Relazione progetto Big Data e Business Intelligence

Francesca Stefano matricola 306826

Linguaggio: Python

Dataset: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

PROGETTAZIONE E STRUTTURA:

L'obiettivo del progetto è, a partire dai dataset di fake e real news, sviluppare un modello in grado di stabilire (con un certo grado di accuracy) se una notizia è vera oppure falsa.

Il task da eseguire è un task di classificazione supervisionato. Come detto l'obiettivo è predire se una notizia è reale oppure falsa. Le performance saranno misurate andando a valutare l'accuracy, matrice di confusione e (attraverso classification_report di sklearn.metrics)precision, recall, f1-score.

Per prima cosa, dopo aver scaricato i file csv, ne ho preso visione analizzando le varie features.

In entrambi i csv vi erano 4 colonne:

-TITLE

-TEXT

-SUBJECT

-DATE

Poiché l'obiettivo finale del progetto è quello di predire se una notizia è reale o meno ho aggiunto in entrambi una colonna Target. Questo perché dopo aver unito i due dataset era necessario sapere la natura delle notizie. Prima di unire i due dataset li ho però esplorati singolarmente, in particolar modo ho ritenuto utile avere informazioni generali sul dataset attraverso datafake.info() e quindi sapere se ci fossero valori nulli (e nel caso positivo prevedere tecniche per gestirli) e infine ottenere gli argomenti del 'subject'

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23481 entries, 0 to 23480
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    23481 non-null  object
 1   text     23481 non-null  object
 2   subject  23481 non-null  object
 3   date     23481 non-null  object
 4   Target   23481 non-null  int64 
dtypes: int64(1), object(4)
memory usage: 917.4+ KB
None
```

```
subject
News               9050
politics           6841
left-news          4459
Government News    1570
US_News             783
Middle-east         778
dtype: int64
title      0
text       0
subject    0
date       0
Target     0
dtype: int64
```

Dall'analisi di quanto riportato in figura è possibile capire la dimensione del dataset, la memoria occupata, ma (informazioni ben più importanti) anche la presenza o meno di valori nulli (e nel mio caso non c'è ne sono) e i vari topics degli articoli: News, politics, left-news, Government News, US-News.

In egual modo ho proceduto per le real news, ottenendo quanto segue:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21417 entries, 0 to 21416
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   title    21417 non-null  object
 1   text     21417 non-null  object
 2   subject  21417 non-null  object
 3   date     21417 non-null  object
 4   Target   21417 non-null  int64
dtypes: int64(1), object(4)
memory usage: 836.7+ KB
```

```
None
subject
politicsNews    11272
worldnews       10145
dtype: int64
title      0
text       0
subject    0
date       0
Target     0
dtype: int64
```

Anche in questo caso non sono presenti valori nulli e i subject sono invece diversi da quelli del dataset precedente. Infatti, in questo caso come subject vi sono solamente politicsNews (mentre in fake vi era politics) e worldnews.

Prima di unire i due dataset mi sono fatta stampare da ciascuno 15 campioni (scelti randomicamente) nella colonna dei 'text'.

```
13021    Does anyone even care that this American man w...
6206     If you ve ever been on Twitter then you know t...
21515    Because #BlackCopKillersLivesMatter right?Supp...
8114     Ever since the end of the ridiculous exercise ...
6209     Republicans plan to use laws about voter ID to...
14745    Is anyone else concerned that the Left was abl...
483      If Donald Trump hadn t proven himself to be th...
17881     Last week, President Trump made a public anno...
18770    The poll below is why people shouldn t trust p...
17266    WAS IT HILLARY OR THE STATE DEPARTMENT? We kno...
21279    For the umpteenth time, Obama takes the opport...
22650     21st Century Wire asks HAVE YOUR SHOUT: Apple...
14795    Have you ever noticed how DHS Director Jeh Joh...
16122    Wow! This is really epic! Paris Dennard nails ...
15047    Rep. Trey Gowdy, chairman of the Benghazi Sele...
Name: text, dtype: object
```

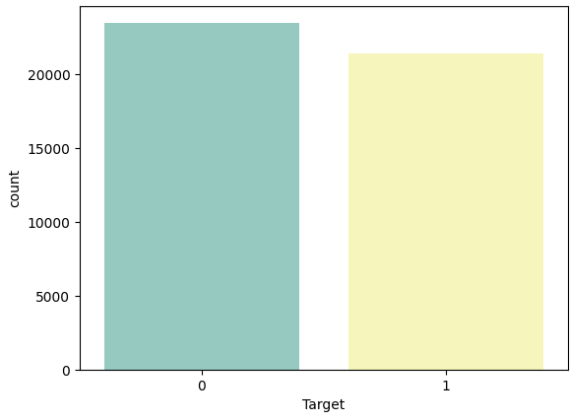
FAKE

```
8699     CLEVELAND (Reuters) - City police on bicycles ...
19798    BEIJING (Reuters) - The youth wing of China s ...
12529    ANKARA (Reuters) - Russia and Turkey agree tha...
13611    TEGUCIGALPA (Reuters) - The Honduran president...
21042    LIMA (Reuters) - Teachers in Peru started retu...
16309    CAIRO (Reuters) - Egyptian security forces kil...
15795    SYDNEY (Reuters) - The U.N. High Commissioner ...
5496     WASHINGTON (Reuters) - When Japanese first lad...
19316    NEW YORK (Reuters) - U.S. President Donald Tru...
13340    ATHENS (Reuters) - Tayyip Erdogan will travel ...
1601     WASHINGTON (Reuters) - U.S. Senator Elizabeth ...
3846     WASHINGTON (Reuters) - The U.S. Senate Intelli...
15494    SRINAGAR, India (Reuters) - Indian soldiers ki...
17204    TOKYO/UNITED NATIONS (Reuters) - The United St...
172      WASHINGTON (Reuters) - Two of President Donald...
Name: text, dtype: object
```

REAL

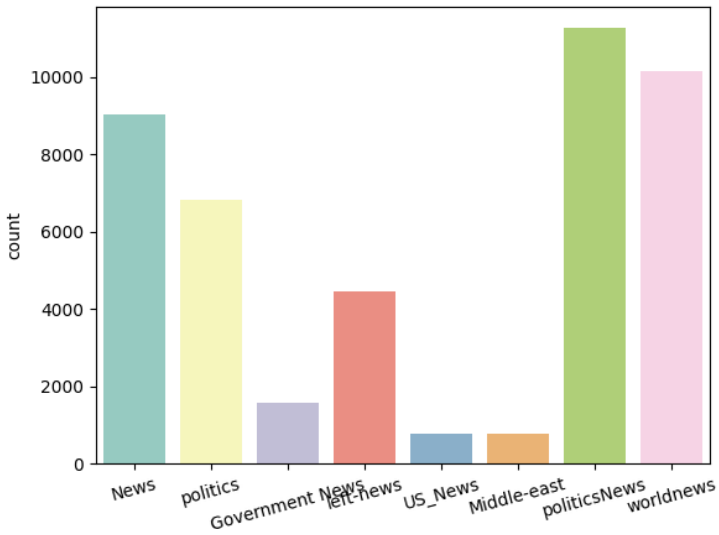
Analizzando questi campioni si può facilmente intuire anche un'altra differenza fra i due tipi di news: le real presentano nel text la formula CITY (Reuters)

Dopodiché ho concatenato i due dataset in uno unico denominandolo df e per esplorarlo ho deciso di visualizzare in base al target il count delle real e fake news ottenendo quanto segue:

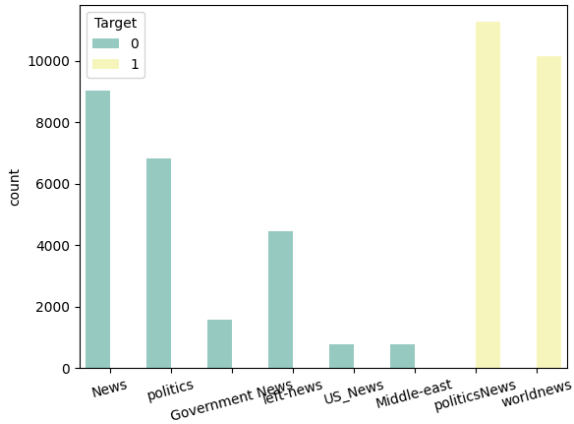


Ho deciso inoltre di visualizzare i valori dei subject (dei dataset uniti) in modo da sapere come le news si distribuivano in base all'oggetto del articolo e di graficarlo.

```
[15 rows x 5 columns]
subject
politicsNews       11272
worldnews          10145
News                9050
politics            6841
left-news           4459
Government News     1570
US_News              783
Middle-east          778
dtype: int64
```



Questo istogramma mi ha permesso di capire come, in base al subject, le varie news si distribuiscono, e indagando la relazione fra il target - subject, ho visualizzato attraverso gli istogrammi quanto intuito prima: in questo dataset se si è di fronte a una politicalNews o una worldnews, allora sicuramente queste saranno real (e il contrario per gli altri subject)



Finita questa prima parte di esplorazione dei dati sono passata alla parte di pre-processing.

Per prima cosa mi sono fatta stampare una lista di stopwords inglesi (e ne ho stampate a video 100 per prenderne visione) e dopodiché le ho rimosse dal mio text. Ho poi creato una funzione il cui obiettivo era rendere tutti i caratteri minuscoli, eliminare gli URL , sostituire a un due spazi bianchi solamente uno, parentesi e punti di domanda e di esclamazione (da un punto di vista sintattico in realtà i punti di domanda come quelli di esclamazione hanno un significato ben preciso in un frase, anzi ne possono modificare il significato, e quindi per questo motivo in un primo momento non li avevo eliminati, ma ricercando su articoli di NLP in molti suggerivano di eliminarli).Una volta creata questa funzione ho creato una nuova colonna nel mio dataset che ho chiamato sempre 'text' che altro non è l'unione della colonna 'title' con 'text' e a questa nuova colonna ho applicato la mia funzione.

Dopodiché ho eliminato la colonna ‘title’ (perché comunque già inclusa nella nuova colonna ‘text’), ‘subject’ (per la corrispondenza 1:1 fra il subject e le real/fake news) e infine la colonna ‘date’ (anche in questo caso all’inizio non ero certa di eliminare la colonna, in quanto la frequenza di pubblicazione di una rivista/giornale può essere un dato interessante per il progetto).

	text	Target
12108	offensive satanic display allowed next to nati...	0
44626	factbox key issues in the nafta renegotiations...	1
14025	daughter of sunni muslim george clooney wife o...	0
39412	australia's famed uluru outback monolith to be...	1
16348	emotional trump endorsement from former fbi as...	0
2002	rep ted lieu torches evil donald trump over h...	0
41643	tokyo governor koike no need for big change in...	1
18337	watch one woman reports the weather in sweden ...	0
14665	muslim brotherhood affiliate invited to obama ...	0
21340	illegal immigrants caught squatting in deploye...	0
7259	microsoft forced to remove racist sexist robo...	0
29706	democrats take aim at mnuchin as confirmation ...	1
29516	trump preparing executive orders to reduce u s...	1
13216	nfl star delivers tough message about extermin...	0
28307	trump's cut to flood map program could trigger...	1

<class 'list'>

Esempio di 15 campioni del dataset dopo aver applicato la funzione e aver eliminato le colonne

A questo punto ho preceduto definendo X e y (y è il valore target che il modello deve predire, mentre per X ho creato una lista di liste in cui ogni lista è una lista di parole).
L’ obiettivo ora era riuscire a “trasformare le parole in vettori”; per fare questo ho utilizzato Word2Vec.
Per fare questo ho scelto una serie di parametri che sarebbero poi stati inseriti nel mio modello:
sentences=X
vector_size=100
window=10
min_count=1
Nel mio caso quindi ogni parola sarà convertita con un vector_size pari a 100.
Ho poi posto la massima distanza fra la parola corrente e quella predetta in un’unica frase pari a 10(di default è posto pari a 5).

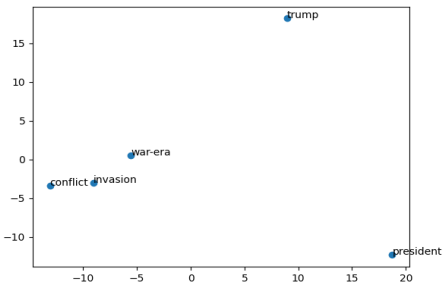
Affinché ci sia tale conversione è necessario aspettare circa un minuto e dopodiché ho ottenuto che:

```
Wait...
Word2Vec<vocab=339037, vector_size=100, alpha=0.025>
339037
```

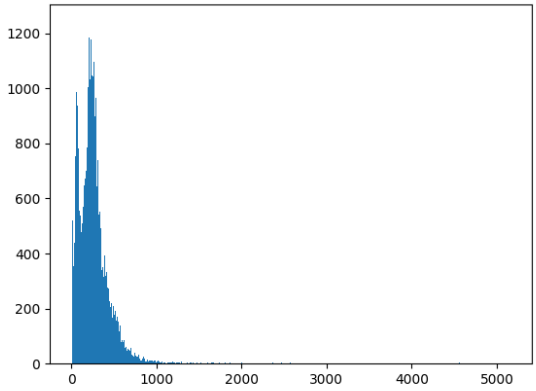
Il mio vocabolario ha 339037elementi.

Al fine di esplorare Word2Vec ho testato funzionalità quali most_similar e la cosine_similarity. Come esempi ho cercato le parole più simili a trump(president-elect, pig-pile, etc), war(war-era, invasion,conflict) e twitter(tweets, hashtag, etc)

Ho poi attraverso la PCA creato un set di parole di prova per fare visualizzazione di word2Vec



Ho provato anche a tokenizzare per ottenere una sequenza di numeri assegnati univocamente alle parole e ho graficato la distribuzione delle parole nel mio dataset.



A questo punto ho preceduto facendo lo split tra training e test set testando vari modelli alla ricerca di quello che potesse fornirmi un'accuracy migliore

Per primo ho provato con la logist regression, ottenendo quanto segue:

```
0.38455827765404604
Confusion Matrix logistic regression:
 [[5556 1489]
 [3691 2734]]
Accuracy logistic regression :
 61.54417223459539
Report logistic regression :
              precision    recall  f1-score   support

           0       0.60      0.79      0.68      7045
           1       0.65      0.43      0.51      6425

    accuracy                           0.62     13470
   macro avg       0.62      0.61      0.60     13470
weighted avg       0.62      0.62      0.60     13470
```

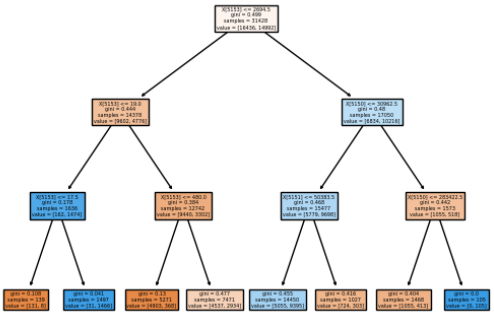
Come modello non mi permette di ottenere grandi prestazioni, in quanto l'accuracy risulta essere del 62% e anche analizzando la matrice di confusione si evince che vengono predetti molti falsi positivi. Inoltre, ho calcolato anche l'errore medio assoluto e questo risulta essere di 0.38

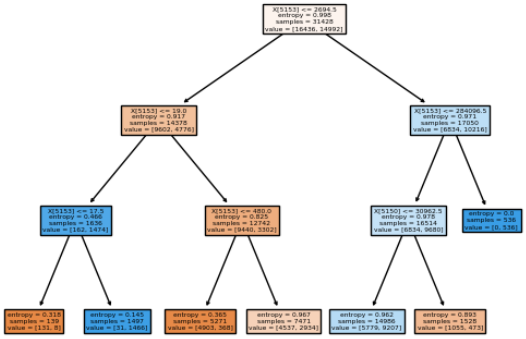
Ho proceduto con il decision tree sia con, come misura di purezza, gini ed entropy (anche se di fatto la scelta di una o dell'altra non modifica in modo determinante il risultato)

```
Report :
              precision    recall  f1-score   support

           0       0.74      0.65      0.69      7045
           1       0.66      0.75      0.70      6425

    accuracy                           0.70     13470
   macro avg       0.70      0.70      0.69     13470
weighted avg       0.70      0.70      0.69     13470
```

Ho ottenuto una precisione del 70%, per cui andando a migliorare i parametri potrei ottenere risultati ancora migliori.

In questo caso avevo posto `max_depth=3,min_samples_leaf=5`





Ho quindi modificato questi due parametri ponendo `max_depth=7,min_samples_leaf=5.`
Ottenendo risultati migliori e facendo vari test ho infine posto i due valori a, rispettivamente, 15 e 17. Ottenendo quanto segue:

```
Accuracy :
 83.34818114328137
Report :
              precision    recall  f1-score   support

           0       0.89      0.78      0.83      7045
           1       0.79      0.89      0.84      6425

    accuracy                           0.83     13470
   macro avg       0.84      0.84      0.83     13470
weighted avg       0.84      0.83      0.83     13470
```

(Le immagini relative ai decision tree aventi come max_depth e min_sample_leaf pari a 15 e 17 si possono trovare nella cartella del progetto, poiché la loro dimensione non consentiva visualizzarle al meglio all' interno di questo documento)

Ho poi pensato di provare ad utilizzare una rete neurale

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 64)                329920

 dense_1 (Dense)             (None, 64)                4160

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 334,145
Trainable params: 334,145
Non-trainable params: 0
```

A livello di parametri della rete ho posto come funzione di attivazione dell'output layer la funzione sigmoide che predice i valori 0-1 perché in base alla mia colonna Target ciò che la rete neurale deve predire è proprio uno di questi due valori.

```
[[6526  519]
 [5541  884]]
              precision    recall  f1-score   support

           0       0.54      0.93      0.68      7045
           1       0.63      0.14      0.23      6425

    accuracy                           0.55     13470
   macro avg       0.59      0.53      0.45     13470
weighted avg       0.58      0.55      0.46     13470
```

Come si può evincere da quanto qui riportato il grado di accuracy della rete è del 55%, in particolar modo si può riscontrare(dalla matrice di confusione) una predizione dei falsi positivi molto elevata.

Per evitare l'overfitting ho inserito nel codice l'EarlyStopping, ponendo patience=5, ossia il numero di epoche senza alcun miglioramento dopo le quali l'allenamento verrà interrotto, e min_delta = 0.001, ossia la variazione minima nella quantità monitorata per qualificarsi come miglioramento, quindi una variazione assoluta inferiore a min_delta non verrà conteggiata come miglioramento.
Infine ho realizzato un codice per ricercare i migliori iperparametri da porre nella mia rete neurale (il codice inerente a questa parte risulta essere commentato, in quanto l'esecuzione richiede più di 3 ore e quindi una volta che mi ha fornito il risultato l' ho commentato per motivi computazionali).