



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

Confronto tra Reti Neurali Ricorrenti per l'Analisi dei Sentimenti sui Tweet di Twitter: Un'Indagine tra LSTM, BiLSTM e GRU.

RELATORE

Prof. Fabio **PALOMBA**

Dott. Stefano **LAMBIASE**

Università degli Studi di Salerno

CANDIDATO

Gennaro CECCO

Matricola: 0512107786

Anno Accademico 2022-2023

Questa tesi è stata realizzata nel

sesa^{lab}
SOFTWARE ENGINEERING
SALERNO

"Il fallimento nella vita è inevitabile. È il modo in cui apprendiamo e cresciamo da esso che conta".

- Stephen Hawking

Abstract

L'analisi dei sentimenti applicata ai tweet rappresenta un campo cruciale nell'era dei social media, offrendo insight preziosi sulle opinioni e le emozioni degli utenti. Questo studio esplora l'efficacia di modelli di reti neurali, inclusi LSTM, BiLSTM e GRU, nel riconoscere i sentimenti nei brevi e informali messaggi di Twitter.

I risultati indicano che la BiLSTM emerge come il modello più performante, raggiungendo un'accuracy del 90.76% sui dati di test, anche se con un leggero overfitting del 6.73%. La sua capacità di elaborare informazioni in entrambe le direzioni si traduce in una migliorata capacità di catturare pattern complessi. La LSTM, pur mostrando un'accuracy del 90.74%, evidenzia un overfitting leggermente inferiore (6.53%). La GRU, con un'accuracy del 89.95%, si colloca in una posizione intermedia, mostrando prestazioni accettabili.

La scelta tra i modelli dipende dalle esigenze specifiche dell'applicazione: la BiLSTM potrebbe essere preferibile per massimizzare la rilevazione degli eventi positivi, mentre la LSTM potrebbe essere più adatta a evitare falsi positivi. La nostra analisi dettagliata fornisce una guida utile per orientare questa scelta in base agli obiettivi specifici del problema.

Indice

Elenco delle Figure	iii
Elenco delle Tabelle	v
1 Introduzione	1
1.1 Contesto	1
1.2 Motivazioni ed Obiettivi	2
1.3 Risultati	3
1.4 Struttura della tesi	3
2 Stato dell'Arte	4
2.1 Sentiment Analysis	4
2.2 RNN utilizzate	5
3 Lavoro Svolto	8
3.1 Obiettivo	8
3.2 Metodologia	9
3.2.1 Dataset	10
3.2.2 Implementazione Reti Neurali Ricorrenti	13
3.2.3 Benchmark	19

4	Risultati	22
4.1	Risultati Addestramento	22
4.2	Risultati dei test di validazione isolati sui modelli	27
4.3	Matrici di confusione	29
4.4	Explainable AI: LIME	35
5	Conclusioni	46
	Bibliografia	48

Elenco delle figure

3.1	Step eseguiti per il raggiungimento dell'obiettivo.	10
3.2	Configurazione RNN-LSTM.	17
3.3	Configurazione RNN-BiLSTM.	18
3.4	Configurazione RNN-GRU.	19
4.1	Dati addestramento LSTM	23
4.2	Dati addestramento BiLSTM	24
4.3	Dati addestramento GRU	26
4.4	Validazione Reti su dati che non ha mai visto	28
4.5	Matrice di Confusione LSTM	30
4.6	Matrice di Confusione BiLSTM	32
4.7	Matrice di Confusione GRU	34
4.8	Analisi Lime-LSTM sul tweet positivo	36
4.9	Analisi Lime-LSTM sul tweet negativo	38
4.10	WordCloud-LSTM	39
4.11	Analisi Lime-BiLSTM sul tweet positivo	40
4.12	Analisi Lime-BiLSTM sul tweet negativo	41
4.13	WordCloud-BiLSTM	42
4.14	Analisi Lime-GRU sul tweet positivo	43
4.15	Analisi Lime-GRU sul tweet negativo	44

4.16 WordCloud-GRU	45
------------------------------	----

Elenco delle tabelle

4.1	Risultati delle valutazioni delle reti LSTM, BiLSTM e GRU	28
4.2	Valori Matrice di confusione LSTM	30
4.3	Valori di Precision, Recall e F1-Score modello LSTM	31
4.4	Valori Matrice di confusione BiLSTM	32
4.5	Valori di Precision, Recall e F1-Score modello BiLSTM	33
4.6	Valori Matrice di confusione GRU	34
4.7	Valori di Precision, Recall e F1-Score modello GRU	35
5.1	Confronto tra LSTM, BiLSTM e GRU su Precision, Recall e F1-Score .	46

CAPITOLO 1

Introduzione

1.1 Contesto

Il nostro lavoro si colloca nell'ambito dell'analisi dei sentimenti applicata ai dati provenienti dai social media, con un'attenzione particolare verso i tweet. Questo settore è di grande importanza oggi, considerando che i social media rappresentano un terreno fertile per le opinioni e le emozioni degli utenti. L'obiettivo principale è sviluppare modelli di intelligenza artificiale in grado di interpretare con precisione ed efficienza il tono emotivo dei messaggi testuali. Nei contesti delle piattaforme di social media, come Twitter, la comprensione del sentimento è cruciale per vari settori. Ad esempio, aziende e organizzazioni possono trarre grandi benefici dall'analisi dei sentimenti per valutare la percezione del pubblico riguardo ai propri prodotti, servizi o iniziative. Inoltre, un'analisi accurata dei sentimenti può essere utilizzata per monitorare l'andamento dell'opinione pubblica su questioni rilevanti, consentendo risposte tempestive a eventi o tendenze emergenti. La nostra applicazione si focalizza specificamente sulla valutazione dei sentimenti nei tweet, un formato di comunicazione breve e spesso informale. Questa scelta è motivata dalla diffusione ampia di Twitter come piattaforma di microblogging e dalla sua significativa influenza nelle conversazioni online. L'analisi dei sentimenti nei tweet può fornire insight

preziosi su come le persone percepiscono e reagiscono a vari argomenti, eventi o tendenze culturali. L'utilizzo di modelli avanzati basati su reti neurali, come le LSTM, BiLSTM e GRU, nell'analisi dei sentimenti mira a superare le limitazioni dei metodi tradizionali, consentendo una comprensione più approfondita e contestualizzata del linguaggio naturale. L'integrazione di approcci di spiegabilità, come LIME, aggiunge un ulteriore strato di interpretabilità alle previsioni del modello, rendendo il processo decisionale più trasparente e comprensibile.

1.2 Motivazioni ed Obiettivi

In un mondo sempre più immerso nell'utilizzo di social media, come Twitter, emergono nuove sfide nell'interpretazione delle brevi e spesso informali espressioni di sentimenti presenti nei tweet. Per rispondere a questa esigenza, ci siamo impegnati nello sviluppo di modelli avanzati di intelligenza artificiale. La nostra missione principale è stata quella di esplorare come le reti neurali LSTM, BiLSTM e GRU possano analizzare efficacemente il sentiment nei tweet. Questi modelli promettono di catturare le sfumature linguistiche e le relazioni semantiche in modo più avanzato rispetto agli approcci tradizionali. La ricerca è stata guidata dalla volontà di superare le sfide uniche poste dalla sinteticità e dall'informalità dei tweet. La crescente necessità di comprendere a fondo le opinioni degli utenti espresse attraverso i social media ha ulteriormente motivato il nostro impegno. Il nostro studio si inserisce nel panorama della ricerca sull'analisi dei sentimenti, mirando a fornire un'analisi comparativa delle performance dei modelli LSTM, BiLSTM e GRU su dati specifici dei tweet. I nostri obiettivi specifici sono stati guidati dalla volontà di valutare le prestazioni di ciascun modello attraverso metriche chiave come accuracy, precision, recall e F1-score. L'introduzione di tecniche di spiegabilità, come LIME, è stata una componente cruciale per rendere il processo decisionale dei modelli più chiaro e interpretabile. Nel complesso, miriamo a fornire un quadro completo delle capacità e dei limiti di questi modelli nell'analisi dei sentimenti applicata ai tweet, contribuendo così alla letteratura esistente in questo campo dinamico.

1.3 Risultati

I dettagli dei risultati ottenuti dalla nostra analisi dei sentimenti sui tweet rivelano che la LSTM ha ottenuto un'accuracy del 90.74%, pur mostrando un leggero overfitting del 6.53%. La BiLSTM ha superato la LSTM, conseguendo un'accuracy del 90.76% e dimostrando una maggiore capacità di riconoscere pattern complessi, sebbene con un overfitting leggermente più elevato del 6.73%. La GRU ha riportato risultati accettabili con un'accuracy del 89.95%, ma con un overfitting del 6.98%. Il confronto tra i modelli mette in luce la dipendenza della scelta dal contesto specifico dell'applicazione. Se l'obiettivo principale è massimizzare la rilevazione degli eventi positivi, la BiLSTM risulta preferibile. Al contrario, se è essenziale evitare falsi positivi, la LSTM potrebbe essere più adatta. La decisione finale potrebbe richiedere un bilanciamento tra precisione e recall, e la nostra analisi approfondita fornisce una guida preziosa per orientare questa scelta in base agli obiettivi specifici del problema.

1.4 Struttura della tesi

La Sezione 2 illustra lo stato dell'arte correlato al sentiment analysis e all'utilizzo dei modelli RNN. La Sezione 3 riporta la metodologia e le tecniche utilizzate per raggiungere gli obiettivi della tesi, mentre la Sezione 4 riporta i risultati ottenuti. La sezione 5 conclude la tesi con l'analisi dei risultati.

CAPITOLO 2

Stato dell'Arte

In questo capitolo viene fornita una panoramica dello stato dell'arte sul deep learning e delle RNN applicato al contesto del sentiment analysis

2.1 Sentiment Analysis

I social media costituiscono un canale di comunicazione significativo attraverso il quale gli individui condividono le proprie opinioni e punti di vista su una vasta gamma di argomenti. Questo continuo flusso di interazioni genera una considerevole quantità di informazioni non strutturate. Le imprese devono elaborare in dettaglio tali sentimenti per analizzare i dati e ottenere informazioni commerciali rilevanti[1]. L'analisi del sentimento, o opinion mining, è un settore attivo nella ricerca di elaborazione del linguaggio naturale, focalizzato sull'identificare, estrarre e organizzare i sentimenti espressi dagli utenti sui social network. Molte ricerche recenti adottano approcci di apprendimento automatico per affrontare le sfide legate alle scelte di rappresentazione dei dati, coinvolgendo esperti del settore nella creazione di estrattori di caratteristiche potenti. Il deep learning, in particolare, si è dimostrato un modello computazionale efficace nel rilevare rappresentazioni semantiche complesse dei testi [2]. L'acquisizione di dati da piattaforme sociali come Twitter offre significa-

tive opportunità per raggiungere obiettivi cruciali come la previsione di mercato, la raccomandazione di prodotti e la comprensione del sentimento degli utenti. Tuttavia, la gestione di dati non strutturati su questi social media è una sfida complessa. Per superare questo ostacolo, gli algoritmi di apprendimento profondo emergono come una soluzione efficace per affrontare con successo tali complessità e sfruttare appieno il potenziale informativo di questi dati sociali [3]. Il sentiment analysis è considerato una branca del Natural Language Processing (NLP), che a sua volta è un campo di studio dedicato a consentire alle macchine di comprendere, interpretare e generare il linguaggio umano per interagire in modo più naturale con gli esseri umani.

2.2 RNN utilizzate

Il modello linguistico costituisce il fondamento del processing del linguaggio naturale (NLP). L'utilizzo diffuso del modello basato sulla rete neurale ricorrente (RNN) come tecnologia di modellizzazione linguistica è dovuto alla sua potente capacità di apprendimento automatico e analisi di dati sequenziali[4]. Nel contesto più ampio, il Natural Language Processing (NLP) emerge come una disciplina cruciale nell'Intelligenza Artificiale (IA), agendo come interfaccia tra utenti e computer per agevolare interazioni attraverso vari algoritmi computazionali. Gli algoritmi NLP attuali hanno molteplici applicazioni, tra cui la rilevazione di messaggi indesiderati, l'analisi di resoconti di riunioni, la risposta automatica a domande, la sintesi di articoli mediatici, la traduzione automatica e altro ancora. Il Deep Learning, un'estensione dell'apprendimento automatico e dell'IA, istruisce i computer a imparare dall'esperienza, emulando il processo cognitivo umano. Recentemente, modelli di Deep Learning hanno dimostrato successo in diverse attività di processing del linguaggio naturale, sfruttando architetture come CNN (Convolutional Neural Network), RNN (Recurrent Neural Network) e modelli di attenzione [5]. Le reti neurali ricorrenti (RNN) si distinguono come un approccio innovativo per navigare nei dati sequenziali, come il testo. La loro caratteristica distintiva risiede nella capacità di conservare una "memoria" delle informazioni passate durante l'analisi di una sequenza di dati. Questa memoria sequenziale permette alle RNN di catturare e considerare il contesto temporale delle informazioni, risultando cruciale per l'analisi di testi e dati strutturati

in modo sequenziale. A differenza delle reti neurali tradizionali, le RNN sono dotate di uno "stato nascosto" che si aggiorna costantemente ad ogni passaggio temporale, influenzato sia dai dati di input attuali che dallo stato nascosto precedente. Questa caratteristica consente alle RNN di sviluppare una comprensione dinamica del contesto sequenziale, particolarmente rilevante nelle applicazioni di sentiment analysis, dove la valutazione di una parola o frase può dipendere strettamente dal contesto circostante.

Nello studio di questa tesi, i modelli RNN utilizzati sono:

- LSTM: è un tipo di RNN progettato per superare le limitazioni nel catturare dipendenze a lungo termine nelle sequenze, utilizzando una struttura complessa di celle di memoria.
- BiLSTM: è una variante di LSTM che elabora una sequenza sia in avanti che all'indietro, catturando contesti temporali sia passati che futuri per una migliore comprensione del contesto.
- GRU: è un'unità RNN che, come LSTM, gestisce il problema della scomparsa del gradiente utilizzando un meccanismo di gating, ma con una struttura più semplice, risultando spesso in una computazione più efficiente rispetto alle LSTM.

Le RNN-LSTM vengono utilizzate particolarmente nello studio del sentiment analysis. L'articolo *Sentiment Analysis of US Airlines Tweets Using LSTM/RNN* tratta dell'utilizzo delle RNN-LSTM per eseguire sentiment analysis, inoltre introduce le BiLSMT come RNN per aumentare le prestazioni rispetto alla LSTM[6]. Questo tipo di rete neurale, nel campo del sentiment analysis, è stata utilizzata anche per fare previsioni sui sentimenti degli investitori del mercato azionario[7]. Per superare i limiti raggiunti con le LSTM, vengono introdotte nel campo del sentiment analysis anche le BiLSTM. Secondo l'articolo *DICE: Deep Intelligent Contextual Embedding for Twitter Sentiment Analysis*[8] l'utilizzo della BiLSTM combinato con una tecnica di "Incorporamento Contestuale Intelligente Profondo" ha dato risultati molto positivi rispetto ad altri modelli utilizzati. Come ultima tipologia di RNN utilizzata abbiamo la GRU. Questo tipo di rete, nel campo del sentiment analysis viene utilizzato in

combinazione con altre tipologie di RNN come le LSTM o le BiLSTM[9][10]. Nel contesto di questo lavoro di tesi, le GRU sono state utilizzate nella loro interezza, senza combinarle con altri modelli, in modo da poter effettuare un confronto oggettivo dei 3 modelli. Le tecniche di analisi tramite RNN vedono la loro applicazione anche per valutare la polarità delle recensioni dei film [11] e questo fa capire quanto sia importante l'opinione pubblica, quindi in maniera intrinseca rende di estrema importanza uno studio approfondito sui modelli di deep learning e il loro utilizzo nel sentiment analysis anche in situazioni critiche, come il Covid-19. Anche su questo argomento sono stati fatti numerosi studi, uno dei tanti che abbiamo analizzato è stato quello di Ayşenur Topbaş et al. [12]

CAPITOLO 3

Lavoro Svolto

In questo capitolo viene descritta la metodologia applicata per valutare le prestazioni delle diverse reti implementate e le tecniche utilizzate per poterle implementare.

3.1 Obiettivo

L'obiettivo di questo lavoro di tesi è valutare e analizzare 3 tipi differenti di reti neurali ricorrenti, con lo scopo di individuare quale tra queste abbia dato le prestazioni migliori per l'analisi dei sentimenti dei tweet da Twitter. Per raggiungere questo obiettivo si è scelto di utilizzare Twitter come social media in quanto ha milioni di utenti che condividono pensieri, opinioni e informazioni ogni giorno. Ciò offre un vasto pool di dati disponibili per l'analisi dei sentimenti. Inoltre Twitter copre una vasta gamma di argomenti, dalle notizie correnti alle tendenze di intrattenimento, alle discussioni su prodotti e servizi. Questa varietà di argomenti permette di esaminare sentimenti su vari settori. Una volta ottenuti i dati, abbiamo costruito le nostre 3 reti neurali e confrontato i risultati per poterle valutare.

3.2 Metodologia

L'analisi dei sentimenti, o opinion mining, è un campo attivo nella ricerca di elaborazione del linguaggio naturale[2] ed è per questo che è nata l'idea di confrontare diversi tipi di RNN in modo da individuare la rete con le prestazioni più alte. Per poter arrivare al raggiungimento del nostro obiettivo abbiamo dovuto dividere il nostro sviluppo in steps. Il primo passo vede la ricerca del dataset, la nostra fonte di dati, che permette al modello di apprendere relazioni tra gli input sequenziali e l'output. Il secondo passo è stato quello di Estrazione delle features, ovvero quel processo in cui selezioniamo solo le proprietà di maggior peso oppure cruciali per il corretto funzionamento della nostra rete. Una volta ottenuti i nostri dati, andiamo a rimuovere dal testo tutte quelle parole che fanno rumore oppure caratteri di punteggiatura che la rete non considera. Tutto questo viene fatto nella fase di pulizia del testo. Eseguita questa fase, i dati sono pronti per l'utilizzo ma prima dobbiamo costruire i nostri modelli: LSTM-BiLSTM-GRU. Dopo la costruzione segue lo step che riguarda l'addestramento della rete. Una volta terminato, il modello può essere già utilizzato, ma visto che questa tesi ha come obiettivo quello di confrontare diversi tipi di RNN, abbiamo aggiunto tecniche di Explainable-AI per poter meglio capire il peso che la rete attribuisce alle parole nei tweet e da qui dedurre se l'addestramento è andato come da programma. Infine abbiamo la fase di validazione e da qui la valutazione dei nostri modelli.

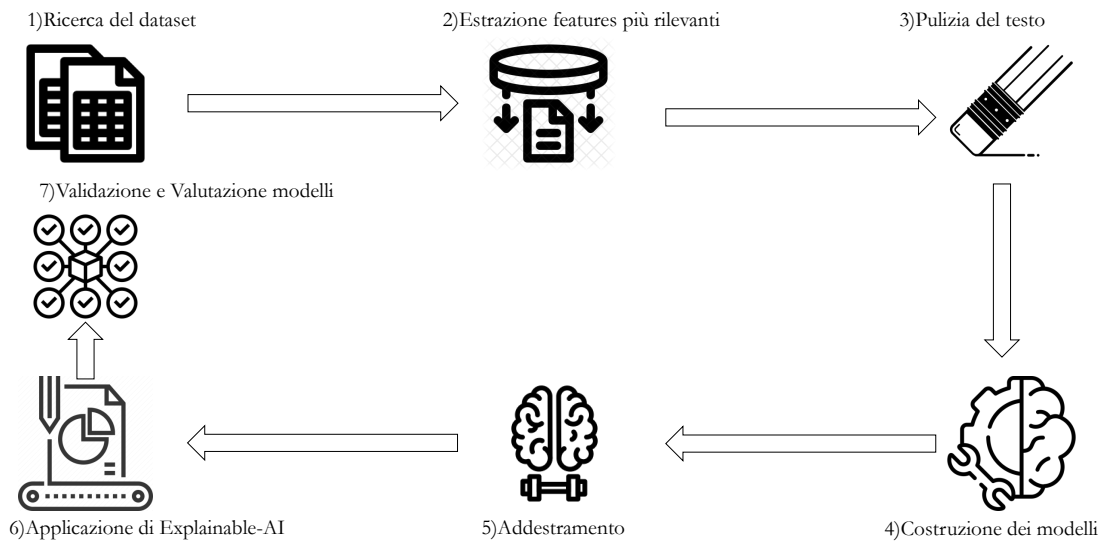


Figura 3.1: Step eseguiti per il raggiungimento dell'obiettivo.

3.2.1 Dataset

I dati utilizzati per l'addestramento delle reti vengono da due dataset differenti. I dataset in questione sono:

- TwitterData.csv (www.kaggle.com)
- OtherNegativi.csv è stato creato prendendo tweet negativi dal dataset training.1600000.processed.noemoticon.csv (www.kaggle.com)

In alcuni casi è possibile che il dataset soffra del problema dello **sbilanciamento**, ovvero: situazione in cui il numero di campioni in ciascuna classe di un problema di classificazione è significativamente diverso. Questa disparità può presentare sfide nella fase di addestramento e valutazione dei modelli di apprendimento automatico. Questo era il problema di cui soffriva il primo dataset TwitterData.csv. In questo dataset avevamo un numero di tweet positivi maggiore rispetto a quello dei tweet negativi, erano state adottate tecniche di *undersampling*, quindi ridurre il numero di istanze dei tweet positivi, ma questo ha portato ad un eccessivo overfitting da parte del modello. E' stata utilizzata anche la tecnica del *oversampling*, quindi aumentare il numero di istanze negative, ma anche in questo caso le prestazioni dei modelli erano sempre basse. Per risolvere il problema della classe di minoranza è stato uti-

lizzato un altro dataset, OtherNegativi.csv. Questo dataset contiene esattamente il numero di istanze di tweet negativi per avere un bilanciamento con quelli positivi. Per creare questo insieme di dati, è stato utilizzato originariamente il dataset training.1600000.processed.noemoticon.csv che contiene circa 1.600.000 tweet. Da questo sono stati presi i tweet mancanti, esclusivamente negativi, e creato un altro dataset. Una volta uniti questi due insiemi di dati abbiamo avuto un totale di 144.498 tweet per i nostri modelli. TwitterData.csv vede solo due caratteristiche, ovvero:

- **clean-text:** indica il testo vero e proprio del tweet
- **category:** indica la polarità del tweet, 0 indica una polarità positiva, -1 una polarità negativa.

Sono state prese in considerazione entrambe le features. Per quanto riguarda OtherNegativi.csv, come detto anche in precedenza, questo dataset è stato creato prendendo una parte dei tweet negativi da training.1600000.processed.noemoticon.csv, che vede come features:

- **target:** indica la polarità del tweet, 0 indica una polarità negativa, 2 indica una polarità neutrale, 4 indica una polarità positiva.
- **ids:** indica l'id del tweet.
- **date:** indica la data del tweet.
- **flag:** indica che se il sistema non ha una query specifica da elaborare, il flag viene impostato su "NO-QUERY" per indicare l'assenza di una query valida.
- **user:** indica l'utente che ha tweettato
- **text:** indica il testo del tweet.

Analizzando le caratteristiche di questo dataset, vengono prese in considerazione solo: target e text in quanto sono quelle più importanti e significative al fine dello studio di questa tesi. Una volta identificate le features da prendere in considerazione, si devono eseguire delle operazioni per garantire la compatibilità con l'altro dataset. Poiché per il nostro scopo sono necessari solo i tweet negativi da questo dataset, vengono

selezionati solo quelli con polarità 0. Una volta ottenuti i tweet, la loro polarità viene mappata a -1 e inserita nel dataset creato denominato 'OtherNegativi.csv'.

Una volta ottenuti i dati, questi devono subire una fase di *pre-elaborazione* prima di essere lemmatizzati e tokenizzati. Questa fase di text-preprocessor prevede:

- Rimozione di possibili righe nulle nell'insieme di dati.
- Rimozione di URL, Tag HTML e Emoji: Rimuove gli URL, i tag HTML e le emoji usando espressioni regolari.
- Rimozione dell'Username: Rimuove gli username di Twitter (ad esempio, "@username").
- Trasformazione in Minuscolo: Converte l'intero testo in minuscolo.
- Riduzione dei Caratteri Ripetuti: Rimuove i caratteri ripetuti, mantenendo solo un'occorrenza.
- Rimozione di Caratteri Speciali Aggiuntivi.
- Gestione delle Negazioni: Trasforma la parola "not" in "not_" per gestire le negazioni.
- Rimozione delle Stopword: Rimuove le comuni stopwords inglesi usando NLTK, Natural Language Toolkit (NLTK) è una libreria Python open-source progettata per l'elaborazione del linguaggio naturale (NLP).

Una volta terminate queste operazioni andiamo ad eseguire un'ulteriore operazione che è quella della lemmatizzazione, ovvero ridurre le parole di una frase al loro lemma o forma di base. Il lemma è la forma canonica di una parola, rappresentando la radice linguistica comune a tutte le sue forme flessive. Ad esempio, lemmatizzando il verbo "running", si otterrebbe il lemma "run". La lemmatizzazione aiuta a standardizzare le parole, semplificando l'analisi del testo e consentendo la riduzione delle diverse forme di una parola a una forma base comune. Come ultima operazione troviamo la tokenizzazione dei tweet che consiste nel suddividere un testo in unità più piccole, chiamate token, che possono essere parole, frasi o addirittura caratteri, a seconda del livello di dettaglio desiderato. Il tokenizer utilizzato è della libreria python

"keras", che esegue una tokenizzazione di tipo sequenziale, quindi suddividerà ogni testo in parole individuali e assegnerà un numero intero unico a ciascuna parola, questa suddivisione permetterà di creare il vocabolario che verrà utilizzato dal nostro modello il nostro modello.

3.2.2 Implementazione Reti Neurali Ricorrenti

Questo progetto di tesi vede l'implementazione di 3 reti neurali ricorrenti:

- LSTM: Long short-term memory
- BiLSTM: Bidirectional Long Short-Term Memory
- GRU: Gated recurrent unit

Per implementare queste 3 versioni è stata utilizzata la libreria Keras ed il framework TensorFlow sia per la costruzione vera e propria del modello che per il suo addestramento. Prima di dare i nostri dati puliti alle reti, dobbiamo eseguire un'ultima operazione che è quella del padding, che consiste nell'aggiungere una serie di 0 a una sequenza in modo da raggiungere una lunghezza desiderata. Come lunghezza abbiamo prefissato un valore pari alla lunghezza del tweet più lungo all'interno del dataset. Quindi una volta che abbiamo eseguito anche l'operazione di padding possiamo creare le nostre reti. Per tutte le varianti RNN implementate, troviamo in comune:

la funzione di attivazione sigmoide, utilizzata per problemi di classificazione binaria. La funzione di attivazione è una componente fondamentale nelle reti neurali artificiali ed è utilizzata per introdurre non linearità nei singoli neuroni o unità di una rete. Questa non linearità è cruciale perché permette alle reti neurali di imparare relazioni complesse tra le variabili di input e di adattarsi a modelli più complessi. La formula della funzione di attivazione sigmoide è:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

- $\sigma(x)$ rappresenta l'output della funzione sigmoide
- e è la base del logaritmo naturale

- x è l'input della funzione sigmoide è la somma ponderata degli input provenienti dal layer precedente, inclusi i pesi e il bias

La funzione di attivazione sigmoide "schiaffia" l'output a valori compresi tra 0 e 1. Come funzione di perdita è stata utilizzata la *binary_crossentropy*. Il suo obiettivo è misurare quanto bene il modello sta performando rispetto agli obiettivi desiderati durante la fase di addestramento. La sua formula è:

$$L(y, \hat{y}) = -(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y}))$$

- $L(y, \hat{y})$ è la perdita per un singolo esempio
- y è l'etichetta reale della classe (0 o 1)
- \hat{y} è l'output predetto dal modello, una probabilità compresa tra 0 e 1.

Questa formula misura la discrepanza tra le previsioni del modello \hat{y} e l'etichetta reale y . La binary cross-entropy è particolarmente adatta per problemi di classificazione binaria. L'ottimizzatore che è stato utilizzato è *Adam*. Un ottimizzatore opera utilizzando il gradiente della funzione di perdita rispetto ai parametri del modello. Il gradiente indica la direzione e l'ampiezza del cambiamento più rapido nella funzione di perdita. Gli ottimizzatori regolano i parametri muovendosi nella direzione opposta al gradiente, cercando di raggiungere il minimo della funzione di perdita. L'ottimizzatore Adam adatta il tasso di apprendimento per ciascun parametro individualmente, calcolando le stime del primo e secondo momento dei gradienti inoltre applica una correzione del bias per contrastare la tendenza dell'ottimizzatore ad avere una bassa varianza nei primi passaggi. Nel nostro modello, stiamo utilizzando l'accuratezza come metrica principale. L'accuratezza è una misura della percentuale di previsioni corrette fatte dal modello rispetto al totale delle previsioni. È stata scelta questa metrica perché vogliamo valutare quanto bene il nostro modello riesce a classificare correttamente le istanze. In un problema di classificazione binaria, l'accuratezza fornisce un'indicazione diretta della percentuale di previsioni corrette. La formula è:

$$\text{Accuracy} = \frac{\text{TP}}{\text{Total}}$$

In un problema di classificazione binaria, True Positives TP rappresenta il numero di casi in cui il modello ha correttamente previsto la classe positiva, e $Total$ rappresenta il totale delle istanze nel set di dati di valutazione. Inoltre controlliamo anche la binary cross-entropy loss, che è una misura comune di quanto le previsioni del modello si discostino dalle etichette reali in un problema di classificazione binaria. La formula è:

$$\text{Binary Cross-Entropy Loss} = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)]$$

- N è il numero totale di esempi nel set di dati.
- y_i è l'etichetta vera della i -esima istanza (0 o 1).
- \hat{y}_i è la previsione del modello per la i -esima istanza.

Un problema delle RNN e in generale dei modelli di ML è l'overfitting che si verifica quando il modello si adatta troppo ai dettagli specifici dei dati di addestramento temporali. Questo comportamento eccessivo porta alla perdita della capacità di generalizzare su nuove sequenze o situazioni. Poiché le RNN gestiscono dati sequenziali, l'overfitting può essere causato dalla complessità eccessiva della struttura del modello o dalla disponibilità limitata di dati temporali per l'addestramento. Per evitare questo, oltre alle tecniche di dropout e l'inserimento dei livelli di Batch, utilizziamo altre due tecniche che sono:

- **Early Stopping** : L'early stopping si basa sul monitoraggio della performance del modello su un set di dati di validazione. Durante l'addestramento, la metrica di valutazione, nel nostro caso la *loss* viene misurata sul set di dati di validazione ad intervalli regolari (epoche). Se la performance sul set di dati di validazione smette di migliorare per un certo periodo *patience*, l'addestramento viene interrotto prematuramente.
- **Reduce Learning Rate on Plateau**: Riduce il tasso di apprendimento se la metrica presa in considerazione, nel nostro caso la *loss* non migliora per un certo periodo *patience*. Nel nostro caso il tasso di apprendimento viene ridotto di un *factor* pari a 0.5, quindi il tasso di apprendimento viene ridotto del 50%. Il tasso

di apprendimento non viene ridotto al di sotto di un valore stabilito ovvero il min_lr che nel nostro caso corrisponde al $1e-5$.

Essendo che questo elaborato ha come obiettivo, il confronto tra 3 modelli di RNN, si è cercato di dare un impostazione simile a tutte e 3 le reti, infatti in tutti i modelli abbiamo:

- Embedding
- **Layer LSMT/BiLSMT/GRU**
- Batch Normalization Layer
- Dense Layer

Il livello di embedding converte le parole in vettori densi di dimensione 32. Il layer di batch serve a normalizzare l'output ed è stato aggiunto per ridurre l'overfitting. L'ultimo è un layer denso con funzione di attivazione *sigmoide*.

Per l'implementazione della rete LSTM, abbiamo aggiunto il layer *LSTM*, composto da 16 unità che sono i neuroni della nostra rete, inoltre a questo livello è stato aggiunto un dropout, tecnica che disattiva casualmente i neuroni della rete, dello 0.4. L'aggiunta del dropout permette di ridurre l'overfitting.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 44, 32)	3490560
lstm (LSTM)	(None, 16)	3136
batch_normalization (Batch Normalization)	(None, 16)	64
dense (Dense)	(None, 1)	17
Total params: 3493777 (13.33 MB)		
Trainable params: 3493745 (13.33 MB)		
Non-trainable params: 32 (128.00 Byte)		

Figura 3.2: Configurazione RNN-LSTM.

Per l'implementazione della rete BiLSMT, abbiamo aggiunto il layer *Bidirectional LSTM*, composto da 16 unità. Anche per questo livello del seguente modello è stato aggiunto il dropout.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 44, 32)	3490560
bidirectional (Bidirectional)	(None, 32)	6272
batch_normalization (Batch Normalization)	(None, 32)	128
dense (Dense)	(None, 1)	33
Total params: 3496993 (13.34 MB)		
Trainable params: 3496929 (13.34 MB)		
Non-trainable params: 64 (256.00 Byte)		

Figura 3.3: Configurazione RNN-BiLSTM.

Per l'implementazione della rete GRU, abbiamo aggiunto il layer *GRU*, composto da 16 unità. Anche per questo livello del seguente modello è stato aggiunto il dropout.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 44, 32)	3490560
gru (GRU)	(None, 16)	2400
batch_normalization (Batch Normalization)	(None, 16)	64
dense (Dense)	(None, 1)	17
Total params: 3493041 (13.32 MB)		
Trainable params: 3493009 (13.32 MB)		
Non-trainable params: 32 (128.00 Byte)		

Figura 3.4: Configurazione RNN-GRU.

3.2.3 Benchmark

Una volta costruiti i nostri modelli, per avviare la fase di valutazione dobbiamo iniziare dal loro addestramento. L'addestramento di una rete neurale è il processo attraverso il quale la rete apprende i modelli e le rappresentazioni dei dati a partire dai dati di addestramento. Durante questa fase, il modello ottimizza i suoi parametri in modo che la sua uscita si avvicini quanto più possibile alle etichette corrette associate ai dati di addestramento. L'addestramento dei modelli dura un totale di 10 epoche, in quanto verso la sesta/settimana epoca il modello non migliora e quindi le condizioni di arresto, descritte in precedenza, fermano l'addestramento. Ogni epoca vede una batch_size di 32, perchè vogliamo aiutare il modello nella generalizzazione poichè con una dimensione di batch così impostata, il modello viene aggiornato più frequentemente. Inoltre viene utilizzato un 20% dei dati in input in ogni epoca del modello, per la validazione. Questo split dei dati aiuta a capire come si comporta la rete su dati che non ha mai visto. Per essere sicuri che i risultati riportati nella fase di addestramento e validazione siano precisi, si è pensato di prendere un altro 20% dei dati all'inizio della creazione del dataset per avviare un processo di validazione

isolato. Questo processo isolato viene eseguito solo al termine dell'addestramento della rete. I primi valori che consideriamo per valutare le prestazioni dei nostri modelli, sono i report delle fasi di addestramento. Oltre ai due test di validazione che vengono eseguiti per monitorare le prestazioni del nostro modello andiamo a creare le matrici di confusione (una per ogni modello). Essa mostra il numero di predizioni, corrette ed errate fatte dal modello su un set di dati di test, organizzate in una tabella. La matrice di confusione è composta da quattro elementi principali:

- **True Positives (TP):** Il modello ha correttamente previsto la classe positiva.
- **True Negatives (TN):** Il modello ha correttamente previsto la classe negativa.
- **False Positives (FP):** Il modello ha erroneamente previsto la classe positiva quando in realtà era negativa (Errore di Tipo I).
- **False Negatives (FN):** Il modello ha erroneamente previsto la classe negativa quando in realtà era positiva (Errore di Tipo II).

Per creare le matrici di confusione abbiamo utilizzato le librerie *sklearn* e *seaborn*. Dopo aver addestrato il modello e ottenuto le previsioni sul set di dati di validazione (X_{val}), convertiamo le probabilità predette in predizioni binarie utilizzando la soglia dello 0.5. Successivamente, andiamo a calcolare la matrice di confusione utilizzando la funzione *confusion_matrix* di *sklearn.metrics*. Per poter dare un senso o per meglio dire spiegare le scelte fatte dal nostro modello ci avvaliamo di *LIME*. *LIME* opera secondo il principio di spiegabilità locale, fornendo una comprensione dettagliata delle decisioni del modello su esempi specifici. La fase di spiegazione *LIME* inizia con l'inizializzazione di un oggetto *LimeTextExplainer*, che funge da ponte tra il modello e il processo di spiegazione. Questo oggetto è configurato con le classi di output, "Negative" e "Positive", e diventa la chiave per generare spiegazioni locali. La funzione *predict_fn* è di importanza cruciale per *LIME*. Questa funzione personalizzata, traduce i testi in sequenze numeriche, le paddeggia alle dimensioni richieste dal modello e restituisce le probabilità di predizione. Questo permette a *LIME* di manipolare i testi e valutare come tali manipolazioni influenzano le previsioni del modello, che è un aspetto fondamentale per ottenere spiegazioni significative. Con l'approccio utilizzato esploriamo diverse istanze di testo e dimensioni di testo nella funzione

explore_instances_and_text_sizes, LIME genera spiegazioni sotto forma di feature più influenti. Queste feature evidenziano quali parole o token hanno avuto un impatto significativo sulla predizione del modello per una specifica istanza.

CAPITOLO 4

Risultati

In questo capitolo vengono analizzate le performance e i risultati dei modelli RNN.
Nello specifico vengono analizzate le metriche, le matrici e le spiegazioni LIME

4.1 Risultati Addestramento

Questa analisi preliminare ci fornisce insights fondamentali sul comportamento dinamico dei nostri modelli durante la fase di apprendimento.

Il primo addestramento che andremo ad analizzare è quello che riguarda il modello LSTM.

```

Epoch 1/10
2890/2890 [=====] - 177s 60ms/step - loss: 0.3666 - accuracy: 0.8356 - val_loss: 0.2831 - val_accuracy: 0.8785 - lr: 0.0010
Epoch 2/10
2890/2890 [=====] - 167s 58ms/step - loss: 0.2243 - accuracy: 0.9106 - val_loss: 0.2475 - val_accuracy: 0.8957 - lr: 0.0010
Epoch 3/10
2890/2890 [=====] - 161s 56ms/step - loss: 0.1628 - accuracy: 0.9391 - val_loss: 0.2410 - val_accuracy: 0.9054 - lr: 0.0010
Epoch 4/10
2890/2890 [=====] - 155s 54ms/step - loss: 0.1283 - accuracy: 0.9531 - val_loss: 0.2605 - val_accuracy: 0.9026 - lr: 0.0010
Epoch 5/10
2890/2890 [=====] - 154s 53ms/step - loss: 0.1049 - accuracy: 0.9619 - val_loss: 0.2640 - val_accuracy: 0.9087 - lr: 0.0010
Epoch 6/10
2890/2890 [=====] - 154s 53ms/step - loss: 0.0767 - accuracy: 0.9731 - val_loss: 0.2817 - val_accuracy: 0.9095 - lr: 5.0000e-04
C:\Users\Genny\PycharmProjects\Sentiment_Analysis\venv\lib\site-packages\keras\src\engine\training.py:3080: UserWarning: You are saving your model as an HDF5 file via 'model.save'
  saving_api.save_model(
Learning rate minimo raggiunto: 0.0005000000237487257
L'addestramento è stato interrotto all'epoca 6 a causa di early stopping.
La percentuale di overfitting è: 6.53%

```

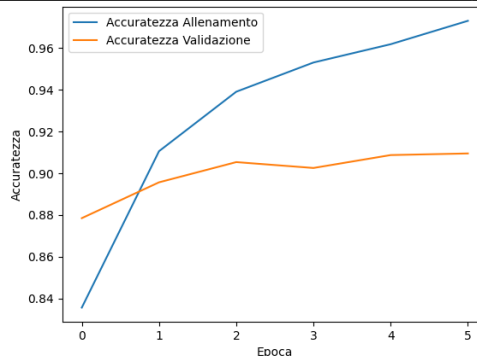


Figura 4.1: Dati addestramento LSTM

Come evidenziato nella Figura 4.1, l'addestramento della rete ha dimostrato un notevole successo, raggiungendo una loss pari a 0.07 e un'accuracy del 97%. Questi valori risultano estremamente soddisfacenti e indicano che la rete ha appreso in maniera efficace le rappresentazioni complesse dei dati di addestramento. Per garantire che il nostro modello non abbia imparato a "memoria" le rappresentazioni dei dati, è essenziale esaminare i risultati sulla porzione di dati non vista durante l'addestramento, ovvero il set di validazione. Notiamo che la loss di validazione è leggermente più elevata rispetto a quella di addestramento, attestandosi a 0.28. Questo fenomeno è comune e può essere interpretato come un segno di una leggera capacità del modello di generalizzare su nuovi dati. L'accuracy di validazione, anch'essa con un valore del 90%, conferma questa tendenza. Nonostante la leggera differenza nei valori di loss e accuracy tra addestramento e validazione, il modello presenta comunque un ottimo livello di performance. L'overfitting, misurato come la differenza percentuale tra le due accuracies, è del 6.53%. Questo indica che il modello ha una buona capacità di generalizzazione, evitando un addestramento eccessivamente aderente ai dati di addestramento specifici.

Il secondo addestramento che andremo ad analizzare è quello che riguarda il modello BiLSTM.


```

Epoch 1/10
2890/2890 [=====] - 205s 69ms/step - loss: 0.3596 - accuracy: 0.8397 - val_loss: 0.2700 - val_accuracy: 0.8840 - lr: 0.0010
Epoch 2/10
2890/2890 [=====] - 200s 69ms/step - loss: 0.2190 - accuracy: 0.9132 - val_loss: 0.2502 - val_accuracy: 0.8952 - lr: 0.0010
Epoch 3/10
2890/2890 [=====] - 196s 68ms/step - loss: 0.1570 - accuracy: 0.9413 - val_loss: 0.2405 - val_accuracy: 0.9060 - lr: 0.0010
Epoch 4/10
2890/2890 [=====] - 199s 69ms/step - loss: 0.1183 - accuracy: 0.9567 - val_loss: 0.2689 - val_accuracy: 0.9068 - lr: 0.0010
Epoch 5/10
2890/2890 [=====] - 198s 68ms/step - loss: 0.0960 - accuracy: 0.9657 - val_loss: 0.2644 - val_accuracy: 0.9117 - lr: 0.0010
Epoch 6/10
2890/2890 [=====] - 214s 74ms/step - loss: 0.0678 - accuracy: 0.9746 - val_loss: 0.2894 - val_accuracy: 0.9109 - lr: 5.0000e-04
C:\Users\Genny\PycharmProjects\Sentiment_Analysis\venv\lib\site-packages\keras\src\engine\training.py:3000: UserWarning: You are saving your model as an HDF5 file via 'model.save'
saving_api.save_model(
Learning rate minimo raggiunto: 0.0005000000237487257
L'addestramento è stato interrotto all'epoca 6 a causa di early stopping.
La percentuale di overfitting è: 6.73%

```

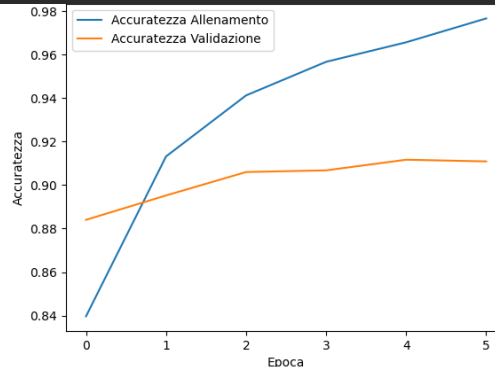


Figura 4.2: Dati addestramento BiLSTM

L'addestramento della rete RNN-BiLSTM ha prodotto risultati estremamente positivi. La loss è stata ridotta a un valore notevole di 0.06, accompagnata da un'accuracy del 97%. Questi risultati confermano la notevole abilità della rete nel comprendere e apprendere le complesse rappresentazioni dei dati di addestramento. La loss di validazione, pari a 0.28, riflette una leggera differenza rispetto alla loss di addestramento. Analogamente, l'accuracy di validazione, con un valore del 91%, evidenzia la capacità del modello di generalizzare su nuovi dati. Rispetto alla RNN-LSTM, la RNN-BiLSTM mostra miglioramenti significativi, suggerendo una maggiore robustezza nella gestione delle rappresentazioni dei dati. Un confronto diretto con la RNN-LSTM rivela che la RNN-BiLSTM ha riportato valori migliori nelle metriche di validazione, nonostante registri tempi di addestramento leggermente maggiori. L'overfitting, misurato al 6.73%, è leggermente superiore ma ancora accettabile. Questo indica che, nonostante la tendenza all'overfitting leggermente maggiore, la RNN-BiLSTM offre prestazioni superiori su nuovi dati rispetto alla sua controparte. In conclusione, la Figura 4.2 sottolinea il successo dell'addestramento della RNN-BiLSTM, evidenziando miglioramenti rispetto alla RNN-LSTM in termini di metriche di validazione. L'overfitting leggermente più elevato è bilanciato dalle prestazioni superiori su dati non osservati durante l'addestramento, posizionando la RNN-BiLSTM

come una promettente architettura per il nostro compito di analisi dei sentimenti. Il terzo addestramento che andremo ad analizzare è quello che riguarda il modello GRU.

```

Epoch 1/10
2890/2890 [=====] - 126s 43ms/step - loss: 0.3792 - accuracy: 0.8298 - val_loss: 0.2678 - val_accuracy: 0.8881 - lr: 0.0010
Epoch 2/10
2890/2890 [=====] - 130s 45ms/step - loss: 0.2348 - accuracy: 0.9051 - val_loss: 0.2444 - val_accuracy: 0.8998 - lr: 0.0010
Epoch 3/10
2890/2890 [=====] - 138s 48ms/step - loss: 0.1788 - accuracy: 0.9387 - val_loss: 0.2437 - val_accuracy: 0.9084 - lr: 0.0010
Epoch 4/10
2890/2890 [=====] - 134s 46ms/step - loss: 0.1425 - accuracy: 0.9461 - val_loss: 0.2573 - val_accuracy: 0.8993 - lr: 0.0010
Epoch 5/10
2890/2890 [=====] - 128s 44ms/step - loss: 0.1175 - accuracy: 0.9568 - val_loss: 0.3045 - val_accuracy: 0.8981 - lr: 0.0010
Epoch 6/10
2890/2890 [=====] - 129s 45ms/step - loss: 0.0889 - accuracy: 0.9689 - val_loss: 0.2971 - val_accuracy: 0.9013 - lr: 5.0000e-04
C:\Users\Genny\PycharmProjects\Sentiment_Analysis\venv\lib\site-packages\keras\src\engine\training.py:3080: UserWarning: You are saving your model as an HDF5 file via 'model.save'
  saving_api.save_model(
Learning rate minimo raggiunto: 0.0005000000237487257
L'addestramento è stato interrotto all'epoca 6 a causa di early stopping.
La percentuale di overfitting è: 6.98%

```

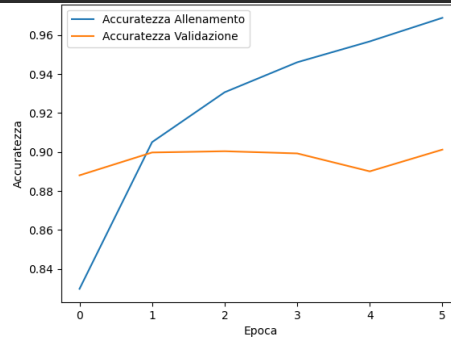


Figura 4.3: Dati addestramento GRU

I risultati dell'addestramento della rete RNN-GRU, evidenziati da una loss di 0.08 e un'accuracy del 96% sebbene soddisfacenti, richiedono un'ulteriore analisi per garantire che la rete abbia appreso in modo significativo senza cadere nell'eccessiva memorizzazione dei dati di addestramento. La loss di validazione si attesta a 0.29, mostrando una leggera discrepanza rispetto alla loss di addestramento. L'accuracy di validazione, pari al 90%, evidenzia la capacità del modello di generalizzare su nuovi dati, anche se con un rendimento inferiore rispetto alle architetture precedenti. Comparando direttamente con le architetture RNN-LSTM e RNN-BiLSTM, la RNN-GRU mostra valori leggermente inferiori nelle metriche di validazione. Questo suggerisce che, nonostante la sua capacità di apprendimento, la RNN-GRU potrebbe non essere altrettanto robusta nel generalizzare su dati non osservati durante l'addestramento. L'overfitting, misurato al 6.98%, indica una maggiore propensione del modello a memorizzare i dati di addestramento rispetto alle architetture precedenti. Questo è un aspetto importante da considerare, poiché un overfitting eccessivo potrebbe compromettere la capacità di adattamento del modello su nuovi dati.

4.2 Risultati dei test di validazione isolati sui modelli

Abbiamo riservato il 20% del dataset per un test di validazione isolato. Questa fase di valutazione avviene successivamente all'addestramento e fornisce una visione delle capacità di generalizzazione del modello. L'obiettivo è assicurare che il modello sia in grado di affrontare dati mai visti prima, evitando potenziali bias legati al set di addestramento e valutando la sua capacità di generalizzare dopo l'apprendimento. Questo monitoraggio continuo delle prestazioni ci consente di valutare l'andamento del modello nel tempo. Non solo aiuta a identificare segnali di overfitting, ma fornisce anche un'opportunità dinamica di regolare gli iperparametri per ottimizzare le prestazioni del modello. In questo modo, il test di validazione isolato post-addestramento funge da verifica finale delle capacità di generalizzazione del modello, mentre i test di validazione in corso d'opera permettono di guidare e perfezionare il processo di addestramento. Questa combinazione di approcci di valutazione offre una panoramica completa delle prestazioni dei modelli di Sentiment Analysis in diverse fasi del loro sviluppo, garantendo solidità e adattabilità.

1) Evaluate LSTM

```
Evaluate su dati che il modello non ha mai visto...
904/904 [=====] - 4s 4ms/step - loss: 0.2344 - accuracy: 0.9074
Loss: 0.2343820482492447, Accuracy: 0.9074048399925232
```

2) Evaluate BiLSTM

```
Evaluate su dati che il modello non ha mai visto...
904/904 [=====] - 6s 7ms/step - loss: 0.2344 - accuracy: 0.9076
Loss: 0.23439015448093414, Accuracy: 0.9076470732688904
```

3) Evaluate GRU

```
Evaluate su dati che il modello non ha mai visto...
904/904 [=====] - 4s 4ms/step - loss: 0.2399 - accuracy: 0.8995
Loss: 0.23991702497005463, Accuracy: 0.8994809985160828
```

Figura 4.4: Validazione Reti su dati che non ha mai visto

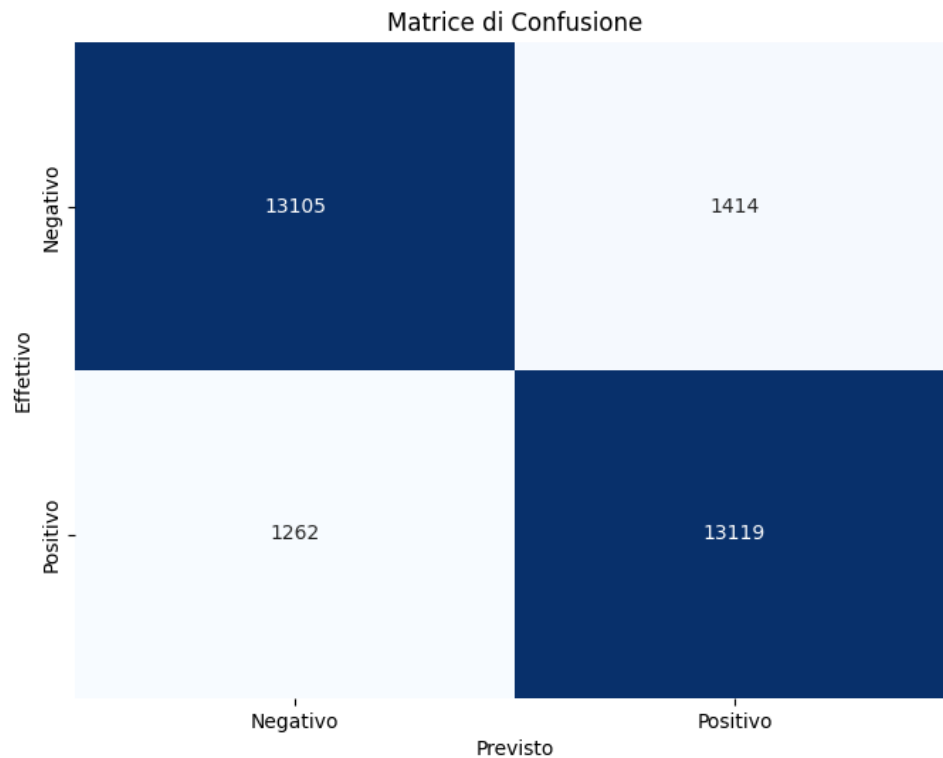
Modello	Loss	Accuracy (%)
LSTM	0.2344	90.74
BiLSTM	0.2344	90.76
GRU	0.2399	89.95

Tabella 4.1: Risultati delle valutazioni delle reti LSTM, BiLSTM e GRU

Per la rete LSTM, la loss pari a 0.2344 indica quanto bene il modello si adatti ai dati di valutazione, dove un valore più basso è preferibile. L'accuracy del 90.74% riflette la percentuale di previsioni corrette rispetto al totale dei casi. Questi risultati indicano un buon equilibrio tra precisione e generalizzazione della rete sui dati sconosciuti. La rete BiLSTM presenta risultati molto simili alla LSTM, con una loss di 0.2344 e un'accuracy di 90.76%. Questa architettura ha dimostrato di mantenere un livello elevato di precisione e adattamento ai dati di valutazione. La rete GRU mostra risultati leggermente inferiori rispetto alle reti LSTM e BiLSTM. La loss di 0.2399 e un'accuracy del 89.95% indicano comunque buone performance, ma potrebbero essere associate a una leggera diminuzione della precisione rispetto alle altre architetture.

4.3 Matrici di confusione

Di seguito vengono riportate le matrici di confusione delle RNN implementate. La prima matrice che andremo ad analizzare è quella relativa al modello LSTM:

**Figura 4.5:** Matrice di Confusione LSTM

True Negative (TN)	False Positive (FP)	False Negative (FN)	True Positive (TP)
13,105	1,414	1,262	13,119

Tabella 4.2: Valori Matrice di confusione LSTM

Con questi valori è possibile calcolare altre metriche di valutazione, come la *Precision* che misura la proporzione di istanze positive predette correttamente rispetto a tutte le istanze positive predette dal modello, la formula è:

$$\text{Precision} = \frac{TP}{TP+FP}$$

nel nostro caso la *Precision* assume un valore di 0.90. Oltre a questa metrica è possibile calcolare anche la *Recall* e la *F1-Score*. La *Recall* definita come il rapporto tra i veri positivi (TP) e la somma di veri positivi (TP) e falsi negativi (FN). La formula è:

$$\text{Recall} = \frac{TP}{TP+FN}$$

mentre la F1-Score la possiamo vedere come una media ponderata di precision e recall. La formula è:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

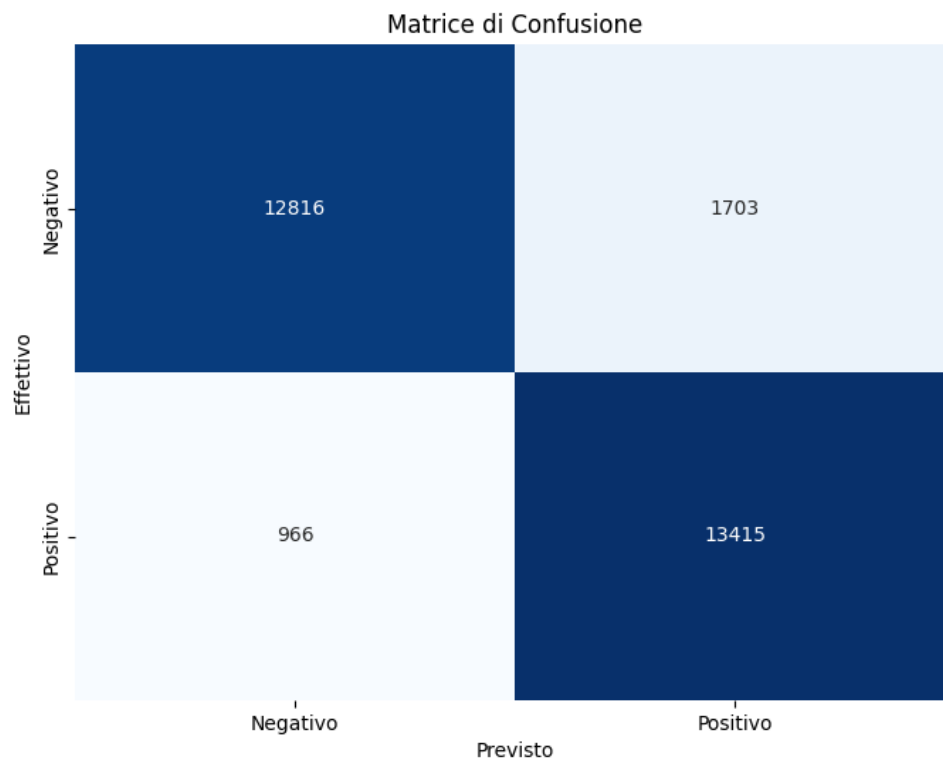
Il valore della Recall è pari a 0.91, mentre quello della F1-Score è pari a 0.90. Questi valori ci dicono che il nostro modello LSTM ha prestazioni molto solide.

- Precision = 0.90: significa che il 90% delle istanze classificate come positive dal modello sono effettivamente positive. Un valore così alto indica che il modello ha una bassa probabilità di classificare erroneamente le istanze negative come positive.
- Recall = 0.91: indica che il 91% delle istanze positive totali è stata correttamente identificata dal modello. Un valore alto indica che il modello riesce a catturare la maggior parte delle istanze positive presenti nei dati.
- F1-Score = 0.90: indica un buon equilibrio tra precisione e capacità del modello di catturare correttamente le istanze positive.

Precision	Recall	F1-Score
0.90	0.91	0.90

Tabella 4.3: Valori di Precision, Recall e F1-Score modello LSTM

La seconda matrice che andremo ad analizzare sarà quella relativa al modello BiLSTM:

**Figura 4.6:** Matrice di Confusione BiLSTM

True Negative (TN)	False Positive (FP)	False Negative (FN)	True Positive (TP)
12,816	1,703	966	13,415

Tabella 4.4: Valori Matrice di confusione BiLSTM

Con i valori ricavati dalla matrice di confusione abbiamo:

Precision	Recall	F1-Score
0.88	0.93	0.90

Tabella 4.5: Valori di Precision, Recall e F1-Score modello BiLSTM

La precisione è leggermente più bassa rispetto al modello precedente. Tuttavia, i valori sono ancora molto buoni e indicano che il modello ha una buona capacità di classificare correttamente gli esempi positivi, minimizzando i falsi positivi e i falsi negativi.

La terza matrice che andremo ad analizzare sarà quella relativa al modello GRU:

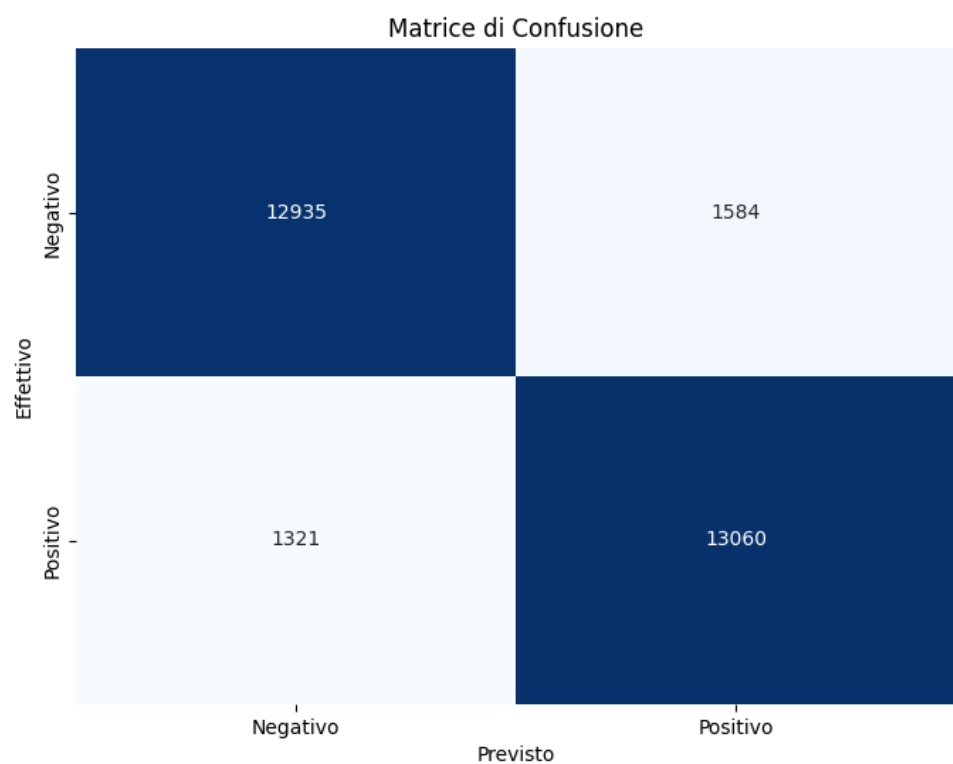


Figura 4.7: Matrice di Confusione GRU

True Negative (TN)	False Positive (FP)	False Negative (FN)	True Positive (TP)
12,935	1,584	1,321	13,060

Tabella 4.6: Valori Matrice di confusione GRU

Con i valori ricavati dalla matrice di confusione abbiamo:

Precision	Recall	F1-Score
0.89	0.90	0.89

Tabella 4.7: Valori di Precision, Recall e F1-Score modello GRU

Il modello mantiene una buona capacità di classificazione, con una precisione migliorata rispetto alla BiLSTM a scapito di una leggera diminuzione della recall. L’F1-Score, che bilancia precisione e recall, è rimasto stabile.

4.4 Explainable AI: LIME

Sono state date diverse istanze di esempi di tweet a LIME ma per la finalità di questa tesi, mostreremo due istanze, una positiva e una negativa.

- Positiva = **The product is really fantastic.**
- Negativa = **I did not like it at all, terrible purchase.**

La prima analisi Lime viene eseguita sul modello LSTM:

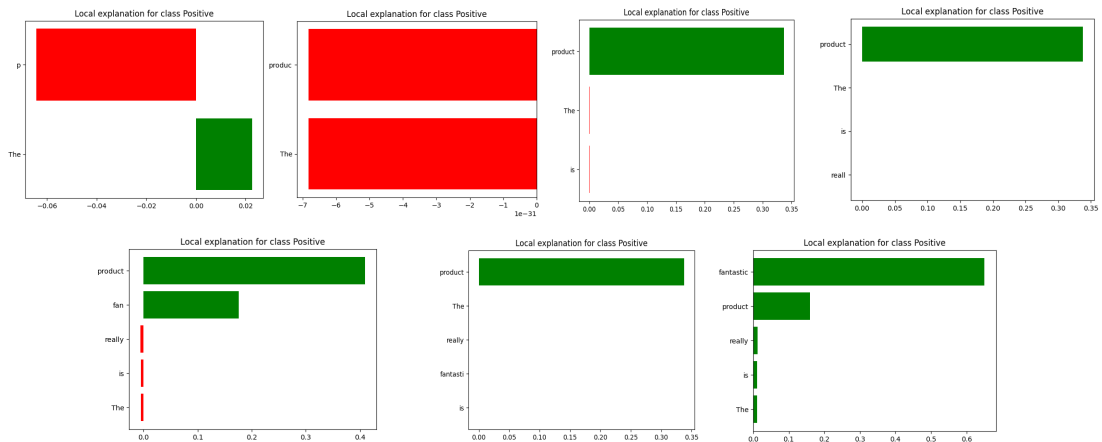


Figura 4.8: Analisi Lime-LSTM sul tweet positivo

Test_instance positiva (con lunghezza variabile):

- "The p": L'etichetta reale è "Positive", il che significa che il tweet dovrebbe essere classificato come positivo. Tuttavia, la previsione del modello è "Negative", il che indica che il modello ha classificato erroneamente questo breve testo come negativo. La spiegazione Lime mostra i contributi di alcune parole al verificarsi di questa previsione. Ad esempio, la parola "p" contribuisce negativamente alla previsione, mentre la parola "The" contribuisce positivamente.
- "The produc": la previsione del modello è ancora "Negative". La spiegazione Lime mostra che la parola "produc" ha un contributo molto piccolo alla previsione, mentre la parola "The" contribuisce negativamente.
- "The product is": la previsione del modello è ancora "Negative". La spiegazione Lime indica che la parola "product" ha un contributo positivo considerevole alla previsione, ma altre parole contribuiscono negativamente.
- "The product is really fan": In questo caso la previsione del modello diventa positiva. Le parole che hanno avuto un peso positivo sono state: "product", "fan", "really".

- "The product is really fantasti": la previsione torna negativa anche se non si nota dalla Figura 4.8
- "The product is really fantastic!": porta la previsione a essere "Positive", che è la corretta etichetta. La spiegazione Lime mostra il contributo positivo di parole come "product", "fantastic", "really" alla previsione corretta.

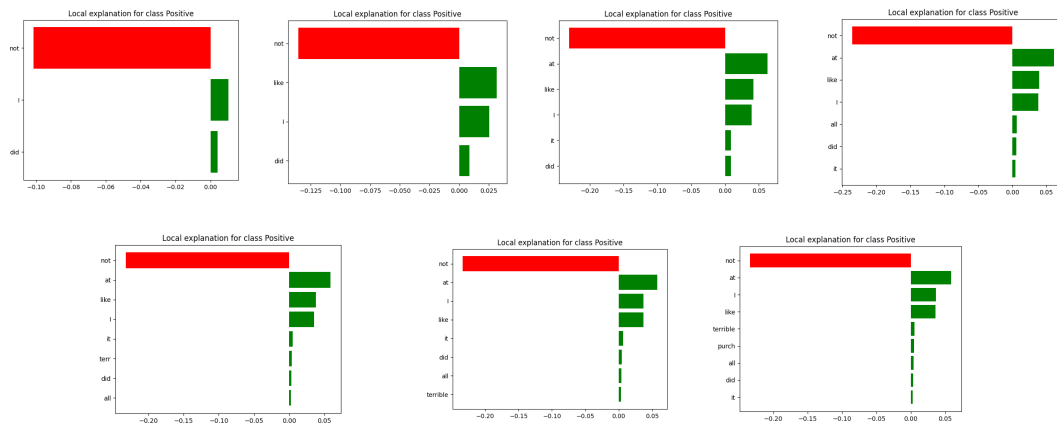


Figura 4.9: Analisi Lime-LSTM sul tweet negativo

Test_instance negativa (con lunghezza variabile):

Da questa analisi possiamo vedere che le parole più rilevanti sono: "not", "like", "terrible", e "purchase".

- Parola "not": Contribuisce in modo significativo a una previsione negativa, con un peso negativo considerevole, indicando che la presenza di questa parola è associata a una probabilità più bassa di una previsione positiva.
- Parola "like": Ha un peso positivo, ma minore rispetto a "not", indicando che la presenza di questa parola contribuisce leggermente a una previsione positiva.
- Parola "terrible": Ha un peso positivo significativo, contribuendo in modo rilevante a una previsione positiva.
- Parola "purchase": Ha un peso positivo, ma minore rispetto a "terrible" e contribuisce a una previsione positiva.

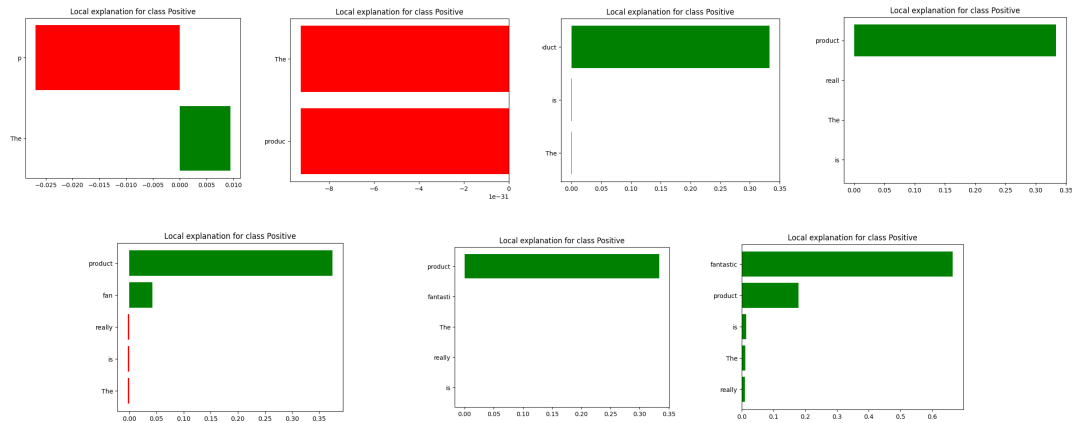


Figura 4.11: Analisi Lime-BiLSTM sul tweet positivo

La seconda analisi Lime viene eseguita sul modello BiLSTM: Test_instance positiva (con lunghezza variabile):

Parole come "fantastic" e "product" sono identificate come chiavi per una previsione corretta del sentimento positivo. La presenza di tali parole ha un impatto positivo sulle previsioni del modello, suggerendo che il modello associa correttamente queste parole a un sentimento positivo. Questa constatazione è consistente con l'aspettativa intuitiva che parole positive siano indicative di un tono positivo nella frase. D'altra parte, parole più generiche come "The" e frammenti di parole senza un significato apparente hanno un impatto negativo sulle previsioni del modello. La presenza di queste parole sembra confondere il modello e portare a errori nel riconoscimento del sentimento. Ciò suggerisce che il modello potrebbe non essere sufficientemente robusto nel discernere il significato corretto delle frasi, specialmente quando influenzato da parole comuni che non contribuiscono significativamente al contenuto emotivo, ma considerando un tweet completo in cui non ci sono frammenti di parole, il modello riesce ad identificare le parole chiave e quindi ad arrivare ad una corretta previsione.

Test_instance negativa (con lunghezza variabile):

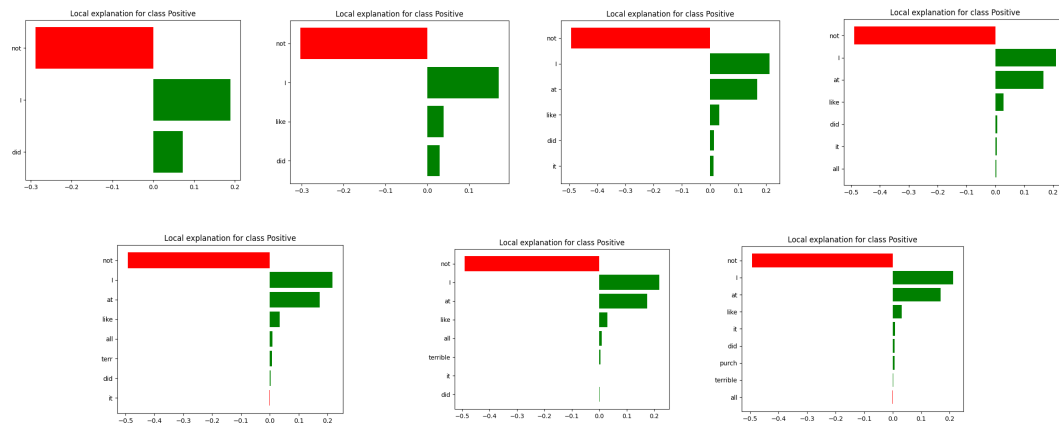


Figura 4.12: Analisi Lime-BiLSTM sul tweet negativo

Parole come "not" e "terrible," emergono come determinanti per una corretta previsione del sentimento negativo all'interno del modello di analisi del sentimento. La presenza di tali termini ha un impatto negativo significativo sulle previsioni del modello, indicando che il sistema associa accuratamente queste parole a un tono negativo all'interno della frase. Questo risultato è in linea con l'aspettativa intuitiva che parole negative siano indicative di un'atmosfera negativa nel contesto. Al contrario, parole più generiche come "I" e "did," oltre a termini comuni come "like" e "it," contribuiscono positivamente alle previsioni del modello, ma in misura minore rispetto alle parole chiave negative. La presenza di queste parole generiche sembra essere associata a un contesto neutro o positivo.

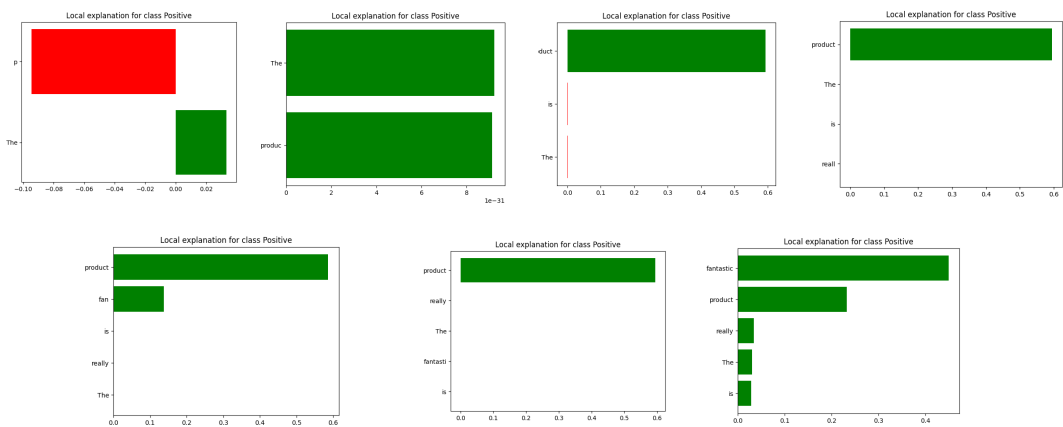


Figura 4.14: Analisi Lime-GRU sul tweet positivo

Test_instance negatvia (con lunghezza variabile):

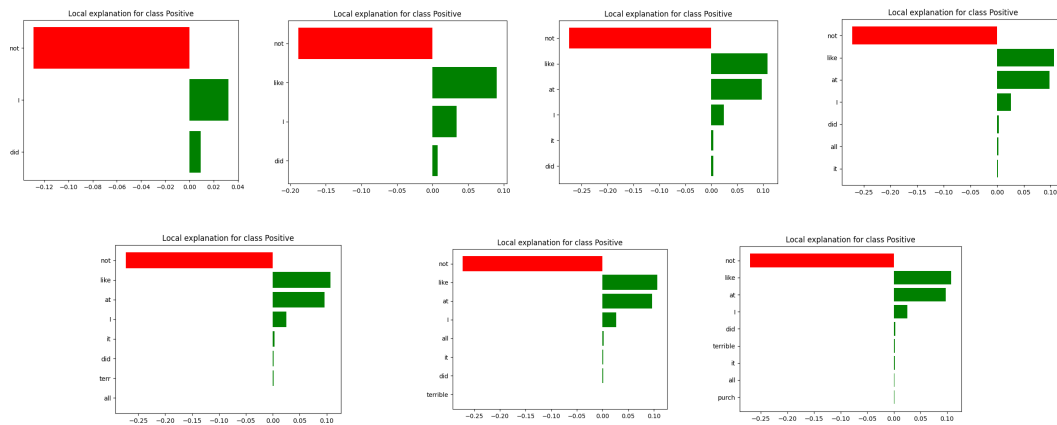


Figura 4.15: Analisi Lime-GRU sul tweet negativo

La presenza di parole come "not," "like," "at," e "terrible" ha un impatto significativo sulla previsione del modello, contribuendo a rafforzare l'etichetta negativa. La parola "not" è fortemente associata a un impatto negativo, come indicato dal suo peso negativo nell'analisi Lime. Allo stesso modo, le parole "like" e "at" hanno contributi positivi alla previsione del sentimento negativo. La presenza di parole specifiche, come "terrible," conferma ulteriormente la natura negativa della frase, evidenziando come il modello riconosca correttamente la forza emotiva di parole più intense. D'altra parte, la presenza di parole comuni come "I" e "did" sembra avere un impatto minore sulla previsione del modello, indicando che il modello si focalizza maggiormente su termini più descrittivi e emotivamente carichi.

CAPITOLO 5

Conclusioni

Questo capitolo riassume il lavoro svolto ed analizza i risultati ottenuti

Nel corso di questo progetto di tesi, abbiamo esplorato e addestrato tre diverse architetture di reti neurali ricorrenti (RNN) per l'analisi dei sentimenti su un dataset di tweet. Le architetture considerate sono LSTM (Long Short-Term Memory), BiLSTM (Bidirectional LSTM) e GRU (Gated Recurrent Unit). Di seguito, forniamo un'analisi dettagliata delle prestazioni di ciascun modello:

Modello	Precision	Recall	F1-Score
LSTM	0.90	0.91	0.90
BiLSTM	0.88	0.93	0.90
GRU	0.89	0.90	0.89

Tabella 5.1: Confronto tra LSTM, BiLSTM e GRU su Precision, Recall e F1-Score

Il modello LSTM ha dimostrato prestazioni solide durante l'addestramento e la validazione, raggiungendo un'accuracy del 90.74% sui dati di test. Questo evidenzia la sua abilità nell'apprendere rappresentazioni semantiche complesse dai tweet.

Tuttavia, va sottolineato che l'overfitting del 6.53% indica una leggera differenza tra le prestazioni sui dati di addestramento e quelli di validazione.

La BiLSTM, sfruttando la sua capacità di elaborare informazioni sia da sinistra a destra che da destra a sinistra, ha dimostrato prestazioni superiori rispetto alla LSTM. Con un'accuracy del 90.76% sui dati di test e un overfitting del 6.73%, la BiLSTM si distingue per la sua capacità di catturare pattern complessi nei tweet. Nonostante un overfitting leggermente superiore rispetto alla LSTM, la BiLSTM si è confermata come la migliore in termini di recall.

Il modello GRU ha conseguito risultati accettabili, registrando un'accuracy del 89.95% sui dati di test. Tuttavia, l'overfitting del 6.98% suggerisce che potrebbero essere necessarie ulteriori regolazioni dei parametri o un aumento del numero di dati di training per ottimizzare le prestazioni. La GRU si posiziona in una posizione intermedia tra LSTM e BiLSTM in termini di rendimento complessivo.

Nel confronto tra i modelli, emerge che la BiLSTM ha prestazioni superiori, specialmente in termini di recall, mentre la LSTM si distingue per la migliore precisione. La scelta tra questi modelli dovrebbe essere guidata dalle esigenze specifiche dell'applicazione. Se l'obiettivo principale è massimizzare la rilevazione degli eventi positivi, la BiLSTM potrebbe risultare preferibile. D'altra parte, se è cruciale evitare falsi positivi, la LSTM potrebbe essere più indicata. Questi risultati suggeriscono che la scelta del modello ideale deve tener conto degli obiettivi specifici del problema, potendo richiedere in certi contesti un compromesso bilanciato tra precisione e recall.

Bibliografia

- [1] A. Yadav and D. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artificial Intelligence Review*, vol. 53, pp. 4335–4385, 2020. [Online]. Available: <https://doi.org/10.1007/s10462-019-09794-5> (Citato a pagina 4)
- [2] D. Tang, B. Qin, and T. Liu, "Deep learning for sentiment analysis: successful approaches and future challenges," *WIREs Data Mining and Knowledge Discovery*, vol. 5, pp. 292–303, 2015. (Citato alle pagine 4 e 9)
- [3] V. Tyagi, A. Kumar, and S. Das, "Sentiment analysis on twitter data using deep learning approach," in *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2020, pp. 187–190. (Citato a pagina 5)
- [4] J. Xiao and Z. Zhou, "Research progress of rnn language model," in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2020, pp. 1285–1288. (Citato a pagina 5)
- [5] R. Patel and S. Patel, "Deep learning for natural language processing," in *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*, M. S. Kaiser, J. Xie, and V. S. Rathore, Eds. Singapore: Springer Nature Singapore, 2021, pp. 523–533. (Citato a pagina 5)

-
- [6] R. Monika, S. Deivalakshmi, and B. Janet, "Sentiment analysis of us airlines tweets using lstm/rnn," in *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, 2019, pp. 92–95. (Citato a pagina 6)
- [7] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and lstm," *Neural Computing & Applications*, vol. 32, pp. 9713–9729, 2020. (Citato a pagina 6)
- [8] U. Naseem and K. Musial, "Dice: Deep intelligent contextual embedding for twitter sentiment analysis," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 953–958. (Citato a pagina 6)
- [9] R. Ni and H. Cao, "Sentiment analysis based on glove and lstm-gru," in *2020 39th Chinese Control Conference (CCC)*, 2020, pp. 7492–7497. (Citato a pagina 7)
- [10] M. M. Abdelgwad, T. H. A. Soliman, A. I. Taloba, and M. F. Farghaly, "Arabic aspect based sentiment analysis using bidirectional gru based models," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6652–6662, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1319157821002482> (Citato a pagina 7)
- [11] Y. M. Wazery, H. S. Mohammed, and E. H. Houssein, "Twitter sentiment analysis using deep neural network," in *2018 14th International Computer Engineering Conference (ICENCO)*, 2018, pp. 177–182. (Citato a pagina 7)
- [12] A. Topbaş, A. Jamil, A. A. Hameed, S. M. Ali, S. Bazai, and S. A. Shah, "Sentiment analysis for covid-19 tweets using recurrent neural network (rnn) and bidirectional encoder representations (bert) models," in *2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, 2021, pp. 1–6. (Citato a pagina 7)

Ringraziamenti

A mio fratello:

Pilastro fondamentale del mio successo, grazie per avermi supportato e "sopportato" in quei periodi in cui credevo di non riuscire a farcela, in cui credevo di non essere abbastanza. Grazie per essere sempre stato al mio fianco, per la tua comprensione incondizionata e per la forza che trasmetti nei momenti difficili. La tua presenza ha reso la mia vita più ricca e significativa, e per questo ti sono infinitamente grato. So che la vita è un cammino imprevedibile e spesso complicato, ma con te al mio fianco, ho imparato che la bellezza della vita risiede proprio nella sua imprevedibilità. Le sfide che affrontiamo insieme ci rendono più forti, e le gioie che condividiamo sono ancor più luminose a causa degli ostacoli superati. Spero dal profondo del cuore che questo traguardo, tanto mio quanto tuo, ti possa far capire che la vita non è già scritta, ma è come la creiamo giorno dopo giorno con le nostre scelte, le nostre azioni e la nostra capacità di adattarci alle sfide. Grazie per essere la mia fonte di ispirazione, il mio compagno di avventure e il mio migliore amico. La tua presenza nella mia vita è un dono inestimabile, e non vedo l'ora di condividere ancora molti momenti speciali insieme.

A mia madre:

In questo momento speciale, voglio dedicarti un ringraziamento sincero, perché sei stata la prima a intraprendere con me questo straordinario viaggio chiamato vita. Sei stata la prima ad avermi guardato con occhi pieni di amore e sincera comprensione.

Sei stata la prima a tenermi tra le braccia, il luogo più sicuro e confortante che abbia mai conosciuto. Hai pianto di felicità nei momenti speciali, condividendo emozioni uniche che rimarranno impresse nel mio cuore per sempre. Grazie perchè mi hai insegnato i valori fondamentali della vita. Hai dedicato il tuo tempo, la tua energia e il tuo amore infinito per farmi diventare la persona che sono oggi. Hai affrontato ogni sfida con coraggio, senza farmi mai mancare nulla. Questo traguardo è anche tuo, poiché sei stata la radice della mia crescita e del mio successo perchè sei stata la prima a credere in me, ancor prima che io iniziassi a credere in me stesso. Grazie per essere stata la mia più grande sostenitrice, la mia motivatrice, e per avermi insegnato che non ci sono limiti a ciò che possiamo realizzare.

A mio padre:

Grazie papà per aver rappresentato la forza silenziosa dietro ogni mia conquista, la voce incoraggiante quando ho affrontato le sfide e il sorriso fiero nei momenti di successo. Il tuo amore e il tuo impegno sono il fondamento su cui poggia la mia vita. Grazie papà per aver condiviso con me le tue esperienze, i tuoi fallimenti e i tuoi successi, insegnandomi che ogni momento della vita è un'opportunità di crescita. Grazie per essere stato il mio mentore, il mio consigliere e il mio amico più fidato. Grazie per avermi insegnato che la vera forza risiede nella capacità di affrontare le difficoltà con dignità e speranza.

A mia nonna:

Le parole sembrano troppo piccole per esprimere la gratitudine che provo nei tuoi confronti. Grazie per essere stata la fonte inesauribile di amore e affetto. I tuoi abbracci sono stati il rifugio sicuro nei momenti difficili. La tua saggezza è stata la bussola che ha orientato il mio percorso. Grazie per i tuoi consigli, le tue storie di vita che sono diventate le mie lezioni più preziose. Mi hai insegnato che la tenacia è la chiave per affrontare le tempeste, che la gentilezza è la lingua universale e che l'amore è la forza che supera ogni ostacolo. Grazie per essere stata la mia migliore amica. Le risate condivise, i segreti custoditi e i momenti di silenziosa comprensione hanno reso la nostra relazione un tesoro prezioso. Ogni momento passato con te è un ricordo che conservo gelosamente nel mio cuore. La tua presenza nella mia vita è il regalo più grande, e sono eternamente grato per ogni istante trascorso con te.

A zio Gino:

Sei un faro luminoso nella mia vita, una presenza che ha reso il mio cammino più ricco e significativo nonostante le circostanze che ci separano fisicamente, il tuo spirito e la tua forza d'animo continuano a ispirarmi. Grazie per avermi trasmesso la tua capacità di trovare significato anche nelle circostanze più difficili. La tua prospettiva positiva e la tua ricerca continua di speranza sono un faro di luce che illumina la via. Grazie per avermi insegnato che la speranza può fiorire anche nei luoghi più inaspettati. Il tuo incoraggiamento ha giocato un ruolo fondamentale nel mantenermi motivato e concentrato sui miei obiettivi accademici. Hai creduto in me quando le sfide sembravano insormontabili, e questo ha reso ogni passo del mio percorso educativo più significativo. Il tuo costante sostegno ha alimentato la mia determinazione a raggiungere il successo. Le tue parole di saggezza e il tuo incoraggiamento instancabile hanno illuminato il cammino, rendendo ogni conquista un motivo di gioia condivisa. Sono grato di averti come zio e mentore, e il tuo incoraggiamento è un dono che porterò sempre con me.

A Rosa:

L'amica di una vita intera. La tua presenza nella vita di mio fratello ha portato una luce radiante e una gioia che si riflette in ogni momento trascorso insieme. Grazie per la tua gentilezza e il tuo cuore generoso. Hai portato un tocco speciale alla nostra famiglia con la tua presenza amorevole e il tuo spirito positivo. Grazie per le infinite chiacchierate, grazie per i mille consigli e grazie per il tuo affetto incondizionato. Sei davvero una benedizione, e sono entusiasta di condividere il futuro con una persona così straordinaria come te.

Grazie al mio fantastico gruppo *The System Errors*, composto da Domenico, Luigi, Emanuele, Gianpaolo, Manuel e Pasquale, desidero ringraziarvi perché con voi ho capito che l'Università può andare oltre i semplici libri. Ricordo il periodo di IS come se fosse ieri, quanto ci siamo divertiti, ammettetelo (non ridete, che se mi siedo e ci ripenso, inizio a piangere). Ho imparato molto con voi, ma soprattutto ho imparato moltissimo da ognuno di voi.

Gio:

Non devo ringraziare solo i "senior" di questo fantastico gruppo, ma anche un altro

membro che fin dal primo giorno in cui ha messo piede qui, in questa gabbia di matti, ci ho stretto un incredibile rapporto: "Giò". Grazie per tutte le volte in cui sei stato un consigliere sincero, grazie per tutte quelle volte in cui ho trovato un amico che mi ascoltasse, e soprattutto grazie perché anche con la tua conoscenza ho capito veramente che l'università non è solo esami, esami ed esami.

Domenico:

Non posso non ringraziare una persona di questo fantastico gruppo: "Domenico". (Non iniziare a ridere, che se sto leggendo questi ringraziamenti non finisco più). Dom, sei stato il primo "collega" universitario molto simile a me; infatti, fin da subito ci siamo trovati bene insieme. Forse tu mi hai insegnato la lezione più cara che potessi mai imparare in questo percorso accademico, ovvero quella di non perdere mai "la determinazione" e "il coraggio". Pochi sanno del tuo percorso, e altrettanto pochi sanno le motivazioni che ti hanno spinto a fare determinate scelte, ma tu non hai mollato e hai inseguito l'obiettivo conquistandolo. Quindi, i ringraziamenti sono d'obbligo, perché ho capito che non bisogna mai arrendersi alle avversità e bisogna continuare a inseguire i propri sogni. Grazie "Domè".

Marco:

Grazie Marco (lo so non ti ho nominato nei The System Errors però guarda il lato positivo...). Anche con te sono cresciuto e grazie a te ho ricevuto una lezione molto importante (non parlo di quelle fatte in classe, lo so non c'era bisogno di specificare però si mi andava) ovvero quella di non fermarmi alle apparenze. Ricordo ancora la frase che dissi il primo giorno che ti vidi "come mi sta antipatico quello", cerchiamo di fare i formali, la frase già la conosci. Però non so come siamo diventati amici e questa insieme ad altre è stata un'enorme fortuna per me. Grazie per i mille consigli preziosi sullo studio e soprattutto grazie per essere stato il mio limitatore di "ansia", già sai shhhhh. Spero di fortificare questa splendida amicizia nel percorso comune che abbiamo scelto.

Peppe:

Grazie Peppe per i nostri momenti di allegria e spirito positivo, perché hanno reso indimenticabile questa esperienza accademica. Grazie, Peppe, per aver condiviso con me gioie e risate durante questo percorso significativo.

Le farmaciste e l'ingegnere:

Grazie a Flavia, Sara, Candida e Lucia, ragazze che con l'informatica c'entrano ben poco, ma amiche dalle mille risate, dalle mille ansie ehhh si dai mille caffè. Vi ho conosciuto per puro caso, ovviamente sempre fortunato nella vita no? Si scherza, si scherza... è stato questo scherzare con voi che ha reso il mio percorso più "leggero" e divertente. Un augurio che voglio fare ad altri studenti come me è proprio quello di trovare nel loro percorso amiche come voi.

CoScienze:

Grazie a CoScienze, la mia seconda casa, la mia seconda famiglia. Grazie a te (a voi tutti) ho rivalutato in meglio l'università, ho imparato a mettermi in gioco per gli altri. Voglio ringraziare tutto lo splendido direttivo (Rebecca, Leonardo Schiavo, Marisa, Mattia Matteo, Zero, Leonardo Monaco, Antonio L.M e Nick e tutte le new entry) per l'incredibile impegno messo in ogni singola attività firmata CoScienze. Da voi ho imparato a trasmettere quella sicurezza che a volte mancava a me, ho imparato a gioire per obiettivi comuni e soprattutto ho capito che con voi non si è mai soli. Sono infinitamente orgoglioso di dire "faccio parte del direttivo di CoScienze" (Schiavo lo sappiamo è quello allargato fai il bravo).

Leo M:

Parlando di questa splendida famiglia, non posso non ringraziare anche solo con poche righe la persona che è riuscita a farmi amare questo aspetto universitario, Leonardo M. Grazie Leo per essere stato lì in quei momenti in cui avevo bisogno di un aiuto e grazie per aver avuto sempre una parola giusta in qualsiasi situazione, forse è stato questo tuo essere disponibile con tutti che mi ha invogliato a mettermi in gioco per gli altri e non posso fare altro che ringraziarti per avermi fatto vivere a pieno questo mio percorso universitario.

Voglio ringraziare inoltre i miei amici i quali non c'entrano con il mondo universitario ma sono una parte fondamentale delle mie giornate.

Antonio:

Vorrei iniziare dalla persona che reputo più "coraggiosa" (tanto lo sappiamo cosa voglio dire) Antonio H. perchè con me, con noi non sei mai cambiato sei sempre stato la persona che ho conosciuto 7/8 anni fa. Ammiro tutti i sacrifici che fai e soprattutto

ammiro la tua tenacia nell'inseguire i sogni. Grazie per essere stato sempre lì anche in quei periodi in cui l'università mi soffocava ma tu, da amico leale, eri sempre pronto a supportarmi.

Paki:

Grazie Pasquale perchè anche con il tuo supporto sono riuscito a gestire i periodi più stressanti. Le infinite chiacchierate con te mi alleggerivano dal peso dei libri. Sei sempre stato un gran lavoratore(non ridere jaa che in fondo lavori un pò) e la cosa che ho sempre ammirato di te è la voglia di mettersi in gioco nonostante gli sbagli oppure la poca esperienza, non ti sei mai tirato indietro. Non cambiare mai il tuo splendido carattere.

Alessandro:

Ci siamo conosciuti forse anche un pò per caso, però è proprio questo il bello delle amicizie, quelle vere, non sai mai quando possono arrivare. Grazie per avermi sempre motivato e spronato e soprattutto grazie per l'enorme fiducia che hai avuto nei miei confronti nel raggiungere questo obiettivo che credevo tanto lontano. Grazie perchè sei sempre stato lì anche nei momenti più brutti. Grazie per essere sempre stato il fratello maggiore che non ho mai avuto.

Giusy:

In questi due anni e passa di amicizia, hai aggiunto colori brillanti alle mie giornate. Grazie per le risate contagiose, gli abbracci calorosi e le avventure strampalate che abbiamo vissuto insieme. Non vedo l'ora di creare ancora più ricordi felici e di condividere momenti spassosi con te nei prossimi anni.

A tutti i miei amici:

La vostra presenza nella mia vita è un tesoro inestimabile, e non posso fare a meno di ringraziarvi per ogni sorriso condiviso e per ogni momento di sostegno. Grazie per la vostra gentilezza, comprensione e affetto. Siete parte integrante della mia vita, e non posso che sentirmi fortunato per avere amici così straordinari.

Francesco S:

A Francesco S. desidero esprimere la mia sincera gratitudine per tutto quello che hai fatto per me. La tua guida e il tuo insegnamento hanno avuto un impatto straordinario sulla mia vita e sul mio approccio a questo mondo. Grazie a te, ho imparato ad amare questo ambiente in modo profondo e appassionato. La tua dedizione nel

condividere le tue conoscenze e la tua esperienza è stata ispiratrice, rendendo ogni lezione un'opportunità di crescita e scoperta. Grazie, Ciccio, per essere stato non solo un insegnante, ma anche un mentore e un amico.

La famiglia mia tutta:

Infine vorrei ringraziare tutti i miei cugini e zii (non vi elenco perchè siete veramente tanti). non posso fare a meno di esprimere la mia profonda gratitudine nei vostri confronti. La nostra famiglia è un tesoro, e voi, cari cugini e zii, siete parti essenziali di quel tesoro. Grazie per essere sempre presenti nelle diverse fasi della mia vita. Le risate condivise, le esperienze condivise e anche le sfide superate insieme hanno reso i nostri legami ancora più forti. La vostra presenza ha reso il cammino della vita più ricco e significativo. Grazie di cuore per essere la meravigliosa famiglia che siamo. La vostra presenza rende la vita più bella