FONDAMENTI DI INTELLIGENZA ARTIFICIALE

Project Documentation - TeachTuner

Sentiment Analysis for Education

Nocera Salvatore[1]* | **Mattia Fattoruso**[2]*

Students information

[1]Matricola: 0512117510

[2]Matricola: 0512118639

Correspondence

Dipartimento di Informatica, Università degli Studi di Salerno, Fisciano, SA, Italia

Contact details

s.nocera7@studenti.unisa.it

m.fattoruso11@studenti.unisa.it

Questo progetto applica tecniche di analisi del sentiment al contesto educativo per automatizzare l'interpretazione dei feedback degli studenti. Il progetto prevede la raccolta di commenti tramite form personalizzati, la normalizzazione dei dati testuali e l'utilizzo di modelli di machine learning e deep learning per classificare le opinioni in sentiment positivi, negativi o neutri. I risultati sono presentati attraverso report intuitivi e visualizzazioni grafiche, offrendo uno strumento pratico e scalabile per migliorare la qualità dell'insegnamento e supportare decisioni informate basate sui dati.

1 | INTRODUCTION

TeachTuner è un progetto che esplora come l'analisi del sentiment possa essere applicata al contesto educativo. L'obiettivo è fornire a educatori e istituzioni uno strumento innovativo per raccogliere e analizzare feedback degli studenti, migliorando la qualità dell'insegnamento e ottimizzando le strategie didattiche. Utilizzando tecnologie di machine learning e deep learning, il progetto automatizza l'elaborazione del linguaggio naturale per classificare opinioni e sentimenti, favorendo decisioni informate e basate sui dati.

La documentazione che segue descrive in dettaglio le scelte metodologiche, le tecnologie utilizzate e le sfide affrontate, evidenziando i vantaggi di un sistema progettato per l'analisi emotiva nel settore educativo.

Link Repository Tutti i dettagli, il codice sorgente e la documentazione tecnica completa del progetto sono disponibili nella nostra repository ufficiale: `https://github.com/Pr1vat30/machine_learning_project.git`

1.1 | What is sentiment analysis

La sentiment analysis, nota anche come opinion mining, è un ramo dell'elaborazione del linguaggio naturale (NLP, Natural Language Processing) e della linguistica computazionale che si concentra sull'identificazione, l'estrazione e la quantificazione del tono emotivo e delle informazioni soggettive contenute nei dati testuali. Questo campo riveste

* Equally contributing authors.

un ruolo cruciale nella comprensione degli atteggiamenti, delle emozioni e delle opinioni espresse dagli individui nella comunicazione scritta o parlata. Analizzando il linguaggio, la sentiment analysis mira a classificare il testo come positivo, negativo o neutro e, in applicazioni più avanzate, a identificare emozioni specifiche come gioia, rabbia o tristezza.

Alla base della sentiment analysis vi è l'esigenza di comprendere il comportamento umano, le preferenze e gli atteggiamenti in un mondo sempre più dominato dalla comunicazione digitale. In un'epoca in cui i social media, le recensioni online e i feedback dei clienti sono centrali nelle interazioni, organizzazioni e ricercatori devono elaborare enormi volumi di dati testuali per trarre informazioni significative. La sentiment analysis colma questa lacuna automatizzando l'interpretazione del testo e offrendo un modo sistematico per monitorare l'opinione pubblica, valutare la soddisfazione dei clienti e analizzare le tendenze emotive nel tempo.

L'importanza della sentiment analysis trascende settori e discipline. Nel mondo del business, consente alle aziende di ottimizzare le strategie di marketing, migliorare le relazioni con i clienti e monitorare la percezione del brand. In politica, la sentiment analysis viene utilizzata per misurare l'opinione pubblica su politiche, campagne o candidati. Oltre alle sue applicazioni pratiche, lo studio della sentiment analysis contribuisce anche ai progressi nella linguistica computazionale e nell'intelligenza artificiale, ampliando le capacità delle macchine di comprendere il linguaggio umano.

I fondamenti teorici della sentiment analysis si basano sull'assunto che il linguaggio trasmetta non solo informazioni, ma anche segnali emotivi. Analizzando le proprietà semantiche e sintattiche del testo, gli algoritmi possono dedurre i sentimenti sottostanti. Ad esempio, la presenza di aggettivi come "eccellente" o "terribile", l'uso dei punti esclamativi o persino la disposizione sintattica delle frasi possono indicare un sentimento positivo o negativo. Inoltre, la sentiment analysis tiene conto della natura contestuale del linguaggio, riconoscendo che il significato e il tono delle parole spesso variano in base al contesto circostante.

- *"This phone is amazing! The battery lasts all day!"* verrebbe probabilmente classificata come positiva.
- *"The product arrived late, and it doesn't work as expected"* sarebbe registrata come negativa.
- *"The package was delivered on Thursday"* potrebbe essere classificata come neutrale.

1.2 | How sentiment analysis work

Il funzionamento della sentiment analysis si basa su una combinazione di principi linguistici, modelli statistici e tecniche computazionali avanzate. Alla sua base, la sentiment analysis implica l'elaborazione e l'analisi dei dati testuali per classificarli in base al loro contenuto emotivo. Questo processo può essere suddiviso in diverse fasi fondamentali, ognuna delle quali svolge un ruolo cruciale nel trasformare il testo grezzo in intuizioni significative sul sentiment.

La prima fase della sentiment analysis è il data preprocessing, che consiste nella pulizia e preparazione dei dati testuali per l'analisi. I dati testuali raccolti da diverse fonti, come post sui social media, recensioni o email, sono spesso non strutturati e contengono rumore sotto forma di errori di battitura, gergo, emoji e informazioni irrilevanti. Il preprocessing generalmente include passaggi come la tokenization, che suddivide il testo in singole parole o frasi; la rimozione di stopwords, ovvero parole comuni come "e" o "il" che non hanno un significato rilevante; e la standardizzazione del testo tramite la conversione in minuscolo o la riduzione delle parole alle loro radici (stemming). Questi passaggi di preprocessing garantiscono che i dati siano in un formato adatto per l'analisi.

Una volta preprocessato il testo, la fase successiva consiste nell'feature extraction, in cui il testo viene trasformato in una rappresentazione numerica che può essere elaborata da algoritmi di machine learning. Questo può essere realizzato attraverso tecniche come il bag-of-words (BoW), il term frequency-inverse document frequency (TF-IDF) o le word embeddings. BoW e TF-IDF rappresentano il testo in base alla frequenza delle parole, mentre le word embeddings, come Word2Vec o GloVe, codificano le parole in vettori densi che catturano le relazioni semantiche.

Il cuore della sentiment analysis risiede nell'applicazione di algoritmi per classificare il testo in base al sentiment. Questi algoritmi possono variare da sistemi basati su regole a modelli di machine learning e, più recentemente, architetture di deep learning. I sistemi basati su regole si affidano a dizionari predefiniti di parole connotate dal punto di vista emotivo e a regole sintattiche per determinare la polarità del sentiment. Ad esempio, un approccio basato su regole potrebbe classificare una frase come positiva se contiene parole come "felice" o "straordinario." Sebbene semplice, questo metodo spesso fatica a gestire il linguaggio dipendente dal contesto e le espressioni complesse.

I modelli di machine learning, invece, utilizzano dataset etichettati per addestrare classificatori in grado di prevedere il sentiment. Algoritmi come la logistic regression, le support vector machines (SVMs) e il naive Bayes sono stati storicamente scelte popolari per la sentiment analysis. Questi modelli apprendono pattern dato dati riuscendo ad associare le caratteristiche del testo alle etichette di sentiment. Tuttavia, richiedono quantità significative di dati etichettati e spesso non riescono a cogliere le sfumature del linguaggio.

L'avvento del deep learning ha migliorato significativamente la precisione e la flessibilità della sentiment analysis. Le reti neurali, in particolare le recurrent neural networks (RNNs), le convolutional neural networks (CNNs) e le architetture basate su transformer, hanno dimostrato prestazioni eccezionali nella classificazione del sentiment.

La sentiment analysis può essere condotta a diversi livelli di granularità, tra cui:

- Document-level analysis, che valuta il sentiment complessivo di un testo, come un'intera recensione o articolo.
- Sentence-level analysis, che si concentra sul sentiment di singole frasi.
- Aspect-level analysis, che analizza il sentiment relativo a specifici attributi o caratteristiche all'interno del testo.

Nonostante i progressi effettuati, l'efficacia della sentiment analysis dipende dalla capacità di affrontare diverse sfide. L'ambiguità del linguaggio, la presenza di sarcasmo e ironia e la natura dinamica dei significati delle parole rappresentano ostacoli significativi. Inoltre, la necessità di elaborare dati multilingue e specifici per determinati domini richiede tecniche sofisticate e dataset di addestramento diversificati.

1.3 | Sentiment analysis in education

L'applicazione della sentiment analysis nell'ambito dell'educazione si è rivelata una strada promettente per migliorare i risultati di apprendimento, favorire il coinvolgimento degli studenti e arricchire l'esperienza educativa complessiva. Analizzando i dati testuali generati nei contesti educativi — come i feedback degli studenti, i forum di discussione e i contenuti accademici — la sentiment analysis consente a educatori e istituzioni di ottenere una comprensione più profonda delle emozioni, degli atteggiamenti e delle necessità degli studenti.

Uno degli utilizzi principali della sentiment analysis in ambito educativo è la valutazione dei feedback degli studenti. Le istituzioni educative raccolgono regolarmente feedback attraverso sondaggi, valutazioni dei corsi e piattaforme online per misurare la qualità dell'insegnamento, dei contenuti dei corsi e il livello complessivo di soddisfazione degli studenti. Analizzare manualmente questi feedback può essere un processo lungo e soggetto a bias. La sentiment analysis automatizza tale processo categorizzando i commenti come positivi, negativi o neutri, consentendo agli educatori di individuare i punti di forza e le aree che necessitano di miglioramento.

La sentiment analysis riveste inoltre un ruolo fondamentale nel monitoraggio delle emozioni e del benessere mentale degli studenti. Analizzando testi provenienti da forum di discussione, social media o persino compiti scritti dagli studenti, è possibile rilevare segnali di stress, frustrazione o disimpegno. Ad esempio, studenti che esprimono sentimenti negativi riguardo al carico di lavoro o a un argomento specifico potrebbero beneficiare di un supporto o di interventi aggiuntivi. Questo approccio proattivo contribuisce a creare un ambiente di apprendimento più inclusivo e supportivo, affrontando le dimensioni emotive dell'educazione spesso trascurate.

Nel contesto dell'educazione online e delle piattaforme di e-learning, la sentiment analysis può migliorare la personalizzazione delle esperienze di apprendimento. Analizzando interazioni come post nei forum, consegne di compiti o registri di chat, la sentiment analysis fornisce informazioni sulle risposte emotive degli studenti ai materiali didattici e alle strategie di insegnamento. Queste informazioni permettono ai sistemi di apprendimento adattivo di personalizzare contenuti e risorse in base alle esigenze individuali, migliorando sia il coinvolgimento sia i risultati di apprendimento.

Inoltre, la sentiment analysis è stata utilizzata per studiare le dinamiche delle interazioni in classe e le pratiche didattiche. Analizzando le trascrizioni delle discussioni in aula o le registrazioni delle lezioni, i ricercatori possono valutare il tono emotivo e i livelli di coinvolgimento degli studenti. Tendenze positive nei sentiment possono indicare strategie di insegnamento efficaci, mentre tendenze negative potrebbero segnalare la necessità di apportare modifiche. Questo approccio fornisce agli educatori informazioni utili per affinare i metodi pedagogici e favorire un ambiente scolastico più coinvolgente e supportivo dal punto di vista emotivo.

Analizzando dataset su larga scala, come post sui social media o forum pubblici, è possibile misurare l'opinione pubblica su iniziative, politiche e riforme educative. Ad esempio, i responsabili politici potrebbero utilizzare la sentiment analysis per valutare l'accoglienza di un nuovo quadro curriculare o l'impatto di modifiche ai metodi di valutazione standardizzati. Questo approccio basato sui dati garantisce che le decisioni siano allineate con le esigenze e le aspettative di studenti, genitori ed educatori.

Nonostante il suo potenziale, l'applicazione della sentiment analysis nell'educazione deve affrontare alcune sfide. Un ostacolo significativo è rappresentato dalle questioni etiche e dalle preoccupazioni relative alla privacy nell'analisi dei dati generati dagli studenti. Le istituzioni devono garantire che i processi di raccolta e analisi dei dati rispettino le normative sulla privacy e i diritti degli studenti. Inoltre, la natura contestuale del linguaggio nei contesti educativi richiede che i modelli di sentiment analysis siano ottimizzati per cogliere le sfumature specifiche del dominio. Ad esempio, una critica di uno studente a un compito difficile potrebbe riflettere impegno anziché disimpegno, richiedendo un'interpretazione accurata.

2 | PROPOSED PROJECT

Nei seguenti paragrafi, descriveremo perché abbiamo deciso di affrontare il problema della sentiment analysis e perche ci siamo voluti focalizzare sul contesto educativo. Acceneremo inoltre alle prime scelte riguardanti come abbiamo deciso di strutturare il progetto e ad altre informazioni relative al problema in esame.

2.1 | Domain choices

2.1.1 - Why chose education

La scelta di applicare la sentiment analysis al contesto educativo nasce dalla crescente esigenza di comprendere meglio le percezioni, le opinioni e le esperienze degli studenti in relazione al materiale didattico, alle lezioni e agli approcci pedagogici adottati. In un panorama in cui la qualità dell'educazione è fortemente influenzata dalla capacità degli insegnanti e delle istituzioni di rispondere ai bisogni degli studenti, l'analisi automatizzata delle opinioni offre un'opportunità unica per raccogliere e sintetizzare informazioni critiche.

La sentiment analysis consente di identificare rapidamente tendenze, punti di forza e aree di miglioramento, riducendo il tempo e le risorse necessari per l'analisi manuale del feedback. Inoltre, l'educazione rappresenta un campo in cui le emozioni e le percezioni svolgono un ruolo determinante nel determinare l'efficacia dei processi di apprendimento.

Concentrandoci su questo dominio applicativo, possiamo contribuire a migliorare la qualità del sistema educativo, fornendo ai docenti e agli amministratori strumenti per prendere decisioni basate sui dati. La nostra scelta riflette quindi la necessità di utilizzare tecnologie avanzate per migliorare l'efficienza e l'efficacia del settore educativo, valorizzando le opinioni degli studenti come fonte di informazioni strategiche.

2.1.2 - What we focus on

Nell'applicare la sentiment analysis al contesto educativo, abbiamo scelto di concentrarci su specifici aspetti che riguardano la valutazione di materiali didattici, lezioni e approcci pedagogici. Sebbene l'analisi delle emozioni e della salute mentale degli studenti sia un'area di interesse importante, abbiamo deliberatamente deciso di non focalizzarci su questo ambito, né sulla personalizzazione della didattica. Questi aspetti richiederebbero modelli altamente specializzati e l'adozione di misure etiche più rigorose, che esulano dagli obiettivi di questo progetto.

Invece, il nostro interesse si concentra sull'analisi delle recensioni e dei commenti relativi a contenuti come manuali, dispense e corsi online, nonché sul feedback riguardante le lezioni erogate e le metodologie di insegnamento. La scelta di questo focus deriva dalla volontà di offrire un supporto concreto agli educatori, fornendo loro strumenti per migliorare la qualità delle risorse e dei metodi utilizzati.

Questo approccio si distingue per la sua praticità e immediatezza, permettendo di identificare rapidamente aree di miglioramento e di adottare strategie che rispondano in modo diretto alle esigenze espresse dagli studenti. Tale scelta, inoltre, è in linea con la crescente importanza attribuita all'uso di dati per ottimizzare l'efficacia dell'educazione in contesti sia tradizionali che digitali.

2.2 | Application choices

2.2.1 - Why chose web application

La scelta di sviluppare un'applicazione web come artefatto del progetto è stata determinata dalla necessità di garantire accessibilità, flessibilità e facilità d'uso a un vasto numero di utenti, tra cui studenti, docenti e amministratori. Le applicazioni web rappresentano un mezzo ideale per l'implementazione di strumenti basati su sentiment analysis, poiché consentono la raccolta in tempo reale di dati provenienti da diverse utenze; inoltre incentivano all'ammodernamento dei metodi didattici andando ad integrarle facilmente ad ogni tipo di contesto.

Rispetto ad altre tipologie di artefatti, come software desktop o applicazioni mobili, le applicazioni web offrono il vantaggio di essere indipendenti dalla piattaforma e di poter essere utilizzate su qualsiasi dispositivo dotato di un browser. Questo approccio consente inoltre un aggiornamento continuo delle funzionalità e l'implementazione di nuove metodologie di analisi senza la necessità di interventi complessi da parte degli utenti finali.

La nostra scelta riflette quindi l'importanza di fornire uno strumento versatile e scalabile, in grado di rispondere alle esigenze mutevoli del contesto educativo e di favorire la diffusione dell'innovazione tecnologica nel settore.

2.2.2 - What the web-app does

L'applicazione web sviluppata per questo progetto mira a offrire una soluzione pratica ed efficace per l'analisi delle opinioni e delle percezioni degli studenti, in linea con gli obiettivi identificati per il dominio applicativo. L'applicazione consente di raccogliere e analizzare automaticamente feedback relativi a materiali didattici, lezioni e approcci pedagogici, fornendo risultati chiari e sintetici sotto forma di report visuali e metriche.

Questo strumento permette agli educatori di monitorare il sentiment generale degli studenti e di identificare eventuali aree di criticità, supportando un processo decisionale informato e basato sui dati.

L'applicazione è stata progettata per garantire un'esperienza utente intuitiva, rendendo semplice l'utilizzo delle funzionalità di analisi anche a personale non esperto di tecnologie avanzate. L'approccio adottato riflette la volontà di mettere a disposizione un sistema accessibile e altamente efficace per migliorare la qualità dell'educazione, valorizzando il ruolo centrale delle opinioni degli studenti nella progettazione di esperienze di apprendimento più coinvolgenti e significative.

2.3 | Methodology choices

2.3.1 - Why choose ML/DL

La scelta di focalizzarsi su approcci basati su machine learning (ML) e deep learning (DL) rispetto a metodi rule-based o algoritmi di ricerca tradizionali è stata guidata dalla complessità e dalla variabilità del linguaggio naturale utilizzato nei contesti educativi. Gli approcci rule-based, pur essendo semplici da implementare, risultano spesso inadeguati per gestire la polisemia, le espressioni idiomatiche e la complessità del feedback degli studenti, che può includere sarcasmo, ironia e ambiguità.

Al contrario, i modelli di ML e DL offrono la possibilità di apprendere direttamente dai dati, garantendo una maggiore capacità di adattamento alle sfumature linguistiche e ai contesti specifici. Le reti neurali profonde, in particolare, permettono di catturare relazioni semantiche e sintattiche complesse, migliorando significativamente la precisione

dell'analisi del sentiment. Inoltre, l'utilizzo di tecnologie avanzate come BERT o GPT ha reso possibile lo sviluppo di modelli pre-addestrati che possono essere facilmente adattati al dominio educativo, riducendo i tempi e i costi di implementazione. La nostra scelta riflette quindi l'obiettivo di ottenere risultati accurati e affidabili, sfruttando il potenziale delle tecnologie più avanzate nel campo dell'elaborazione del linguaggio naturale.

2.3.2 - Pipeline step choices

Nello sviluppo dell'applicazione, particolare attenzione è stata dedicata alla scelta delle metodologie di preprocessing e dei modelli utilizzati, al fine di garantire risultati precisi e significativi nell'analisi del sentiment. Il preprocessing dei dati testuali ha incluso fasi fondamentali come la tokenizzazione, la rimozione di stopword, la lemmatizzazione e la gestione dei caratteri speciali, per assicurare che i testi fossero adeguatamente normalizzati e pronti per l'analisi. Questi passaggi sono stati fondamentali per ridurre il rumore nei dati ed estrarre le caratteristiche linguistiche più rilevanti.

Per quanto riguarda i modelli, abbiamo scelto di utilizzare approcci classici di machine learning, come le Support Vector Machines (SVM) e la logistic regression, che offrono una solida capacità di catturare pattern nei dati strutturati e interpretabili. Questi modelli sono stati ulteriormente ottimizzati e addestrati su dataset specifici del dominio educativo, garantendo così una maggiore accuratezza nell'analisi delle opinioni relative ai materiali didattici e alle metodologie di insegnamento. La combinazione di un preprocessing rigoroso e di modelli classici ben consolidati ha permesso di ottenere un sistema performante, in grado di soddisfare le esigenze del progetto e di fornire risultati utili e affidabili.

3 | MACHINE LEARNING PIPELINE

Per lo sviluppo del progetto, abbiamo adottato il modello CRISP-DM (Cross-Industry Standard Process for Data Mining), una metodologia flessibile e consolidata per la gestione di progetti di data science. La scelta del CRISP-DM è stata motivata dalla natura del nostro progetto, che è di dimensioni relativamente ridotte e non coinvolge un numero elevato di figure professionali con competenze altamente differenziate. In questo contesto, la necessità di garantire una rigorosa coerenza socio-consequenziale tra i diversi attori non rappresenta una priorità assoluta.

3.1 | Business understanding

Prima di iniziare a lavorare con i dati, è opportuno effettuare un riepilogo delle scelte prese e delle decisioni effettuate fino a questo momento. Come descritto nel paragrafo 2.1, abbiamo deciso di focalizzarci sull'ambito educational, con particolare attenzione all'analisi dei commenti e delle recensioni fornite dagli studenti. Su tali commenti verrà applicata una document-level sentiment analysis, mentre eventuali miglioramenti o approfondimenti derivanti da livelli di granularità più elevati saranno lasciati alle considerazioni finali.

Tutti i modelli che saranno sviluppati e testati si baseranno su approcci di machine learning (ML) o deep learning (DL). Per ciascun modello, saranno dettagliate le caratteristiche specifiche adottate nel nostro caso di studio. Successivamente, ogni modello sarà valutato e confrontato con gli altri, al fine di individuare la soluzione più performante da implementare e utilizzare nell'applicazione finale.

Nel corso di questa analisi, prenderemo in considerazione anche l'impatto dei diversi tipi di word embedding sulle

prestazioni dei modelli. Una volta completate tutte le analisi e valutazioni necessarie, procederemo con lo sviluppo dell'applicazione vera e propria, integrando le soluzioni più efficaci emerse durante il processo.

3.2 | Data understanding

I modelli che intendiamo analizzare saranno addestrati su dataset composti essenzialmente da commenti o recensioni lasciate da studenti riguardanti aspetti legati alla didattica. Sebbene sia comune per le istituzioni raccogliere questo tipo di dati a fini di analisi e miglioramento, spesso tali informazioni non vengono rese pubbliche a causa di vincoli legati alla privacy. Questo rende difficile reperire online dataset già predisposti e di dimensioni sufficienti per un'analisi approfondita. Di conseguenza, è necessario esplorare altre fonti da cui trarre i dati, come forum o blog di discussione.

In particolare, durante le nostre ricerche online, ci siamo imbattuti in due dataset disponibili su Mendeley che potrebbero fungere da base per costruire un dataset adatto al nostro scopo. Il primo, è stato ottenuto dall'estrazione dei commenti presenti sull'omonimo sito. Questo dataset include diverse informazioni, come la media delle valutazioni assegnate ai professori e il dipartimento di appartenenza, ma l'aspetto di maggiore interesse per il nostro caso d'uso è costituito dai commenti testuali. Tuttavia, i commenti non risultano pre-labellati per quanto riguarda il sentiment, il che rende necessaria un'ulteriore fase di elaborazione per poterli utilizzare.

Il secondo dataset è stato invece progettato specificamente per la sentiment analysis e raccoglie feedback di studenti dell'American International University-Bangladesh. I commenti sono già etichettati in tre classi di sentiment (positivo, negativo e neutro) e sono stati sottoposti a una fase iniziale di pulizia e preprocessing. Tuttavia, sono emersi alcuni problemi significativi, tra cui uno squilibrio marcato tra le classi, con una predominanza eccessiva di sentiment positivi rispetto agli altri, e la presenza di numerosi commenti duplicati con etichette contraddittorie.

Considerando la scarsità di alternative facilmente accessibili, riteniamo opportuno partire dai due dataset individuati, adattandoli alle nostre necessità attraverso una fase di raffinamento che comprenda una rietichettatura dei dati ove necessario. Al termine di tale fase, intendiamo combinare i due dataset ottenuti per ottenere una base dati più robusta, bilanciata e adatta al nostro obiettivo. In futuro, potrebbe essere interessante esplorare tecniche avanzate di web scraping o altre metodologie per ottenere commenti più vari e rappresentativi

3.3 | Data engineering

3.3.1 - Datasets refine process

Partiamo dal primo dataset che contiene circa 20.000 record costituiti da diverse feature, tra cui i commenti. La prima operazione che abbiamo effettuato consiste nell'eliminazione delle feature non rilevanti per il problema, al fine di isolare esclusivamente i commenti. Una volta fatto questo, ci ritroviamo ad affrontare la principale criticità del dataset: i commenti non sono classificati. Poiché il numero di record è troppo elevato, non è possibile procedere con un'annotazione manuale. Per tale motivo, abbiamo optato per l'utilizzo di un modello pre-addestrato specificamente progettato per la classificazione dei testi.

Dato che abbiamo scelto una classificazione in tre sentimenti – negativo, positivo e neutro – abbiamo individuato Twitter-roBERTa-base for Sentiment Analysis, disponibile su Hugging Face, come modello adeguato alle nostre esigenze. Va comunque sottolineato che l'impiego di un modello pre-addestrato introduce un margine di errore nella classifi-

cazione, che dovrà essere considerato nei risultati finali.

In scenari differenti, non sarebbe stato possibile fare affidamento su strumenti di questo tipo, e le annotazioni avrebbero dovuto essere effettuate manualmente, da parte dello sviluppatore stesso o da un gruppo di esperti del settore. Una volta classificati i commenti, abbiamo salvato il dataset risultante in un file temporaneo, per poi procedere con l'analisi del secondo.

Il secondo dataset individuato, è invece composto da circa 2 milioni di commenti, già annotati e parzialmente pre-processati. Sebbene un numero elevato di dati possa essere generalmente vantaggioso, in questo caso non disponiamo della potenza di calcolo necessaria per gestirli interamente. Inoltre, risulta evidente che il creatore del dataset abbia privilegiato la quantità dei campioni rispetto alla loro qualità.

Anche senza un'analisi statistica approfondita, è emerso, fin dalle prime centinaia di righe, che diversi commenti sono ripetuti più volte, talvolta con annotazioni differenti. Questo rappresenta un problema rilevante, poiché tali incongruenze potrebbero confondere i modelli, compromettendo i risultati.

In base a queste considerazioni, abbiamo quindi deciso di effettuare una significativa riduzione della dimensione del dataset, selezionando casualmente un sottoinsieme di 100.000 record dalle 2 milioni di entry disponibili. Per garantire che non vi siano duplicati nella selezione, abbiamo utilizzato la funzione sample() della libreria pandas. Successivamente, abbiamo isolato le feature rilevanti – commento e sentimento – e salvato il risultato in un dataset temporaneo.

Dopo aver ottenuto i due dataset raffinati, questi sono stati uniti in un unico dataset, che è stato successivamente sottoposto a un'analisi approfondita.

3.3.2 - Dataset analysis

Una delle prime e più rilevanti operazioni effettuate è stata l'analisi della distribuzione delle classi. La presenza di un dataset bilanciato è di fondamentale importanza, in quanto uno squilibrio significativo tra le classi potrebbe introdurre problemi di bias e varianza, compromettendo l'efficacia dell'addestramento dei modelli.

A tal proposito, è stato osservato che uno dei dataset utilizzato nella fase precedente, presentava un forte squilibrio a sfavore della classe positiva. Questo andamento si è confermato anche nel dataset unificato e raffinato, come evidenziato dalla distribuzione mostrata nel seguente istogramma.

Oltre allo sbilanciamento marcato delle classi, un altro aspetto rilevante emerso riguarda la ripetizione di numerose entry all'interno del dataset. Questo fenomeno ha portato a una netta prevalenza di alcune parole ricorrenti nei commenti, tra cui "good", "bad" e "teacher". L'evidenza di tale concentrazione lessicale è emersa sia attraverso una prima rappresentazione visiva tramite word cloud, sia attraverso un'analisi più approfondita condotta su unigram, bigram e trigram.

Infine, abbiamo ritenuto utile analizzare alcune statistiche descrittive relative ai commenti, concentrandoci in particolare sulla lunghezza media delle parole in relazione al sentimento espresso. Quest'analisi si è dimostrata utile per comprendere meglio le caratteristiche stilistiche e semantiche dei commenti, fornendo informazioni sufficienti per proseguire.

3.3.3 - Dataset balancing

Prima di procedere con operazioni come la lemmatizzazione o la rimozione di stopword, è necessario bilanciare la distribuzione delle classi nel dataset. In questi casi, le due principali strategie adottate sono l'undersampling e l'oversampling. Considerando che la classe positiva rappresenta una porzione significativamente maggiore rispetto alle altre classi, sarà necessario ridimensionarla.

Tuttavia, ridurre esclusivamente il numero di elementi della classe positiva fino a renderla uniforme con le altre comporterebbe una drastica diminuzione della dimensione complessiva del dataset, aumentando così il rischio di bias e compromettendo la capacità del modello di apprendere pattern significativi dai dati. Pertanto, abbiamo scelto di applicare una serie di operazioni di oversampling sulle restanti classi prima di allineare la classe positiva.

Per creare campioni sintetici delle classi neutral e negative, abbiamo deciso di procedere con la generazione di campioni sintetici tramite prompt engineering utilizzando un modello preaddestrato. Per garantire una maggiore generalizzazione, abbiamo utilizzato un large language model (LLM). In particolare, abbiamo adottato un tool chiamto OIlama, che consente l'utilizzo locale di modelli avanzati, e il modello Llama3 di Meta. La generazione di commenti sintetici tramite Llama è stata affrontata con due approcci distinti:

1. Approccio basato sui dati esistenti: Questo approccio prevedeva la generazione di nuovi commenti a partire dalle entry esistenti del dataset. Il modello creava commenti sintetici rispettando vincoli come la lunghezza massima e il formato dell'output, ma mantenendo il sentimento desiderato. Tuttavia, questa strategia si è dimostrata poco efficace, poiché i commenti generati risultavano spesso simili tra loro.
2. Approccio basato su prompt variabili: Per aumentare la diversità, abbiamo limitato il modello a rispondere esclusivamente a un sottoinsieme di prompt da noi forniti. Questi prompt, centrati sul sentimento desiderato, venivano variati a intervalli regolari. La scelta dei prompt avveniva casualmente tra le diverse opzioni disponibili, permettendo al modello di generare risposte più eterogenee rispetto all'approccio precedente. Ogni 2500 campioni sintetici generati, i prompt venivano completamente sostituiti, mantenendo però il tema e il sentimento specificati. In questo modo, siamo riusciti a incrementare sia il numero di entry negative che quello delle entry neutral.

Una volta generati i campioni sintetici necessari, li abbiamo integrati con il dataset originale raffinato.Ancora una volta evidenziamo il potenziale margine di errore introdotto dall'utilizzo di dati sintetici generati tramite Llama3.

Sebbene queste tecniche abbiano permesso di bilanciare il dataset e aumentare la rappresentatività delle classi sottorappresentate, il rischio di introdurre rumore o dati non completamente rappresentativi dei pattern reali deve essere tenuto in considerazione durante l'analisi dei risultati del modello.

3.3.4 - Data preprocessing

Con le operazioni effettuate abbiamo finalmente ottenuto un dataset completo, abbastanza generalizzato e ben distribuito tra le tre classi si sentiment. Completiamo questa prima fase di preprocessing con un'analisi del dataset finale, che andremo ad utilizzare per l'addestramento dei nostri modelli.

Il preprocessing è una fase fondamentale nell'elaborazione di un dataset utilizzato per elaborazione di linguaggio naturale. Consiste nell'applicare una serie di trasformazioni ai dati per migliorarne la qualità e renderli adatti all'analisi. Questa fase è cruciale per eliminare elementi di disturbo, uniformare i dati e facilitare l'identificazione di pattern

rilevanti. Un preprocessing accurato non solo riduce il rumore nei dati, ma aumenta l'efficacia e la precisione dei modelli di machine learning o deep learning. Nel nostro caso, al dataset generato, abbiamo effettuato le seguenti trasformazioni:

1. Lowercase: convertire tutto il testo in minuscolo è una tecnica di base per evitare che variazioni nella capitalizzazione, come "Buono" e "buono", vengano interpretate come termini differenti dal modello.
2. Clean text spaces and digits: rimuovere spazi ridondanti e disordinati garantisce un testo strutturato correttamente, migliorando il processo di tokenizzazione e l'efficienza dei successivi passaggi. Ciò vale anche per i numeri, i quali non rappresentano informazioni rilevanti per il modello
3. Remove punctuation: la punteggiatura spesso non aggiunge informazioni rilevanti in un contesto di Sentiment Analysis. Rimuoverla consente di focalizzarsi esclusivamente sul contenuto semantico del testo.
4. Remove long lines: le righe molto lunghe possono contenere rumore, come descrizioni prolisse o informazioni irrilevanti. Eliminandole, si semplifica il dataset e si concentra l'analisi su dati più rilevanti. Questa scelta è stata presa anche in considerazione della media della lunghezza delle recensioni, la quale ha evidenziato una scarsa presenza di commenti lunghi più di 25/30 parole ciascuna
5. Remove stopwords: le stopwords sono parole molto frequenti e comuni, come articoli, congiunzioni o preposizioni (ad esempio: "e", "ma", "di"). Questi termini, pur essendo utili a livello sintattico, non apportano valore semantico all'analisi. La loro rimozione riduce il volume di dati elaborati dal modello e permette di concentrarsi sulle parole che hanno un impatto diretto sull'identificazione del sentimento.
6. Lemmatization: la lemmatizzazione è una fase chiave per uniformare il vocabolario del testo. Questo processo riduce le parole alla loro forma base o lemma, considerando il loro significato. Ad esempio, "camminando", "camminava" e "camminato" vengono ricondotti a "camminare". Ciò permette di eliminare informazioni che non influenzano il contenuto semantico e aiuta il modello a riconoscere i concetti, piuttosto che le sole occorrenze superficiali delle parole. Questa operazione è essenziale per migliorare la coerenza e ridurre la complessità del dataset.
7. Remove short lines: le righe molto brevi contengono raramente informazioni utili, come commenti privi di contesto o espressioni generiche. La loro eliminazione garantisce un dataset più pulito e rappresentativo. In questa trasformazione andiamo principalmente ad eliminare valori nulli generati dalle fasi precedenti

3.3.4 - Data embedding

Nel campo del Natural Language Processing (NLP), la rappresentazione del testo è uno step fondamentale per trasformare i dati testuali, intrinsecamente non strutturati, in un formato numerico che i modelli di machine learning o deep learning possano interpretare. Poiché i modelli lavorano esclusivamente con numeri, convertire parole o frasi in vettori numerici consente di analizzare, elaborare e trarre informazioni significative dal linguaggio naturale.
tecniche di rappresentazione del testo mirano a preservare il significato semantico, le relazioni sintattiche e i contesti delle parole. Le metodologie più semplici includono il bag of words (BoW) e il TF-IDF (Term Frequency-Inverse Document Frequency) , che si basano su contatori di frequenze per costruire vettori di rappresentazione. Tuttavia, queste tecniche ignorano l'ordine delle parole e i loro significati contestuali.

Con l'avanzamento dell'NLP, tecniche più sofisticate come i word embeddings sono state sviluppate per superare i limiti dei metodi tradizionali. I word embeddings mappano le parole in uno spazio vettoriale continuo in cui la similarità semantica tra le parole è preservata. Questo approccio ha rivoluzionato il campo, poiché permette di rappresentare le parole in modo denso e ricco di significato. Di seguito elenchiamo le tecniche di embedding che abbiamo deciso di utilizzare per valutare quanto le performance dei modelli dipendano da esse

TF-IDF Embedding La tecnica TF-IDF (Term Frequency-Inverse Document Frequency) è uno degli approcci più conosciuti e utilizzati per rappresentare il testo in modo numerico. Il suo obiettivo è assegnare a ogni parola un peso che rifletta la sua importanza in un documento specifico, tenendo conto anche della sua rilevanza rispetto all'intero corpus. In altre parole, TF-IDF permette di identificare quanto una parola contribuisce al significato di un documento, escludendo termini comuni che appaiono ovunque, come articoli o congiunzioni.

TF-IDF combina due concetti chiave: Term Frequency (TF), che misura quanto frequentemente una parola appare in un documento specifico; Inverse Document Frequency (IDF), che valuta quanto una parola è rara nel corpus complessivo, penalizzando i termini troppo comuni.

Il risultato è una rappresentazione in cui ogni documento è descritto da un vettore di parole pesate in base alla loro importanza relativa.

TF-IDF è semplice da implementare e particolarmente efficace per applicazioni di machine learning tradizionali, come SVM o Naive Bayes. È ampiamente utilizzato per classificare documenti, recuperare informazioni (come nei motori di ricerca) o rilevare spam. La sua leggerezza lo rende ideale per scenari in cui le risorse hardware sono limitate.

TF-IDF presenta infatti alcune criticità. Non considera il contesto delle parole: termini ambigui come "banca" (che può riferirsi a un fiume o a un istituto finanziario) vengono trattati allo stesso modo, indipendentemente dalla frase. Inoltre, non riesce a cogliere relazioni semantiche o sinonimi. Queste limitazioni lo rendono inadatto a compiti più complessi, come la generazione di testo o l'analisi avanzata del sentiment, dove tecniche più sofisticate sono preferibili.

Word2Vec Embedding Word2Vec è una delle tecniche di word embedding più rivoluzionarie nella storia dell'elaborazione del linguaggio naturale. Sviluppata da Google nel 2013, si basa su reti neurali per rappresentare ogni parola come un vettore in uno spazio continuo, dove parole simili si trovano vicine. Questo approccio consente di catturare sia il significato semantico delle parole sia le loro relazioni contestuali.

Word2Vec utilizza due metodi principali per generare queste rappresentazioni:

- CBOW (Continuous Bag of Words), che prevede una parola basandosi sul contesto circostante. Ad esempio, dato un frammento come "Il gatto — sul divano", il modello può prevedere "dorme".
- Skip-Gram, che lavora in modo opposto, cercando di prevedere il contesto circostante partendo da una parola specifica. Ad esempio, partendo da "cane", il modello può prevedere parole come "abbaia" o "giardino".

Aspetto rivoluzionario di Word2Vec è la capacità di catturare relazioni semantiche attraverso i vettori. Parole con significati simili si trovano vicine nello spazio vettoriale, mentre relazioni come sinonimia o analogia emergono naturalmente. Ad esempio, Word2Vec è in grado di rappresentare relazioni come "re" e "regina".

Word2Vec supera le limitazioni delle tecniche precedenti come TF-IDF, producendo rappresentazioni dense che catturano meglio le relazioni tra parole. È ideale per task come il clustering di documenti, la traduzione automatica e la classificazione testuale. Inoltre, la sua efficienza e flessibilità ne hanno fatto una pietra miliare nell'NLP.

Nonostante i suoi punti di forza, Word2Vec ha alcune debolezze. Le rappresentazioni generate sono statiche: ogni

parola ha un unico vettore, indipendentemente dal contesto in cui si trova. Ad esempio, "banca" avrà lo stesso embedding sia che si parli di un fiume, sia che ci si riferisca a un istituto finanziario. Questo limite lo rende inadatto a task che richiedono una comprensione più contestuale, per cui modelli come BERT risultano più efficaci.

BERT Pre-trained Embedding BERT (Bidirectional Encoder Representations from Transformers) rappresenta un passo avanti rispetto a Word2Vec, introducendo un nuovo livello di comprensione contestuale. Sviluppato da Google AI nel 2018, BERT sfrutta l'architettura Transformer per analizzare il contesto delle parole in modo bidirezionale. Ciò significa che il modello tiene conto di tutte le parole in una frase, sia quelle a sinistra che quelle a destra, per determinare il significato di un termine specifico.

BERT si distingue per il suo approccio all'addestramento, che include due tecniche principali:

- Masked Language Modeling (MLM): Durante l'addestramento, alcune parole nella frase vengono mascherate, e il modello impara a prevederle basandosi sul contesto. Ad esempio, nella frase "Il gatto [MASK] sul divano", BERT impara che la parola mancante potrebbe essere "dorme".
- Next Sentence Prediction (NSP): BERT apprende anche le relazioni tra frasi consecutive, una capacità fondamentale per attività come la risposta a domande o la comprensione testuale.

BERT rappresenta un punto di svolta per l'elaborazione del linguaggio naturale grazie alla sua capacità di cogliere il contesto globale delle parole. Questa caratteristica lo rende ideale per compiti complessi, come l'analisi del sentiment, la traduzione automatica e la generazione di testo. Inoltre, grazie al pre-training su enormi quantità di dati, BERT può essere facilmente adattato a compiti specifici attraverso il fine-tuning.

Nonostante le sue potenzialità, BERT richiede risorse computazionali significative per l'addestramento e l'inferenza. Questo lo rende meno accessibile per progetti con risorse hardware limitate. Inoltre, la sua complessità lo rende meno pratico per applicazioni semplici o in tempo reale, dove modelli più leggeri come Word2Vec o TF-IDF possono essere più appropriati.

3.4 | Model creation

In questa sezione vengono descritti i modelli implementati per affrontare il problema della sentiment analysis. Ogni modello è stato selezionato e sviluppato per sfruttare specifiche caratteristiche dei dati e del task, al fine di garantire un'analisi accurata ed efficiente. Tra i modelli proposti, figurano algoritmi classici e consolidati, come il Multinomial Naive Bayes, il Support Vector Machine e la Logistic Regression, affiancati da approcci più avanzati basati su reti neurali, come le LSTM. Ogni metodo viene descritto dettagliatamente, con un focus sulle motivazioni alla base della scelta, le definizioni formali e i passaggi fondamentali per il loro funzionamento.

3.4.1 - Multinomial Naive Bayes

Overview Il Multinomial Naive Bayes (MNB) è un algoritmo probabilistico che si è dimostrato particolarmente efficace nella classificazione di dati testuali, come nel caso della sentiment analysis. Si basa sul principio del Naive Bayes, utilizzando il teorema di Bayes per calcolare la probabilità che un testo appartenga a una determinata classe in base alle sue caratteristiche. La principale differenza tra il Naive Bayes generico e la sua variante multinomiale risiede nell'assunzione di una distribuzione multinomiale per i dati. Questa caratteristica rende l'MNB particolarmente adatto

a lavorare con dati discreti, come le parole che compongono un testo.

Nel contesto dell'elaborazione del linguaggio naturale, l'MNB modella la frequenza delle parole nei testi per identificare modelli che caratterizzano le diverse classi. È particolarmente utile quando si lavora con dati ad alta dimensionalità e con distribuzioni sparse, entrambi tratti distintivi dei testi. Grazie alla sua semplicità ed efficienza computazionale, il modello è largamente utilizzato per applicazioni come la classificazione testuale e la sentiment analysis.

Formal Definition L'obiettivo principale del Multinomial Naive Bayes è quello di determinare a quale classe appartiene un determinato testo (ad esempio, positivo, negativo o neutro) sulla base delle parole che lo compongono. Il modello esegue questa classificazione analizzando la probabilità che il testo in esame sia associato a ciascuna classe, tenendo conto della frequenza delle parole sia all'interno del testo stesso sia nei dati di addestramento.

Per fare ciò, l'MNB calcola una serie di probabilità. In primo luogo, considera la distribuzione iniziale delle classi nel dataset, ovvero quanto frequentemente ogni classe compare nei dati di addestramento. Successivamente, stima quanto è probabile che ogni parola del testo appaia nei documenti di una specifica classe. Infine, queste informazioni vengono combinate per determinare la probabilità complessiva che il testo appartenga a ciascuna classe.

Un aspetto importante del Multinomial Naive Bayes è la capacità di gestire la presenza di parole che non sono mai apparse nei dati di addestramento, evitando che queste causino problemi nel calcolo delle probabilità. Questo è possibile grazie all'applicazione di tecniche di smoothing, come il Laplace Smoothing, che aggiungono una piccola correzione ai conteggi delle parole per evitare valori nulli.

Fundamental Steps Quando il Multinomial Naive Bayes viene utilizzato per classificare un nuovo testo, il processo segue una serie di passaggi chiari e ben definiti:

1. Calcolo delle probabilità iniziali delle classi: Il primo passo consiste nel determinare quanto frequentemente ogni classe (ad esempio, positivo, negativo o neutro) compare nei dati di addestramento. Questo passaggio stabilisce una sorta di punto di partenza per il calcolo delle probabilità.
2. Stima della probabilità delle parole nelle classi: Successivamente, il modello analizza quanto è probabile che ciascuna parola presente nel testo appartenga a una determinata classe. Questo viene fatto analizzando la frequenza con cui ogni parola compare nei documenti di ciascuna classe durante la fase di addestramento.
3. Applicazione di tecniche di smoothing: Per evitare che parole nuove o poco comuni, non presenti nei dati di addestramento, compromettano l'efficacia del modello, si applicano correzioni ai conteggi delle parole. Questo garantisce che anche le parole rare abbiano una probabilità minima associata.
4. Determinazione della probabilità complessiva delle classi: Utilizzando le probabilità calcolate nei passaggi precedenti, il modello combina tutte le informazioni disponibili per stimare la probabilità complessiva che il testo appartenga a ciascuna classe.
5. Assegnazione della classe: Infine, il modello assegna al testo la classe che ha la probabilità complessiva più alta.

Motivazioni scelta Abbiamo scelto di implementare il Multinomial Naive Bayes per una serie di motivi che ne evidenziano l'efficacia e l'idoneità per la sentiment analysis:

- Adattabilità ai dati testuali: L'MNB è particolarmente adatto alla natura dei dati testuali, poiché tratta in modo

efficace la frequenza delle parole. Questo lo rende ideale per applicazioni in cui le parole stesse costituiscono le caratteristiche principali.

- Efficienza computazionale: Grazie alla sua semplicità, l'MNB è un modello leggero che può essere addestrato e utilizzato per fare previsioni in modo molto rapido, anche con dataset di grandi dimensioni o in contesti che richiedono analisi in tempo reale.
- Gestione della sparsa distribuzione dei dati: I dati testuali spesso presentano una distribuzione sparsa, in cui molte parole appaiono raramente nei documenti. Il Multinomial Naive Bayes gestisce bene questa caratteristica, concentrandosi sulle parole più frequenti e significative.
- Riduzione del rischio di overfitting: Il modello, basandosi sull'assunzione di indipendenza tra le caratteristiche (le parole), evita di catturare relazioni complesse che potrebbero portare a un sovradattamento ai dati di addestramento. Questo lo rende più robusto quando applicato a dati nuovi.
- Compatibilità con tecniche di embedding: L'MNB si integra perfettamente con tecniche di rappresentazione testuale come il TF-IDF, che pesano le parole in base alla loro importanza relativa nel corpus. Questa combinazione permette al modello di distinguere meglio le parole più significative e di ignorare quelle irrilevanti, migliorando le prestazioni complessive nella classificazione.

3.4.2 - Support Vector Machine

Overview Il Support Vector Machine (SVM) è un potente algoritmo di apprendimento supervisionato ampiamente utilizzato per compiti di classificazione, inclusa la sentiment analysis. L'idea alla base dell'SVM è rappresentare i dati come punti in uno spazio multidimensionale, dove ogni dimensione corrisponde a una caratteristica del dato (ad esempio, una parola in un testo). L'obiettivo principale dell'SVM è trovare un iperpiano che separi in modo ottimale i punti appartenenti a classi diverse.

Nel caso di dati facilmente separabili, il modello cerca di massimizzare il margine, ovvero la distanza tra l'iperpiano e i punti più vicini di ciascuna classe, noti come vettori di supporto. Questo margine massimo garantisce una separazione più robusta e generalizzabile. Tuttavia, quando i dati non sono linearmente separabili (come accade spesso nei problemi reali), l'SVM utilizza tecniche avanzate, come le funzioni kernel, per trasformare i dati in uno spazio di dimensioni superiori. In questo spazio trasformato, diventa più semplice trovare una separazione lineare tra le classi. Questa capacità di adattarsi anche a dati complessi rende l'SVM estremamente versatile ed efficace.

What is a Kernel? Un elemento fondamentale del Support Vector Machine è il concetto di kernel. Il kernel è una funzione che consente di trasformare i dati in uno spazio a dimensioni più elevate, dove la separazione tra le classi può diventare più semplice. Questa trasformazione è particolarmente utile quando i dati non possono essere separati linearmente nello spazio originale.

Ad esempio, immagina un insieme di punti distribuiti in modo tale che sia impossibile tracciarvi una linea retta che separi le classi. Applicando una funzione kernel, i punti vengono "spostati" in uno spazio superiore, dove quella stessa separazione lineare diventa possibile. In altre parole, il kernel consente al modello di trovare soluzioni in spazi più complessi senza dover calcolare esplicitamente le trasformazioni. Questo approccio rende l'SVM adatto a una vasta gamma di problemi, anche molto complessi.

Quando invece i dati sono già separabili linearmente, si può utilizzare un kernel lineare che lascia i dati nel loro spazio originale, permettendo al modello di trovare una separazione semplice ed efficace.

Fundamental Steps Il processo seguito da un SVM per classificare i dati può essere suddiviso in tre passaggi principali:

1. Rappresentazione nello spazio delle caratteristiche: I dati vengono rappresentati come vettori in uno spazio multidimensionale, dove ogni dimensione corrisponde a una caratteristica (ad esempio, la presenza o la frequenza di una parola in un testo).
2. Individuazione dell'iperpiano ottimale: Durante la fase di addestramento, l'SVM cerca di individuare l'iperpiano che separa meglio i dati appartenenti a classi diverse. Il criterio chiave per questa separazione è la massimizzazione del margine, ossia la distanza tra l'iperpiano e i punti più vicini di ciascuna classe. Questo garantisce una classificazione robusta e generalizzabile.
3. Classificazione di nuovi dati: Una volta individuato l'iperpiano ottimale, l'SVM utilizza questa separazione per classificare i nuovi dati. La posizione di ciascun punto rispetto all'iperpiano determina la classe a cui appartiene.

Why choose it? Abbiamo deciso di implementare il Support Vector Machine per diversi motivi che ne evidenziano l'idoneità al nostro progetto:

- Separazione ottimale tra le classi: L'SVM è progettato per trovare la separazione migliore tra classi, rendendolo ideale per compiti come la sentiment analysis, dove spesso le classi (positivo, negativo, neutro) hanno confini complessi.
- Efficacia con dati ad alta dimensionalità: I dati testuali, come quelli utilizzati nella sentiment analysis, sono caratterizzati da un gran numero di caratteristiche (ad esempio, ogni parola rappresenta una dimensione). L'SVM eccelle in questi contesti, poiché può gestire efficacemente spazi a molte dimensioni.
- Robustezza al rumore: I dati testuali spesso contengono rumore, come parole non informative, errori di battitura o contenuti ambigui. L'SVM è noto per la sua capacità di resistere a questi problemi, concentrandosi solo sui punti più rilevanti, i vettori di supporto.
- Compatibilità con le tecniche di word embedding: Quando i testi vengono trasformati in rappresentazioni numeriche dense, come quelle generate da tecniche di embedding (ad esempio, Word2Vec), l'SVM è in grado di sfruttare queste rappresentazioni per migliorare la classificazione. Le proprietà semantiche catturate dagli embedding si combinano con la capacità dell'SVM di trovare separazioni ottimali, migliorando le performance complessive.

3.4.3 - Logistic Regression

Overview La Logistic Regression è un algoritmo di classificazione lineare ampiamente utilizzato per prevedere la probabilità che un'osservazione appartenga a una determinata classe. Nonostante il termine "regressione" possa far pensare a un modello di tipo predittivo-continuo, si tratta a tutti gli effetti di un modello di classificazione.

Il cuore della Logistic Regression è rappresentato dalla funzione logistica, conosciuta anche come funzione sigmoide. Questa funzione trasforma la combinazione lineare delle variabili di input in una probabilità compresa tra 0 e 1. In altre parole, il modello calcola la probabilità che un'osservazione appartenga a una specifica classe (ad esempio, sentiment positivo). Una volta calcolata questa probabilità, si utilizza una soglia predefinita (generalmente 0.5) per assegnare l'osservazione a una classe: se la probabilità supera la soglia, l'osservazione viene classificata come appartenente alla classe positiva, altrimenti a quella negativa.

La Logistic Regression è particolarmente efficace quando le classi sono separabili linearmente o quando esiste una relazione logaritmica tra le variabili di input e la variabile target.

Come funziona Il funzionamento della Logistic Regression può essere riassunto in pochi passaggi chiave. Per iniziare, il modello combina le variabili di input in un'espressione lineare (cioè somma pesata delle caratteristiche più un termine di bias). Questa combinazione lineare rappresenta il "punteggio grezzo" per ciascuna classe. Successivamente, il punteggio viene trasformato in una probabilità utilizzando la funzione logistica, che "comprime" il valore in un intervallo compreso tra 0 e 1.

Una volta calcolata la probabilità per una determinata classe, il modello prende una decisione: se questa probabilità è superiore a una soglia prestabilita (tipicamente 0.5), assegna l'osservazione alla classe positiva. Al contrario, se è inferiore, l'osservazione viene classificata come negativa.

La Logistic Regression utilizza un processo iterativo per ottimizzare i pesi associati alle variabili di input, in modo da ridurre al minimo gli errori nelle previsioni. Durante questa fase di addestramento, il modello apprende quali caratteristiche sono più rilevanti per distinguere tra le classi.

Fundamental Steps Il processo di classificazione della Logistic Regression può essere suddiviso in quattro fasi:

1. Costruzione del modello: Il primo passaggio consiste nel definire una combinazione lineare delle variabili di input. Questa rappresenta il punto di partenza per calcolare la probabilità che un'osservazione appartenga a una determinata classe.
2. Definizione della funzione di perdita: Il modello utilizza una funzione di perdita per valutare quanto le sue previsioni si discostano dai valori effettivi. Una delle funzioni più comuni è la Log Loss (o Binary Cross-Entropy), che penalizza fortemente le previsioni errate.
3. Ottimizzazione dei pesi: Durante l'addestramento, il modello utilizza tecniche di ottimizzazione, come il Gradient Descent, per modificare i pesi associati alle variabili di input. L'obiettivo è minimizzare la funzione di perdita e migliorare la capacità del modello di effettuare previsioni accurate.
4. Classificazione: Una volta addestrato, il modello utilizza la funzione logistica per calcolare la probabilità che un nuovo dato appartenga a una determinata classe. In base a questa probabilità e alla soglia scelta, il modello classifica l'osservazione.

Why choose it? La Logistic Regression rappresenta una scelta solida e ben motivata per diversi motivi, che ne evidenziano l'efficacia e la praticità, specialmente nel contesto della sentiment analysis:

- Semplicità e interpretabilità: È un modello lineare facile da comprendere e implementare. La Logistic Regression non solo classifica i dati, ma fornisce anche una probabilità associata a ogni previsione, permettendo di valutare il livello di confidenza del modello nelle sue decisioni. Inoltre, la semplicità della formula rende il modello particolarmente trasparente, facilitando l'interpretazione dei risultati.
- Efficienza nei dati linearmente separabili: Se le classi sono separabili linearmente, la Logistic Regression offre una soluzione semplice ed efficace. Questo la rende una scelta ideale quando i dati mostrano confini chiari tra sentiment positivo, negativo o neutro.
- Velocità di addestramento e inferenza: La Logistic Regression è un modello relativamente semplice dal punto di vista computazionale, il che garantisce tempi di addestramento e di inferenza rapidi. Questa caratteristica è particolarmente utile per applicazioni che richiedono analisi in tempo reale, come la sentiment analysis di commenti o recensioni.

- Interpretabilità dei pesi: I pesi associati alle variabili di input forniscono informazioni preziose sull'importanza relativa di ciascuna caratteristica. Ad esempio, in un task di sentiment analysis, i pesi possono aiutare a identificare quali parole influenzano maggiormente la classificazione di un testo come positivo o negativo.
- Compatibilità con tecniche di rappresentazione testuale: La Logistic Regression lavora bene con rappresentazioni come il TF-IDF, che assegna un peso alle parole in base alla loro rilevanza nel corpus. Questa combinazione permette di ottenere modelli più precisi e mirati per la sentiment analysis.

LSTM (Long Short-Term Memory)

Overview Le Long Short-Term Memory (LSTM) sono una variante avanzata delle reti neurali ricorrenti (RNN) progettate per superare i limiti delle RNN tradizionali. Questi limiti emergono soprattutto nella gestione delle dipendenze a lungo termine, ovvero quando le informazioni rilevanti in una sequenza si trovano a una distanza significativa dalle altre.

Le LSTM sono particolarmente efficaci nell'analisi di dati sequenziali, come i testi, in cui le relazioni tra parole distanti possono influenzare profondamente il significato complessivo. Per esempio, in una frase complessa, una parola all'inizio potrebbe determinare il sentimento generale dell'intera frase. Grazie alla loro struttura innovativa, le LSTM riescono a mantenere informazioni rilevanti nel tempo e a ignorare quelle meno utili, rendendole ideali per task come la sentiment analysis.

Architettura L'architettura delle LSTM è basata su una struttura simile a quella delle RNN tradizionali, ma introduce un elemento chiave: la cellula di memoria. Questa cellula rappresenta una sorta di "banca dati" interna, progettata per memorizzare informazioni importanti e mantenere il contesto lungo la sequenza. A differenza delle RNN, le LSTM gestiscono questa memoria tramite un sistema di tre porte principali, che agiscono come meccanismi di controllo:

- Input Gate: Questa porta determina quali nuove informazioni devono essere aggiunte alla memoria. Serve a identificare gli elementi importanti dell'input corrente, valutando se essi sono rilevanti per il contesto complessivo.
- Forget Gate: Come suggerisce il nome, questa porta decide quali informazioni devono essere eliminate dalla memoria. È utile per "dimenticare" dati non più rilevanti o potenzialmente fuorvianti, consentendo al modello di focalizzarsi su ciò che è importante.
- Output Gate: Questa porta controlla quali informazioni memorizzate devono essere utilizzate per generare l'output. L'output finale, quindi, rappresenta una combinazione del contesto attuale e delle informazioni precedenti, filtrate attraverso questa porta.

Grazie a questo sistema di porte, le LSTM sono in grado di memorizzare informazioni importanti per lunghi periodi di tempo e ignorare quelle non necessarie, superando così il problema della perdita di informazioni che caratterizza le RNN tradizionali.

Passaggi Fondamentali Il funzionamento delle LSTM può essere riassunto in una serie di passaggi principali:

1. Elaborazione dell'Input: Ogni parola (o token) nella sequenza viene analizzata dal modello. L'LSTM valuta se l'informazione associata a quel termine deve essere memorizzata o ignorata, utilizzando l'input gate per prendere questa decisione.
2. Aggiornamento della Memoria: Le informazioni rilevanti vengono aggiunte alla cellula di memoria, mentre quelle non necessarie vengono eliminate grazie al forget gate. Questo passaggio consente al modello di mantenere una

memoria “pulita” e focalizzata sul contesto più significativo.

3. Generazione dell’Output: Basandosi sulle informazioni presenti nella cellula di memoria, il modello produce un output. Questo output rappresenta una previsione basata sia sull’input corrente che sul contesto accumulato lungo la sequenza.
4. Aggiornamento dello Stato Nascosto: L’output viene utilizzato per aggiornare lo stato nascosto del modello, che rappresenta la memoria a breve termine dell’LSTM. Questo stato nascosto viene poi passato al passo successivo nella sequenza, garantendo che le informazioni essenziali siano disponibili per le previsioni future.

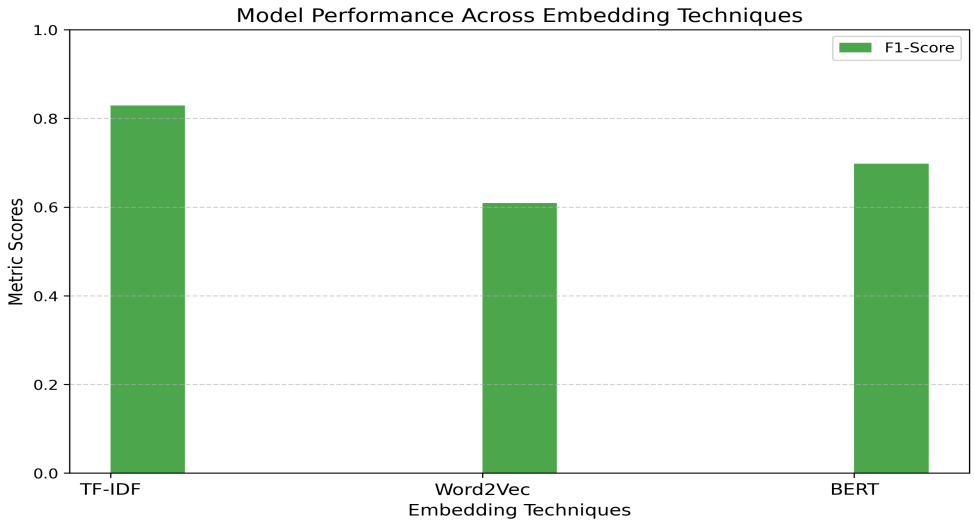
Why We Choose It Le LSTM sono particolarmente indicate per la sentiment analysis per una serie di motivi che ne evidenziano la potenza e l’efficacia:

- Gestione delle Dipendenze a Lungo Termine: Una delle caratteristiche principali delle LSTM è la loro capacità di comprendere e utilizzare le relazioni tra parole distanti all’interno di un testo. Questo è fondamentale nella sentiment analysis, dove il significato di una frase può dipendere da parole che si trovano in punti molto distanti della sequenza.
- Riconoscimento del Contesto: Molto spesso, una parola assume significato solo in relazione al contesto in cui è inserita. Ad esempio, la parola “incredibile” può essere positiva o negativa a seconda del tono e delle altre parole circostanti. Le LSTM eccellono nel mantenere il contesto rilevante e nel fare previsioni accurate tenendo conto delle relazioni tra le parole.
- Efficacia nell’Analisi di Sequenze di Testo: Le LSTM sono state progettate specificamente per lavorare con dati sequenziali, come i testi, dove l’ordine delle parole è cruciale per comprendere il significato. Questo le rende una scelta ideale per compiti di elaborazione del linguaggio naturale (NLP), come la classificazione del sentiment.
- Flessibilità per Dati Complessi: Le LSTM possono gestire frasi lunghe e strutturate in modo complesso, dove i confini tra sentiment positivo, negativo o neutro non sono immediatamente evidenti. La loro capacità di memorizzare informazioni rilevanti e dimenticare quelle inutili consente di ottenere previsioni più affidabili.

3.5 | Model evaluation

In questo paragrafo analizzeremo le prestazioni dei diversi modelli implementati, concentrandoci principalmente sulla metrica F1-score, poiché essa rappresenta una misura più completa ed equilibrata della performance, combinando sia la precisione che il recall. Successivamente, confronteremo i vari modelli per determinare quale di essi si adatti meglio al nostro task di classificazione, tenendo conto delle specifiche caratteristiche e necessità del progetto.

3.5.1 - Prestazioni del Multinomial Naive Bayes Il Multinomial Naive Bayes (MNB) ha mostrato una performance significativamente migliore con la rappresentazione TF-IDF rispetto a Word2Vec e BERT. Questo risultato è spiegabile considerando le caratteristiche intrinseche del modello MNB e le peculiarità degli embedding utilizzati.

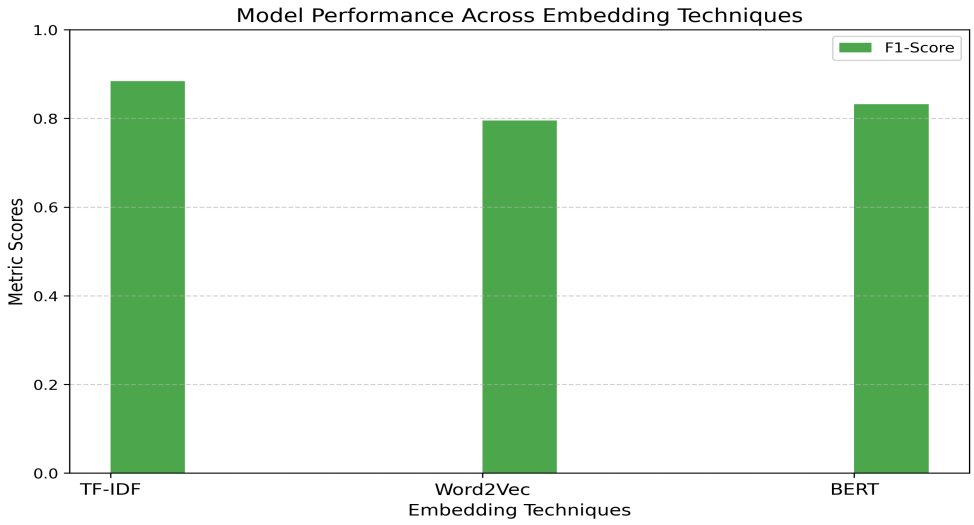


MNB - TF-IDF Il Multinomial Naive Bayes si adatta particolarmente bene a rappresentazioni sparse e discrete come quelle prodotte dal metodo TF-IDF. La natura lineare e interpretabile di TF-IDF permette al modello di sfruttare le assunzioni di indipendenza condizionale tra le caratteristiche, risultando in una classificazione più precisa. In particolare, TF-IDF consente al MNB di rappresentare efficacemente l'importanza relativa dei termini in un documento rispetto al corpus e di seguire l'assunzione multinomiale, in cui ogni termine contribuisce in modo indipendente alla classificazione. Questo rende TF-IDF particolarmente adatto per il MNB.

MNB - Word2Vec Word2Vec genera vettori densi in uno spazio continuo, che catturano le relazioni semantiche tra le parole. Tuttavia, questa rappresentazione presenta delle criticità quando viene utilizzata con il MNB: - I vettori generati da Word2Vec non sono sparsi e non rispettano l'assunzione di indipendenza condizionale del modello. - La rappresentazione di Word2Vec è indipendente dal contesto del documento, riducendo la capacità del MNB di riconoscere le relazioni tra i termini e le classi target in modo efficace.

MNB - BERT Le rappresentazioni prodotte da BERT sono dense e contestuali, catturando significati complessi delle frasi e delle parole all'interno di un determinato contesto. Tuttavia: - La natura densa e contestuale delle embedding di BERT non si adatta alle assunzioni matematiche di MNB, che richiede input lineari e interpretabili. - La complessità delle rappresentazioni di BERT è progettata per essere utilizzata con modelli avanzati, come le reti neurali profonde, che sono in grado di sfruttarne appieno la ricchezza semantica.

3.5.2 - Prestazioni della Support Vector Machine La Support Vector Machine (SVM) ha ottenuto ottimi risultati con tutte le tipologie di codifica dei dati testuali. Tuttavia, la migliore performance è stata registrata utilizzando il metodo TF-IDF. Questo risultato può essere attribuito principalmente alle caratteristiche intrinseche del modello SVM e alle specifiche proprietà degli embedding

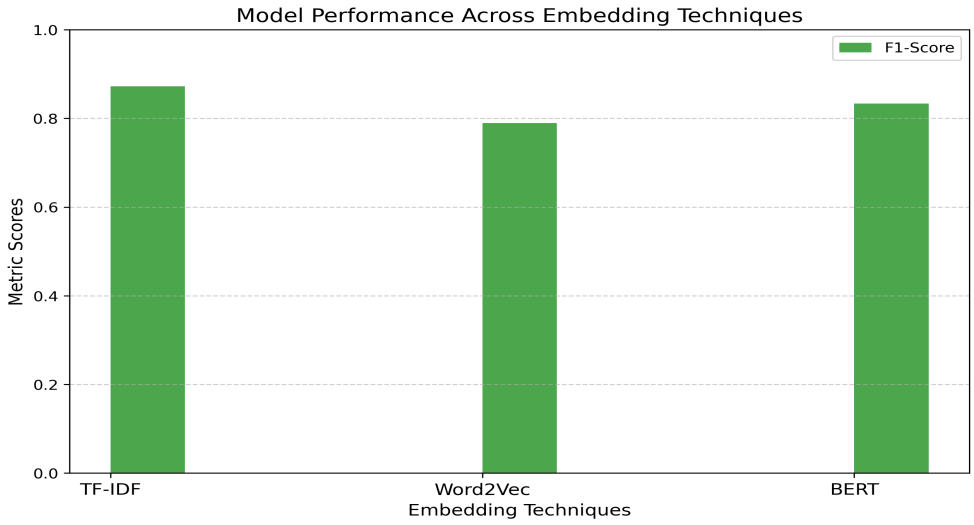


SVM - TF-IDF TF-IDF funziona meglio con le SVM perché le rappresentazioni sparse e di alta dimensionalità risultano facilmente separabili in uno spazio lineare. Le SVM sono particolarmente adatte a lavorare con dati sparsi e a trovare un iperpiano ottimale per separare le classi. La struttura sparsa di TF-IDF consente alla SVM di operare con maggiore efficacia rispetto ad altri metodi di embedding, che generano rappresentazioni più dense.

SVM - Word2Vec Word2Vec, essendo una rappresentazione densa e continua, non cattura direttamente i termini più discriminanti per la classificazione. SVM, che è progettate per lavorare con dati sparsi e separabili linearmente, non riesce a sfruttare pienamente la struttura semantica complessa di Word2Vec, risultando meno efficace in questo contesto.

SVM - BERT BERT genera rappresentazioni contestuali ad alta dimensionalità, che sono troppo complesse per essere gestite efficacemente da SVM. Sebbene BERT sia un modello potente, SVM non è ottimizzato per lavorare con embedding così densi e ricchi. Le rappresentazioni di BERT richiedono modelli più avanzati, come le reti neurali profonde, in grado di sfruttare appieno la complessità semantica delle embedding.

3.5.3 - Prestazioni di Regressione Logistica La Regressione Logistica come il Support Vector Machine ha ottenuto performance perlopiù stabili con tutte le tipologie di codifiche, anche lui come il precedente modello ha ottenuto performance migliore con codifica TF-IDF.

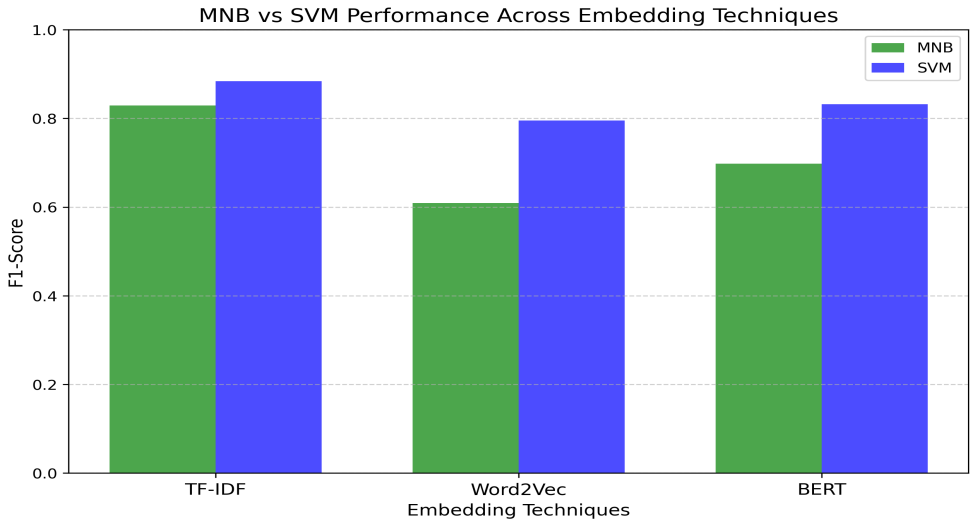


LR - TF-IDF TF-IDF funziona bene con Logistic Regression perché le rappresentazioni sparse e ad alta dimensionalità si adattano efficacemente alla natura lineare del modello. La Logistic Regression, come le SVM, è progettata per gestire dati sparsi, e la struttura di TF-IDF permette di separare le classi in modo ottimale. Inoltre, la semplicità del modello aiuta a interpretare meglio i risultati, rendendo TF-IDF un'opzione ideale quando si desidera un approccio semplice e veloce per la classificazione dei testi.

LR - Word2Vec Word2Vec , essendo una rappresentazione densa e continua, non cattura immediatamente i termini più discriminanti per la classificazione, rendendolo meno adatto a Logistic Regression . Anche se Word2Vec può catturare relazioni semantiche più profonde tra le parole, la sua struttura densa e continua non si adatta facilmente alla struttura lineare di Logistic Regression , che predilige rappresentazioni più sparse e meno complesse.

LR - BERT Le rappresentazioni generate da BERT sono altamente complesse e contestuali, rendendo difficile per Logistic Regression sfruttarle in modo ottimale. Anche se BERT è un modello potente, la sua alta dimensionalità e la natura ricca delle sue embedding richiedono modelli più complessi, come le reti neurali profonde, che sono in grado di gestire la complessità semantica delle rappresentazioni contestuali. La Logistic Regression , essendo un modello lineare, non è in grado di gestire adeguatamente la complessità delle embedding di BERT .

3.5.4 - Confronto MNB-SVM Dal confronto diretto tra i due modelli, emerge chiaramente che il modello SVM ottiene risultati significativamente migliori con qualsiasi tipo di codifica dei dati testuali rispetto a MNB.



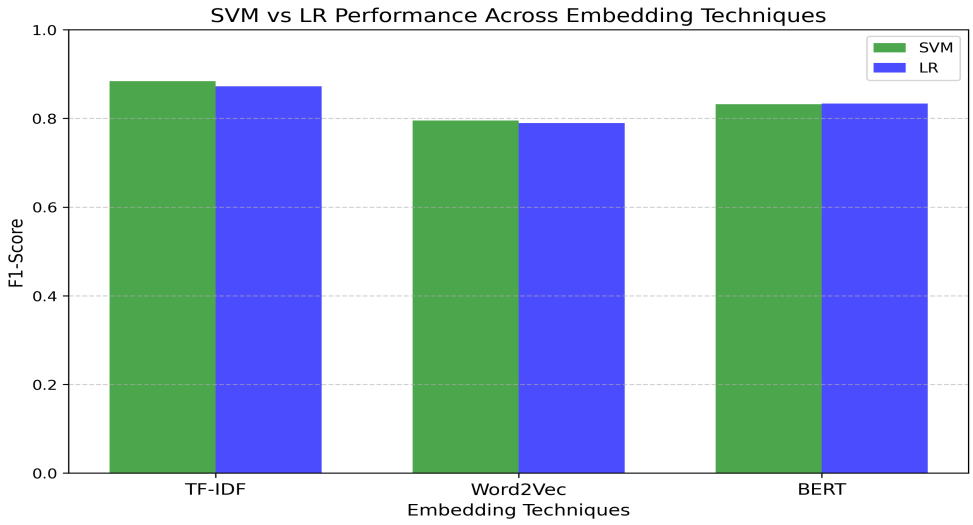
In particolare, nei casi migliori la F1-score di SVM è 0.87, mentre quella di Naive Bayes è 0.83, a favore di SVM. Questo risultato evidenzia la capacità di SVM di gestire spazi di caratteristiche ad alta dimensione in modo più efficace, permettendo una separazione più precisa delle classi. Inoltre, SVM riesce a catturare relazioni complesse tra le caratteristiche, anche quando queste non sono linearmente separabili, rendendolo più adatto a compiti di classificazione di testo dove le interazioni tra le parole e le classi sono più articolate.

Al contrario, Naive Bayes , pur essendo un modello semplice e veloce, tende a soffrire quando le assunzioni di indipendenza tra le caratteristiche non sono valide, come avviene frequentemente nelle rappresentazioni più complesse dei dati testuali. Infine la sua capacità di generalizzare sui dati di test può essere limitata quando si lavora con feature ad alta dimensione

Nel nostro contesto, dove sia la precisione che la velocità sono fattori cruciali, la scelta tra SVM e MNB dipende da come bilanciare questi due aspetti. Se l'accuratezza è la priorità assoluta e non ci sono problemi significativi con il tempo di risposta, SVM è la scelta migliore grazie alla sua capacità di separare in modo più preciso le classi e di gestire relazioni complesse tra le caratteristiche.

Tuttavia, considerando che il costo computazionale è altrettanto importante nel nostro task, e visto che la differenza di precisione tra i due modelli è relativamente piccola (0.04), potrebbe essere più conveniente optare per Naive Bayes. Questo modello, infatti, offre una performance decente con tempi di calcolo più rapidi e una minore complessità computazionale. Se il margine di differenza in termini di precisione è accettabile, Naive Bayes si rivela una scelta vantaggiosa quando la velocità di esecuzione è fondamentale.

3.5.5 - Confronto SVM-LR Analizzando il grafico, possiamo osservare che le performance dei due modelli sono estremamente simili. In effetti, SVM risulta vincente in questa comparazione con uno scarto di circa due centesimi nei primi due embedding. Tuttavia, nel terzo embedding, SVM ottiene un risultato leggermente inferiore rispetto a Logistic Regression , con una differenza di pochi millesimi.



In questo contesto, proclamare un vincitore tra SVM e Regressione Logistica è complesso. Di primo impatto, potrebbe sembrare che SVM sia il modello migliore, grazie alla sua capacità di ottenere performance superiori in termini di precisione. Tuttavia, se si considera anche il costo computazionale , la situazione cambia notevolmente.

Infatti, SVM risulta essere un modello significativamente più complesso rispetto alla Regressione Logistica, non solo dal punto di vista algoritmico ma anche computazionale. L'addestramento di SVM richiede una maggiore quantità di risorse e tempo , specialmente con dataset di grandi dimensioni e quando le rappresentazioni dei dati sono ad alta dimensionalità. In un contesto come il nostro, dove la velocità di risposta del modello è cruciale, ogni secondo di tempo di elaborazione è prezioso.

Va anche sottolineato che la Regressione Logistica è un modello che presuppone relazioni lineari tra le caratteristiche, il che la rende meno adatta per problemi dove le interazioni complesse tra le variabili sono decisive. Tuttavia, nel nostro caso, il distacco tra i due modelli è così piccolo che può essere considerato trascurabile in molti casi. Pertanto, risulta vantaggioso sacrificare quel piccolo margine di precisione per ridurre la complessità computazionale. La Regressione Logistica , essendo un modello più semplice e meno costoso dal punto di vista computazionale, permette di ottenere risposte più rapide senza compromettere in modo significativo la qualità delle previsioni.

3.6 | Deployment

Dall'analisi delle prestazioni dei modelli, è emerso sin da subito un problema significativo di dipendenza dai dati di addestramento. Più precisamente, i dataset utilizzati per il training e il testing non si sono dimostrati sufficientemente generalizzati, generando risultati eccessivamente ottimistici, noti come overperformance. Questa criticità ha influenzato l'affidabilità delle metriche ottenute, che riflettevano più la capacità dei modelli di adattarsi ai dati forniti che non una reale capacità di generalizzazione su dati nuovi o mai visti prima.

Nonostante questa limitazione, le considerazioni sui modelli e sulle loro prestazioni in relazione alle tecniche di em-

bedding utilizzate rimangono valide. Ogni modello ha mostrato caratteristiche distintive che, in termini di accuratezza e capacità predittiva, possono essere sfruttate efficacemente in contesti specifici. Tuttavia, per valutare il comportamento dei modelli in modo più realistico, abbiamo deciso di testarli manualmente introducendo dati di input in maniera dinamica. Questo approccio, sebbene soggettivo, ci ha permesso di osservare le prestazioni dei modelli in un contesto applicativo reale.

Dai test condotti manualmente, è emerso che il modello basato su Naive Bayes si è distinto per la sua capacità di effettuare predizioni accurate con tempi di risposta accettabili, anche in presenza di richieste multiple. La semplicità computazionale del modello e il suo adattamento ai dati testuali lo hanno reso una scelta particolarmente valida, soprattutto in contesti in cui la rapidità delle risposte è fondamentale. Tuttavia, è importante sottolineare che questa osservazione è puramente empirica e potrebbe non riflettere una valutazione oggettiva o generalizzabile.

Nonostante le buone prestazioni di Naive Bayes, non escludiamo la possibilità di un riadattamento dei modelli. Questo potrebbe prevedere l'addestramento su dataset specificamente progettati per la sentiment analysis, caratterizzati da un migliore bilanciamento e qualità delle etichette. Tale approccio consentirebbe di ridurre il rischio di overfitting, migliorando la capacità dei modelli di generalizzare su dati nuovi. In questa prospettiva, il riaddestramento non rappresenterebbe solo un miglioramento metodologico, ma un passaggio fondamentale per garantire l'affidabilità dell'applicazione finale.

Infine, indipendentemente dal modello selezionato per l'implementazione finale, la scelta dovrà tenere conto sia delle prestazioni predittive che dei requisiti di efficienza computazionale, con l'obiettivo di garantire un equilibrio tra accuratezza e velocità in contesti reali.

4 | IMPLEMENTED APPLICATION

L'applicazione che utilizzerà il modello di sentiment analysis sarà strutturata principalmente in due sezioni: una dedicata agli utenti (studenti) e l'altra riservata alle istituzioni o ai singoli docenti. Gli studenti, utilizzando un codice fornito loro dai docenti, potranno accedere a semplici form in cui sarà possibile inserire commenti su un tema stabilito dall'insegnante. Una volta aggiunti i commenti desiderati, potranno tornare alla homepage ed eventualmente accedere ad altri form disponibili. I docenti, invece, avranno la possibilità di visualizzare informazioni generali relative ai form creati e ai commenti ricevuti, creare nuovi form specificando un titolo e una descrizione/tema, e infine accedere a una visualizzazione dettagliata delle informazioni relative a ciascun form.

Tecnologie utilizzate Per la realizzazione dell'applicazione, abbiamo deciso di utilizzare tecnologie moderne che garantissero un elevato riuso e personalizzazione dei componenti, oltre a consentire una semplice integrazione con quelli già esistenti. Segue un riassunto delle tenologie usate per frontend e backend

- Frontend: È stata adottata una combinazione di Vue, TypeScript e TailwindCSS, poiché queste tecnologie risultano già familiari al nostro team e rispondono pienamente alle esigenze progettuali. I vari moduli frontend interagiscono con il backend tramite chiamate fetch che sfruttano il protocollo HTTP, mentre per la gestione della navigazione interna è stato utilizzato Vue Router.

- Backend: Per il backend, abbiamo ritenuto che l'implementazione di un semplice modulo FastAPI fosse sufficiente a soddisfare le funzionalità richieste dal frontend. Non essendo necessario gestire dati particolarmente variegati o complessi, abbiamo deciso di non adottare un servizio di database dedicato. Al suo posto, i dati vengono memorizzati e gestiti utilizzando file JSON, replicando la logica dei database non relazionali.

Data la configurazione altamente flessibile, ma poco robusta, non ci siamo concentrati su aspetti come il multithreading per gestire accessi simultanei o sull'integrazione di sistemi di autenticazione e registrazione. Questi elementi, tuttavia, rappresentano spunti per possibili miglioramenti futuri. Inoltre, è stata predisposta un'istanza specifica del modello di machine learning (ML) selezionato in precedenza, salvando i binari e le informazioni necessarie per il suo funzionamento.

Flusso operativo Il flusso delle azioni più comuni che un utente può compiere sulla piattaforma è il seguente:

1. Creazione del form: Il docente crea un nuovo form, inserendo un titolo e una descrizione/tema. Una volta creato, viene generato un codice univoco che il docente condivide con gli studenti.
2. Accesso degli studenti: Gli studenti accedono al form utilizzando il codice fornito dal docente e inviano uno o più commenti sul tema specificato e poi tornano alla homepage.
3. Elaborazione dei commenti: I commenti vengono elaborati dal modello di sentiment analysis, memorizzati nel sistema ed inviate alle varie interfaccie frontend dei docenti.
4. Aggiornamento delle informazioni: Le informazioni relative al singolo form e quelle generali vengono automaticamente aggiornate e rese disponibili al docente.
5. Feedback ai studenti: In base ai risultati dell'analisi dei sentimenti estratti dai commenti, i docenti possono fornire un feedback agli studenti, completando così il ciclo di interazione.

5 | FINAL CONSIDERATIONS

5.1 | General note

Nonostante il suo potenziale, la sentiment analysis deve affrontare diverse sfide. Una delle più rilevanti è rappresentata dall'innata complessità del linguaggio umano. Elementi come il sarcasmo, l'ironia e le sfumature culturali sfuggono spesso a una classificazione diretta, creando difficoltà anche per gli algoritmi più avanzati. Inoltre, l'ambiguità di alcune parole e la natura in continua evoluzione del linguaggio richiedono aggiornamenti e perfezionamenti costanti nei modelli di sentiment analysis. Un'ulteriore sfida è data dalla necessità di elaborare dati multilingue, il che implica l'uso di strumenti in grado di operare efficacemente in contesti linguistici diversi.

Negli ultimi anni, i progressi nel machine learning e nel deep learning hanno significativamente migliorato le capacità della sentiment analysis. Le reti neurali, in particolare le recurrent neural networks (RNNs) e i transformers, hanno permesso ai modelli di catturare schemi complessi nel linguaggio e di aumentare la precisione nella classificazione dei sentiment. Inoltre, i modelli linguistici pre-addestrati, come BERT (Bidirectional Encoder Representations from Transformers), hanno rivoluzionato il settore fornendo rappresentazioni testuali robuste che possono essere ottimizzate per compiti specifici di sentiment analysis.

5.2 | Model improvements

Per quanto riguarda i modelli da noi selezionati, addestrati e valutati, esistono diversi elementi che ampliano i margini di miglioramento. In primo luogo, la raccolta e l'utilizzo di dataset più ampi, accuratamente processati e annotati, potrebbe incrementare il livello di generalizzazione e ridurre il bias presente nei modelli. Questo approccio contribuirebbe anche a mitigare l'errore introdotto dalle scelte progettuali adottate, come l'utilizzo dello stesso modello per classificare i dati utilizzati nell'addestramento, una pratica che può compromettere l'accuratezza delle valutazioni.

Un altro aspetto da considerare riguarda i modelli stessi. Sarebbe possibile addestrare e testare reti neurali più complesse rispetto a quelle utilizzate, oppure esplorare tecniche avanzate come il fine-tuning di modelli pre-addestrati (foundation models) per adattarli alla sentiment analysis. Inoltre, l'adozione di approcci ensemble, che combinano metodologie rule-based con modelli di ML e DL, potrebbe migliorare ulteriormente la precisione.

Approcci che includano livelli di granularità più elevati rappresentano un altro potenziale vantaggio, nonostante le complessità che comportano. Lo sviluppo di modelli specifici per il rilevamento del sarcasmo o dell'ironia — contesti in cui anche i modelli più avanzati spesso faticano — risulterebbe particolarmente utile. Non meno importante sarebbe la creazione di modelli multilingua, capaci di operare in contesti linguistici diversi, o modelli multimodali, che integrino informazioni provenienti da diverse fonti, come il tono vocale o le espressioni facciali, per aumentare l'efficacia della sentiment analysis nel mondo reale.

5.3 | Application improvements

Anche l'applicazione web sviluppata presenta margini di miglioramento significativi. Ad esempio, l'esperienza utente potrebbe essere ampliata, includendo sezioni interattive che spieghino il funzionamento dei modelli di sentiment analysis, oppure offrendo strumenti ludici che mostrino il loro impatto nel mondo reale. Per la sezione dedicata alle istituzioni, si potrebbero integrare statistiche più dettagliate, infografiche chiare e la possibilità di creare form più articolati, rendendo l'interfaccia più informativa e intuitiva.

6 | CONCLUSION

In conclusione, la sentiment analysis è uno strumento potente per interpretare le dimensioni emotive e soggettive del testo. Sfruttando tecniche computazionali per analizzare il linguaggio, offre preziose intuizioni sugli atteggiamenti e le opinioni umane, diventando indispensabile in numerosi ambiti, dal business alla politica, fino all'accademia e oltre. Con l'evolversi del settore, la sentiment analysis promette di approfondire la nostra comprensione della comunicazione umana e di migliorare la nostra capacità di prendere decisioni basate sui dati in un mondo sempre più complesso.