Progetto Machine Learning – Fake News Detection

Colab Link:
https://colab.research.google.com/drive/1N52su2YBX0eNUsLstwgwAQXaRfUJDm0O?usp=sharing

Gruppo composto da:

- Pulerà Francesca 870005
- Zarantonello Massimo 866457

Indice

1. Descrizione del Progetto

Le fake news costituiscono una sfida rilevante nell'attuale panorama mediatico. Con l'emergere dei social media e la semplicità con cui le informazioni si diffondono online, le fake news sono presenti in ogni ambito, con possibili impatti negativi sulla fiducia pubblica e sul benessere sociale. Incidenti come elezioni influenzate, disinformazione riguardante la salute pubblica e danni alla reputazione individuale sono solo alcune delle manifestazioni delle fake news nella società contemporanea.

1.1. Introduzione

Riconoscere e contrastare le fake news è quindi diventato imperativo per preservare la verità e l'integrità dell'informazione. La classificazione accurata delle notizie è fondamentale, poiché consente di identificare e isolare le informazioni false, proteggendo così il pubblico da potenziali danni. L'apprendimento automatico offre uno strumento potente per affrontare questa sfida, grazie alla sua capacità di analizzare grandi quantità di dati testuali e identificare modelli significativi.

Per il presente progetto, è stato scelto di adoperare un dataset testuale reperito su Kaggle (consultabile al link: https://www.kaggle.com/datasets/jainpooja/fake-news-detection) specificamente incentrato sul rilevamento delle fake news. Da qui il titolo "Fake News Detection".

1.2. Obiettivi

Gli obiettivi principali del progetto includono l'implementazione e successiva analisi di tre modelli di classificazione supervisionata del testo, in grado di discriminare tra fake news e notizie autentiche. Inoltre, si mira a comprendere le caratteristiche linguistiche distintive delle fake news al fine di migliorare la capacità del modello di identificarle in modo accurato e affidabile.

1.3. Struttura della relazione

La relazione sarà strutturata in modo da fornire una panoramica completa del progetto, includendo una descrizione dettagliata della metodologia utilizzata, i risultati ottenuti e le future direzioni della ricerca.

2. Analisi del Dataset

Il dataset utilizzato per questo progetto è stato accuratamente selezionato per garantire una distribuzione bilanciata tra notizie autentiche e notizie false ottenute da articoli online.

2.1. Unione dei due dataset “fake” e “true”

Originariamente, il dataset consisteva di due insiemi distinti: uno contenente notizie vere e l'altro contenente fake news. Entrambi presentavano la medesima struttura composta da quattro colonne (escludendo l'indice): "**title**" per il titolo dell'articolo, "**text**" per il contenuto testuale dell'articolo, "**subject**" per il tema trattato e "**date**" per la data di pubblicazione dell'articolo. Per uniformare la struttura e consentire un’analisi coerente, si è scelto di aggiungere una colonna “**class**” a ciascun dataset, che indicasse la classe di appartenenza di ogni documento: **0** per la classe “**fake**” e **1** per la classe “**true**”.

Successivamente, questi due insiemi sono stati integrati in un'unica fonte dati, le cui righe sono state casualmente permutate al fine di permettere un mescolamento delle notizie reali e false, ottenendo così un dataset di 44898 istanze che si presenta in questa forma:

index	title	text	subject	date	class
0	Donald Trump Sends Out Embarrassing New Year’s Eve Message; This is Disturbing	Donald Trump just couldn t wish all Americans a Happy New Year and leave it at that. Instead, he had to give a shout out to his enemies, haters and the very dishonest fake news media. The former reality show star had just one job to do and he couldn t do it. As our Country rapidly grows stronger and smarter, I want to wish all of my friends, supporters, enemies, haters, and even the very dishonest Fake News Media, a Happy and Healthy New Year, President Angry Pants tweeted. 2018 will be a great year for America! As our Country rapidly grows stronger and smarter, I want to wish all of my friends, supporters, enemies, haters, and even the very dishonest Fake News Media, a Happy and Healthy New Year. 2018 will be a great year for America! Donald J. Trump (@realDonaldTrump) ……..	News	December 31, 2017	0
1	U.S., North Korea clash at U.N. forum over nuclear weapons	GENEVA (Reuters) - North Korea and the United States clashed at a U.N. forum on Tuesday over their military intentions towards one another, with Pyongyang s envoy declaring it would never put its nuclear deterrent on the negotiating table. Japan, well within reach of North Korea s missiles, said the world must maintain pressure on the reclusive country to rein in its nuclear and missile programs and now was not the time for a resumption of multi-party talks. North Korea has pursued its weapons programs in defiance of U.N. Security Council sanctions and ignored all calls, including from major ally China, to stop, prompting a bellicose exchange of rhetoric between the North and the United States. North Korea justifies its weapons programs, including its recent threat to fire missiles towards the U.S. Pacific territory of Guam, by pointing to perceived U.S. hostility, such as military exercises with South Korea this week. U.S. disarmament ambassador Robert …….	worldnews	August 22, 2017	1
2	Sheriff David Clarke Becomes An Internet Joke For Threatening To Poke People ‘In The Eye’	On Friday, it was revealed that former Milwaukee Sheriff David Clarke, who was being considered for Homeland Security Secretary in Donald Trump s administration, has an email scandal of his own.In January, there was a brief run-in on a plane between Clarke and fellow passenger Dan Black, who he later had detained by the police for no reason whatsoever, except that maybe his feelings were hurt. Clarke messaged the police to stop Black after he deplaned, and now, a search warrant has been executed by the FBI to see the exchanges.Clarke is calling it fake news even though copies of the search warrant are on the Internet. I am UNINTIMIDATED by lib media attempts to smear and discredit me with their ……	News	December 30, 2017	0

Tabella 1: Prime tre righe del dataset generato dall'unione di Fake e True

Ciò che si è ritenuto importante fare in una fase preliminare è stata proprio l'analisi del dataset, esplorandone il contenuto:

- Inizialmente, ci si è focalizzati sull'**analisi del tipo di notizie** presenti nel dataset. Come illustrato nei grafici qui di seguito (*Figura 2* e *Figura 3*), ottenuti dall'estrazione della feature "subject" del dataset stesso, si può osservare che le notizie sono prevalentemente di natura politica.

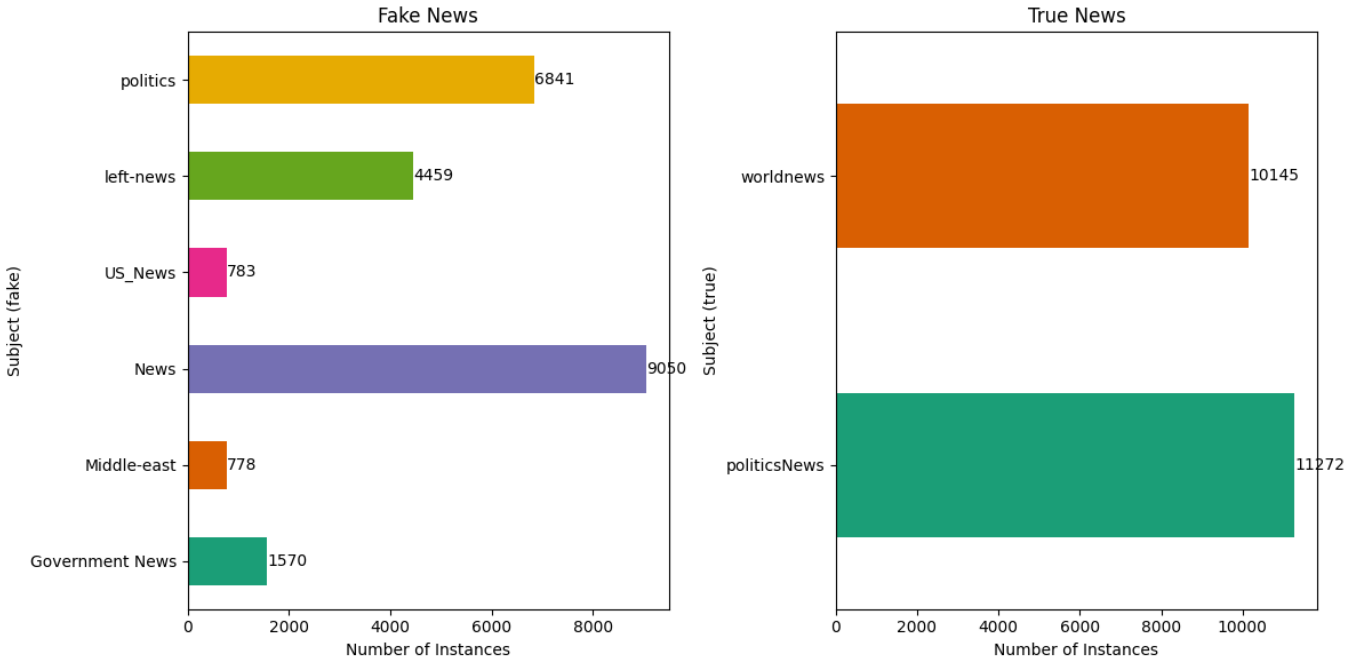


Figura 1: Raggruppamento per 'subject' dei due dataset distinti e calcolo delle dimensioni di ciascun gruppo per le notizie false e vere

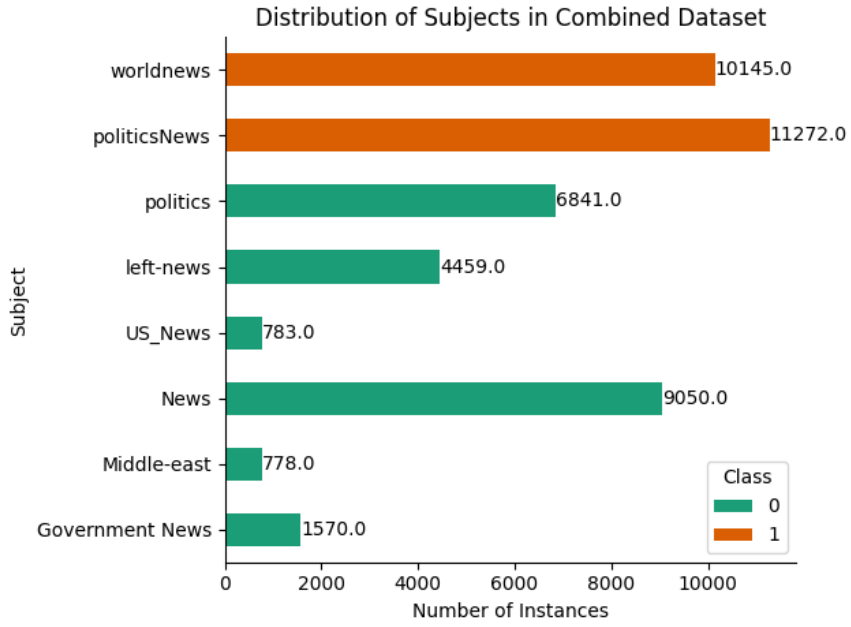


Figura 2: Raggruppamento per 'subject' e calcolo delle dimensioni di ciascun gruppo per le notizie false e vere

- Successivamente, è stata condotta un'analisi della lunghezza di ciascun articolo nel dataset al fine di valutare la loro complessità e completezza. Questo ha permesso di identificare gli articoli che potrebbero essere considerati troppo brevi o "poveri" di contenuto per essere adeguatamente valutati come reali o falsi. Tale analisi ha contribuito a garantire che solo gli articoli con una lunghezza significativa e rilevante fossero inclusi nel processo di valutazione e classificazione delle fake news.
  Si sono aggiunte due colonne al dataset per poter selezionare ed individuare più rapidamente le notizie in base alla loro lunghezza:
  1) "text_length", che indica la lunghezza del testo in termini di caratteri
  2) "word_count", che indica la quantità di parole di cui è composto l'articolo.

Statistiche sulla lunghezza dei testi:	
count	44898
mean	2469,109693
std	2.171.617.091
min	1.000.000
25%	1.234.000.000
50%	2.186.000.000
75%	3.105.000.000
max	51.794.000.000

Tabella 2: Statistiche descrittive della lunghezza dei testi in termini di caratteri

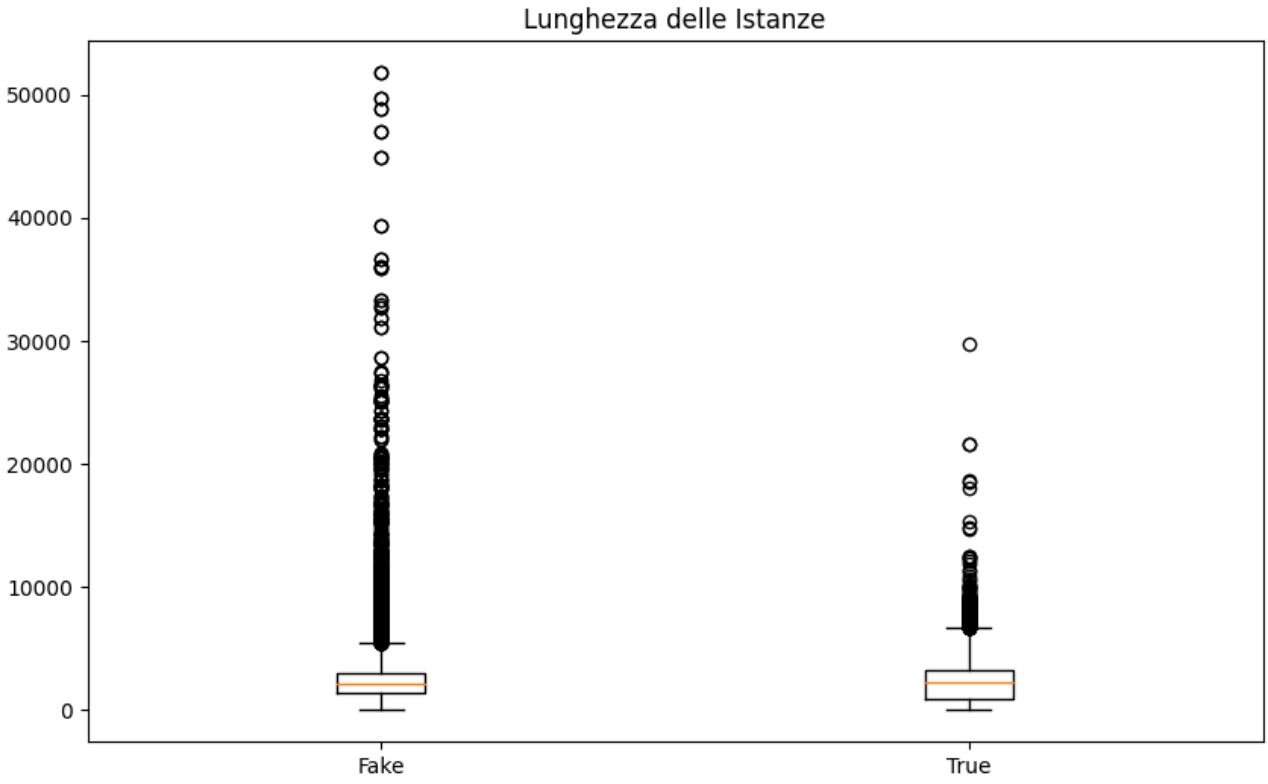


Figura 3: Boxplot della lunghezza degli articoli del dataset

Nella *Figura 5* è presentato un boxplot che illustra la distribuzione della lunghezza delle istanze nel dataset. Analizzando il grafico, emergono evidenti punti al di fuori dei baffi del boxplot, indicativi di outliers nelle lunghezze dei testi. È interessante notare che questi outliers sono più frequenti nei testi relativi alle fakenews rispetto a quelli delle notizie vere. Questo fenomeno potrebbe essere dovuto a diversi fattori. Ad esempio, le fakenews potrebbero tendere ad essere più sensazionalistiche o contenere informazioni aggiuntive o non rilevanti, aumentando così la loro lunghezza rispetto alle notizie reali. Tuttavia, è importante esaminare attentamente la natura di questi outliers per comprendere se rappresentano errori nei dati, casi eccezionali o caratteristiche legittime del fenomeno in studio. La presenza di outliers nelle fakenews potrebbe essere utile nell'identificare pattern distintivi o caratteristiche peculiari di questo tipo di contenuti, ma richiede ulteriori analisi per determinare la loro rilevanza e impatto sull'obiettivo dell'analisi.

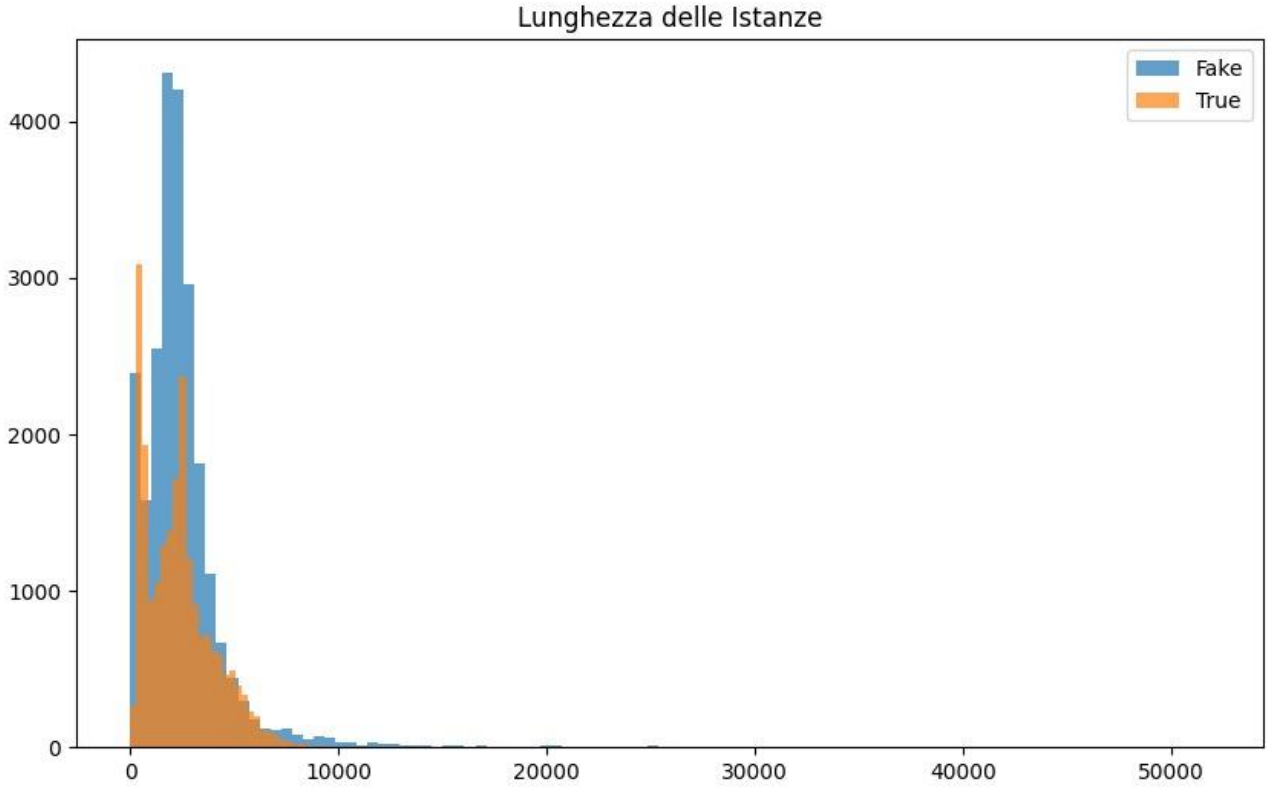


Figura 4: Istogramma della lunghezza dei testi in termini di caratteri

2.2. Rimozione degli articoli con meno di 20 parole

In seguito, si è deciso di rimuovere gli articoli che contenevano meno di 20 parole, identificandone precisamente 1131. Questa operazione ha portato ad alleggerire leggermente il dataset, seppur in misura limitata. La decisione di eliminare gli articoli con una lunghezza così ridotta è stata presa al fine di garantire che i dati fossero sufficientemente informativi e rappresentativi per l'analisi successiva. Nonostante la riduzione modesta del dataset, questa operazione ci ha permesso di concentrarci sui documenti più significativi e di migliorare la qualità complessiva dei dati utilizzati nell'analisi.

2.3. Rimozione di features

Per la stessa ragione delineata nel paragrafo precedente, abbiamo eliminato alcune delle feature che non contribuivano all'obiettivo di classificazione delle notizie. In particolare, abbiamo rimosso le colonne "title", "data", "subject", "word_count" e "text_length". Questa scelta ci ha consentito di ridurre il peso del dataset, concentrandoci sulle feature più informative e rilevanti, ovvero "text" e "class".

Si ottiene quindi il dataset “news” di questo tipo:

index	text	class
0	Donald Trump just couldn t wish all Americans ...	0
1	House Intelligence Committee Chairman Devin Nu...	0
2	BRUSSELS (Reuters) - NATO allies on Tuesday we...	1
3	LONDON (Reuters) - LexisNexis, a provider of l...	1

Tabella 3: Prime quattro righe del dataset “news”

Conclusivamente, il dataset è composto da due feature e un totale di 43,729 istanze, corrispondenti agli articoli analizzati. Tra queste, 22,313 sono identificate come fake news e 21,416 come real news, determinando così un dataset bilanciato in termini di distribuzione delle classi (*Figura 8*).

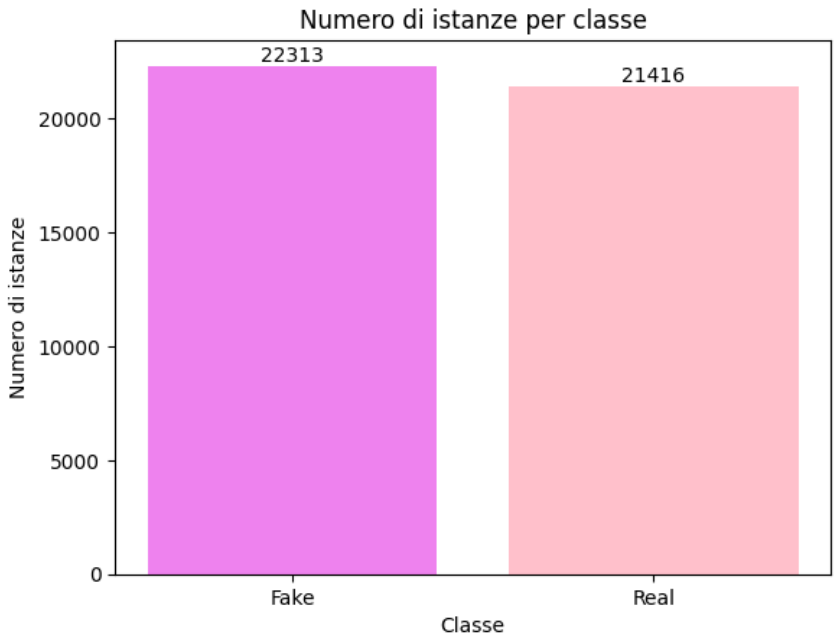


Figura 5: numero di istanze per ciascuna classe

3. Preprocessing

I dati di tipo testuale richiedono una fase di preparazione prima di poter essere elaborati. Questa prima revisione dei dati è essenziale per poter eliminare dai documenti delle parole che non danno alcun contributo utile al fine di elaborazioni successive.

Prima di procedere alla modellazione dei testi è dunque necessaria una fase di preparazione e pulizia dei dati; le operazioni più importanti sono la rimozione della punteggiatura, la lemmatizzazione, la tokenizzazione e la rimozione delle *stopword*.

3.1. Conversione in lowercase

Si converte il testo in minuscolo per garantire una rappresentazione uniforme del testo. In questo modo, parole con la stessa forma ma scritte in maiuscolo o minuscolo verranno trattate allo stesso modo durante l'analisi, evitando ambiguità e semplificando il processo di elaborazione del testo.

La conversione in minuscolo, inoltre, aiuta a ridurre la varianza nei dati testuali, poiché le parole che differiscono solo per la maiuscola o minuscola vengono considerate equivalenti.

PRIMA:

```
0    If there s one thing this election succeeded i...
1    MADRID (Reuters) - A Spanish audit office has ...
2    WASHINGTON (Reuters) - Some prominent Republic...
3    Who s laughing now  funny  guy?We asked everyo...
4    Wow! This young Asian student nails it! He spe...
```

DOPO:

```
0    if there s one thing this election succeeded i...
1    madrid (reuters) - a spanish audit office has ...
2    washington (reuters) - some prominent republic...
3    who s laughing now  funny  guy?we asked everyo...
4    wow! this young asian student nails it! he spe...
```

3.2. Creazione di una funzione che processa il testo

Si introduce una funzione, denominata *processTexts*, la quale esegue una serie di operazioni di pre-processing su un testo in linguaggio naturale. Qui di seguito una spiegazione e argomentazione delle singole operazioni:

- Sostituzione di '\n' con uno spazio vuoto: la sequenza '\n' indica un carattere di nuova riga. La sostituzione di questo carattere con uno spazio vuoto (' ') è utile per

rimuovere le interruzioni di linea, che possono interferire con l'analisi del testo. Ad esempio, se il testo originale contiene una nuova riga dopo ogni parola, la sostituzione di '\n' con uno spazio consente di trattare l'intero testo come una singola sequenza continua di parole.

- Rimozione del testo racchiuso tra parentesi quadre: questo passaggio utilizza espressioni regolari per rimuovere tutti i testi racchiusi tra parentesi quadre, inclusi eventuali contenuti testuali o codice sorgente. Questo può essere utile per eliminare note, citazioni, o informazioni aggiuntive che non sono rilevanti per l'analisi e interferiscono con i modelli di apprendimento automatico.
- Rimozione dei link URL: utilizzando nuovamente le espressioni regolari, questa operazione rimuove tutti i link URL presenti nel testo. Questo è utile per eliminare collegamenti ipertestuali, indirizzi web o altri dati strutturati che non contribuiscono alla comprensione del contenuto testuale e interferiscono con l'analisi.
- Rimozione dei tag HTML: questa operazione utilizza nuovamente le espressioni regolari per eliminare tutti i tag HTML presenti nel testo. Questi tag possono includere formattazione, struttura o metadati aggiuntivi presenti nei dati testuali, che in questo caso non sono rilevanti per l'analisi del testo e potrebbero interferire con i modelli di elaborazione del linguaggio naturale.
- Rimozione dei numeri: questo passaggio utilizza nuovamente le espressioni regolari per rimuovere tutti i caratteri numerici presenti nel testo. Questo è utile quando si desidera analizzare solo il contenuto testuale del testo e si vogliono eliminare numeri, cifre o altri dati numerici.
- Rimozione del carattere '_': questo passaggio semplicemente rimuove il carattere underscore ('_') dal testo. Questo carattere è presente molto spesso nei dati testuali, ma non è rilevante per l'analisi del testo.

3.3. Rimozione della punteggiatura

La rimozione della punteggiatura è spesso una pratica comune nell'elaborazione del linguaggio naturale (NLP) per diversi motivi:

- Riduzione del rumore: la punteggiatura, come virgole, punti, parentesi, ecc., non aggiunge necessariamente significato al testo e può essere considerata "rumore" durante l'analisi del testo. Rimuovendo la punteggiatura, si riduce il rumore nei dati testuali, consentendo all'algoritmo di concentrarsi sulle parole significative.
- Uniformità: rimuovendo la punteggiatura, si ottiene una rappresentazione del testo più uniforme e pulita. Ciò semplifica il processo di tokenizzazione e analisi del testo, in quanto non è necessario gestire i segni di punteggiatura come token separati.
- Dimensionalità ridotta: mantenere la punteggiatura potrebbe portare a un aumento della dimensionalità dei dati, specialmente in grandi corpus di testo. Rimuovendo la punteggiatura, si riduce il numero di caratteri distinti da considerare durante l'analisi, semplificando così il processo computazionale e riducendo la complessità del modello.
- Prevenzione dell'overfitting: la presenza della punteggiatura potrebbe portare a un addestramento eccessivamente specifico su caratteristiche non rilevanti nei dati

testuali. Rimuovendo la punteggiatura, si riduce la possibilità di overfitting, consentendo al modello di generalizzare meglio sui dati di test.

PRIMA:

```
0  if there s one thing this election succeeded i...
1  madrid (reuters) - a spanish audit office has ...
2  washington (reuters) - some prominent republic...
3  who s laughing now  funny  guy?we asked everyo...
4  wow! this young asian student nails it! he spe...
```

DOPO:

```
0  if there s one thing this election succeeded i...
1  madrid reuters  a spanish audit office has dem...
2  washington reuters  some prominent republicans...
3  who s laughing now  funny  guywe asked everyon...
4  wow this young asian student nails it he speak...
```

3.4. Tokenizzazione

La tokenizzazione, nel contesto del trattamento del linguaggio naturale (NLP), è il processo di suddividere un testo in unità più piccole, chiamate token. Un token può essere una parola, una frase, un simbolo o qualsiasi altra unità significativa all'interno del testo.

In questo lavoro di analisi del testo, si utilizza il word_tokenizer, uno strumento fornito da librerie NLP come NLTK (Natural Language Toolkit), per eseguire la tokenizzazione delle parole. Questo è un passaggio fondamentale nel pre-processing del testo, e ha diversi obiettivi e benefici:

- Suddivisione in unità significative: la tokenizzazione suddivide il testo in unità di significato, come parole o simboli, facilitando l'analisi e l'interpretazione del testo.
- Standardizzazione del testo: la tokenizzazione aiuta a standardizzare il testo, consentendo di trattare ogni parola come un'unità separata indipendentemente dalla formattazione o dalla struttura del testo.
- Creazione di vocabolari: la tokenizzazione delle parole è fondamentale per costruire vocabolari di parole uniche presenti nel testo, che possono essere utilizzati per l'analisi delle frequenze delle parole, l'identificazione di pattern linguistici e altre operazioni di elaborazione del linguaggio naturale.
- Preparazione per l'elaborazione automatica: la tokenizzazione prepara il testo per l'elaborazione automatica da parte di algoritmi di machine learning e altre tecniche di analisi del testo. Trattando ogni parola come un'unità separata, è possibile rappresentare il testo in forma numerica, che può essere elaborata da algoritmi di machine learning per compiti come la classificazione del testo, l'analisi del sentiment o la generazione di testo.

PRIMA:

```
0  if there s one thing this election succeeded i...
1  madrid reuters  a spanish audit office has dem...
2  washington reuters  some prominent republicans...
```

```
3   who s laughing now  funny  guywe asked everyon...
4   wow this young asian student nails it he speak...
```

DOPO:

```
0   [if, there, s, one, thing, this, election, suc...
1   [madrid, reuters, a, spanish, audit, office, h...
2   [washington, reuters, some, prominent, republi...
3   [who, s, laughing, now, funny, guywe, asked, e...
4   [wow, this, young, asian, student, nails, it, ...
```

3.5. Rimozione di stopwords

Le stopwords sono un insieme di parole utilizzate frequentemente nel linguaggio e presenti in tutti i testi, quali articoli, congiunzioni, pronomi etc... Questi non aggiungono nessuna informazione riguardo l'argomento del quale si parla in un documento, anzi, porterebbero problemi di stima vista l'alta frequenza con cui vengono utilizzati, dunque vengono rimossi dal *corpus*. Nel contesto specifico del nostro caso, sono state considerate le stopwords della lingua inglese poiché i documenti presenti nel dataset erano scritti in inglese.

PRIMA:

```
0   [if, there, s, one, thing, this, election, suc...
1   [madrid, reuters, a, spanish, audit, office, h...
2   [washington, reuters, some, prominent, republi...
3   [who, s, laughing, now, funny, guywe, asked, e...
4   [wow, this, young, asian, student, nails, it, ...
```

DOPO:

```
0   [one, thing, election, succeeded, bringing, he...
1   [madrid, reuters, spanish, audit, office, dema...
2   [washington, reuters, prominent, republicans, ...
3   [laughing, funny, guywe, asked, everyone, boyc...
4   [wow, young, asian, student, nails, speaks, ev...
```

3.6. Lemmatizzazione

La lemmatizzazione, invece, ha lo scopo di trovare una radice comune, la quale rappresenta un insieme di parole che hanno approssimativamente lo stesso significato. Questo procedimento non si basa però semplicemente sul troncare la parola, ma su un meccanismo molto più complesso. Ad esempio, lemmatizzare la serie di parole correre, corro, corriamo e correremo, produrrebbe il lemma "correre". Questa pratica non è particolarmente complessa quando si tratta della lingua inglese, con la sua grammatica relativamente semplice e poche forme irregolari.

PRIMA:

```
4   [wow, young, asian, student, nails, speaks, ev...
```

DOPO:

3.7. Rimozione delle parole con meno di due caratteri

Le parole con meno di due caratteri sono spesso poco informative e non contribuiscono significativamente al significato o al contenuto del testo. Rimuovendo queste parole, è possibile ridurre il rumore nei dati e concentrarsi sulle parole più rilevanti per l'analisi.

3.8. Wordclouds

Di seguito sono presentate due wordclouds che rappresentano il dataset: una prima della fase di preprocessing e una dopo.

La prima wordcloud mostra il dataset così com'è, senza alcuna manipolazione. Questo include la presenza di punteggiatura, caratteri speciali e parole non lemmatizzate. La seconda wordcloud, invece, è stata generata dopo aver applicato il preprocessing al dataset, incluso il trattamento della punteggiatura, la lemmatizzazione delle parole e altre operazioni di pulizia del testo.

Queste visualizzazioni forniscono un'indicazione visiva delle parole più frequenti presenti nel dataset originale e nel dataset preprocessato, consentendo una comparazione diretta tra i due e mostrando come il preprocessing abbia contribuito a migliorare la qualità dei dati e a concentrarsi sulle informazioni rilevanti.

Figura 6: Wordcloud relativa al dataset prima del preprocessing

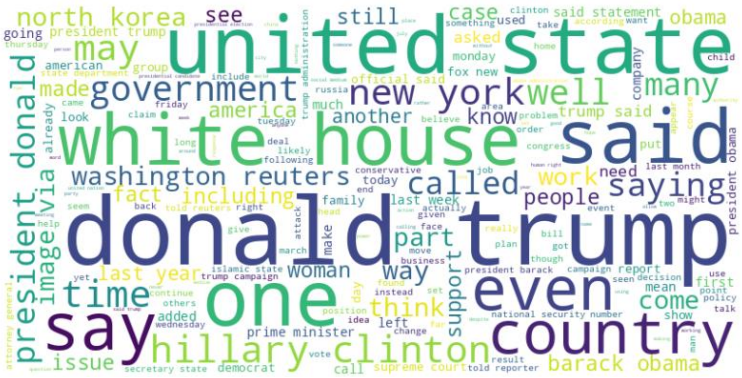


Figura 7: Wordcloud relativa al dataset dopo il preprocessing

Nel testo originale, prima del preprocessing, è evidente la presenza di singoli caratteri come "s" e parole di due caratteri come "us". Tuttavia, nella wordcloud derivante dal testo preprocessato, vengono considerate solo le parole con più di due caratteri. Inoltre, si osserva una significativa riduzione della frequenza di parole come "said" dopo il preprocessing, fenomeno spiegato dalla lemmatizzazione che considera il contesto di utilizzo delle parole per determinarne il lemma. Ad esempio, "saw" potrebbe essere lemmatizzato come "see" come verbo, ma come "saw" come sostantivo; allo stesso modo, "said" potrebbe essere lemmatizzato come "say" in alcuni contesti.

3.9. Frequenza delle parole

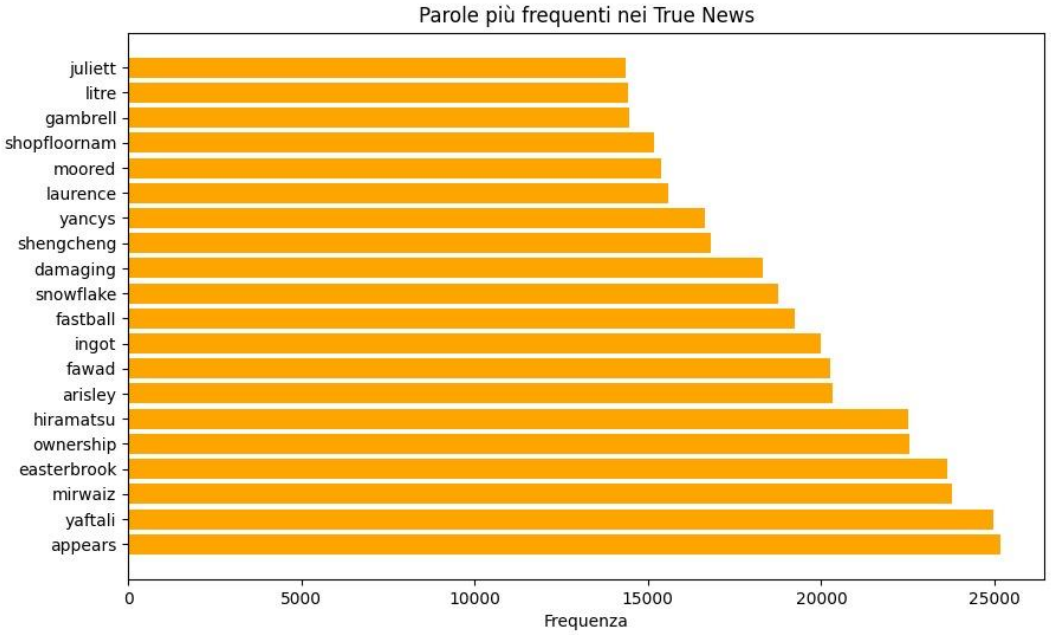


Figura 9: Grafico delle 20 parole più frequenti nel dataset True News

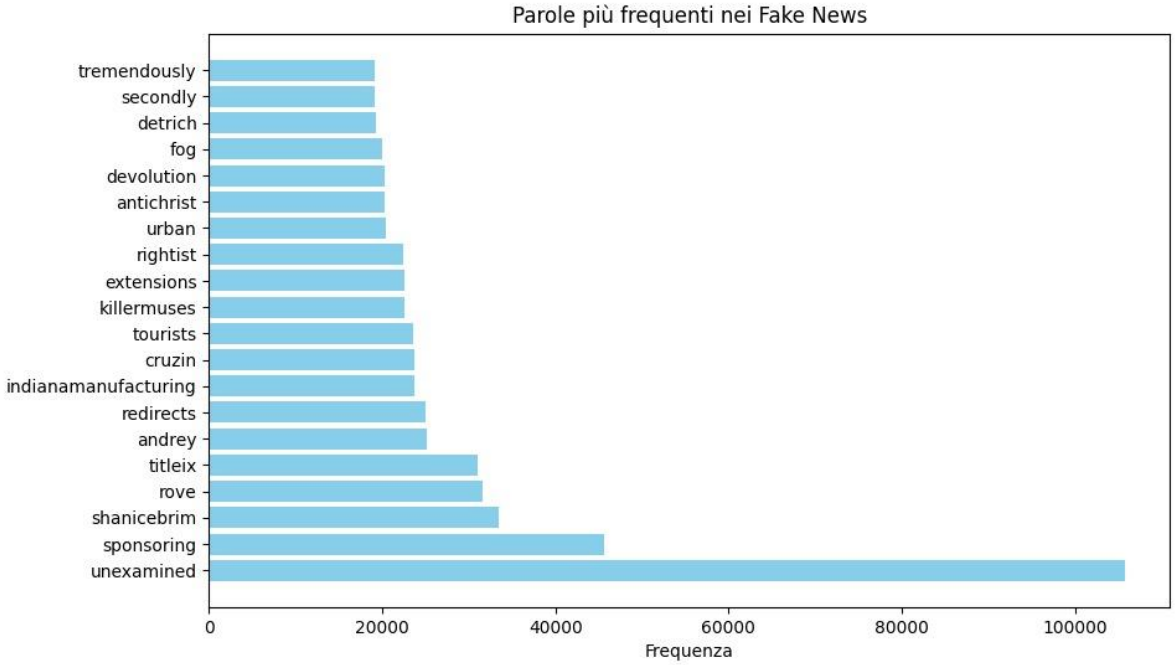


Figura 8: Grafico delle 20 parole più frequenti nel dataset Fake News

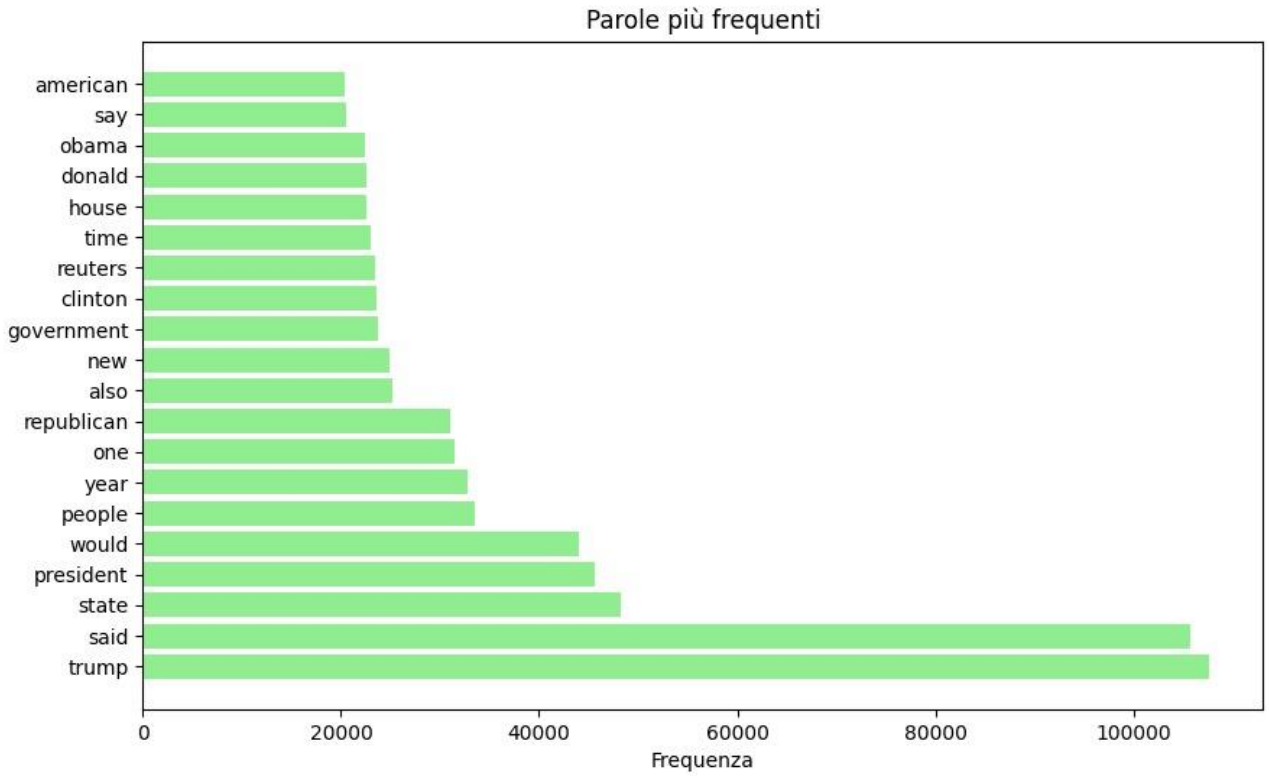


Figura 10: Grafico delle 20 parole più frequenti nel dataset News

Dai grafici presentati emerge una discrepanza significativa tra le parole più frequenti nei dataset delle Fake news e delle True News, con un'intersezione nulla tra i due. Questo fenomeno potrebbe suggerire l'esistenza di pattern distintivi e ricorrenti nelle fake news, che le differenziano notevolmente dalle notizie reali. Inoltre, si osserva che le frequenze delle parole più comuni nel dataset delle fake news sono molto più elevate rispetto a quelle delle True News, indicando una maggiore ripetitività o enfasi su certe parole o concetti all'interno delle fake news.

La scelta di eliminare le feature poco influenti per il progetto è considerata come una forma di feature selection (selezione delle feature più rilevanti o informative per il progetto), mentre le operazioni di tokenizzazione, lemmatizzazione e rimozione delle stopwords sono considerate tecniche di feature extraction (estrarre le caratteristiche significative dai dati testuali) nel processo di preprocessing di un dataset.

4. Addestramento dei Modelli

Dopo aver completato la fase di preprocessing del dataset, il capitolo procede con l'addestramento dei modelli per la classificazione delle fake news. Questa fase richiede la suddivisione del dataset stesso in un set di addestramento e un set di test, e la trasformazione degli input testuali in rappresentazioni numeriche, ovvero il processo noto come vettorizzazione.

Solo successivamente verranno addestrati i vari modelli utilizzando entrambe le tecniche di vettorizzazione descritte nel paragrafo 4.2.

4.1. Suddivisione in train e test

La suddivisione è essenziale per valutare l'efficacia e le prestazioni dei modelli di machine learning prima dell'applicazione pratica. Si è scelto di utilizzare l' 80% per addestrare i modelli, consentendo loro di apprendere i pattern e le caratteristiche dei dati. I restanti 20% dei dati vengono riservati per valutare le prestazioni dei modelli su dati non visti durante l'addestramento, fornendo così una stima imparziale della loro capacità di generalizzazione.

Questo approccio è cruciale per evitare il overfitting (sovrapprendimento), che si verifica quando un modello si adatta eccessivamente ai dati di addestramento e non è in grado di generalizzare correttamente su nuovi dati.

Dopo aver effettuato la suddivisione in set di addestramento e di test, si procede alla fase di validazione. Durante questa fase, che avviene dopo l'addestramento dei modelli, vengono utilizzate diverse tecniche di ottimizzazione e valutazione per selezionare i migliori iperparametri dei modelli e ottimizzarne le prestazioni.

Una delle tecniche più comuni utilizzate durante la fase di validazione è la grid search (utilizzata in questo progetto), che consiste nell'esplorare sistematicamente un insieme di combinazioni predefinite di iperparametri dei modelli al fine di identificare quella che massimizza le prestazioni del modello su un insieme di dati di validation. In questa fase, viene quindi effettuata una suddivisione ulteriore del set di addestramento in un insieme di dati di validation, che rappresenta generalmente un terzo dei dati di addestramento, e un insieme di dati di training rimanente.

Dunque, il nostro set di addestramento sarà composto da 34.982 record, mentre il set di test conterrà 8.746 record. Inoltre, nel set di addestramento, la distribuzione delle classi è approssimativamente del 51.15% per la classe 0 e del 48.85% per la classe 1. Nel set di test, la distribuzione è leggermente diversa, con circa il 50.53% dei record appartenenti alla classe 0 e il 49.47% alla classe 1.

Figura 12: Wordcloud dell'insieme di train

Figura 11: Wordcloud dell'insieme di test

L'osservazione dell'evidente presenza frequente delle parole "trump" e "said" nel test set rispetto al training set può essere attribuita a una casualità. Questa discrepanza potrebbe essere il risultato della naturale variazione dei dati, dove il 20% del dataset utilizzato per il test set comprende semplicemente un numero maggiore di testi riguardanti Donald Trump rispetto al training set. In altre parole, la distribuzione casuale dei documenti tra il training set e il test set potrebbe aver portato a una maggiore rappresentazione di certe parole nel test set rispetto al training set, senza che ciò rifletta necessariamente una differenza significativa nella distribuzione reale delle parole tra i due set.

4.2. Rappresentazione vettoriale del corpus

La vettorizzazione consiste nel tradurre le feature estratte in vettori numerici, che possono essere elaborati dai modelli di machine learning. Questa rappresentazione numerica consente ai modelli di applicare algoritmi matematici per analizzare e trovare pattern nei dati testuali, cosa che non sarebbe possibile con i dati in forma di testo grezzo.

Nel contesto del progetto si sono scelti due diversi vettorizzatori: CountVectorizer e TfidfVectorizer.

4.2.1. Count Vectorizer

Questa decisione è stata guidata dalla semplicità intrinseca della seguente tecnica e dalla sua capacità di adeguarsi alle specificità dell'ambito di studio. Entriamo nei dettagli sul suo funzionamento e sulle ragioni che ne hanno fatto l'opzione più adatta per questa classificazione binaria:

- CountVectorizer: è un metodo di vettorizzazione del testo che si basa sul calcolo delle occorrenze delle parole in un dato documento. Il processo inizia con la creazione di un vocabolario che contiene tutte le parole uniche all'interno del corpus di testo. Successivamente, ogni documento viene rappresentato come un vettore, in cui ogni elemento corrisponde alla frequenza di una parola all'interno del documento.
- Processo Operativo:
    - Costituzione del vocabolario: CountVectorizer inizia individuando tutte le parole uniche nel corpus e utilizzandole per creare un vocabolario.
    - Calcolo delle occorrenze: per ogni documento nel corpus, CountVectorizer conta le volte in cui ciascuna parola del vocabolario appare nel documento, generando un vettore di conteggio per ogni documento.
    - Rappresentazione numerica: ciò risulta in una matrice in cui ogni riga rappresenta un documento e ogni colonna rappresenta una parola del vocabolario, con i valori che indicano il numero di occorrenze della parola nel documento.
- Applicazione alla Classificazione di Fake News: diversi fattori specifici al contesto di classificazione binaria di fake news hanno motivato la scelta di CountVectorizer:
    - Conservazione delle Informazioni di Frequenza: la frequenza delle parole gioca un ruolo fondamentale nel distinguere tra notizie vere e false. CountVectorizer cattura queste informazioni, evidenziando le differenze chiave attraverso il conteggio delle occorrenze delle parole.
    - Adattabilità ai dati testuali semplici: nell'ambito delle notizie, le informazioni più importanti spesso derivano da parole chiave. CountVectorizer, contando le occorrenze delle parole, riesce a catturare efficacemente tali informazioni, contribuendo alla comprensione del contenuto delle notizie.
    - Sensibilità alla lunghezza del testo: data la variazione nella lunghezza del testo tra notizie vere e false, CountVectorizer si adatta bene, perché si basa sulla frequenza relativa delle parole piuttosto che sulla loro presenza assoluta.
    - Facilità di interpretazione: la rappresentazione numerica prodotta da CountVectorizer è di facile interpretazione. La presenza e la frequenza delle parole nel vettore risultante forniscono immediati spunti sulla struttura delle notizie.
- Tempo di calcolo: tempo impiegato per l'estrazione delle features con CountVectorizer: - Durata: 5.608 secondi

4.2.2. Tfidf Vectorizer

Esaminiamo il suo funzionamento e perché è pertinente per il nostro compito di classificazione binaria.

- Vectorizer TF-IDF: TF-IDF (Term Frequency-Inverse Document Frequency) è un metodo di vettorizzazione del testo che attribuisce un peso a ciascuna parola basato sulla sua frequenza nel documento e sulla sua rarità nel corpus complessivo. Quindi, le parole frequenti in un documento ma rare nel corpus avranno un peso maggiore.
- Funzionamento:

    - Calcolo della Frequenza delle Parole (TF): calcola la frequenza di ciascuna parola nel documento, indicando la frequenza di comparsa di una parola nel documento.
    - Calcolo dell'Inverse Document Frequency (IDF): calcola l'inverso della frequenza con cui una parola appare nel corpus complessivo. Questo valore è più alto per le parole rare e più basso per le parole comuni.
    - Moltiplicazione di TF e IDF: moltiplica la frequenza delle parole (TF) per l'Inverse Document Frequency (IDF) per ottenere il valore TF-IDF di ciascuna parola nel documento.
    - Rappresentazione numerica: i documenti sono rappresentati come vettori di TF-IDF, in cui ciascun elemento corrisponde al valore TF-IDF di una parola nel documento.
- Applicazione alla classificazione di fake news: la scelta di usare il Vectorizer TF-IDF è stata motivata da considerazioni specifiche al nostro contesto di classificazione binaria di fake news:
    - Ponderazione dell'importanza delle parole: TF-IDF assegna pesi alle parole basandosi sulla loro rilevanza nel documento e nel corpus, permettendo di catturare meglio l'importanza delle parole specifiche nel distinguere tra notizie reali e false.
    - Riduzione dell'impatto delle parole comuni: riducendo il peso delle parole comuni, TF-IDF si concentra sulle parole più informative e discriminative, contribuendo ad aumentare la capacità predittiva del modello.
    - Adattabilità a diversi contesti: TF-IDF è adatto a contesti in cui la frequenza delle parole chiave può variare notevolmente tra i documenti, come nel caso delle notizie con diverse lunghezze e stili.
    - Gestione delle parole rare: l'IDF privilegia le parole rare, permettendo al modello di attribuire maggiore importanza a termini meno frequenti ma potenzialmente più informativi nelle fake news.
- Tempo di calcolo: tempo impiegato per l'estrazione delle feature con CountVectorizer: - Durata: 5.647 secondi

I tempi di estrazione delle feature sono quindi praticamente gli stessi. Al termine della vettorizzazione avrò un totale di 178.070 parole.

4.3. Creazione di una funzione di Performance Evaluation

Si è scelto di creare una funzione chiamata *prediction_evaluation*, al fine di valutare le prestazioni di un modello di classificazione binaria.

Essa prende in input le etichette reali (y_test) e le etichette predette dal modello (y_pred). Di seguito un riassunto delle operazioni che svolge:

- Calcola diverse metriche di valutazione delle prestazioni del modello, come l'accuratezza, la precisione, ila recall (o sensibilità), il punteggio F1 e la specificità, ovvero la capacità del modello di classificare correttamente gli esempi negativi, e

cioè quanti veri negativi sono stati classificati correttamente rispetto al totale dei veri negativi.
- Calcola e visualizza la matrice di confusione, che mostra il numero di predizioni corrette e errate fatte dal modello.
- Calcola e visualizza la curva ROC (Receiver Operating Characteristic) insieme all'area sotto la curva (ROC AUC). Questa curva rappresenta la relazione tra il tasso di veri positivi e il tasso di falsi positivi al variare della soglia di classificazione.

In questo modo si ha una valutazione completa e visuale delle prestazioni del modello di classificazione, consentendo di comprendere meglio come il modello si comporta nel fare predizioni su nuovi dati. La matrice di confusione fornisce una panoramica dei diversi tipi di predizioni fatte dal modello, mentre la curva ROC e l'area sotto la curva (AUC) forniscono informazioni sulla capacità discriminativa del modello.

4.4. Creazione di una funzione di Calcolo dei Tempi

La funzione *save_time_stamp* è stata impiegata per registrare i tempi di esecuzione di diverse operazioni durante il processo di estrazione delle feature tramite vettorizzatori, addestramento e test del modello, nonché ottimizzazione degli iperparametri, ed è quindi utile per registrare informazioni correlate all'interno di un file di testo.

Viene calcolata la durata totale dell'operazione sottraendo il tempo di inizio dal tempo di fine. Questo valore rappresenta il tempo trascorso per eseguire un'operazione.

4.5. Classificatore Baesyano

Nella scelta del modello per la classificazione delle fake news, si è scelto il Naive Bayes principalmente per la sua comprovata efficacia nell'elaborazione del testo. Il Naive Bayes, sebbene di semplice implementazione, si distingue per la sua potenza nel trattare grandi volumi di dati testuali, rendendolo una scelta ideale per le applicazioni di elaborazione del linguaggio naturale.

Una delle caratteristiche più apprezzate del modello Naive Bayes è la sua capacità di gestire numerose features indipendenti, comune nei dati testuali. Questa flessibilità lo rende particolarmente adatto per la classificazione delle fake news.

Inoltre, il Naive Bayes offre un basso costo computazionale, rendendolo adatto per l'uso su larga scala e con set di dati di grandi dimensioni.

Nel contesto delle fake news, dove il rumore nei dati è diffuso, il Naive Bayes si dimostra efficace nel gestire informazioni incoerenti o non pertinenti, contribuendo alla corretta classificazione delle notizie.

Un aspetto chiave che rende il modello Naive Bayes efficace è il principio di indipendenza condizionale, dove ogni parola o feature contribuisce indipendentemente alla probabilità finale. Questa caratteristica è cruciale nel rilevamento delle fake news, dove le singole parole possono fornire indizi significativi sulla veridicità della notizia.

Infine, l'utilizzo della grid search di sklearn per l'ottimizzazione degli iperparametri può migliorare ulteriormente le prestazioni dei modelli Naive Bayes, consentendo di individuare i migliori parametri per la classificazione delle fake news. Questo approccio offre una maggiore flessibilità rispetto alla scelta di iperparametri predefiniti, portando a risultati ottimizzati e più accurati.

4.5.1. HyperParametri

Il classificatore bayesiano naïve, noto anche come classificatore di Bayes, è un'importante tecnica di apprendimento automatico basata sul teorema di Bayes. Questo classificatore assume che l'effetto di un particolare attributo su una classe sia indipendente dagli altri attributi.

I parametri che sono stati scelti per le varie esecuzioni del classificatore bayesiano sono i seguenti:

- alpha: Questo parametro rappresenta la probabilità a priori del verificarsi di un dato evento. In altre parole, è una speculazione o un'ipotesi fatta sulla probabilità di un evento prima di aver raccolto e analizzato i dati. Di default è impostata a 1.
- fit_prior: Questo parametro determina se si dovrebbe apprendere la probabilità delle classi prima dell'esecuzione. Se si imposta questo parametro come false, allora si utilizza una probabilità uniforme. Questo significa che tutte le classi avranno la stessa probabilità, indipendentemente dalla distribuzione dei dati. Nel nostro caso è impostata di default a True.

4.5.2. Performance Evaluation

Il modello bayesiano, insieme agli altri modelli considerati, è stato sottoposto ad addestramento e test utilizzando entrambe le tecniche di vettorizzazione, TF-IDF e CountVectorizer (CV). Questa metodologia è stata adottata al fine di valutare le prestazioni del modello in diverse rappresentazioni dei dati testuali e per determinare quale vettorizzatore si adatta meglio al problema di classificazione delle fake news.

Grazie alla funzione di valutazione delle prestazioni descritta nel paragrafo 4.3, è possibile ottenere rapidamente una valutazione delle prestazioni del modello, espressa attraverso le metriche di Accuracy, Precision, Recall, F1 e ROC AUC:

Guardando i risultati ottenuti, si possono fare diverse considerazioni:

Performance evaluation with cv	
Accuracy	0.9556368625657443
Precision	0.9483268836785795
Recall	0.9627917725907095
F1	0.9555045871559633
ROC AUC	0.9557113422808718
Specificità	0.9486309119710342
Precision negativa:	0.9630140133241443

Performance evaluation with tfidf	
Accuracy	0.9392865309855934
Precision	0.9305807622504537
Recall	0.9480009244280102
F1	0.93921007441328
ROC AUC	0.9393772442913981
Specificità	0.9307535641547862
Precision negativa:	0.9481327800829875

1) Si osserva che il modello Bayesiano addestrato con CountVectorizer (cv) ottiene prestazioni leggermente migliori rispetto a quello addestrato con TF-IDF in tutte le metriche valutate. In particolare, l'Accuracy, la Precision, il Recall, l'F1 e l'Area sotto la curva ROC (ROC AUC) sono tutti superiori nel caso del vettorizzatore cv rispetto a TF-IDF.

2) Entrambi i modelli hanno valori elevati di Precision e Recall, indicando che sono in grado di classificare correttamente un'elevata percentuale di istanze positive (fake news) e negative (real news). La Precision indica la proporzione di istanze identificate come fake news che sono effettivamente fake, mentre il Recall indica la proporzione di fake news nel dataset che sono state correttamente identificate dal modello.

3) L'F1-score tiene conto sia di Precision che di Recall e fornisce una misura complessiva delle prestazioni del modello. Entrambi i modelli hanno un F1-score elevato, indicando un buon equilibrio tra Precision e Recall.

4) L'Area sotto la curva ROC (ROC AUC) misura la capacità del modello di distinguere tra le classi positive e negative. Entrambi i modelli hanno valori elevati di ROC AUC, il che significa che sono in grado di classificare correttamente le istanze con un'alta probabilità.

In generale, i risultati indicano che entrambi i modelli Bayesiani sono efficaci nella classificazione delle fake news, con prestazioni leggermente migliori ottenute utilizzando il vettorizzatore CountVectorizer. Tuttavia, è importante considerare anche altri fattori, come la dimensione e la rappresentatività del dataset, e confrontare questi risultati con altri modelli per determinare la scelta migliore per l'applicazione specifica.

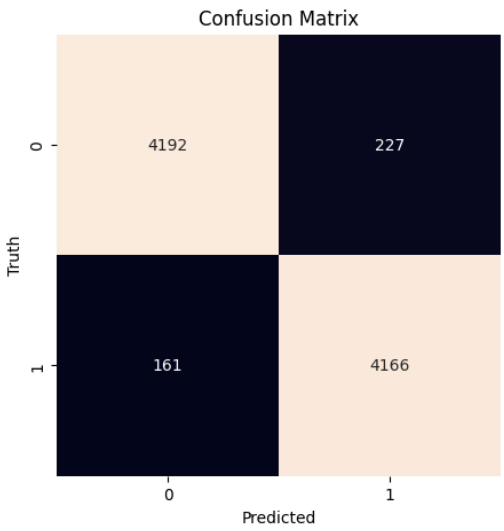


Figura 13: Confusion matrix con cv

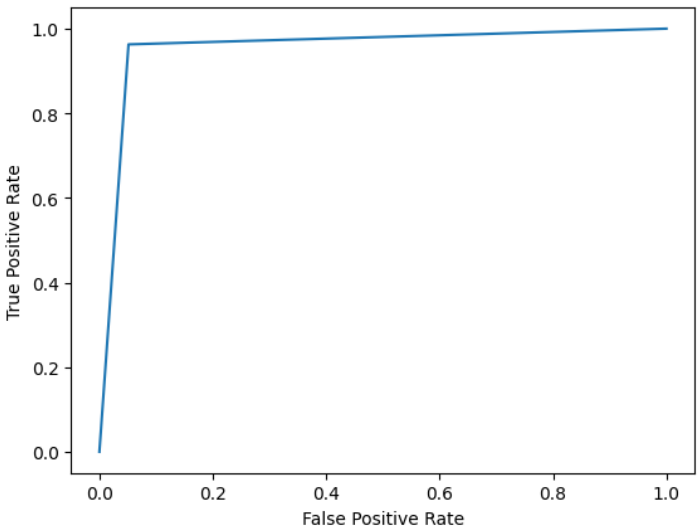


Figura 14: Curva ROC con cv

Osservando la confusion matrix, inoltre, si può affermare che:

1) Bilanciamento tra positivi e negativi: il numero di casi positivi (TP + FN) e negativi (TN + FP) non è sbilanciato, il che è un buon segno. Se uno dei due fosse molto più grande dell'altro, potrebbe indicare un problema di sbilanciamento dei dati che potrebbe influenzare le prestazioni del modello.
2) Alto numero di TN e TP: il fatto che ci siano un alto numero di True Negatives (TN) e True Positives (TP) indica che il modello sta classificando correttamente la maggior parte dei casi negativi e positivi. Questo è un buon risultato e suggerisce che il modello sta apprendendo efficacemente i pattern nei dati.
3) Errore di classificazione: tuttavia, è importante notare che ci sono anche False Positives (FP) e False Negatives (FN), il che significa che il modello sta commettendo errori di classificazione. Potrebbe essere utile esaminare più da vicino i casi in cui il modello ha commesso errori per capire quali potrebbero essere le cause e se ci sono modi per migliorare le prestazioni.

Lo stesso discorso si può fare per il modello con TFIDF, che ottiene risultati leggermente inferiori ma pur sempre ottimi:

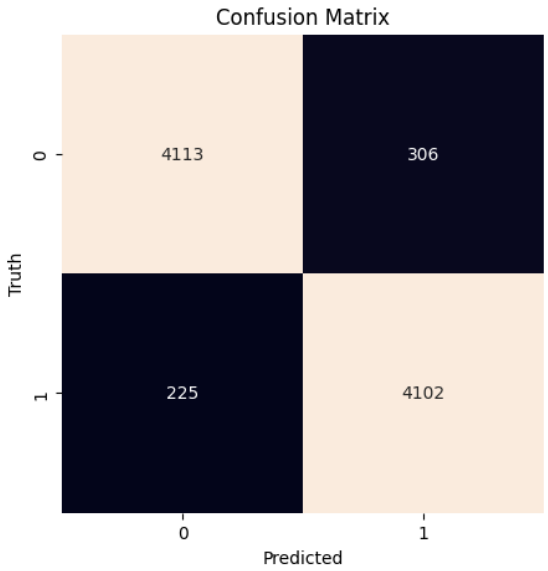


Figura 15: Confusion matrix con tfidf

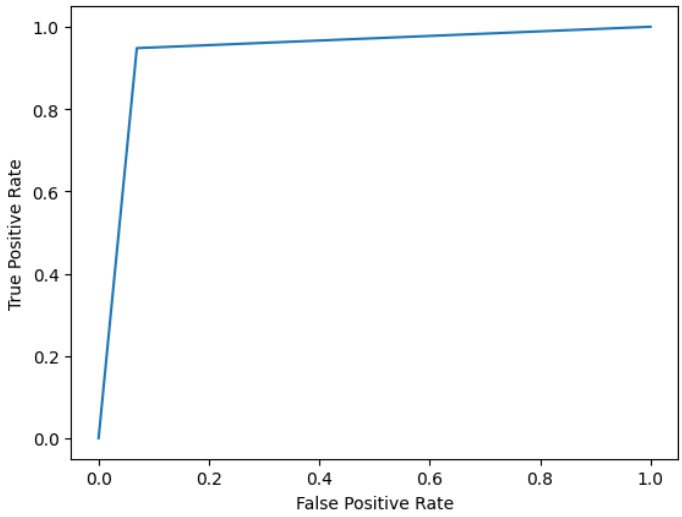


Figura 16: Curva ROC con tfidf

4.5.3. Fine Tuning

Il "fine-tuning" è un processo di ottimizzazione che viene utilizzato nell'ambito del machine learning e del deep learning per migliorare le prestazioni di un modello già addestrato. In generale, il fine-tuning si riferisce alla pratica di regolare gli iperparametri o i pesi di un modello già addestrato per adattarlo meglio a un compito specifico o per migliorare le sue prestazioni su nuovi dati.

Nonostante i risultati soddisfacenti ottenuti con il modello bayesiano, si è intrapresa la strada del miglioramento ulteriore esplorando le potenzialità della ricerca degli iperparametri ottimali. Questa decisione è stata motivata dalla volontà di massimizzare le prestazioni del modello e ottenere risultati ancora migliori. Per condurre questa ricerca in modo efficiente

ed efficace, si è optato per l'utilizzo del metodo grid poiché, nonostante la notevole quantità di dati a disposizione, questo approccio ha dimostrato di comportarsi in modo robusto ed efficace, garantendo una copertura completa dello spazio degli iperparametri. Questo ci ha fornito la sicurezza di esplorare in modo esaustivo le possibili combinazioni e individuare quella ottimale per il nostro modello, senza dover ricorrere a soluzioni computazionalmente più onerose o complesse. Inoltre, l'approccio basato su grid ci ha offerto una struttura organizzata e sistematica per condurre l'analisi dei parametri, consentendo una valutazione accurata delle prestazioni del modello in diverse configurazioni iperparametriche.

4.5.3.1. Grid Search

La Grid Search è una tecnica di ottimizzazione dei parametri utilizzata nell'addestramento dei modelli di machine learning. Consiste nel definire una griglia di combinazioni di iperparametri e valutare le prestazioni del modello per ciascuna combinazione tramite una procedura di cross-validation. In questo modo, è possibile individuare la combinazione ottimale di iperparametri che massimizza le prestazioni del modello rispetto a una metrica di valutazione specifica, come l'accuratezza o l'F1-score. La Grid Search è particolarmente utile quando si desidera esplorare un ampio spazio di iperparametri per trovare la migliore configurazione per il modello.

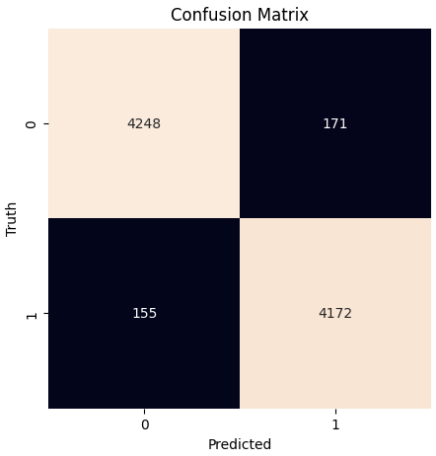


Figura 17: Confusion matrix con cv dopo il tuning degli hyperparametri

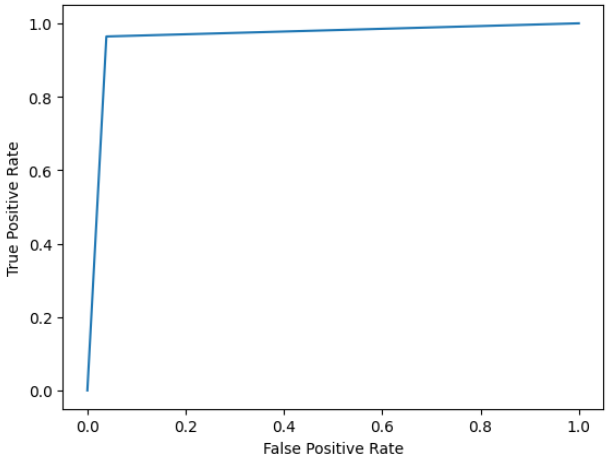
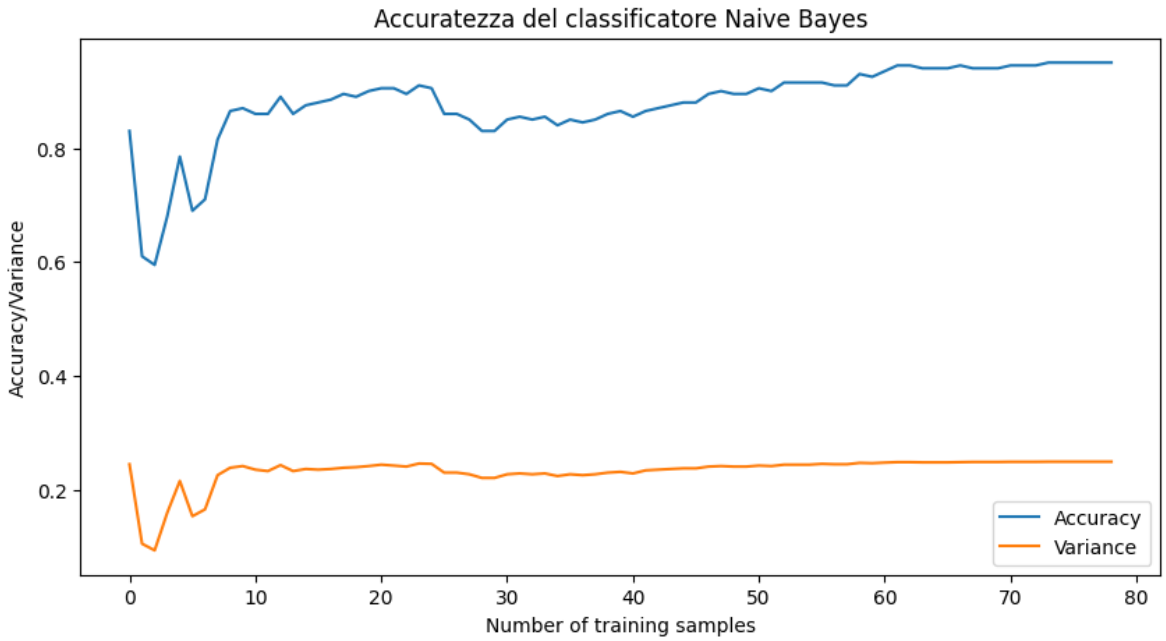


Figura 18: Curva ROC con cv dopo il tuning degli hyperparametri

Dall'analisi della matrice di confusione emerge un miglioramento delle prestazioni del modello. L'accuracy, è aumentata, passando da 0.9556 a 0.9627. Stessa cosa per il fine tuning con tfidf:

Performance evaluation post tuning with tfidf	
Accuracy	0.957809284244226
Precision	0.9770973963355835
Recall	0.9366766813034435
F1	0.9564601769911505
ROC AUC	0.9575893024077752
Specificità	0.9785019235121069
Precision negativa:	0.9404088734232275



Il grafico mostra l'andamento delle prestazioni di un modello Naive Bayes ottimizzato attraverso la ricerca di iperparametri, utilizzando le feature estratte tramite CountVectorizer.

All'inizio del processo di addestramento, si osserva un'accuracy relativamente elevata, indicando che il modello ha una buona capacità iniziale di apprendere dai dati di addestramento. Tuttavia, contemporaneamente, la varianza mostra un livello iniziale alto, indicando una considerevole fluttuazione nelle prestazioni del modello.

Con l'aggiunta progressiva di testi al set di addestramento, si evidenziano oscillazioni significative nell'accuracy, manifestate attraverso una serie di valli e picchi. Tale comportamento suggerisce che il modello sta reagendo in modo sensibile alle nuove istanze introdotte.

In un secondo momento, si osserva un miglioramento graduale dell'accuracy, che converge verso un valore stabile. Questo indica che, con l'aumentare del numero di testi di addestramento, il modello ha acquisito una comprensione più robusta del problema e delle caratteristiche del testo in questione.

Tuttavia, è importante notare che la varianza segue una traiettoria simile, attraversando le medesime fluttuazioni prima di raggiungere anch'essa un valore stabile. La varianza costante attorno a 0.27 suggerisce che, nonostante l'accuracy abbia raggiunto una stabilità, il modello continua a mostrare una certa sensibilità alle variazioni nei dati di addestramento.

In sintesi, mentre l'accuracy riflette il miglioramento del modello nel catturare i pattern nei dati, la varianza segnala che il modello mantiene una certa suscettibilità a fluttuazioni nonostante il processo di addestramento prolungato. Questo equilibrio tra miglioramento e stabilità è fondamentale per garantire che il modello possa generalizzare efficacemente su nuovi dati.

4.5.4. Tempi di Calcolo

Per quanto riguarda i tempi di calcolo, abbiamo utilizzato la funzione *save_time_stamp* descritta nel paragrafo 4.4. I tempi sono stati molto brevi, con un'efficienza notevole riscontrata soprattutto nell'utilizzo di CountVectorizer.

TEMPI	Addestramento del Modello	Predizione del Modello	Addestramento del Modello Migliore	Predizione del Modello Migliore
MultinomialNB con CV	0.036 secondi	0.01 secondi	2.323 secondi	0.011 secondi
MultinomialNB con TF-IDF	0.042 secondi	0.011 secondi	2.825 secondi	0.012 secondi

4.6. Decision Tree Classifier

La scelta del modello **Decision Tree** è stata guidata dalla sua natura esplicativa e dalla capacità di gestire dati non lineari, fornendo un approccio chiaramente interpretabile alla classificazione delle fake news. La flessibilità di questo modello si adatta bene all'obiettivo di distinguere tra notizie vere e false, particolarmente in un contesto di alta dimensionalità.

Due caratteristiche principali degli alberi decisionali sono:

1. **Interpretabilità**: Gli alberi decisionali sono modelli facilmente interpretabili, poiché le regole di decisione sono rappresentate sotto forma di albero. Questo permette di comprendere facilmente il processo decisionale seguito dal modello.
2. **Versatilità**: Gli alberi decisionali possono gestire sia dati numerici che categorici e possono essere utilizzati per risolvere una vasta gamma di problemi di machine learning. Possono anche essere combinati con tecniche come il pruning per evitare l'overfitting e migliorare le prestazioni del modello.

Gli alberi di decisione possono gestire variabili categoriche, ma è preferibile convertire le variabili categoriche in variabili numeriche o booleane per garantire una migliore compatibilità con l'algoritmo e per ottenere risultati più significativi.

4.6.1. Performance Evaluation

Dal training e testing dell'albero di decisione con le due vettorizzazioni si ottengono i seguenti risultati:

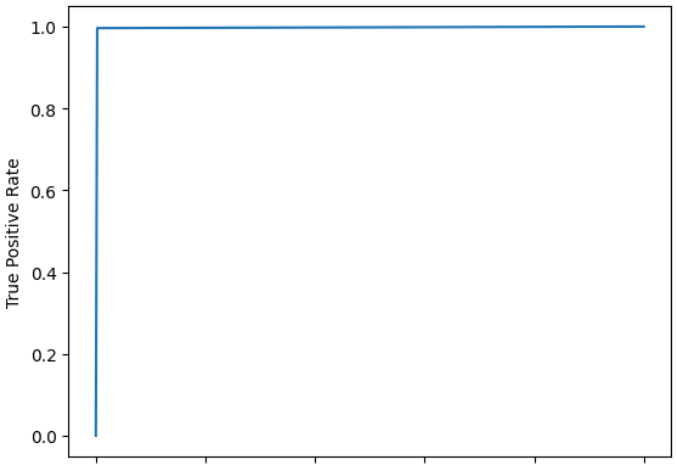


Figura 19: Curva ROC con cv

4.6.1.1. CV

Performance evaluation with cv	
Accuracy	0.9969128744568946
Precision	0.9974548819990745
Recall	0.9963022879593252
F1	0.9968782518210197
ROC AUC	0.9969065184987846
Specificità	0.997510749038244
Precision negativa:	0.9963833634719711

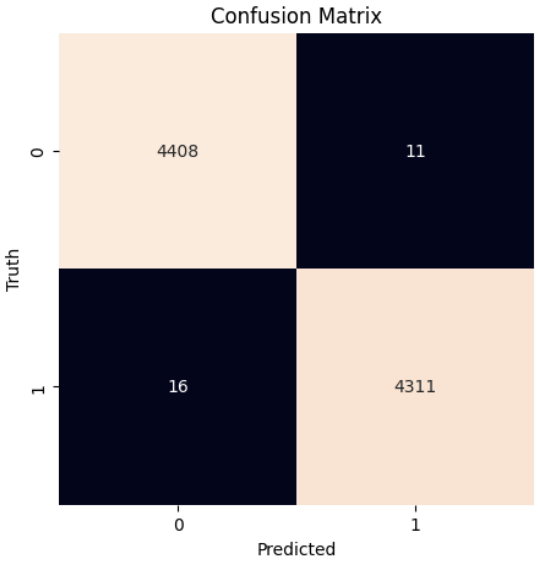


Figura 20: Confusion matrix con cv

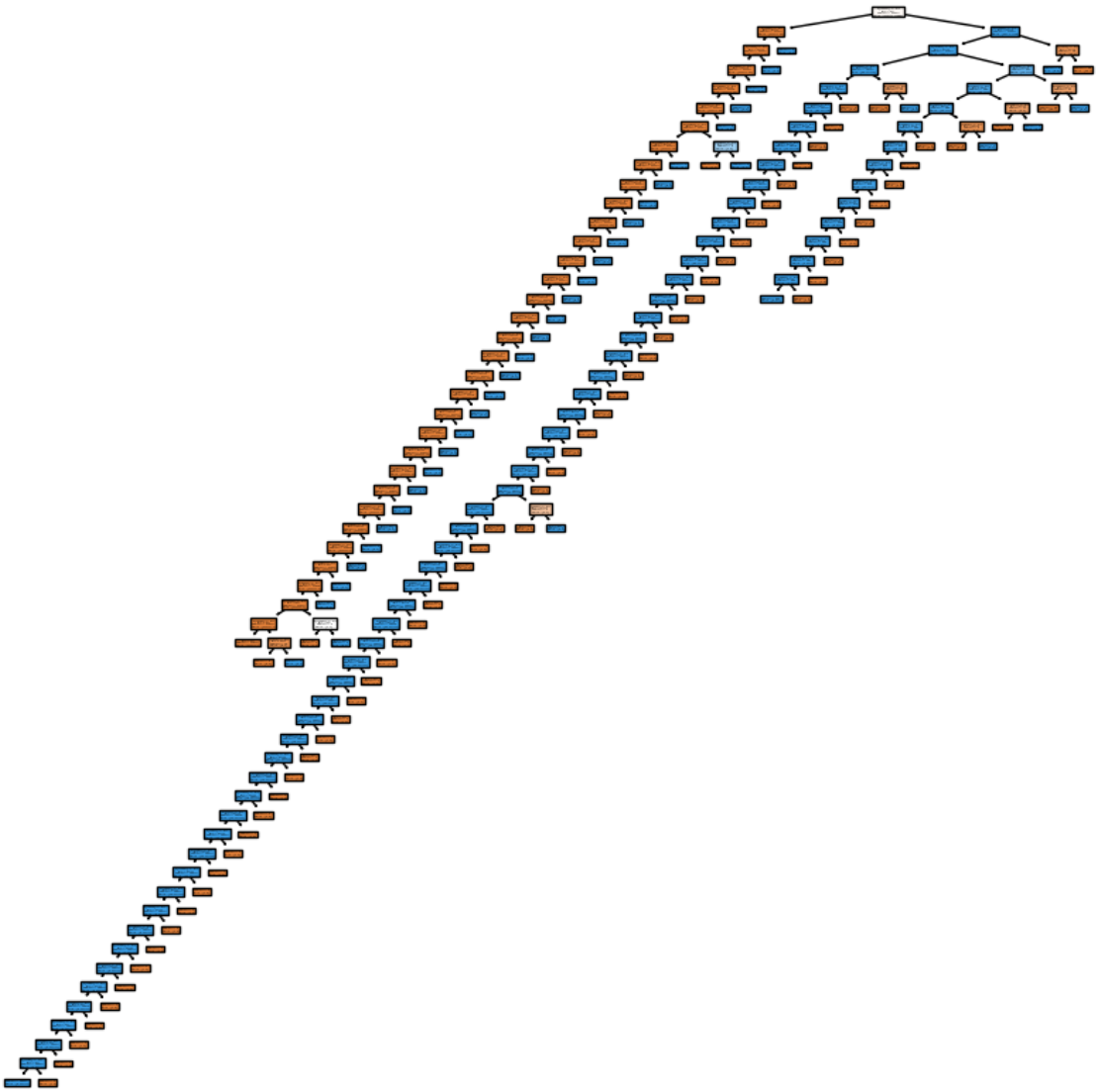


Figura 21: Albero risultante con cv

4.6.1.2. TFIDF

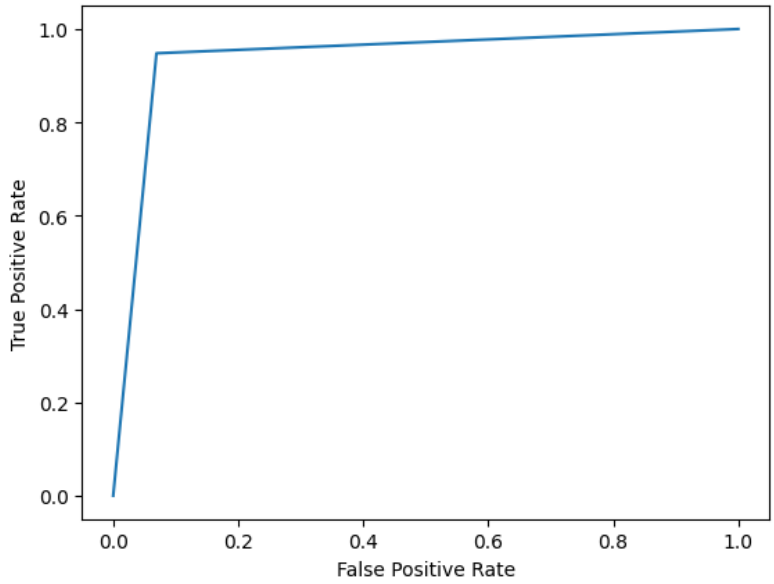
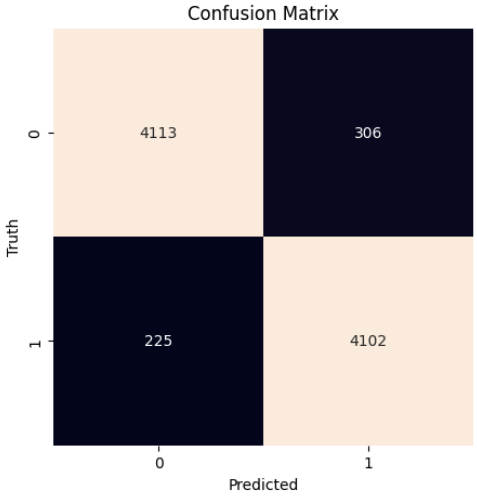


Figura 22: ROC tfidf



I risultati mostrano in entrambi i casi prestazioni eccezionali del modello, con valori molto elevati per tutte le metriche di valutazione, tra cui Accuracy, Precision, Recall, F1, ROC AUC, Specificità e Precisione negativa. Tali valori indicano che il modello è in grado di classificare correttamente la stragrande maggioranza dei casi, sia positivi che negativi, con un'elevata precisione e sensibilità.

Tuttavia, il fatto che l'albero di decisione sia molto profondo può sollevare alcune preoccupazioni riguardo alla complessità del modello. Un albero decisionale molto profondo potrebbe indicare che il modello è stato addestrato ad adattarsi eccessivamente ai dati di training, memorizzando dettagli specifici che potrebbero non generalizzare bene su nuovi dati (overfitting). Inoltre, nodi sbilanciati possono rendere più difficile l'interpretazione dell'albero di decisione, poiché la struttura dell'albero potrebbe non riflettere in modo accurato la relativa importanza delle caratteristiche nei dati.

Pertanto, nonostante i risultati estremamente positivi, potrebbe essere necessario valutare attentamente se la complessità del modello è giustificata dalla sua capacità di generalizzare

su dati non visti. Potrebbe essere utile esaminare alternative per ridurre la complessità del modello, ad esempio attraverso la potatura dell'albero decisionale o l'ottimizzazione degli iperparametri.

4.6.2. Fine Tuning

Per affrontare il problema degli alberi di decisione sbilanciati, è possibile adottare diverse strategie:

1. Potatura dell'albero: ridurre la complessità dell'albero eliminando nodi che non aggiungono significativamente alla sua capacità predittiva. Questo può aiutare a prevenire l'overfitting e a semplificare l'albero, ed è il metodo adottato nel progetto.
2. Utilizzo di criteri di divisione alternativi: esplorare criteri di divisione alternativi che tengano conto del bilanciamento dei dati nelle varie ramificazioni dell'albero.
3. Modifica dei parametri di crescita: regolare i parametri di crescita dell'albero, come la profondità massima o il numero minimo di campioni richiesti per suddividere un nodo, per ottenere una struttura più bilanciata dell'albero.
4. Utilizzo di ensemble di alberi: utilizzare tecniche di ensemble come Random Forest o Gradient Boosting, che combinano più alberi di decisione e possono essere più robusti degli alberi sbilanciati.

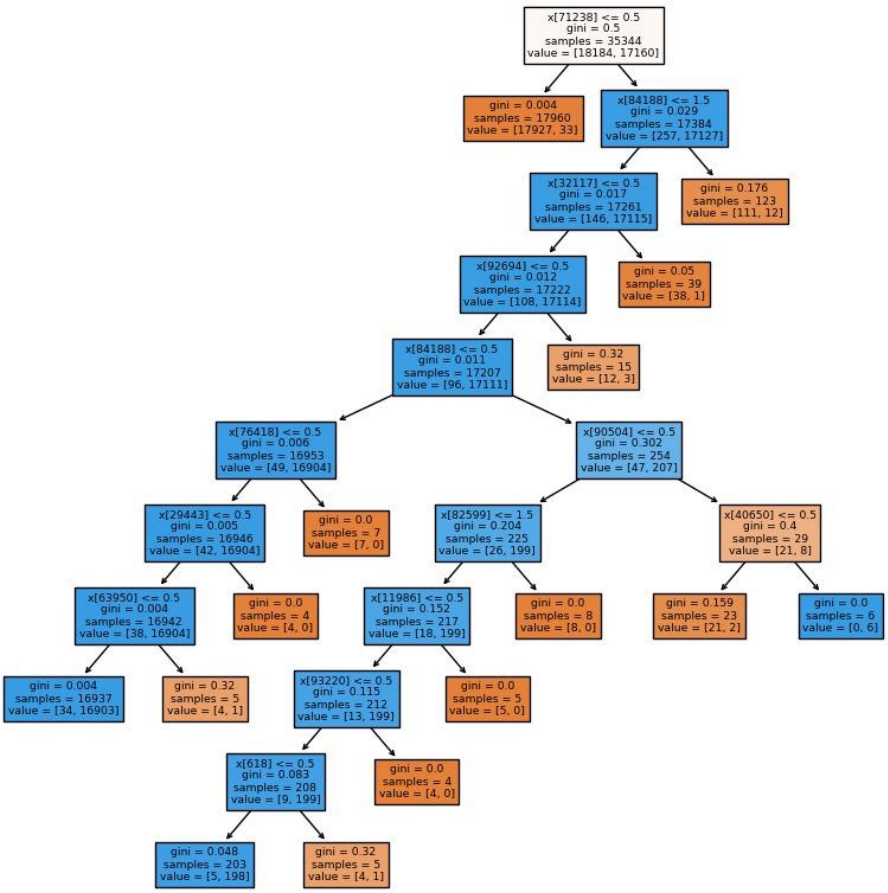
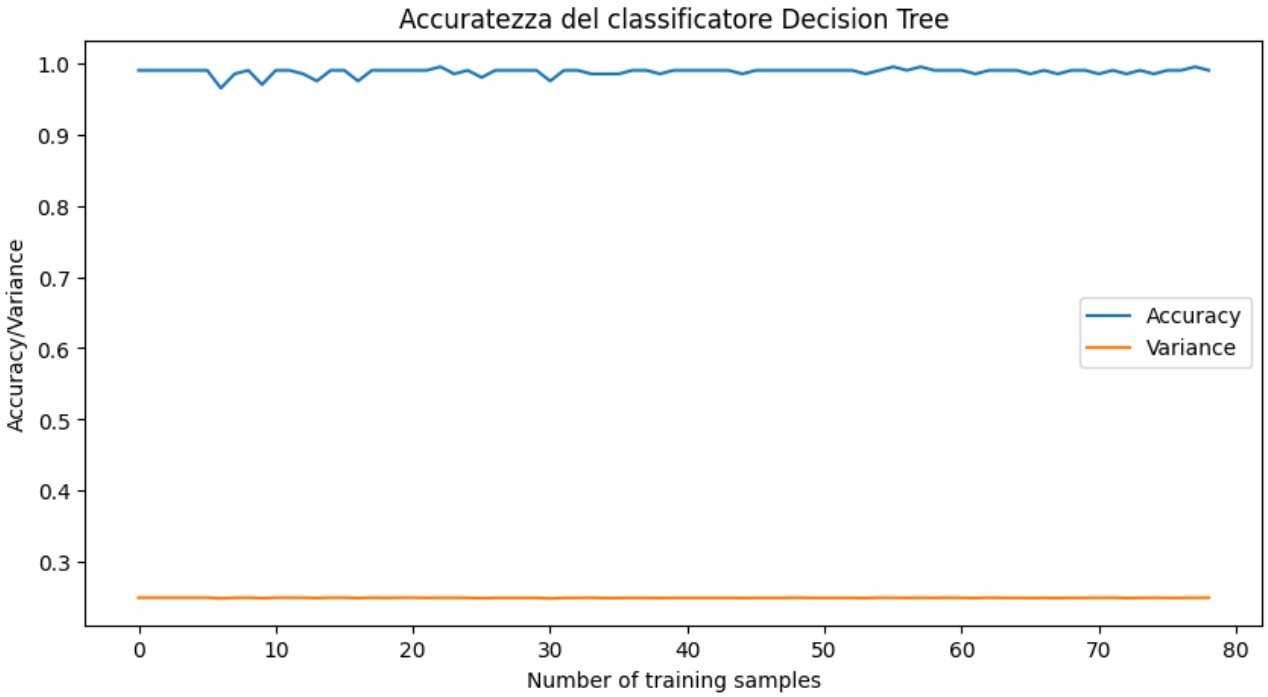


Figura 23: Albero potato post tuning degli iperparametri

Vediamo come varia l'accuratezza dell'albero decisionale in funzione della complessità dell'albero. Lavoro dunque sul parametro di complessità, che stima quello che è il vantaggio di aggiungere o togliere degli split in funzione degli errori di classificazione, e quindi al fine di accorciare l'albero.

Il parametro di complessità corrisponde quindi al minimo miglioramento che il nostro modello è in grado di garantire al variare dei nodi. Esso mi suggerisce quando fermarmi a tagliare l'albero.

L'ottimizzazione dei parametri del Decision Tree è stata una tappa cruciale nel processo di sviluppo del modello. Utilizzando tecniche come la Grid search, abbiamo individuato i parametri che massimizzano le prestazioni del Decision Tree nella distinzione tra true e fake news. Questo passaggio è stato fondamentale per garantire la capacità predittiva ottimale del nostro modello. In questo modo non si sono ottenuti risultati evidentemente superiori, ma si sono accorciati significativamente i tempi di computazione.



Il grafico illustra l'andamento delle prestazioni di un modello Decision Tree ottimizzato mediante la ricerca di iperparametri, con le feature ottenute attraverso CountVectorizer.

Inizialmente, si osserva un'accuratezza relativamente elevata, indicando che il modello è in grado di apprendere efficacemente dai dati di addestramento. Tuttavia, si presentano leggeri affossamenti in alcuni punti del grafico, suggerendo possibili variazioni nella performance del modello su specifici dati.

Con l'incremento progressivo dei dati di addestramento, si nota una notevole mitigazione degli affossamenti iniziali. L'accuratezza, inizialmente soggetta a fluttuazioni, si stabilizza gradualmente a valori leggermente superiori rispetto al punto di partenza. Questo fenomeno

suggerisce che, con l'aumento della quantità di dati di addestramento, il modello migliora la sua capacità di generalizzazione e supera le difficoltà iniziali.

D'altra parte, la varianza rimane costantemente a livello zero per tutta la durata dell'esecuzione del modello. Questo comportamento indica che il modello Decision Tree ottimizzato mostra una stabilità significativa nelle sue prestazioni, nonostante le fluttuazioni iniziali dell'accuratezza. La varianza zero suggerisce che il modello è coerente e robusto, mantenendo una performance costante su diverse parti del dataset di addestramento.

4.6.3. Tempi di Calcolo

TEMPI	Addestramento del Modello	Predizione del Modello	Addestramento del Modello Migliore	Predizione del Modello Migliore
DecisionTree con CV	8.759 secondi	0.1 secondi	23.029 minuti	0.011 secondi
DecisionTree con TF-IDF	24.504 secondi	0.011 secondi	60.05 minuti	0.012 secondi

- DecisionTree con CV: l'addestramento del modello utilizzando il CountVectorizer richiede un tempo relativamente breve, con circa 8.759 secondi. Questo suggerisce che il processo di addestramento è efficiente e veloce. Anche il tempo necessario per la predizione del modello è molto breve, con soli 0.1 secondi. Ciò indica che una volta addestrato, il modello è in grado di effettuare predizioni istantanee.
- DecisionTree con TF-IDF: l'utilizzo del TF-IDF per l'addestramento del modello richiede un tempo notevolmente maggiore rispetto al CountVectorizer, con circa 23.029 minuti. Questo suggerisce che il processo di addestramento è più lungo e potenzialmente più complesso. Tuttavia, il tempo per la predizione del modello rimane relativamente breve, con soli 0.011 secondi. Ciò suggerisce che una volta addestrato, il modello è ancora in grado di effettuare predizioni rapidamente.

In sintesi, il Decision Tree si è rivelato un modello efficace per la classificazione binaria di fake news, grazie alla sua chiara interpretabilità e alla capacità di gestire dati non lineari. La combinazione di CountVectorizer e TF-IDF ha arricchito la rappresentazione del testo, mentre l'ottimizzazione dei parametri e la gestione dello sbilanciamento hanno contribuito a massimizzare le sue prestazioni.

4.7. Support Vector Machine

Il modello di Support Vector Machines (SVM) è stato adottato con attenzione nella nostra analisi delle fake news, considerando la sua notevole capacità di affrontare spazi ad alta dimensionalità e di trattare dati non lineari, specialmente in scenari in cui il numero di dimensioni supera il numero di campioni, fornendo un'adeguata flessibilità per le nostre esigenze specifiche.

Per affrontare la complessità del testo associato alle notizie, abbiamo integrato anche qui le tecniche di CountVectorizer e TF-IDF. CountVectorizer si è dimostrato utile nel catturare la frequenza delle parole, mentre TF-IDF ha contribuito a pesare le parole in base alla loro importanza nel contesto delle fake news.

4.7.1. HyperParametri

Nel caso del classificatore SVM, ci sono vari parametri che vengono presi in considerazione durante le varie esecuzioni. Questi parametri sono fondamentali per la configurazione e l'efficienza del modello:

- C: Questo parametro ha un ruolo cruciale perché controlla il compromesso tra l'ottenimento di un margine il più largo possibile e la minimizzazione delle violazioni di questo margine. In sostanza, regola l'equilibrio tra la semplicità del modello e il tasso di errore di classificazione sul set di allenamento.
- Kernel: Questo parametro specifica la natura dell'iper-piano utilizzato per separare i dati. Vari tipi di kernel possono essere utilizzati a seconda della natura dei dati e del problema specifico che si intende risolvere. Alcuni dei kernel più utilizzati sono:
    - rbf: noto anche come kernel gaussiano, misura la similarità di due punti in uno spazio di dimensioni infinite. Questo kernel è particolarmente utile quando i dati non sono linearmente separabili.
    - polinomiale: questo kernel utilizza una formula polinomiale per misurare la similarità tra i punti. È adatto per problemi in cui esiste una relazione polinomiale tra le caratteristiche.
    - lineare: il kernel lineare utilizza il prodotto vettoriale dei valori per determinare la loro similarità. È il più semplice dei kernel e funziona bene quando i dati sono linearmente separabili.

4.7.2. Fine Tuning

Una fase cruciale è stata l'ottimizzazione dei parametri del nostro modello SVM. Grazie a una ricerca accurata, abbiamo identificato la combinazione ottimale di parametri che ha massimizzato l'efficacia del nostro classificatore nel distinguere tra notizie vere e false.

Per ottenere questa combinazione, abbiamo utilizzato la Grid Search, ma a differenza degli altri modelli, per trovare la combinazione ottimale di iper-parametri, abbiamo applicato la tecnica k-fold (più precisamente 3-fold) fornita tramite il parametro cv della classe GridSearch.

Questo ci ha permesso di dividere il set di allenamento in tre sottoinsiemi separati e di utilizzare uno di questi tre come set di validazione. Il modello ha eseguito le run necessarie, producendo il miglior modello, ovvero quello con la migliore combinazione di iper-parametri.

È importante notare che la ricerca è stata eseguita sulle SVM utilizzando solo il kernel rbf. Questa scelta è stata fatta sia per ragioni di tempistiche che per valutare quanto

efficacemente un kernel diverso da quello lineare potesse essere utilizzato con il dataset in questione.

Dai risultati ottenuti, possiamo osservare che un kernel lineare riesce a separare le notizie false da quelle vere in modo più preciso e più rapido rispetto a un kernel rbf progettato specificamente per la suddivisione di feature non lineari.

Ciò dimostra che i nostri dati presentano caratteristiche linearmente separabili che un kernel come quello lineare riesce a cogliere con maggiore efficacia.

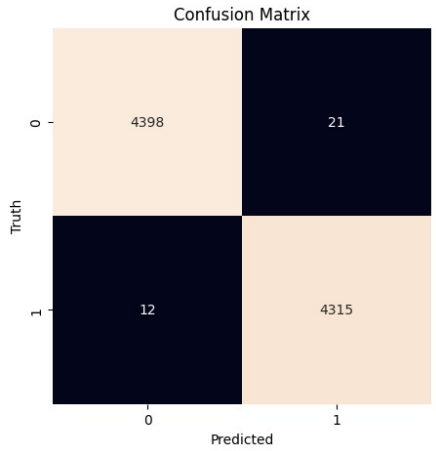
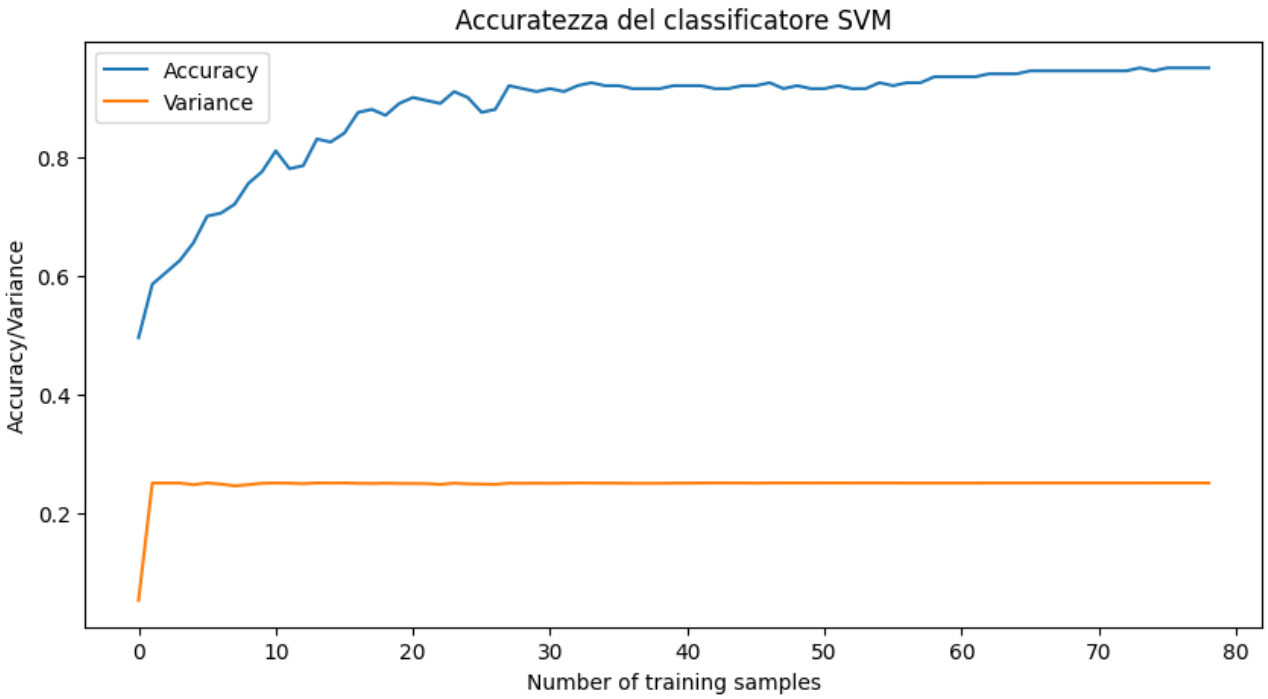


Figura 24: Confusion Matrix del miglior modello di svm con kernel lineare



Inizialmente, si osserva che la SVM, con un numero limitato di dati di addestramento, presenta una capacità predittiva scarsa, manifestata da un'accuratezza pari a 0.5, la quale suggerisce una performance casuale. Ciò si riflette in una varianza pari a 0, indicando che la previsione è costante e non varia con l'aggiunta di nuovi dati.

Con l'aumento progressivo dei dati di addestramento, si evidenzia un notevole miglioramento dell'accuratezza, indicando che il modello sta apprendendo in maniera più efficace dai dati. Tuttavia, contemporaneamente si verifica un drastico aumento della varianza. Questo suggerisce che, nonostante l'aumento dell'accuratezza, il modello diventa più sensibile alle variazioni nei dati di addestramento.

È interessante notare che, nonostante l'aumento della varianza, quest'ultima rimane mantenuta a un livello non troppo elevato. Ciò potrebbe indicare che la SVM sta beneficiando dell'incremento di dati di addestramento senza compromettere eccessivamente la sua capacità di generalizzazione.

4.7.3. Tempi di Calcolo

TEMPI	Addestramento	Predizione	Ricerca del Modello Migliore	Predizione del Modello Migliore
SVM con CV	131.336 secondi	15.531 secondi	1838.015 secondi	90.705 secondi
SVM con TF-IDF	586.929 secondi	49.895 secondi	3015.806 secondi	115.89 secondi

5. Modelli a confronto

5.1. Scelta di Accuracy

Per questo specifico progetto, abbiamo deciso di scegliere l'accuratezza come metrica dominante per valutare la qualità dei modelli. Ci sono diversi motivi chiave che hanno guidato questa decisione:

1. La semplicità e l'intuitività dell'accuratezza: Essenzialmente, l'accuratezza è una misura che ci indica la percentuale di previsioni corrette sul numero totale di previsioni. È un concetto semplice da comprendere e intuitivo, il che lo rende una scelta ideale per una metrica di valutazione.
2. Il bilanciamento delle classi nel nostro dataset: Nel nostro dataset, le classi che comprendono le fake-news e le real-news sono bilanciate. Questo significa che per ottenere un alto punteggio di accuratezza, il modello deve essere in grado di fare previsioni corrette su entrambe le classi, non favorendo una classe rispetto all'altra.
3. L'importanza della precisione delle previsioni su entrambe le classi: Non siamo particolarmente interessati a fare previsioni più precise su una classe specifica, ma piuttosto, il nostro obiettivo principale è che tutte le news, indipendentemente dal fatto che siano fake o reali, siano classificate correttamente. Questo è fondamentale per la natura del nostro progetto.

5.2. Analisi dei Grafici di Addestramento

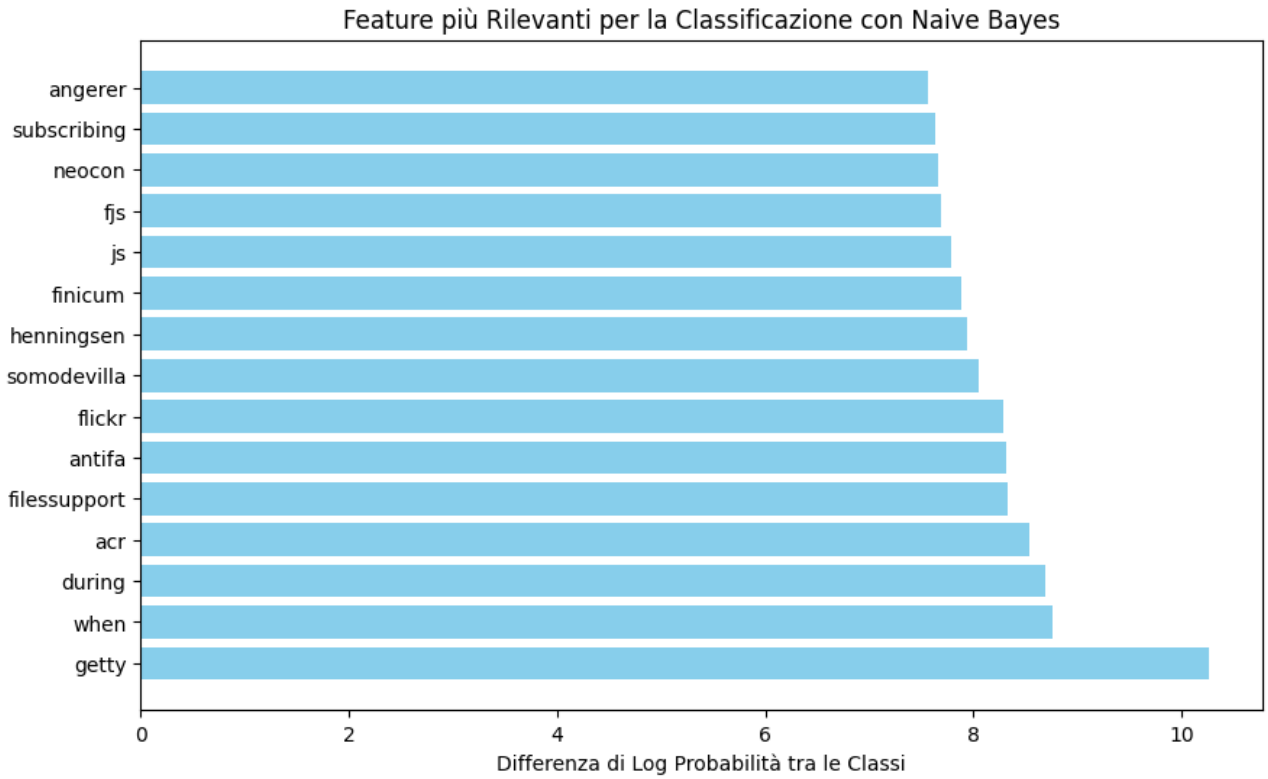
Al fine di valutare la capacità del modello di generalizzare su dati mai visti, abbiamo introdotto la varianza come indicatore chiave. La rappresentazione grafica della varianza facilita la valutazione visiva delle variazioni nella performance del modello durante l'addestramento.

Attraverso l'analisi dei grafici, si vuole identificare pattern emergenti nell'accuratezza del modello al variare delle dimensioni del tain-set. La varianza è stata utilizzata per individuare segnali di overfitting o underfitting, guidando eventuali aggiustamenti nella complessità del modello.

La metodologia di visualizzazione dell'accuratezza e della varianza offre una panoramica esaustiva della performance del modello, fornendo informazioni per ottimizzare le prestazioni e garantire una migliore capacità di generalizzazione su nuovi dati.

Per ogni grafico dato il train-set viene fornito a intervalli di 10 istanze, il modello viene allenato su quelle istanze e vengono poi prodotti in output l’accuratezza, rispetto ad un test-set separato inizialmente, e la varianza, che vengono poi rappresentate nel grafico.

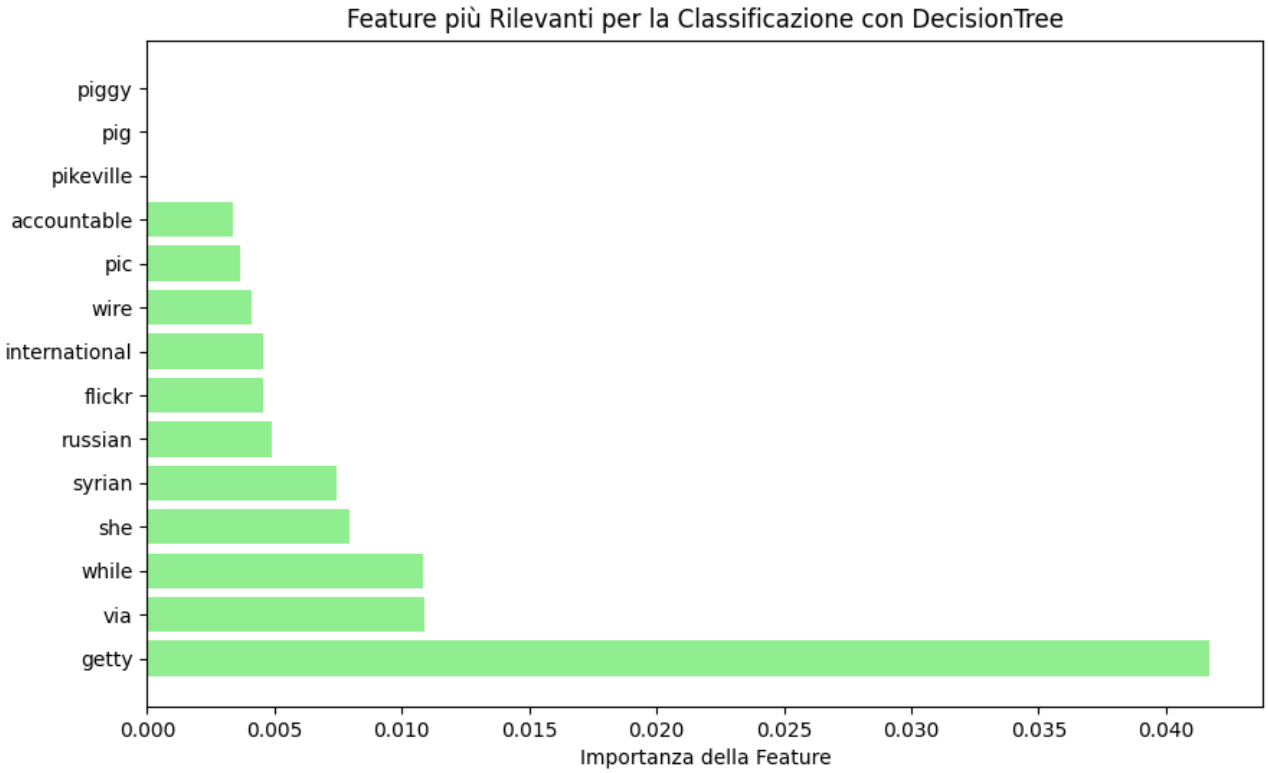
5.3. Grafici di importanza feature



I barplot mostra chiaramente i pesi delle feature nel modello Naive Bayes Multinomiale, calcolati come la differenza di probabilità tra le classi (fake e true news). Ogni barra rappresenta una feature specifica, mentre la lunghezza della barra riflette il peso della feature nella classificazione del testo.

Dal grafico, si osserva che alcune feature hanno pesi notevolmente alti come (getty, when e during), indicando che contribuiscono in modo significativo alla capacità del modello di distinguere tra fake e true news. Queste feature potrebbero svolgere un ruolo chiave nel processo decisionale del modello e sono importanti indicatori nella classificazione del testo.

Il fatto che il modello raggiunga un'accuracy del 94.2% suggerisce che queste feature, con i loro pesi distintivi, sono efficaci nella discriminazione delle classi. L'accuratezza elevata indica che il modello ha generalizzato bene su nuovi dati e ha catturato i pattern rilevanti nei testi per effettuare previsioni accurate.

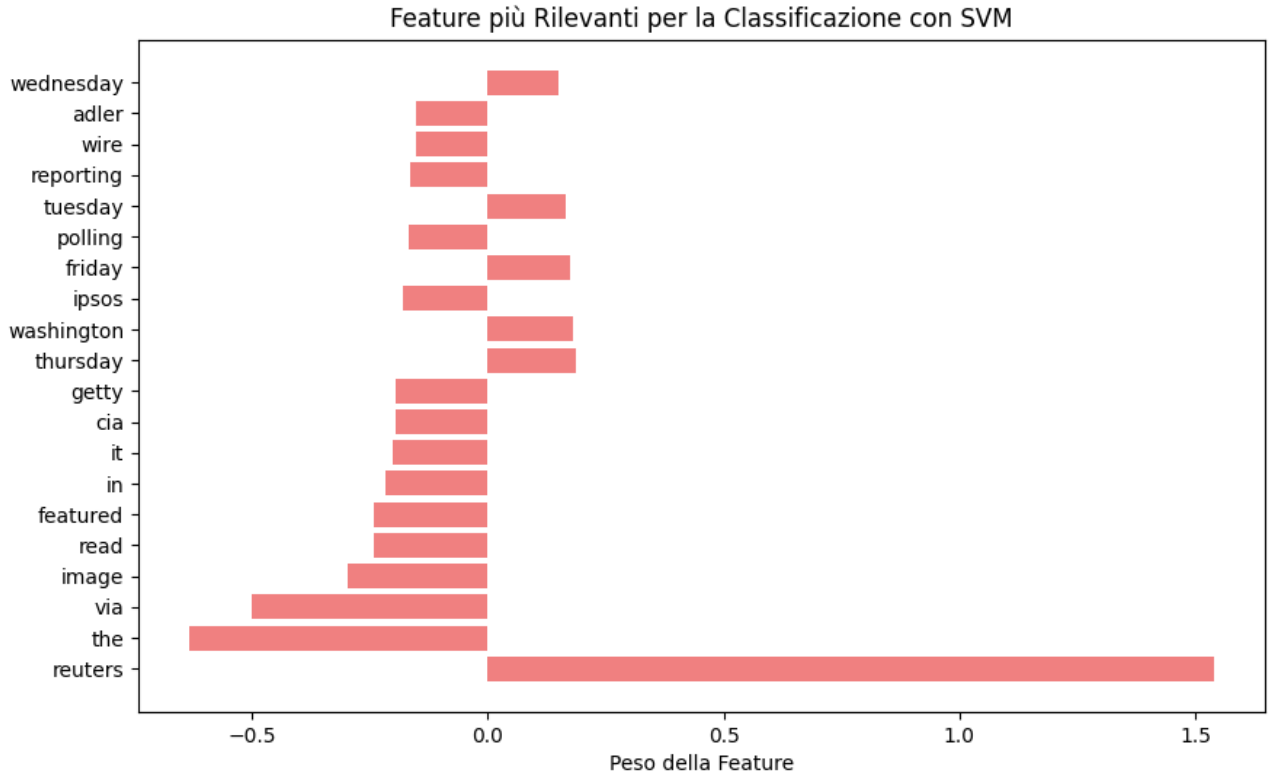


Il barplot mostra le 16 feature più rilevanti per la classificazione con il Decision Tree. Ogni barra rappresenta una feature specifica, mentre la lunghezza della barra riflette l'importanza relativa della feature nella presa di decisioni del modello. Le feature con barre più lunghe sono considerate più rilevanti ai fini della classificazione.

Il grafico evidenzia quali aspetti del dataset hanno maggior peso nella decisione del Decision Tree. Feature con importanze più alte giocano un ruolo più significativo nel processo decisionale del modello. Questa visualizzazione fornisce un'opportunità chiara per identificare quali aspetti del testo hanno avuto un impatto maggiore sulla classificazione.

Il termine "getty" ha un valore molto più alto rispetto alla seconda parola in termini di importanza nelle feature del modello, ciò suggerisce che, secondo il modello di classificazione, il termine "getty" è molto più informativo e discriminante nella distinzione tra le classi target rispetto a tutte le altre parole.

Nel contesto di un modello di classificazione del testo, l'importanza di una parola o di una feature deriva dal contributo che fornisce nel processo decisionale del modello. Se "getty" ha un peso significativamente maggiore, può indicare che questa parola è associata a caratteristiche distintive che il modello considera cruciali per la classificazione.



Il barplot mostra le 20 feature più rilevanti per la classificazione con la SVM. Le feature sono riportate sull'asse Y, mentre sull'asse X è rappresentato il peso associato a ciascuna feature. Le barre più lunghe indicano feature che hanno un maggiore impatto nella capacità del modello di discriminare tra le classi target.

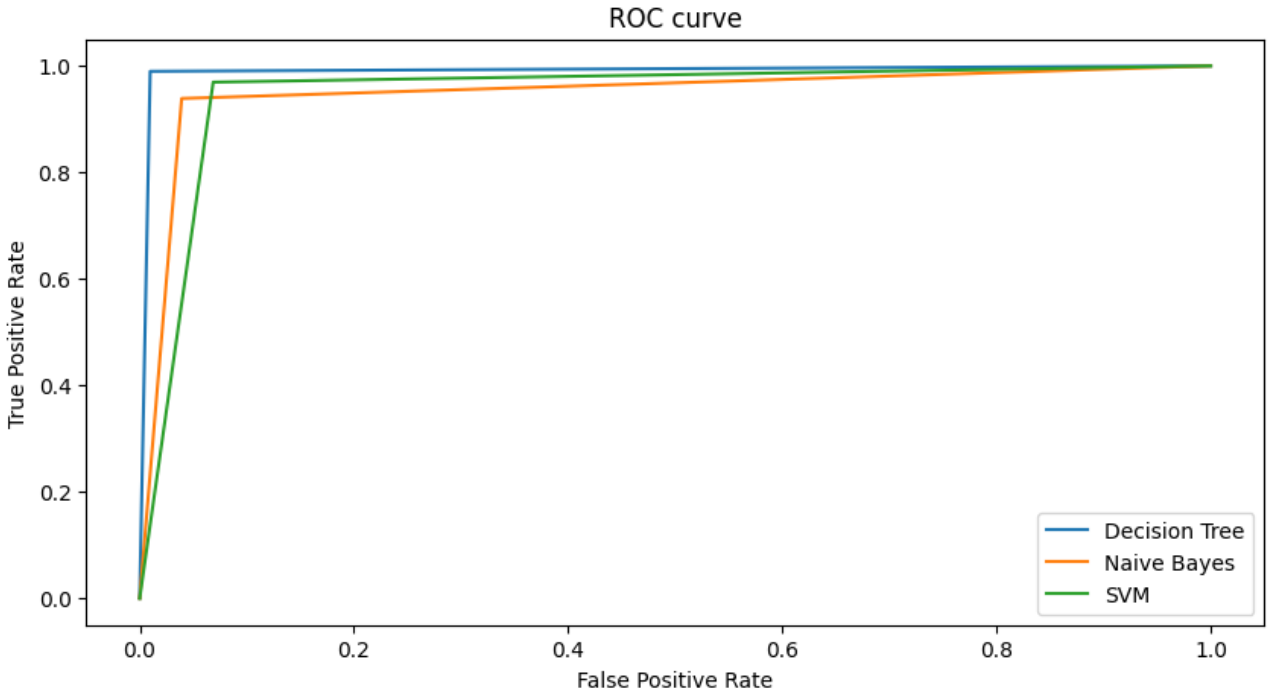
Questo grafico offre un'interpretazione visuale delle caratteristiche più cruciali secondo il modello SVM. Feature con pesi più alti sono considerate più influenti nel processo decisionale, indicando quali aspetti del testo sono considerati più determinanti nella classificazione.

I valori positivi e negativi associati alle feature nelle analisi di pesi di un modello di machine learning, come in questo caso con un modello SVM, indicano la direzione dell'effetto della feature sulla decisione del modello.

Nel contesto di un modello SVM lineare, i pesi rappresentano i coefficienti del piano decisionale, e le feature i vettori di supporto.

Ogni feature contribuisce in modo proporzionale al suo peso nel calcolo della funzione di decisione lineare. L'output di questa funzione viene poi utilizzato per determinare la classe predetta.

5.4. Curve ROC



I valori AUC ottenuti forniscono una panoramica delle prestazioni relative dei tre modelli:

- Decision Tree AUC: 0.9899: L'AUC estremamente vicino a 1 indica che il modello Decision Tree ha dimostrato una straordinaria capacità di separare le classi positive e negative. La curva ROC del Decision Tree è vicina all'angolo in alto a sinistra, suggerendo che il modello ha una bassa sensibilità alle variazioni delle soglie di classificazione e riesce a discriminare in modo eccellente tra le classi.
- Naive Bayes AUC: 0.9497: Un AUC superiore a 0.9 per il modello Naive Bayes suggerisce una buona capacità discriminante, sebbene leggermente inferiore rispetto al Decision Tree. Il modello Naive Bayes ha dimostrato di essere robusto nella distinzione tra le classi, ma potrebbe essere più sensibile alle variazioni delle soglie di classificazione rispetto al Decision Tree.
- SVM AUC: 0.9503: Il modello SVM mostra un'efficace capacità discriminante, con un AUC simile a quello del Naive Bayes. La curva ROC dell'SVM evidenzia la sua buona capacità di separazione tra le classi, sebbene potrebbe mostrare una leggera sensibilità alle variazioni delle soglie di classificazione.

Complessivamente, tutti e tre i modelli presentano prestazioni solide, con il Decision Tree che si distingue per l'AUC più alto, indicando una maggiore capacità di classificazione. Questi risultati sono incoraggianti e suggeriscono che i modelli stanno svolgendo bene nella discriminazione tra le classi target del problema.

5.5. Confronto dei misclassificati

Il confronto dei dati misclassificati non ha fornito ulteriori informazioni rilevanti, poiché la classe di appartenenza degli esempi classificati erroneamente non mostra una

sproporzione significativa, e i testi misclassificati non presentano feature eclatanti, come lunghezza estrema o numero anomalo di parole.

L'assenza di evidenti differenze nelle caratteristiche dei testi e nella distribuzione delle classi suggerisce che le misclassificazioni potrebbero derivare da pattern ricorrenti tipici dell'altro tipo di notizia. In altre parole, potrebbero esserci determinati schemi linguistici o strutture di frase che sono più comuni in un tipo di notizia e che possono ingannare il modello.

La classificazione basata sulla frequenza relativa e inversa delle parole nel testo può essere sensibile a questi pattern ricorrenti. Ad esempio, se alcune parole con frequenza più alta sono tipiche delle notizie reali, la presenza di queste parole in una notizia fake può portare a una classificazione errata.

È importante considerare l'aspetto semantico e contestuale delle parole, poiché la frequenza non tiene conto del significato delle parole all'interno del contesto.

6. Considerazioni finali e Sviluppi futuri

L'attuale stadio del progetto rappresenta solo una tappa iniziale nel percorso di miglioramento della capacità predittiva dei modelli di classificazione delle notizie fake e reali. Numerose opportunità di ottimizzazione e sviluppo sono ancora inesplorate, tra queste possiamo notare:

1. Rimozione delle Parole Comuni: Per migliorare ulteriormente il pre-processing, potrebbe essere vantaggioso non solo rimuovere le parole comuni, ma anche eseguire un'analisi più approfondita delle parole per identificare e rimuovere quelle che non contribuiscono significativamente alla distinzione tra le notizie fake e reali. Ciò potrebbe comportare una maggiore efficienza computazionale durante le fasi successive.
2. Stratificazione dell'Integrazione di Vettorizzatori: Quando si integra più vettorizzatori, come word2vec, GloVe e BERT, potrebbe essere utile eseguire una stratificazione o una ponderazione dei contributi di ciascun vettorizzatore. Questo approccio potrebbe bilanciare l'importanza delle rappresentazioni semantiche con quelle basate sulla frequenza delle parole, migliorando così la completezza della rappresentazione del testo.
3. Clustering Semantico con word2vec: L'aggiunta di un modello di clustering non supervisionato sfruttando rappresentazioni semantiche come word2vec è un'idea innovativa. Potrebbe essere interessante esplorare come il clustering basato sul significato semantico delle frasi potrebbe rivelare relazioni o pattern nascosti nelle notizie, contribuendo così a una comprensione più approfondita del dataset.

Questi miglioramenti mirano a raffinare ulteriormente il processo di pre-processing, integrare in modo più bilanciato le rappresentazioni vettoriali e sperimentare con l'uso di modelli di clustering semantico, offrendo così un approccio più sofisticato e completo alla classificazione delle notizie fake e reali.