UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

DATA ANALYTICS

PROGETTO

# Fake News Analysis

*Authors:*
Davide Creati - 869274 - d.creati@campus.unimib.it
Martino Pettinari - 866496 - m.pettinari2@campus.unimib.it

# Indice

# 1 Introduzione e Obiettivi

Negli ultimi anni, la diffusione di fake news attraverso dichiarazioni, canali di comunicazione tradizionale e social network, ha assunto un'importante fetta sul totale delle notizie, minacciando l'affidabilità delle informazioni, influenzando facilmente l'opinione pubblica.

Il nostro progetto ha l'obiettivo di analizzare il contesto delle fake news, in particolare dichiarazioni e affermazioni in ambito politico e affrontare il problema della classificazione delle notizie.

## 1.1 Motivazioni Analisi

Per guidare le nostre analisi e assicurarci di indirizzare correttamente il lavoro, abbiamo definito una serie di domande a cui ci siamo posti l'obiettivo di rispondere. Le domande in questione sono le seguenti:

- *Come si evolvono le notizie (in questo caso dichiarazioni) nel tempo e qual è la loro correlazione con gli eventi politici?*
  Questa domanda ha l'obiettivo di indirizzare l'analisi sotto un punto di vista temporale della disinformazione ed esplorare eventuali legami tra la sua diffusione e i contesti politici specifici.
- *Esistono pattern linguistici distintivi associati alle classificazioni delle affermazioni?*
  L'obiettivo è identificare caratteristiche linguistiche, come l'uso di determinate parole, frasi o strutture grammaticali, che possano differenziare le notizie false da quelle vere.
- *È possibile sviluppare un classificatore affidabile per distinguere le notizie vere da quelle false?*
  Questa domanda ha un focus che punta alla creazione di un modello predittivo capace di identificare se una affermazione sia affidabile con un grado di accuratezza che sia accettabile.

Per rispondere a queste domande, abbiamo svolto il progetto partendo da un'approfondita analisi esplorativa dei dati e lo sviluppo di modelli di classificazione per il riconoscimento di notizie false.

## 1.2 Introduzione al Dataset

Per approfondire il tema delle fake news, abbiamo utilizzato il dataset **LIAR2** [1], una versione estesa e migliorata del precedente dataset *LIAR*, ampiamente utilizzato nella ricerca sulla classificazione delle fake news. LIAR2 è stato sviluppato per risolvere alcune delle principali problematiche del dataset originale, aggiungendo nuove informazioni e una struttura più robusta di dati.

Il dataset utilizza una classificazione *multi-classe* composta da sei categorie di veridicità, assegnate da esperti fact-checker in base al contenuto e al contesto di ogni dichiarazione.

Grazie a queste migliorie, il dataset LIAR2 rappresenta un insieme di dati più fedele e utile per la costruzione di modelli di classificazione robusti, in grado di affrontare l'analisi e la classificazione di notizie ed affermazioni nel contesto politico americano.

# 2 Dataset e Preprocessing

In questo capitolo verrà presentato nel dettaglio il dataset utilizzato per il progetto e successivamente, verranno illustrate le fasi cruciali di preprocessing che abbiamo applicato ai dati per renderli idonei all'analisi e alla modellazione, garantendo una maggiore qualità nei dati in base ai nostri obiettivi.

## 2.1 Analisi del Dataset

Il dataset LIAR2 [1] contiene circa 23.000 esempi contenenti dichiarazioni fatte da esponenti politici o raccolte da canali social in ambito della politica americana. I dati hanno una partizione 8:1:1 in training, validation e test set.

Tutte le dichiarazioni contenute nel dataset sono state raccolte e annotate da *PolitiFact*, una piattaforma indipendente di fact-checking. Gli esperti di PolitiFact utilizzano un processo di verifica, che consiste in:

1. Richiesta di prove alla fonte che ha formulato la dichiarazione.
2. Analisi di precedenti fact-checks e confronto con fonti ufficiali.
3. Ricerca avanzata attraverso strumenti come Google Fact-check Explorer.
4. Consultazione di esperti, documenti ufficiali, articoli e contenuti multimediali.
5. Assegnazione di una valutazione finale tramite la *Truth-O-Meter*, che attribuisce la classe di veridicità.

PolitiFact fornisce inoltre per ogni dichiarazione una motivazione testuale dettagliate e l'informazione sullo storico di credibilità degli oratori (informazioni presenti tra le features del dataset).

Il dataset ha diverse features per ogni dichiarazione, in particolare alcune di queste si focalizzano sull'affermazione fatta:

- **Statement**: Il testo della dichiarazione da valutare.
- **Subject**: Argomento o tema trattato nella dichiarazione.
- **Speaker**: Nome dell'autore della dichiarazione (spesso questa informazione può essere il canale social da cui proviene).
- **Speaker Description**: Profilo esteso del soggetto (inclusi ruoli passati, affiliazioni o biografia politica).
- **State Info**: Area geografica a cui si riferisce la dichiarazione (es. National, New York, ecc.).
- **Date**: Data in cui è stata pronunciata la dichiarazione.
- **Justification**: Spiegazione testuale dettagliata che motiva l'etichetta assegnata.
- **Contex**: Il contesto in cui è stata fatta la dichiarazione.

Oltre alle informazioni legate alla dichiarazione, ci sono anche features che arricchiscono il dataset con informazioni sullo storico delle affermazioni fatte da uno speaker ovvero, il numero di dichiarazioni storiche per ciascuna delle seguenti classi: *True*, *Mostly-true*, *Half-true*, *Barely-true*, *False*, *Pants-on-fire*. Questo insieme di sapere, per ogni speaker, quante dichiarazioni (per ogni grado di verità) sono state fatte in passato e permette di capire il livello di credibilità di ogni soggetto all'interno dei dati.

Le stesse categorie sono state usate per classificare le notizie, in modo da rappresentare le possibili mezze verità di alcune notizie. Le categorie rappresentano:

- **Pants-on-fire**: Affermazione completamente falsa, spesso palesemente assurda o deliberatamente fuorviante.
- **False**: Dichiarazione oggettivamente falsa, senza basi fattuali.
- **Barely-true**: Contiene elementi parzialmente veri, ma l'affermazione complessiva è sostanzialmente fuorviante.
- **Half-true**: Affermazione che presenta una miscela bilanciata di verità e falsità.
- **Mostly-true**: Affermazione sostanzialmente corretta, con alcuni dettagli secondari imprecisi o fuori contesto.
- **True**: Affermazione pienamente supportata dai fatti, senza ambiguità o imprecisioni rilevanti.

Il dataset offre una ricca collezione di dati, includendo informazioni testuali, numeriche e contestuali per ogni dichiarazione. Per le nostre analisi, abbiamo selezionato le features più pertinenti, basandoci sulle domande che guidano lo sviluppo del progetto.

## 2.2 Preprocessing

Prima di poter analizzare o utilizzare un qualsiasi insieme di dati, è fondamentale assicurarsi che questi siano puliti, coerenti e strutturati nel modo corretto. In questa fase ci siamo focalizzati su un insieme di operazioni che ci hanno permesso di preparare i dati grezzi per le successive analisi o per l'addestramento di modelli.

### 2.2.1 Bilanciamento

Prima di iniziare ad eseguire step di pulizia dei dati, abbiamo eseguito un'analisi sul dataset riguardante la distribuzione delle classi, al fine di verificarne il bilanciamento dei dati. Come mostrato in Figura 1, la distribuzione appare generalmente equilibrata, con la maggior parte delle classi comprese tra l'11% e il 16% del totale. Tuttavia, si può notare una classe con una concentrazione anomala di dati: la classe `false` risulta significativamente più rappresentata, coprendo circa il 28% delle istanze.

Nonostante questo sbilanciamento, non si è ritenuto opportuno intervenire tramite tecniche di bilanciamento (come *oversampling*, *undersampling* o pesatura delle classi). La scelta è motivata dal fatto che questa distribuzione riflette una caratteristica della realtà del dominio applicativo, ovvero una tendenza più marcata (in ambito politico) a produrre affermazioni false rispetto a quelle vere. Applicare un bilanciamento potrebbe compromettere la correttezza del dataset, introducendo un bias che avrebbe potuto alterare la capacità del modello di apprendere correttamente le dinamiche reali del fenomeno.

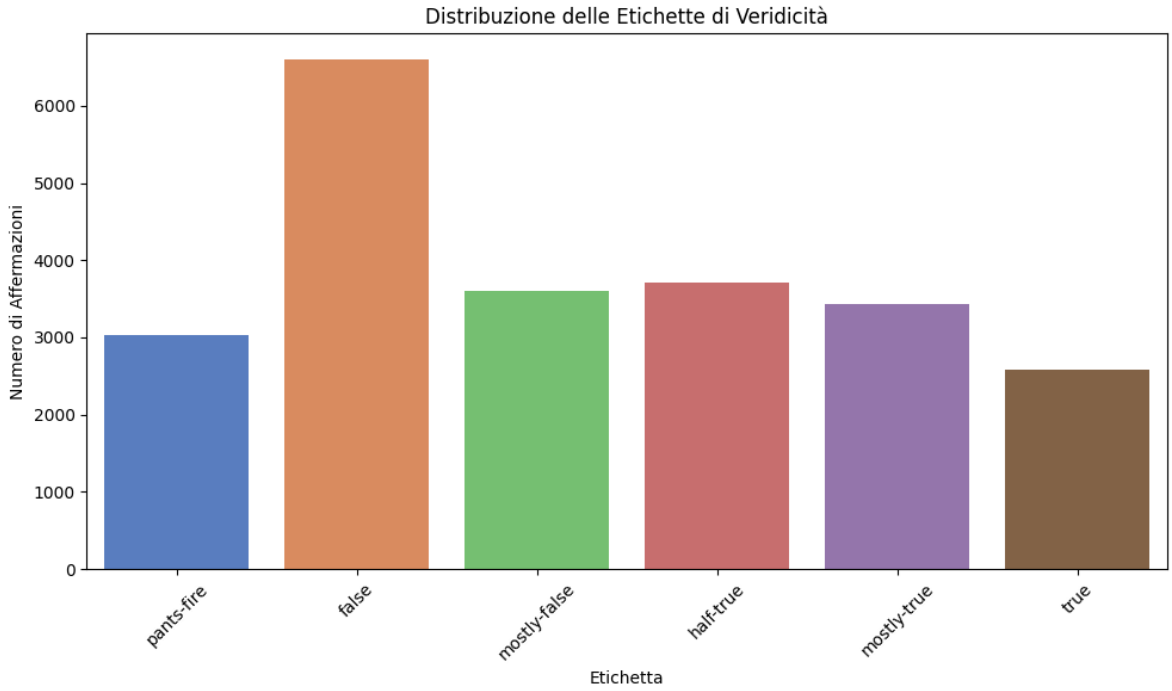


Figura 1: Bilanciamento dei Dati

### 2.2.2 Dati Mancanti

Un'altra analisi condotta sul dataset ha riguardato l'identificazione dei valori mancanti, analizzando i dati è emerso che solo quattro colonne presentano dati assenti: `subject`, `context`, `speaker_description` e `state_info`. Le prime tre mostrano una percentuale trascurabile di valori mancanti (tutte inferiori all'1%), mentre `state_info` mostra un'incidenza significativamente più elevata, pari al 24,57% del totale, come evidenziato nella Tabella 1.

Considerata che `state_info` rappresenta un'informazione che non può essere dedotta tramite tecniche di imputazione standard, non si è ritenuto necessario procedere con un'analisi approfondita della tipologia del dato mancante. A causa dell'elevata incidenza e dell'impossibilità di una stima affidabile, si è deciso di rimuovere tutte le istanze in cui questo campo risulta assente. La stessa scelta è stata applicata anche alle righe con valori mancanti nelle altre tre colonne. In ogni caso, l'impatto complessivo sulla dimensione del dataset risulti minimo.

### 2.2.3 Duplicati

Un'ulteriore analisi è stata dedicata per l'identificazione di eventuali duplicati all'interno del dataset, con l'obiettivo di garantire l'unicità delle istanze e prevenire possibili distorsioni. La verifica è stata effettuata considerando l'intero insieme di attributi e ha dimostrato che non sono presenti record duplicati, dunque non si è reso necessario alcun intervento di rimozione o pulizia in tal senso.

| Nome | Valori Mancanti o Nulli | Percentuale Mancante |
|---|---|---|
| id | 0 | – |
| label | 0 | – |
| statement | 0 | – |
| date | 0 | – |
| subject | 207 | 0.90% |
| speaker | 0 | – |
| speaker_description | 10 | 0.04% |
| state_info | 5641 | 24.57% |
| true_counts | 0 | – |
| mostly_true_counts | 0 | – |
| half_true_counts | 0 | – |
| mostly_false_counts | 0 | – |
| false_counts | 0 | – |
| pants_on_fire_counts | 0 | – |
| context | 154 | 0.67% |
| justification | 0 | – |

Tabella 1: Valori mancanti per colonna nel set di dati.

### 2.2.4 Credibility Score e Disinfo label

Come ultimo step del preprocessing, sono state generate due nuove variabili al fine di arricchire l'informazione disponibile nel dataset originale.

La prima feature introdotta è il `credibility_score`, rappresenta una stima della credibilità complessiva di ciascuno speaker, calcolata sulla base delle affermazioni precedenti classificate nel dataset. Il punteggio è espresso come valore numerico continuo compreso tra 0 e 1, dove valori più alti indicano una maggiore credibilità.

Per ogni riga del dataset, è stato calcolato il numero totale di affermazioni riferite allo speaker:

$$\begin{aligned}\text{total\_statements} = \text{true\_counts} + \text{mostly\_true\_counts} + \text{half\_true\_counts} \\ + \text{mostly\_false\_counts} + \text{false\_counts} + \text{pants\_on\_fire\_counts}\end{aligned}$$

Successivamente, la feature `credibility_score` è stata definita come rapporto tra le affermazioni vere o per lo più vere e il totale delle affermazioni attribuite allo speaker:

$$\text{credibility\_score} = \begin{cases} \frac{\text{true\_counts}+\text{mostly\_true\_counts}}{\text{total\_statements}} & \text{se total\_statements} > 0 \\ 0 & \text{altrimenti} \end{cases}$$

Al termine del calcolo, la variabile ausiliaria `total_statements` è stata rimossa dal dataset, in quanto non più necessaria.

La seconda feature creata, denominata `disinfo`, è un dato binario che sintetizza in maniera più generale la natura dell'affermazione: vera o falsa. Nello specifico, per ogni record del dataset, il valore della variabile è assegnato come segue:

$$\text{disinfo} = \begin{cases} 1 & \text{se la label è } \texttt{false}, \texttt{mostly-false} \text{ o } \texttt{pants-on-fire} \\ 0 & \text{altrimenti} \end{cases}$$

Questa trasformazione consente di distinguere rapidamente tra affermazioni potenzialmente fuorvianti e affermazioni affidabili.

# 3 Analisi Esplorativa

L'analisi esplorativa dei dati ha rappresentato una fase cruciale del progetto, in quanto ci ha permesso di approfondire in modo concreto le dinamiche e i fenomeni di disinformazione presenti nel dataset. Attraverso l'utilizzo di dashboard interattive, è stato possibile visualizzare e interpretare l'evoluzione temporale delle affermazioni. Oltre a ciò, l'analisi si è concentrata anche sull'individuazione di pattern ricorrenti nelle affermazioni false, con l'obiettivo di comprendere se esistano elementi linguistici o strutturali comuni tra le diverse dichiarazioni.

## 3.1 Ricerca di Pattern Semantici

La prima analisi che abbiamo svolto ha l'obiettivo di verificare l'esistenza di pattern semantici ricorrenti all'interno delle affermazioni (`statement`) presenti nel dataset, valutando il grado di similarità tra le istanze appartenenti alla stessa classe. L'obiettivo è comprendere se affermazioni etichettate in modo simile presentino effettive somiglianze a livello linguistico e contenutistico.

L'analisi non si è limitata a valutare la coerenza semantica all'interno delle singole classi, così come definite dalle etichette originali, ma ha considerato anche una versione semplificata delle etichette, chiamata *disinfo*, che distingue tra affermazioni vere e false. Questo approccio ci ha permesso di capire se raggruppare le classi in categorie di veridicità più ampie renda più evidente la separazione nei contenuti, facilitando così l'identificazione di eventuali differenze linguistiche o tematiche.

Questa indagine rappresenta un passaggio fondamentale per valutare la coerenza del dataset rispetto all'etichettatura fornita, nonché per individuare eventuali ambiguità che potrebbero influenzare negativamente le prestazioni dei modelli di classificazione. L'analisi è stata condotta tramite la trasformazione delle frasi in rappresentazioni vettoriali utilizzando tecniche di *Natural Language Processing* (NLP), seguita dall'applicazione di metriche di similarità per misurare la coesione intra-classe e la separazione tra categorie di veridicità.

Per questo scopo, è stato utilizzato *Sentence-BERT* (SBERT) [2], una variante del modello BERT progettata per la generazione efficiente di *sentence embeddings*. SBERT utilizza un'architettura siamese o triplet per produrre vettori semantici confrontabili, superando i limiti del modello BERT tradizionale nel confronto diretto tra frasi. L'uso di metriche come la *cosine similarity* consente di stimare il grado di affinità semantica tra affermazioni.

Dato che gli statement sono frasi tendenzialmente con poche parole, SBERT si rivela particolarmente adatto a catturarne le relazioni semantiche latenti. Ogni affermazione è stata quindi convertita in un vettore numerico a dimensione fissa, preservandone il significato contestuale e rendendo possibile l'analisi quantitativa della similarità tra istanze.

### 3.1.1 Creazione degli embeddings

Per ottenere rappresentazioni vettoriali delle affermazioni presenti nel dataset, è stata utilizzata la funzione `SentenceTransformer.encode()`, appartenente all'omonima libreria Python. Tale libreria semplifica l'uso di modelli pre-addestrati basati su SBERT, progettati per generare *sentence embeddings* ottimizzati per il confronto semantico tra frasi.

Nel nostro caso, le frasi presenti nella colonna `statement` sono state fornite in input alla funzione `encode()`, ottenendo per ciascuna una rappresentazione nello spazio semantico.

Tali vettori sono stati successivamente utilizzati per valutare la similarità intra-classe attraverso l'applicazione di metriche di distanza, con l'obiettivo di individuare pattern linguistici ricorrenti o eventuali anomalie semantiche.

Il processo che utilizza `SentenceTransformer` per generare gli embedding semantici, si articola nei seguenti passaggi:

1. **Tokenizzazione**: La frase viene suddivisa in token tramite il tokenizer del modello pre-addestrato.
2. **Embedding dei token**: Ogni token è convertito in un vettore numerico tramite i pesi del modello.
3. **Contestualizzazione**: I vettori sono elaborati dai livelli del modello Transformer per catturare il contesto semantico.
4. **Pooling**: I vettori dei token vengono aggregati in un singolo vettore di frase mediante tecniche come:
   - `Mean Pooling`: Media dei vettori non mascherati;
   - `CLS Token`: Utilizzo del vettore del token speciale `[CLS]`.
5. **Output**: Viene restituito un vettore semantico utilizzabile per analisi quali clustering, similarità o classificazione.

### 3.1.2 Analisi dei risultati

Al fine di visualizzare la distribuzione semantica delle affermazioni nello spazio latente, è stata applicata una tecnica di riduzione dimensionale agli embedding precedentemente calcolati. In particolare, è stato utilizzato l'algoritmo *UMAP* [3], con una proiezione nello spazio tridimensionale ($\mathbb{R}^3$).

L'algoritmo costruisce un grafo locale ponderato che rappresenta la struttura dei dati nello spazio originale e successivamente ne ottimizza una proiezione in uno spazio a bassa dimensionalità, cercando di preservare la struttura topologica locale e globale.

Nello sviluppo dell'analisi, UMAP è stato preferito ad altre tecniche di riduzione come PCA o t-SNE, poiché offre un buon compromesso tra fedeltà locale e globale della struttura dei dati, è computazionalmente efficiente su dataset di grandi dimensioni, ed è in grado di mantenere le relazioni semantiche apprese durante la fase di embedding.

L'obiettivo principale di questa operazione è stato quello di esplorare la possibilità che gli embedding generati da SBERT presentino una naturale separazione nello spazio semantico, coerente con le etichette di classe.

Come si può notare nelle Figure 2 e 3, non si osserva una netta separazione tra le diverse classi all'interno dello spazio proiettato. Le istanze appaiono distribuite in maniera omogenea, senza far emergere pattern o cluster distinti che ne evidenzino una struttura semantica ben definita.

Si può notare, una maggiore densità di punti corrispondenti alla classe *false* in alcune aree dello spazio, fenomeno riconducibile principalmente alla prevalenza numerica di tale classe rispetto alle altre. Questa concentrazione locale non implica necessariamente una coesione semantica, ma riflette piuttosto l'effetto della distribuzione sbilanciata delle etichette nel dataset.

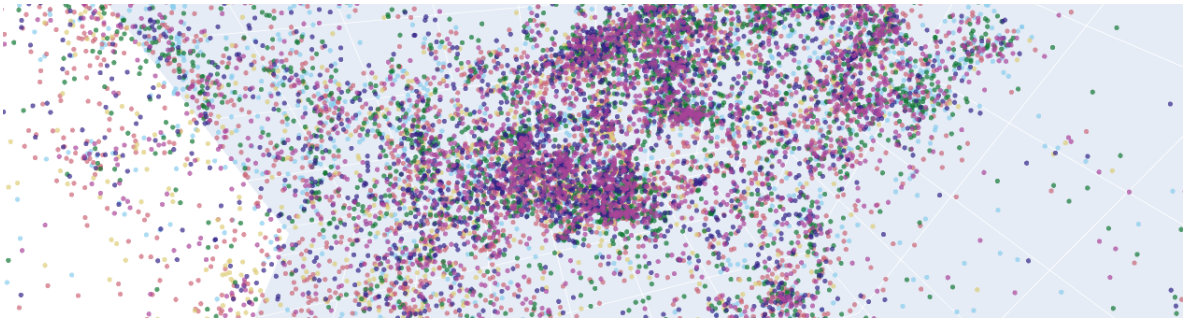

Figura 2: Visualizzazione tramite UMAP degli embedding delle affermazioni, colorati secondo le classi originali presenti nel dataset.

Figura 3: Visualizzazione tramite UMAP degli embedding delle affermazioni, colorati in base alla label aggregata *disinfo*, che distingue tra affermazioni vere e false.

Poiché non è stato possibile rilevare visivamente nessuna struttura o pattern distinti nella distribuzione delle affermazioni, l'analisi è stata estesa mediante l'impiego di metriche quantitative, al fine di ottenere valutazioni chiare e non ambigue.

In particolare, sono state considerate due metriche principali: la *cosine similarity*, utilizzata per misurare il grado di similarità semantica tra coppie di affermazioni, e il *silhouette score*, che permette di valutare la coesione intra-classe e la separabilità inter-classe nel contesto della rappresentazione vettoriale degli embedding.

La prima analisi quantitativa effettuata ha riguardato il calcolo della *cosine similarity*, una metrica ampiamente utilizzata nel campo dell'elaborazione del linguaggio naturale per valutare la similarità tra vettori in spazi ad alta dimensionalità. La *cosine similarity* misura l'angolo tra due vettori, fornendo un valore compreso tra -1 e 1, dove valori più vicini a 1 indicano una maggiore similarità semantica.

Questa metrica adatta particolarmente bene all'analisi degli *embedding* ottenuti tramite modelli come SBERT, in quanto consente di quantificare con precisione quanto due frasi siano semanticamente affini, indipendentemente dalla loro magnitudine.

Nell'ambito della nostra analisi, la *cosine similarity* è stata calcolata considerando sia le classi originali presenti nel dataset, sia la label aggregata *disinfo*, che distingue tra affermazioni vere e false. L'obiettivo è stato quello di ottenere una valutazione numerica della similarità intra-classe e inter-classe, al fine di confermare o meno la presenza di coesione semantica all'interno delle categorie definite.

Nella Figura 4 si può osservare il risultato dell'analisi, la similarità intra-classe, rappresentata dalla diagonale della matrice, tende ad assumere valori maggiori rispetto alla similarità inter-classe, fenomeno osservabile soprattutto per le classi appartenenti alla porzione

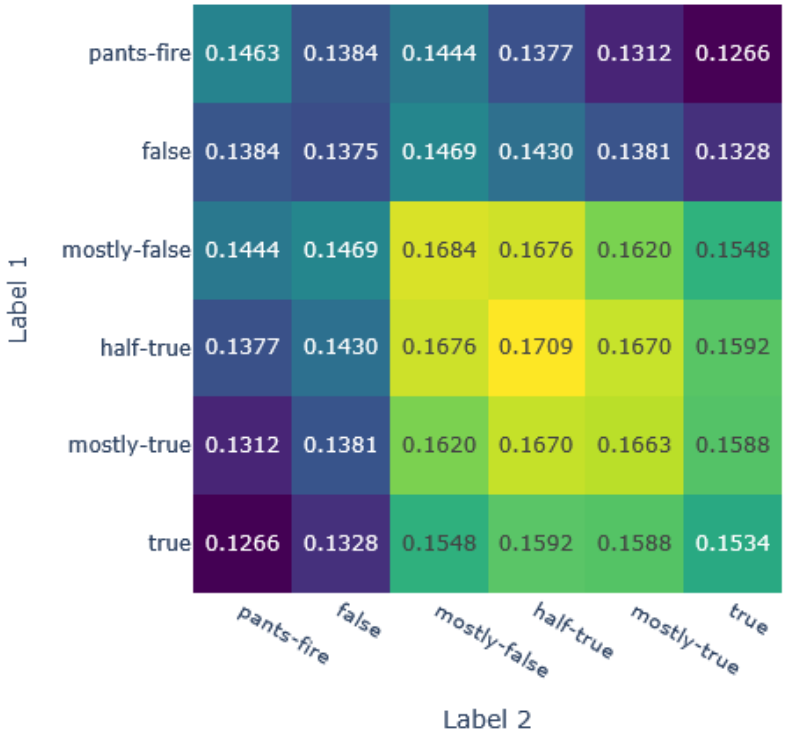


Figura 4: Matrice di confusione multi-label con i risultati del cosine similarity

"vera" delle etichette. Tuttavia, non si riscontra una separazione netta e ben definita tra le diverse classi.

Inoltre, classi semanticamente e concettualmente vicine, come *half-true* e *mostly-true*, presentano elevati valori di similarità interna, complicando ulteriormente la distinzione precisa tra le categorie. Le classi semanticamente più distanti mostrano valori di similarità inferiori, indicativi di una maggiore distanza semantica, sebbene tali valori non differiscano in maniera significativa dalla media complessiva.

Si può affermare che l'analisi non ha messo in evidenza differenze semantiche marcate tra le affermazioni appartenenti alle diverse classi, suggerendo una certa sovrapposizione nei contenuti e nelle strutture linguistiche delle istanze.

Come mostrato nella Figura 5, che riporta i risultati del calcolo della *cosine similarity* tra le classi della colonna *disinfo*, si osserva nuovamente l'assenza di una separazione netta tra le categorie. Tuttavia, si nota una maggiore coesione all'interno della classe delle affermazioni considerate "vere", a conferma di quanto evidenziato nell'analisi basata sulle label originali.

La seconda analisi quantitativa condotta ha riguardato il calcolo del *Silhouette Score*, una metrica utilizzata per valutare la qualità dei risultati di clustering, misurando quanto ciascun elemento sia coerente con il proprio cluster rispetto ai cluster vicini. Il Silhouette Score per un punto $i$ è definito come:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

dove:

- $a(i)$ è la distanza media tra $i$ e tutti gli altri punti nel medesimo cluster (coesione).

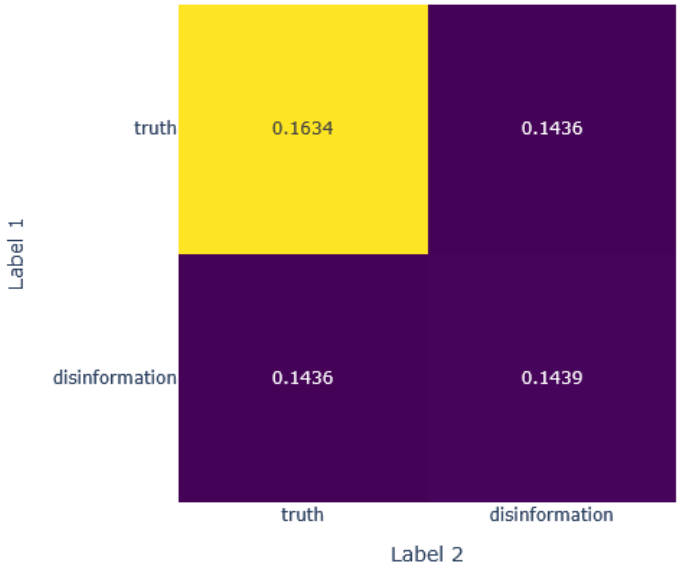


Figura 5: Matrice di confusione disinfo con i risultati del cosine similarity

- $b(i)$ è la distanza media tra $i$ e tutti i punti nel cluster più vicino (separazione).

Il valore del Silhouette Score varia tra $-1$ e $+1$:

- Valori prossimi a $+1$ indicano che il punto è ben posizionato all'interno del proprio cluster e lontano dai cluster vicini.
- Valori prossimi a $0$ suggeriscono che il punto si trova sul confine tra due cluster.
- Valori negativi indicano che il punto potrebbe essere assegnato al cluster sbagliato.

In questo contesto, il Silhouette Score è stato impiegato per valutare la coesione intra-classe e la separazione inter-classe delle affermazioni nel dataset, sia considerando le classi originali sia la label aggregata *disinfo*. L'obiettivo è ottenere una misura numerica della qualità del clustering semantico delle affermazioni, al fine di integrare le osservazioni qualitative precedenti con una valutazione quantitativa.

I risultati dell'analisi del *Silhouette Score* evidenziano valori medi prossimi allo zero per entrambe le etichettature considerate. In particolare, per le classi originali del dataset, il valore del Silhouette Score calcolato utilizzando la *cosine* è pari a $-0.0122$, mentre per la label aggregata *disinfo* il valore è di $0.0096$. Questi valori suggeriscono una scarsa coesione intra-classe e una limitata separazione inter-classe.

In generale, un valore medio del Silhouette Score vicino a zero indica che le affermazioni sono distribuite in modo tale da non appartenere chiaramente a un singolo cluster, ma piuttosto si trovano in prossimità dei confini tra più cluster. Questo comportamento è particolarmente evidente nelle classi associate alla porzione "vera" delle etichette, dove la sovrapposizione semantica tra le affermazioni rende difficile una distinzione netta tra le categorie.

I risultati ottenuti suggeriscono che le affermazioni nel dataset presentano una struttura semantica complessa e interconnessa, con una bassa separabilità tra le diverse classi.

## 3.2 Spiegazioni globali sulle affermazioni

Per comprendere l'influenza di singole parole (o più nello specifico dei token) sul processo decisionale del modello, è stata impiegata la tecnica *SHapley Additive exPlanations* (SHAP) [4]. I valori SHAP permettono di quantificare il contributo di ciascun token all'output predittivo, rendendo esplicito se un elemento lessicale influisce positivamente o negativamente sulla classificazione.

L'impiego di SHAP in questo contesto ci ha permesso di ottenere:

- Una valutazione trasparente e quantitativa del ruolo di ogni token nella predizione individuale.
- L'identificazione di possibili bias lessicali o artefatti linguistici sfruttati dal modello;
- Un livello di analisi interpretativa complementare a quello effettuato tramite embedding, focalizzato sul singolo elemento lessicale anziché sulla struttura globale del testo.

A differenza dell'analisi precedente, basata su SBERT, in questa fase abbiamo utilizzato DistilBERT [5] in combinazione con SHAP per valutare l'impatto individuale dei token sulla decisione del modello, senza misure di similarità ma tramite interpretabilità locale. L'approccio consente di identificare vocaboli o frammenti di testo che determinano maggiormente l'attribuzione alle classi.

Per ottimizzare tempi e risorse computazionali, l'analisi è stata limitata a un campione di circa 50 affermazioni. Questa scelta ha permesso di generare gli `shap_values` in tempi ragionevoli, senza compromettere la qualità delle spiegazioni locali.

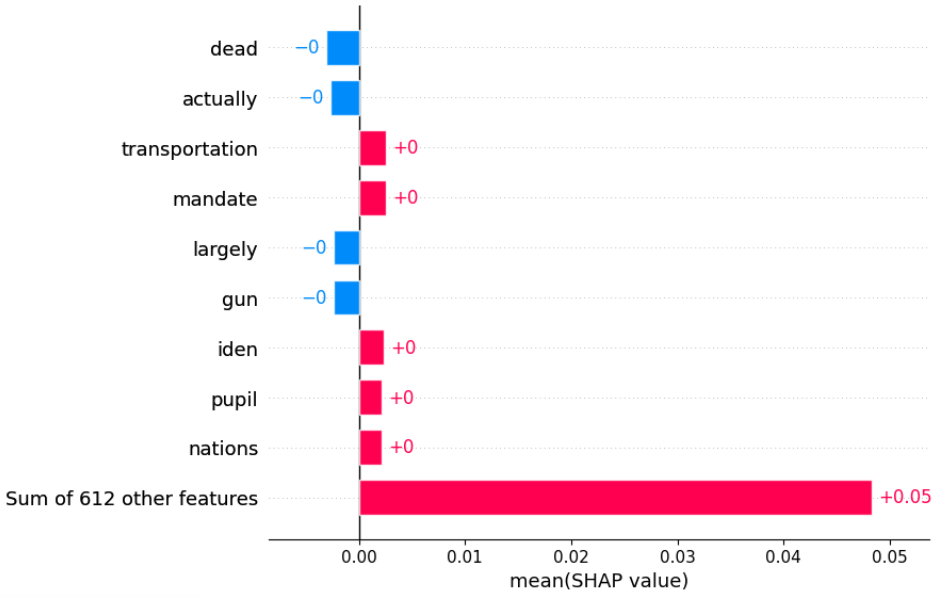


Figura 6: SHAP on pants-on-fire label

La Figura 6 presenta un grafico a barre generato tramite i valori SHAP, focalizzato sull'interpretazione dell'importanza media delle feature rispetto alla previsione della classe `pants-on-fire`. Le parole mostrate nel grafico rappresentano i token che hanno contribuito maggiormente (in senso positivo o negativo) alla classificazione dei campioni appartenenti a questa classe.

Dall'analisi del grafico si osserva che l'impatto medio dei singoli token risulta generalmente molto basso, con valori SHAP prossimi allo zero. Tale evidenza suggerisce che nessun termine individuale esercita un'influenza significativa sulla decisione del modello per la classe in esame.

Ne consegue che il modello non si basi su poche parole chiave fortemente distintive per classificare un'affermazione come appartenente a `pants-on-fire`, ma opera invece sulla base di un pattern diffuso, aggregando il contributo di numerosi token con impatto ridotto. Questo comportamento può indicare una strategia predittiva più distribuita e meno suscettibile a bias legati a termini isolati.

Si può dedurre che non esiste una correlazione marcata tra le singole parole e la classificazione nella classe `pants-on-fire`, poiché tutti i valori SHAP individuali risultano prossimi allo zero, denotando un'influenza limitata di ciascun token preso singolarmente. I grafici relativi alle altre classi riportano pattern molto simili a quello osservato per la classe `pants-on-fire`, con singoli token caratterizzati da valori SHAP medi prossimi allo zero e nessuna parola in grado di esercitare un'influenza predominante (grafici visualizzabili nel notebook Google Colab, omessi per leggibilità e poco significato per i risultati).

È importante sottolineare che l'analisi condotta in questa sezione non ha un valore oggettivo fortemente rappresentativo ai fini della valutazione complessiva del dataset. A causa dei vincoli computazionali legati all'utilizzo del metodo SHAP, l'esperimento è stato limitato a un sottoinsieme di soli 50 esempi, a fronte di un dataset di oltre 23.000 affermazioni. Dunque, questa analisi deve essere interpretata come una stima esplorativa e non esaustiva, il cui scopo è quello di fornire indicazioni preliminari sulla possibile distribuzione dell'importanza lessicale appresa dal modello, piuttosto che determinare con certezza le dinamiche predittive sottostanti.

## 3.3 Visualizzazione dinamica dei dati

Per migliorare l'analisi condotta, è stata sviluppata una dashboard interattiva che permette di esplorare visivamente i dati del dataset, con particolare attenzione ai fenomeni legati alla disinformazione.

L'interfaccia, basata su componenti dinamici e interattivi, è stata strutturata in due principali sezioni tematiche accessibili tramite schede (tab):

- **Frequenza**: Consente di analizzare, anno per anno, la distribuzione delle affermazioni rilasciate dai principali speaker del dataset. La dashboard mostra le fonti (speaker) più attive per ciascun anno, correlando la *frequenza* delle dichiarazioni con il relativo *credibility score*. La visualizzazione permette di individuare gli speaker più attivi di oggi anno, visualizzando anche il loro livello di affidabilità delle informazioni. Ciò fa emergere eventuali picchi di attività legati a singole fonti. Inoltre, la dashboard visualizza i dati a partire dal 2007, in quanto gli anni precedenti risultano avere pochi dati rappresentati nel dataset. In particolare, la mancanza di dati per interi anni o il basso numero di affermazioni non consentivano una suddivisione temporale coerente e una corretta visualizzazione.

- **Percentuale**: Offre una serie di strumenti per l'esplorazione approfondita dei pattern di disinformazione nel tempo. In particolare:

  - È possibile selezionare un singolo speaker e visualizzare, tramite un grafico a linee, l'evoluzione temporale della percentuale di affermazioni false (`false`, `pants-on-fire`, `mostly-false`) rilasciate.
  - È disponibile un istogramma personalizzabile che mostra il conteggio complessivo di dichiarazioni veritiere e non per le fonti più frequenti nel dataset.
  - Una heatmap evidenzia, per ciascuna fonte con un numero sufficiente di affermazioni, la proporzione di disinformazione, permettendo di visualizzare gli speaker con tasso di dichiarazioni false (sul totale dal relativo speaker) maggiore in assoluto.

- **Veridicità**: Fornisce una visione temporale globale, analizzando l'andamento della *veridicità media* delle dichiarazioni nel tempo, insieme al *numero medio di affermazioni* rilasciate. Grazie a una media mobile a 30 giorni e a uno slider temporale interattivo, è possibile osservare come si evolvono nel tempo la qualità delle affermazioni e la loro frequenza, con la possibilità di focalizzarsi su intervalli temporali specifici. Anche in questo caso, la dashboard mostra i dati a partire dal 2007 per i motivi precedentemente elencati.

- **Contesto**: Analizza la disinformazione in relazione al contesto in cui le affermazioni sono state rilasciate (es. interviste, dichiarazioni pubbliche, post social). La dashboard permette di visualizzare:

  - Un istogramma che mette a confronto, per i contesti più frequenti, la distribuzione delle affermazioni vere e false.
  - Una heatmap che evidenzia la proporzione di disinformazione per ciascun contesto, limitata a quelli con un numero sufficiente di dati. Questo consente di identificare i contesti comunicativi più esposti alla diffusione di notizie false.

- **Tema**: Consente di esplorare la disinformazione per argomento trattato (es. economia, sanità, sicurezza). La dashboard offre:

  - Un istogramma dinamico che confronta il numero di affermazioni veritiere e false per i temi più discussi.
  - Una heatmap che mostra la proporzione di disinformazione per ciascun tema, evidenziando quelli più soggetti a dichiarazioni ingannevoli.

Nel complesso, la dashboard permette di visualizzare in modo intuitivo i seguenti dati:

- La frequenza delle dichiarazioni per le fonti (o speaker) più popolari nel tempo, con divisione annua.
- L'andamento della disinformazione per singolo speaker.
- Le fonti maggiormente associate alla diffusione di contenuti falsi.
- Una visualizzazione globale su trend di veridicità delle notizie.

### 3.3.1 Analisi delle dashboard

L'utilizzo della dashboard permette di eseguire un'analisi dinamica, consentendo di individuare trend significativi nel tempo e di osservarne l'evoluzione dei dati.

#### Frequenza degli speaker

In particolare, nella tab `Frequenza` è possibile visualizzare i 30 speaker più attivi per ciascun anno. L'osservazione dell'andamento dei grafici nel corso degli anni evidenzia variazioni significative nella dimensione dei punti rappresentati nei diagrammi, in particolare alcune sorgenti di affermazioni acquisiscono sempre più rilevanza, mentre altri tendono a ridursi progressivamente.

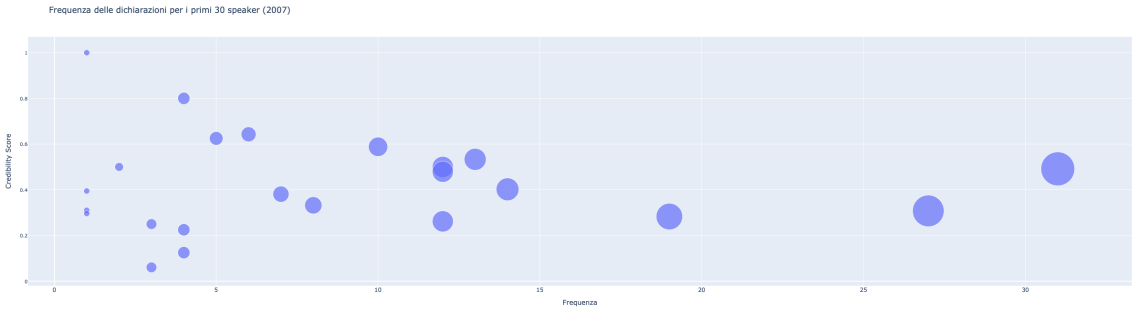


Figura 7: 30 speakers principali nel 2007

Come si osserva nella Figura 7, nel 2007 (primo anno visualizzabile attraverso la dashboard) la distribuzione della frequenza tra gli speaker è relativamente omogenea. Pur essendoci alcune fonti con un numero maggiore di affermazioni, la maggior parte degli speaker è ben distinguibile nel diagramma. Tutti gli elementi rappresentano singoli individui che hanno rilasciato dichiarazioni pubbliche, dato che non sono ancora presenti social media come fonti. In particolare, lo speaker più rilevante risulta essere Hillary Clinton, con un credibility score prossimo a 0,5.

Spostandosi verso la fine del periodo osservabile, si nota un cambiamento radicale del diagramma. L'avvento dei social media ha trasformato il modo in cui vengono diffuse le affermazioni, amplificando la voce di utenti senza un ruolo politico ufficiale e modificando la composizione delle fonti più attive.

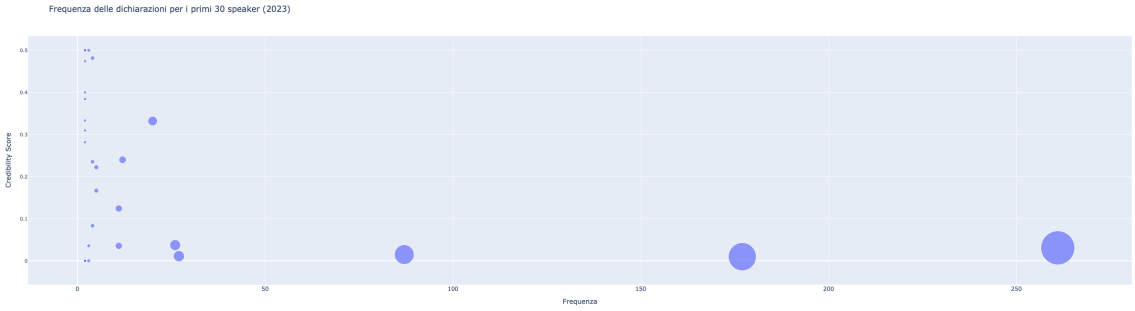


Figura 8: 30 speakers principali nel 2023

Nella Figura 8, si notano alcuni elementi di dimensioni molto maggiori rispetto agli altri, questi punti corrispondono a canali social come Instagram, Facebook o blog. Al contrario, gli altri speaker risultano marginali nel diagramma (seppure siano i 30 più frequenti del rispettivo anno). Inoltre, è evidente come questi nuovi attori principali abbiamo un credibility score molto basso (inferiore a 0,1), suggerendo che, nonostante la loro elevata frequenza di affermazioni, non siano affidabili.

### Affidabilità speaker

Un'ulteriore analisi può essere svolta utilizzando la tab `Percentuale`, che permette di osservare l'andamento nel tempo della proporzione di affermazioni false per ciascun speaker. Questa sezione della dashboard è particolarmente utile per confrontare l'affidabilità di personaggi pubblici nel corso degli anni.

Due figure politiche molto attuali e interessanti da analizzare, Donald Trump e Joe Biden (entrambe tra i 10 speaker più frequenti in modo assoluto nel dataset), mostrano comportamenti comunicativi differenti. Nella Figura 9, si nota un trend chiaramente inclinato verso una maggiore diffusione di notizie false in merito alla comunicazione di Donald Trump. Il suo grafico evidenzia un primo incremento attorno al 2015 (in concomitanza con l'inizio della campagna presidenziale), seguito da un'impennata costante a partire dal 2018, fino a raggiungere valori vicini o superiori all'80% nel 2022. In tutto l'intervallo osservabile, Trump non scende mai sotto il 65% di affermazioni false, evidenziando una comunicazione costantemente poco affidabile secondo i dati.

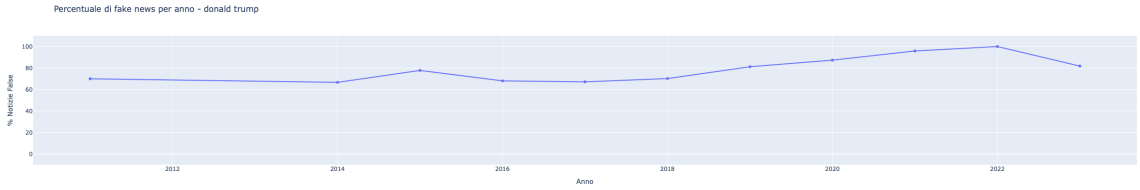


Figura 9: Evoluzione temporale veridicità affermazioni di Donald Trump

Viceversa, nella Figura 10, si osserva un comportamento molto diverso: nel 2007 si registra un picco di affermazioni false, dovuto al numero ridotto di dichiarazioni fatte in quell'anno (esattamente 8 dichiarazioni presenti nel dataset). Tuttavia, negli anni successivi, pur con alcune oscillazioni, la percentuale di affermazioni false si mantiene sempre al di sotto del 50%, suggerendo una comunicazione mediamente più veritiera e coerente nel tempo. La curva associata a Biden riflette quindi un profilo comunicativo tendenzialmente più affidabile rispetto a quello di Trump.

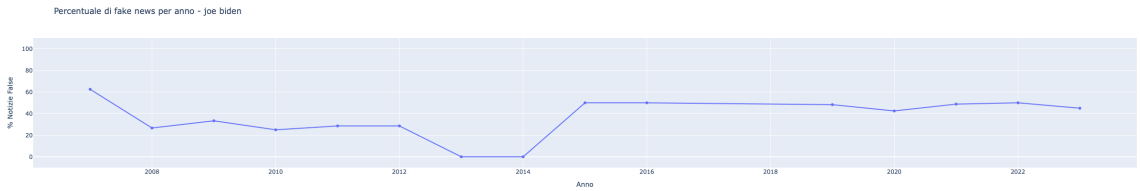


Figura 10: Evoluzione temporale veridicità affermazioni di Joe Biden

Queste differenze evidenziano come la dashboard, attraverso visualizzazioni mirate, permetta non solo di monitorare la frequenza e l'impatto delle fonti informative, ma anche di valutare l'affidabilità percepita dei singoli speaker nel corso del tempo.

### Evoluzione veridicità

Infine, la tab `Veridicità` consente di osservare l'andamento temporale complessivo delle quantità e veridicità delle affermazioni fatte. In questo grafico, la linea blu rappresenta la media di veridicità delle dichiarazioni, mentre la linea arancione mostra il numero medio di affermazioni registrate nello stesso intervallo temporale.

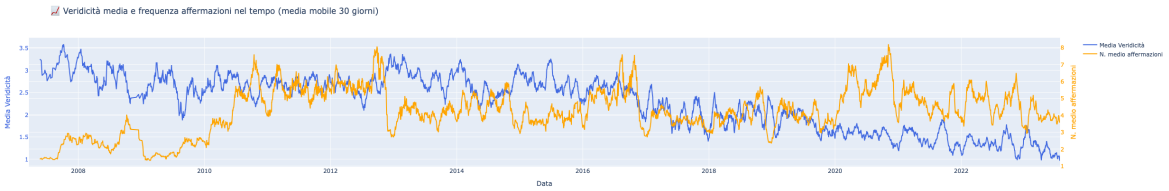


Figura 11: Evoluzione temporale complessiva veridicità e quantità statement

Analizzando nella Figura 11 l'andamento della veridicità (linea blu), si nota una tendenza decrescente nel tempo. A partire dal 2007, la veridicità media delle affermazioni tende a diminuire, indicando un calo nella qualità dell'informazione diffusa da diversi speaker o canali social. Questo trend riflette l'influenza che hanno avuto i social network in merito alla divulgazione di informazioni in ambito politico in America.

Con un trend opposto, si osserva una crescita generale del volume di affermazioni (linea arancione), con picchi negli anni 2012, 2016 e 2020. Questi momenti coincidono con le principali elezioni presidenziali statunitensi, periodi caratterizzati da un'intensa attività comunicativa e da una maggiore diffusione di informazioni non vere.

La sovrapposizione di queste due informazioni mette in evidenza un aspetto cruciale: aumenta la quantità di affermazioni, ma diminuisce la loro qualità media. Si può dedurre che ci sia un legame tra l'aumento della comunicazione in ambito politico con la perdita di affidabilità nei contenuti diffusi.

### Analisi sui diversi contesti

La Figura 12 rappresenta il diagramma osservabile nella tab `Contesto`. Analizza la distribuzione delle affermazioni in relazione al contesto, ordinando i contesti in ordine decrescente sulla base del numero totale di disinformazioni rilevate. In particolare, la colonna verde rappresenta il numero di affermazioni classificate come false, mentre la colonna rossa indica quelle ritenute veritiere.

Dall'analisi emerge che i primi tre contesti, corrispondenti prevalentemente a piattaforme di social media, presentano una maggiore incidenza di affermazioni false rispetto a quelle vere. Ciò suggerisce una maggiore diffusione di contenuti disinformativi nei canali social e in particolar modo, se può notare come siano quelli che dominano per quantità di affermazioni sul totale.

Per fornire una visione più dettagliata sul contesto, è stato realizzato un ulteriore diagramma che rappresenta, in ordine decrescente, i contesti comunicativi in base alla proporzione

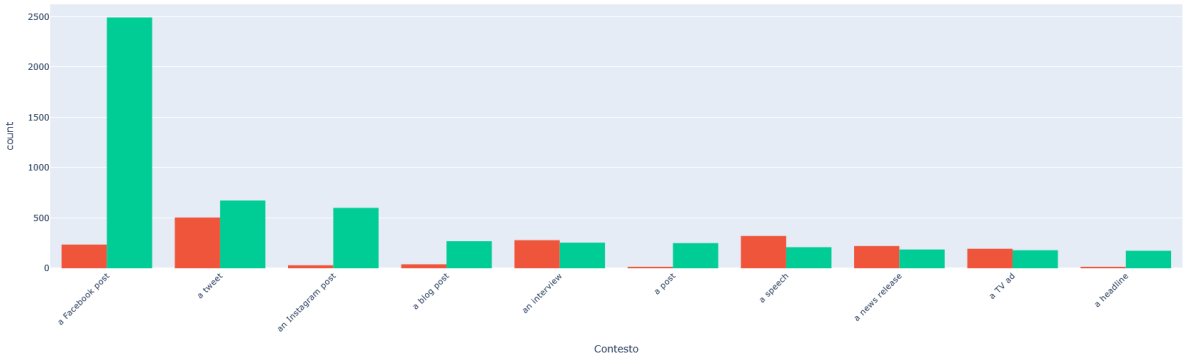


Figura 12: Analisi dei contesti con più disinformazione

di affermazioni false rispetto al totale delle affermazioni registrate per ciascun contesto. In modo da valutare non solo il volume assoluto di disinformazione, ma anche la densità relativa della stessa all'interno dei diversi contesti comunicativi. Si può notate come, anche questo diagramma, segua quello precedentemente visto, assecondando i canali social come quelli con più disinformazione.

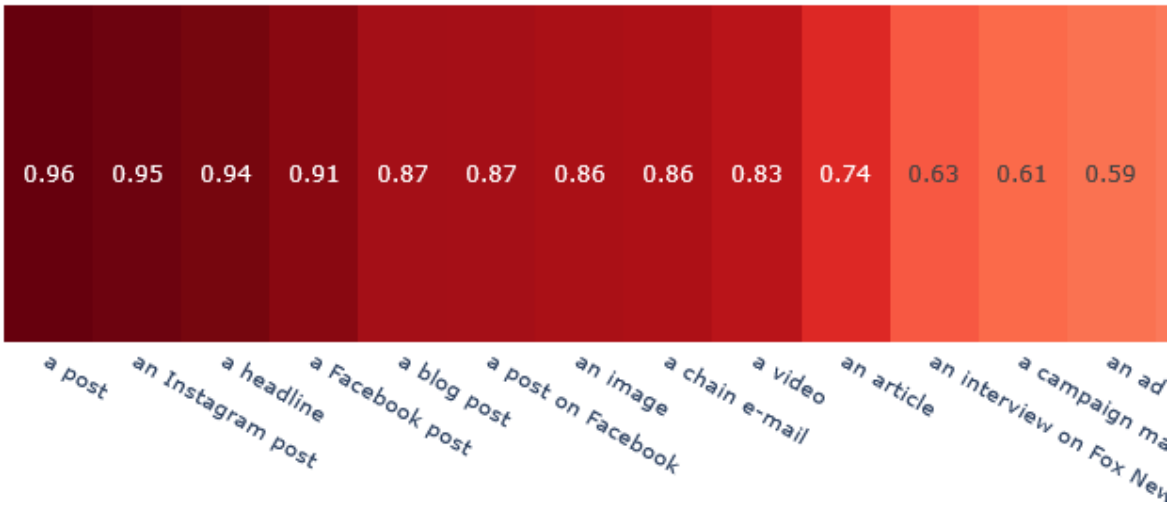


Figura 13: Proporzione di disinformazione nei contesti

### Analisi dei temi (argomenti)

L'ultima tab presente nella dashboard, ovvero `Tema`, raggruppa la divisione per temi o argomenti dei vari statement. Abbiamo applicato gli stessi metodi utilizzati per la tab `Contesto`, per quanto riguarda la logica di aggregazione dei dati.

Nella Figura 14, si possono osservare i principali temi presentiti nel dataset. Emerge come le categorie relative alle *elezioni* e alla *health-care* presentino i livelli più elevati di disinformazione all'interno del dataset. Inoltre ha anche un peso importante il tema legato al *coronavirus* che però è una tematica limitata principale ad un periodo temporale.

Nella tab si può anche analizzare la heatmap che mostra i temi con più disinformazione sul totale di statement. L'analisi conferma la tendenza riscontrata nei temi legati alle *elezioni* e alla *health-care*, che si distinguono per un'elevata incidenza di affermazioni false rispetto

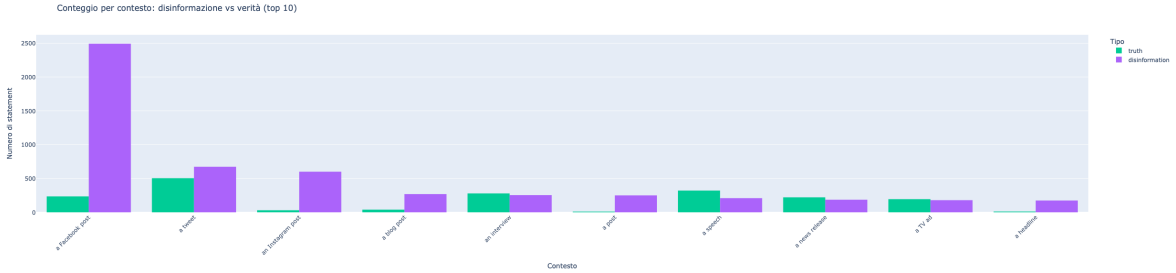


Figura 14: Analisi dei contesti con più disinformazione

a quelle veritiere. Tali dati sono stati rilevati attraverso l'attività di verifica dei fact-checker, indicando una particolare criticità informativa in questi ambiti tematici.

# 4 Classificazione

In questo capitolo descriviamo il processo di classificazione delle notizie, il cui obiettivo è assegnare a ciascuna dichiarazione presente nel dataset un livello di verità, assegnando una delle 6 categorie presenti.

Per raggiungere questo scopo, è stato seguito un flusso di lavoro articolato in diverse fasi: dalla preparazione dei dati, alla trasformazione del linguaggio naturale in rappresentazioni numeriche, fino all'addestramento e alla valutazione di diversi modelli di classificazione.

Abbiamo adottato un approccio ibrido, in cui il modello BERT [6] è stato utilizzato sia come estrattore di caratteristiche semantiche (feature extractor), sia come modello da ottimizzare tramite fine-tuning, al fine di confrontare l'efficacia delle due strategie.

Nel dettaglio, sono stati adottati due approcci distinti:

- **Classificatore esterno (feature extraction)**
  In questo approccio, BERT è stato utilizzato come modello pre-addestrato senza alcuna modifica ai suoi pesi. Le rappresentazioni testuali (embedding), ottenute dal token di classificazione, vengono utilizzate come input di un classificatore esterno. Inoltre, diversi tipi di features sono stati combinati con gli embedding per testare diverse possibilità di training del modello.

- **Fine-tuning di BERT**
  In questo caso, il modello bert-base-cased è stato ulteriormente addestrato sul task di classificazione, facendo un fine-tuning. Questo approccio consente a BERT di adattare le sue rappresentazioni in modo più specifico al contesto.

Nel seguito del capitolo verranno descritti nel dettaglio tutte le fasi del processo, i modelli impiegati e i risultati ottenuti, con confronti tra gli approcci utilizzati.

Durante lo sviluppo dei diversi classificatori è stato utilizzato BERT nella sua versione `bert-base-uncased`, questa scelta è stata motivata dal fatto di concentrare l'attenzione del modello sul contenuto semantico delle affermazioni, riducendo l'influenza di variabili come la formattazione testuale che non apportano informazioni rilevanti ai fini della classificazione della veridicità.

## 4.1 Classificatore esterno

In questo approccio, abbiamo utilizzato BERT per ottenere gli embeddings degli statement e poi abbiamo implementato un modello che, prendendo in input i dati prodotti da BERT, classifichi ogni statement.

In particolare, il tokenizer di BERT è stato utilizzato per convertire gli *statement* testuali in token, ottenendo i tensori, i quali contengono i dati che per la nostra analisi sono i più importanti, ovvero: `input_ids` e `attention_mask`.

Questi tensori sono stati forniti in input al modello BERT, ottenendo il tensore `pooler_output`, ovvero l'embedding del token speciale `[CLS]`. Questo vettore, lungo 768 dimensioni, rappresenta una sintesi semantica dell'intera affermazione ed è stato utilizzato come input per un classificatore esterno.

Una volta ottenuti gli embedding per i set di `train`, `validation` e `test`, sono stati testati diversi classificatori, per la classificazione di ogni statement nel livello di verità corrispettivo.

### 4.1.1 Valutazione diversi classificatori

Per valutare l'efficacia degli embedding ottenuti da BERT come base per la classificazione, abbiamo sperimentato diverse tipologie di classificatori. Siamo partiti utilizzando solo gli embedding del testo con l'obiettivo di confrontare architetture differenti, per identificare il miglior compromesso tra accuratezza, tempo di addestramento e generalizzazione. In particolare, in questa prima fase, sono stati provati 3 diversi modelli: 2 reti neurali e una random-forest.

Le due varianti di reti variano in termini di architettura, in particolare, sono state valutate le seguenti configurazioni:

- **Architettura base**
  In questo caso sono presenti 2 layer nascosti con 128 e 64 neuroni con funzione di attivazione `ReLU`, e layer finale `softmax` per classificazione multiclass.

- **Architettura profonda**
  In questa variante sono presenti 3 layer nascosti, con rispettivamente 256, 128 e 64, con regolarizzazione tramite `Dropout(0.3)` e `BatchNormalization`.

Entrambe le architetture sono state allenate con ottimizzatone `Adam` e `categorical_crossentropy` come funzione di loss. Inoltre, sono state usate in entrambe i casi 10 epoche per il training, provando in modo sperimentale valori diversi, abbiamo costatato valori di accuracy abbassarsi sul validation set.

Come terza architettura, abbiamo provato ad utilizzare una modello `RandomForest Classifier`, testato sugli embedding originali. Il vantaggio principale di questo modello risiede nella sua robustezza rispetto al rumore.

### 4.1.2 Risultati classificazione statement

Eseguendo le fasi di addestramento e valutazione, i risultati sperimentali indicano che il modello basato su rete neurale con due soli layer nascosti rappresenta la soluzione più bilanciata in termini di accuratezza. Questa architettura basica ha raggiunto un'accuratezza sul set di test di quasi circa il 32% (31,99%), risultando la più performante tra le soluzioni testate.

Il modello neurale più complesso, composto da più layer e tecniche aggiuntive di regolarizzazione come `Dropout` e `BatchNormalization`, ha ottenuto risultati leggermente inferiori al 30% di accuratezza (29,56%). L'incremento della complessità architetturale non ha portato a un reale miglioramento prestazionale.

Per quanto riguarda il modello basato su **Random Forest** ha mostrato prestazioni decisamente inferiori. Nonostante diversi test per provare diverse architetture in modo iterativo dei suoi iperparametri principali (numero di alberi, profondità massima, numero di feature per split, ecc.), l'accuratezza in fase di test non ha mai superato circa il 15%. Questo risultato suggerisce che la natura distribuita e densa degli embedding di BERT non si adatta facilmente a modelli basati Random Forest, che faticano a catturare relazioni non lineari ad alta dimensionalità.

Complessivamente, questi risultati mostrano che, anche senza fine-tuning diretto del modello BERT, è possibile ottenere buone performance, considerando la classificazione su 6 classi.

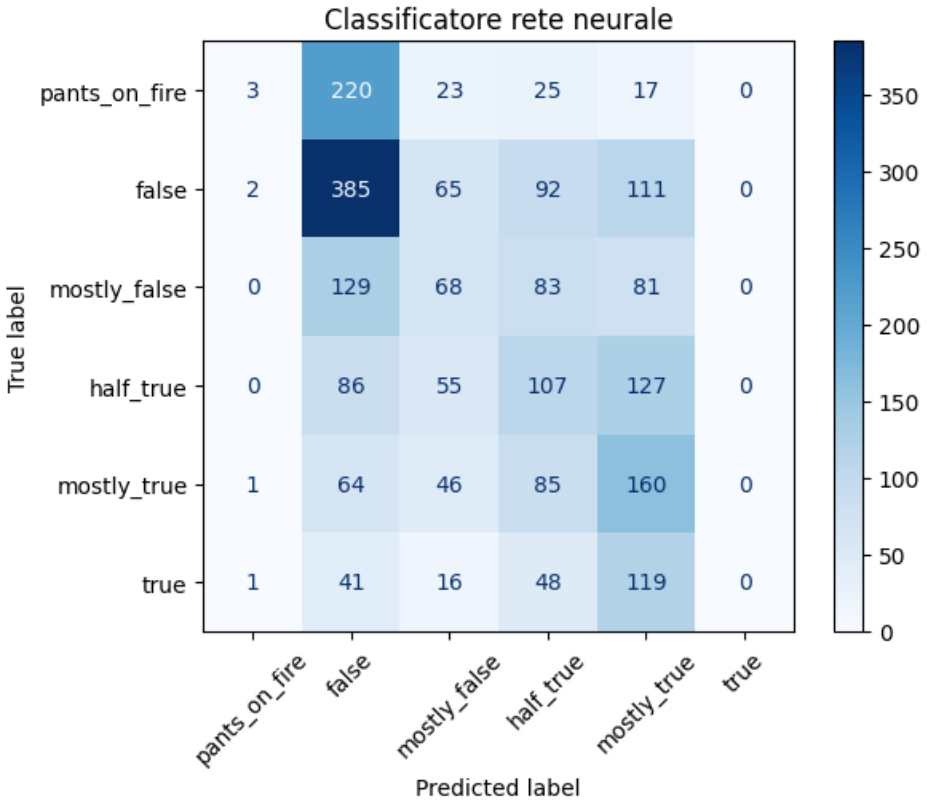


Figura 15: Matrice di confusione classificazione rete con 2 livelli

Per la visualizzazione dei risultati dell'architettura più performante, nella Figura 15 è presente la matrice di confusione delle predizioni. Il diagramma evidenzia una marcata tendenza del modello a prediligere la classe più frequente nel dataset, ovvero `false`, che viene frequentemente predetta anche per istanze appartenenti ad altre categorie. Questa tendenza può anche essere dovuta al fatto che nel dataset sono presenti più affermazioni false dato che rispecchiano la situazione reale delle dichiarazioni che vengono fatte. Inoltre, la classe `true` non è mai predetta. Infine, si può notare come le classi `false` e `mostly_true` raccolgano numerose istanze delle classi delle classi vicine.

Il modello mostra difficoltà nel discriminare correttamente tra classi vicine, come `half_true`, `mostly_true` e `mostly_false`, dove si osservano forti sovrapposizioni. Questo comportamento è attribuito sia alla somiglianza tra le etichette che ad affermazioni molto simili appartenenti a classi diverse ma vicine semanticamente. In particolare, le classi meno rappresentate (`pants_on_fire`, `true`) tendono ad essere frequentemente confuse con classi più comuni.

Si può dedurre come la matrice di confusione conferma che, sebbene il modello riesca a catturare alcune relazioni tra classi, soffre ancora di un certo bias verso le categorie più centrali che risultano essere anche quelle più ibride di verità.

Per valutare il modello nella capacità di predire se un'affermazione sia vera o false, è stata realizzata una seconda matrice di confusione aggregando le sei classi originali in due macro-categorie: `true` (contenente le etichette `true`, `mostly_true` e `half_true`) e `false` (contenente `false`, `mostly_false` e `pants_on_fire`). I risultati di questa classificazione binaria sono riportati nella Figura 16.

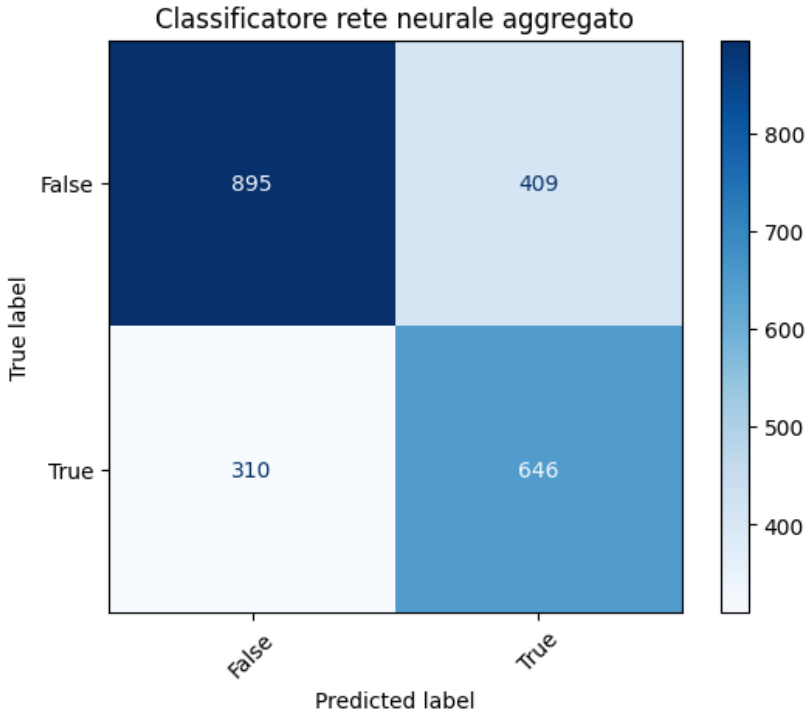


Figura 16: Matrice di confusione classificazione rete con 2 livelli con risultati aggregati

Per testare questa predizione, e stato utilizzato lo stesso modello con una aggregazione dei risultati, ciò ha mostrato una accuratezza di circa il 68% (68,19%). Dalla matrice emerge l'efficacia nella classificazione delle affermazioni false da quelle veritiere. Tuttavia, si può osservare la presenza di molti falsi negativi e falsi positivi, riflettendo la realtà delle dichiarazioni raccolte. Poiché le etichette `half_true` e `mostly_true` racchiudono affermazioni parzialmente vere, è plausibile che il modello fatichi a collocarle nella categoria corretta, accentuando l'ambiguità tra le due macro-classi.

In generale, la classificazione binaria migliora la leggibilità delle prestazioni del modello e dimostra che, pur non perfetto, l'approccio basato su rete neurale riesce ad apprendere la distinzione tra verità e falsità.

### 4.1.3 Riduzione dimensionalità

Per valutare l'impatto della dimensionalità sull'apprendimento, sono stati condotti esperimenti di riduzione dimensionale sugli embedding generati da BERT, applicando sia la *Principal Component Analysis* (PCA) che l'*Uniform Manifold Approximation and Projection* (UMAP). Entrambe le tecniche sono state testate utilizzando il classificatore basato su rete neurale che ha mostrato le migliori performance nella fase precedente, ovvero il modello basico di rete neurale.

La PCA, pur essendo una tecnica lineare, ha prodotto risultati migliori rispetto a UMAP. In particolare, una riduzione a circa 50 componenti, utilizzata per allenare il modello, ha raggiunto un'accuratezza di poco superiore al 33% (33.14%) sui dati di test. Al contrario, l'utilizzo di UMAP, nonostante la sua natura non lineare, ha portato a risultati inferiori, con un'accuratezza ferma al 27% (27.35%).

Inoltre, considerando la versione aggregata delle classi (vero/falso), l'accuratezza ottenuta con la configurazione PCA a 50 componenti ha raggiunto circa il 67% (67.08%). Questo risultato conferma che, sebbene la riduzione dimensionale comporti una certa perdita di dettaglio, il modello riesce comunque a mantenere una buona capacità discriminativa nei confronti della veridicità generale delle affermazioni.

Per ciascun approccio, sono stati sperimentati diversi numeri di componenti principali, al fine di individuare il punto di equilibrio tra riduzione della complessità e mantenimento dell'informazione discriminativa. In particolare, il miglior risultato osservato è con la PCA utilizzando 50 componenti.

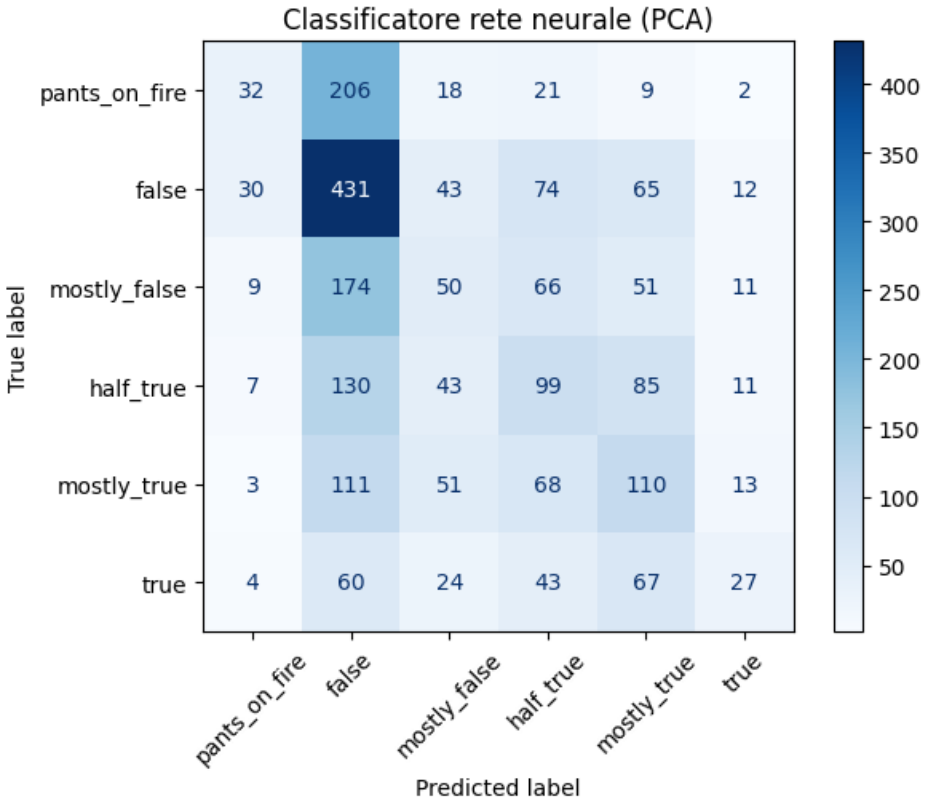


Figura 17: Matrice di confusione classificazione con PCA

Anche per questo approccio, con la matrice di confusione, presente nella Figura 17 possiamo osservare come ci sia una sovrapposizione delle classi centrali, simile allo scenario precedentemente osservato nella classificazione senza riduzione delle componenti.

### 4.1.4 Classificazione con features aggiuntive

Al fine di migliorare la capacità del modello, è stata esplorata la possibilità di utilizzare ulteriori features, da affiancare agli embedding testuali ottenuti tramite BERT. In particolare, sono state considerate le seguenti features:

- **Speaker:** L'autore dell'affermazione può essere un'informazione rilevante per la classificazione, poiché alcuni individui tendono a fare dichiarazioni più veritiere o meno rispetto ad altri.

- **Subject:** Ogni affermazione è associata a uno o più argomenti tematici, che possono fornire indizi sul tipo di contenuto trattato e sulle sue probabilità di veridicità. L'informazione sugli argomenti può contribuire a distinguere affermazioni appartenenti a contesti con tassi di veridicità differenti.

- **Year:** L'anno in cui l'affermazione è stata pronunciata, dato estratto dalla data. In particolare, è stato incluso solo l'anno dato che nei dati non sono presenti eventi storici collegati a giornate specifiche dunque la dvisione in anni può aiutare il modello a valutare il contesto storico o politico di riferimento.

Queste features sono state concatenate all'embedding testuale, precedentemente ridotto a 50 dimensioni tramite PCA. Anche se questa tecnica non abbia prodotto i risultati migliori in assoluto, è stata preferita per un motivo: la riduzione dimensionale consente di contenere le informazioni all'embedding testuale, dando però maggiore rilevanza alle features aggiuntive all'interno del modello. Il vettore risultante dalla concatenazione è stato quindi utilizzato come input per il classificatore neurale. L'architettura della rete è stata opportunamente adattata per gestire la nuova struttura dell'input, pur mantenendo inalterata la configurazione generale dei layer impiegata nel modello base.

### 4.1.5 Preprocessing delle features aggiuntive

Per poter integrare le features aggiuntive con gli embedding testuali, è stato necessario applicare un processo di preprocessing specifico a ciascun tipo di dato, al fine di ottenere rappresentazioni numeriche scalate in modo coerente.

Il preprocessing è stato differenziato in base alla tipologia delle variabili:

- **Feature numeriche:** `year`
  La variabile `year`, rappresentando un'informazione temporale, è stata scalata tramite `MinMaxScaler`. Questa tecnica ha trasformato il valore in un intervallo compreso tra 0 e 1, preservando la distribuzione relativa degli anni e rendendo il dato adatto all'input della rete neurale. L'utilizzo di uno scaler lineare semplice è motivato dalla natura già ben distribuita della variabile e dall'assenza di valori anomali significativi.

- **Feature categoriche:** `speaker` e `subject`
  Entrambe le variabili categoriche sono state inizialmente trasformate tramite `Label Encoder`, che associa a ciascuna categoria un valore intero univoco. Tuttavia, poiché si hanno tanti valori differenti, portando ad avere numeri di encoding elevati, è stato applicato un ulteriore step di normalizzazione con lo `StandardScaler`.

L'utilizzo degli scaler ha due principali motivazioni: in primo luogo, standardizzare le distribuzioni ha reso le codifiche numeriche più neutre rispetto al modello, limitando l'influenza dalle label create con l'encoding. In secondo luogo, lo scaling ha permesso di riportare le feature aggiuntive su una scala comparabile con quella degli embedding ridotti tramite PCA, evitando che differenze di ordine di grandezza tra le variabili potessero compromettere la stabilità e l'efficacia dell'apprendimento del classificatore.

Una volta preprocessate tutte le features, sono state concatenate all'embedding testuale ridotto tramite PCA (50 componenti), formando un vettore di input esteso fornito alla rete (50 componenti + 3 features).

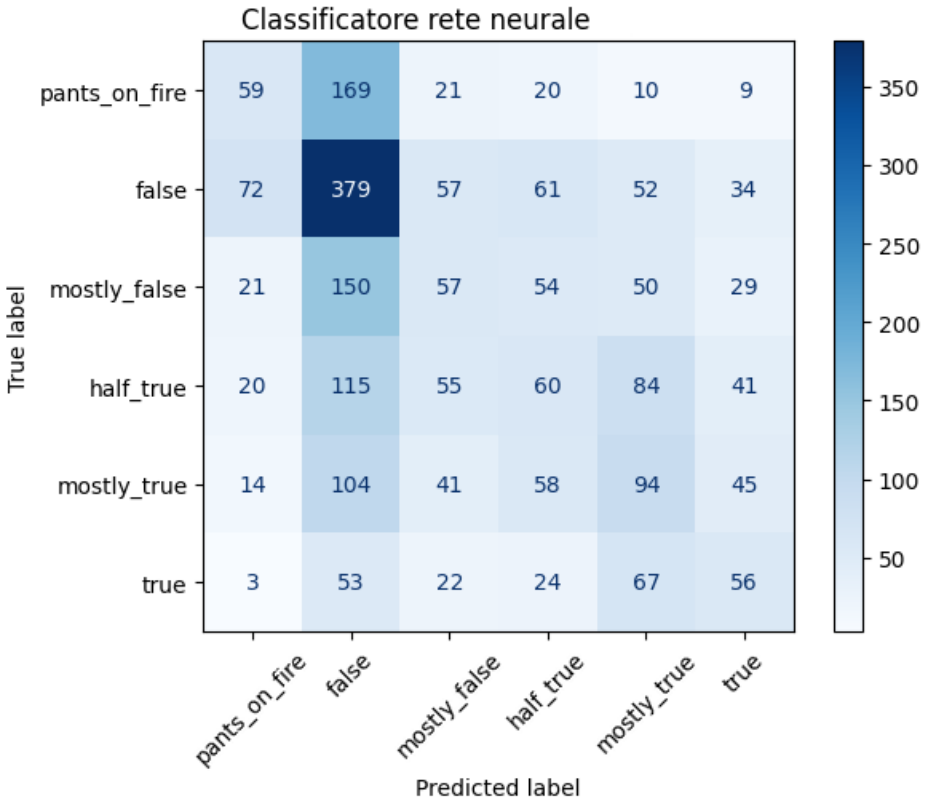


Figura 18: Matrice di confusione classificazione con features aggiuntive

### 4.1.6 Risultati classificazione con features aggiuntive

L'integrazione delle features selezionate (`speaker`, `subject` e `year`) con gli embedding testuali ridotti tramite PCA ha portato a un lieve miglioramento parziale della capacità predittiva del modello.

In particolare, l'accuratezza complessiva sul set di test ha raggiunto circa il 31% (31.19%), un risultato leggermente inferiore rispetto al modello che utilizzava solo le componenti della PCA ma comunque superiore ad altri approcci. Tuttavia, considerando un'aggregazione binaria delle classi (vero/falso), l'accuratezza aggregata si attesta a un circa 70% (66.99%), evidenziando che il modello è in grado di cogliere efficacemente la classificazione generale delle affermazioni.

La matrice di confusione in Figura 18 mostra una maggiore distribuzione corretta delle predizioni rispetto a classi vere, inclusa una leggera ripresa delle predizioni corrette della classe `true`, precedentemente trascurata. Nonostante ci siano ancora alcune difficoltà nel distinguere classi semanticamente vicine (ad esempio, `mostly_false` e `half_true`), l'aggiunta di informazioni extra ha probabilmente aiutato il modello a contestualizzare meglio alcune affermazioni, in particolare quelle provenienti da determinati speaker o soggetti ricorrenti.

In conclusione, l'esperimento conferma che la combinazione tra embedding testuali ridotti a 50 componenti e l'aggiunta di features ausiliarie riesce a fornire un contributo informativo utile al classificatore. Sebbene non cambi drasticamente le performance complessive, questo approccio si è rivelato particolarmente efficace nella classificazione binaria, portando ai migliori risultati ottenuti finora.

## 4.2 Fine-tuning BERT

Dopo aver sperimentato diverse configurazioni con classificatori, abbiamo condotto un esperimento di fine-tuning diretto del modello `BERT`, con l'obiettivo di adattare completamente i pesi del modello pre-addestrato al compito di classificazione.

Il modello di partenza utilizzato è `bert-base-uncased`, a cui è stato aggiunto uno strato finale per la classificazione in sei categorie di verità. Il fine-tuning è stato gestito tramite la libreria `Transformers` di Hugging Face, utilizzando la classe `Trainer` per semplificare processo di addestramento.

Sono stati esplorati due approcci per il fine-tuning del modello `BERT`, in particolare:

- **Approccio base (solo statement)**
  In questo approccio al modello è stato eseguito un fine-tuning esclusivamente sul contenuto testuale dello statement, ovvero la dichiarazione da classificare. È stato aggiunto un livello finale di classificazione a sei classi direttamente sopra BERT.

- **Approccio con concatenazione di features**
  In questa configurazione sperimentale, oltre allo statement, sono state concatenate direttamente al testo anche altre informazioni contestuali, come `speaker_description`, `subject` e `year` (utilizzando la forma testuale e non l'encoding visto precedentemente). L'intera sequenza, risultante dalla concatenazione, è stata poi utilizzata come input per il fine-tuning del modello.

### 4.2.1 Approccio base (solo statement)

Nel primo approccio, ciascun esempio è stato rappresentato unicamente dallo `statement`. Il testo è stato tokenizzato tramite il `tokenizer` associato a `bert-base-uncased`, per poi essere passato al modello BERT, a cui è stato aggiunto un livello per la classificazione. Il fine-tuning ha coinvolto l'intero modello, aggiornando anche i pesi interni di BERT durante l'addestramento.

### 4.2.2 Approccio con concatenazione

Nel secondo approccio abbiamo testato se l'aggiunta di informazioni aggiuntive potesse migliorare la performance del modello. Le stringhe dei campi `speaker_description`, `subject` e `year` sono state concatenate direttamente al testo principale, utilizzando il seguente schema:

```
"Statement:  " + statement + " Speaker:  " +
speaker_description + " Subject:  " + subject + " Date:
" + date
```

Questa sequenza estesa è stata poi fornita direttamente in input a BERT, dopo aver tokenizzato i testi, trattando tutte le informazioni come se facessero parte dello stesso testo. In questo modo, tutte le informazioni sono state trattate in modo omogeneo come testo, permettendo a BERT di apprendere eventuali correlazioni semantiche tra i contenuti.

### 4.2.3 Limitazioni e scelte progettuali

Non sono stati esplorati approcci più sofisticati per la gestione del fine-tuning, principalmente per motivi computazionali. Il fine-tuning di BERT richiede già un carico elevato in termini di memoria e tempo, provando metodi più articolati la GPU fornita non era sufficiente (GPU gratuita disponibile su Google Colab). Per questo motivo, ci siamo concentrati su strategie che ci permettessero di mantenere il carico computazionale entro i limiti consenti.

### 4.2.4 Risultati

Il fine-tuning del modello BERT ha portato a risultati migliori rispetto agli approcci basati su classificatori esterni. Sono state valutate le due tecniche di fine-tuning utilizzate:

- **Modello BERT fine-tunato solo su statement**
  In questa configurazione, il modello è stato addestrato esclusivamente sulla dichiarazione testuale. L'accuratezza ottenuta sul set di test è stata pari a 35.71%, mentre la classificazione aggregata (vero/falso) ha raggiunto un valore pari a 72.17%.

- **Modello BERT con concatenazione di features**
  Utilizzando la sequenza estesa precedentemente analizzata, l'accuratezza sul set di test è salita a 37.98%, mentre l'accuratezza aggregata ha raggiunto 73.61%. Questo suggerisce che, sebbene l'aggiunta testuale delle features non comporti un miglioramento drastico, può comunque fornire un contributo utile all'accuratezza finale del modello.

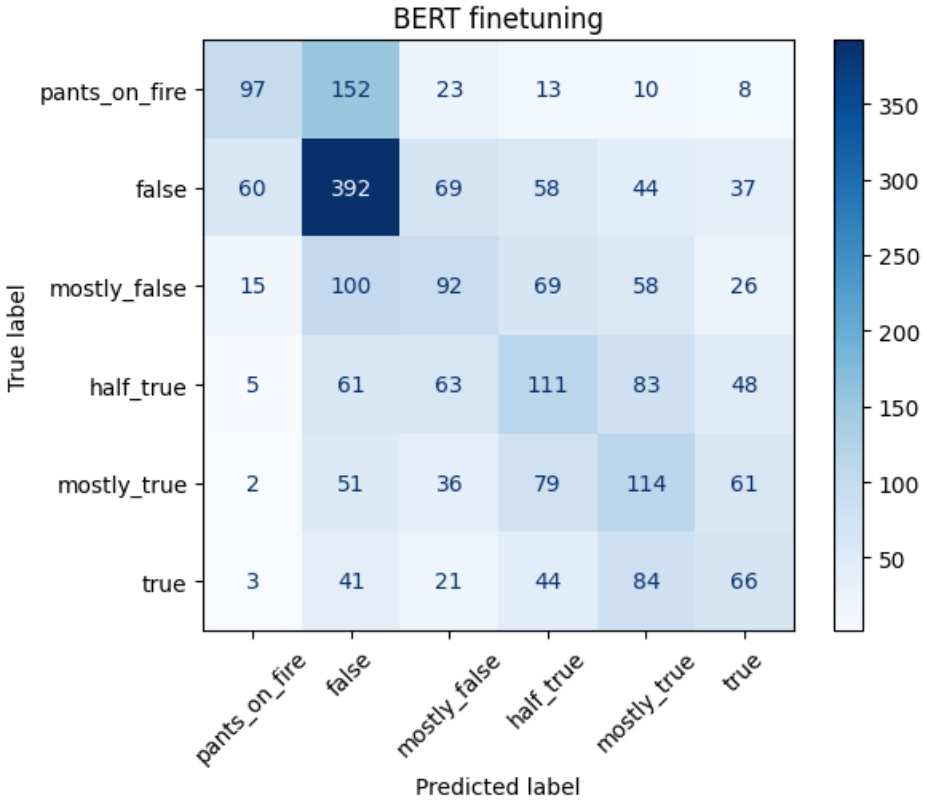


Figura 19: Matrice di confusione fine-tuning BERT con concatenazione features

La matrice di confusione riportata in Figura 19 mostra le prestazioni del modello BERT con concatenazione delle features. Le classi `false` e `pants_on_fire` sono quelle meglio riconosciute, con un numero elevato di predizioni corrette rispetto alle altre classi. Tuttavia, anche in questo caso, c'è sovrapposizione tra classi semanticamente vicine anche se minore agli scenari precedenti. Ad esempio, le classi `half_true`, `mostly_true` e `mostly_false` sono quelle con maggiore sovrapposizione.

In generale, la distribuzione degli errori è coerente con quella osservata in altri modelli, ma con un numero maggiore di classificazioni corrette, in particolare per le classi più rappresentate nel dataset. Il miglioramento nella classificazione aggregata (vero/falso) rispetto al modello base conferma che l'aggiunta di informazioni contestuali, anche se gestita in modo rudimentale, può offrire benefici in termini di accuratezza. Nella Figura 20 si può vedere la matrice di confusione ottenuta con le classi predette aggregate tra vero e falso.

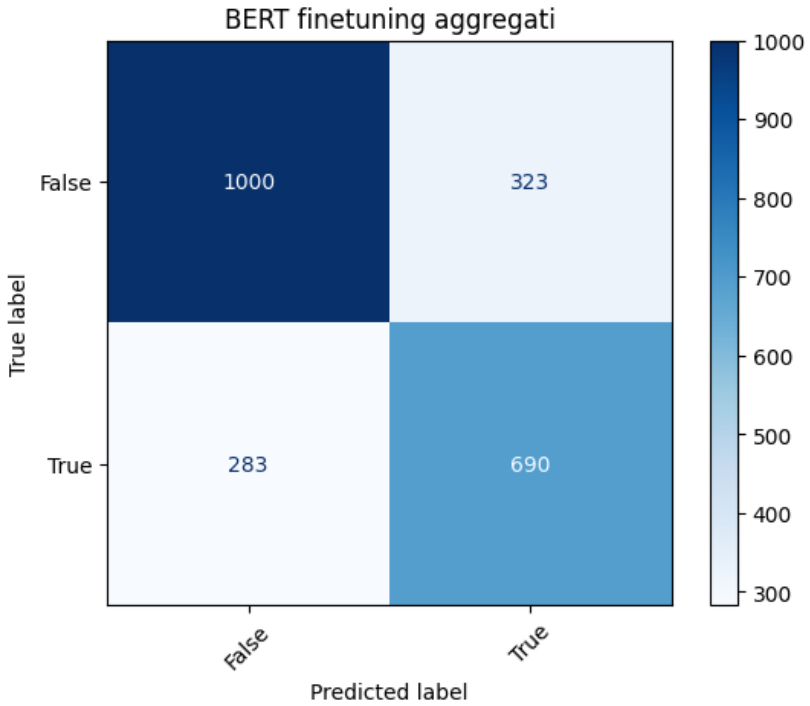


Figura 20: Matrice di confusione fine-tuning BERT con concatenazione features con risultati aggregati

Entrambe le varianti basate su fine-tuning diretto del modello linguistico superano ampiamente le performance ottenute tramite classificatori applicati su embedding. L'integrazione delle informazioni nel testo mostra un potenziale migliorativo, seppur limitato, anche in assenza di strategie più avanzate. In generale, il fine-tuning ha dimostrato di essere una soluzione efficace per migliorare le performance del modello.

## 4.3 Confronto

Nel classificare le notizie con sei livelli di verità, abbiamo esplorato due approcci principali basati su BERT: come estrattore di feature per classificatori esterni e tramite fine-tuning diretto.

In particolare, utilizzando BERT come estrattore di feature, il modello migliore è basato su una rete neurale con due layer nascosti. Questa configurazione ha raggiunto un'accuratezza del 32% per la classificazione a sei classi mentre del 68,2% per la classificazione aggregata (vero/falso). Abbiamo anche notato un leggero miglioramento utilizzando la PCA per ridurre gli embedding a 50 componenti nel caso di classificazione aggregata, raggiungendo un'accuratezza di circa 67%.

Nonostante questi approcci abbiano portato risultati incentivanti, l'approccio con fine-tuning ci ha permesso di migliorare la classificazione. Addestrato solo con gli statement, il modello ha raggiunto quasi il 36% di accuratezza sulle sei classi e il 72% per la classificazione aggregata.

Il miglior risultato in assoluto è stato ottenuto con il fine-tuning di BERT che integrava feature aggiuntive concatenate direttamente nel testo di input. Questo approccio ha portato l'accuratezza al 38% per le sei classi e quasi al 73% per la classificazione aggregata vero/falso.

# 5 Conclusioni

## 5.1 Discussione dei risultati rispetto agli obiettivi del progetto

Lo sviluppo del progetto ha come obiettivo analizzare la disinformazione tramite l'analisi delle affermazioni presenti nel contesto politico. In particolare, sono state esplorate tre principali direzioni di ricerca, corrispondenti alle domande iniziali che hanno guidato il progetto. Di seguito riportiamo le riflessioni e i risultati ottenuti per ciascuna di esse.

**Come si evolvono le notizie nel tempo e qual è la loro correlazione con gli eventi politici?**

Tramite gli strumenti di analisi dinamica (dashboard) è stato possibile comprendere come la quantità di affermazioni false è evoluta nel corso degli anni. Come illustrato precedentemente, si può notare come ci sia un continuo incremento di notizie non attendibili con il passare del tempo. Una delle cause principali di questo trend è dovuta alla diffusione dei social network.

L'analisi evidenzia che ci sono momenti storici particolari per cui si possono notare incrementi nel numero di affermazioni etichettate come false. Questi momenti storici sono collegati a diversi fattori all'interno della società politica americana. Ad esempio, a partire da inizio 2016 si nota un forte incremento delle notizie false dovuto all'intensificazione dell'attività delle organizzazioni di *fact-checking* (tra il 2015 e il 2016), che ha portato a una identificazione e catalogazione delle affermazioni false o ingannevoli.

Oltre a ciò, tutti i picchi in corrispondenza di periodi con molta attività di comunicazione e alti tassi di disinformazione sono corrispondenti agli ultimi processi di elezione, in particolare nel 2012, 2016 e 2020. Un'ulteriore evidenza di questa dinamica emerge dall'analisi per singolo speaker. Ad esempio, Donald Trump mostra una percentuale di affermazioni false costantemente superiore al 65%, con un trend in crescita a partire dal 2018, mentre Joe Biden mantiene valori di disinformazione generalmente più bassi e sotto il 50%. Questi risultati evidenziano come la veridicità delle informazioni sia fortemente influenzata dal soggetto che le diffonde, oltre che dal contesto temporale.

Dunque, possiamo dire che è presente una forte correlazione tra eventi politici importanti e il tasso di disinformazione, anche dovuto ad una maggiore attività delle persone coinvolte e dei canali social. Oltre a ciò, si può affermare che il tasso di dichiarazioni fatte che non sono attendibili è in continua crescita, causato da una maggiore propagazione di contenuti falsi, in particolar modo a causa dei social network.

**Esistono pattern linguistici distintivi associati alle classificazioni delle affermazioni?**

In seguito all'analisi che abbiamo condotto, possiamo sostenere che da quanto è emerso, non è possibile individuare una correlazione semantica sufficientemente marcata tra il contenuto delle affermazioni e la classificazione associata (intesa come veridicità o falsità lungo le diverse sfumature previste dal dataset). In particolare, le informazioni testuali contenute nelle affermazioni non sembrano presentare pattern linguistici o semantici sistematicamente distintivi, in grado di discriminare in maniera affidabile tra le diverse classi, suggerendo una complessità intrinseca nella relazione tra linguaggio utilizzato e giudizio di veridicità.

**È possibile sviluppare un classificatore affidabile per distinguere le notizie vere da quelle false?**

L'addestramento di diversi modelli ha dimostrato che è possibile ottenere performance accettabili, soprattutto nel contesto di una classificazione binaria (vero/falso), dove l'accuratezza ha raggiunto circa il 73% nel caso del modello BERT fine-tunato con concatenazione di informazioni ausiliarie. Dunque, per rispondere alla domanda, è possibile costruire un classificatore che riesce mediamente e predire in modo corretto 3 affermazioni su 4, se consideriamo di classificare una affermazione in tema politico come vera o false. Se si vuole aspirare ad un classificatore accurato per le 6 non si ha una affidabilità che sia sufficiente per una classificazione.

## 5.2 Considerazioni finali

Nel complesso, il progetto ha mostrato che modelli linguistici avanzati, come BERT, offrono una solida base per l'analisi automatica della veridicità delle affermazioni. L'integrazione di feature ausiliarie e tecniche di riduzione dimensionale ha portato a miglioramenti parziali, mentre il fine-tuning diretto del modello ha dimostrato il maggiore potenziale. Tuttavia, le limitazioni computazionali hanno imposto scelte progettuali semplificate, come la gestione testuale delle informazioni strutturate.

Nel completo, il lavoro ha dimostrato come sia possibile estrarre valore informativo da dati, combinando metodi statistici, tecniche di NLP e strumenti di visualizzazione interattiva.

# Riferimenti bibliografici


[1] C. Xu and M.-T. Kechadi, "An enhanced fake news detection system with fuzzy deep learning," *IEEE Access*, vol. 12, pp. 88 006–88 021, 2024.

[2] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[3] T. Sainburg, L. McInnes, and T. Q. Gentner, "Parametric umap embeddings for representation and semisupervised learning," *Neural Computation*, vol. 33, no. 11, pp. 2881–2907, 2021.

[4] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020. [Online]. Available: https://arxiv.org/abs/1910.01108

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805