Università degli Studi Milano-Bicocca
Scuola di Scienze
Corso di Laurea Magistrale in Data Science

Text Mining Project

Possiamo individuare le fake news?

Gruppo di Lavoro
Alberto Carlone matr. 726894
Davide Miori matr. 813692
Alice Ondei matr. 826399

Anno Accademico 2021-2022

Introduzione

Il XXI secolo è stato caratterizzato dall'esplosione dei social network i quali, favoriti dalla digitalizzazione globale, hanno permesso una comunicazione più rapida tra gli individui. Proprio per questa ragione, tale strumento viene spesso utilizzato per fare propaganda politica o come mezzo di informazione; si è bombardati giornalmente da numerose notizie, molte delle quali però con scarsa attendibilità. Il fenomeno che ne consegue è la proliferazione delle *fake news*, alcune delle quali facilmente identificabili, altre invece richiedono una buona conoscenza di fondo relativamente al tema in questione.
Strumenti come il Text Mining, congiuntamente all'utilizzo di modelli di Machine Learning, possono avere un ruolo cruciale nell'aiutare il processo di identificazione di tali notizie fuorvianti.

L'elaborato si presuppone dunque l'obiettivo di classificare un'informazione come "Real" o "Fake", applicando tecniche di Text Mining per la gestione dei dati testuali e modelli di Machine Learning per la classificazione dei documenti in una delle due classi.
Trattandosi di notizie di carattere politico, in un secondo momento si cercherà di capire se le notizie rilasciate dai maggiori politici statunitensi possano essere riconosciute e suddivise in base alla fonte originaria. Verrà quindi svolta una Cluster Analysis con l'intento di raggruppare gli articoli provenienti dai dieci politici con il maggior numero di notizie pubblicate.

Dati e Pre-Processing

Il dataset esaminato è costituito da circa diecimila articoli ottenuti dal sito Politifact (https://www.politifact.com/), un portale di fact-checking di notizie relative alla politica statunitense, caratterizzati da sei attributi:

- **News_Headline**: informazioni relative al contenuto.
- **Link_Of_News**: url per recuperare l'articolo.
- **Source**: autore del testo originale.
- **Stated_On**: data di pubblicazione del testo.
- **Date**: data di pubblicazione su Politifact.
- **Label**: classe di interesse divisa in sei categorie True, Mostly-True, Half-True, Barely-True, False e Pants-Fire.

Il primo passo del pre-processing è stato recuperare il corpo degli articoli dall'url nella colonna **Link_Of_News** mediante scraping, eseguito grazie alla libreria *BeautifulSoup*: sfruttando il codice HTML delle url è possibile individuare il tag specifico in cui è situato l'intero testo. Per facilitare lo scraping, avente una durata totale di diverse ore, è stata effettuata una suddivisione del dataset in 10 blocchi, successivamente unificati per ottenere il dataset definitivo. A seguire, siccome negli articoli pubblicati su Politifact sono presenti alcune frasi in cui viene specificata esplicitamente la classe di appartenenza (l'attributo **Label**), si è deciso di rimuovere queste dal testo prima di procedere. Vista la numerosità del dato, sono stati anche rimossi gli articoli aventi meno di cinque frasi.
Successivamente, ci si è occupati di eliminare link, emoticons, simboli speciali e caratteri appartenenti ad alfabeti diversi da quello latino oltre ad esplicitare le forme abbreviate inglesi (ad

esempio *aren't* con *are not* oppure *cause* con *because*). Alcuni caratteri speciali sono stati individuati successivamente alla prima iterazione del pre-processing.
Infine, sono state applicate la rimozione delle *stopword* e della punteggiatura. La tokenizzazione e la lemmatizzazione, eseguita con POS Tagging, vengono applicate tramite una funzione appositamente costruita da attribuire al parametro *tokenizer* della funzione *TfidfVectorizer* in modo da uniformare il testo e renderlo funzionale alla rappresentazione selezionata. All'interno della funzione creata, il POS Tagging viene effettuato sui tre elementi principali di una frase: nomi (*nouns*, tag che iniziano con NN), verbi (*verbs*, tag che iniziano con VB) e aggettivi (*adjective*, tag che iniziano con JJ).

Come anticipato, la variabile **Label** contiene sei classi[1] sulle quali è possibile svolgere una classificazione multi-classe. Tuttavia, poiché i **Label** originali sono molto sbilanciati, si è deciso di assegnare le sei categorie ad uno dei due poli - rappresentati da Real e Fake - ottenendo quindi un problema di classificazione binaria.

Prima di allenare il classificatore sono state estratte le feature dal testo mediante la creazione della matrice TF-IDF (Term Frequency-Inverse Document Frequency), della quale sono state mantenute solo le parole con una df (document frequency) compresa tra 0,01 e 0,5, in modo da escludere parole troppo comuni e troppo rare (e quindi poco caratterizzanti).
In seguito, si è deciso di aggiungere un ulteriore set di feature con l'intento di migliorare le performance del classificatore. I nuovi attributi sono: il conteggio dei caratteri, il conteggio delle parole, il rapporto dei due precedenti, il conteggio della punteggiatura e il conteggio delle parole con iniziale maiuscola.

Text Representation

Per la Text Representation è stata scelta una rappresentazione tramite TF-IDF, in cui ogni documento è rappresentato da un vettore di parole con pesi differenti.
Questi pesi vengono calcolati attraverso una funzione composta dal prodotto di due elementi:

- Term Frequency, calcolata come $\frac{tf_{t,d}}{max_{ti} tf_{ti,d}}$, e
- Inverse Document Frequency $log(\frac{N}{df_t})$

Con t il termine di cui si sta calcolando il peso, d il documento di riferimento, $tf_{t,d}$ la frequenza del termine t all'interno del documento d, df_t il numero di documenti d in cui compare il termine t e N il numero totale di documenti.

---

1 https://www.politifact.com/article/2018/feb/12/principles-truth-o-meter-politifacts-methodology-i/#Truth-O-Meter%20ratings

Con questa rappresentazione si ottengono dei pesi che aumentano sia al crescere della frequenza di un determinato termine all'interno del documento, sia con la rarità della sua presenza tra tutti i documenti.

In una fase preliminare è stata valutata anche la possibilità di utilizzare una rappresentazione tramite Word2Vec (o Doc2Vec) ma a causa della non particolare voluminosità del dato sia in termini di documenti (poco meno di 10.000) sia in termini di parole uniche (circa 60.000 dopo il pre-processing) le performance ottenute non risultavano soddisfacenti.

Text Classification

Per la fase di classificazione sono stati selezionati due modelli: la Regressione Logistica e il Random Forest.
Prima dell'allenamento il dataset è stato suddiviso in training e test set conservando per il primo l'80% delle osservazioni. Inoltre, la divisione è avvenuta stratificando per la variabile risposta **Label** quindi mantenendo invariata la proporzione di Real e Fake.
Per quanto riguarda le variabili esterne alla matrice TF-IDF è stata applicata una normalizzazione al fine di avere valori che giacessero in uno spazio dimensionale facilmente confrontabile. L'ultimo passo prima dell'allenamento è stata la Feature Selection mediante chi-quadrato, la quale ha permesso di selezionare le feature che meglio spiegassero la classe di interesse. Per l'esattezza si è passati da 3380 a 228 variabili applicando un p-value limite di 0,90.
Conclusa la fase di preparazione sono stati allenati i due modelli sul training set per poi verificarne le performance sul test. A tal fine sono state utilizzate le metriche più comuni tra cui l'accuratezza (siccome la classe non è sbilanciata), la curva Precision-Recall e l'AUC ottenuta rappresentando la curva ROC. Inoltre, è stato considerato anche il tempo di esecuzione.
La Figura 1 mostra i parametri impostati e gli score relativi ai due modelli utilizzati. Il Random Forest mostra performance migliori in termini di accuracy e delle altre metriche considerate in media rispetto alle due classi.

```
LogisticRegression(class_weight='balanced', penalty='l1', random_state=42,
                   solver='liblinear')
Train Accuracy      0.75901
Test Accuracy       0.73829
Test Precision      0.75713
Test Recall         0.73829
Test F1             0.73777
Test F2             0.73573
Classification report:
              precision    recall  f1-score   support

        FAKE    0.83099   0.65677   0.73368      1078
        REAL    0.66727   0.83747   0.74274       886

    accuracy                        0.73829      1964
   macro avg    0.74913   0.74712   0.73821      1964
weighted avg    0.75713   0.73829   0.73777      1964
```

```
RandomForestClassifier(class_weight='balanced', n_estimators=500,
                       random_state=42)
Train Accuracy      1.00000
Test Accuracy       0.75967
Test Precision      0.76220
Test Recall         0.75967
Test F1             0.76018
Test F2             0.75969
Classification report:
              precision    recall  f1-score   support

        FAKE    0.79823   0.75232   0.77459      1078
        REAL    0.71835   0.76862   0.74264       886

    accuracy                        0.75967      1964
   macro avg    0.75829   0.76047   0.75862      1964
weighted avg    0.76220   0.75967   0.76018      1964
```

Figura 1: Score relativi alla Regressione Logistica (a sinistra) e Random Forest (a destra).

Anche osservando le curve di ROC e la curva Precision-Recall nella Figura 2 il classificatore Random Forest risulta, seppur leggermente, migliore rispetto alla Regressione Logistica.
A livello computazionale il primo algoritmo impiega 314 millisecondi per prevedere le 1964 osservazioni del test mentre il Random Forest impiega 2,64 secondi. In conclusione, quest'ultimo performa leggermente meglio in termini previsionali impiegando un tempo lievemente maggiore nella previsione. Quest'ultimo fattore, tuttavia, non presenta una differenza sostanziale al punto da spingerci a scegliere la Regressione Logistica come algoritmo migliore. Il Random Forest dunque rappresenta il modello più performante tra i due selezionati.

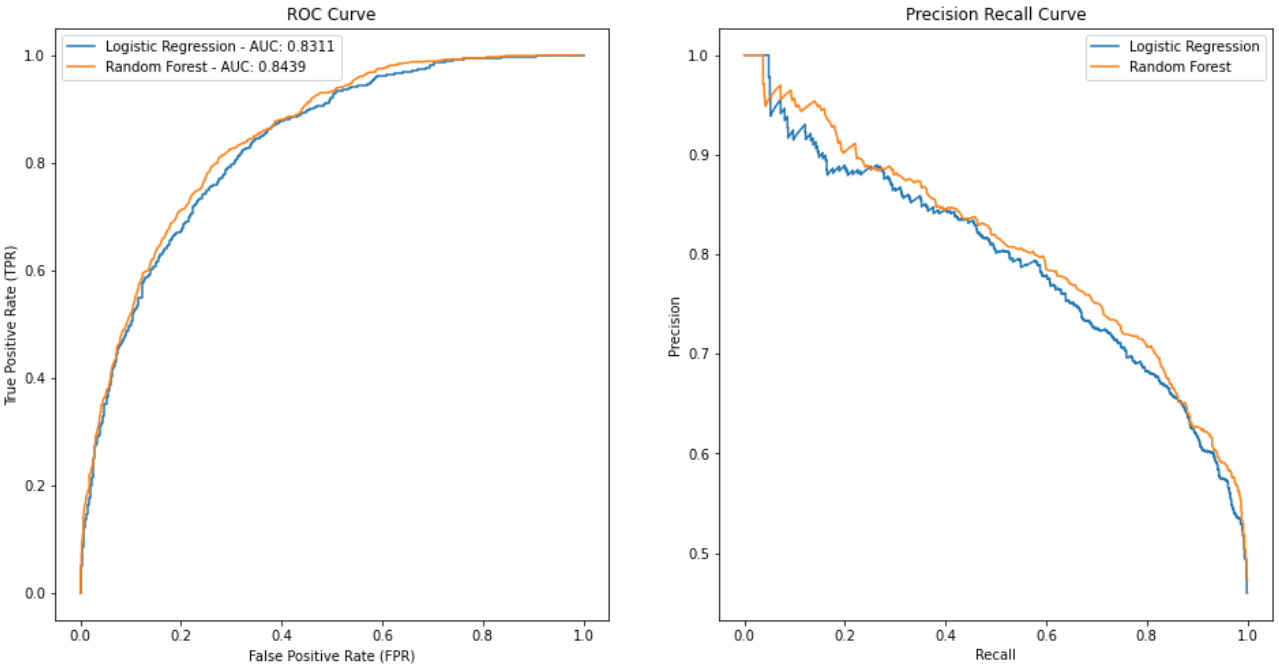

Figura 2: ROC curve (a sinistra) e Precision-Recall Curve (a destra).

Text Clustering

Come già specificato precedentemente, tra le variabili presenti nel dataset proposto rientra anche l'attributo **Source**, esso indica l'autore della notizia originaria. Tra queste fonti, come facile immaginare, figurano diversi personaggi politici; per questo si è deciso di cercare di capire se fosse possibile suddividere in cluster gli articoli proprio sulla base dell'autore di riferimento.
Dato che alcune fonti presentano un esiguo numero di articoli, si è deciso di contare il numero per ognuna di esse per poi selezionare i 10 politici con il numero di articoli rilasciati più alto. Tra questi troviamo: Joe Biden, Jeb Bush, Rick Scott, Marco Rubio, Scott Walker, Ted Cruz, Barack Obama, Bernie Sanders, Hillary Clinton e, ovviamente, Donald Trump. Si evidenzia che tra le principali fonti su cui viene effettuato un *fact-checking* sono presenti i principali social network, questi non vengono considerati in quanto fonti molto eterogenee e difficilmente suddivisibili.
Così come per la fase di classificazione, è stata creata la matrice TF-IDF ed è stata successivamente svolta una Feature Selection mediante chi-quadrato applicando un p-value limite di 0,99.
Tramite la prima rappresentazione tramite t-SNE possiamo notare, osservando la Figura 3a, come la maggior parte delle **Source** sia distinguibile, salvo qualche eccezione dovuta ad esempio alla preponderanza di documenti relativi a Donald Trump rispetto alle altre fonti.

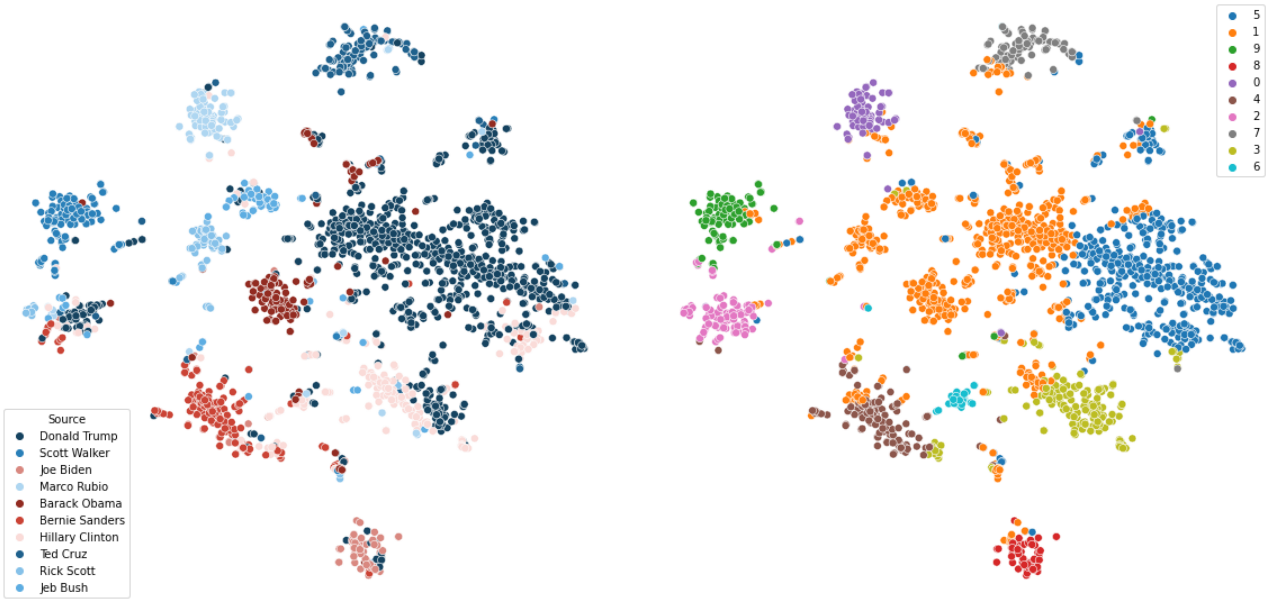

Figura 3: a. Cluster generati da t-SNE; b. Cluster generati dal K-Means con le coordinate del t-SNE.

Per l'effettivo clustering degli articoli è stato utilizzato il metodo K-Means. Dalla Figura 3b, ottenuta sfruttando le stesse coordinate di t-SNE, si può vedere anche in questo caso come, seppur con qualche eccezione, risultino sufficientemente evidenti diversi cluster già individuati precedentemente. Si nota inoltre come alcuni cluster, ad esempio il numero 2, non trovino corrispondenza con una fonte precisa.

Per approfondire l'assegnazione dei cluster è stato realizzato un istogramma, rappresentato dalla Figura 4, per ogni fonte considerata: come già intuibile tramite il confronto grafico tra i due scatterplot, diverse fonti sono state attribuite al cluster 1 invece che a cluster separati (Barack Obama, Rick Scott, Jeb Bush) mentre fonti come Joe Biden e Scott Walker sono state assegnate in maniera pressoché omogenea ad un singolo cluster.

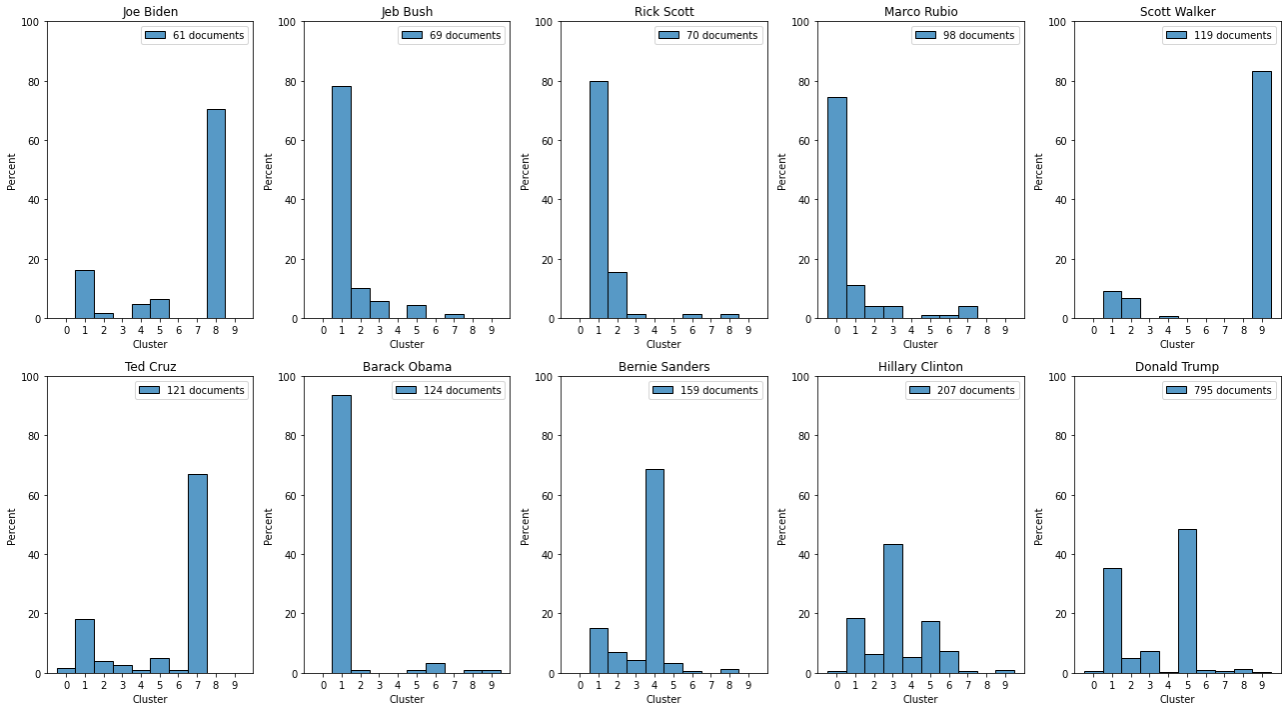

Figura 4: Divisione percentuale della fonte rispetto ai cluster.

Per vagliare le impressioni tratte dai grafici si confrontano le etichette generate dal cluster con quelle reali; per la valutazione sono state prese in considerazione due metriche differenti: il Rand Index, utilizzato per confrontare la similarità tra due cluster e in qualche modo analogo all'accuratezza utilizzata per le classificazioni, e il Fowlkes-Mallows Index, corrispondente alla media geometrica tra la *precision* e la *recall*. Sono stati ottenuti, rispettivamente, un Rand Index pari a 0,75 e un Fowlkes-Mallows index pari a 0,42. La differenza tra i due score è da imputare all'assegnazione del cluster 1 a diverse fonti, soprattutto al numero elevato di documenti attribuiti a Donald Trump, che portano ad avere un punteggio di *precision* e *recall* molto basso.

Conclusione

Considerata anche la difficile situazione che ci troviamo a vivere da ormai due anni, è sempre più evidente l'importanza di avere accesso a informazioni affidabili e veritiere; ne consegue la necessità di sapersi orientare all'interno di questo mare magnum di notizie. Si è quindi voluto addestrare un classificatore che potesse essere a supporto dell'identificazione di notizie fasulle; ciò è stato possibile grazie all'applicazione congiunta di tecniche di Text Mining e di Machine Learning. Dopo una fase di pulizia e pre-processing dei testi, il modello più performante è risultato il Random Forest, il quale raggiunge un'accuratezza di 0,76. Si ottengono poi prestazioni analoghe anche nel secondo task, il quale aveva come obiettivo quello di separare in gruppi, mediante una Cluster Analysis, articoli provenienti dai 10 politici con numero di news pubblicate maggiore. Le previsioni del K-Means, confrontate con la ground truth a disposizione, portano ad un valore di Rand Index (paragonabile all'accuratezza in ambito di classificazione) pari a 0,75.

Questi risultati non sono ottimi ma si possono considerare soddisfacenti, dal momento che 3 osservazioni su 4 vengono identificate in maniera corretta. Per migliorare le prestazioni è possibile effettuare una più oculata e specializzata operazione di pre-processing del testo, è possibile inoltre utilizzare classificatori più complessi, facendo eventualmente utilizzo di tecniche di apprendimento profondo. Un'altra possibilità è quella di estrarre feature più valide e efficaci nello spiegare la variabile risposta. Come sempre, una maggior presenza di dati può portare a performance migliori, ciò permetterebbe anche la possibilità di rivolgersi a tecniche di embedding, applicabili soltanto in presenza di un maggior numero di documenti.