

Intelligenza artificiale per l'analisi delle news :

Dati, inferenza e apprendimento automatico

Membri del gruppo di lavoro:

- Antonio Pio Mennuni (758395) , a.mennuni8@studenti.uniba.it
- Danilo Santo (758454) , d.santo3@studenti.uniba.it

Repository GitHub del progetto:

- <URL repo associato, contenente il materiale completo>

Indice

Introduzione 1

Dati utilizzati 2

Descrizione dei dati	2.1
Sommario	2.2

Pre-processing dei dati 3

Introduzione	3.1
Operazioni offline	3.1.1
Operazioni online	3.1.2
Aggiunta nuova feature	3.2
Sommario	3.3

Creazione della Knowledge Base 4

introduzione	4.1
Fatti e regole	4.2
Applicazione delle regole	4.3
Estrazione delle nuove feature dalle regole	4.4
Sommario	4.5

Apprendimento supervisionato 5

introduzione	5.1
Scelta dei modelli di apprendimento	5.2
Analisi delle prestazioni e confronto tra modelli	5.3
Sommario	5.4

Apprendimento non supervisionato 6

Scelta dei modelli di apprendimento	6.1
Analisi dei risultati	6.2
Sommario	6.3

Introduzione 1

Il caso di studio in oggetto si pone come obiettivo lo svolgimento di una analisi su news di politica e attualità, ai fini di poter identificare possibili correlazioni tra le feature delle news (come la lunghezza del testo, tipo di notizia o presenza di parole che possano far pensare ad una fake news) e successivamente costruire un KBS (knowledge-based system) che aiuti nella discriminazione tra fake news e true news. Utilizzando tecniche di apprendimento supervisionato e inferenze basate su logica, il caso di studio (come già detto in precedenza) mira a creare un KBS in grado di poter estrapolare feature aggiuntive, per i dati in possesso al fine di poter ampliare la quantità informativa disponibile e poter utilizzare modelli predittivi a partire dai dati posseduti. Viene anche impiegato l'utilizzo di tecniche di apprendimento non supervisionato, con l'obiettivo di identificare pattern di similarità tra i dati che possano aiutarci nella discriminazione delle news. Inoltre, l'utilizzo di queste tecniche non supervisionate viene anche impiegato per trovare altre informazioni da utilizzare nella creazione della base di conoscenza e nell'addestramento dei modelli predittivi supervisionati (infatti, si vedrà nel seguito che la scelta di ricavare la feature **"num_words"** in fase di pre-processing non è stata una scelta casuale. Si rimanda alla lettura del **capitolo 6**). Ad oggi, in un mondo ipertecnologico dove siamo sommersi di notizie continuamente, saper discriminare tra una fake news e una true news è importantissimo, considerando che la manipolazione delle news è uno strumento potentissimo per veicolare i pensieri e le opinioni delle persone. Ai fini di una più completa e corretta comprensione del caso di studio, si consiglia al lettore di eseguire l'intero software avviabile dal file **main.py** , presente nella directory **/python** .

Dati utilizzati 2

Descrizione dei dati 2.1

I dati impiegati all'interno del caso di studio provengono da due fonti principali e sono interamente in lingua inglese. La prima fonte è **figshare.com**, da cui è stato scaricato un dataset in formato **csv** chiamato "**Fake_Real.csv**" e contenente circa 35.000 fake news e true news di politica e attualità. La seconda fonte è **kaggle.com**, da cui sono stati scaricati due dataset in formato **csv** chiamati rispettivamente, **Fake.csv** contenente 23.502 fake news e **True.csv** contenente 21.417 true news. Gli ultimi due dataset citati contengono ovviamente news di politica e attualità. Alla fine di questo paragrafo questo sono presenti i rispettivi link dei siti web da cui sono stati scaricati i dataset utilizzati. Passiamo ora alla descrizione dei dataset utilizzati.

Fonte 1:

https://figshare.com/articles/dataset/Fake_and_True_News_Dataset/13325198?utm_source=chatgpt.com&file=25670870

Fonte 2:

<https://www.kaggle.com/datasets/clmentbisailon/fake-and-real-news-dataset?resource=download&select=True.csv>

- **Fake_Real.csv**

Come già citato in precedenza, questo dataset contiene 23.502 fake news di politica e attualità. Esso è costituito dai seguenti campi mostrati nella tabella sottostante:

Campo	Significato
id	ID della notizia
text	Testo della notizia
subject	Tipo di notizia (politica o attualità)
tof	Indica se la notizia è fake o real

Di seguito un'immagine del file aperto in Excel :

A1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
id,text,subject,target																				
0,new york	reuters us environmental group sierra club asked environmental protection agency's inspector general investigate whether agency's head scott pruitt violated internal policies said believe carbon dioxide																			
1,washington	reuters us air force asked industry friday proposals replace minuteman iii intercontinental ballistic missile nuclear cruise missile military moves ahead costly modernization aging atomic weapons systems																			
2,saturday	paul ryan posted photo instagram photo mass selfie large group capitol hill interns long social media commentators noticed something wrong photo in post ryan wrote think sets record number capitol hill interns																			
3,america	keeps waiting word hillary indicted obama waiting right moment fbi allowed job put hillary behind bars joe biden elizabeth warren standing wings one thing know sure corrupt administration anything could																			
4,religion	peace ht weasel zippers, left-news, Fake																			
5,leave	obama politicize money think past presidents money obama made money like postage stamp made political racial really expect pitiful man abraham lincoln person depicted 5 bill much longer the treasury department																			
6,knew	donald trump would nightmare supreme court nominee judge neil gorsuch trying present sort legal scholar enforce law fairly regardless personal views however past coming back haunt turns misogynistic base																			
7,like	soldiers oden vigilante group reported february national framed vigilante group tired waiting inept government protect citizens dangerous criminals arriving boat nations intention assimilating country decided ir																			
8,kiev	reuters outgoing us vice president joe biden called monday sanctions russia stay urging world stand against coercion aggression against president elect donald trump mooted ending measures possible deal moscow sp																			
9,detroit	reuters quality problems prompted two michigan governor rick snyder's top lawyers urge flint moved back detroit water system months decision draw water supply flint river according emails released friday																			
10,please	rt identify disgusting nyu professor slandering police nazi sympathizers blue lives matter picture twitter com 2wgrnj6md democrats trump young democrats trump february 3 2017, politics, Fake																			
11,valetta	reuters police believe bomb killed prominent journalist malta attached beneath car triggered remotely government spokeswoman said thursday giving first details investigation daphne caruana galizia reno																			
12,email	sent april 15 2011 upstanding attorney general eric holder wrote issa idiot cronies never gave damn happening thousands mexicans killed guns country want reality cripple atf suck gun lobby politics worst ma																			
13,rome	reuters brother man killed two women knife outside marseille train station last week probably foreign fighter syria iraq italian investigators said monday italian police arrested 25-year-old tunisian anis hannac																			
14,british	pop singer rebecca ferguson daughter white british woman jamaican father says invited perform donald trump inauguration response may rsvp president elect team expecting the former uk x factor runner up s																			
15,rep	maxine waters dca physical embodiment antitrump resistance congress openly called orange one impeached multiple occasions rails national television every opportunity latest attack donald comes via favori																			
16,washington	reuters us republican senator mike lee utah said monday support legislation overhaul us tax system leaving two republicans undecided bill approaches final senate vote week after just finished reading																			
17,washington	reuters despite president donald trump full-throated support saudi arabia united states appears signaling desire riyadh take cautious approach regional power struggle iran experts say trump administr																			
18,washington	reuters democratic presidential candidate hillary clinton praised peace deal colombia rebel group farc statement thursday after ceas president ian ll ensure united states remains partner process people																			

- **True.csv**

Come già citato in precedenza, questo dataset contiene 21.417 true news di politica e attualità. Esso è costituito dai seguenti campi mostrati nella tabella sottostante:

Campo	Significato
title	Titolo della news
text	Testo della news
subject	Tipo di news (attualità o politica)
date	Data di pubblicazione della news

Di seguito un'immagine del file aperto in Excel :

A1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	title, text, subject, date																			
2	As U.S. budget fight looms, Republicans flip their fiscal script, "WASHINGTON (Reuters) - The head of a conservative Republican faction in the U.S. Congress, who voted this month for a huge expansion of the national d																			
3	U.S. military to accept transgender recruits on Monday: Pentagon, "WASHINGTON (Reuters) - Transgender people will be allowed for the first time to enlist in the U.S. military starting on Monday as ordered by federal co																			
4	Senior U.S. Republican senator: 'Let Mr. Mueller do his job', "WASHINGTON (Reuters) - The special counsel investigation of links between Russia and President Trump's 2016 election campaign should continue withc																			
5	FBI Russia probe helped by Australian diplomat tip-off: NYT, "WASHINGTON (Reuters) - Trump campaign adviser George Papadopoulos told an Australian diplomat in May 2016 that Russia had political dirt on Democr																			
6	Trump wants Postal Service to charge 'much more' for Amazon shipments, "SEATTLE/WASHINGTON (Reuters) - President Donald Trump called on the U.S. Postal Service on Friday to charge "much more" to ship pac																			
7	White House, Congress prepare for talks on spending, immigration, "WEST PALM BEACH, Fla./WASHINGTON (Reuters) - The White House said on Friday it was set to kick off talks next week with Republican and Democra																			
8	Trump says Russia probe will be fair, but timeline unclear: NYT, "WEST PALM BEACH, Fla (Reuters) - President Donald Trump said on Thursday he believes he will be fairly treated in a special counsel investigation into R																			
9	Factbox: Trump on Twitter (Dec 29) - Approval rating, Amazon, "The following statements were posted to the verified Twitter accounts of U.S. President Donald Trump, @realDonaldTrump and @POTUS. The opinions e																			
10	Trump on Twitter (Dec 28) - Global Warming, "The following statements were posted to the verified Twitter accounts of U.S. President Donald Trump, @realDonaldTrump and @POTUS. The opinions expressed are his c																			
11	Alabama official to certify Senator-elect Jones today despite challenge: CNN, "WASHINGTON (Reuters) - Alabama Secretary of State John Merrill said he will certify Democratic Senator-elect Doug Jones as winner on Thu																			
12	Jones certified U.S. Senate winner despite Moore challenge, "Reuters) - Alabama officials on Thursday certified Democrat Doug Jones the winner of the state's U.S. Senate race, after a state judge denied a challenge																			
13	New York governor questions the constitutionality of federal tax overhaul, "NEW YORK/WASHINGTON (Reuters) - The new U.S. tax code targets high-tax states and may be unconstitutional, New York Governor Andrew Cu																			
14	Factbox: Trump on Twitter (Dec 28) - Vanity Fair, Hillary Clinton, "The following statements were posted to the verified Twitter accounts of U.S. President Donald Trump, @realDonaldTrump and @POTUS. The opinions																			
15	Trump on Twitter (Dec 27) - Trump, Iraq, Syria, "The following statements were posted to the verified Twitter accounts of U.S. President Donald Trump, @realDonaldTrump and @POTUS. The opinions expressed are his																			
16	Man says he delivered manure to Mnuchin to protest new U.S. tax law, " (In Dec. 25 story, in second paragraph, corrects name of Strong's employer to Mental Health Department, not Public Health Department.) By Bei																			
17	Virginia officials postpone lottery drawing to decide tied statehouse election, "Reuters) - A lottery drawing to settle a tied Virginia legislative race that could shift the statehouse balance of power has been indefinitely p																			
18	U.S. lawmakers question businessman at 2016 Trump Tower meeting: sources, "WASHINGTON (Reuters) - A Georgian-American businessman who met then-Miss Universe pageant owner Donald Trump in 2013, has bei																			
19	Trump on Twitter (Dec 26) - Hillary Clinton, Tax Cut Bill, "The following statements were posted to the verified Twitter accounts of U.S. President Donald Trump, @realDonaldTrump and @POTUS. The opinions expresse																			
20	U.S. appeals court rejects challenge to Trump voter fraud panel, "Reuters) - A U.S. appeals court in Washington on Tuesday upheld a lower court's decision to allow President Donald Trump's commission investi																			

Sommario 2.1

Le feature presenti nei vari dataset sembrano adatte allo scopo del caso di studio (se pur poche). Una volta raccolti i vari dataset questi devono essere compattati per ottenere un unico dataset in modo da avere tutto ciò che serve per passare alle fasi successive. Prima però è necessaria un'attenta fase di pre-processing dei dati, poiché questi devono essere puliti e formattati correttamente visto che devono essere successivamente usati per formalizzare la conoscenza in possesso. Si precisa che il numero di fake news e true news nel complesso è bilanciato e lo è anche dopo la fase di pre-processing dei dati.

Pre-processing dei dati 3

Introduzione 3.1

Nella fase di pre-processing dei dati vengono eseguite tutte le operazioni necessarie per la pulizia di dati ridondanti, non utili per lo scopo del caso di studio o mancanti, e le operazioni necessarie per l'integrazione dei dati provenienti dai tre diversi dataset. Questa fase risulta essenziale poiché, come detto in precedenza, i dati costituiscono la conoscenza in possesso che verrà formalizzata nelle fasi successive per permettere il corretto sviluppo del KBS. Le librerie python utilizzate negli script di questa fase sono le seguenti : **pandas** (per i dataframe), **skit-learn** (per la generazione dei pesi tf-idf) e **numpy** (per la gestione degli array, matrici e calcoli scientifici.)

Passiamo ora all'analisi di tutte le varie operazioni.

Operazioni offline 3.1.1

Le operazioni che stanno per essere descritte sono state effettuate offline (attraverso l'utilizzo di appositi script python), cioè non vengono eseguite durante l'esecuzione dell'intero programma, avviabile attraverso l'esecuzione dello script **main.py** presente nella directory **/python** . Di conseguenza, i dataset **True.csv** e **Fake.csv** non sono Disponibili nella directory **/csv** , in cui saranno presenti solo i dataset relativi alla fase di pre-processing online e alle altre fasi. La scelta di non integrare queste operazioni, e i rispettivi dataset, all'interno del programma finale è data dal fatto che l'esecuzione avrebbe portato (insieme a tutte le altre operazioni) ad un dispendio di tempo e spazio

molto più elevato di quello attuale. Tutto ciò quindi, avrebbe rallentato e appesantito molto l'esecuzione del programma.

-

Per prima cosa si è deciso di non considerare i campi **“title”** e **“date”** presenti nei dataset **Fake.csv** e **True.csv**, poiché considerati non utili agli scopi del caso di studio.

-

È stato inserito nei dataset **Fake.csv** e **True.csv** il campo **“id”**, poiché presente solo nel dataset **Fake_Real.csv**. Questo campo risulta estremamente utile non solo nelle fasi di pre-processing per la pulizia dei dati, ma anche nelle successive fasi, visto che aiuta a sapere con quale riga del dataset si sta lavorando.

-

I dataset **True.csv** e **Fake.csv** non hanno il campo **“tof”** che specifica se la news è fake o no. Alla luce di questo, per l'integrazione con il dataset **Fake_Real.csv**, è stato necessario inserire questo campo in entrambi i dataset, con l'accortezza di settare il valore **'True'** al dataset delle true news e il valore **'Fake'** al dataset delle fake news. Dopo l'esecuzione di queste operazioni tutti e tre i dataset hanno gli stessi campi, che si elencano sotto :

Id	text	subject	Tof
----	------	---------	-----

-

Adesso i dataset hanno gli stessi campi, perciò possono essere integrati. E' stato deciso di prendere circa 8.000 news dai dataset **True.csv** e **Fake.csv** per un totale di circa 16.000 news e integrarle nel dataset **Fake_Real.csv** che da questo momento diventa l'unico dataset a disposizione nel caso di studio.

-

Come ultima operazione di pre-processing offline è stata eseguita un'operazione di ricerca ed eliminazione delle news con un testo uguale o molto simile, poiché questo avrebbe potuto portare nelle fasi di addestramento dei modelli predittivi, ad un'intersezione tra training set e test set. Se questo accade ci sono forti probabilità che un modello predittivo vada in **overfitting**, non riuscendo più a discriminare bene tra fake news e true news nella fase di test o nella messa in esercizio con altri dataset mai visti prima dal modello. Per evitare ciò, è stato impiegato l'uso di uno script python che utilizza diverse librerie apposite. Questo script riesce a trovare testi di news simili o uguali sfruttando la similarità del coseno. Prima però, è necessario trasformare i testi delle news in matrici di pesi **tf-idf(term frequency-invers document frequency)**. Dopo di che si sfrutta la similarità del coseno con una soglia di similarità del 50%, per individuare ed eliminare dal dataset le coppie di news con un testo almeno simile per il 50%. Questo ha ridotto abbastanza il numero di news. Adesso, il dataset **Fake_Real.csv**, che ha subito una prima fase di pulizia, si trova nella directory **/csv** sotto lo stesso nome. Non rimane che pulirlo da eventuali news con valori mancanti o nulli, operazione eseguita nella fase di pre-processing online.

Operazioni online 3.1.2

Le operazioni che stanno per essere descritte fanno parte della fase di pre-processing online dei dati (anche essa eseguita con appositi script python), cioè, quella fase di pre-processing che viene eseguita nel programma finale, avviabile con l'esecuzione dello script **main.py** che ricordiamo essere presente nella directory **/python** . Obiettivo di questa fase è pulire il dataset

Fake_Real.csv , già precedentemente arricchito e pulito nella fase di pre-processing offline, da eventuali dati mancanti o nulli nei vari campi. Per fare questo, vengono impiegati i seguenti script python :

last_field_cleaner.py

second_field_cleaner.py

first_field_cleaner.py

third_field_cleaner.py

Questi script python, disponibili nella directory **/python/pre_processing** si occupano rispettivamente di identificare ed eliminare quelle news del dataset che hanno valore nullo o mancante nel campo **"id"**, **"text"**, **"subject"** e **"tof"**

cioè nei campi del dataset. Ognuno degli script crea un nuovo file csv con solo le news valide. Quindi ad esempio, lo script **last_field_cleaner.py** creerà nella directory **/csv** un nuovo file csv denominato **dataset_no_last_empty_field.csv** che conterrà solo le news che non hanno valore mancante o nullo nel campo **“tof”**. Ora il dataset **Fake_Real.csv** è stato totalmente pulito da news simili o uguali e da news con valori mancanti o nulli. Tutte le news valide, che ammontano a 16412, sono state esportate in un nuovo file csv denominato **clean_dataset.py** disponibile nella directory **/csv** . Queste operazioni hanno molto ridotto il numero di news del dataset, ma erano operazioni necessarie per avere un dataset pronto all’uso nelle fasi successive. Inoltre, come già citato in precedenza, il dataset anche dopo le fasi di pre-processing h un numero di notizie fake e true bilanciato. Per verificare questo si utilizza uno script python, disponibile nella directory **/python/pre_processing** e denominato **count_true_false.py** .

A seguire sono allegate delle immagini relative alla fase di pre-processing online e all’esecuzione dello script **count_true_false.py** .

```
5
6 try:
7     # Carica il file CSV
8     data = pd.read_csv(file_path)
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS C:\Users\Antonio\Desktop\progetto_icon> & C:/Users/Antonio/AppData/Local/Microsoft/WindowsApps/python3.13/python.exe C:/Users/Antonio/Desktop/progetto_icon/python/pre_processing/count_true_false.py
Righe con 'true': 7517
Righe con 'fake': 8891
Totale righe nel file (esclusa la prima): 16408
PS C:\Users\Antonio\Desktop\progetto_icon>

Risultato dell'esecuzione dello script **count_true_false.py**

```
4 file_path = "csv/dataset_no_second_empty_field.csv"
5 output_file = "csv/dataset_no_first_empty_field.csv"
6
7 try:
8     # Caricare il CSV
9     data = pd.read_csv(file_path)
10    # Rimuovere le righe con campi vuoti all'inizio
11    data = data[data[0].str.strip() != '']
12    # Salvare il file pulito
13    data.to_csv(output_file, index=False)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS C:\Users\Antonio\Desktop\progetto_icon> & C:/Users/Antonio/AppData/Local/Microsoft/WindowsApps/python3.13/python.exe C:/Users/Antonio/Desktop/progetto_icon/python/pre_processing/first_field_cleaner.py
File pulito salvato in: csv/dataset_no_first_empty_field.csv
Rimosse 0 righe non valide.
PS C:\Users\Antonio\Desktop\progetto_icon>

Risultato dell'esecuzione dello script **first_field_cleaner.py**

```
17     valid_rows = data.iloc[1:][~(second_field.isna() | second_field.str.strip().eq(""))]
18
19     # Salva le righe valide in un nuovo file
20     valid_rows.to_csv(output_file_cleaned, index=False)
21
22     print(f"Totale righe nel file pre rimozione (esclusa la prima): {len(data) - 1}")
23     print(f"Rimosse {len(data) - 1 - len(valid_rows)} righe con secondo campo nullo o vuoto.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS Python

PS C:\Users\Antonio\Desktop\progetto_icon> & C:/Users/Antonio/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/User
ogetto_icon/python/pre_processing/sd_field_cleaner.py
Totale righe nel file pre rimozione (esclusa la prima): 44895
Rimosse 632 righe con secondo campo nullo o vuoto.
File pulito salvato in: csv/dataset_no_second_empty_field.csv
PS C:\Users\Antonio\Desktop\progetto_icon>

Risultato dell'esecuzione dello script **sd_field_cleaner.py**

```
19     # Salva le righe valide in un nuovo file
20     valid_rows.to_csv(output_file_cleaned, index=False)
21
22     print(f"Rimosse {len(data) - len(valid_rows)} righe con terzo campo nullo o vuoto.")
23     print(f"File pulito salvato in: {output_file_cleaned}")
24
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS C:\Users\Antonio\Desktop\progetto_icon> & C:/Users/Antonio/AppData/Local/Microsoft/WindowsApps/python3.1
ogetto_icon/python/pre_processing/third_field_cleaner.py
Rimosse 36 righe con terzo campo nullo o vuoto.
File pulito salvato in: csv/clean_dataset.csv
PS C:\Users\Antonio\Desktop\progetto_icon>

Risultato dell'esecuzione dello script **third_field_cleaner.py**

Aggiunta nuova feature 3.2

Questa è una delle due fasi in cui vengono aggiunte feature (campi) al dataset. Si ricorda che da ora in poi il dataset a cui fare riferimento è **clean_dataset.py** disponibile nella directory **/csv** . In questa fase si sfrutta uno script python, presente nella directory **/python/pre_processing** e denominato **num_words_for_texts.py** , per contare il numero di parole del testo per ogni news. Questa nuova informazione verrà poi aggiunta come campo al dataset. Quindi dopo l'esecuzione dello script il dataset avrà un nuovo campo chiamato

“num_words” che specifica il numero di parole presenti nel campo **“text”** .

L'intero dataset, con il campo aggiuntivo e avvalorato per ogni riga, è stato esportato in un nuovo file csv denominato

clean_dataset_with_num_words.csv presente nella directory **/csv** . Come già citato in precedenza, la scelta del calcolo e della creazione di questo nuovo campo non è stata casuale. Attraverso un algoritmo di apprendimento non supervisionato chiamato **k-means** è stato possibile ricavare che c'è una correlazione tra la lunghezza del testo delle news, il tipo di news (con cui ci si riferisce tramite il campo **“subject”**) e la veridicità della stessa. In particolare, si è visto che la correlazione tra lunghezza del testo e veridicità è molto forte. In seguito, questa nuova feature verrà sfruttata dai modelli predittivi per

discriminare meglio tra fake news e true news. Questo motiva abbondantemente la scelta del calcolo di questa feature. Per una trattazione più approfondita si rimanda alla lettura del **capitolo 6** di questa Documentazione.

Sommario 3.3

Dopo le numerose fasi di pre_processing effettuate e il calcolo e l'aggiunta della nuova feature, si possiede un dataset ampio, pulito e pronto per essere utilizzato dal KBS che formalizzerà quella conoscenza e la sfrutterà per estrapolare ulteriori informazioni da tramutare in feature ingegnerizzate che aiuteranno ulteriormente i modelli predittivi che verranno addestrati nelle fasi successive. Tutto ciò verrà effettuato attraverso l'uso del linguaggio di programmazione logica **ProLog**, che sfrutta meccanismi di inferenza basata su logica del primo ordine, cioè ricava nuove informazioni a partire da quelle Esistenti.

Creazione della Knowledge Base 4

Introduzione 4.1

-

La Knowledge Base è stata scritta in linguaggio Prolog e consiste di una serie di fatti riguardanti la sola classe di individui, ovvero le news, (rappresentate dalle loro feature che sono gli individui veri e propri della kb) e di una serie di regole per inferire conoscenza a partire dai fatti esistenti. Sarà quindi impiegata la libreria “**pyswip**” che permette di interfacciarsi con **SWI-ProLog**, un popolare interprete Prolog. Questa libreria è utile per integrare la logica dichiarativa di Prolog in applicazioni Python, combinando il meglio dei due linguaggi: la potenza di elaborazione logica di Prolog con la flessibilità e la praticità di Python. Oltre a **pyswip** , sono state anche utilizzate anche le librerie **pandas** e **os**. Per questioni organizzative si è scelto di tenere fatti e regole (quindi la KB) in due file separati. I due file in questione sono **rules.pl** e **facts.pl** , entrambi disponibili nella directory **progetto_icon** .

-

Per svolgere gli opportuni calcoli sulla KB. Gli individui selezionati per il caso di studio non sono altro che le feature del dataset **clean_dataset.csv** che ricordiamo essere le seguenti :

Id , text , subject , tof , num_words

le relazioni individuate inizialmente si riducono ad una sola:

News(Id,text,subject,tof,num_words)

mentre successivamente verranno estrapolate e create due nuovi tipi di relazioni, le quali diverranno poi feature del dataset (si veda nel proseguo). La scelta di codificare le news come relazioni e le loro feature come individui è dovuta al fatto che le news (cioè l'insieme delle loro feature) sono viste come fatti a sé stanti ed è molto più conveniente lavorare sulle feature come individui, visto che il cuore di una notizia sta proprio nelle sue caratteristiche.

Fatti e regole 4.2

Si specifica che tutti i predicati definiti in corso d'opera sono predicati dinamici. La scelta è motivata dal fatto che usare un predicato dinamico in ProLog permette di poter aggiungere e rimuovere ulteriori fatti alla kb a runtime, senza dover far ripartire l'esecuzione e quindi la rilettura del file

facts.pl . Questa scelta si è rivelata fondamentale per la corretta esecuzione degli script, visto che a runtime vengono aggiunte diverse migliaia di fatti riguardanti i predicati generati attraverso le regole di inferenza.

-

La KB si compone inizialmente di un solo tipo di fatto che è il seguente :

News(Id,text,subject,tof,num_words)

E che rappresenta una notizia a sé stante, cioè un qualcosa di vero e assodato, come già detto in precedenza. Successivamente all'applicazione delle regole sulla KB, verranno creati due nuovi tipi di fatti, sugli individui esistenti, che verranno menzionati più tardi. I fatti iniziali relativi alle news vengono creati nello script **generate_kb.py** , presente nella directory **python/prolog_code** , e caricati poi nel file ProLog **fact.pl** . Lo script python non fa altro che leggere le news dal dataset **clean_dataset_with_num_words.csv** e creare in **fact.pl** un fatto di tipo news per ogni notizia letta. Ovviamente prima di ciò lo script definisce i predicati dinamici come prime righe del file **fact.pl**

Di seguito immagini relative al file **fact.pl** e **generate_kb.py**

```

news(155, "athens reuters leading member greece deadly guerrilla group november 17 began twoday leave prison thursday pa
news(156, "dr ben carson earned dubious honor refuting idea brain surgeon equals smart clearly least beyond operating ta
news(157, "gay community feel triumphating actor bill clinton pal kevin spacey coming gay man excuse distraction attempti
news(159, "women saudi arabia given right drive many conservative men believe separation sexes believe women smart enoug
news(162, "wow really serious problem antiamerican jackhole president", "Government News", "Fake", 7).
news(164, "dubai reuters saudi arabia saturday suspended dialogue qatar accusing distorting facts report phone call lead
news(169, "donald trump continues showered adulation conservatives one rightwing radio host seriously regretting role ri
news(171, "SKOPJE Reuters A senior official of Macedonia s secret police was sentenced to 18 months in prison for destr
news(172, "hillary may gotten away lying public decades underestimated time around enormous power chicago thugs valerie
news(173, "secret military suffered obama years us military hindered overbearing inexperienced white house president bar
news(175, "It s not even a slight exaggeration to say that when the Supreme Court decided that rightwing religious bigot
news(176, "seoul reuters north korea used chinese-made trucks display missiles massive military parade last week accordin
news(178, "MEXICO CITY Reuters A key committee in Mexico's Senate on Thursday ratified a deal with the United States t
news(179, "well last day eight years obama presidency eight years saw swarm friends acquaintances regurgitate asinine co
news(181, "los angeles reuters caitlyn jenner joined list celebrities considering running political office united states
news(185, "remember racist black panther punk started ericshpeppardchallenge involved stomping american flag yeah well ad
news(186, "white supremacists celebrating donald trump rise top republican party kkk leader david duke bragged radiospea
news(187, "An Islamist group with secretive compounds around the US is arming up in anticipation of raids by the incomin
news(188, "gadsden ala reuters us senate race rocked allegations sexual misconduct republican candidate roy moore women
news(189, "cedar rapids iowa reuters us president donald trump siege washington defended record raucous political rally
news(190, "moscow reuters russian foreign minister sergei lavrov said wednesday main part battle islamic state syria acc
news(192, "LANGLEY Va Reuters US President Donald Trump told CIA officers on Saturday that he has great respect for the
news(194, "This is just idiotic A couple of liberal brainiacs decided to steal a Trump sign The only problem is that the
news(195, "21st century wire says tuesday president donald trump signed executive action removing legal obstacles holdin

```

Immagine relativa al file **facts.pl**

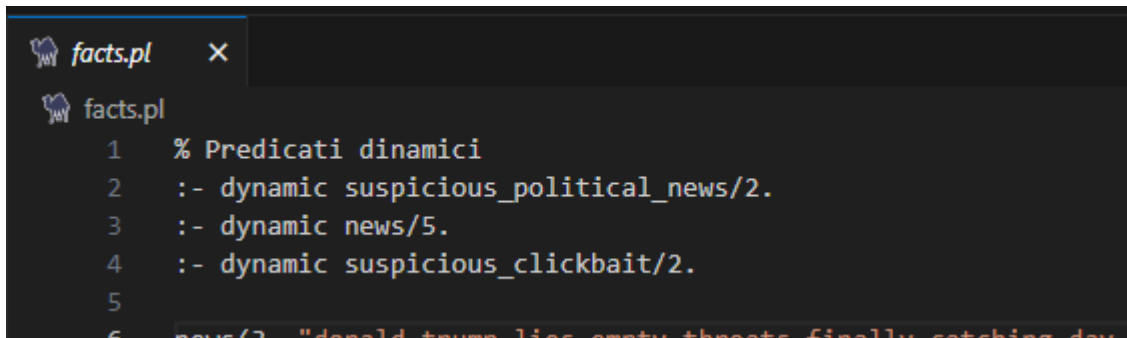
```

3
4 def create_prolog_kb(csv_file, kb_file):
5     # Leggere il dataset
6     df = pd.read_csv(csv_file)
7
8     # Creare la directory per il file di output se non esiste
9     os.makedirs(os.path.dirname(kb_file), exist_ok=True)
10
11     # Aprire il file di output per scrivere i fatti
12     with open(kb_file, 'w', encoding='ISO-8859-1') as f:
13         # Dichiarazione di predicati dinamici
14         f.write("% Predicati dinamici\n")
15         f.write(":- dynamic suspicious_political_news/2.\n")
16         f.write(":- dynamic news/5.\n")
17         f.write(":- dynamic suspicious_clickbait/2.\n\n")
18
19         #scrivi i fatti nel file prolog dei fatti
20         for _, row in df.iterrows():
21             row_id = row['id']
22             text = row['text'].replace("'", '"').replace('\n', ' ')
23             subject = row['subject'].strip()
24             tof = row['tof']
25             num_words = row['num_words']
26             fact = f"news({row_id}, \"{text}\", \"{subject}\", \"{tof}\", {num_words}).\n"
27             f.write(fact)
28
29     #crea i percorsi per il file csv e il file prolog
30     base_path = os.path.abspath(os.path.dirname(__file__))
31     csv_path = os.path.abspath(os.path.join(base_path, '..\\..', 'csv'))

```

Ln 2, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.1 (Micro

Immagine relativa al file **generate_kb.py**



```
1 % Predicati dinamici
2 :- dynamic suspicious_political_news/2.
3 :- dynamic news/5.
4 :- dynamic suspicious_clickbait/2.
5
6 news(2, "donald trump lies empty threats finally catching day")
```

Qui è possibile osservare i predicati dinamici dichiarati nel file **facts.py**

-

Nel file **rules.pl** sono state dichiarate tre regole di inferenza che permetteranno al motore ProLog di ricavare nuove informazioni a partire da quelle già esistenti nel file **facts.pl** .

Inoltre, viene anche impiegato l'uso di una lista di stringhe.

La lista è la seguente :

```
clickbait_keywords([
    "shocking", "incredible", "you won't believe", "unbelievable",
    "revealed", "exposed", "this will change everything", "what happened next",
    "the truth about", "secret", "trump", "donald", "vaccine", "covid19", "COVID19"
]).
```

Le regole utilizzate sono le seguenti :

REGOLA 1

detect_suspicious_political_news(Id) :-

news(Id, _, Subject, _, NumWords),

member(Subject, ["politics", "politicsNews"]),

NumWords > 500.

Questa regola di inferenza è stata scritta per poter identificare all'interno dei fatti di tipo News, quali fossero quelli aventi come valore di subject **"politics"** o **"politicsNews"** e come valore di num_words un numero maggiore di 500. In altre parole, la regola cerca tutte le news di politica con lunghezza del testo maggiore di 500 e ne estrae l'id. La regola quindi, identifica quelle news di politica potenzialmente fake. La stesura di questa regola ha due

motivazioni principali. La prima è che solitamente le notizie di politica sono quelle più suscettibili al fenomeno delle fake news. Basti pensare a tutte le volte in cui interi governi sono caduti o hanno rischiato di cadere per via di fake news, o a tutti i vantaggi che un politico potrebbe avere nel diffondere fake news che danneggino l'opposizione.

La seconda motivazione risiede ancora una volta nell'analisi effettuata attraverso l'algoritmo di apprendimento non supervisionato **K-means** . Questo ha rivelato che se i testi delle notizie del dataset in analisi superano le 500/600 parole, la percentuale di fake

news inizia ad aumentare di molto, fino ad arrivare a notizie con lunghezza oltre le 1000 parole che indica la presenza di innumerevoli fake news.

REGOLA 2

detect_suspicious_clickbait(Id) :-

**news(Id, Text, _, _, _), clickbait_keywords(Keywords),
member(Keyword, Keywords) ,sub_String(Text, _, _, _, Keyword).**

Questa regola di inferenza è stata scritta per poter identificare all'interno dei fatti di tipo news, quali fossero quelli aventi all'interno di text parole nella lista clickbait_keywords. In altre parole, questa regola controlla se il testo di una notizia contiene parole clickbait, cioè notizie potenzialmente false. Fatto ciò, la regola estrae gli id di queste news.

REGOLA 3

very_suspicious_news(Id) :-

**detect_suspicious_political_news(Id),
detect_suspicious_clickbait(Id).**

Questa regola racchiude la semantica delle due regole precedenti, estraendo gli id di quelle che sono news di politica sospette e contenenti anche parole clickbait nel testo. È abbastanza coerente pensare che una news possa essere potenzialmente fake se è di politica, ha una lunghezza del testo molto elevata e contiene parole clickbait.

A seguire un'immagine del file **rules.pl**

```
rules.pl
1  % Regola per identificare le notizie sospette
2  detect_suspicious_political_news(Id) :-
3      news(Id, _, Subject, _, NumWords),      % La notizia ha un id, un subject e un numero di parole
4      member(Subject, ["politics","politicsNews"]), % Il subject deve essere 'politics' o 'politicsNews'
5      NumWords > 500.                          % Il numero di parole deve essere maggiore di 500
6
7
8  % Lista di parole chiave clickbait
9  clickbait_keywords([
10     "shocking", "incredible", "you won't believe", "unbelievable",
11     "revealed", "exposed", "this will change everything", "what happened next",
12     "the truth about", "secret", "trump", "donald", "vaccine", "covid19", "COVID19"
13 ]).
14
15
16 % Regola per identificare notizie sospette per clickbait
17 detect_suspicious_clickbait(Id) :-
18     news(Id, Text, _, _, _),
19     clickbait_keywords(Keywords),
20     member(Keyword, Keywords),
21     sub_string(Text, _, _, _, Keyword).
22
23
24 % Regola che combina le regole detect_suspicious_clickbait(Id) e detect_suspicious_political_news(Id)
25 % Che si avvera se una notizia contiene parole clickbait ed è una notizia di politica sospetta
26 very_suspicious_news(Id) :-
27     detect_suspicious_political_news(Id),
28     detect_suspicious_clickbait(Id).
```

Applicazione delle regole 4.3

Emerge subito, che l'applicazione delle regole sarà utile per estrarre tutti gli id di quelle news che possono essere potenzialmente fake. Lo script che si occupa dell'applicazione delle regole della kb per l'estrapolazione di nuovi importanti fatti è denominato **apply_rules_and_update_kb.py** presente nella directory **python/prolog_code** . Questo script inizialmente, carica i fatti e le regole ProLog, presenti nei file **facts.pl** e **rules.pl**, all'interno della memoria dell'interprete ProLog di **SWI-ProLog** . Quindi viene creato un oggetto di classe ProLog e viene chiamato il metodo **consult()** passando come argomento il percorso del file specificato. Fatto ciò, viene chiamata la funzione **execute_rule(prolog,query)** che si occupa di eseguire la regola specificata dal parametro query. L'esecuzione della regola non è altro che una query ProLog che ha il nome della testa della regola che si vuole eseguire. Questa funzione viene sfruttata per eseguire le regole **detect_suspicious_political_news(Id)** e **detect_suspicious_clickbait(Id)** . Verrà chiamata due volte, una per ogni regola, e restituirà ad ogni chiamata gli id identificati dalla regola. Successivamente, gli id estratti andranno in input alla funzione **Update_prolog_facts(facts_file, detected_news_id,predicate)** che prende in input anche il percorso del file **facts.pl** e il nome del predicato della regola che ha permesso l'estrazione degli id. Questa funzione non farà altro che scrivere nel file **facts.pl** una nuova serie di fatti, con due argomenti, che specificano se una notizia, identificata nel predicato dall'id, è potenzialmente fake (politica sospetta nel caso della prima regola e clickbait nel caso della seconda). Quindi i due nuovi tipi di fatti avranno due

argomenti: l'id della notizia e un flag booleano che specifica se la notizia è sospetta o meno.

Estrazione delle nuove feature dalle regole 4.4

Una volta arricchito il file **facts.pl** con la nuova serie di fatti è facile estrarre informazioni da quei fatti per etichettare le news del dataset come potenzialmente sospette perché clickbait o perché sospette di politica. Nello script **apply_rules_and_update_kb.py** sono presenti due funzioni di esportazione dati e una funzione di merge dei dati. Queste, si occupano rispettivamente di esportare i dati dei due nuovi fatti del file **facts.pl** a dei file csv ed unire questi due csv risultanti in un unico nuovo dataset csv finale, denominato **final_dataset.csv** , con le seguenti feature :

id, text, subject, tof, num_words, suspicious_political_news, suspicious_clickbait_news
e disponibile nella directory **/csv** . Il dataset è ora arricchito con due nuove feature.

A seguire alcune immagini dei file **apply_rules_and_update_kb.py** , **final_dataset.csv** e del file **fact.pl** dopo il suo aggiornamento.

```
apply_rules_and_update_kb.py  final_dataset.csv X
csv > final_dataset.csv
1 id,text,subject,tof,num_words,suspicious_political_news,suspicious_clickbait_news
2 3,donald trump lies empty threats finally catching day hearing former fbi director james comey committees con
3 4,USA president nominated christopher wray new director fbi president announced pick tweet wednesday nominati
4 5,like garden guns today president first lady president trump spoke national rifle association convention atl
5 6,nearly 30000 police officers gathered tribute nypd police officer brian moore send message newly sworn doj
6 9,HOUSTON Reuters Venezuela s former ambassador to the United Nations Rafael Ramirez has left the United Sta
7 12,sarah palin thing mouth began moving brain way involved time managed say stuff offensive stupid stunned en
8 14,seldom life truly head scratching moments make question life sanity donald trump nominated nobel peace pri
9 15,The Internet is going crazy after photos videos and gifs have been circulated of Melania Trump with her hu
10 18,ashamed say voting donald trump black trump supporter brandon tatum delivers powerful brave message black
11 19,MUNICH Germany Reuters A German prosecutor urged judges on Tuesday to give a sentence of life imprisonmen
12 22,washington reuters us president donald trump monday demanded investigators apologize looking russian inter
13 24,Probably the most highly coveted endorsement by either Democratic Presidential candidate would be Senator
14 25,bangkok reuters 45billion plan redevelop thailand industrial east go ahead regardless whoever takes power
15 26,UNITED NATIONS Reuters The US State Department on Thursday announced more than 697 million in additional
16 28,lany czech republic reuters czech election winner andrej babis attempt form minority government shunned pa
17 29,A woman who was born in Nazi Germany says that what reminds her of Hitler more than anything else isn t Do
18 30,A federal judge in Washington DC has recused herself from a second case involving Trump dossier firm Fusio
19 31,new york reuters presidentelect donald trump housing secretarydesignate ben carson spoke entertainer talk
20 32,washington reuters top democrat senate intelligence committee said tuesday committee conduct credible inve
21 33,WASHINGTONUNITED NATIONS Reuters US President Donald Trump on Wednesday threatened to cut off financial a
22 34,nyp reported players bob mcnair employs happy comments recent nfl owners meetingmcnair sparked outrage say
23 36,washington reuters republican presidential candidate ted cruz said make major announcement 4 pm wednesday
24 37,It looks like Bill O Reilly is getting the last laugh When the Fox News Channel fired cable news giant Bil
25 38,riyadh reuters president donald trump taking air force one latest bad news pinged inboxes besieged staff r
26 39,Attention conservative men This one is for youAfter reading a pair of antiabortion letters written by men
27 40,reuters lobbyist soviet counterintelligence officer participated last year meeting senior aides us preside
```

Immagine relativa al dataset **final_dataset.csv**

```
32804 suspicious_political_news(28969, 'false').
32805 suspicious_political_news(28970, 'false').
32806 suspicious_political_news(28972, 'false').
32807 suspicious_political_news(28973, 'true').
32808 suspicious_political_news(28976, 'false').
32809 suspicious_political_news(28979, 'false').
32810 suspicious_political_news(28980, 'false').
32811 suspicious_political_news(28981, 'false').
32812 suspicious_political_news(28982, 'false').
32813 suspicious_political_news(28983, 'false').
32814 suspicious_political_news(28986, 'false').
32815 suspicious_political_news(28987, 'false').
32816 suspicious_political_news(28989, 'false').
32817 suspicious_political_news(28991, 'false').
32818 suspicious_political_news(28992, 'false').
32819 suspicious_political_news(28994, 'false').
32820 suspicious_political_news(28995, 'false').
32821 suspicious_political_news(28996, 'false').
32822 suspicious_political_news(28997, 'false').
32823 suspicious_political_news(28998, 'false').
32824 suspicious_political_news(29000, 'false').
32825 suspicious_clickbait_news(3, 'true').
32826 suspicious_clickbait_news(4, 'true').
32827 suspicious_clickbait_news(5, 'true').
32828 suspicious_clickbait_news(6, 'false').
32829 suspicious_clickbait_news(9, 'false').
32830 suspicious_clickbait_news(12, 'true').
32831 suspicious_clickbait_news(14, 'true').
32832 suspicious_clickbait_news(15, 'false').
32833 suspicious_clickbait_news(18, 'true').
32834 suspicious_clickbait_news(19, 'false').
```

Immagine relativa al file **facts.pl** dopo l'aggiornamento

```

5
6
7 # Funzione per eseguire la regola Prolog e ottenere i risultati
8 def execute_rule(prolog,query):
9     print("Esecuzione della regola...")
10    # Recupera tutte le notizie sospette (Id)
11    news = list(prolog.query(f"{query}"))
12    print("regola eseguita")
13    # Estrai gli Id delle notizie sospette
14    detected_news_id = [r['Id'] for r in news]
15
16    return detected_news_id
17
18

```

Immagine relativa alla funzione `execute_rule(prolog, query)`

```

# Funzione per aggiungere il predicato 'predicate' nel file Prolog
def update_prolog_facts(facts_file, detected_news_id,predicate):
    with open(facts_file, "a", encoding="ISO-8859-1") as f:
        with open(facts_file, "r", encoding="ISO-8859-1") as f_read:
            existing_facts = f_read.readlines()

    for line in existing_facts:
        if line.startswith("news"):
            # Estrai l'ID dalla riga del fatto
            parts = line.split(",")
            news_id = int(parts[0].split("(")[1])

            # Aggiungi il predicato 'suspicious_political_news'
            if news_id in detected_news_id:
                f.write(f"\n{predicate}({news_id}, 'true').")
            else:
                f.write(f"\n{predicate}({news_id}, 'false').")

```

Immagine relativa alla funzione `update_prolog_facts(facts_file, detected_news_id, predicate)`

Sommario 4.5

Ciò che è stato fatto in questa fase è semplicemente sfruttare il motore ProLog per estrapolare nuove informazioni che potessero diventare delle vere e proprie feature ingegnerizzate da utilizzare nell'addestramento dei modelli predittivi. Ed è proprio quello che è stato portato a termine. Ora il nuovo dataset **final_dataset.csv** ha a disposizione due nuove feature rispetto al suo predecessore **clean_dataset_with_num_words.csv** .

L'obiettivo della fase successiva sarà quello che di addestrare i modelli predittivi, sfruttando tutte le feature a disposizione da questo momento in poi.

Apprendimento supervisionato 5

Introduzione 5.1

Questa fase è il cuore del KBS che si sta sviluppando, perché è proprio l'obiettivo che si prefigge di fare questo caso di studio. Verranno addestrati, con il dataset **final_dataset.csv** , una serie di **modelli di apprendimento supervisionato** per affrontare un task di **classificazione booleana**. Questo fornirà al KBS la possibilità di discriminare al meglio possibile tra fake news e true news. Codesto è il comportamento intelligente che deve essere esibito e che fa del nostro KBS un vero e proprio software di intelligenza artificiale. Tutte le feature impiegate in questa fase (input e target) provengono ovviamente dal

dataset **final_dataset.csv** . La feature target da predire è “**tof**” che essendo booleana rende il task di predizione, un task di classificazione booleana. Le feature di input utilizzate sono tutte quelle del dataset, tranne la feature target e la feature “**id**” che agli scopi del task non risulta utile. Successivamente, bisognerà valutare le prestazioni dei modelli per poter scegliere quello che si adatta meglio al presente task di classificazione. Per l’implementazione di questi ultimi vengono impiegate principalmente le librerie python **scikit-learn** (per l’implementazione dei modelli) e **matplotlib** (per la visualizzazione dei grafici relativi alle prestazioni dei modelli) oltre che alle librerie classiche **pandas** e **numpy** già utilizzate nelle fasi precedenti. Tutti gli script python relativi a questa fase sono presenti nella directory **/python** .

Scelta dei modelli di apprendimento 5.2

Per questioni organizzative e temporali, tra i tanti modelli di apprendimento supervisionato disponibili sono stati scelti principalmente i seguenti modelli :

- 1) Naïve Bayes** (Per eseguire il task con un modello probabilistico)
- 2) Random forest** (Per eseguire il task con un modello base, ma più robusto)
- 3) K-nearest neighbors (KNN)**

Come si evince dall’elenco sottostante, sono stati utilizzati tre modelli completamente diversi l’uno dall’altro per spaziare al meglio nella valutazione finale, che ricordiamo essere atta a scegliere il modello che meglio risolve il task di classificazione.

1)

Attraverso il modello **Naïve Bayes** , addestriamo sui dati un modello probabilistico che sfrutta tutta la sua potenza utilizzando esclusivamente il testo della notizia, cioè le parole che diventano a tutti gli effetti le feature di input del modello. Inoltre, **Naïve Bayes** è particolarmente adatto a lavorare con i testi, poiché l'assunzione di indipendenza tra le parole di un testo (necessaria per il calcolo delle probabilità sulle frequenze delle parole) è molto sensata nel contesto del classificatore Bayesiano. In altri termini, assumere che le parole di un testo siano indipendenti tra loro è un'assunzione utilissima al funzionamento del modello, seppure un'assunzione molto forte.

2)

L'utilizzo di un modello come il **Random forest** non è una scelta casuale. Esso si basa sul modello di base "**Albero di decisione**" creando però una foresta di alberi di decisione, addestrati assegnando ad ognuno una porzione di training set e di feature di input. Questo ci garantisce un modello più robusto del modello di base e quindi una più robusta performance finale. Per questo modello gli **iperparametri** impostati sono il numero di alberi che compone la foresta, **n_estimators** e il numero di feature assegnate ad ogni albero per apprendere l'albero stesso, **max_features** .

3)

Il modello **KNN** è un modello completamente diverso dai due precedenti. Esso è un modello che calcola la sua predizione sulla base delle distanze tra l'esempio da classificare e i k vicini più vicini ad esso (da qui il nome del modello). Quindi non c'è una vera e propria fase di addestramento, cosa che succedeva nei due modelli precedentemente esposti. Semplicemente mantiene tutti i dati di training. Si potrebbe pensare che questo è un modello più scarso in fatto di prestazioni, visto che non ha una fase di training, ma invece risulta un ottimo modello che offre una modalità di classificazione completamente diversa e che in molti casi può tornare utile. Per questo modello l'unico **iperparametro** impostato è stato k , il numero dei vicini più vicini.

È doveroso specificare che per i valori di alcune feature non numeriche è stato necessario adottare tecniche di codifica dei dati, ai fini della corretta esecuzione dei modelli. Nel **Naïve Bayes** è stata utilizzata la codifica **bag of words** per i testi, utile ad individuare le frequenze empiriche delle parole nei vari testi. Nel **random forest** e nel **KNN** è stata utilizzata la codifica **one-hot encoding** per codificare numericamente i valori della feature **subject** che ricordiamo specificare il tipo associato alla news. Ancora nel **KNN**, è stata utilizzata la matrice di pesi **tf-idf** per codificare le parole dei testi numericamente ed associare un peso dato dal livello di discriminazione che la parola possiede nel testo a cui appartiene e nell'intero insieme dei testi. Inoltre, poiché **KNN** basa le sue predizioni sul calcolo delle distanze dei vari esempi proiettati nello spazio, è stata necessaria una ulteriore operazione sui valori numerici, derivanti dalle varie codifiche, per normalizzarli al fine di non distorcere

i risultati. In ultimo, tutti e tre i modelli adottano una codifica numerica del valore della feature target. Semplicemente, al valore Fake viene associato uno 0 e al valore True il valore 1. È stato scelto, per ogni modello, di dividere i dati del dataset **final_dataset.csv** in un 30% riservato al test set e un 70% riservato al training set. Tutti gli script dedicati ai modelli sono stati divisi in directory dedicate, ognuna relativa ad un determinato modello e contenente gli script necessari. Tutte le directory e i file sono presenti nella directory **/python** . Ogni modello ha uno script dedicato che prende in input il dataset **final_dataset.csv** , estrae i valori delle feature e quando necessario applica determinate codifiche ai valori per adattarli al contesto del modello, divide il dataset in training set e test set ed esegue tutte le istruzioni atte all'apprendimento, alla valutazione del modello e alla generazione dei grafici esplicativi per le prestazioni registrate. Si procede ora all'analisi delle prestazioni dei vari modelli.

Analisi delle prestazioni e confronto tra modelli 5.3

Prima di tutto, ogni modello dopo la fase di addestramento esegue la fase di test sul test set, producendo il valore della metrica scelta per la valutazione, cioè **l'accuratezza** . Fatto ciò, sarà anche calcolata la **curva ROC (Receiver Operating Characteristic)** , e **AUC (area sotto la curva ROC)** , che permettono di valutare la capacità di discriminazione del modello misurando il miglioramento rispetto ad un modello casuale. La curva ROC mostra le

performance di un classificatore binario ed è tracciata mettendo in relazione il tasso di veri positivi (**True Positive Rate**) e il tasso di falsi positivi (**False Positive Rate**) a vari livelli di soglia. L'AUC rappresenta l'area sotto questa curva e fornisce una misura del potere discriminante del modello. Un valore AUC di 0.5 indica una prestazione casuale, mentre un AUC di 1 indica una perfetta capacità di distinguere tra le classi. Per garantire una maggiore robustezza e affidabilità dei risultati ottenuti, è stata utilizzata la tecnica di validazione incrociata, nota come **K-Fold Cross Validation** con $k = 5$. Questo metodo ha previsto la suddivisione del test set in 5 fold e ogni fold viene utilizzato sia per l'addestramento che per la validazione. In questo modo, è stata prodotta una stima più precisa delle performance del modello, riducendo la variabilità dovuta alla casualità nella suddivisione dei dati. Durante la validazione incrociata, è stata scelta l'**accuratezza**, come metrica principale per valutare le performance dei modelli. L'accuratezza, in particolare, misura la percentuale di classificazioni corrette rispetto al totale delle classificazioni effettuate, e viene massimizzata per identificare il miglior modello. Per ciascun modello, il valore medio dell'accuratezza è stato calcolato come la media delle accuratze ottenute nella **K-Fold Cross Validation**. Inoltre, è stata calcolata la **deviazione standard** dell'accuratezza per ciascun modello per valutare la variabilità dei risultati. In questo modo è stato possibile ottenere non solo una misura della performance media, ma anche una valutazione della consistenza e della stabilità del modello.

Passiamo ora alla vera e propria analisi dei risultati, ottenuta confrontando i valori delle prestazioni ottenuti dai vari modelli nella fase di test. Denotiamo con \bar{x} l'accuratezza media della validazione incrociata e con ρ la deviazione standard dell'accuratezza media.

La seguente tabella illustra i risultati

Modello	Configurazione migliore degli iperparametri	\bar{x} e ρ dell'accuratezza	Area sotto la curva di ROC
Naïve Bayes	Default	$\bar{x} = 0.94$ e $\rho = 0.004$	0.98
Random Forest	n_estimators = 5, max_features = 'log2'	$\bar{x} = 0.98$ e $\rho = 0.005$	0.99
K-NN	k = 10	$\bar{x} = 0.95$ e $\rho = 0.003$	0.79

Prestazioni dei classificatori in termini di accuratezza media, deviazione standard e ROC AUC

I valori della seconda colonna, a partire da sinistra, sono stati scelti per garantire un corretto bilanciamento tra massimizzazione dei risultati e prevenzione dell'**overfitting**.

Valutiamo ora singolarmente i risultati ottenuti per comprendere quale modello è migliore tra quelli analizzati e se, in generale, le prestazioni del task di classificazione rispecchiano le aspettative iniziali. Di seguito, oltre ai risultati precedenti, notiamo in figura anche le relative curve ROC che forniscono informazioni rilevanti riguardanti la qualità dei risultati ottenuti e la capacità da parte del modello di generalizzare su esempi non visti.

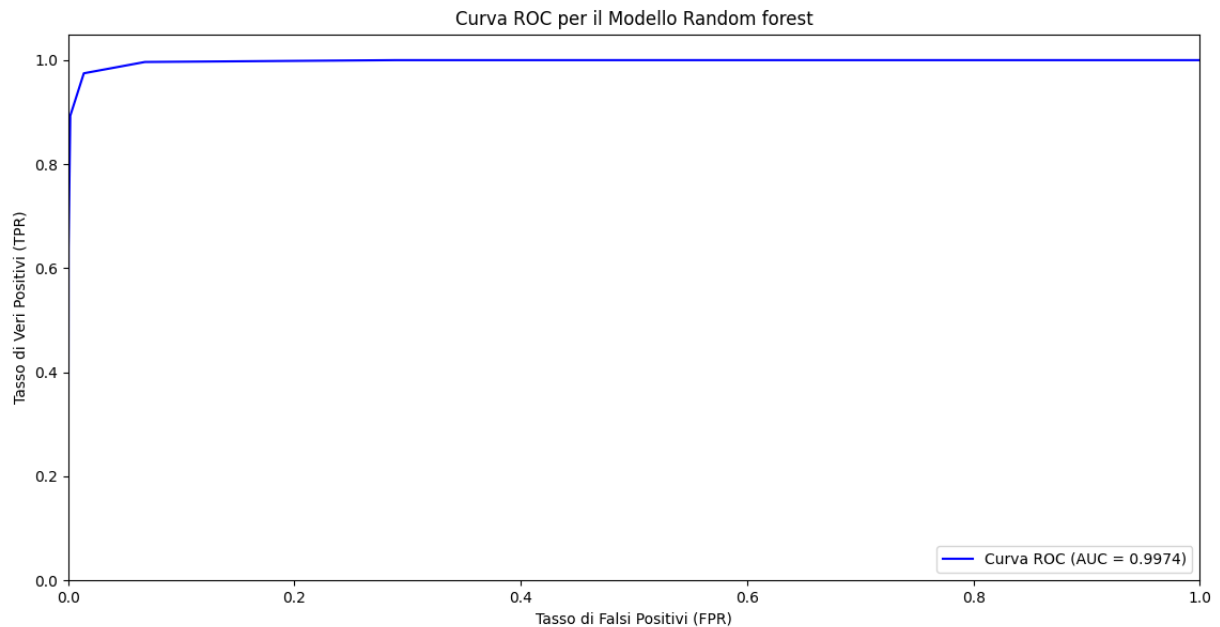


Immagine relativa alla curva ROC per il modello random forest

Il primo modello analizzato è il Random Forest, che ha prodotto una accuratezza media di $\bar{x} = 0.98$ e una deviazione standard pari a $\sigma = 0.005$. Come è possibile vedere dalla figura, l'area sotto la curva ROC è di 0.99. Un valore dell'area sotto la curva ROC così alto indica che il modello presenta quasi le stesse prestazioni di classificatore che riesce a discriminare perfettamente tra le due classi. Quindi nel complesso, i valori delle metriche analizzate sono decisamente molto ottimali e suggeriscono che il modello è particolarmente efficace per il presente task di classificazione.

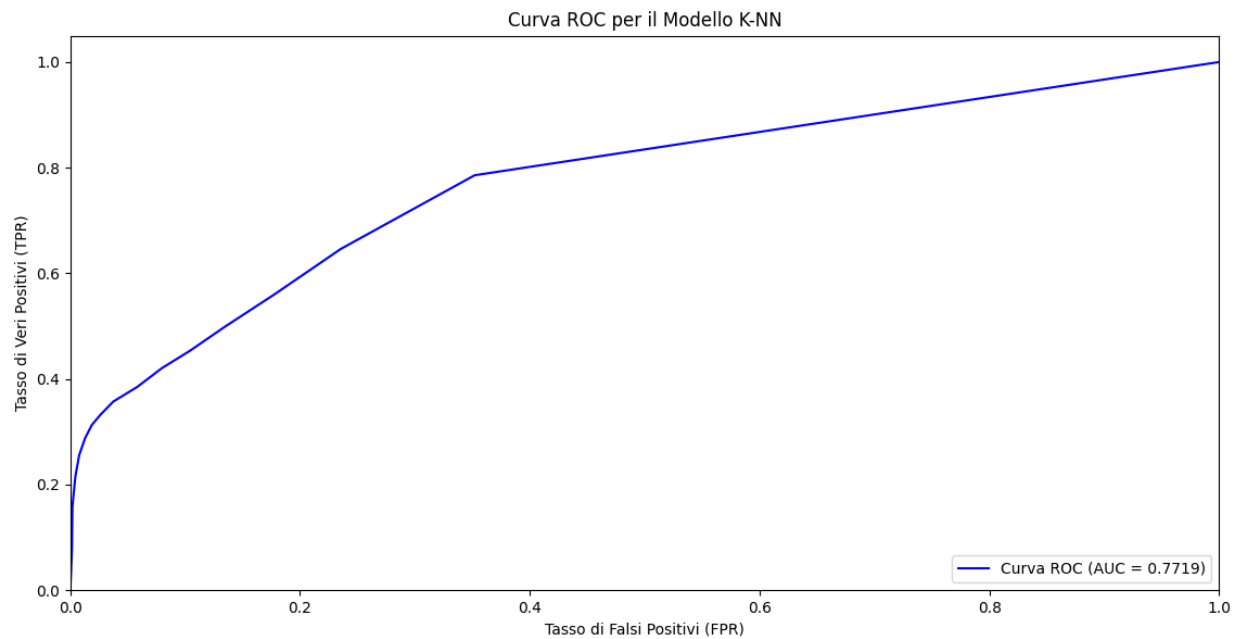


Immagine relativa alla curva ROC per il modello KNN

Il secondo modello analizzato è il KNN, che ha prodotto una accuratezza media di $\bar{x} = 0.95$ e una deviazione standard pari a $\sigma = 0.003$. Come è possibile vedere dalla figura, l'area sotto la curva ROC è di 0.77. Un tale valore dell'area indica che il modello si discosta abbastanza dalle prestazioni di un classificatore casuale, effettuando una classificazione discretamente buona. Inoltre, i valori delle metriche analizzate suggeriscono che il modello riscontra molta facilità nella discriminazione delle due classi.

Immagine relativa alla curva ROC per il modello KNN

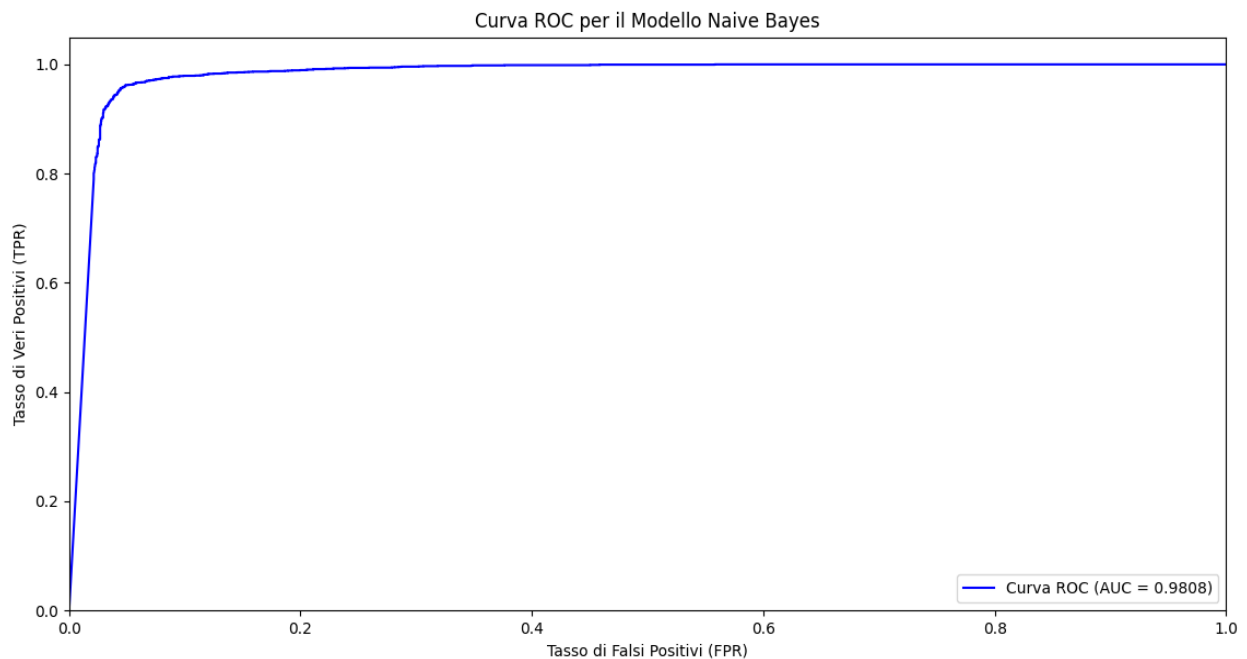


Immagine relativa alla curva ROC per il modello Naïve Bayes

Il terzo e ultimo modello analizzato è il Naïve Bayes, che ha prodotto una accuratezza media di $\bar{x} = 0.94$ e una deviazione standard pari a $p = 0.004$. Come è possibile vedere dalla figura, l'area sotto la curva ROC è di 0.98. Valori delle metriche così alti indicano che il modello presenta quasi le stesse prestazioni di un classificatore che riesce a discriminare perfettamente tra le due classi. Quindi, questi valori sono decisamente molto ottimali e suggeriscono che il modello è particolarmente efficace per il presente task di classificazione.

Sommario 5.4

Tirando le somme dei risultati ottenuti, possiamo dire che tutti e tre i modelli analizzati hanno ottime prestazioni sul presente dataset. Quindi non avrebbe senso parlare di un modello migliore visto che tutti e tre quasi si equivalgono in fatto di prestazioni. Tuttavia, l'analisi effettuata porta a enunciare delle conclusioni. Anche se i modelli sembrano essere molti robusti e performanti nella discriminazione delle classi, non è esclusa la possibilità che i modelli siano andati in **overfitting** (seppur sono state illustrate varie misure per contrastare questa problematica) o che semplicemente il dataset stesso presenti delle particolari caratteristiche che rendono facile la classificazione. Si pensi ad esempio al fatto che una determinata parola potrebbe comparire molto molto spesso all'interno delle news di una classe, rendendo più facile per il modello calcolare la predizione. Si sarebbe dovuto costruire un dataset più robusto e variegato che per questioni di tempo e di prestazioni computazionali non è stato costruito. Inoltre, l'obiettivo di questo caso di studio non era la costruzione di un potente e funzionante classificatore, ma bensì quello di effettuare un'analisi utilizzando vari strumenti e conoscenze del mondo dell'AI.

Apprendimento non supervisionato 6

Scelta dei modelli di apprendimento 6.1

Nel contesto del caso di studio, si è voluto anche verificare se l'apprendimento non supervisionato potesse fornire spunti di analisi nell'identificazione di pattern di similarità latenti nel dataset **clean_dataset_with_num_words.csv**. Si specifica quindi che quest'analisi è stata condotta prima della creazione della KB e dell'addestramento dei modelli predittivi. A tal fine quindi, è stata condotta un'analisi di clustering utilizzando l'algoritmo K-Means, presente nello script **k-means.py** all'interno della directory **/python/k-means**. Le feature scelte per il clustering sono la feature **num_words** e la feature **subject**. L'obiettivo è stato quindi determinare se queste feature potessero portare alla scoperta di particolari pattern di similarità nei dati sulla base dei valori delle feature scelte e successivamente, determinare la percentuale di fake news e true news all'interno dei cluster risultanti. In altri termini, si vuole verificare, qual è la percentuale di fake news e true news per le news di un certo tipo e con una certa lunghezza del testo. In corso di esecuzione viene generato un grafico 2D, grazie alla riduzione in componenti principali fatta tramite metodo **PCA**, che mostra quanto i cluster sono riusciti a dividersi sulla base delle feature scelte. Per quanto riguarda la scelta del valore dell'iperparametro **k** (numero di cluster risultanti), si è condotta un'analisi volta a trovare il valore ottimale di **k** per i dati in possesso. L'analisi è stata svolta utilizzando il **metodo dell'elbow** (gomito), il quale non fa altro che calcolare per ogni valore di **k** quale sarebbe l'errore risultante dal clustering con **k** cluster, in modo da

generare una curva in uno spazio 2D che mostri l'andamento dell'errore in relazione al numero di cluster (cioè la complessità del modello) . Quindi questo metodo esegue k-means ad ogni valore di k , calcola per ogni cluster la distanza di ogni esempio dal centroide associato e somma tutte queste distanze dividendo per il numero di esempi totale. Si otterrà quindi un punto nello spazio 2D di coordinate “errore” o “inerzia” e “numero cluster” . Lo script che esegue il **metodo dell'elbow** è presente nella directory **/python/k-means** Per l'implementazione di questa fase vengono ancora una volta utilizzate le librerie **matplotlib** , **scikit-learn** , **pandas** , **numpy** e ancora una volta è stata adottata la codifica one-hot encoding per la feature subject. Passiamo ora all'analisi dei risultati.

Analisi dei risultati 6.2

-

Analizziamo ora il risultato ottenuto dall'esecuzione del **metodo dell'elbow** :

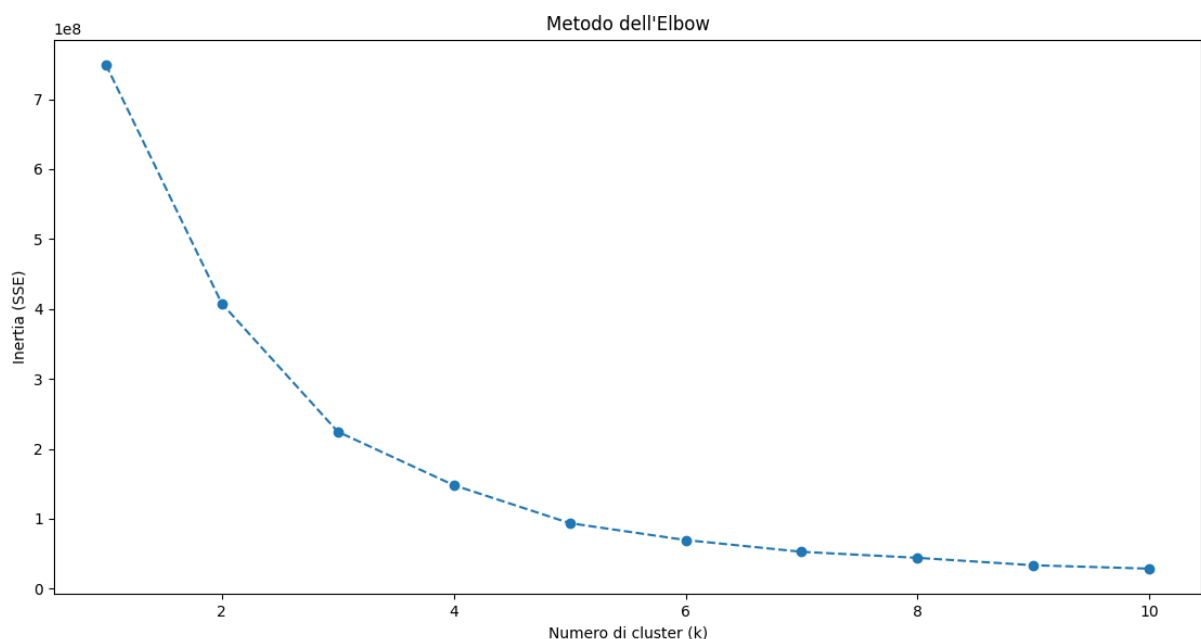


Immagine relativa al grafico per la curva dell'elbow

Come si vede dal grafico soprastante, per valori di k da 1 a 3 l'errore è molto elevato ma scende rapidamente fino ad arrivare ad un valore di k pari a 4 in cui l'errore inizia a diminuire molto più lentamente. Il gomito della curva è quel valore di k per cui la diminuzione dell'errore, passando al valore $k+1$, non giustifica l'incremento di complessità del modello, cioè utilizzare $k+1$ cluster. Il gomito della curva è il valore ottimale k con cui eseguire il clustering. Si evince facilmente, osservando la curva, che il valore ottimale (il gomito della curva) è $k=5$.

-

Analizziamo ora il risultato ottenuto dall'esecuzione dell'algoritmo **k-means** :

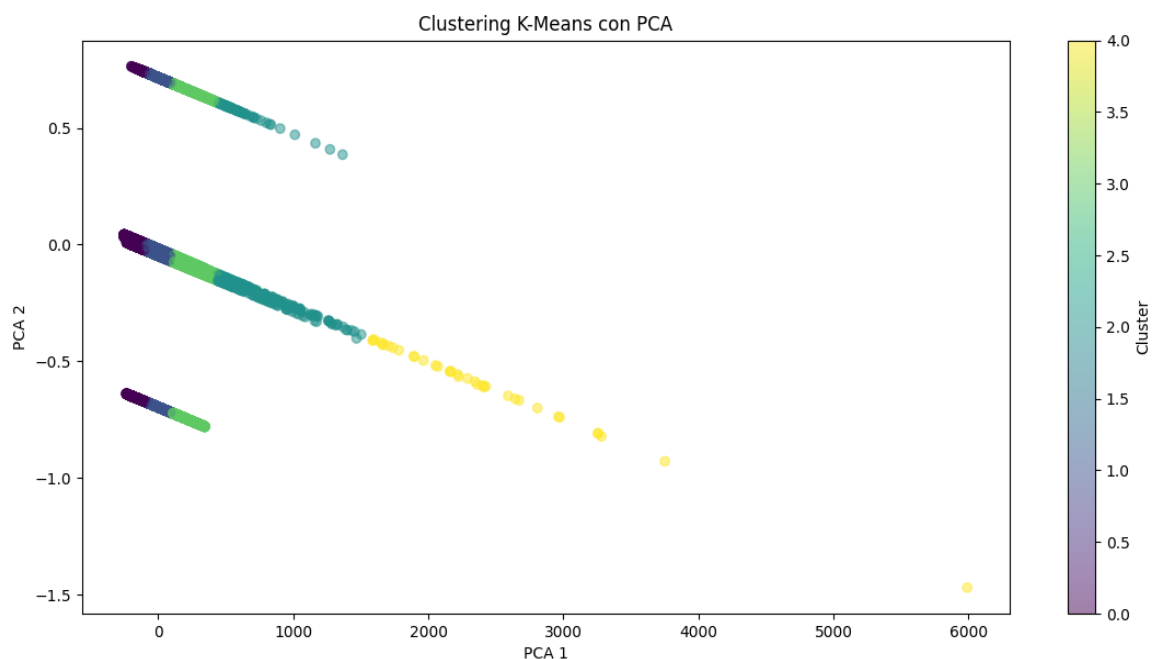


Immagine relativa al grafico **PCA** per il clustering

Il grafico presenta tanti puntini colorati con 5 colori. Ogni colore è associato ad un cluster. Il grafico presente tre linee che indicano la clusterizzazione lungo determinate componenti principali (le feature) scelte dal metodo **PCA**. Si evince facilmente dal grafico che i cluster sono ben separati tra loro lungo le diverse componenti, suggerendo che il clustering eseguito con **k-means** ha portato a dei buoni risultati.

-

Analizziamo ora le percentuali di categorie di documenti presenti e di fake news e true news, in ogni cluster :

```
Cluster 0:
num_words: 85.68
subject_Government News: 0.04
subject_Middle-east: 0.01
subject_News: 0.07
subject_US_News: 0.01
subject_left-news: 0.09
subject_politics: 0.20
subject_politicsNews: 0.30
subject_worldnews: 0.28
```

```
Cluster 1:
num_words: 242.50
subject_Government News: 0.02
subject_Middle-east: 0.01
subject_News: 0.41
subject_US_News: 0.01
subject_left-news: 0.07
subject_politics: 0.10
subject_politicsNews: 0.21
subject_worldnews: 0.16
```

```
Cluster 2:
num_words: 934.20
subject_Government News: 0.07
subject_Middle-east: 0.11
subject_News: 0.21
subject_US_News: 0.10
subject_left-news: 0.26
subject_politics: -0.00
subject_politicsNews: 0.00
subject_worldnews: 0.26
```

```
Cluster 3:
num_words: 452.17
subject_Government News: 0.03
subject_Middle-east: 0.01
subject_News: 0.31
subject_US_News: 0.01
subject_left-news: 0.07
subject_politics: 0.09
subject_politicsNews: 0.32
subject_worldnews: 0.15
```

```
Cluster 4:
num_words: 2648.86
subject_Government News: 0.14
subject_Middle-east: 0.39
subject_News: 0.00
subject_US_News: 0.31
subject_left-news: 0.17
subject_politics: 0.00
subject_politicsNews: 0.00
subject_worldnews: 0.00
```

Numero di elementi per ciascun cluster:

```
Cluster 0: 5497 elementi
Cluster 1: 7130 elementi
Cluster 2: 343 elementi
Cluster 3: 3403 elementi
Cluster 4: 36 elementi
```

Immagini relative ai valori delle feature nei centroidi, per ogni cluster

Percentuali di notizie vere e false per ciascun cluster:

tof	Fake	True
cluster		
0	42.350373	57.649627
1	62.636746	37.363254
2	74.344023	25.655977
3	53.100206	46.899794
4	100.000000	0.000000

Immagini relative alle percentuali di fake news e true news, per ogni cluster

Il **cluster 4** che contiene 36 news con lunghezza media di 2648 parole, ha il 100% fake news distribuite principalmente sulle categorie **Middle-East** , **Government News** e **US News** .

L'interpretazione qui è che news molte lunghe, forse con toni propagandistici o complottisti, sono al 100% fake news in questo dataset.

Il **cluster 2** che contiene 343 news con lunghezza media di 934 parole, ha il 75% fake news distribuite principalmente sulle categorie **Middle-East** , **Left-News** , **Worldnews** .

L'interpretazione qui è che news con una lunghezza non esagerata, ma comunque elevata, che trattano temi geopolitici hanno una forte presenza di fake news. Ancora una volta, news eccessivamente lunghe risultano in buon parte fake.

Il **cluster 1** che contiene 7130 news con lunghezza media di 242 parole, ha il 63% fake news distribuite principalmente sulle categorie **News** , **PoliticsNews** , **Politics** , **left-news** .

L'interpretazione qui è che news trattanti attualità e politica con una lunghezza dei testi medio bassa risultano maggiormente fake, seppur non di troppo.

Il **cluster 3** che contiene 3403 news con lunghezza media di 85 parole, ha il 42% fake news distribuite principalmente sulle categorie **News** , **PoliticsNews** , **Politics** e **Worldnews** .

L'interpretazione qui è che quando le news hanno una lunghezza media attorno alle 450 parole, gli argomenti si spostano sull'attualità con un conseguente calo delle fake news.

Il **cluster 0** che contiene 5497 news con lunghezza media di 452 parole, ha il 53% fake news distribuite principalmente sulle categorie riguardanti politica (circa un 65%) con la restante parte di news trattanti attualità. L'interpretazione qui è che quando le news hanno una lunghezza media molto bassa le notizie sono molto più affidabili.

Sommario 6.3

Tirando le somme di questa analisi, possiamo dire che in generale la varianza presente nei dati valutata sulla base della lunghezza del testo e del tipo di notizia è molto alta, ma tuttavia delle conclusioni utili si possono ricavare. Si capisce facilmente dall'analisi precedente, che quando le news hanno una lunghezza superiore alle 500 parole, i temi principali sono politica e geopolitica (americana e non) e il numero di fake news aumenta drasticamente. Viceversa, quando le news hanno una lunghezza molto bassa il livello di true news (e in particolare notizie di politica e geopolitica) è più alto. Quindi le feature scelte per la clusterizzazione hanno una correlazione con la percentuale di fake news e true news. Questa analisi motiva abbastanza la scelta di aver ricavato la feature **num_words** e di averla utilizzata in una regola **ProLog** per etichettare notizie di politica sospette. Questa analisi ha motivato anche la scelta di utilizzare la feature **subject** per apprendere i modelli **KNN** e **Random forest**.

