnologia
- site:ansa.it/sito/notizie/cronaca

Cercando con queste espressioni si reperiscono risultati che appartengono soltanto ai temi citati sopra.

Ogni articolo e' composto da: (titolo, sottotitolo, testo, tags, categoria). Le tags sono parole chiavi che dovrebbero aiutare il lettore a contestualizzare il contenuto dell'articolo e di conseguenza categorizzarlo in qualche maniera. Si sono estratte da ogni articolo i campi citati sopra i cui valori sono stati salvati in formato JSON[3].

In totale si sono raccolti 2400 articoli[4], che sono stati suddivisi casualmente in training set, validation set e test set.

[3] formato di serializzazione per dati
[4] Dataset scaricabili in formato json all'inidirizzo: *https* : *//github.com/mastershef/big_data*

3 Esperimenti

Prima di poter utilizzare modelli per l'analisi testuale è necessario preprocessare i dati, questo e' stato fatto costruendo la seguente pipeline di preprocessamento: articoli |rimozione stopwords[5] |stemming |rimozione tags html |rimozione punteggiatura |rimozione numeri |rimozione link.

3.1 Baseline

Come basline si e' deicso di classificare utilizzando il modello *DummyClassifier* che classifica ogni articolo secondo la categoria piu' frequente del training set. L'accuratezza ottenuta e' del: 15%.

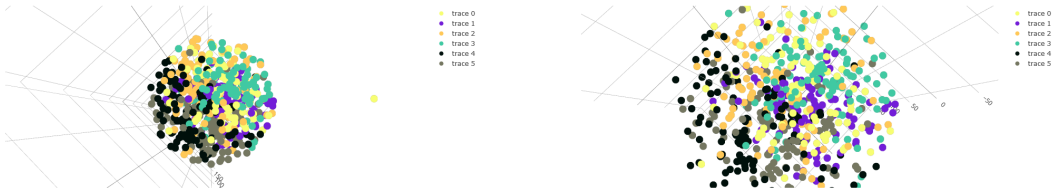
3.2 Analisi esplortive

Si e' notato che le categorie assegnate agli articoli erano in realta' micro-cateogire, percio' un ultimo step di preprocessamento e' stato riclassificare manualmente gli articoli, etichettandoli con le corrispondenti macro-categorie.

Ad esempio: Cultura ← (Libri, Cinema, Film); gli articoli con micro-categoria Libri o Cinema o Film, vengono rietichettati con la macro-categoria Cultura.

Successivamente si e' calcolata la term-document matrix[6] la cui classe di supporto in Python da' la possibilita' di specificare 3 parametri: ngram_range, min_df, max_df; per scegliere il range di ngrammi da prendere in considerazione si e' fatto riferimento al libro Social Media e Sentiment Analysis[3] dove si suggerisce che generalmente n-grammi con piu' di 3 termini non aggiungono contenuto informativo, i parametri max_df, min_df sono invece stati scelti in base alla distribuzione delle "frequenze degli ngrammi nei documenti" del train set.

Per continuare l'esplorazione, si e' cercato di visualizzare come i dati sono raggrupati in categorie riducendo lo spazio delle features con tSNE, l'approccio iniziale e' consistito nell'utilizzare la term document matrix come insieme di variabili esplicative, seguendo la pipeline: articoli preprocessati |Term Document Matrix |tSNE a 3 componenti.



(a) Vista normale (b) Vista ravvicinata

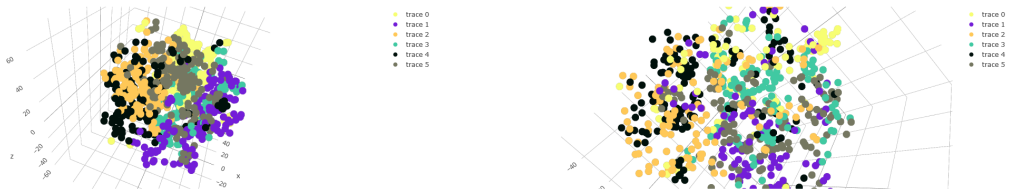
Figura 1: Riduzione della term document matrix da forma (2400, 4640) a forma (2400, 3) via tSNE.

Si vede in Figura 1 come articoli dello stesso tema sono abbastanza distaziati tra loro e sparsi nell'agglomerato di articoli.

[5] Utilizzando le parole dalla libreria TextWiller github.com/livioivil/TextWiller

[6] e' una matrice che ha sulle colonne le singole parole e sulle righe 0 o 1 che indicano se la parola e' presente nel documento alla riga i oppure no e se' presente piu' volte nella cella e' memorizzato il numero di volte che compare in quel documento, calcolato attraverso $sklearn.feature\_extraction.text.CountVectorizer$

Successivamente si e' provato attraverso la stessa pipeline, applicando in piu' Latent Dirichlet Allocation (LDA), impostando come parametri n_components [7] a 6 e learning_decay[8] al valore di default.



(a) Vista normale (b) Vista ravvicinata

Figura 2: Riduzione della matrice da forma (2400, 4640) a forma (2400, 3).

A differenza della Figura 1 in questa figura (esplorazione interattiva [9]) si vede come i diversi temi sono raggruppati in piccoli cluster sparsi in maniera piu' o meno uniforme lungo i tre assi, inoltre articoli dello stesso tema nella Figura 2 sembrano essere meno distanti tra loro. Sempre dalla Figura 2 si evince come un albero potrebbe essere una soluzione accettabile per il problema di classificazione.

3.3 Prove

Si e' deciso di mettere a confronto 3 configurazioni attraverso le quali si vede come l'accuratezza delle predizioni varia al variare della dimensione del training set. Nella seguente tabella si evincono le 3 configurazioni.

Pipeline \ **Trasformation**	LDA-12	LDA-48	Term Frequency
T.D Matrix	✓	✓	✓
LDA	✓	✓	x
Classifier	✓	✓	✓

Tabella: configurazioni LDA-12 e LDA-48, indicano i modelli fittati con 12 e 48 componenti.

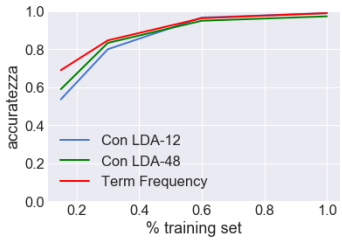


Figura 3: Performace modelli di classificazione sul test set in funzione della dimensione del training set.

[7] numero di topic latenti che LDA dovrebbe individuare

[8] parametro usato per regolare il learning rate, che si consiglia imposare nel range (0.5, 1] per garantire la convergenza asintotica.

[9] https://plot.ly/create/?fid=cristi.gutzu:5&fid=cristi.gutzu:6

4 Risultati finali

Come riportato in Tabella 1, le rappresentazioni LDA-12 e Term Frequency hanno pari accuratezza, e sono i modelli migliori tra quelli che abbiamo stimato, classificando gli articoli dell'insieme di test. Siccome l'accuratezza, che misura la qualita' complessiva del modello, risulta molto buona, ci si e'

Rappresentazione	Accuratezza
Dummy	15.0 %
LDA-12	**99.0 %**
LDA-48	97.3 %
Term Frequency	**99.0 %**

Tabella 1: Risultati utilizzando i diversi modelli di raprresentazione con l'intero training set.

chiesto come il classificatore si comporta anche per le singole classi, per verificarlo abbiamo calcolato ulteriori misure per i due modelli migliori.

Categoria	precision	recall	f1	support
Cronaca	0.96	1.00	0.98	45
Cultura	1.00	1.00	1.00	57
Economia	1.00	0.98	0.99	44
Politica	1.00	0.96	0.98	54
Sport	1.00	1.00	1.00	49
Tech	0.98	1.00	0.99	51
micro avg	0.99	0.99	0.99	300
macro avg	0.99	0.99	0.99	300
weighted avg	0.99	0.99	0.99	300

Tabella 2: Report metriche sul classificatore stimato con LDA-12.

Categoria	precision	recall	f1	support
Cronaca	1.00	1.00	1.00	45
Cultura	1.00	1.00	1.00	57
Economia	0.96	0.98	0.97	44
Politica	1.00	0.96	0.98	54
Sport	1.00	1.00	1.00	49
Tech	0.98	1.00	0.99	51
micro avg	0.99	0.99	0.99	300
macro avg	0.99	0.99	0.99	300
weighted avg	0.99	0.99	0.99	300

Tabella 3: Report metriche sul classificatore stimato con Term Frequency.

Si osserva dalle tabelle che le metriche: precisione, richiamo, f1; sono molto alte per ogni classe, cosa che conferma l'accuratezza elevata ottenuta sul trainig set.

5 Conclusioni

Dai risultati delle analisi si vede che il modello basato sulla rappresentazione con LDA e' comparabile al modello basato sulla rappresentazione con Term Frequency dal punto di vista dell'accuratezza,

tuttavia il modello con LDA riduce lo spazio delle variabili riducendo i costi computazionali nel processo di classificazione.

Riferimenti bibliografici

1. David m. Blei, Andrew Y, Ng, and Michael I. Jordan, Latent Dirichlet Allocation, 2003
2. Hastie T., 2016, Introduction to Statistical Learning , Decision Trees
3. Ceron, Curini, Iacus, 2014, Social Media e Sentiment Analysis.