

UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Scuola di Economia e Statistica

Corso di laurea Magistrale in

BIOSTATISTICA



**Applicazioni NLP e Machine Learning per la valutazione
della gravità dei pazienti e la previsione di mortalità.
Un'applicazione alle Schede di Dimissione Ospedaliera.**

Relatore: Prof. Paolo Berta

Correlatore: Prof. Lorenzo Malandri

Tesi di Laurea di:

Andrea Millone

Matr. N. 846588

Anno Accademico 2024/2025

Indice

1. Abstract	5
2. Introduzione	7
3. Metodi	9
3.1 Fonti e Descrizione dei Dataset	9
3.1.1 Dataset delle SDO italiane	10
3.1.2 Dataset dell'ICD-9-CM	12
3.2 Database Relazionali	12
3.3 Pre-processing e Feature engineering	13
3.3.1 Dati Strutturati	13
3.3.2 Dati non Strutturati	14
4. Analisi statistica	15
4.1 Machine Learning (ML) nella Sanità	15
4.1.1 Random Forest	17
4.1.2 XGBoost	19
4.2 Natural Language Processing (NLP) nella Sanità	23
4.2.1 Rappresentazione delle Parole e Valore Semantico	24
4.2.2 One Hot Encoding	25
4.2.3 Bag of Words (BoW)	25
4.2.4 Term Frequency-Inverse Document Frequency (TF-IDF)	26
4.2.5 N-Gram e Curse of Dimensionality	26
4.2.6 Word Embedding	27
4.2.7 Reti Neurali Artificiali (ANN) e Deep Learning (DL)	29
4.2.8 BiLSTM	34
4.2.9 Il modello Transformer	35
4.2.10 BERT	39
4.2.11 Bio_Discharge_Summary_BERT	41
4.3 Ottimizzazione e Regolarizzazione	42
4.3.1 Split in Training, Test e Validation	42
4.3.2 Feature Selection (LASSO)	44
4.3.3 K-Fold Cross-Validation	45
4.3.4 Parametri di Addestramento:	45
4.3.5 Epoche	45

4.3.6 Learning Rate	46
4.3.7 Ottimizzatori: Adam, AdamW	46
4.3.8 Tecniche di Regularization: Dropout, Batch Normalization	48
4.3.9 Funzione di Costo: Binary Cross-Entropy	48
4.3.10 Bayesian Optimization	49
4.4 Valutazione e Interpretazione dei Modelli	50
4.4.1 Confusion Matrix e Metriche di Valutazione	50
4.4.2 Analisi Grafica: ROC, PRC, Learning Curves	52
4.4.3 Explainable Artificial Intelligence (XAI)	54
4.4.4 LIME (Local Interpretable Model-Agnostic Explanations)	56
5. Risultati	58
5.1 Ambiente Cloud, Strumenti e Librerie Utilizzate	58
5.2 Analisi Descrittiva	60
5.2.1 caratteristiche del campione	60
5.3 Performance dei Modelli	67
5.3.1 Primo Approccio: Random Forest e XGBoost	67
5.3.2 Secondo Approccio: Random Forest (TF-IDF) e XGBoost (BoW)	75
5.3.3 Terzo Approccio: Bio_Discharge_Summary_BERT	82
5.3.4 Quarto Approccio: Modello Ibrido	85
5.4 Interpretazione degli Esiti tramite XAI	90
5.5 Interpretazione dei Risultati	101
5.6 Punti di Forza	102
5.7 Limitazioni dello Studio	103
6. Discussione	104
7. Bibliografia	106

1. Abstract

Negli ultimi anni è emersa l'importanza dei dati "reali", come quelli provenienti dalle Cartelle Cliniche Elettroniche (EHR). L'estrazione di conoscenza da questi Big Data rappresenta una sfida significativa in sanità pubblica per migliorare la qualità delle decisioni cliniche, sviluppare terapie personalizzate, ridurre i costi sanitari e prevedere gli esiti clinici. L'uso di nuovi metodi di apprendimento automatico come il Natural Language Processing (NLP) sta crescendo rapidamente e si è dimostrato uno strumento indispensabile, per analizzare e sintetizzare informazioni utili dai dati testuali non strutturati, come le descrizioni delle diagnosi.

Obiettivi: Gli obiettivi di questo studio sono: (i) sviluppare e confrontare differenti algoritmi di Machine Learning (ML), Deep Learning (DL) ed Natural Language Processing (NLP) per la previsione della mortalità ospedaliera utilizzando i dati provenienti dalle Schede di Dimissione Ospedaliera (SDO) italiane; (ii) valutare l'interpretabilità dei modelli attraverso tecniche di Explainable AI (XAI) che rendono più trasparenti le decisioni degli algoritmi e la comprensione del processo di classificazione.

Analisi dei dati: Nel presente studio è stata condotta un'analisi secondaria dei dati raccolti dalle SDO italiane, relative al periodo 2012–2016 e comprendenti circa 3,5 milioni di pazienti. Il dataset utilizzato risulta composto da dati strutturati (estratti dalle SDO) e da dati non strutturati (descrizioni delle diagnosi ICD-9-CM). Tra i diversi modelli analizzati nello studio (tecniche classiche e innovative di ML e NLP applicate sui dati strutturati e/o sui dati non strutturati) l'approccio ibrido, che combina dati strutturati e non strutturati, ha dimostrato prestazioni superiori (Accuracy= 97%, con una media macro degli F1-score=78% e ROC(AUC)=0.96 Average Precision=0.69), rispetto ai modelli che utilizzano solo una tipologia di dati.

Conclusioni: L'approccio sperimentale attraverso l'uso di tecniche classiche e innovative di ML e NLP ha dimostrato che l'approccio ibrido che combina dati strutturati e non strutturati garantisce le performance migliori nel predire il rischio clinico. Questo lavoro oltre ad aumentare la fiducia nell'intelligenza artificiale in ambito medico, fornisce un contributo significativo per migliorare l'accuratezza e la trasparenza dei sistemi di intelligenza artificiale applicati alla medicina clinica. I risultati di questo studio forniscono indicazioni strategiche per un uso ottimizzato dei Big Data sanitari, con importanti ricadute sulla personalizzazione delle terapie e sull'efficienza nella gestione del sistema sanitario.

2. Introduzione

Tradizionalmente, la ricerca medica si basava su studi controllati con protocolli rigorosi; tuttavia, negli ultimi anni è emersa l'importanza dei dati "Real-world", come quelli provenienti dalle cartelle cliniche elettroniche (EHR, Electronic Health Records). L'estrazione di conoscenza da questi Big Data, caratterizzati dalle 3V – volume (dati nell'ordine dei terabyte o superiori), velocità (generazione continua e ad alta frequenza) e varietà (diversità di formati e fonti) – rappresentano una sfida rilevante, ma offrono anche opportunità senza precedenti per migliorare la qualità delle decisioni cliniche, sviluppare terapie personalizzate, contenere i costi sanitari e prevedere in maniera più accurata gli esiti clinici.

L'uso dei Big Data è particolarmente cruciale in sanità pubblica, dove consente di identificare con maggiore precisione le popolazioni a rischio, monitorare l'evoluzione delle malattie e pianificare interventi di prevenzione. Tuttavia, l'eterogeneità e la natura non strutturata di molti di questi dati richiedono avanzate tecnologie per l'analisi. In questo contesto, il Natural Language Processing (NLP) si è dimostrato uno strumento indispensabile, consentendo di analizzare e sintetizzare informazioni utili dai dati testuali non strutturati, come le descrizioni delle diagnosi, rendendo accessibili informazioni cruciali per il progresso medico-scientifico e per il miglioramento dell'assistenza sanitaria (1,2).

L'Italia, con i suoi 58,9 milioni di abitanti, si configura come uno dei principali paesi europei per la disponibilità di dati amministrativi destinati alla ricerca. In particolare, il database delle Schede di Dimissione Ospedaliera (SDO) rappresenta una risorsa fondamentale, in quanto raccoglie sistematicamente informazioni relative a tutti gli eventi di ospedalizzazione, sia nel settore pubblico che in quello privato, rimborsati dal Servizio Sanitario Nazionale (SSN).

Nel contesto delle Schede di Dimissione Ospedaliera (SDO) italiane, i dati raccolti rappresentano un'importante opportunità per analizzare e comprendere meglio la mortalità ospedaliera. Le SDO italiane contengono sia informazioni strutturate sia non strutturate, l'integrazione di queste due tipologie di dati potrebbe permettere di sviluppare modelli predittivi più accurati, utili per supportare decisioni cliniche personalizzate. Tuttavia, il successo di tali analisi dipende fortemente dalla qualità e dalla preparazione dei dati, evidenziando la necessità di un accurato pre-processing per gestire la complessità dei dataset eterogenei. Le tecnologie avanzate di Intelligenza Artificiale (IA), in particolare il Machine Learning (ML) e il Deep Learning (DL), offrono strumenti potenti per analizzare i dati clinici (5, 6).

Metodi come le reti neurali convoluzionali (CNN) e le reti ricorrenti (LSTM) sono in grado di identificare pattern complessi e migliorare sia la capacità predittiva che quella diagnostica. L'impiego di modelli pre-addestrati, come ClinicalBERT e MedBERT, consente inoltre di utilizzare rappresentazioni linguistiche sofisticate per interpretare con precisione il linguaggio medico (7,8,9). Questi approcci permettono di analizzare grandi volumi di dati e di sviluppare previsioni personalizzate, adattate alle specificità di ciascun paziente. Per implementare queste tecnologie su larga scala, sono necessarie infrastrutture digitali adeguate, una standardizzazione dei dati e una regolamentazione chiara per garantire la sicurezza e la privacy delle informazioni sensibili. L'integrazione di tecniche avanzate di ML e NLP con i dati delle SDO italiane rappresenta dunque un'importante opportunità per migliorare le previsioni cliniche e ottimizzare la gestione sanitaria, sfruttando al massimo il potenziale combinato di dati strutturati e non strutturati.

Gli obiettivi di questo studio sono: (i) valutare l'impatto dell'integrazione di dati strutturati e non strutturati nell'analisi predittiva della mortalità ospedaliera, utilizzando i dati provenienti dalle SDO italiane. (ii) sviluppare e confrontare differenti algoritmi di ML e DL per la previsione della mortalità ospedaliera. A tal fine, verranno utilizzate misure derivate da dati strutturati (ad esempio, età, sesso, durata della degenza, ecc.), dal metodo Elixhauser basato su dati amministrativi ICD-9-CM (10, 11,12) e da dati testuali non strutturati (descrizioni delle diagnosi ICD-9-CM). Le prestazioni dei modelli saranno valutate secondo criteri quali parsimonia, capacità discriminante e calibrazione. (iii) valutare l'interpretabilità dei modelli attraverso tecniche di Explainable AI (XAI), che consentiranno di rendere più trasparenti le decisioni degli algoritmi, facilitando la comprensione delle caratteristiche più rilevanti nel processo di classificazione. Strumenti come LIME (Local Interpretable Model-agnostic Explanations) verranno impiegati per analizzare il contributo delle diverse variabili e garantire che le previsioni dei modelli siano interpretabili dai clinici. (13)

Il presente lavoro si propone di rafforzare la fiducia nell'applicazione dell'intelligenza artificiale in ambito medico, contribuendo in maniera significativa all'incremento dell'accuratezza e della trasparenza nelle previsioni degli esiti clinici. I risultati ottenuti offrono prospettive strategiche per l'ottimizzazione dell'uso dei Big Data sanitari, con rilevanti implicazioni per la personalizzazione delle terapie e il miglioramento dell'efficienza nella gestione del sistema sanitario.

3. Metodi

3.1 Fonti e Descrizione dei Dataset

Nel presente studio abbiamo condotto un'analisi secondaria dei dati estratti dal database Scheda di Dimissione Ospedaliera (SDO) dell'Italia (14,15,16,17), relative al periodo 2012–2016 e comprendenti circa 3,5 milioni di ricoveri provenienti da tutte le regioni del Paese.

Le SDO rappresentano una sintesi delle informazioni cliniche contenute nella cartella clinica e assumono una rilevanza medico-legale e giuridica. Nello specifico le SDO includono informazioni personali della cartella clinica del paziente (esso, data e luogo di nascita, comune di residenza), informazioni cliniche (diagnosi, procedure chirurgiche e diagnostiche, informazioni su ricovero e dimissione) e informazioni relative alla struttura (ad esempio, regione ospedaliera, tipo di attività, tipo di ospedalizzazione) in cui è stato effettuato il ricovero.

L'Italia è un paese eterogeneo con disparità geografiche tra regioni del Nord e del Sud, con il Nord che è un'area con migliori indicatori socioeconomici e una rete più sviluppata di operatori sanitari. Dal 2006 è stato adottato per la classificazione delle malattie e delle procedure, la *Classificazione internazionale delle malattie, nona revisione, modifica clinica ICD-9-CM* aggiornato regolarmente per rispondere alle esigenze cliniche e organizzative (3,4).

Le Linee Guida per la codifica e la gestione del flusso SDO stabiliscono regole dettagliate volte a minimizzare errori e a garantire l'affidabilità dei dati, elementi essenziali per studi epidemiologici e la pianificazione sanitaria. Inoltre, il D.M. del 26 settembre 2023 ha introdotto nuove specifiche tecniche per il flusso SDO, sottolineando l'importanza crescente di questo strumento nella gestione sanitaria e nella ricerca, anche grazie all'impiego di tecniche avanzate come il ML e il natural language processing per l'analisi e la previsione di esiti clinici, quali la mortalità ospedaliera. Questi documenti contengono sia informazioni strutturate (ad esempio, età, sesso, durata della degenza) sia non strutturate (descrizioni delle diagnosi secondo ICD-9-CM).

Il dataset utilizzato in questo studio è stato fornito dal prof. Berta.

3.1.1 Dataset delle SDO italiane

Il dataset iniziale risulta composto dalle variabili riportate in tabella 1.

Variabile	Descrizione	Classe
cod_reg	regione dove è situato l'ospedale	categoriale
cod_ist	codice identificativo dell'istituto	categoriale
reg_res	regione di residenza del paziente	categoriale
prov_res	provincia di residenza del paziente	categoriale
com_res	comune di residenza del paziente	categoriale
cittad	cittadinanza del paziente	categoriale
eta	età del soggetto al momento del ricovero	numerica
mod_dim	modalità di uscita dalla struttura ospedaliera	categoriale
Disc*	codice relativo alla disciplina ospedaliera*	categoriale
rep	codice relativo al reparto ospedaliero	categoriale
dpr	codice relativo alla diagnosi principale secondo il manuale ICD-9	categoriale
dsec1	prima diagnosi concomitante secondo il manuale ICD-9	categoriale
Dsec2	seconda diagnosi concomitante secondo il manuale ICD-9	categoriale
Dsec3	terza diagnosi concomitante secondo il manuale ICD-9	categoriale
Dsec4	quarta diagnosi concomitante secondo il manuale ICD-9	categoriale
Dsec5	quinta diagnosi concomitante secondo il manuale ICD-9	categoriale
Drg24	codice relativo alla prestazione medica	categoriale
codice	codice identificativo del paziente	categoriale
cosp	codice identificativo dell'ospedale	categoriale
id	codice identificativo del ricovero	categoriale
female	indica se il paziente è di sesso femminile	categoriale
edata_dim	data di dimissione del paziente dalla struttura sanitaria	yy/mm/dd
edata_ric	data di ricovero del paziente dalla struttura sanitaria	yy/mm/dd
los (length of stay)	durata di permanenza del paziente nella struttura sanitaria in giorni	numerica
anno_rif	anno del referto	numerica

*Le discipline ospedaliere nelle SDO sono suddivise in discipline Mediche, Chirurgiche, Specialiste, di Emergenza-Urgenza, e Riabilitative.

Tabella 1 elenco variabili dataset delle SDO

Al dataset iniziale delle SDO sono state poi implementate 31 variabili, riportate in tabella 2, derivate dal metodo Elixhauser (10,11,12) sulla base dei codici ICD-9-CM. La presenza o l'assenza di ciascuna delle 31 patologie è stata estratta dalle informazioni relative alla diagnosi principale e alle diagnosi secondarie presenti nelle SDO e/o da altre informazioni apprese da ricoveri successivi. Successivamente, è stata creata tramite l'algoritmo Elixhauser, la variabile numerica Elixsum, che rappresenta la somma totale delle presenze (o assenze) delle 31 patologie e può assumere un punteggio che varia da 0 a 31.

Variabile	Descrizione	Classe
ynel1	presenza o assenza della patologia insufficienza cardiaca congestizia	categoriale
ynel2	presenza o assenza della patologia aritmie cardiache	categoriale
ynel3	presenza o assenza della patologia malattia vascolare	categoriale
ynel4	presenza o assenza della patologia disturbi della circolazione polmonare	categoriale
ynel5	presenza o assenza della patologia disturbi vascolari periferici	categoriale
ynel6	presenza o assenza della patologia ipertensione non complicata	categoriale
ynel7	presenza o assenza della patologia paralisi	categoriale
ynel8	presenza o assenza della patologia altri disturbi neurologici	categoriale
ynel9	presenza o assenza della patologia malattia polmonare cronica	categoriale
ynel10	presenza o assenza della patologia diabete non complicato	categoriale
ynel11	presenza o assenza della patologia diabete complicato	categoriale
ynel12	presenza o assenza della patologia ipotiroidismo	categoriale
ynel13	presenza o assenza della patologia insufficienza renale	categoriale
ynel14	presenza o assenza della patologia malattia del fegato	categoriale
ynel15	presenza o assenza della patologia ulcera peptica esclusa emorragia	categoriale
ynel16	presenza o assenza della patologia AIDS/HIV	categoriale
ynel17	presenza o assenza della patologia linfoma	categoriale
ynel18	presenza o assenza della patologia cancro metastatico	categoriale
ynel19	presenza o assenza della patologia tumore solido senza metastasi	categoriale
ynel20	presenza o assenza della patologia artrite reumatoide/collagene vascolare	categoriale
ynel21	presenza o assenza della patologia coagulopatia	categoriale
ynel22	presenza o assenza della patologia obesità	categoriale
ynel23	presenza o assenza della patologia perdita di peso	categoriale
ynel24	presenza o assenza della patologia disturbi dei fluidi e degli elettroliti	categoriale
ynel25	presenza o assenza della patologia anemia da perdita di sangue	categoriale
ynel26	presenza o assenza della patologia anemia da carenza	categoriale
ynel27	presenza o assenza della patologia abuso di alcol	categoriale
ynel29	presenza o assenza della patologia abuso di droga	categoriale
ynel29	presenza o assenza della patologia psicosi	categoriale
ynel30	presenza o assenza della patologia depressione	categoriale
ynel31	presenza o assenza della patologia ipertensione complicata	categoriale
elixsum	valore indice Elixhauser	numerica

Tabella 2 elenco variabili delle patologie e dell'indice Elixhauser

3.1.2 Dataset dell'ICD-9-CM

Il secondo dataset utilizzato in questo studio (Tabella 3) contiene le descrizioni testuali delle diagnosi corrispondenti ai codici della *Classificazione internazionale delle malattie, nona revisione, modifica clinica ICD-9-CM* (3):

Variabile	Descrizione	Classe
dpr	codice relativo alla diagnosi secondo il manuale ICD-9-CM	categoriale
LONG Description	descrizione testuale della diagnosi nel formato lungo	stringa
SHORT Description	descrizione testuale della diagnosi nel formato corto, con abbreviazioni e acronimi	stringa

Tabella 3 elenco variabili del dataset dell'ICD-9-CM

3.2 Database Relazionali

Per sfruttare appieno le informazioni testuali relative alla diagnosi, è stata effettuata una left join, che ha integrato due dataset, mantenendo inalterata la struttura del dataset principale. Questa operazione ha consentito di arricchire le Schede di Dimissione Ospedaliera con informazioni testuali dettagliate sulle diagnosi. In particolare, il primo dataset conteneva per ciascun soggetto una variabile categorica denominata dpr, che identificava il codice della diagnosi clinica principale, mentre il secondo dataset forniva invece la corrispondenza tra i codici diagnostici e la descrizione testuale della diagnosi, secondo il manuale ICD-9-CM.

L'integrazione basata sulle chiavi diagnostiche, ha permesso di associare a ciascun codice la relativa descrizione testuale. Questo approccio ha permesso di unire dati quantitativi (codici) e qualitativi (descrizioni). Inoltre, grazie a questa procedura, è stato possibile recuperare sia la descrizione delle diagnosi principali sia quella delle diagnosi secondarie, ampliando così il quadro informativo disponibile per le analisi successive.

3.3 Pre-processing e Feature engineering

Il Pre-processing rappresenta una fase essenziale e preliminare all’analisi dei dati, in quanto mira a ottimizzare e standardizzare le informazioni per garantire la robustezza e l’affidabilità dei modelli di classificazione di ML e NLP. Soprattutto nel contesto del NLP, la pre-elaborazione del testo è fondamentale, poiché influisce significativamente sull’efficacia dei modelli predittivi.

La fase di pre-processing rappresenta un passaggio preliminare fondamentale, finalizzato a migliorare la qualità e la coerenza dei dati in ingresso mediante una serie di operazioni mirate. Dopo la tokenizzazione lessicale, che segmenta il testo in unità minime (token), si applicano trasformazioni specifiche per ottimizzare sia i dati strutturati sia quelli non strutturati. Questo approccio consente di ottenere dataset di elevata qualità, facilitando lo sviluppo di analisi più accurate e significative. Nel presente studio, il pre-processing è stato articolato in due fasi distinte, ciascuna dedicata a uno specifico tipo di dati, al fine di garantire un trattamento adeguato e mirato per ciascuna tipologia.

3.3.1 Dati Strutturati

Per i dati strutturati sono state implementate diverse operazioni di pre-processing e feature engineering, finalizzate a garantire la qualità del dataset e a ottimizzare le prestazioni del modello predittivo. In particolare, sono state eseguite le seguenti operazioni:

- Eliminazione dei valori missing nelle variabili: le osservazioni che presentavano valori mancanti sono state eliminate.
- Verifica ed eliminazioni di osservazioni duplicate.
- Verifica, ricodifica della tipologia delle variabili e creazione di nuove variabili: abbiamo controllato la corretta allocazione della classe della variabile tenendo conto della sua natura, in particolare, abbiamo effettuato la dummyzzazione (o one-hot-encoding) nelle variabili legate all’indice Elixhauser le ynel (1 per presenza della patologia e con 0 la sua l’assenza). Ai fini del nostro studio abbiamo creato la variabile target, Binarizzando la variabile modalità di dimissione, con 0 che indica che il soggetto è sopravvissuto e 1 che indica che il soggetto è deceduto.
- Eliminazioni di variabili non significative per l’analisi e ottimizzazione dei tempi di elaborazione.
- *Feature selection:* per evitare il sovrardimensionamento del dataset e ottimizzare l’efficienza computazionale, senza compromettere l’accuratezza del modello, è stato

selezionato un sottoinsieme di variabili significative. Questa operazione è stata realizzata mediante il metodo LASSO, che ha permesso di preservare la capacità di generalizzazione del modello.

3.3.2 Dati non Strutturati

Per i dati non strutturati sono state effettuate le seguenti operazioni di Pre-processing e Feature engineering:

- Creazione della variabile “Diagnosi”: le diagnosi principali e secondarie sono state concatenate in un'unica stringa. Questa procedura ha permesso di raggruppare tutte le informazioni diagnostiche in un unico campo, facilitando così l'analisi del quadro clinico globale del soggetto. Tale metodologia consente di evidenziare, in maniera sinergica, le interrelazioni tra le diverse patologie, offrendo una rappresentazione più accurata e approfondita dello stato sanitario del paziente durante il suo periodo di ricovero. contestualmente, è stata implementata una variabile (denominata “num_words”) che calcola il numero di parole presenti nella stringa "Diagnosi", fornendo così un ulteriore indicatore della complessità diagnostica.
- Conversione del testo in minuscolo (lowercasing): tecnica comunemente utilizzata, utile per ridurre la dimensionalità del vocabolario e le ambiguità legate alla capitalizzazione.
- Premesso che il testo delle diagnosi secondo il manuale ICD-9-CM è già un testo abbastanza standardizzato, si sono effettuate le operazioni di: Rimozione delle stopwords, rimozione di caratteri speciali, numeri e punteggiatura, lemmatizzazione per riportare le parole alla loro forma base.
- Generazione di n-grammi e bi-grammi; per arricchire la rappresentazione testuale. per i modelli del secondo approccio (BoW e TF-IDF).

4. Analisi Statistica

4.1 Machine Learning (ML) nella Sanità

L'impiego del ML nel settore sanitario sta trasformando radicalmente il funzionamento dei sistemi sanitari e l'erogazione delle cure (5). Grazie alla capacità dei modelli ML di identificare pattern complessi tra input e output a partire da enormi quantità di dati, è possibile ottenere supporto decisionale in tempo reale e personalizzare i percorsi di cura (Figura 1).

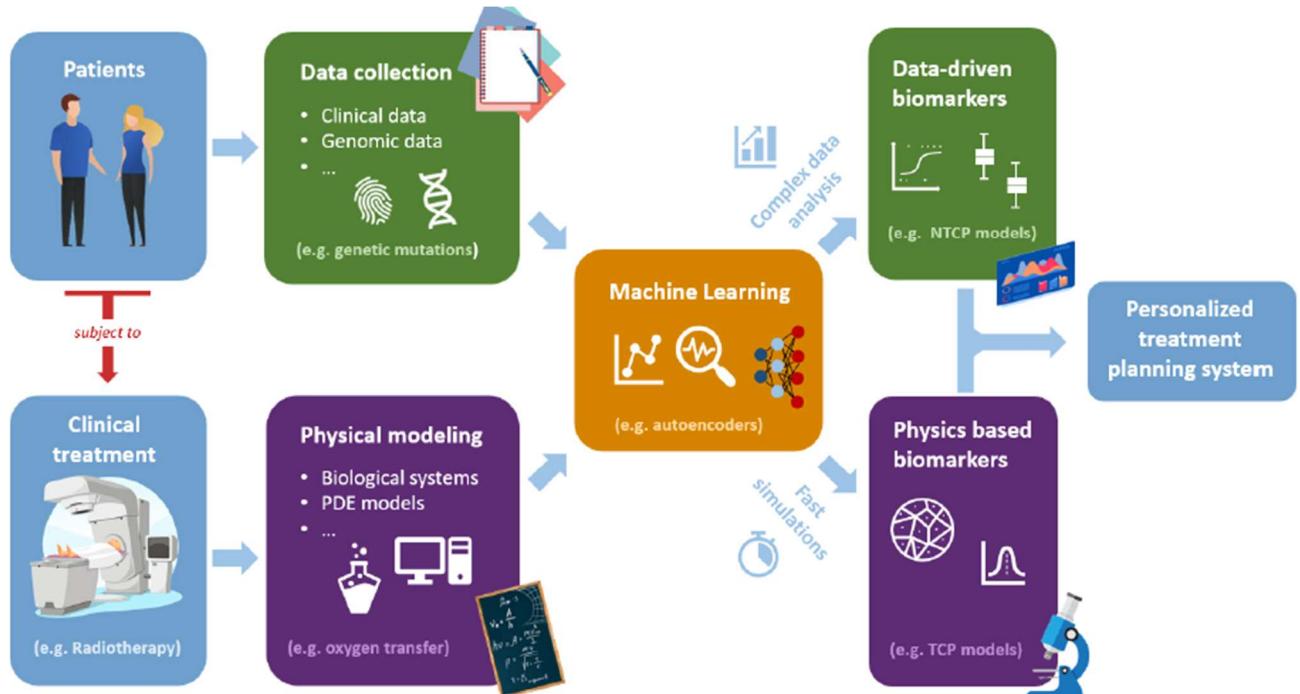


Figura 1 Esempio di quadro unificato per la medicina di precisione in cui gli algoritmi di apprendimento automatico consentono combinazione di approcci basati sui dati e sui modelli

Questo approccio genera un impatto positivo rilevante sia per i pazienti che per il personale medico, contribuendo a un miglioramento sostanziale dell'efficacia e della rapidità nelle decisioni cliniche.

Uno degli ambiti di applicazione più rilevanti del ML in Sanità riguarda l'organizzazione e l'ottimizzazione dei percorsi assistenziali. Sistemi basati su algoritmi di apprendimento automatico sono in grado di analizzare i dati clinici, le cartelle elettroniche e le informazioni provenienti da dispositivi indossabili per:

- Pianificare percorsi di cura personalizzate: In base alla storia clinica, agli esiti di test diagnostici e ad altri parametri rilevanti, gli algoritmi possono suggerire il percorso più

appropriato per ogni paziente, riducendo i tempi di attesa e migliorando la tempestività delle cure.

- Definire piani di trattamento ottimali: Attraverso l'analisi dei dati di outcome, il ML supporta i medici nella scelta delle terapie più efficaci, minimizzando il rischio di errori e permettendo un monitoraggio continuo dell'efficacia del trattamento.

Il ML, integrato con sistemi di intelligenza artificiale, fornisce ai medici strumenti avanzati per l'accesso e l'interpretazione di grandi quantità di dati:

- Diagnosi assistita: Algoritmi di classificazione e di riconoscimento delle immagini, addestrati su dataset di diagnosi precedenti, possono aiutare a individuare patologie in fase precoce, ad esempio tramite l'analisi automatizzata di immagini radiologiche o mammografiche.
- Predizione degli esiti clinici: Modelli predittivi, basati su tecniche di regressione o reti neurali, elaborano i dati storici dei pazienti per stimare il rischio di complicanze, facilitando decisioni cliniche che possono ridurre il tasso di mortalità e migliorare la qualità delle cure.
- Integrazione di fonti eterogenee di dati: I sistemi ML consentono di mettere in relazione dati provenienti da diverse fonti (studi clinici, letteratura medica, registrazioni elettroniche) per fornire ai medici informazioni aggiornate e pertinenti, aiutandoli a tenere il passo con le ultime innovazioni nel campo medico.

L'adozione del ML in medicina non ha l'obiettivo di sostituire il giudizio umano, ma di affiancarlo, automatizzando le attività routinarie e riducendo il carico di lavoro dei professionisti. Tra gli ambiti in cui l'automazione basata su ML si sta dimostrando particolarmente utile si evidenziano:

- Gestione delle cartelle cliniche: Sistemi di elaborazione del linguaggio naturale (NLP) permettono di estrarre informazioni rilevanti dalle cartelle elettroniche, semplificando l'accesso ai dati clinici e riducendo gli errori nella registrazione delle informazioni.
- Robotica assistita: I robot, integrati con algoritmi di ML, possono occuparsi di compiti ripetitivi o di monitoraggio continuo dei pazienti. Ad esempio, robot terapeutici sono impiegati per supportare i malati di Alzheimer, fornendo interazioni personalizzate e monitorando lo stato di salute in tempo reale.
- Prevenzione degli errori umani: L'analisi automatizzata dei dati consente di individuare anomalie e criticità prima che si traducano in errori clinici, contribuendo a ridurre gli effetti negativi legati all'affaticamento del personale sanitario.

L'adozione di soluzioni basate su ML ha un impatto positivo non solo sulla qualità delle cure, ma anche sull'efficienza organizzativa degli ospedali e delle strutture sanitarie. In particolare:

- Riduzione dei tempi di attesa: Ottimizzando la pianificazione delle visite e delle procedure diagnostiche, i sistemi ML permettono di gestire in modo più efficace le risorse, riducendo i tempi di attesa per le cure specialistiche.
- Diminuzione dei costi operativi: L'automazione delle attività amministrative e di monitoraggio, unitamente a diagnosi più rapide ed efficaci, consente una gestione più sostenibile delle risorse economiche e umane, diminuendo le spese mediche e i costi legati a errori di gestione.
- Miglioramento della qualità della vita dei pazienti: Grazie a percorsi assistenziali personalizzati e a un monitoraggio continuo, i pazienti ricevono cure più tempestive e mirate, riducendo la necessità di ricoveri prolungati e migliorando la qualità complessiva della vita.

Il ML rappresenta una svolta significativa per il settore medico-sanitario, non inteso a sostituire il ruolo del medico, ma a potenziarlo e a supportarlo in attività complesse e ripetitive. L'integrazione di algoritmi avanzati e di sistemi di automazione favorisce una maggiore precisione diagnostica, una personalizzazione dei trattamenti e una gestione più efficiente delle risorse, con un impatto diretto sulla riduzione dei costi e sul miglioramento della qualità della cura. In questo contesto, l'adozione del ML diventa un alleato strategico per affrontare le sfide attuali del sistema sanitario, garantendo un'assistenza medica sempre più efficace e orientata al benessere del paziente. In questo studio, per la classificazione basata su dati strutturati sono stati impiegati modelli di ML tradizionali, come Random Forest e XGBoost, che qui di seguito saranno illustrati.

4.1.1 Random Forest

Il modello Random Forest (RF) rappresenta uno degli algoritmi di apprendimento supervisionato più versatili e popolari nel campo del ML, grazie alle sue eccellenti prestazioni, scalabilità e capacità di gestire sia problemi di classificazione che di regressione. Si basa sul concetto di apprendimento ensemble, combinando i risultati di molteplici alberi decisionali per migliorare la robustezza e l'accuratezza delle previsioni. Alla base dell'algoritmo vi è la metodologia del *bagging* (bootstrap aggregating), che prevede la creazione di un insieme di alberi addestrati su campioni casuali con reinserimento (bootstrap) estratti dal dataset originale.

Ad ogni nodo decisionale, RF seleziona un sottoinsieme casuale di caratteristiche (*feature bagging*), riducendo la correlazione tra gli alberi e prevenendo il rischio di overfitting, tipico dei singoli decision tree.



Figura 2 Schema funzionamento del modello Random Forest

La procedura generale (Figura 2) per generare k alberi decisionali, dato un insieme D di addestramento di tuple, è la seguente:

- Campionamento Bootstrap: per ogni iterazione i ($con i = 1, 2, \dots, k$), un training set D_i di d tuple viene campionato con la sostituzione di D . Ciò significa che ogni D_i è un campione *bootstrap* di D e ne consegue che alcune tuple possano verificarsi più di una volta in D_i , mentre altre possano esserne escluse.
- Selezione degli attributi per la divisione: sia F il numero di attributi da utilizzare per determinare la divisione in ciascun nodo, dove F è molto più piccolo del numero di attributi disponibili; in ogni nodo, vengono selezionati casualmente F attributi da utilizzare come candidati per la suddivisione.
- Costruzione dell'albero: per costruire un classificatore dell'albero decisionale M_i si selezionano casualmente, in ogni nodo, gli attributi F come candidati per la divisione nel nodo.

Durante il processo, ogni albero cresce fino alla massima profondità possibile senza potatura, utilizzando metodi come CART (Classification and Regression Trees) per ottimizzare

le suddivisioni dei nodi. Il risultato finale della foresta è ottenuto attraverso un voto a maggioranza nel caso della classificazione o una media delle predizioni nel caso della regressione.

Dal momento che la previsione viene calcolata in base al voto della maggioranza degli alberi decisionali del modello, la funzione di discriminazione può essere definita come segue:

$$H(x) = \operatorname{argmax}_y \sum_{i=1}^k I(h_i(X, \theta_i) = Y)$$

Dove:

- I indica la funzione indicatore, che restituisce 1 se la condizione è vera e 0 altrimenti.
- h è la decisione del singolo albero, rappresenta la predizione del singolo albero i per l'input x , parametrizzato da θ_i .
- y è il valore della variabile di output.
- argmax_y indica il valore y quando viene massimizzato $\sum_{i=1}^k I(h_i(X, \theta_i) = Y)$.

Tra i principali vantaggi del modello vi sono la capacità di gestire dati categoriali e continui, la robustezza ai valori anomali e il rumore, nonché la flessibilità nell'uso di dati con valori mancanti. Tuttavia, RF può risultare meno efficiente in termini di memoria e tempo di calcolo su dataset di grandi dimensioni, soprattutto nella fase di previsione.

4.1.2 XGBoost

XGBoost (eXtreme Gradient Boosting) è un algoritmo di boosting basato su alberi decisionali, ottimizzato per ottenere elevate prestazioni in termini di accuratezza e velocità di addestramento. Il boosting è una tecnica di ensemble learning in cui modelli deboli (weak learners), solitamente alberi di decisione con profondità ridotta, vengono combinati iterativamente in modo che ogni nuovo modello corregga gli errori del precedente. Questa tecnica consente di migliorare la capacità predittiva riducendo errori di bias e varianza.

A differenza di altri metodi ensemble come il bagging (ad esempio, le Random Forest), in cui più modelli vengono addestrati indipendentemente su sottoinsiemi casuali del dataset, il boosting costruisce i modelli in sequenza. L'idea alla base del boosting è di dare più peso agli errori commessi in precedenza, in modo che il modello successivo si concentri su di essi per migliorarne la correzione (Figura 3).

Il processo di boosting segue questi passaggi:

- Si addestra un primo modello $f_1(x)$ su tutto il dataset.
- Si calcola l'errore delle predizioni e si assegnano pesi maggiori agli esempi classificati erroneamente.
- Il modello successivo $f_2(x)$ viene addestrato con una maggiore attenzione ai campioni con errore più elevato.
- Il processo continua iterativamente fino a ottenere un modello finale che combina tutti i modelli intermedi in una somma pesata.
- Il risultato è un predittore più robusto e accurato, capace di correggere progressivamente gli errori.

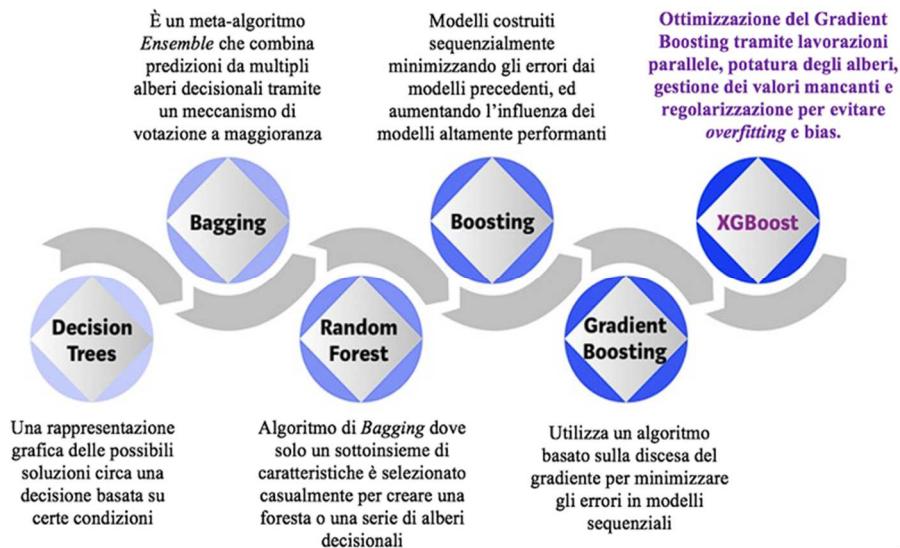


Figura 3 evoluzione degli alberi decisionali al XGBoost

Formalizzazione Matematica

Il modello finale è una combinazione lineare di tutti gli alberi deboli:

$$\hat{y}_i = \sum_{k=1}^K \alpha_k f_k(x_i)$$

Dove α_k sono i pesi associati a ciascun modello $f_k(x)$.

XGBoost implementa il boosting in maniera gradient-based, ovvero utilizza la discesa del gradiente per minimizzare una funzione di perdita.

Il modello viene appreso ottimizzando la seguente funzione obiettivo regolarizzata:

$$L(\phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

dove:

- $l(y_i, \hat{y}_i)$ è la funzione di perdita, che misura la differenza tra le previsioni \hat{y}_i e il valore reale y_i (ad esempio, per la regressione si usa la Loss quadratica $l(y, \hat{y}) = (y - \hat{y})^2$).
- $\Omega(f_k)$ è un termine di regolarizzazione che controlla la complessità del modello e previene l'overfitting.

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2$$

dove T è il numero di foglie nell'albero e ω rappresenta i pesi associati alle foglie. XGBoost utilizza la discesa del gradiente per aggiornare gli alberi a ogni iterazione, costruendo nuovi alberi $f_t(x)$ per correggere gli errori del modello precedente:

$$L^t = \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

dove:

- $g_i = \left(\frac{\delta l(y_i, \hat{y}_i)}{\delta \hat{y}_i} \right)$ è il gradiente della funzione di perdita rispetto alla predizione.
- $h_i = \frac{\delta l(y_i, \hat{y}_i)}{\delta \hat{y}_i^2}$ è l'hessiano (seconda derivata) della funzione di perdita.

Il modello XGBoost consente di aggiornare i pesi degli alberi in base alla direzione e alla curvatura della funzione di perdita, migliorando la stabilità dell'ottimizzazione e introduce diverse ottimizzazioni per migliorare l'efficienza e l'accuratezza:

- Shrinkage (Learning Rate): per controllare la velocità di apprendimento, ogni nuovo albero viene scalato con un fattore η :

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i)$$

Questo aiuta a prevenire aggiornamenti troppo bruschi, migliorando la generalizzazione del modello.

- Column Subsampling: per ridurre la varianza e migliorare la velocità di apprendimento, a ogni iterazione viene selezionato solo un sottoinsieme di feature per determinare il miglior split.
- Sparsity-Aware Split Finding: XGBoost gestisce in modo nativo i valori mancanti, decidendo dinamicamente la direzione migliore da prendere per gli split.
- Parallelizzazione: gli alberi vengono costruiti utilizzando tecniche di pruning e calcolo parallelo per ottimizzare il tempo di addestramento.
- Gamma Regularization: viene introdotto un ulteriore termine di regolarizzazione γ che impone un costo minimo per ogni split, riducendo la crescita non necessaria degli alberi.

XGBoost rappresenta uno degli algoritmi di boosting più efficienti e performanti, in grado di gestire dataset di grandi dimensioni e ottenere risultati superiori rispetto a modelli tradizionali. Grazie alla sua struttura ottimizzata e alla gestione avanzata della regolarizzazione, XGBoost è diventato uno standard per problemi di classificazione e regressione.

4.2 Natural Language Processing (NLP) nella Sanità

Il Natural Language Processing (NLP) rappresenta una tecnologia di punta nel settore sanitario, offrendo soluzioni avanzate per la gestione dei dati clinici non strutturati, come quelli presenti nelle cartelle cliniche elettroniche (EHR), nei documenti scientifici e nella letteratura medica.(18) La natura non strutturata di questi dati ne rende complesso l'utilizzo manuale, poiché l'analisi di referti medici, SDO, annotazioni cliniche e articoli scientifici richiede un elevato livello di competenza specialistica e un notevole dispendio di tempo e risorse economiche.

L'impiego del NLP in ambito sanitario consente di trasformare il testo in dati strutturati, rendendo più efficiente l'estrazione di informazioni rilevanti e migliorando l'accessibilità dei dati per supportare le decisioni cliniche.

L'integrazione del NLP con il ML e il DL sta rivoluzionando la sanità. Analizzando in tempo reale dati clinici quali la storia medica, i risultati dei test diagnostici e altre informazioni, questi modelli offrono un supporto decisionale che consente di formulare diagnosi più accurate e di personalizzare le terapie in base alle specifiche esigenze dei pazienti.

In questo studio per la classificazione di dati non strutturati sono stati impiegati tecniche e modelli di NLP che saranno presentati nel capitolo 5. Di seguito, presentiamo l'evoluzione metodologica che ha progressivamente affinato la rappresentazione del significato delle parole per i task di classificazione (figura 4). In particolare, sarà illustrato il percorso storico, che parte dal One-Hot Encoding, prosegue con i modelli Bag-of-Words (BoW), TF-IDF, n-grammi, word embeddings fino ai modelli preaddestrati, e come ciascuna tecnica contribuisce a migliorare la comprensione e la modellazione del linguaggio naturale.

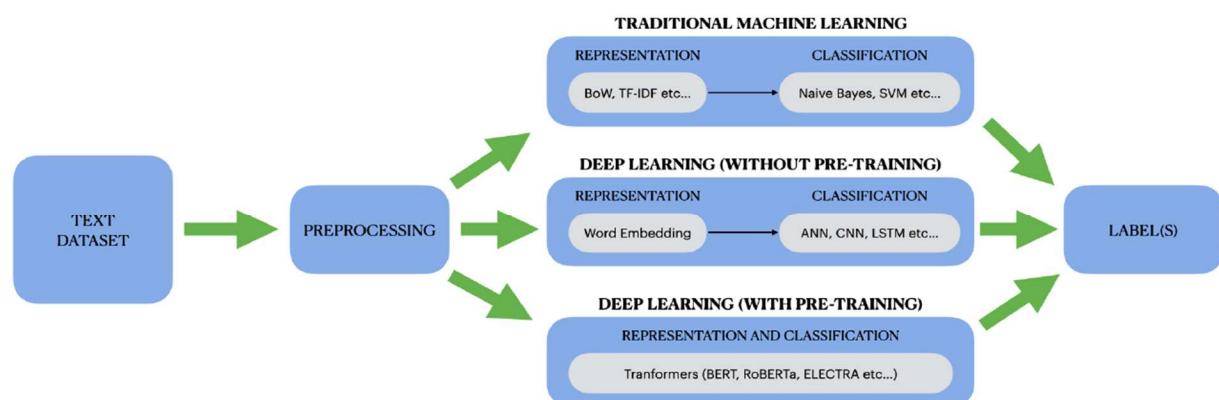


Figura 4 Approcci tradizionali e moderni di NLP

4.2.1 Rappresentazione delle Parole e Valore Semantico

Nel contesto del Natural Language Processing (NLP), uno degli obiettivi principali consiste nel trovare modalità efficaci per rappresentare le parole in una forma interpretabile dai sistemi computazionali, come già ipotizzato da Alan Turing negli anni '50 (Figura 5). Poiché i computer operano esclusivamente su dati numerici, è indispensabile tradurre il linguaggio naturale in vettori matematici, idonei all'elaborazione tramite modelli statistici e algoritmi di machine learning. La sfida risiede nel preservare il significato semantico delle parole, superando la mera rappresentazione simbolica. In questo modo, l'adozione delle tecniche di NLP non solo automatizza l'analisi testuale, ma consente anche di ottenere una comprensione approfondita dei contenuti, interpretando in maniera accurata le strutture e i significati intrinseci del linguaggio naturale.

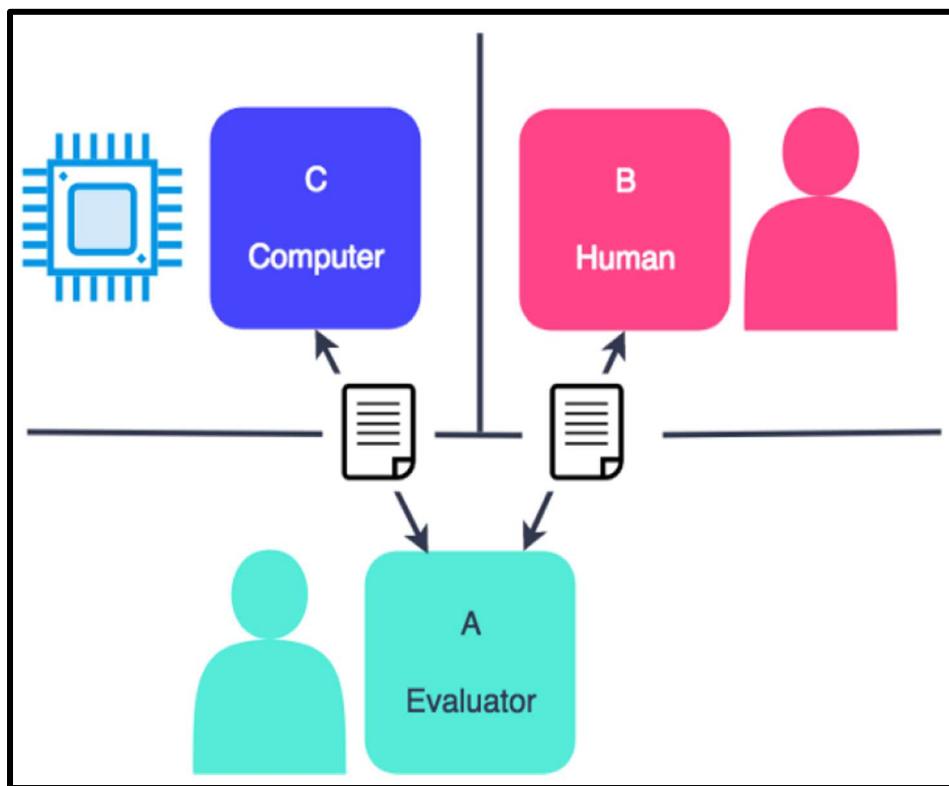


Figura 5 Test di Turing: un valutatore (A) interagisce, senza alcun contatto visivo, con due entità: B, un essere umano, e C, un computer. Se, al termine della comunicazione, il valutatore non riesce a distinguere quale dei due interlocutori sia la macchina e quale l'uomo, allora si considera che C abbia superato il Test di Turing.

4.2.2 One Hot Encoding

Considerando un insieme di N documenti che contengono K parole distinte. La rappresentazione tramite One-Hot Encoding viene costruita seguendo due fasi fondamentali:

- Assegnazione degli identificatori: Ogni parola del dizionario viene associata ad un identificatore numerico compreso tra 0 e $K - 1$.
- Costruzione dei vettori: A ciascuna parola viene assegnato un vettore binario di lunghezza K in cui la posizione corrispondente all'identificatore della parola assume valore 1, mentre tutte le altre posizioni assumono valore 0.

L'One Hot Encoding mappa ogni parola in un vettore univoco e ortogonale, garantisce una rappresentazione semplice ed efficace dal punto di vista implementativo, ma non riesce a catturare le relazioni semantiche tra le parole generando vettori di dimensione elevata, proporzionale al numero totale di parole presenti nel vocabolario.

4.2.3 Bag of Words (BoW)

Estendendo il concetto di One-Hot-Encoding, la rappresentazione dei documenti può essere ottenuta tramite il modello BoW che rappresenta una metodologia semplice ma efficace per trasformare dati testuali in rappresentazioni numeriche, permettendo l'applicazione di algoritmi di apprendimento automatico a testi.

BoW si basa sul conteggio delle occorrenze delle parole all'interno di un documento, ignorando la struttura sintattica e semantica del testo, quindi rimane limitato nel cogliere significati più profondi. Per un insieme di N documenti e un vocabolario costituito da K parole, la BoW si esprime attraverso la costruzione di una matrice M di dimensioni $N \times K$ (nota anche come document-term matrix), in cui:

- Righe e colonne: Ogni riga i (con $i \in [1, N]$) rappresenta un documento, mentre ogni colonna j (con $j \in [1, K]$) corrisponde a una specifica parola.
- Elementi della matrice: L'elemento M_{ij} assume valore 1 se la parola j è presente nel documento i , e 0 altrimenti.

Sebbene il modello BoW perda informazioni sull'ordine delle parole, si dimostra utile in diverse applicazioni di classificazione e modellazione linguistica. Nonostante la sua semplicità, il modello BoW e le sue estensioni presentano limitazioni importanti, come la perdita di informazioni semantiche e la generazione di rappresentazioni di alta dimensionalità.

4.2.4 Term Frequency-Inverse Document Frequency (TF-IDF)

Un ulteriore affinamento della rappresentazione BoW è rappresentato dal modello Term Frequency-Inverse Document Frequency (TF-IDF), che attribuisce un peso diverso ad ogni parola in base alla sua rilevanza. Il peso è calcolato sia in base alla frequenza della parola all'interno di un singolo documento (TF), sia alla sua diffusione nell'intera collezione di documenti (IDF) che contengono la parola.

$$TFIDF = TF * IDF$$

$$TF = \frac{\text{numero di occorrenze della parola specifica nel documento}}{\text{numero totale di parole nel documento}}$$

$$IDF = \log \frac{\text{numero di documenti nel corpus}}{\text{numero di documenti contenenti la parola specifica}}$$

Di conseguenza, il TF-IDF enfatizza i termini distintivi di un documento rispetto a quelli comuni, mitigando l'influenza delle parole più frequenti che spesso non sono significative (come articoli e preposizioni). Sebbene il TF-IDF sia una tecnica semplice e intuitiva, si rivela estremamente potente e viene utilizzata in molteplici ambiti, tra cui i motori di ricerca e la classificazione testuale.

4.2.5 N-Gram e Curse of Dimensionality

Gli *n-grams* sono sequenze contigue di n parole estratte da un testo. A differenza del modello BoW che considera ogni parola singolarmente, ovvero i cosiddetti “unigrammi”, gli *n-grams* catturano l’ordine e le relazioni locali tra le parole. Ad esempio, dalla frase “il gatto è sul tavolo” si ottengono bigrammi come (il, gatto) e (gatto, è).

Questa metodologia risente della problematica *Curse of Dimensionality*; in pratica all’aumentare del valore di n, il numero di possibili *n-grams* cresce esponenzialmente, ciò comporta un aumento significativo della dimensionalità della rappresentazione, con conseguenze negative in termini di memoria e capacità computazionali. Inoltre gli *n-grams* non sempre riescono a cogliere relazioni o dipendenze ad ampio raggio che possono essere importanti per comprendere il significato complessivo del testo.

4.2.6 Word Embedding

il Word Embedding si distingue dalle rappresentazioni precedenti, per la capacità di generare rappresentazioni dense e semantiche delle parole. Questa tecnica, nota anche come rappresentazione distribuzionale, si fonda sull'idea che il significato di una parola possa essere dedotto dal suo contesto, ossia dalle parole che la precedono e la seguono in un corpus di testo. Il risultato è una mappatura delle parole in vettori multidimensionali di numeri reali, in cui la vicinanza geometrica tra vettori riflette la similitudine semantica e sintattica tra le parole (Figura 6). Grazie a queste caratteristiche, le Word Embeddings rappresentano strumenti fondamentali per molteplici applicazioni nell'ambito del NLP.

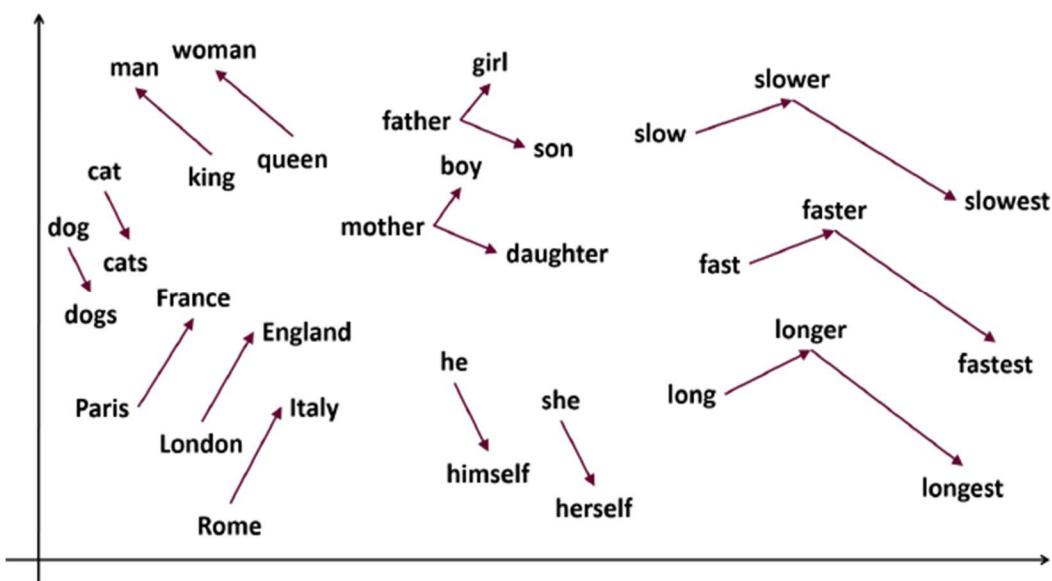


Figura 6 esempio di visualizzazione bidimensionale degli embedding delle parole

Tra metodi di Word Embedding, ci sono modelli basati su reti neurali come Word2Vec, GloVe e FastText, che hanno introdotto una rappresentazione continua che riesce a catturare il significato semantico e sintattico delle parole (19,20,21).

Word2Vec, sviluppato da Mikolov (19), si articola in due varianti principali: Continuous Bag of Words (CBOW) e Skip-Gram (Figura 7). Il modello CBOW utilizza come input un insieme di parole di contesto per predire la parola centrale della finestra considerata. Il modello Skip-Gram opera in modo inverso, utilizzando una parola centrale per stimare le parole del contesto. Entrambi i metodi sfruttano una rete neurale con tre livelli (corrispondenti a input, nascosto e output), nella quale i pesi associati al livello nascosto rappresentano gli embeddings da apprendere.

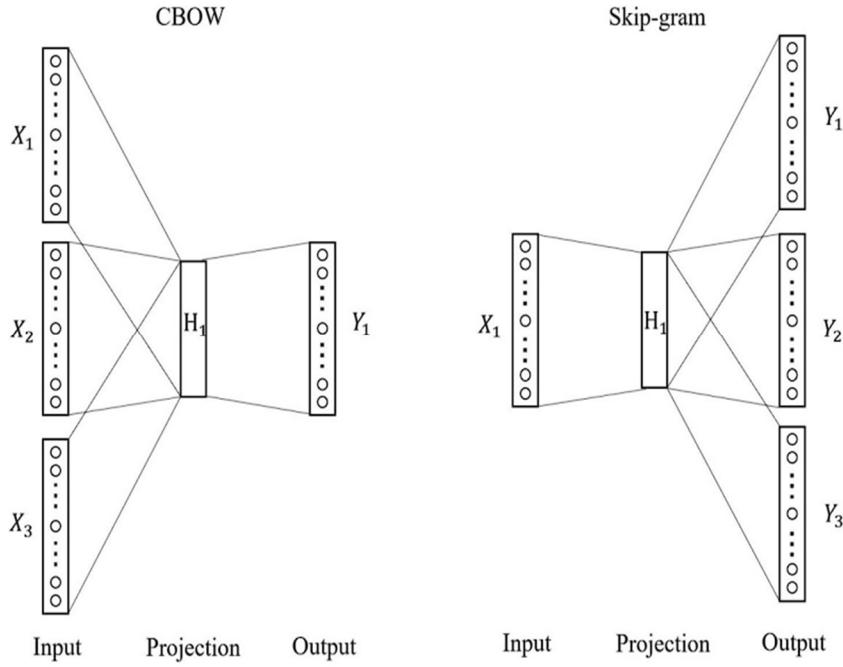


Figura 7 Word2Vec model (Continuous Bag of Words (CBOW) and Skip-Ngram)

L'ottimizzazione dei risultati avviene attraverso il metodo della discesa del gradiente con backpropagation e l'uso di funzioni di perdita, spesso accompagnate da tecniche di approssimazione come Negative Sampling o Hierarchical Softmax, per rendere il calcolo computazionalmente più efficiente. Word2Vec ha il vantaggio di apprendere relazioni semantiche locali tra le parole, risultando particolarmente utile nei compiti di NLP basati sul contesto immediato.

Un approccio differente è adottato da GloVe (*Global Vectors for Word Representation*), introdotto da Pennington (20), che combina le caratteristiche dei modelli predittivi con un approccio basato sulle statistiche di co-occorrenza delle parole.

GloVe costruisce una matrice di co-occorrenza che rappresenta la frequenza con cui le parole compaiono insieme all'interno di un ampio corpus testuale, calcolando successivamente la probabilità di co-occorrenza normalizzata. L'algoritmo esegue quindi una fattorizzazione della matrice, cercando di minimizzare la differenza tra i prodotti scalari degli embeddings e i valori di co-occorrenza attesi, attraverso una funzione di costo ponderata. Questo permette di generare rappresentazioni che catturano relazioni semantiche a livello globale, risultando particolarmente efficaci per l'analisi delle similarità tra parole e per la risoluzione di analogie.

Dal punto di vista delle prestazioni Word2Vec e GloVe presentano differenze sostanziali.

Word2Vec, e in particolare il modello Skip-Gram, è in grado di rappresentare meglio parole rare e ambigue, mentre CBOW si dimostra più veloce nell'apprendimento con dataset di grandi dimensioni. CBOW e Skip-Gram sono due varianti che apprendono relazioni esclusivamente sulla base di finestre di contesto locali.

GloVe, invece, grazie al suo approccio basato sulle statistiche globali del corpus, cattura più efficacemente le relazioni semantiche tra parole, risultando particolarmente utile nei casi in cui sia necessario modellare similarità ad ampio raggio tra termini.

FastText è un metodo per ottenere rappresentazioni vettoriali delle parole che si basa sull'analisi dei componenti interni delle parole stesse. Invece di considerare ogni parola una entità indivisibile, FastText la scomponete in piccoli segmenti, chiamati n-grammi. (21) La rappresentazione vettoriale di una parola viene costruita aggregando i vettori dei suoi n-grammi.

FastText affronta il problema dei termini fuori dal vocabolario (OOV) e migliora la gestione di parole mai comparse durante l'addestramento.

In sintesi i modelli Word2Vec, GloVe e FastText, pur essendo efficaci per le soluzioni che propongono presentano alcuni limiti, in quanto assegnando un unico vettore a ciascuna parola, non riescono a catturare le sfumature contestuali in cui essa viene utilizzata e le variazioni semantiche. Queste limitazioni hanno determinato lo sviluppo di tecniche più avanzate, come gli embedding contestuali, modelli di linguaggio più sofisticati basati su architetture transformer, quali BERT e GPT.

4.2.7 Reti Neurali Artificiali (ANN) e Deep Learning (DL)

Le Reti Neurali Artificiali (*Artificial Neural Network ANN*) sono una delle tecnologie di base nel campo del DL, che simulando il funzionamento del cervello umano, mirano a replicare i meccanismi del ragionamento e dell'apprendimento biologico.

Una rete neurale, pur non essendo un algoritmo nel senso tradizionale del termine, fornisce un framework all'interno del quale vari algoritmi di ML possono interagire per trattare e processare dati complessi. Le ANN sono strutturate in diversi strati: il livello di input, uno o più strati di livello nascosto e il livello di output (Figura 8). Ogni strato è composto da un numero arbitrario di neuroni, unità computazionali, in grado di ricevere segnali di ingresso, elaborarli tramite pesi associati, e applicare una funzione di attivazione che determina l'output. I neuroni sono interconnessi tramite bordi ponderati che rendono il sistema altamente adattabile,

in quanto in risposta agli stimoli ricevuti, riesce a modificare la propria struttura variando i pesi e le connessioni.

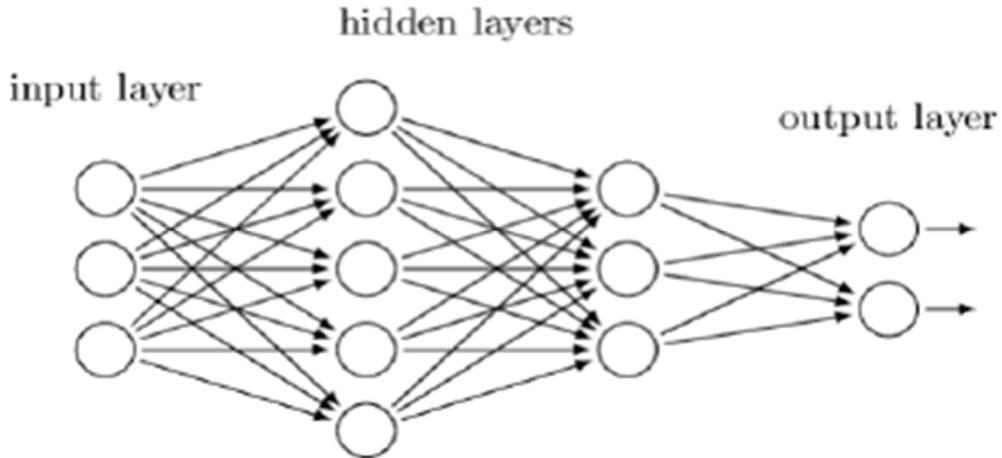


Figura 8 schema rete neurale artificiale

Nel dettaglio:

- *livello di ingresso (input layer)*: riceve in ingresso i dati e contiene neuroni ognuno dei quali rappresenta una specifica feature del dataset.
- *livello nascosto (hidden layer)*: Negli strati nascosti, le caratteristiche in ingresso vengono elaborate tramite somme ponderate che integrano pesi e bias, consentendo alla rete di apprendere nuove proprietà; il risultato di queste elaborazioni viene quindi processato da una funzione di attivazione (come Sigmoide, ReLU o tanh).
- *livello di uscita (output layer)*: Il livello di output fornisce i risultati finali, adattandoli alle esigenze del blocco successivo della rete neurale.

Il singolo neurone artificiale (Figura 9) non è altro che una funzione matematica cui, dato un input un insieme di variabili indipendenti $x = (x_1, x_2, \dots, x_n)$ ed i relativi pesi $w = (w_1, w_2, \dots, w_n)$ restituisce come output: $y = \varphi(v + b)$ dove :

- $v = \sum_{i=1}^m x_i w_i$ è la sommatoria pesata dei dati in input al neurone.
- b (“bias” o “threshold”) è un parametro esterno, solitamente con valore negativo, che viene applicato per gestire la sensibilità della risposta del neurone ai dati in input.
- φ è la funzione di attivazione che determina l’output o l’attivazione del neurone.

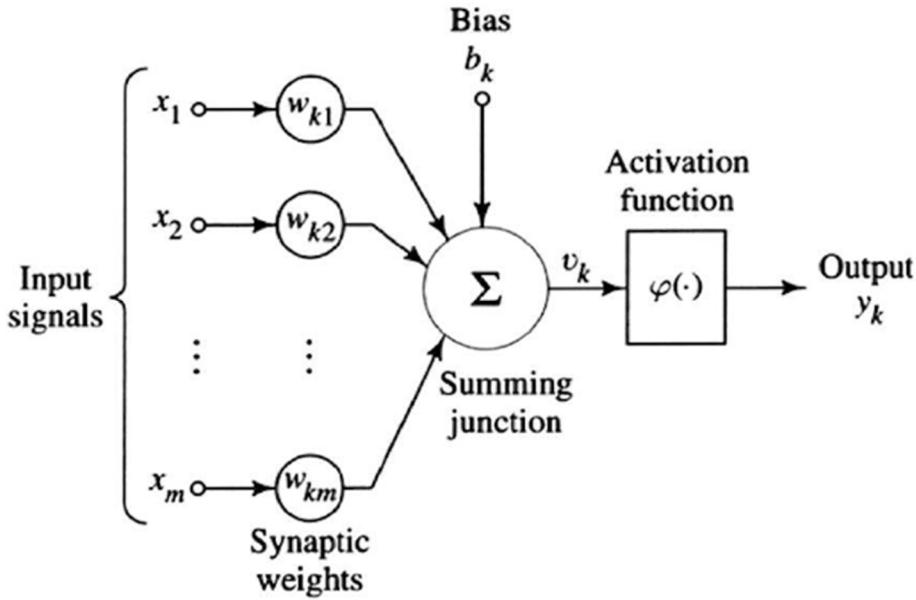


Figura 9 schema funzionamento del calcolo di una rete neurale

Le funzioni di attivazione più adottate sono:

- La funzione logistica o sigmoide:

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

questa funzione viene solitamente impiegata nello strato di output per modelli di classificazione binaria. È una delle funzioni più usate anche se non è esente dal problema della scomparsa del gradiente, che si verifica quando il gradiente assume un valore talmente basso che la rete rifiuta di apprendere ulteriormente.

- La funzione tangente iperbolica (tahn) è una buona alternativa alla sigmoide:

$$g(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

La sua natura è sempre non lineare, ma il suo gradiente è molto più resistente della sigmoide e decidere tra le due dipenderà dalle richieste di robustezza del gradiente stesso. Anch'essa, tuttavia, non è esente dal problema della scomparsa del gradiente.

- la funzione SoftMax:

$$g(x)_i = (e^{z_i}) / (\sum_{j=1}^k e^{z_j}) \quad \text{per } i = 1, \dots, k \text{ e } z = (z_1, \dots, z_k) \in R^k$$

La funzione SoftMax viene solitamente impiegata nello strato di output per modelli di classificazione multclasse.

- La funzione di attivazione ReLU (Rectified Linear Unit) è ampiamente utilizzata nelle reti neurali per introdurre non linearità nel modello. La sua definizione matematica è la seguente:

$$f(x) = \max(0, x)$$

In altre parole, per ogni input x , la funzione restituisce x se x è positivo, altrimenti restituisce 0. Questa semplicità computazionale, insieme alla capacità di mitigare il problema del gradiente scomparso, rende ReLU una scelta popolare nelle architetture di reti profonde.

Il processo di addestramento di una rete neurale si basa sull'ottimizzazione dei pesi attraverso algoritmi come la backpropagation, che utilizza la discesa del gradiente per minimizzare l'errore nel processo di predizione.

$$\min\{J(W)\} = \min \left\{ \frac{1}{N} \sum_{j=1}^N (L_i(f(x_i, W), y_{true})) + \lambda R(W) \right\}$$

dove:

- L_i rappresenta la funzione di perdita calcolata tra la predizione della rete $y_{pred} = f(x_i, W)$ e il relativo valore target y_{true} .
- $R(W)$ è il termine di regolarizzazione che penalizza la complessità del modello.
- λ è il parametro di regolarizzazione.
- N è il numero di campioni considerati nel batch.
- W rappresenta l'insieme dei pesi della rete.

Il DL è emerso come un'evoluzione delle reti neurali tradizionali (ANN), motivato dalla necessità di affrontare problemi sempre più complessi. A differenza delle tecniche di ML convenzionali, che richiedono una selezione manuale delle caratteristiche rilevanti per l'analisi, il DL sfrutta architetture di rete più profonde, dove l'aumento del numero di strati consente di apprendere rappresentazioni sempre più astratte e sofisticate dei dati (Figura 10).

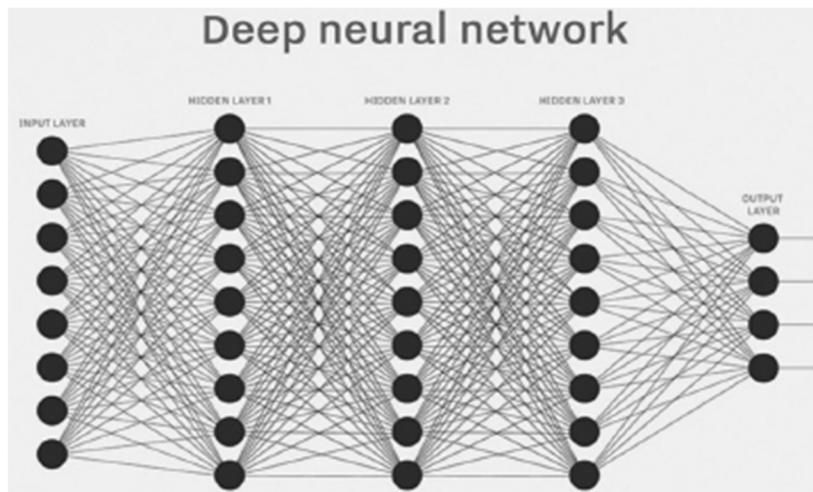


Figura 10 Esempio di rete neurale profonda (Deep Learning)

La caratteristica principale del DL risiede nell'idea che, con una struttura a più livelli, ogni strato successivo può elaborare in modo progressivamente più complesso le informazioni estratte dai livelli precedenti. Questo processo gerarchico di apprendimento consente al modello di catturare interazioni e pattern che sarebbero difficili da definire manualmente, portando a una maggiore capacità di generalizzazione e a migliori risultati in compiti di classificazione, regressione e analisi predittiva.

Il DL ha aperto nuove opportunità per l'analisi dei dati clinici e biomedici, dove le relazioni tra le variabili sono complesse e difficili da modellare esplicitamente. Nonostante i risultati eccezionali ottenuti in numerosi compiti, l'adozione di queste tecnologie è spesso limitata dalla difficoltà di acquisire grandi quantità di dati etichettati e dalla natura "black box" dei modelli, che rende complicato per i medici interpretare le motivazioni alla base delle previsioni.

L'evoluzione del DL è strettamente legata alla disponibilità di grandi quantità di dati e all'uso di strumenti, come le GPU, che riescono ad eseguire operazioni in parallelo su un gran numero di dati. Grazie alla loro architettura parallela, le GPU sono in grado di accelerare in modo significativo operazioni computazionali intensive, come le moltiplicazioni di matrici e le

operazioni vettoriali, che sono fondamentali per l'addestramento e l'inferenza dei modelli di DL.

Le GPU consentono di ridurre drasticamente i tempi di elaborazione rispetto alle CPU tradizionali, facilitando la gestione di dataset di grandi dimensioni e la sperimentazione di modelli complessi. Nel campo del NLP l'applicazione delle tecniche di DL ha rappresentato un'evoluzione importante per quanto concerne la classificazione del testo e l'analisi di documenti clinici, anche se la trasparenza e l'affidabilità dei modelli rimane ancora un obiettivo da raggiungere.

4.2.8 BiLSTM

Le Reti Neurali Ricorrenti Bidirezionali (BiLSTM), rappresentano una variante delle reti Long Short-Term Memory (LSTM) e Recurrent Neural Networks (RNN), che viene ampiamente utilizzata per l'elaborazione di sequenze di dati come il testo. Le RNN e le LSTM sono architetture particolarmente indicate nell'ambito di NLP, in quanto sono in grado, tramite la propagazione dei loro stati nascosti, di mantenere una memoria temporale dei dati precedenti.

La BiLSTM rappresenta una architettura più evoluta, in quanto funziona processando simultaneamente le sequenze di dati in due direzioni. Ogni strato della BiLSTM è composto da due reti LSTM indipendenti. Una LSTM 'forward' che analizza i dati dalla sequenza iniziale a quella finale e una LSTM 'backward', che processa la sequenza in ordine inverso dall'ultimo al primo elemento.

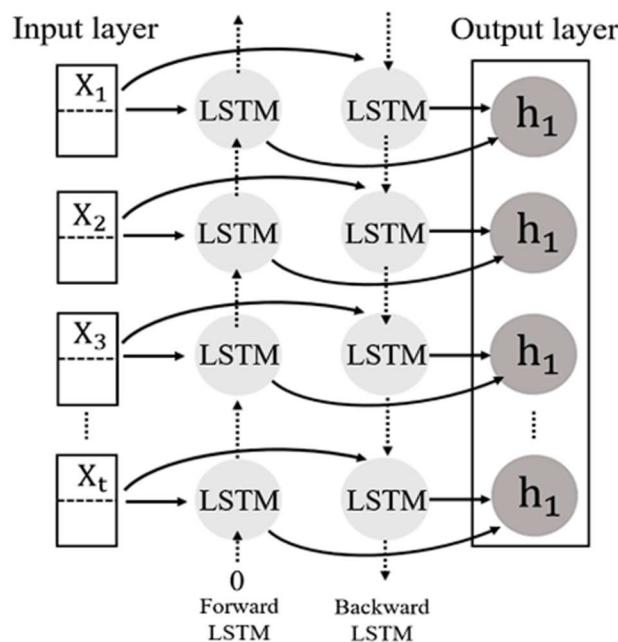


Figura 11 schema di rete neurale BiLSTM

Per ogni elemento della sequenza (Figura 11), gli output prodotti dalle due LSTM vengono concatenati per formare una rappresentazione finale che incorpora il contesto proveniente da entrambe le direzioni. Attraverso questa elaborazione BiLSTM fornisce una visione più completa e contestualizzata dei dati, in quanto consente di catturare meglio le informazioni provenienti dal contesto precedente e successivo in una sequenza di dati, migliorando l'accuratezza nelle previsioni.

4.2.9 Il modello Transformer

L'architettura attualmente più diffusa per la realizzazione di modelli linguistici neurali contestuali è il Transformer, introdotto per la prima volta con il lavoro “*Attention is All You Need*”. (22) Il Transformer è una rete neurale di tipo Encoder-Decoder che sfrutta il meccanismo di *attention* per focalizzarsi selettivamente su porzioni rilevanti di una frase e per costruire rappresentazioni contestuali delle parole.

Nella versione originaria del Transformer (Figura 12), sia l'encoder che il decoder sono composti da sei blocchi ripetuti (encoder e decoder rispettivamente), che condividono la stessa struttura se non per i pesi, inizializzati in modo casuale. Gli encoder sono strutturati in due componenti principali: uno strato di self-attention e una rete neurale feed-forward. I decoder, oltre a questi due livelli, integrano un ulteriore strato di encoder-decoder attention, posizionato tra il self-attention e la rete feed-forward, il quale serve a determinare su quale token concentrarsi ad ogni passo del processo. Ogni token in ingresso viene trasformato in un *word embedding* e processato dal primo encoder, il quale ne codifica il significato. I successivi encoder, invece, ricevono in input gli embedding provenienti dall'encoder precedente, arricchendoli ulteriormente di contesto. Durante il processo, è lo strato di self-attention a consentire ad ogni token di "osservare" gli altri token della sequenza e di considerare le loro posizioni, producendo così un embedding contestuale che riflette il ruolo specifico della parola all'interno della frase.

Il nucleo del modello Transformer è costituito dal meccanismo di self-attention, che consente ai token di interagire tra loro e di pesare reciprocamente la propria rilevanza all'interno di una sequenza. In questo schema, a ciascun token vengono assegnate tre rappresentazioni distinte: query (Q), key (K) e value (V):

- Query (Q) viene impiegata per cercare informazioni negli altri token.
- Key (K) serve a calcolare i pesi dell'attenzione.

- Value (V) fornisce l'output informativo dell'attenzione, in quanto indica la quantità di informazioni da trasferire.

Queste tre rappresentazioni si ottengono moltiplicando l'embedding di ciascun token per tre matrici di peso (WQ , WK , WV) apprese durante l'addestramento. Per ogni token, si calcola quindi il punteggio della self-attention valutando il rapporto tra la sua query e i key degli altri token. Ad esempio, per il token in posizione #1, il punteggio si ottiene calcolando il prodotto scalare tra la sua query e i key dei token nelle altre posizioni.

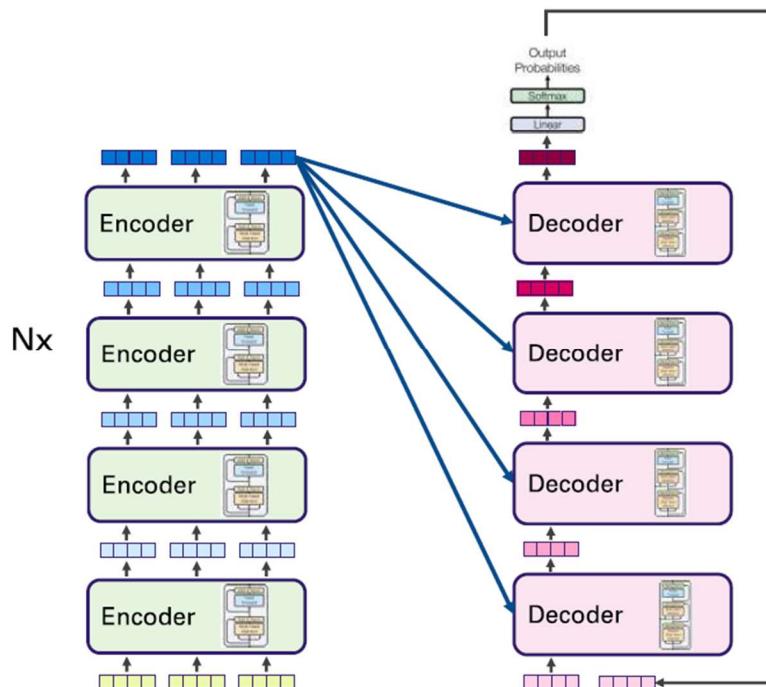


Figura 12 struttura del transformer originale, formato da un numero uguale N di blocchi encoder e decoder.

Questi punteggi vengono normalizzati dividendo per $\sqrt{d_k}$ e passati attraverso uno strato SoftMax. Successivamente, ogni rappresentazione value viene moltiplicata per il relativo punteggio normalizzato, eliminando così le informazioni irrilevanti e preservando quelle essenziali. La somma di tutte le rappresentazioni value pesate produce l'output dello strato di self-attention per ciascun token. La formula fondamentale per il calcolo dell'output dell'attenzione è:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK_T}{\sqrt{d_k}}\right)V$$

Per consentire al modello di concentrarsi su aspetti diversi della medesima parola, l'implementazione originale prevede l'uso della *multi-head attention*. Questa tecnica amplia la

capacità del modello di esaminare contemporaneamente differenti posizioni della sequenza, generando molteplici set di query, key e value (solitamente otto per parola). Poiché lo strato feed-forward accetta in input una singola matrice, i risultati delle diverse teste vengono concatenati e successivamente moltiplicati per una matrice di pesi (Figura 13). Per tenere conto della posizione dei token nella sequenza, viene aggiunto anche un embedding posizionale. Nel decoder, ad ogni passo viene impiegato un set di vettori di attenzione derivanti dall'output dell'ultimo encoder; questi vettori sono usati nello specifico strato di encoder-decoder attention. Inoltre, il decoder finale genera a ogni step un token, il quale viene reinserito nel processo per integrare il contesto già elaborato, continuando così fino al rilevamento del token di fine sequenza.

È importante sottolineare che il layer di self-attention nel decoder è leggermente modificato per evitare che il modello "guardi" i token futuri. A ogni passo, infatti, i token non ancora generati vengono mascherati, impedendo al decoder di accedere a informazioni non disponibili. Infine, il modello non produce direttamente parole in linguaggio naturale: per questo, vengono applicati due strati finali, un layer lineare e un SoftMax, dove il primo proietta gli output in un vettore di logit avente dimensioni pari al vocabolario e il secondo trasforma tali logit in probabilità per selezionare il token più probabile.

I Transformer hanno superato alcune criticità delle RNN. Innanzitutto, la natura sequenziale delle RNN rende difficile sfruttare appieno le capacità di elaborazione parallela offerte da GPU e TPU. Inoltre, mentre un passo di addestramento di una RNN ha una complessità legata alla lunghezza delle sequenze, nei Transformer tale complessità risulta costante. Oltre a questo, il Transformer offre una maggiore interpretabilità, in quanto è possibile visualizzare quali parti della frase vengono considerate durante l'elaborazione o la traduzione di un dato token (22). Infine, a differenza delle RNN, che devono attendere la lettura completa della sequenza per interpretare il significato di una parola, l'encoder del Transformer permette a tutte le parole di interagire simultaneamente, consentendo una comprensione immediata del contesto.

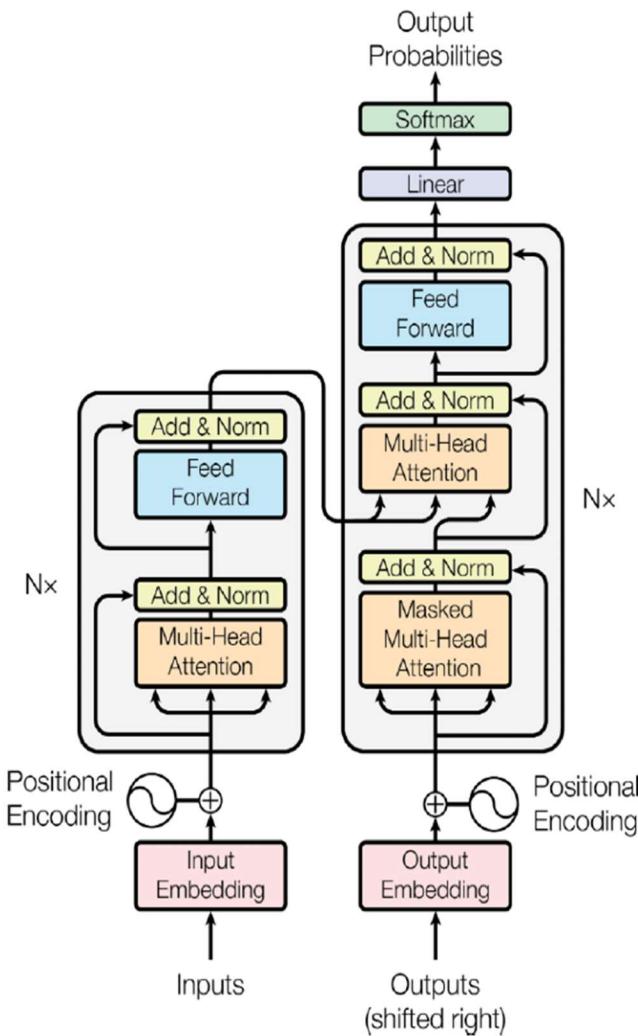


Figura 13 Architettura Transformer Fonte: *Attention is all you need*

L’architettura a doppio componente del Transformer ha portato alla distinzione tra modelli “Encoder-only”, come il noto BERT, e modelli “Decoder-only”, come GPT. Esistono inoltre modelli che adottano la struttura Encoder-Decoder, tra cui T5, anche se con minore frequenza.

I modelli pre-addestrati, hanno rivoluzionato il campo del NLP grazie al transfer learning, un approccio che permette di superare le limitazioni legate alla scarsità di dati e ai lunghi tempi di addestramento. Con questo metodo non è più necessario creare un modello da zero: si utilizza fin da subito un modello pre-addestrato, che ha ottenuto una vasta conoscenza grazie all’addestramento su una grande quantità di dati non etichettati, per poi essere adattato a specifici compiti tramite il fine-tuning. Questo processo consente di ottenere prestazioni elevate anche con dataset modesti e si rivela particolarmente utile in contesti applicativi delicati e complessi, come quello medico, dove la precisione è fondamentale e la disponibilità di dati etichettati può essere limitata.

Tra i principali vantaggi del transfer learning troviamo:

- Utilizzo immediato di modelli pre-addestrati: Non è necessario partire da zero, risparmiando tempo e risorse grazie alla base di conoscenze già acquisita.
- Riduzione dei tempi di addestramento: La fase più dispendiosa, ovvero il pre-training, viene eseguita una sola volta, permettendo un rapido adattamento a nuovi task.
- Efficienza nell'utilizzo dei dati: Anche con dataset ridotti, il modello può ottenere prestazioni elevate grazie alla conoscenza pre-acquisita.
- Adattabilità a diversi domini: La stessa base di conoscenze può essere trasferita e riutilizzata in ambiti specifici, come la diagnostica medica, il supporto alle decisioni cliniche e la gestione delle informazioni sanitarie.
- Riduzione della necessità di dati etichettati: Il pre-training può essere effettuato su dati non etichettati, più facilmente reperibili, risolvendo il problema della scarsità di dati specifici.
- Maggiore precisione in applicazioni critiche: In settori come quello medico, dove errori possono avere conseguenze significative, il transfer learning contribuisce a sviluppare modelli più robusti e affidabili.

In conclusione il transfer learning, rappresenta uno strumento fondamentale per affrontare le sfide del NLP, garantendo un equilibrio ottimale tra efficienza, adattabilità e precisione in vari ambiti applicativi.

4.2.10 BERT

La diffusione di BERT (Bidirectional Encoder Representations from Transformers) da parte di Google nel 2018 ha rappresentato un punto di svolta nel campo del Natural Language Processing. Nell'articolo di Devlin (23) vengono illustrate sia la sua architettura che gli obiettivi che il modello si prefigge di raggiungere.

BERT adotta esclusivamente la parte encoder del Transformer, sfruttando una self-attention bidirezionale per generare rappresentazioni contestuali dei token di input. Nella configurazione standard, detta BERTBASE, il modello è strutturato con 12 livelli di encoder, 12 teste di attenzione e un aumento a 768 unità nascoste nelle reti feed-forward. Sebbene il meccanismo di creazione degli embedding segua il procedimento originario del Transformer, BERT introduce alcune innovazioni, in particolare nella tokenizzazione, realizzata tramite l'algoritmo WordPiece, e nella netta separazione dell'addestramento in due fasi: pre-training e fine-tuning.

Durante il pre-training, fase cruciale per dotare il modello di una base linguistica generica, vengono utilizzati vasti corpus come il Toronto BookCorpus (800 milioni di parole) e la Wikipedia inglese (2,500 milioni di parole). In questa fase, BERT viene addestrato a risolvere due compiti che sostituiscono il tradizionale language modeling: il Masked Language Modeling (MLM) e il Next Sentence Prediction (NSP). Il task MLM, ispirato al cloze test della linguistica, prevede la mascheratura del 15% delle parole nel testo (circa il 90% dei casi sostituite con il token [MASK] e il restante 10% con un token casuale), con l’obiettivo di far predire al modello le parole originarie. Il task NSP, invece, mira a migliorare la capacità del modello di comprendere le relazioni tra coppie di frasi, richiedendogli di classificare se la seconda frase seguia logicamente la prima. Una volta completata la fase di pre-training, BERT è in grado di generare rappresentazioni codificate per ogni token in frasi mai viste, che possono essere impiegate in vari compiti di classificazione.

La fase fine-tuning permette un ulteriore addestramento su un dataset etichettato con una funzione obiettivo modificata rispetto a quella utilizzata in fase di pre-training, al fine di rendere il modello maggiormente specializzato su un task specifico.

Un aspetto distintivo di BERT riguarda la tokenizzazione tramite WordPiece, che consente di costruire un vocabolario contenente sia token speciali (come [PAD], [UNK], [CLS], [SEP] e [MASK]) sia “sottoparole”, ovvero segmenti che, pur non corrispondendo esattamente a morfemi o sillabe, permettono di gestire in modo efficace le parole meno frequenti.

Il processo di tokenizzazione prevede l’identificazione della sotto parte più lunga presente nel vocabolario per ogni parola, partendo dal primo carattere e considerando, per il resto della parola, i caratteri preceduti dal prefisso ‘##’. Inoltre, per garantire che tutte le sequenze abbiano la stessa lunghezza, viene inserito il token [CLS] all’inizio, il token [SEP] tra le frasi e, se necessario, vengono aggiunti token [PAD] per uniformare la lunghezza.

Generalmente, i modelli di linguaggio vengono pre-addestrati su ampi dataset di dominio generico (ad esempio, l’intero corpus di Wikipedia) per poi essere adattati a compiti specifici tramite il fine-tuning. Nel corso del tempo, BERT è stato implementato in numerose varianti che differiscono per dimensione, dataset utilizzati e iperparametri scelti. Tra questi, Bio_Discharge_Summary_BERT è particolarmente rilevante per il nostro studio, essendo stato rispettivamente addestrato su cartelle cliniche, in particolare con riassunti di dimissione ospedaliera.

4.2.11 Bio_Discharge_Summary_BERT

Alsentzer (24), ha sviluppato una serie di modelli BERT specifici per il dominio medico, tra cui Bio_Discharge_Summary_BERT, una variante ottimizzata per l'elaborazione delle schede di dimissione ospedaliera. Il modello Bio_Discharge_Summary_BERT è stato ottenuto mediante il fine-tuning di BioBERT, un modello pre-addestrato su testi biomedici provenienti da PubMed, con un corpus specifico di schede di dimissione clinica. I dati utilizzati per l'addestramento provengono dal dataset MIMIC-III (Medical Information Mart for Intensive Care III), un vasto dataset di dati clinici anonimi provenienti da pazienti ricoverati presso il Beth Israel Deaconess Medical Center (BIDMC) tra il 2001 e il 2012.

Le schede di dimissione utilizzate rappresentano una fonte considerevole di dati clinici, in quanto contengono informazioni su diagnosi (codificate secondo ICD-9-CM), trattamenti, esami di laboratorio, farmaci, dati di follow-up, note cliniche testuali e altri dati utili per la ricerca in ambito sanitario.

La fase di addestramento del modello Bio_Discharge_Summary_BERT, ha previsto, come per la fase di pre-training di BERT, una lunghezza massima della sequenza di 128 token e una learning rate di 5×10^{-5} , per un totale di 150.000 step di addestramento. Il modello risultante è stato poi valutato su diversi task di NLP clinico, tra cui il riconoscimento di entità mediche (NER) e la classificazione testuale.

Le valutazioni sperimentali condotte da Alsentzer hanno mostrato che Bio_Discharge_Summary_BERT offre su task di NLP clinico prestazioni superiori rispetto ai modelli BERT generali e BioBERT. In particolare, il modello ha ottenuto un miglioramento nelle metriche di F1-score per il riconoscimento di entità cliniche nei dataset i2b2 del 2010 e i2b2 del 2012, evidenziando l'importanza dell'uso di un corpus specifico per il dominio clinico.

Inoltre, il modello ha mostrato un'ottima capacità di catturare il linguaggio specialistico delle dimissioni ospedaliere, migliorando la comprensione dei contesti medici rispetto alle versioni di BERT non specializzate. In sintesi il modello Bio_Discharge_Summary_BERT rappresenta un importante passo avanti per l'NLP clinico, dimostrando che l'uso di dati specializzati, come quelli delle schede di dimissione, consente di ottenere prestazioni migliori rispetto ai modelli addestrati su testi biomedici generali. Infatti grazie all'addestramento su dati provenienti dalle schede di dimissione clinica il modello è divenuto più esperto nella classificazione del testo medico e nell'estrazione di informazioni

cliniche. Rimane ancora da affrontare la necessità di integrare dati provenienti da più istituzioni sanitarie e il miglioramento delle performance con dataset clinici.

4.3 Ottimizzazione e Regolarizzazione

Per migliorare le prestazioni dei modelli di ML e NLP e ridurre il tasso d'errore esistono approcci utili di fine tuning, cioè una combinazione di tecniche di ottimizzazione e di regolarizzazione che permettono di affinare il confine decisionale e garantire una maggiore capacità di generalizzazione dei modelli. Un aspetto critico che il fine-tuning intende affrontare è rappresentato dai fenomeni di overfitting e underfitting.

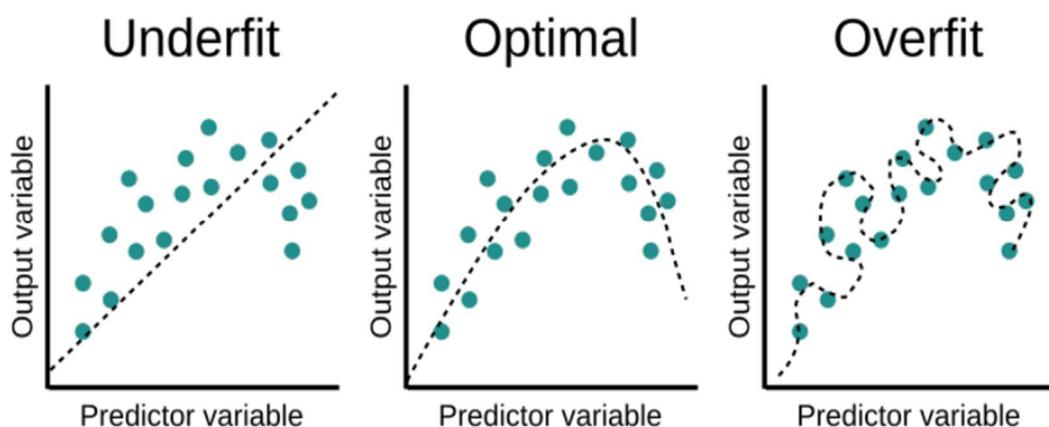


Figura 14 Underfitting, Optimal e Overfitting

L'Overfitting si verifica quando il modello apprende non solo i pattern generali che legano gli input agli output, ma anche dettagli troppo specifici e non generalizzabili caratteristici del dataset di training. Questo fenomeno comporta una diminuzione dell'errore sui dati di addestramento, ma un aumento dell'errore sui dataset di test, poiché il modello ha memorizzato pattern che non si ripetono nei nuovi dati. L'Underfitting si verifica quando il modello non riesce ad apprendere i pattern generali e non è in grado di produrre previsioni accurate sia sul dataset di training che su quello di test.

Il processo di fine-tuning è quindi fondamentale per migliorare le prestazioni complessive del modello e garantire risultati affidabili in applicazioni reali, in quanto riesce a trovare un equilibrio ottimale per ridurre sia il rischio di overfitting che quello di underfitting (Figura 14).

4.3.1 Split in Training, Test e Validation

La pratica di suddividere il dataset in porzioni dedicate all'addestramento, alla validazione e al test è nata dalla necessità di valutare in modo affidabile la capacità di

generalizzazione dei modelli di ML. Inizialmente, questa metodologia è stata adottata per evitare problemi come l'overfitting e per garantire che le prestazioni misurate non fossero semplicemente dovute alla capacità del modello di memorizzare i dati di addestramento.

Suddivisione del dataset:

- Training set: (60-70% dei dati) è il dataset più corposo che viene utilizzato per insegnare al modello le caratteristiche fondamentali del problema, durante l'addestramento, il modello apprende le relazioni tra le variabili di input (variabili indipendenti) e la variabile target (variabile dipendente), adattando i propri parametri per minimizzare l'errore.
- Il Test set: (20-30% dei dati) è un dataset che valuta le prestazioni del modello su dati non visti in fase di training, ottimizzando gli iperparametri ed evitando l'overfitting, per garantire una buona generalizzazione su nuovi dati.
- La Validation set (di solito il 10-20% del dataset) è un campione indipendente usato per valutare le prestazioni finali del modello, una volta completate le fasi di addestramento e ottimizzazione. Durante la validation viene realizzato uno studio della Threshold, che stabilisce un valore di soglia, la decision boundary, delle probabilità a posteriori. Regolare opportunamente la soglia (Threshold) di decisione, cioè il valore a partire dal quale un'istanza viene classificata in una classe piuttosto che nell'altra, può portare a ottenere migliori risultati in termini di accuratezza e di performance (come F1-score, Recall, Precision e specificity). Tale scelta è particolarmente importante quando si lavora con dati sbilanciati.

In questa ricerca, la variabile target è rappresentata dall'esito clinico dei pazienti secondo i dati SDO e risulta composta da due classi:

- La classe positiva che indica i pazienti deceduti ($\approx 5\%$ del campione totale).
- La classe negativa che rappresenta i pazienti sopravvissuti ($\approx 95\%$ del totale).

Nella suddivisione del dataset, utilizzato in questo studio, è stato fondamentale mantenere questa proporzione reale relativa ai due esiti della variabile target (5% vs 95%), questo al fine di garantire che il modello apprendesse e venisse validato su dati che riflettessero fedelmente la realtà.

4.3.2 Feature Selection (LASSO)

Lasso (Least Absolute Shrinkage and Selection Operator) è una tecnica di Feature Selection, che si basa su una regressione lineare con penalizzazione L1. Lasso, applicando una regolarizzazione (L1) riduce progressivamente i coefficienti delle variabili meno rilevanti, fino a farli diventare esattamente zero, identificando in questo modo le funzionalità chiave e le variabili più significative. Questo metodo permette di migliorare la generalizzazione del modello e diminuire il rischio di overfitting. Nel contesto di una regressione lineare, l'obiettivo è stimare il vettore dei coefficienti $\beta = (\beta_1, \beta_2, \dots, \beta_p)$ che meglio descrive la relazione tra una variabile dipendente y e un insieme di p variabili indipendenti rappresentate da X .

Nel caso della regressione lineare classica, si minimizza la funzione di costo (errore quadratico medio):

$$J(\beta) = \frac{1}{2N} \sum_{i=1}^N \left(y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2$$

dove:

- N è il numero di osservazioni.
- x_{ij} rappresenta il valore della j -esima feature per la i -esima osservazione.
- y_i è il valore osservato della variabile dipendente.

Aggiunta della Regolarizzazione L1

Il metodo Lasso introduce un termine di penalizzazione basato sulla norma L1 dei coefficienti, modificando la funzione di costo come segue:

$$J(\beta) = \frac{1}{2N} \sum_{i=1}^N \left(y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

dove:

- $\lambda \geq 0$ è il parametro di regolarizzazione che controlla l'intensità della penalizzazione.

Un λ elevato aumenta la penalizzazione, spingendo molti coefficienti a zero e quindi escludendo i relativi feature. Un λ vicino a zero comporta una penalizzazione minima, avvicinando il modello a una regressione lineare ordinaria senza selezione delle feature.

4.3.3 K-Fold Cross-Validation

La validazione incrociata K-fold valuta la capacità di generalizzazione del modello riducendo overfitting e bias di selezione. Essa consiste nel suddividere il set di dati in K "Fold", dove ciascun fold viene utilizzato a turno come set di test, mentre gli altri fold vengono impiegati per l'addestramento del modello. In ogni iterazione, il modello viene addestrato su dati di training e testato su dati di test e la performance complessiva del modello viene calcolata come la media dei punteggi ottenuti nelle diverse iterazioni.

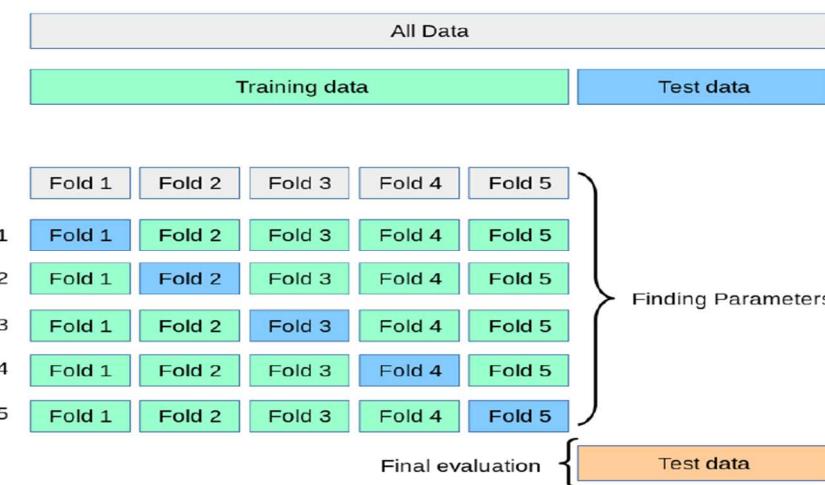


Figura 15 Funzionamento K-fold Cross-Validation

Il numero di fold, che può variare da 5 o 10, viene scelto in base alla dimensione del set di dati, con il 20% dei dati riservato per il test (Figura 15). La *K-fold Cross-Validation* è utile per ridurre il rischio di overfitting, poiché ogni campione del set di dati viene utilizzato sia per l'addestramento che per il test, evitando che il modello memorizzi caratteristiche specifiche dei dati di training.

4.3.4 Parametri di Addestramento:

Per ottimizzare l'addestramento dei modelli di DL, migliorarne le prestazioni e ridurre la loss, è fondamentale adottare tecniche avanzate di fine-tuning. A questo scopo, è essenziale approfondire concetti chiave come l'uso delle epoche, del learning rate, degli ottimizzatori Adam e AdamW, del dropout, della binary cross-entropy e della batch normalization. Questi strumenti sono indispensabili per ottenere risultati di alta qualità.

4.3.5 Epoche

Il concetto di Epoca corrisponde a un ciclo completo di elaborazione durante il quale il modello analizza per una volta tutti i campioni del dataset di training. Durante ciascuna epoca,

il modello esegue aggiornamenti dei pesi basandosi su mini-batch (sottoinsiemi del dataset) per ridurre la funzione di loss.

A differenza della cross-validation, che richiede la suddivisione del dataset in più sottoinsiemi per valutare iterativamente la capacità di generalizzazione del modello, l'uso delle Epoche permette di monitorare e ottimizzare le prestazioni attraverso ripetute esposizioni ai dati. In particolare, combinando l'addestramento su più epochhe con tecniche come l'early stopping e il monitoraggio della perdita su un set di validazione separato, è possibile ottenere stime affidabili della generalizzazione senza dover ricorrere a complesse procedure di cross-validation, che possono risultare computazionalmente onerose in presenza di dataset di grandi dimensioni e di modelli con un elevato numero di parametri. Pertanto, in tali scenari, le Epoche possono essere considerate un metodo alternativo e pragmatico per garantire un'efficace validazione del modello durante il processo di training. Ad esempio, se il dataset contiene N campioni e si utilizza un mini-batch di dimensione B , allora il numero di aggiornamenti (o iterazioni) per epoca è dato da:

$$\text{Numero di iterazioni per epoca} = \frac{N}{B}$$

Il numero totale di Epoche è un iperparametro scelto durante l'addestramento per assicurare un'esposizione adeguata ai dati, evitando sia l'underfitting che l'overfitting. Una gestione ottimale delle epochhe permette al modello di iterare sufficientemente sui dati per apprendere le caratteristiche rilevanti.

4.3.6 Learning Rate

Il Learning Rate (η) è un iperparametro critico nei modelli di DL, poiché determina l'entità degli aggiornamenti applicati ai pesi della rete durante l'ottimizzazione. Un valore troppo elevato può portare a oscillazioni o divergenza, mentre uno troppo ridotto rallenta la convergenza o causa arresti in minimi locali. Nella pratica, il learning rate modula la discesa del gradiente secondo la relazione:

$$w_{t+1} = w_t - \eta \nabla \mathcal{L}(w_t)$$

dove w_t rappresenta i pesi al passo t e $\nabla \mathcal{L}$ è il gradiente della funzione di perdita.

4.3.7 Ottimizzatori: Adam, AdamW

Adam (Adaptive Moment Estimation) è un algoritmo di ottimizzazione che combina i vantaggi di due metodi: il momentum e la normalizzazione del gradiente. Adam adatta i tassi

di apprendimento per ciascun parametro in base alle prime due stime dei momenti (media e varianza) dei gradienti. Le formule principali per l'aggiornamento dei parametri θ in Adam sono:

Calcolo della media mobile dei gradienti:

$$m_t = \beta_1 m_{t-1} + (1 + \beta_1) g_t$$

Calcolo della media mobile dei quadrati dei gradienti:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

dove:

- g_t è il gradiente della funzione di loss rispetto al parametro θ al tempo t .
- β_1 e β_2 sono i parametri di decadimento esponenziale (tipicamente $\beta_1 \approx 0.9$ e $\beta_2 \approx 0.999$).

Correzione del bias delle stime:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Aggiornamento del parametro:

$$\theta_{t+1} = \theta_t - \alpha (\hat{m}_t) / (\sqrt{\hat{v}_t} + \epsilon)$$

dove:

- α è il tasso di apprendimento.
- ϵ è un piccolo valore (ad esempio 10^{-8}) per evitare la divisione per zero.

AdamW è una variante dell'ottimizzatore Adam che corregge alcune problematiche relative alla regolarizzazione con decoupled weight decay. Mentre in Adam la penalizzazione dei pesi (weight decay) è integrata nella stima dei momenti, AdamW separa esplicitamente questo termine, migliorando la regolarizzazione e la convergenza del modello.

L'aggiornamento in AdamW viene quindi modificato aggiungendo un termine di decadenza dei pesi:

$$\theta_{t+1} = \theta - \alpha \left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} - \lambda \theta_t \right)$$

dove λ è il coefficiente del weight decay. Questa separazione permette una gestione più efficace della regolarizzazione, soprattutto in reti neurali profonde.

L'uso degli ottimizzatori avanzati come Adam e AdamW, permette di adattare dinamicamente il tasso di apprendimento e di migliorare la convergenza durante l'addestramento. Le formule esposte evidenziano come questi algoritmi combinino informazioni sui gradienti e le loro varianze, garantendo aggiornamenti dei parametri robusti e stabili, fondamentali per il successo dei modelli di DL.

4.3.8 Tecniche di Regularization: Dropout, Batch Normalization

Il dropout è una tecnica di regolarizzazione che consiste nell'escludere casualmente alcune unità e le loro connessioni durante il calcolo del gradiente e l'aggiornamento dei pesi, con una probabilità definita, impedendo così che le unità si adattino troppo ai dati di addestramento, favorendo una maggiore generalizzazione.

La batch normalization è una tecnica che applicata durante il training, normalizza gli input di ciascun layer per garantire che abbiano una distribuzione con media zero e varianza uno all'interno di ogni mini-batch. Questo approccio aiuta a mitigare il fenomeno della covarianza interna, ovvero la variabilità delle distribuzioni degli input ai layer interni durante l'addestramento, che può rallentare la convergenza e rendere il modello sensibile a iperparametri come il tasso di apprendimento e l'inizializzazione dei pesi.

Relativamente alla batch normalization è corretto aggiungere ancora alcune considerazioni. In primo luogo la batch normalization permette di accelerare la convergenza, riduce la necessità di piccoli tassi di apprendimento e consentendo di utilizzare valori più elevati e favorisce una più rapida minimizzazione della loss. In secondo luogo la batch normalization stabilizza l'addestramento, controllando la varianza degli input, in questo modo riduce il rischio di oscillazioni nei gradienti e facilita un aggiornamento più regolare dei parametri. In terzo luogo agisce come regolarizzatore in quanto grazie al calcolo sui mini-batch introduce una leggera variabilità che può avere un effetto regolarizzante, contribuendo a prevenire l'overfitting.

4.3.9 Funzione di Costo: Binary Cross-Entropy

La funzione di perdita binary cross entropy è particolarmente indicata per problemi di classificazione binaria, dove il compito è discriminare tra due classi distinte. Questa funzione misura la distanza tra le probabilità predette dal modello e le etichette reali, assegnando penalità

maggiori alle predizioni più lontane dal target. Matematicamente, la binary cross entropy è definita come:

$$L(y, \hat{y}) = -[y * \log(\hat{y}) + (1 - y) * \log(1 - \hat{y})]$$

dove:

- y rappresenta l'etichetta reale (0 o 1).
- \hat{y} è la probabilità predetta per la classe positiva.

Questa funzione offre una serie di vantaggi, in primo luogo incoraggia previsioni corrette Penalizzando in modo asimmetrico gli errori, inducendo il modello a produrre output che si avvicinino il più possibile ai valori target. Secondariamente attraverso la derivata della binary cross entropy, in combinazione con tecniche di backpropagation, ottimizza l'aggiornamento dei pesi, contribuendo alla minimizzazione della loss; in terzo luogo in contesti di classificazione binaria si integra bene con l'uso di funzioni di attivazione come la sigmoide, con la binary cross entropy (che mappa i valori in un intervallo [0,1]).

4.3.10 Bayesian Optimization

Il tuning dei parametri, consiste nella ricerca degli iperparametri ottimali di tuning che servono per massimizzare le performance dei modelli di ML. Ad esempio, in una Random Forest, un iperparametro cruciale da ottimizzare è il numero massimo di foglie per ciascun albero, poiché questo parametro incide direttamente sulla capacità del modello di catturare complessità nei dati senza incorrere in overfitting.

In questo contesto, la Bayesian Optimization si presenta come un approccio sistematico ed efficiente per esplorare lo spazio degli iperparametri. L'idea alla base della Bayesian Optimization è quella di modellare la funzione obiettivo $f(x)$ che rappresenta, ad esempio, la performance del modello, in funzione dei suoi iperparametri x tramite un modello probabilistico, comunemente un Gaussian Process (GP). In particolare, si assume che:

$$f(x) \sim GP(m(x), k(x, x'))$$

dove $m(x)$ è la funzione media e $k(x, x')$ è la funzione di covarianza, che cattura la correlazione tra i valori di $f(x)$ in punti diversi dello spazio degli iperparametri.

Per guidare la ricerca, si utilizza una funzione di acquisizione $\alpha(x)$, che bilancia l'esplorazione di nuove aree dello spazio con lo sfruttamento delle regioni già identificate come

promettenti. Un esempio comune di funzione di acquisizione è l'Expected Improvement (EI), definita come:

$$EI(x) = E[\max(0, f(x) - f(x^+))]$$

dove $f(x^+)$ rappresenta il miglior valore osservato fino a quel momento. Questa funzione valuta il potenziale miglioramento che un nuovo set di iperparametri x potrebbe offrire, aiutando a decidere il prossimo punto da testare.

La Bayesian Optimization rappresenta una metodologia estremamente versatile, applicabile a una vasta gamma di modelli, e consente di ottimizzare in modo automatizzato e iterativo gli iperparametri specifici di ciascun modello. Questo approccio permette di esplorare in maniera efficiente lo spazio degli iperparametri, migliorando la capacità predittiva e la generalizzazione del modello, oltre a rendere il processo di tuning più mirato ed efficace. L'integrazione di queste tecniche contribuisce, inoltre, a una gestione ottimale della suddivisione dei dati, alla prevenzione dell'overfitting, alla stabilizzazione e all'accelerazione della convergenza, migliorando complessivamente le prestazioni del modello.

4.4 Valutazione e Interpretazione dei Modelli

Nell'ambito della classificazione, e in particolare nei test diagnostici, l'obiettivo principale è quello di distinguere correttamente tra soggetti appartenenti alle diverse classi del target. Per ottenere questa classificazione si utilizzano le metriche di prestazione impiegate per valutare l'efficacia dei modelli di classificazione. In un test ideale le due popolazioni relative alla classe del target sarebbero perfettamente separate (sopravvissuti vs deceduti); tuttavia, nella pratica si osserva spesso una sovrapposizione che porta alla presenza di falsi positivi (FP) e falsi negativi (FN). Per analizzare e interpretare questi errori si utilizza una la cosiddetta Confusion Matrix o Matrice di Confusione.

4.4.1 Confusion Matrix e Metriche di Valutazione

La Matrice di Confusione è una rappresentazione tabellare che sintetizza i risultati ottenuti dal modello (Figura 16), al suo interno si possono identificare quattro categorie di casi:

- Veri Positivi (TP): il numero di casi in cui il modello ha correttamente identificato un soggetto ad alto rischio (ad esempio, pazienti deceduti in un contesto ospedaliero).
- Veri Negativi (TN): il numero di casi in cui il modello ha correttamente identificato un soggetto a basso rischio (ad esempio, pazienti sopravvissuti).

- Falsi Positivi (FP): il numero di errori in cui il modello ha classificato erroneamente come ad alto rischio soggetti che in realtà appartengono alla categoria a basso rischio.
- Falsi Negativi (FN): il numero di errori in cui il modello ha classificato erroneamente come a basso rischio soggetti che in realtà appartengono alla categoria ad alto rischio.

		Actual	
		Positive	Negative
Predicted	Positive	TP	FN
	Negative	FP	TN

Figura 16 Confusion Matrix

Queste definizioni possono variare a seconda del contesto applicativo, ma rappresentano il punto di partenza per il calcolo di diverse metriche di performance. Per valutare l'efficacia di un classificatore, vengono comunemente utilizzate le seguenti metriche:

- Accuratezza (AC): l'accuratezza misura la qualità complessiva del modello ed è definita come il rapporto tra il numero di casi correttamente classificati e il numero totale di casi:

$$AC = \frac{TP + TN}{TP + FP + TN + FN}$$

- Misclassification Error (Loss): il tasso di errore, spesso indicato anche come loss in alcuni contesti applicativi. Questa misura quantifica la percentuale di predizioni errate rispetto al totale e viene espressa dalla seguente formula:

$$\text{Tasso di Errore} = \frac{FP + FN}{TP + TN + FP + FN}$$

dove FP (falsi positivi) e FN (falsi negativi) rappresentano, rispettivamente, i casi in cui il modello ha erroneamente assegnato una classificazione positiva o negativa. Un tasso di errore elevato indica una maggiore incidenza di predizioni sbagliate, mentre un valore basso evidenzia una migliore capacità del modello di distinguere correttamente tra le classi.

- Sensibilità (Recall o True Positive Rate - TPR): la sensibilità indica la capacità del modello di identificare correttamente i soggetti ad alto rischio. Si calcola come il rapporto tra i veri positivi e la somma di veri positivi e falsi negativi:

$$Recall = SENS = TPR = \frac{TP}{FN + TP}$$

- Specificità (True Negative Rate - TNR): la specificità misura la capacità del test di riconoscere correttamente i soggetti a basso rischio, ed è definita come il rapporto tra i veri negativi e la somma di veri negativi e falsi positivi:

$$SPEC = TNR = \frac{TN}{TN + FP}$$

- Precision: la precisione valuta la capacità del modello di identificare correttamente solo i casi realmente positivi, ossia il rapporto tra veri positivi e il totale dei casi classificati come positivi (somma di TP e FP):

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

- F1-Score: l'F1-score è una misura che combina la precisione e la sensibilità in un'unica metrica, risultando particolarmente utile quando si vuole bilanciare il trade-off tra questi due aspetti. Varia tra 0 e 1, con valori più elevati che indicano migliori prestazioni:

$$F1score = \frac{Recall * Precision}{Recall + Precision} * 2$$

4.4.2 Analisi Grafica: ROC, PRC, Learning Curves

Oltre alle metriche numeriche, esistono rappresentazioni grafiche che risultano particolarmente utili per confrontare e ottimizzare i modelli di classificazione:

- La Curva ROC (Receiver Operating Characteristics) rappresenta la sensibilità (TPR) in funzione del tasso di falsi positivi (1-specificità) per differenti valori di soglia. L'area sotto la curva (AUC) sintetizza la capacità discriminante del modello, con valori vicini a 1 che indicano un'elevata capacità di distinguere tra soggetti ad alto e basso rischio.
- La Precision Recall Curve (PRC) è particolarmente utile in contesti con dataset sbilanciati, poiché si focalizza sulla capacità del modello di prevedere correttamente la classe minoritaria. Essa traccia la relazione tra precisione e recall per vari valori di soglia, con una curva ideale che si avvicina all'angolo in alto a destra. In questo contesto, l'Average Precision (AP) funge da metrica sintetica che riassume la performance complessiva, calcolando la media ponderata delle precisioni ottenute ad ogni incremento di recall. Un valore di AP vicino a 1 indica una notevole capacità del

modello di identificare correttamente la classe di interesse, rendendo questa misura particolarmente preziosa per valutare modelli in scenari diagnostici e in presenza di squilibri tra le classi.

- Le learning curves (Figura 17) sono uno strumento grafico essenziale per comprendere e migliorare il comportamento dei modelli di ML. Attraverso l'analisi visiva del trend degli errori (o di un'altra metrica di performance) sul training e sul test set, è possibile identificare eventuali problemi di overfitting o underfitting, valutare la necessità di ulteriori dati e ottimizzare i parametri del modello per ottenere prestazioni migliori.

La scelta tra ROC e PRC dipende dalla natura del dataset, quando si tratta di identificare correttamente i pazienti ad alto rischio o basso rischio in un contesto di mortalità ospedaliera, entrambe le tecniche sono fondamentali per ottimizzare i modelli di classificazione.

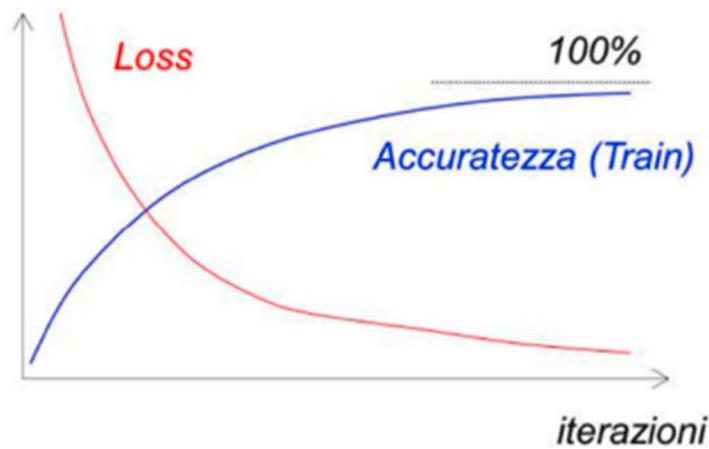


Figura 17 obiettivo delle Learning Curves

Il calcolo delle learning curves prevede generalmente i seguenti passaggi:

- Suddivisione del Training Set: Il dataset di training viene diviso in sottoinsiemi di dimensioni crescenti.
- Addestramento del Modello: Per ciascun sottoinsieme, il modello viene addestrato e, al termine di ogni ciclo di addestramento, si calcola l'errore o la metrica della performance scelta. Questo calcolo viene effettuato sia sul training-set sia sul test-set separato. I valori di errore o di performance vengono memorizzati per ogni dimensione del training set (o per ogni epoca) e per ciascun set (training e test).
- Tracciamento del Grafico: infine, si realizza un grafico dove l'asse delle ascisse rappresenta la dimensione del training set (o il numero di epoche) e l'asse delle ordinate

indica l'errore o la metrica di performance. La sovrapposizione delle due curve permette di visualizzare come evolve l'apprendimento del modello.

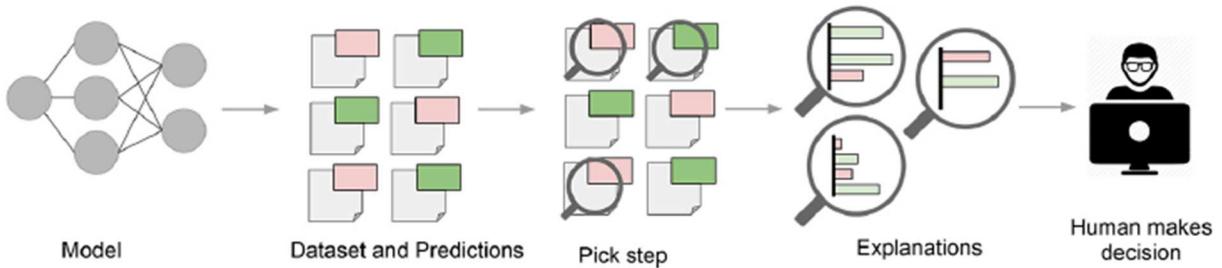
In sintesi la valutazione accurata di un modello di apprendimento automatico richiede l'impiego congiunto di metriche quantitative e strumenti grafici, che consentono di avere una panoramica completa delle prestazioni del modello e di comprendere il comportamento del modello a fronte delle differenti soglie decisionali, al fine di risultare particolarmente utile in contesti diagnostici dove l'identificazione corretta dei soggetti ad alto rischio è cruciale.

4.4.3 Explainable Artificial Intelligence (XAI)

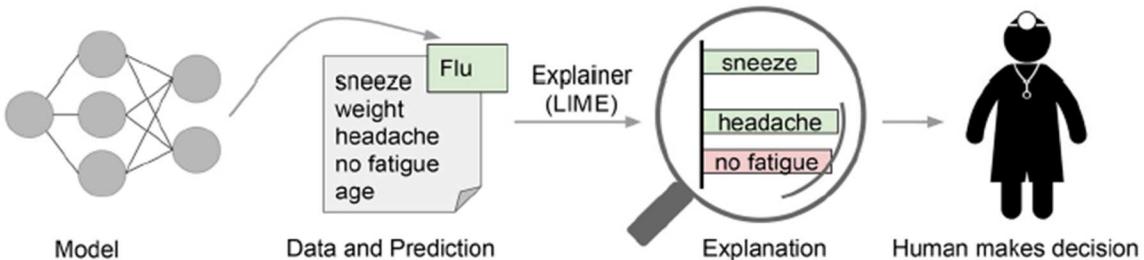
L'impiego sempre più diffuso di tecniche di AI, ML e DL ha sollevato il problema della “black box”, ovvero la difficoltà di comprendere il funzionamento interno degli algoritmi, specialmente quelli più complessi come le reti neurali adottate nel DL.

Questi modelli, costituiti da migliaia di neuroni e interconnessioni, elaborano le informazioni attraverso processi complessi e non direttamente interpretabili dagli esseri umani, rendendo difficile tracciare il ragionamento che conduce a una specifica decisione. Questo aspetto risulta particolarmente critico in ambito medico, dove le predizioni generate dall'intelligenza artificiale possono influenzare decisioni cliniche di fondamentale importanza. La scarsa interpretabilità di un modello non solo può ridurre la fiducia degli operatori sanitari, ma anche amplificare eventuali bias presenti nei dati di addestramento, con potenziali implicazioni etiche e operative significative (Figura 18).

Per risolvere tali problematiche è nata l'eXplainable AI (XAI), un ambito di ricerca che mira a migliorare la trasparenza e l'interpretabilità degli algoritmi, consentendo di comprendere le scelte effettuate dal modello, individuare errori e favorire una maggiore affidabilità e adozione in termini comprensibili per gli esseri umani. (13,25,26,27,28,29,30)



(a) Explaining a model to a human decision-maker



(b) Explaining individual predictions to a human decision-maker

Figura 18 esempi di modello e spiegazione individuale

Una strategia efficace per ottenere spiegazioni interpretabili consiste nell'utilizzo degli Additive Feature Attribution Methods. Questi metodi consentono di decomporre l'output di un modello complesso in termini interpretabili, mostrando come ogni feature contribuisca in modo additivo alla previsione finale.

La formulazione generale è la seguente:

$$\hat{y} = \phi_0 + \sum_{j=1}^n \phi_j x_j$$

dove:

- ϕ_0 rappresenta il valore atteso (baseline) della predizione.
- x_j indica il valore della j -esima feature.
- ϕ_j è il contributo associato alla feature j .

In questo contesto, il valore assoluto dei pesi $|\phi_j|$ può essere utilizzato per valutare l'importanza di ciascuna feature a livello globale, calcolando la media dei contributi assoluti su un insieme di istanze:

$$I_j = \frac{1}{m} \sum_{i=1}^m |\phi_j^{(i)}|$$

Questa formulazione permette di ottenere sia spiegazioni locali, riferite a singole predizioni, sia una visione globale dell'impatto complessivo delle feature sul modello. L'uso del valore assoluto garantisce infatti che l'importanza di una feature sia considerata indipendentemente dalla direzione del suo effetto (positivo o negativo) sulla previsione.

Un aspetto cruciale degli Additive Feature Attribution Methods è che essi rispettino tre proprietà fondamentali, indispensabili per garantire l'affidabilità delle spiegazioni:

- Local Accuracy (Accuratezza Locale), La somma dei contributi delle feature deve riprodurre esattamente la predizione del modello interpretabile g per una specifica istanza x' . In termini formali, vale:

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^M \phi_i x'_i$$

- Missingness (Mancanza di Feature e Peso Nullo), Se una feature risulta assente nell'input semplificato x' (cioè, $x'_i = 0$), il suo contributo deve essere nullo:

$$x'_i = 0 \Rightarrow \phi_i = 0$$

- Consistency (Consistenza delle Feature), Se, in un modello modificato, l'impatto di una feature aumenta rispetto al modello originale, il suo contributo nell'attribuzione finale non deve diminuire. Ciò significa che se l'influenza di una feature aumenta, il suo contributo nel modello interpretabile deve riflettere tale cambiamento senza ridursi. Formalmente:

$$f_{x'}(z') - f_{x'}(z' \setminus \{i\}) \geq f_x(z') - f_x(z' \setminus \{i\}) \Rightarrow \phi_{-i}(f', x) \geq \phi_{-i}(f, x)$$

4.4.4 LIME (Local Interpretable Model-Agnostic Explanations)

Il metodo LIME è un esempio di Additive Feature Attribution Methods, sviluppato per fornire spiegazioni interpretabili di modelli complessi “black-box”, concentrandosi sull'analisi locale delle decisioni. Esaminando il processo decisionale per una singola istanza, permette di evidenziare il contributo di ciascuna feature nella specifica predizione.

L'approccio di LIME si basa sulla creazione di modelli surrogati, ovvero modelli semplici e interpretabili (come modelli lineari o alberi decisionali) progettati per approssimare il comportamento del modello complesso in una regione ristretta dello spazio dei dati. Per costruire il modello surrogato, LIME perturba l'input originale generando dati sintetici simili e osserva come varino le predizioni del modello black-box. Questi campioni perturbati vengono poi utilizzati per addestrare il modello surrogato, che cerca di replicare il comportamento del

modello complesso soltanto in prossimità della predizione analizzata, garantendo così l'accuratezza locale (local accuracy). La funzione di ottimizzazione che guida questa costruzione è definita come:

$$\xi(x) = \arg \min_{g \in G} [L(f, g, \pi_x) + \Omega(g)]$$

dove $L(f, g, \pi_x)$ misura la fedeltà del modello surrogato g rispetto al modello originale f in una regione locale definita da π_x , mentre $\Omega(g)$ rappresenta un termine di regolarizzazione che vincola la complessità del modello, garantendone l'interpretabilità.

LIME può essere applicato a diversi tipi di dati – tabulari, testuali e immagini – offrendo spiegazioni intuitive e facilmente comprensibili, sebbene la sua approssimazione lineare possa non catturare relazioni non lineari complesse in alcuni contesti. Nel contesto clinico, l'interpretabilità delle decisioni dei modelli di ML è cruciale. L'utilizzo di XAI rappresenta una strategia efficace per superare la natura “black box” dei modelli complessi. Questi strumenti permettono di interpretare le scelte effettuate dai modelli, di individuare errori e di supportare decisioni critiche, migliorando così la trasparenza, l'affidabilità e l'adozione dell'AI in contesti sensibili come quello clinico.

5. Risultati

In questa sezione saranno presentati e analizzati i risultati ottenuti dall'applicazione di diverse metodologie di ML e NLP al dataset in esame. L'obiettivo è sfruttare in modo ottimale sia le informazioni strutturate (es. indicatori medici, dati anagrafici) sia quelle non strutturate (es. diagnosi testuali) per la classificazione dei soggetti in base al rischio di mortalità, valutando le prestazioni dei modelli implementati (31,32,33,34).

Il percorso espositivo si articola inizialmente con una dettagliata analisi esplorativa dei dati con statistiche descrittive e grafici, per poi passare alla presentazione comparativa dei modelli sviluppati. In particolare, sono stati sperimentati quattro approcci distinti: il primo applica modelli ML classici sui dati tabulari; il secondo integra le rappresentazioni semplici testuali all'interno di modelli ML sui dati non strutturati; il terzo sfrutta modelli avanzati di NLP per l'analisi esclusiva dei dati testuali; il quarto combina in maniera ibrida le informazioni provenienti da entrambe le tipologie di dati. Per ogni approccio sono state adottate le necessarie tecniche di fine tuning e di regolarizzazione – che hanno garantito l'ottimizzazione delle performance e la stabilità dei risultati. Infine, sono stati utilizzati strumenti di Explainable AI per rendere trasparente il funzionamento dei modelli.

In questa sezione verranno presentate in modo approfondito e comparativo le performance e le caratteristiche distintive dei diversi approcci adottati. L'analisi metterà in evidenza il contributo di ciascun metodo nella comprensione e nell'ottimizzazione delle informazioni contenute nelle SDO, evidenziando il loro impatto sulla qualità della classificazione del rischio di mortalità.

5.1 Ambiente Cloud, Strumenti e Librerie Utilizzate

Per eseguire tutte le analisi, è stato impiegato un abbonamento a Google Colab Pro Plus. Google Colab, abbreviazione di *Google Collaboratory*, è un ambiente di sviluppo online basato su Jupyter Notebook che permette di scrivere ed eseguire codice Python direttamente nel browser. Per definizione, si caratterizza per:

- Esecuzione in Cloud: Non richiede installazioni locali, poiché il codice viene eseguito sui server di Google.
- Accesso a Risorse Hardware Avanzate: Offre CPU, GPU e TPU, facilitando il lavoro con algoritmi complessi e modelli di ML.

- Collaborazione e Condivisione: È integrato con Google Drive, permettendo una facile condivisione e collaborazione sui progetti.
- Interattività: Consente di combinare codice, testo, grafici e dati in un unico documento interattivo.

Questi elementi, uniti alle funzionalità avanzate del servizio, hanno permesso di gestire elaborazioni complesse in tempi brevi e con maggiore efficienza, facilitando l'utilizzo del linguaggio di programmazione Python, strumento fondamentale per lo sviluppo di applicazioni di ML, analisi dei dati e NLP.

Python si distingue per la sua natura interpretata, orientata agli oggetti e per una sintassi chiara e intuitiva, rendendolo ideale in ambito data science, intelligenza artificiale e DL. La sua flessibilità consente di adottare paradigmi procedurali, funzionali e orientati agli oggetti, ed è supportato da una community dinamica che offre ampie risorse, tra cui documentazione, tutorial e assistenza tecnica.

Tra le librerie più rilevanti per l'analisi dei dati, Pandas fornisce strutture dati potenti e flessibili per manipolare e analizzare dati etichettati o relazionali, coprendo l'intero ciclo del data processing. NumPy rappresenta il pilastro per l'elaborazione numerica grazie agli array multidimensionali e alle funzioni matematiche avanzate. Per la visualizzazione dei dati, Matplotlib consente di creare grafici bidimensionali, mentre Seaborn, basandosi su Matplotlib, offre strumenti per la generazione di grafici statistici integrati con i dati di Pandas. In ambito ML, Scikit-learn è una libreria fondamentale che offre un'ampia gamma di algoritmi per l'apprendimento supervisionato e non supervisionato, supportando ogni fase dello sviluppo dei modelli: dal preprocessing (tramite *sklearn.preprocessing*), alla selezione degli iperparametri (con *sklearn.model_selection*), fino alla valutazione delle prestazioni (con *sklearn.metrics*). Per il DL, TensorFlow si presenta come una piattaforma end-to-end che consente di sviluppare e distribuire modelli avanzati, integrando Keras, un'API di alto livello per la costruzione e gestione di reti neurali organizzate in layers e modelli complessi. Parallelamente, PyTorch sfrutta la potenza delle GPU per accelerare l'addestramento dei modelli, garantendo flessibilità e velocità, mentre librerie come gensim e NLTK offrono strumenti specifici per il ML applicato all'analisi testuale. Inoltre, la libreria Transformers permette di utilizzare modelli pre-addestrati per numerose applicazioni NLP, facilitando l'adozione di soluzioni all'avanguardia.

Grazie a questo vasto ecosistema di strumenti e all'integrazione facilitata dal servizio cloud, Python si configura come la piattaforma ideale per sviluppare modelli predittivi basati

su tecniche di ML e NLP, applicabili anche a dataset complessi, come le SDO italiane per la valutazione della mortalità ospedaliera.

5.2 Analisi Descrittiva

5.2.1 caratteristiche del campione

Le caratteristiche dei ricoveri dei pazienti mediante statistiche descrittive sono presentate nella tabella 4.

Statistiche descrittive	
Femmine n. (%)	1.911.949 (55,36%)
Maschi n. (%)	1.541.621 (44,63%)
Età—anni medi (DS)	68,2 (13,02)
Durata mediana della degenza ospedaliera (Q1-Q3)	7 (4-12)
Punteggio Elixhauser, media (DS)	1,39 (1,49)
Numero parole diagnosi, media (DS)	15,01 (9,62)
regione— n. (%)	
Piemonte	280.333 (8,11%)
Lombardia	635.512 (18,4%)
Veneto	290.158 (8,4%)
Emilia Romagna	271.872 (7,87%)
Toscana	252.493 (7,31%)
Lazio	318.113 (9,21%)
Puglia	213.471 (6,18%)
Campania	264.054 (7,64%)
Sicilia	266.195 (7,7%)
Tutte le altre regioni	661.346(19,14%)
Mortalità Ospedaliera	168.395(4,9%)

Tabella 4 Caratteristiche del campione

Tra i 3.453.570 pazienti inclusi, il 55,4% (n = 1.911.949) sono di sesso femminile e il 44,6% (n = 1.541.621) di sesso maschile. Il tasso di mortalità è pari al 4,9% (n = 168.395). La distribuzione dei pazienti per regione evidenzia una concentrazione più elevata in alcune aree, come la Lombardia (18,4%), seguita dal Lazio (9,21%) e dal Piemonte (8,11%). Le statistiche descrittive evidenziano che:

- Età (eta): La media è di circa 68 anni, con un minimo di 18 e un massimo di 115 anni; il 25° percentile è 61, il 50° percentile (mediana) 70 e il 75° percentile 77 anni.

- Durata della degenza (los): La durata media di ricovero è di circa 9,69 giorni, con un range che va da 0 a 862 giorni; il 25° percentile è di 4 giorni, la mediana di 7 e il 75° percentile di 12 giorni.
- Indice di Elixhauser (elixsum): Il valore medio è di 1.39, con valori che spaziano da 0 a 31; il 25° percentile è 0, la mediana 1 e il 75° percentile 2.
- Lunghezza delle diagnosi (num_words): In media, le diagnosi contengono circa 15 parole (minimo 0, massimo 136), con il 25° percentile a 7 parole, la mediana a 13 e il 75° percentile a 21 parole.

Questi indicatori suggeriscono che, a livello complessivo, il 50% del campione presenta diagnosi composte da almeno 13 parole, ha un'età pari o superiore a 70 anni e un periodo di ricovero mediano pari a 7 giorni.

Le statistiche descrittive stratificate per sesso mostrano che:

- Età e Durata del Ricovero: Non emergono differenze sostanziali nella media dell'età e nella durata del ricovero tra maschi e femmine.
- Indice Elixhauser e Lunghezza Diagnostica: I pazienti maschi presentano valori medi leggermente superiori sia per l'indice Elixhauser (1.74 contro 1.29 nelle femmine) sia per la lunghezza delle diagnosi (15.74 parole contro 14.07 nelle femmine).
- Estremi dei Valori: Si osserva una differenza nei giorni di degenza massimi, con le femmine che raggiungono fino a 862 giorni rispetto ai 735 giorni dei maschi.
- Mediana: Il 50° percentile indica che i maschi hanno un'età mediana leggermente superiore (71 anni) rispetto alle femmine (69 anni) e che le diagnosi maschili contengono in media 14 parole, mentre in quelle femminili il numero mediano è di 11 parole.

Le statistiche descrittive delle variabili analizzate, filtrate per anno di referto (2012–2016):

- Età (eta): Nei vari anni il numero di pazienti analizzati è simile (tra circa 684.900 e 698.842 osservazioni per anno). L'età media si attesta in maniera stabile, variando da 67,95 anni nel 2012 a 68,41 anni nel 2016, con una deviazione standard intorno a 13 anni. Il valore minimo è costante a 18 anni, mentre il massimo oscilla tra 112 e 115 anni (eccetto il 2016, dove il massimo risulta 106). I percentili indicano una distribuzione

abbastanza uniforme: il 25° percentile si attesta intorno ai 61 anni e la mediana (50° percentile) intorno ai 70 anni, mentre il 75° percentile varia tra 77 e 78 anni.

- Durata della degenza (los): La durata media del ricovero varia leggermente tra gli anni, con valori medi compresi tra 9,47 giorni (2016) e 9,93 giorni (2012) e una deviazione standard di circa 10 giorni. Il 25° percentile è costantemente a 4 giorni e la mediana a 7 giorni per ogni anno. Il 75° percentile risulta generalmente intorno ai 11–12 giorni. I valori massimi, tuttavia, mostrano una notevole variabilità: dal valore più elevato di 862 giorni registrato nel 2013, fino a 549 giorni (2014) e 435 giorni (2015), mentre nel 2012 e 2016 i massimi sono rispettivamente 735 e 741 giorni.
- Indice di Elixhauser (elixsum): Questo indice presenta una media che si attesta intorno a 1,44 nel 2012 e 2013, per poi diminuire progressivamente fino a 1,30 nel 2016, con una deviazione standard intorno a 1,50 in ciascun anno. Il 25° percentile è pari a 0 e la mediana a 1 in tutti gli anni, mentre il 75° percentile è costantemente 2. Il valore massimo, stabile nel tempo, è pari a 31.
- Lunghezza delle diagnosi (num_words): Il numero medio di parole per diagnosi mostra una leggera diminuzione, passando da 15,27 parole nel 2012 a 14,74 parole nel 2016, con una deviazione standard attorno a 9,70. Il 25° percentile è costantemente a 7 parole e la mediana si attesta a 13 parole per il 2012, 2013, 2014 e 2015, mentre nel 2016 la mediana scende a 12 parole. Il 75° percentile varia intorno a 20–21 parole, mentre il valore massimo varia significativamente da 107 parole nel 2012 fino a 136 parole nel 2014.

L'analisi dei dati annuali evidenzia una notevole stabilità nelle caratteristiche demografiche (età) e cliniche (durata della degenza, indice di complessità e lunghezza delle diagnosi) dei pazienti ospedalizzati. Le variazioni osservate riguardano principalmente i valori estremi, in particolare per la durata del ricovero e per la lunghezza massima delle diagnosi, mentre i valori medi e i percentili restano abbastanza costanti nel periodo 2012–2016.

Le statistiche descrittive in base alla variabile target (sopravvissuti vs. deceduti) rivelano differenze significative:

- Età: L'età media dei pazienti deceduti è superiore (73.02 anni) rispetto a quella dei sopravvissuti (67.93 anni), con mediane rispettivamente di 75 e 70 anni.
- Durata della degenza: I deceduti hanno una durata media di ricovero di circa 13.91 giorni (con una deviazione standard elevata pari a 17.49), mentre i sopravvissuti

presentano una media di 9.47 giorni (DS = 9.42); le mediane risultano essere 9 giorni per i deceduti e 7 giorni per i sopravvissuti.

- Indice Elixhauser: Il valore medio nei deceduti è 2.18 (mediana pari a 2), ben superiore rispetto a quello dei sopravvissuti, pari a 1.35 (mediana di 1).
- Lunghezza delle diagnosi: Anche il numero di parole nelle diagnosi è maggiore nei deceduti, con una media di 20.76 parole (mediana di 20) contro 14.70 parole (mediana di 12) nei sopravvissuti.

Queste differenze indicano che i pazienti deceduti tendono ad essere più anziani, a subire ricoveri più prolungati e a ricevere diagnosi più dettagliate con un numero maggiore di parole, probabilmente riflesso di una maggiore complessità clinica.

Le analisi regionali mostrano variazioni significative nelle caratteristiche dei pazienti ospedalizzati:

- Età: La mediana dell'età varia tra le regioni, con valori compresi tra 67 e 73 anni. In particolare, per la Liguria (mediana 73 anni), Toscana e provincia autonoma di Trento (mediana 71 anni) presentano una popolazione più anziana.
- Durata della degenza: La mediana dei giorni di ricovero è generalmente compresa tra 7 e 8 giorni, tuttavia alcune regioni si distinguono per una gestione più prolungata dei ricoveri: il Molise presenta una mediana di 19 giorni, la Basilicata di 15 giorni e la Puglia di 14 giorni.
- Indice Elixhauser: La maggior parte delle regioni mostra una mediana pari a 1 o 2, ma il Molise si distingue per un valore mediano di 2, suggerendo una maggiore complessità clinica.
- Lunghezza delle diagnosi: Il numero mediano di parole varia tra 10 e 19. Il Molise (19 parole) e Basilicata (15 parole) evidenziano descrizioni diagnostiche particolarmente dettagliate, il che potrebbe riflettere una maggiore complessità diagnostica, considerando che le compilazioni delle cartelle di dimissione ospedaliera sono effettuate seguendo un'unica legenda standardizzata.
- Valori estremi: Alcune regioni, come il Molise, si distinguono per avere valori estremi sia in termini di durata del ricovero che di indice Elixhauser, mentre altre mostrano differenze anche nella lunghezza delle diagnosi.

Regioni con valori estremi:

- Regione del Molise si distingue per valori più elevati nella durata del ricovero (19 giorni di degenza mediana) e complessità clinica (Elixsum mediana di 2), suggerendo una popolazione di pazienti più critica.
- Regione della Basilicata mostra una durata del ricovero più alta rispetto alla media (15 giorni) e un numero maggiore di parole nelle diagnosi (15 parole mediana).
- Regione Toscana e la provincia autonoma di Trento presentano mediane di età più alte (71 anni), suggerendo una maggiore proporzione di pazienti anziani ospedalizzati.

Nel grafico 1 è riportata la matrice di correlazione per le variabili continue (età, durata della degenza, indice Elixhauser e numero parole nelle diagnosi), che evidenzia le seguenti relazioni:

- Età e durata della degenza ($r = 0,09$): Il coefficiente di correlazione pari a 0,09 evidenzia una relazione quasi nulla tra l'età dei pazienti e la durata del ricovero. Questo potrebbe suggerire che l'età, da sola, non è un fattore determinante nella lunghezza della degenza ospedaliera. È probabile che altre variabili – quali la gravità della patologia, le complicanze o le strategie di gestione clinica – esercitino un'influenza maggiore sul LOS.
- Età e Elixhauser ($r = 0,12$): Il valore del coefficiente ($r = 0,12$) indica una leggera correlazione positiva tra l'età dei pazienti e il punteggio Elixhauser, il quale rappresenta il carico di comorbidità e la complessità clinica. Questo legame suggerisce che, in media, i pazienti di età avanzata tendono ad accumulare un numero leggermente superiore di condizioni patologiche. Tuttavia, dato il carattere debole della correlazione, l'età spiega solo una piccola frazione della variabilità nel punteggio Elixhauser. Ciò implica che, pur essendo l'invecchiamento associato ad un aumento del rischio clinico, altri fattori – come lo stile di vita, l'accesso alle cure o la presenza di fattori genetici – possono avere un impatto più significativo sulla complessità clinica complessiva.
- durata della degenza e Elixhauser ($r = 0,10$): la correlazione positiva tra la durata del ricovero e il punteggio Elixhauser risulta molto debole. Questa associazione è di entità marginale e suggerisce che la complessità clinica rappresentata dall'Elixhauser incide solo parzialmente sulla durata della degenza. È quindi plausibile che altri elementi – come la gestione clinica, la tipologia del trattamento o l'efficienza dei percorsi di cura – possano essere determinanti nel definire la durata della degenza.

- numero parole nelle diagnosi ed Elixhauser ($r = 0,52$): Si osserva una correlazione molto positiva, come da attese all'aumentare del valore dell'indice Elixhauser – indicatore della complessità clinica e del carico di comorbidità – corrisponde un aumento del numero di parole nelle diagnosi. Questa relazione conferma che i pazienti con un maggior numero di condizioni cliniche o una maggiore complessità vengono descritti in maniera più dettagliata nelle cartelle di dimissione, probabilmente a causa della necessità di documentare accuratamente le numerose patologie presenti.
- numero parole nelle diagnosi ed età ($r = 0,11$): La correlazione positiva, seppur debole, tra il numero di parole nelle diagnosi e l'età suggerisce che con l'avanzare dell'età si tende a registrare descrizioni diagnostiche leggermente più lunghe.
- numero parole nelle diagnosi e durata della degenza ($r = 0,15$): Anche in questo caso, la correlazione è positiva ma debole, indicando che i pazienti con periodi di ricovero più lunghi tendono ad avere diagnosi leggermente più dettagliate (ossia, un numero maggiore di parole).

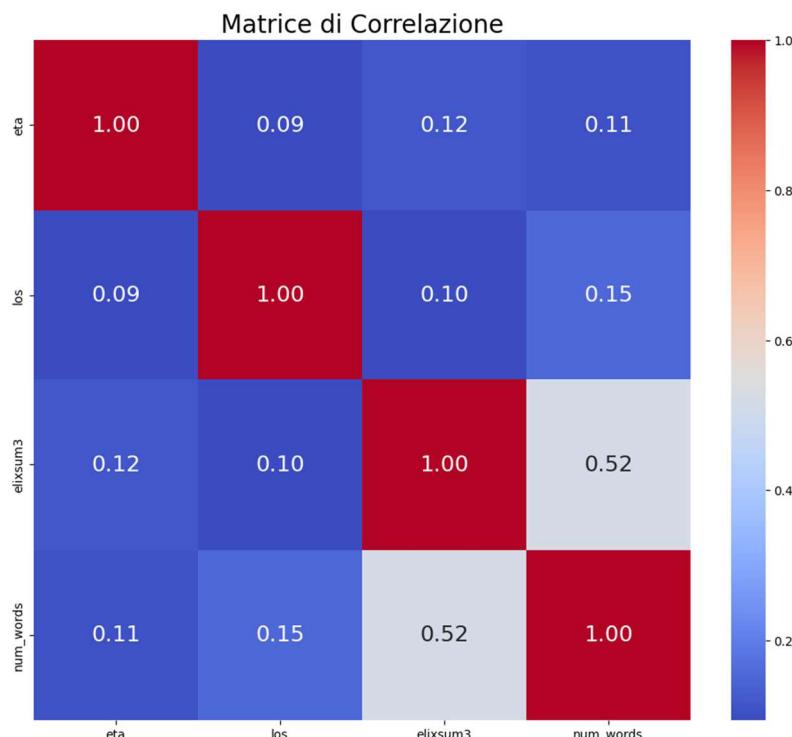


Grafico 1 matrice di correlazione variabili continue
(età, durata della degenza, indice Elixhauser e numero parole nelle diagnosi)

Come evidenziato in precedenza, l'associazione più significativa emersa dall'analisi è quella tra la lunghezza delle diagnosi e l'indice Elixhauser, mentre tutte le altre combinazioni mostrano correlazioni più deboli.

Diagnosi principali	N (%)	Decessi N (%)
Osteoartrosi, localizzata, primaria, parte inferiore della gamba	217.952 (6,31%)	119 (<0%)
Osteoartrosi, localizzata, primaria, regione pelvica e coscia	191.582 (5,55%)	153 (<0%)
Infarto subendocardico, episodio iniziale di cura	122.874 (3,55%)	1.518 (4,41%)
Tumori maligni della prostata	101.704 (2,9%)	2.368 (6,89%)
Ipertrofia (benigna) della prostata con ostruzione urinaria e altri LUTS	90.328(2,6%)	46 (<0%)

Tabella 5 frequenza Prevalenza delle 5 diagnosi Principali

Variabile	Sopravvissuto (N=3285175)	Decessi (N=168395)	Totale (N=3453570)
insufficienza cardiaca congestizia	114.971 (3,50%)	15.586 (9,26%)	130.557 (3,78%)
aritmie cardiache	172.722 (5,26%)	14.246 (8,46%)	186.968 (5,41%)
malattia vascolare	150.229 (4,57%)	6.643 (3,95%)	156.872 (4,54%)
disturbi della circolazione polmonare	18.721 (0,57%)	3.652 (2,17%)	22.373 (0,65%)
disturbi vascolari periferici	61.459 (1,87%)	5.439 (3,23%)	66.898 (1,94%)
ipertensione non complicata	411.426 (12,52%)	8.723 (5,18%)	420.149 (12,17%)
paralisi	9.523 (0,29%)	770 (0,46%)	10.293 (0,30%)
altri disturbi neurologici	42.054 (1,28%)	4.264 (2,53%)	46.318 (1,34%)
malattia polmonare cronica	110.068 (3,35%)	7.966 (4,73%)	118.034 (3,42%)
diabete non complicato	209.327 (6,37%)	9.273 (5,51%)	218.600 (6,33%)
diabete complicato	24.613 (0,75%)	1.419 (0,84%)	26.032 (0,75%)
ipotiroidismo	22.605 (0,69%)	528 (0,31%)	23.133 (0,67%)
insufficienza renale	88.233 (2,69%)	9.357 (5,56%)	97.590 (2,83%)
malattia del fegato	31.213 (0,95%)	5.364 (3,19%)	36.577 (1,06%)
ulcera peptica esclusa emorragia	1.453 (0,04%)	64 (0,04%)	1.517 (0,04%)
AIDS/HIV	1.797 (0,05%)	253 (0,15%)	2.050 (0,06%)
linfoma	8.047 (0,25%)	1.257 (0,75%)	9.304 (0,27%)
cancro metastatico	501.174 (15,26%)	69.117 (41,05%)	570.291 (16,51%)
tumore solido senza metastasi	1.223.106 (37,22%)	106.925 (63,50%)	1.330.031 (38,52%)
artrite reumatoide/collagene vascolare	10.070 (0,31%)	531 (0,32%)	10.601 (0,31%)
coagulopatia	11.648 (0,35%)	2.304 (1,37%)	13.952 (0,40%)
obesità	41.534 (1,26%)	748 (0,44%)	42.282 (1,22%)
perdita di peso	27.992 (0,85%)	21.083 (12,52%)	49.075 (1,42%)
disturbi dei fluidi e degli elettroliti	20.235 (0,62%)	3.654 (2,17%)	23.889 (0,69%)
anemia da perdita di sangue	24.124 (0,73%)	1.164 (0,69%)	25.288 (0,73%)
anemia da carenza	11.886 (0,36%)	457 (0,27%)	12.343 (0,36%)
abuso di alcol	5.197 (0,16%)	797 (0,47%)	5.994 (0,17%)
abuso di droga	683 (0,02%)	53 (0,03%)	736 (0,02%)
psicosi	4.726 (0,14%)	302 (0,18%)	5.028 (0,15%)
depressione	12.681 (0,39%)	254 (0,15%)	12.935 (0,37%)
ipertensione complicata	83.185 (2,53%)	3.871 (2,30%)	87.056 (2,52%)

Tabella 6 Prevalenza delle comorbilità individuali dal metodo Elixhauser in base allo stato di dimissione del paziente (N=3.453.570)

Le tabelle 5 e 6 mostrano rispettivamente le cinque diagnosi principali che compaiono più frequentemente e la distribuzione delle varie patologie nei pazienti sopravvissuti e deceduti, evidenziando alcune relazioni importanti tra le condizioni cliniche e gli esiti clinici.

Ecco una descrizione delle comorbilità più associate alla mortalità ospedaliera:

- Insufficienza cardiaca congestizia: La percentuale di pazienti con insufficienza cardiaca congestizia è significativamente più alta tra i deceduti (9,26%) rispetto ai sopravvissuti (3,50%). Similmente anche le aritmie cardiache è una patologia più presente tra i pazienti deceduti (8,46%).
- Cancro metastatico: La percentuale di pazienti con cancro metastatico è molto più alta tra i deceduti (41,05%) rispetto ai sopravvissuti (15,26%).
- Tumore solido senza metastasi: Anche se i pazienti con tumore solido senza metastasi rappresentano una grande percentuale di entrambi i gruppi (37,22% nei sopravvissuti e 63,50% nei deceduti).
- Perdita di peso: La perdita di peso è associata a una percentuale molto più alta tra i deceduti (12,52%) rispetto ai sopravvissuti (0,85%).

5.3 Performance dei Modelli

5.3.1 Primo Approccio: Random Forest e XGBoost

L'obiettivo principale di questo studio è individuare, tra i soggetti analizzati, quelli a maggior rischio di mortalità ospedaliera. Per valutare il contributo dei dati strutturati al modello predittivo, nel primo approccio sono stati impiegati modelli di ML, come il Random Forest e XGBoost.

La selezione delle variabili più rilevanti è stata effettuata mediante la tecnica di regolarizzazione LASSO, che ha evidenziato come le variabili maggiormente influenti siano patologie come insufficienza cardiaca congestizia, cancro metastatico, perdita di peso, tumore solido senza metastasi, e altre variabili come durata ricovero, età, disciplina ospedaliera e l'indice Elixhauser.

I modelli sono stati validati suddividendo il dataset in training (70%), test (20%) e validation (10%), integrando una cross-validation a 3 fold per garantire una robusta valutazione della generalizzabilità delle soluzioni proposte.

Per ottimizzare ulteriormente la capacità predittiva, i parametri di tuning sono stati ricercati tramite Bayesian Optimization. In particolare, per il modello Random Forest sono stati individuati i seguenti parametri ottimali: 250 alberi (n_estimators), funzione di scelta delle

feature impostata su "log2" (max_features), profondità massima pari a 10 (max_depth), minimo numero di campioni richiesto per effettuare una divisione pari a 2 (min_samples_split) e minimo numero di campioni per foglia pari a 3 (min_samples_leaf). Per il modello XGBoost, invece, i parametri ottimali sono risultati essere: 50 alberi (n_estimators), profondità massima pari a 6 (max_depth), learning rate pari a 0.05, subsample del 50% e colsample_bytree pari a 1.0.

	Random Forest			XGBoost				
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	N	
Survived (0)	0.96	1.00	0.98	0.96	1.00	0.98	657,836	
Deceased (1)	0,75	0,28	0,4	0,74	0,27	0,4	33,679	
Accuracy	-	-	0,96	-	-	0,96	690,715	
Macro Avg	0,86	0,64	0,69	0,85	0,63	0,69	690,715	
Weighted Avg	0,95	0,96	0,95	0,95	0,96	0,95	690,715	

Tabella 7 Classification report dei modelli Random Forest e XGBoost

I risultati ottenuti dai modelli Random Forest e XGBoost per la classificazione dei pazienti a rischio di mortalità, riportati nella tabella 7, mostrano performance interessanti ma con margini di miglioramento. Per la classe “Survived” (0), entrambi i modelli hanno raggiunto una precisione del 96%, un recall del 100% e un F1-score del 98%. Questi valori confermano un’ottima capacità di identificazione dei soggetti non a rischio, garantendo una elevata affidabilità nelle previsioni per questa classe. Tuttavia, per la classe “Deceased” (1), le prestazioni risultano inferiori, con una precisione intorno al 75% circa, un recall del 28% circa e un F1-score del 40%.

Questa discrepanza suggerisce una maggiore difficoltà dei due modelli nel riconoscere tempestivamente i pazienti a rischio, probabilmente a causa di una scarsità d’informazione dei dati strutturati o di uno squilibrio della variabile target intrinseco nei dati di training.

L’accuratezza complessiva dei modelli si attesta al 96%, con una media macro degli F1-score pari al 69% e una media ponderata del 95%.

Questi indicatori, se da un lato testimoniano una buona performance generale, dall’altro sottolineano l’esigenza di incrementare la sensibilità del modello nei confronti dei casi più critici.

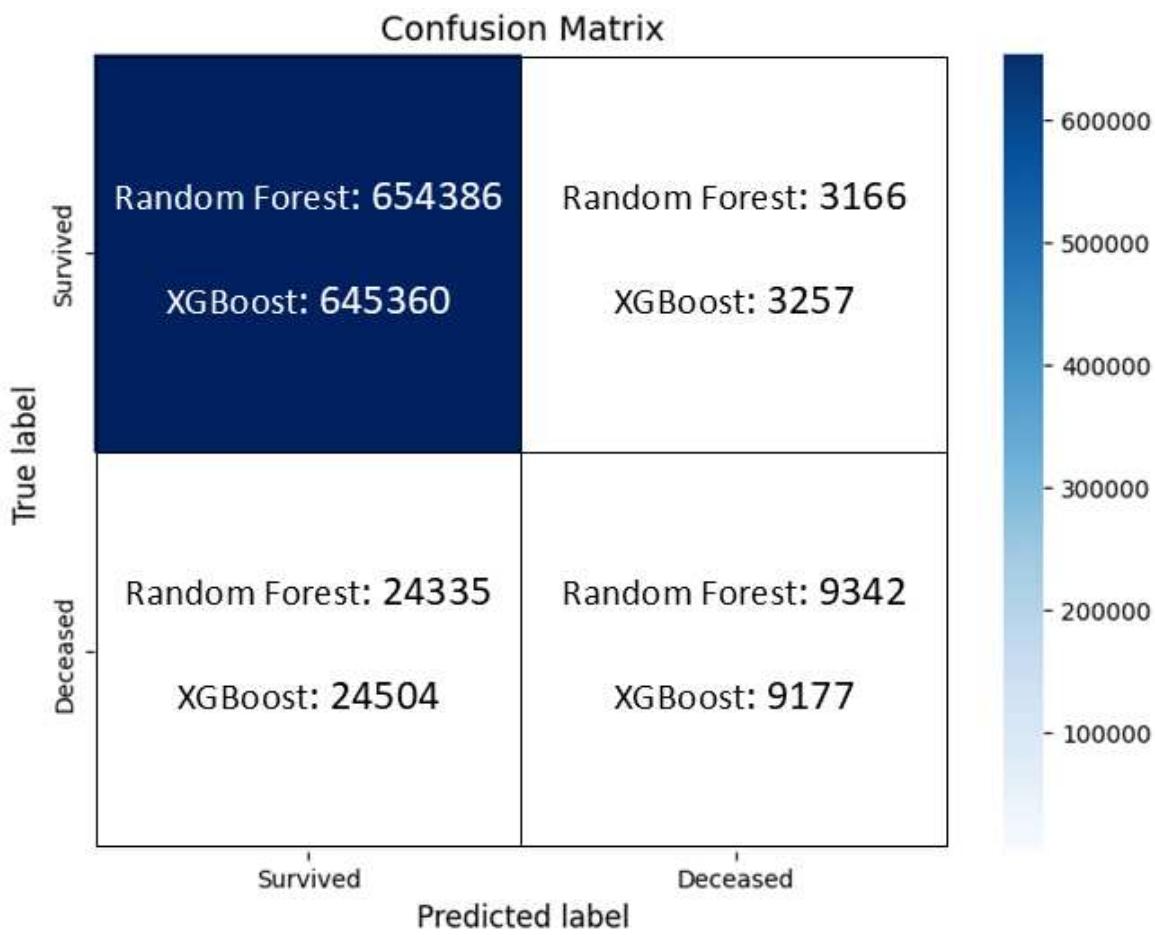


Grafico 2 confusion matrix dei modelli Random Forest e XGBoost

L'analisi della matrice di confusione, illustrata nel Grafico 2, conferma le osservazioni emerse dal classification report. Il dato particolarmente preoccupante è il numero elevato di falsi negativi di entrambi i modelli (24,335 *Random Forest* e 24,504 *XGBoost*), che spiega il basso recall per la classe “Deceased”. Questa situazione evidenzia una problematica centrale dei due modelli: sebbene l'alta accuratezza complessiva (96%) e l'elevata capacità di identificare i sopravvissuti possano sembrare rassicuranti, la mancata identificazione dei soggetti a rischio rappresenta un limite significativo, soprattutto in un contesto clinico dove ogni falso negativo può avere ripercussioni critiche.

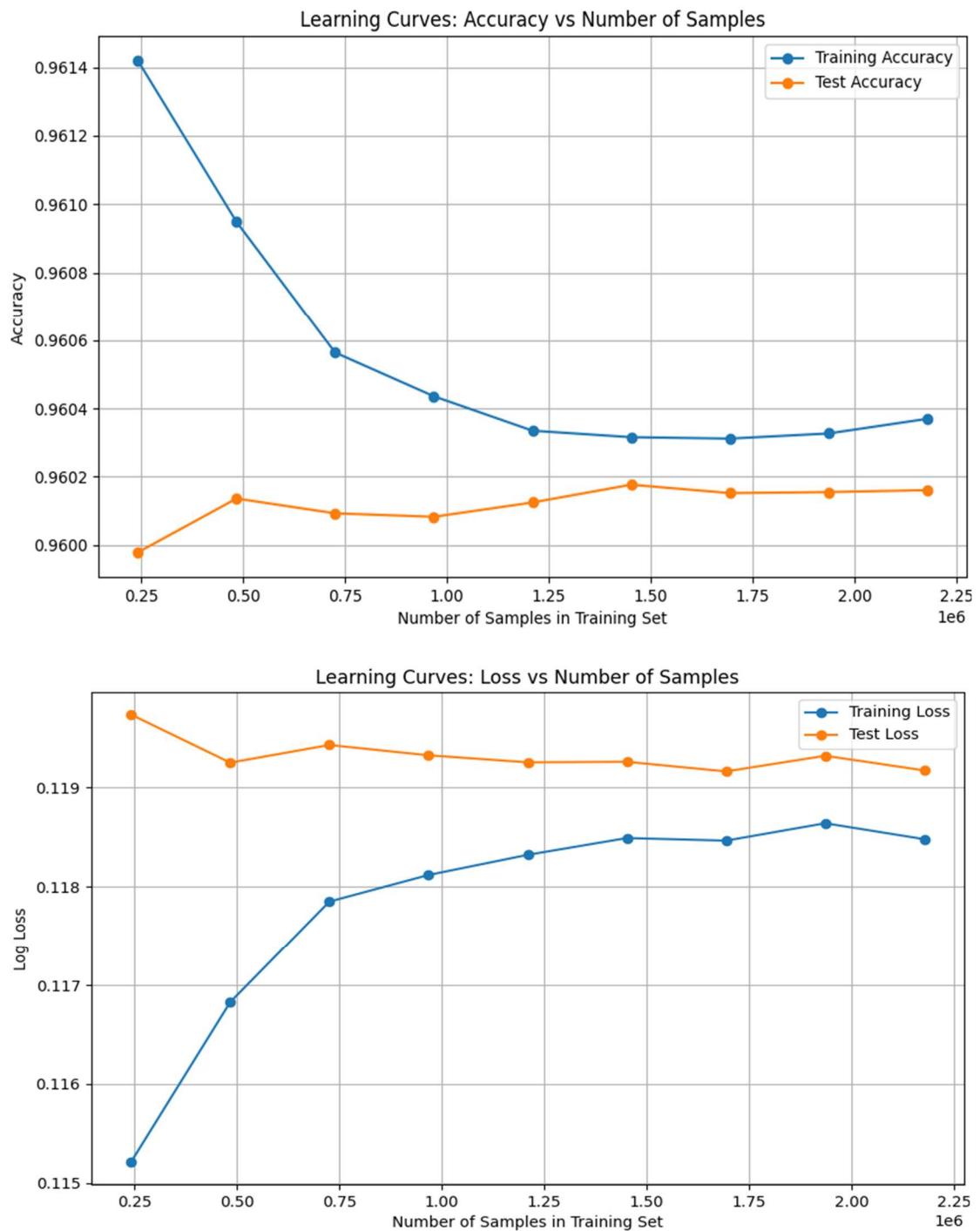


Grafico 3,4 Learning Curves Accuracy e Loss del modello Random Forest

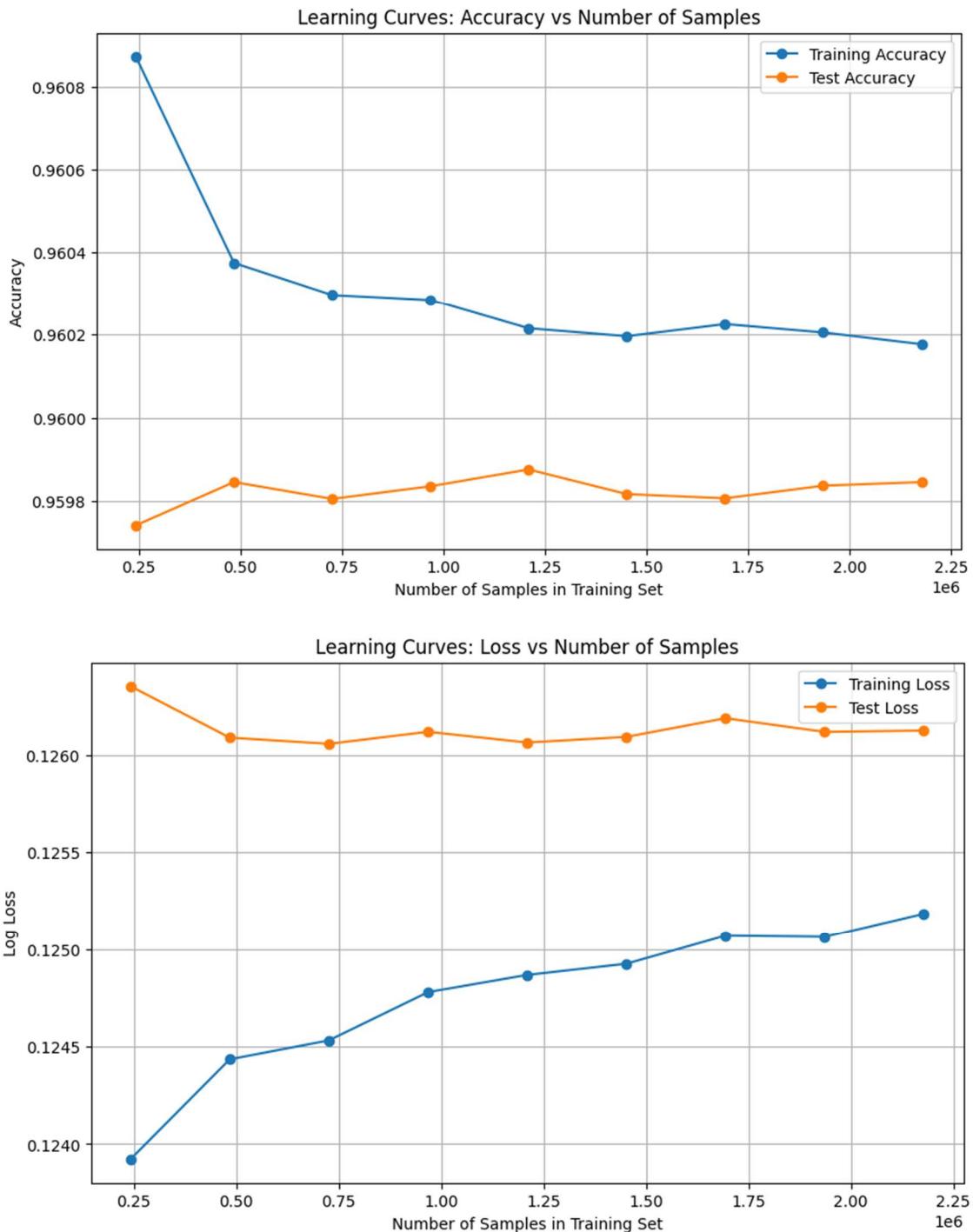


Grafico 5,6 Learning Curves Accuracy e Loss del modello XGBoost

Le Learning Curve dei modelli Random Forest e XGBoost (Grafici 3,4,5 e 6) mostrano che con l'aumentare del numero di campioni, sia l'accuracy che la loss tendono a stabilizzarsi. I modelli dimostrano nel complesso una discreta capacità di generalizzazione, che risulta lievemente più alta per il modello Random Forest. Tuttavia, per entrambi i modelli, nelle prime fasi del training, si osserva un leggero divario tra i valori di loss e accuracy del training set rispetto a quelli del test set. Queste differenze suggeriscono la presenza di un lieve overfitting.

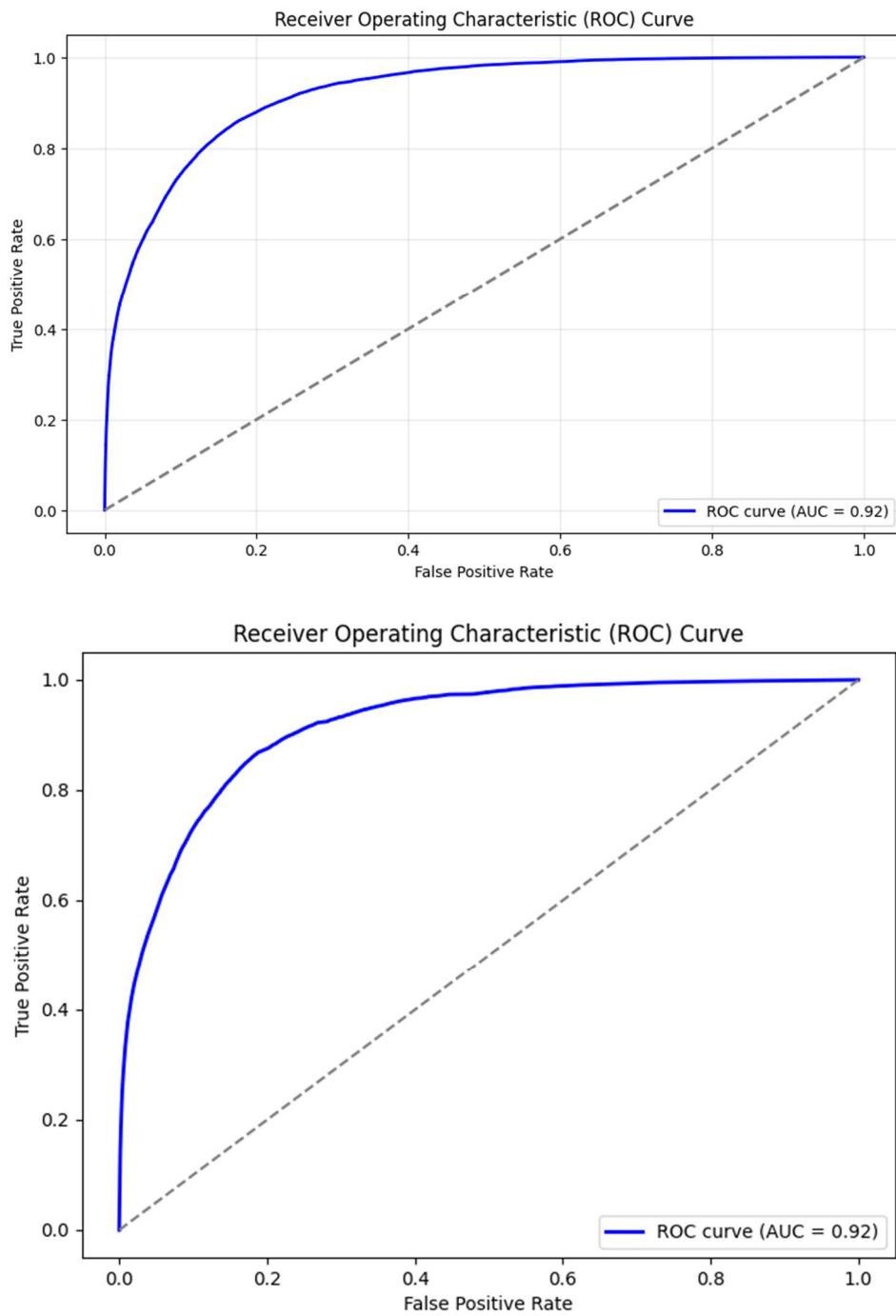


Grafico 7,8 curve ROC dei modelli Random Forest e XGBoost

I valori di AUC ROC dei modelli Random Forest e XGBoost pari a 0.92 (Grafici 7 e 8) suggeriscono una buona capacità discriminante complessiva del modello, ovvero una forte separazione tra le classi “Survived” e “Deceased”. Tuttavia, in presenza di dati fortemente sbilanciati, l’AUC ROC tende a essere influenzato dalla performance sulla classe maggioritaria. In altre parole, i modelli potrebbero apparire performanti secondo la metrica ROC, ma soprattutto per il fine clinico di identificare pazienti a rischio, è necessario approfondire ulteriormente la valutazione con metriche più sensibili allo sbilanciamento.

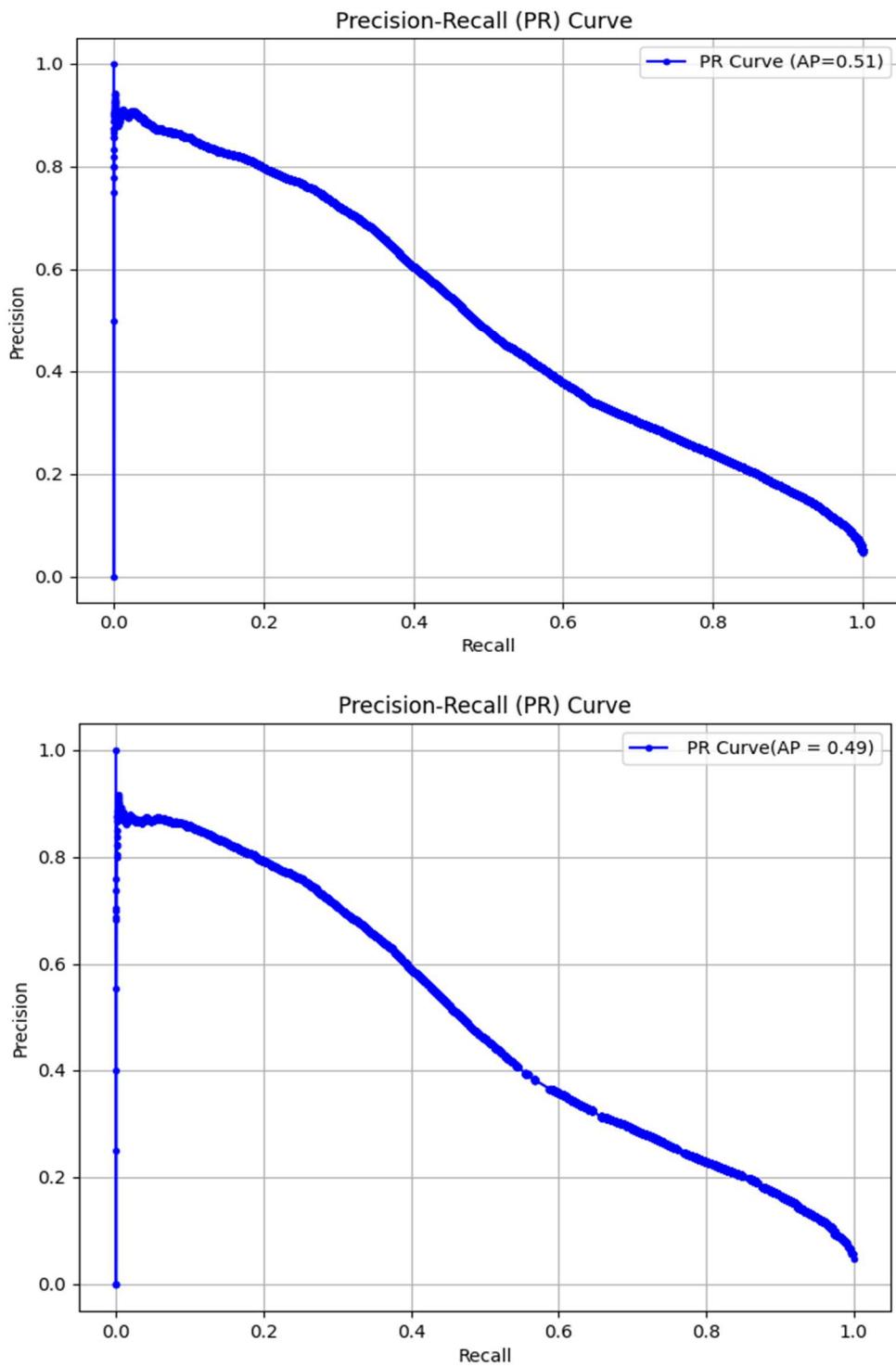


Grafico 9,10 curve Precision-Recall dei modelli Random Forest e XGBoost

L'analisi della curva Precision-Recall risulta particolarmente utile in contesti con dataset sbilanciati. Generalmente, un'Average Precision (AP) superiore a 0.5 è considerato accettabile, mentre valori oltre 0.7 indicano prestazioni eccellenti. In questo caso, un'AP prossimo a 0.5 (Grafici 9 e 10) evidenzia che per la classe "Deceased" i modelli faticano a identificare correttamente questa categoria.

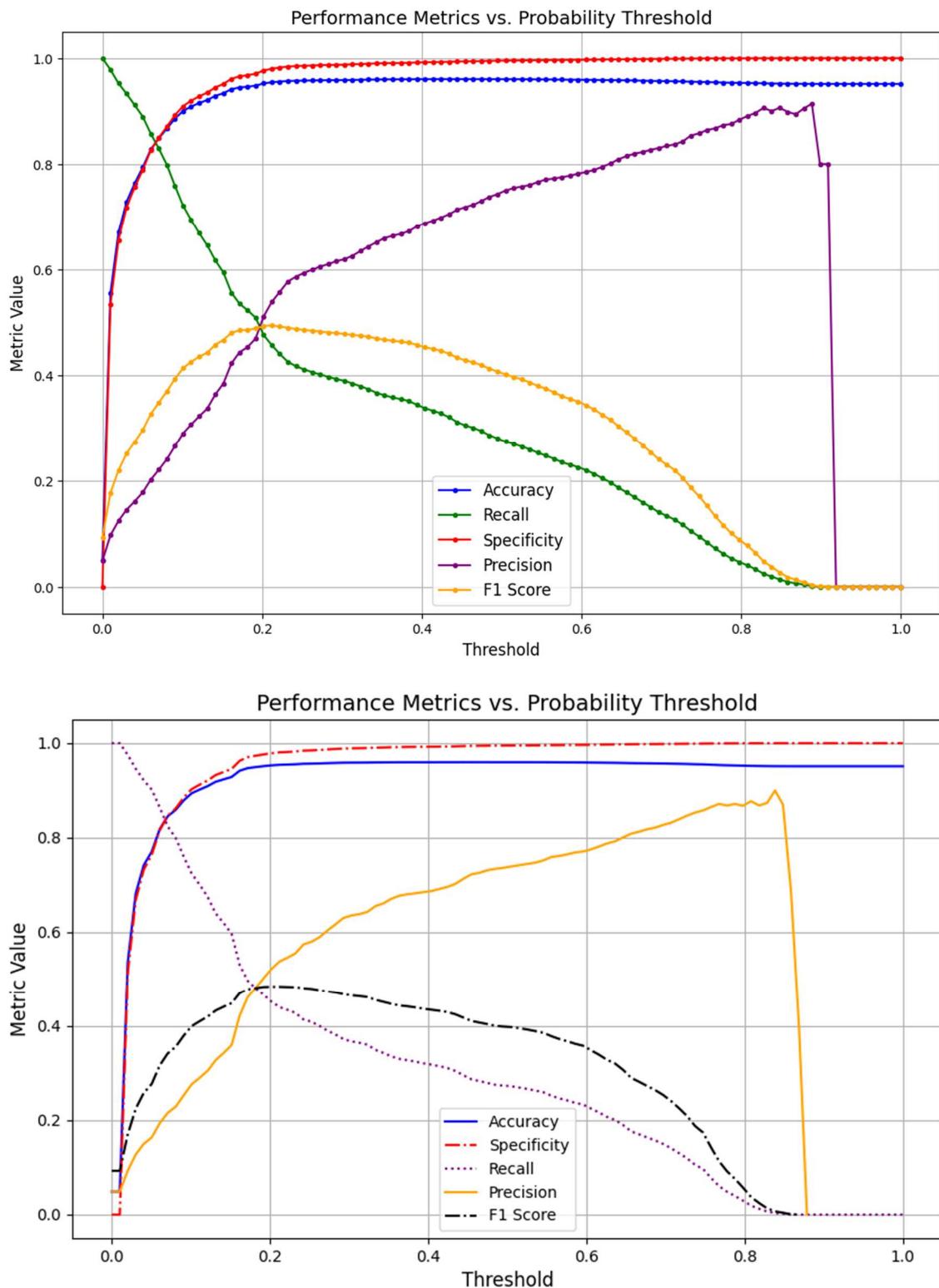


Grafico 11,12 Grafico Threshold vs Metrics del modello Random Forest e XGBoost

I grafici 11 e 12 mostrano l'andamento di accuratezza, precisione, recall e F1-score al variare della soglia di decisione. Queste rappresentazioni permettono di identificare il threshold ideale per classificare correttamente i pazienti a rischio, garantendo un bilanciamento ottimale delle performance. In particolare, i dati indicano che, se il costo di un falso negativo è elevato,

la soglia standard di 0.5 potrebbe non essere la scelta più efficace. L'analisi del threshold di classificazione relativa ai modelli *Random Forest* e *XGBoost* mostra che un possibile abbassamento della soglia (ad esempio 0,18-0,2) potrebbe corrispondere a un valore di Recall per la classe “Deceased” intorno al 49%, migliorando così la capacità dei modelli di identificare i soggetti a rischio. Tuttavia, questa operazione comporta un inevitabile decremento della precisione per la classe “Deceased” e un aumento importante dei falsi positivi.

5.3.2 Secondo Approccio: Random Forest (TF-IDF) e XGBoost (BoW)

Nel secondo approccio, sono state impiegate tecniche NLP per la classificazione dei pazienti, con particolare attenzione alla rappresentazione testuale delle diagnosi tramite il modello Bag-of-Words (BoW) e la tecnica TF-IDF.

Per la rappresentazione tramite Bag of Words (BoW) sono stati selezionati i 500 termini più frequenti, ottenuti a seguito di un accurato preprocessing che ha previsto la conversione in minuscolo, la lemmatizzazione, la rimozione di stopwords e caratteri speciali, nonché l'inclusione di bigrammi e n-grammi.

Dopodiché il modello XGBoost con rappresentazione BoW, è stato addestrato con una suddivisione dei dati in 70% per il training, 20% per il test e 10% per la validation, e validato mediante una cross-validation a 3 fold.

I parametri ottimizzati tramite Bayesian Optimization sono risultati: colsample_bytree pari a 0.751, gamma pari a 0.0675, learning_rate pari a 0.252, max_depth pari a 8, min_child_weight pari a 3, n_estimators pari a 151 e subsample pari a 0.792. Parallelamente, per la rappresentazione tramite TF-IDF è stato impiegato un modello Random Forest, anch'esso ottimizzato tramite Bayesian Optimization, che ha restituito i seguenti parametri: n_estimators pari a 100, max_depth pari a 36, min_samples_split pari a 20 e min_samples_leaf pari a 3.

	Random Forest (TF-IDF)			XGBoost (BoW)			
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	N
Survived (0)	0.96	1.00	0.98	0.97	0.99	0.98	657,836
Deceased (1)	0.78	0.29	0.42	0.73	0.32	0.44	33,679
Accuracy	-	-	0,96	-	-	0,96	690,715
Macro Avg	0.87	0.64	0.70	0.85	0.66	0.71	690,715
Weighted Avg	0.96	0.96	0.95	0.95	0.96	0.95	690,715

Tabella 8 Classification report del modello XGBoost con rappresentazione BoW e del modello Random Forest con rappresentazione TF-IDF

I risultati ottenuti, dal classification report (Tabella 8), presentano performance interessanti, ma con margini di miglioramento. Similmente a quanto visto nel primo approccio per la classe “Survived” (0) entrambi i modelli che utilizzano tecniche classiche di NLP, raggiungono una precisione intorno al 97%, un recall del 99-100% e un F1-score del 98%, e confermano un’ottima capacità di identificazione dei soggetti non a rischio. Per la classe “Deceased” (1) le prestazioni risultano leggermente migliori rispetto ai modelli che utilizzavano dati strutturati, con una precisione del 73-78%, un recall del 29-32% e un F1-score del 42-44%.

Quanto sopra riportato, evidenzia come una classificazione basata solo sul testo può portare ad un miglioramento dei risultati proprio in quanto si “catturano” informazioni che di solito sono difficili da implementare in dati tabulari. Tuttavia rimane ancora non ideale la sensibilità dei due modelli nel riconoscere tempestivamente i pazienti a rischio di mortalità. Molto probabilmente questo approccio, che utilizza le rappresentazioni BoW e TF-IDF, risulta ancora poco capace di catturare in modo adeguato la complessità semantica delle parole che formano le diagnosi, che sono fondamentali per discriminare efficacemente i soggetti a rischio di mortalità.

L’accuratezza complessiva dei modelli si attesta al 96%, con una media macro degli F1-score intorno al 70% e una media ponderata del 95%.



Grafico 13 confusion matrix dei modelli XGBoost (BoW) e Random Forest (TF-IDF)

Come si osserva nel grafico 13, il dato particolarmente critico è il numero elevato di falsi negativi che caratterizza entrambi i modelli (23933 Random Forest e 22986 XGBoost). La mancata identificazione dei soggetti a rischio di mortalità rappresenta un limite importante, soprattutto in ambito clinico.

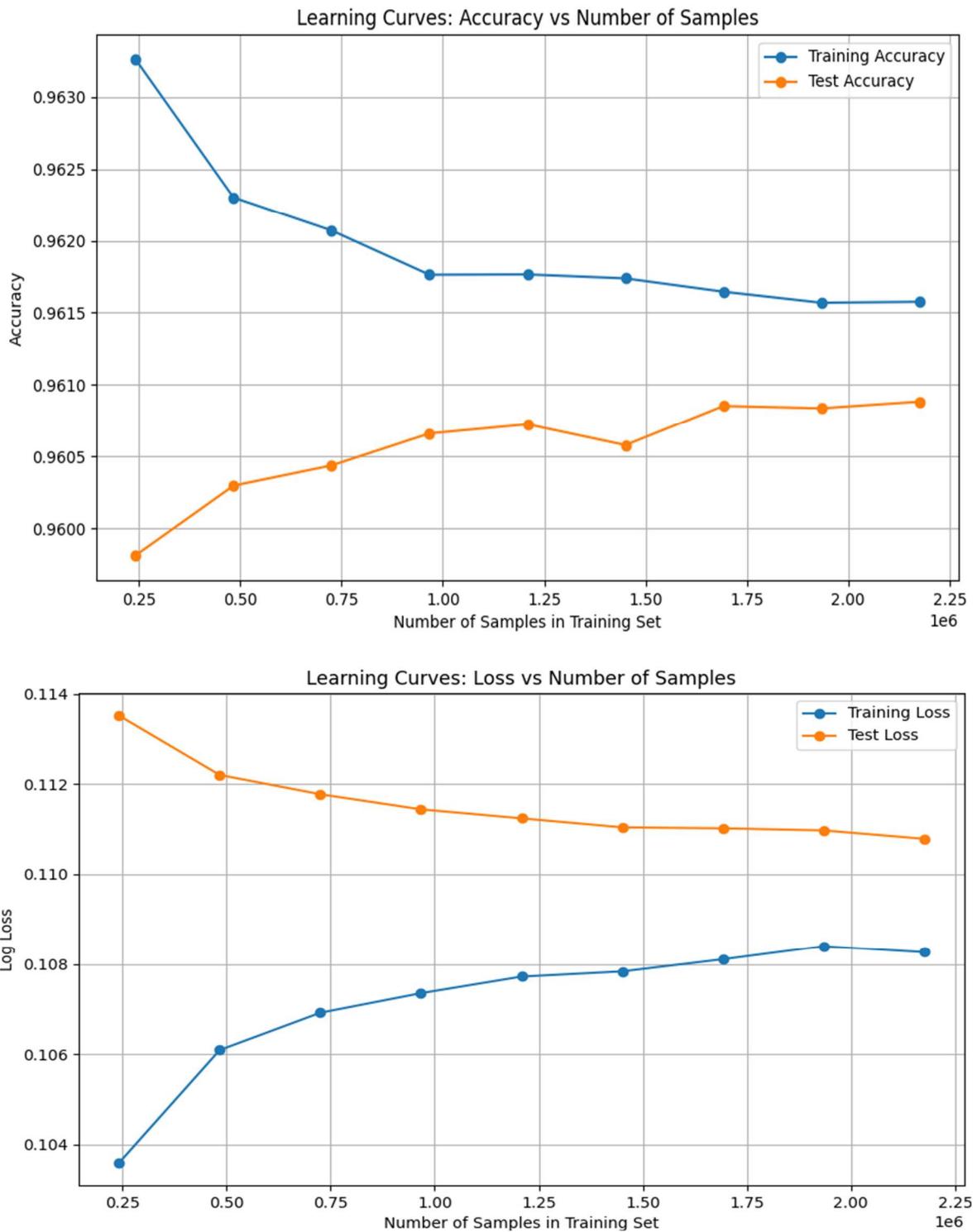


Grafico 14,15 Learning Curves Accuracy e Loss del modello XGBoost (BoW)

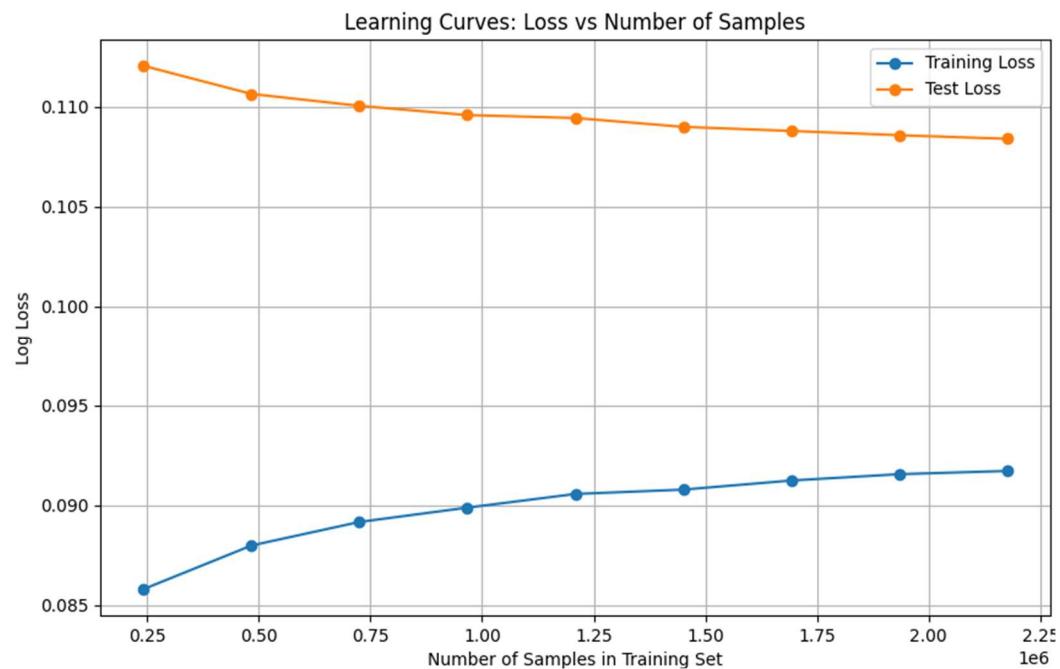
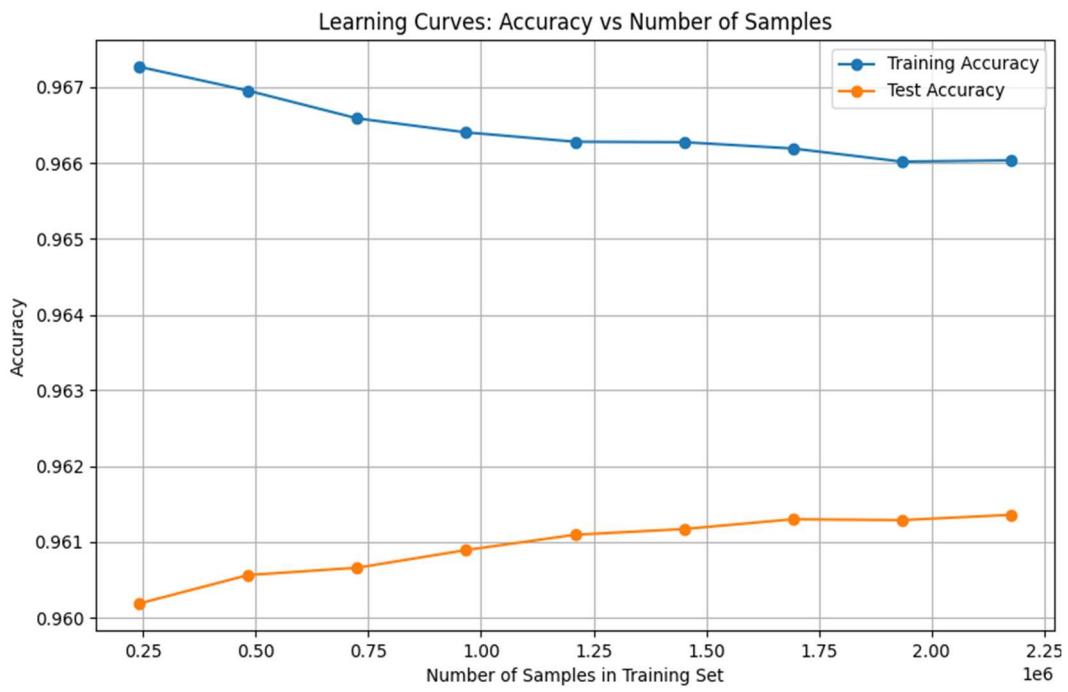


Grafico 16,17 Learning Curves Accuracy e Loss del modello Random Forest (TF-IDF)

Le Learning Curve dei modelli Random Forest e XGBoost (Grafici 14,15,16 e 17) mostrano che con l'aumentare del numero di campioni, sia l'accuracy che la loss tendono a consolidarsi. I due modelli dimostrano nel complesso una discreta capacità di generalizzazione, che risulta lievemente più alta per il modello XGBoost. Tuttavia, per entrambi i modelli, nelle prime fasi del training, si osserva una certa distanza tra i valori di loss e accuracy del training set rispetto a quelli del test set. Queste differenze suggeriscono la presenza di un lieve overfitting.

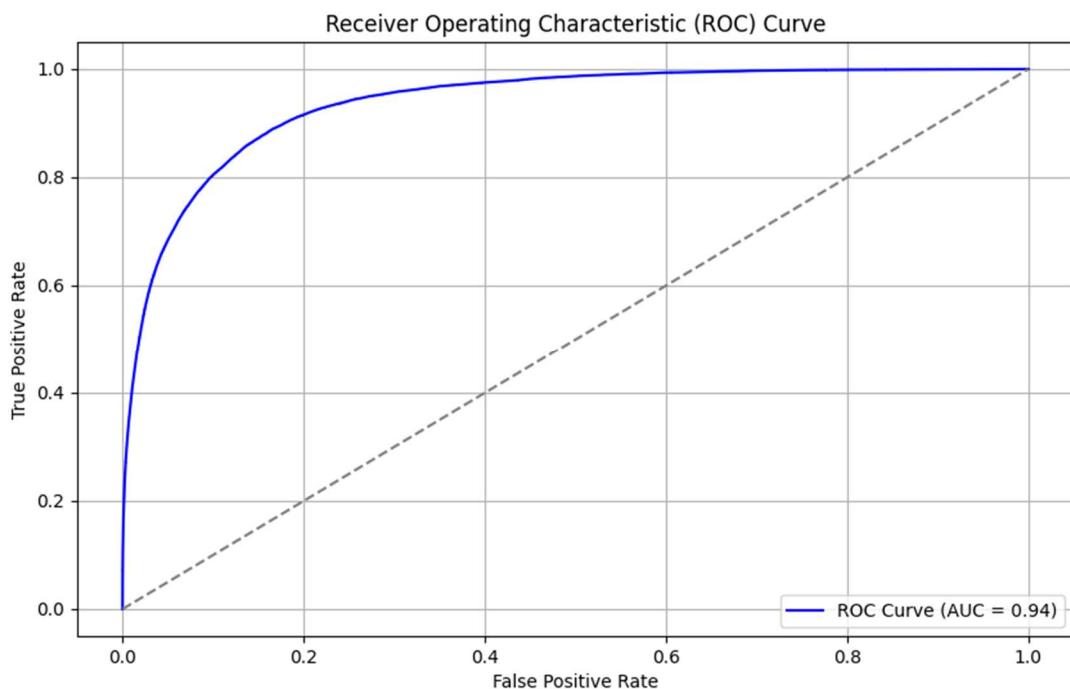
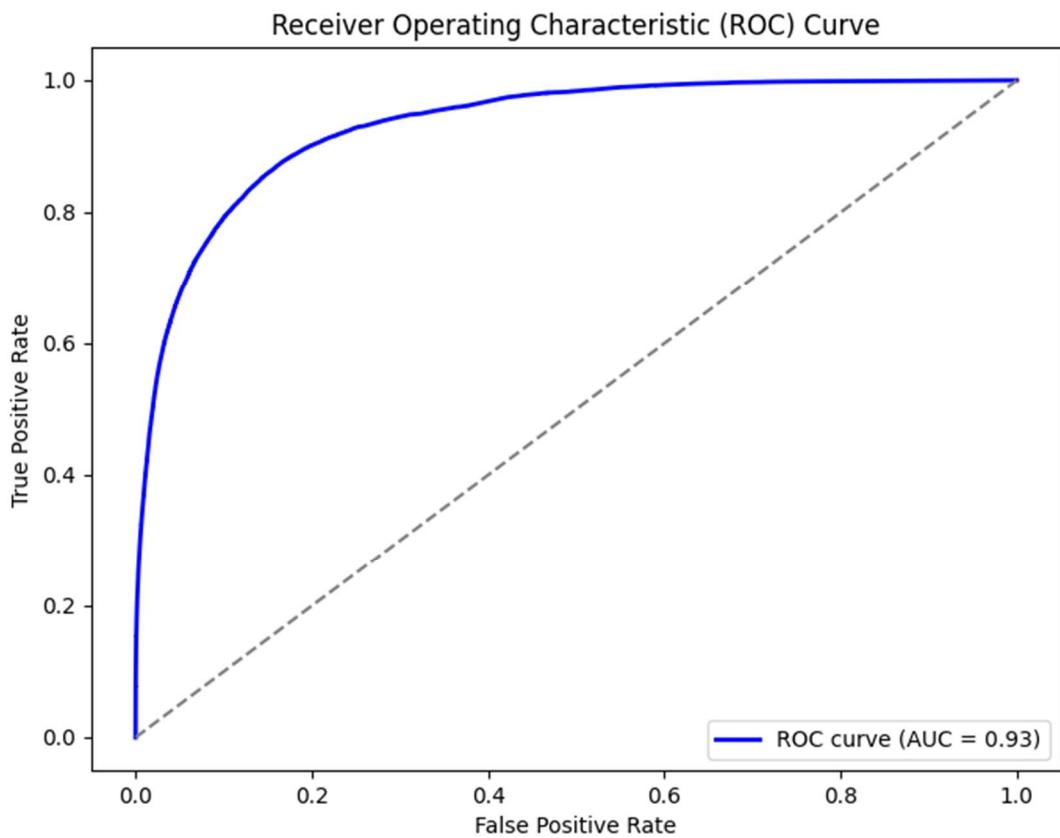


Grafico 18,19 curve ROC dei modelli XGBoost (BoW) e Random Forest (TF-IDF)

I valori di AUC ROC dei modelli XGBoost (BoW) e Random Forest (TF-IDF) risultano pari a 0.93-0.94 cioè mostrano una buona capacità di discriminare tra le classi “Survived” e “Deceased”. In altre parole i modelli distinguono bene tra le due classi ma non riescono a compensare il basso recall osservato per la classe “Deceased” (*Grafici 18 e 19*).

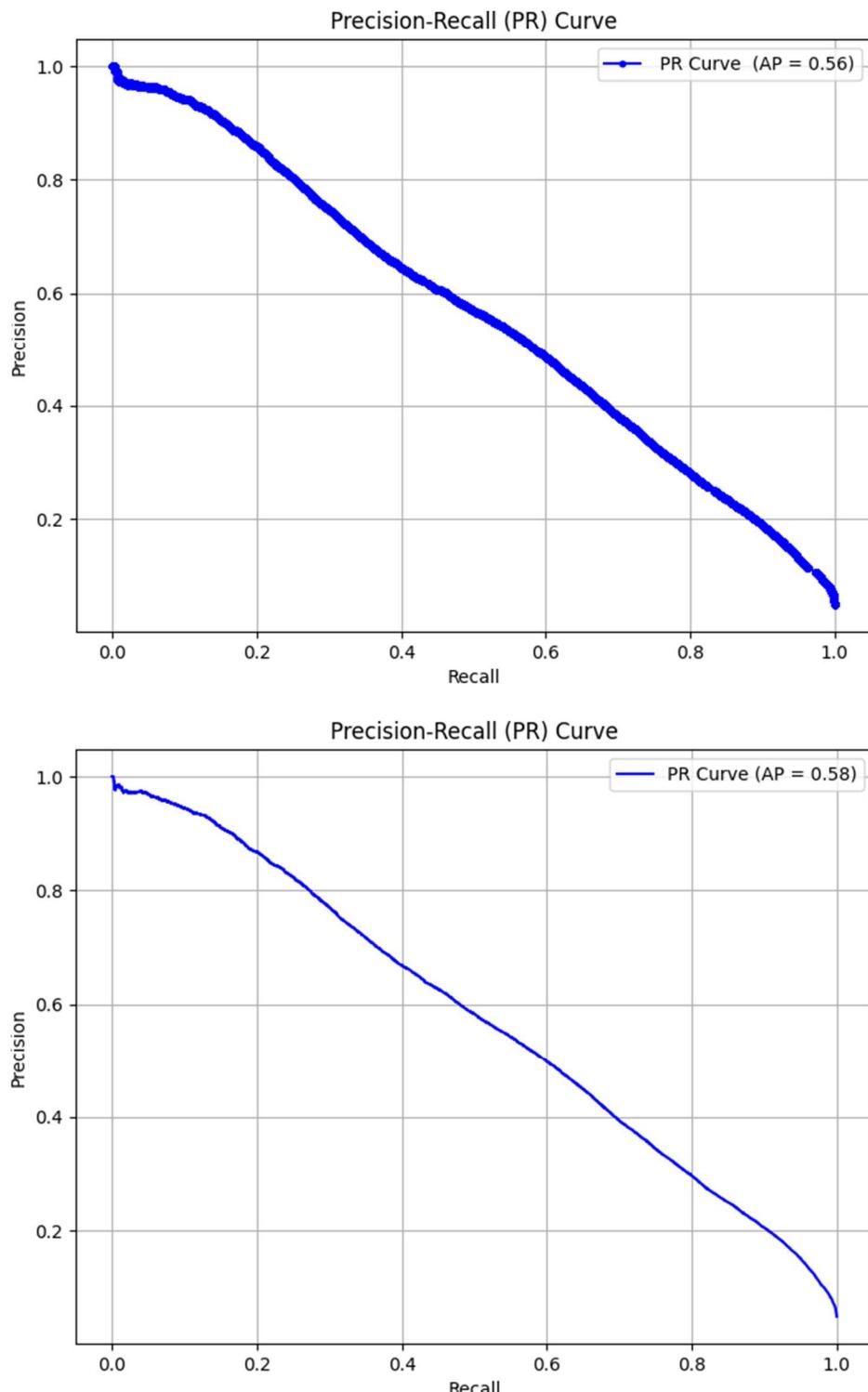


Grafico 20,21 curve Precision-Recall dei modelli XGBoost (BoW) e Random Forest (TF-IDF)

In questo approccio i valori di AP raggiungono un valore di 0.58 Random Forest (TF-IDF) e di 0.56 XGBoost (BoW). Rispetto ai modelli dei primi approcci che utilizzano solo dati strutturati, aumenta l'Average Precision dei modelli basati sul testo delle diagnosi. I valori rimangono in ogni caso ancora non ottimali e mostrano una scarsa abilità dei modelli di identificare correttamente la classe di d'interesse (Grafici 20 e 21).

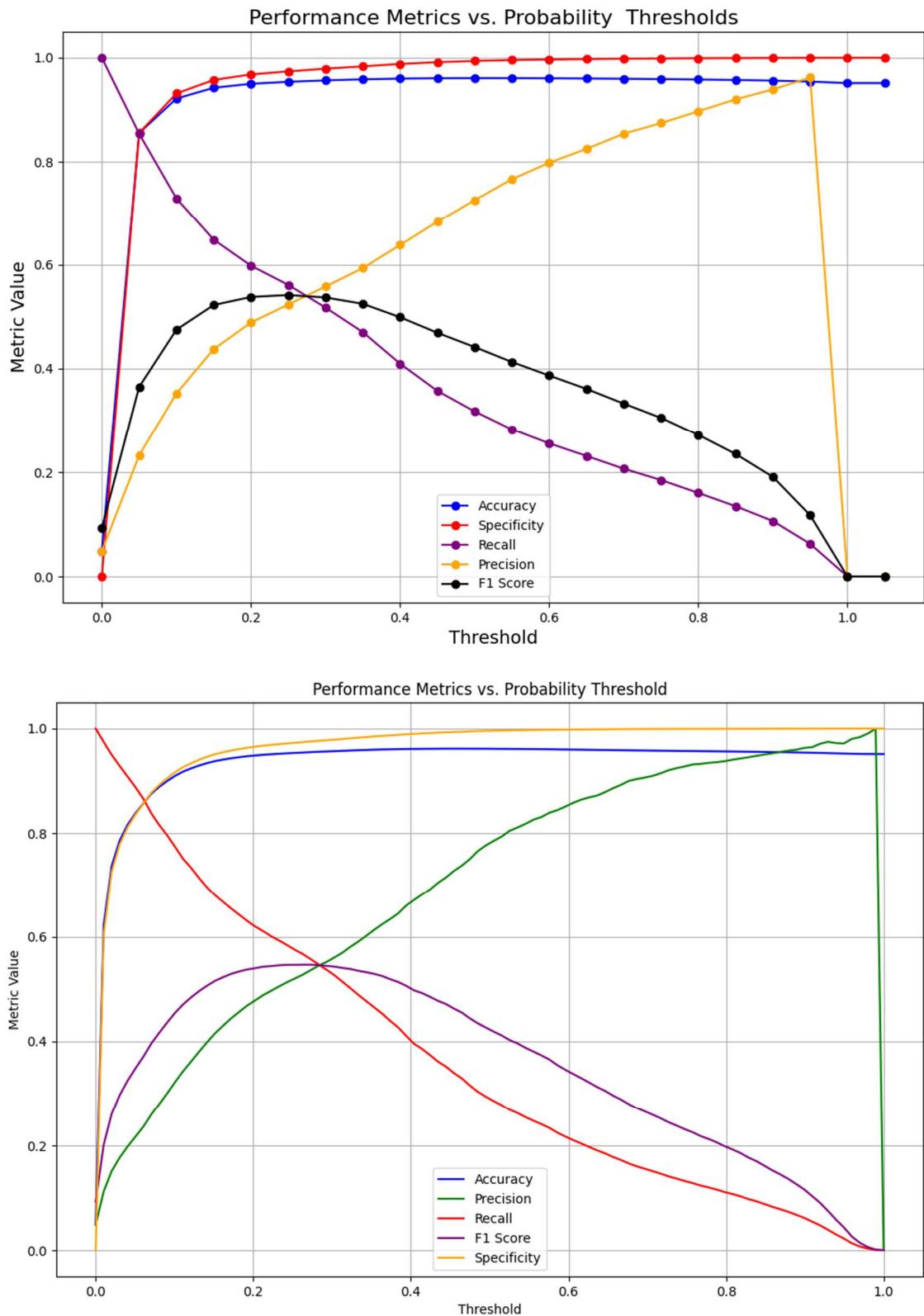


Grafico 22,23 Grafico Threshold vs Metrics dei modelli XGBoost (BoW) e Random Forest (TF-IDF)

L’analisi del threshold di classificazione relativa ai modelli Random Forest (TF-IDF) e *XGBoost* (BoW) (*Grafici 22 e 23*) mostra che un possibile abbassamento della soglia (ad esempio 0.26-0.27) potrebbe corrispondere a un valore di Recall per la classe “Deceased” intorno al 55-56%, migliorando così la capacità dei modelli di identificare i soggetti a rischio. Tuttavia, questa operazione comporta un inevitabile decremento della precisione per la classe “Deceased” e un aumento importante dei falsi positivi.

5.3.3 Terzo Approccio: Bio_Discharge_Summary_BERT

Nel terzo approccio, finalizzato a migliorare la rappresentazione dei valori testuali delle diagnosi, è stato adottato uno strumento avanzato di NLP basato su modelli pre-addestrati, nello specifico il modello Bio_Discharge_Summary_BERT.

Tale modello è stato sottoposto a un accurato processo di preprocessing testuale, che ha compreso la conversione del testo in minuscolo, l’eliminazione di caratteri speciali e la rimozione delle stopwords, al fine di ottenere una rappresentazione più omogenea e pulita del contenuto.

Il dataset è stato, come consuetudine, suddiviso in tre insiemi distinti: training, test e validation. L’addestramento è stato condotto utilizzando l’ottimizzatore AdamW, con un learning rate pari a 1e-5 e un weight decay di 0.01, per 1 epoca, impiegando un batch size di 32 e impostando una lunghezza massima dei token pari a 90.

	Precision	Recall	F1-Score	N
Survived (0)	0.97	0.99	0.98	657,836
Deceased (1)	0.72	0.41	0.53	33,679
Accuracy	-	-	0,96	690,715
Macro Avg	0.84	0.70	0.75	690,715
Weighted Avg	0.96	0.96	0.96	690,715

Tabella 9 Classification report modello Bio_Discharge_Summary_BERT

I risultati riportati nel classification report (Tabella 9), da parte del modello pre-addestrato, Bio_Discharge_Summary_BERT, mostrano in generale un progressivo incremento delle performance. In dettaglio per la classe “Survived” (0) il modello raggiunge una precisione del 97%, un recall del 99% e un F1-score del 98%. Questi valori confermano ancora un’ottima capacità di identificazione dei soggetti non a rischio. Invece, per la classe “Deceased” (1), le

prestazioni risultano migliorate nel complesso, anche se non ancora in modo ottimale, con una precisione del 72%. Il modello preaddestrato Bio_Discharge_Summary_BERT per la classe “Deceased” ottiene F1-score pari 0.53 e un recall di 0.41, valori più alti rispetto a quelli ottenuti dai modelli che usano le rappresentazioni TF-IDF e BoW (F1-score pari 0.42-0.44 e un recall di 0.29-0.32).

L’accuratezza complessiva del modello si attesta al 96%, con una media macro degli F1-score pari al 75% e una media ponderata del 96%.

Questa discrepanza suggerisce una sempre difficoltà nel riconoscere tempestivamente i pazienti a rischio, ma ridotta rispetto agli approcci precedenti, probabilmente a causa di una rappresentazione testuale più accurata data dai meccanismi che caratterizzano un modello preaddestrato, tenendo conto che stiamo lavorando con uno squilibrio intrinseco nei dati di training data dalle classi del target.

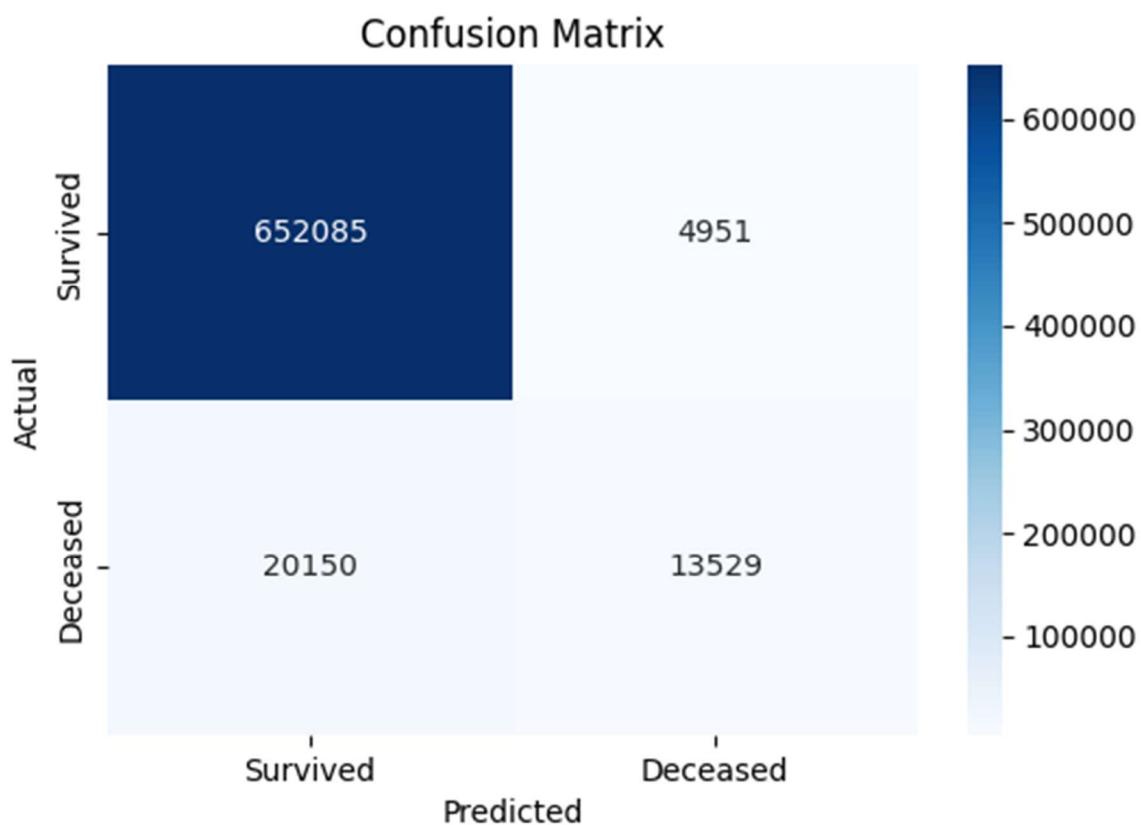


Grafico 24 matrice confusione Bio_Discharge_Summary_BERT

Come si osserva dal *Grafico 24* il modello Bio_Discharge_Summary_BERT mostra che il dato particolarmente preoccupante dei falsi negativi, che nei precedenti approcci era elevato, adesso si è ridotto ulteriormente a 20150 casi a favore dei veri positivi che salgono a 13529.

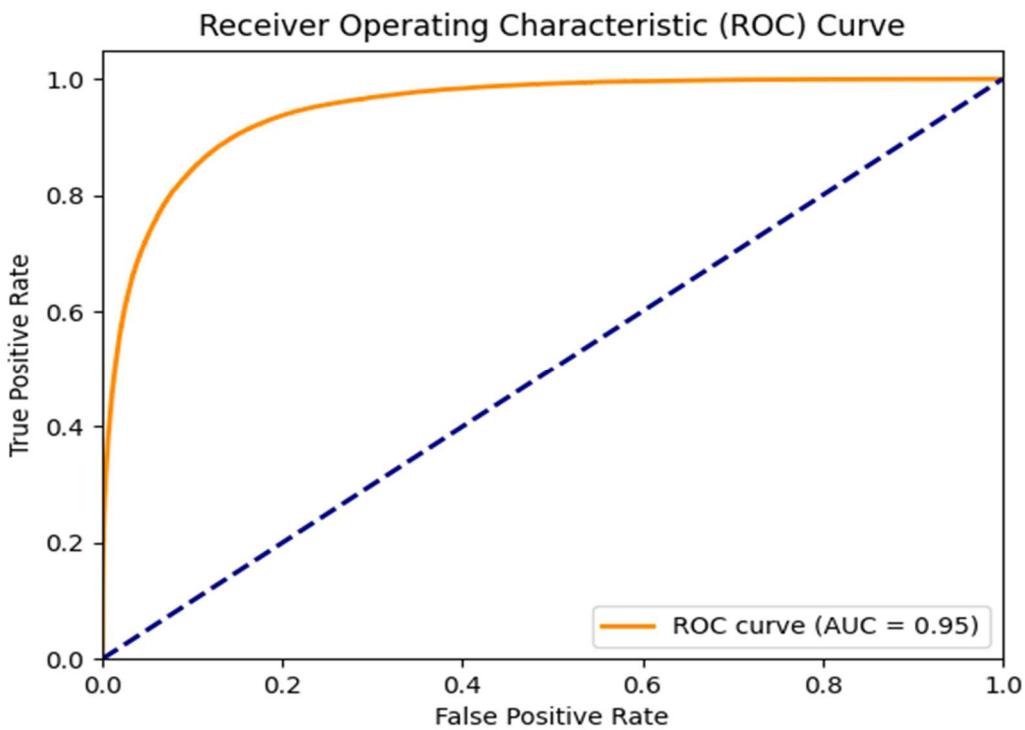


Grafico 25 curva ROC del modello *Bio_Discharge_Summary_BERT*

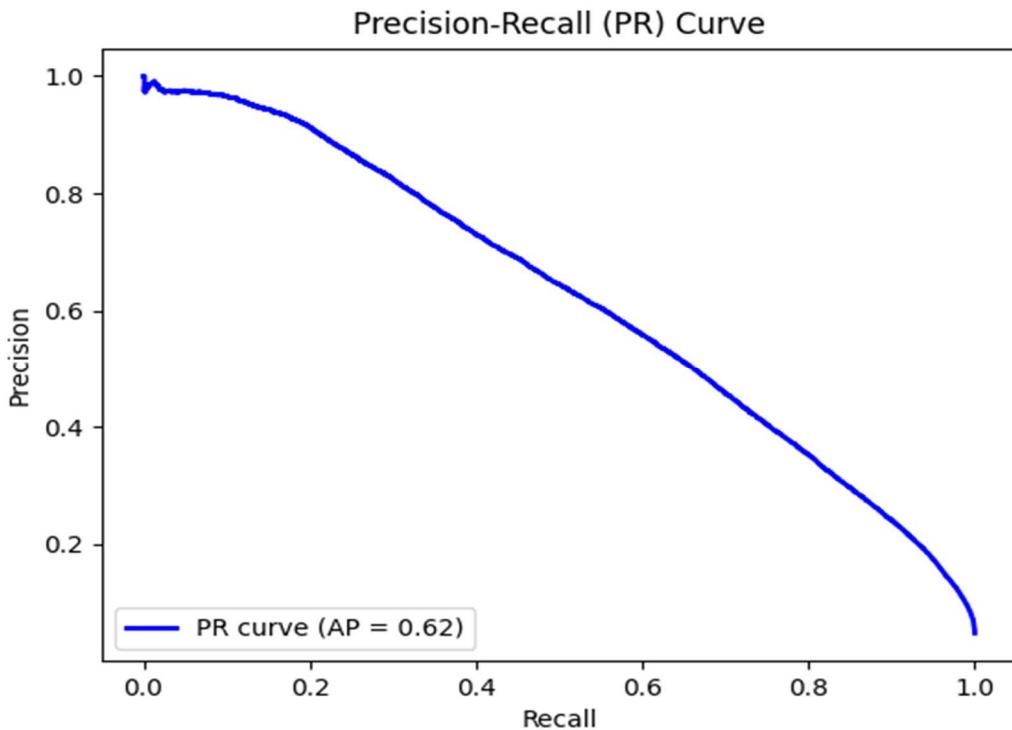


Grafico 26 curva Precision-Recall del modello *Bio_Discharge_Summary_BERT*

La curva ROC del modello Bio_Discharge_Summary_BERT (*Grafico 25*) evidenzia un discreto aumento nella capacità discriminante. Infatti l'area sotto la curva (AUC) mostra un

valore pari a 0.95 indicativo di una buona performance della capacità del modello di differenziare tra pazienti sopravvissuti e deceduti.

Rispetto ai modelli basati su BoW o TF-IDF, l'AP del modello pre-addestrato è aumentata da 0.58 a 0.62 (*Grafico 26*). Sebbene la performance complessiva rimanga moderata, questo incremento evidenzia un compromesso tra precisione e recall per la classe "Deceased" più bilanciato, migliorando la capacità del modello di identificare correttamente i pazienti a rischio.

5.3.4 Quarto Approccio: Modello Ibrido

Nel quarto approccio si è valutata la classificazione dei soggetti attraverso una combinazione ibrida che integra tecniche NLP e dati strutturati, al fine di sfruttare al meglio sia le informazioni testuali contenute nelle diagnosi sia i dati clinici, anagrafici, ecc.

In particolare, il modello pre-addestrato Bio_Discharge_Summary_BERT ha estratto rappresentazioni semantiche a partire dalla diagnosi, tali rappresentazioni vengono successivamente trattate da un LSTM bidirezionale, in grado di catturare le informazioni da entrambi le direzioni del testo, e da un meccanismo di attention che mette in evidenza le informazioni più rilevanti. Dopo l'elaborazione del testo il modello applica tecniche di Pooling globale per ridurre la dimensionalità e catturare le caratteristiche più significative. Parallelamente, a queste rappresentazioni testuali è stato concatenato un insieme di variabili strutturate, selezionato mediante sperimentazioni che hanno evidenziato che il miglior gruppo di variabili fossero le seguenti: durata ricovero, sesso, età, regione, disciplina ospedaliera, l'indice Elixhauser e le 31 patologie che lo compongono.

Queste variabili strutturate vengono elaborate mediante layer densi con funzione di attivazione ReLU, seguiti da una Batch Normalization ed una fase di Dropout, al fine di migliorare la generalizzazione del modello e prevenire l'overfitting. Le rappresentazioni derivate dalla componente testuale e dalle variabili strutturate vengono quindi concatenate e passate attraverso un ulteriore layer denso per la classificazione finale, il cui output è una probabilità di appartenenza alla classe di interesse, ottenuta tramite una funzione di attivazione sigmoide.

I testi sono stati accuratamente pre-processati con lowercase, rimozione di stopwords e l'eliminazione di caratteri speciali e poi tokenizzati e normalizzati con una lunghezza massima di 90 token.

La suddivisione del dataset in training (70%), test (20%) e validation (10%) e l’addestramento a 5 epoche ha permesso di validare in modo robusto le performance del modello.

I parametri ottimizzati tramite Bayesian Optimization sono risultati: batch size di 32, l’ottimizzatore Adam con un learning rate pari a 0.000272, un dropout rate di 0.223, 128 unità LSTM, 64 e 32 unità dense.

	Precision	Recall	F1-Score	N
Survived (0)	0.97	0.99	0.98	657,836
Deceased (1)	0.79	0.47	0.59	33,679
Accuracy	-	-	0,97	690,715
Macro Avg	0.88	0.73	0.78	690,715
Weighted Avg	0.96	0.97	0.96	690,715

Tabella 10 Classification report del modello Ibrido

Come per i modelli precedenti per la classe “Survived” (0) il modello ibrido raggiunge una precisione del 97%, un recall del 99% e un F1-score del 98% (Tabella 10). Questi valori quindi confermano un’ottima capacità del modello nel riconoscimento dei soggetti non a rischio. Per la classe “Deceased” (1) invece, le prestazioni del modello ibrido risultano migliori rispetto a quelle mostrate dai modelli precedenti, la precisione raggiunge un valore pari al 79%, un recall del 47% e un F1-score del 59 % (Tabella 10).

Questo risultato è dovuto molto probabilmente alla capacità del modello ibrido di unire le informazioni derivanti dalle due tipologie di dato. La rappresentazione testuale accurata dei modelli pre-addestrato unita ai valori delle variabili strutturate permette di identificare meglio i soggetti a rischio, nonostante il problema dello sbilanciamento.

L’accuratezza complessiva del modello si attesta al 97%, con una media macro degli F1-score pari al 78% e una media ponderata del 96%.

L’analisi della confusion matrix (*Grafico 27*) fornisce conferma delle osservazioni emerse dal classification report. Il dato dei falsi negativi che era particolarmente preoccupante nei precedenti approcci adesso si riduce notevolmente a 17,792 casi, a riprova che questo approccio è molto promettente.

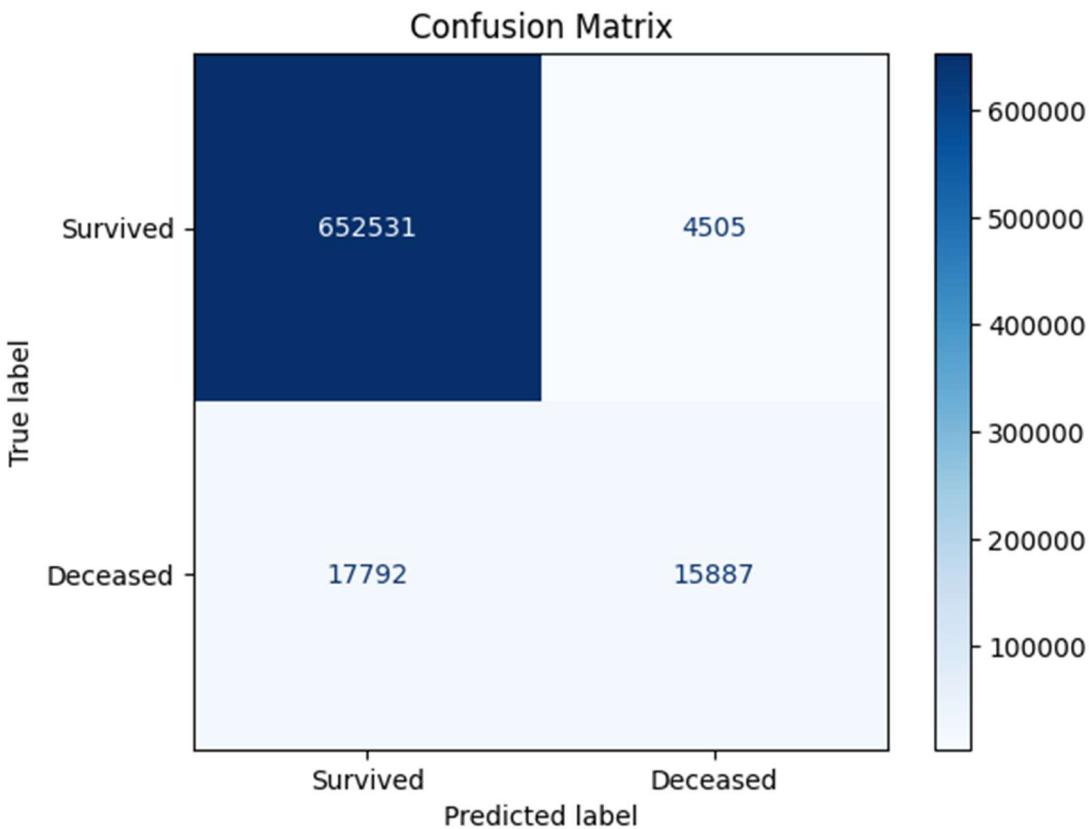


Grafico 27 matrice confusione del modello Ibrido

Le curve di apprendimento del modello ibrido evidenziano che, con l'aumentare del numero di epoche, il divario tra le metriche di accuracy e di loss calcolate sul training set e sul test set si riduce progressivamente fino a stabilizzarsi. (*Grafici 28 e 29*) Questo andamento dimostra una solida capacità di generalizzazione e una costante riduzione dell'overfitting; il modello è in grado di mantenere una buona performance non solo sui dati di addestramento ma anche su nuovi dati non visti durante la fase di training.

La curva ROC per il modello ibrido evidenzia un lieve aumento nella capacità di distinguere tra le due classi rispetto ai modelli testati in precedenza (*Grafico 30*). L'area sotto la curva (AUC) raggiunge un valore pari a 0.96, indicando una buona performance nel discriminare tra pazienti sopravvissuti e deceduti.

Rispetto ai modelli degli approcci precedenti, l'Average Precision del modello ibrido raggiunge un valore pari a 0.69, questo indica che il modello, in media, è in grado di mantenere una precisione del 69% su tutte le soglie di richiamo considerate. Questo risultato potrebbe costituire una performance solida, posiziona il modello in una fascia intermedia, mostrando una buona affidabilità nelle previsioni del rischio di mortalità e un buon potenziale per miglioramenti futuri (*Grafico 31*).

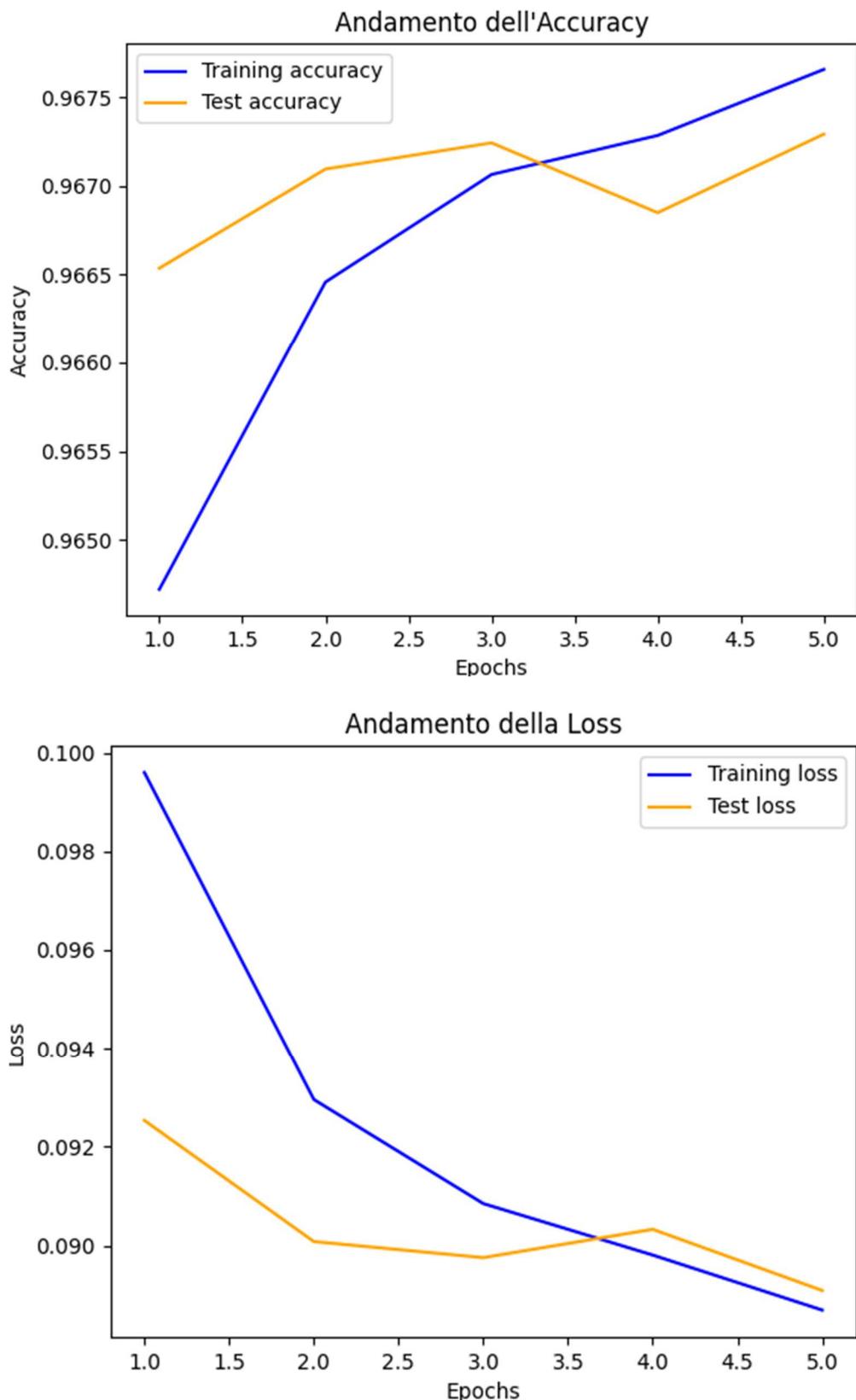


Grafico 28,29 Learning Curves Accuracy e Loss del modello Ibrido

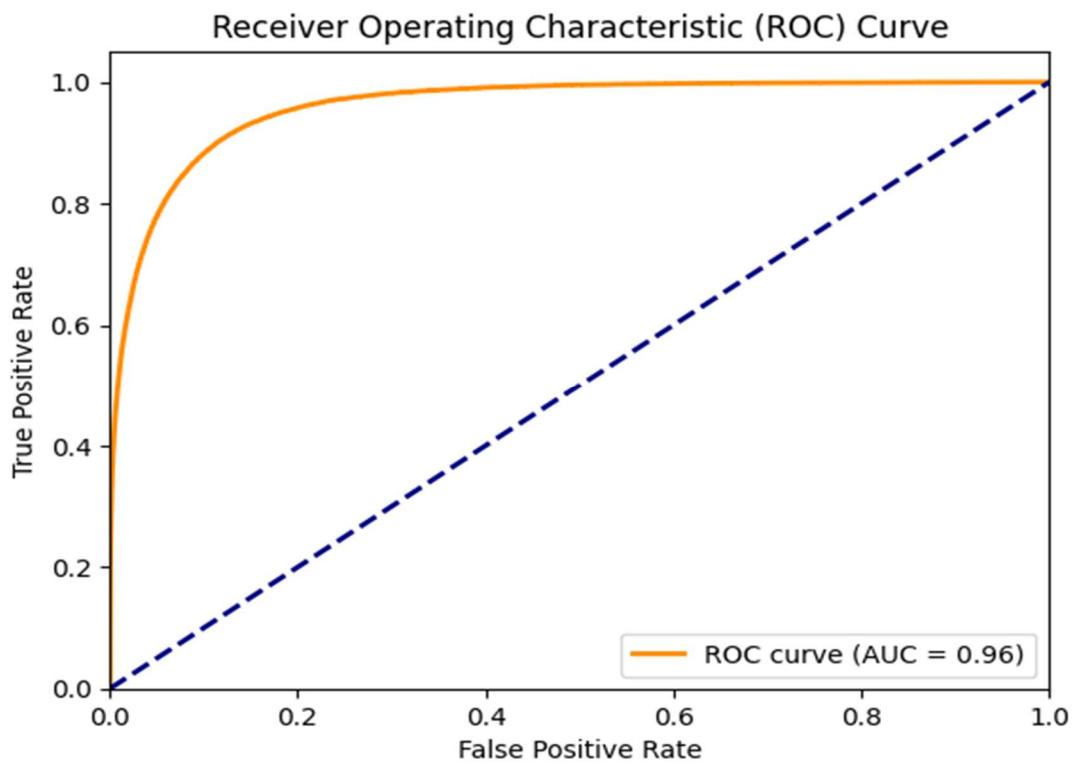


Grafico 30 curva ROC del modello Ibrido

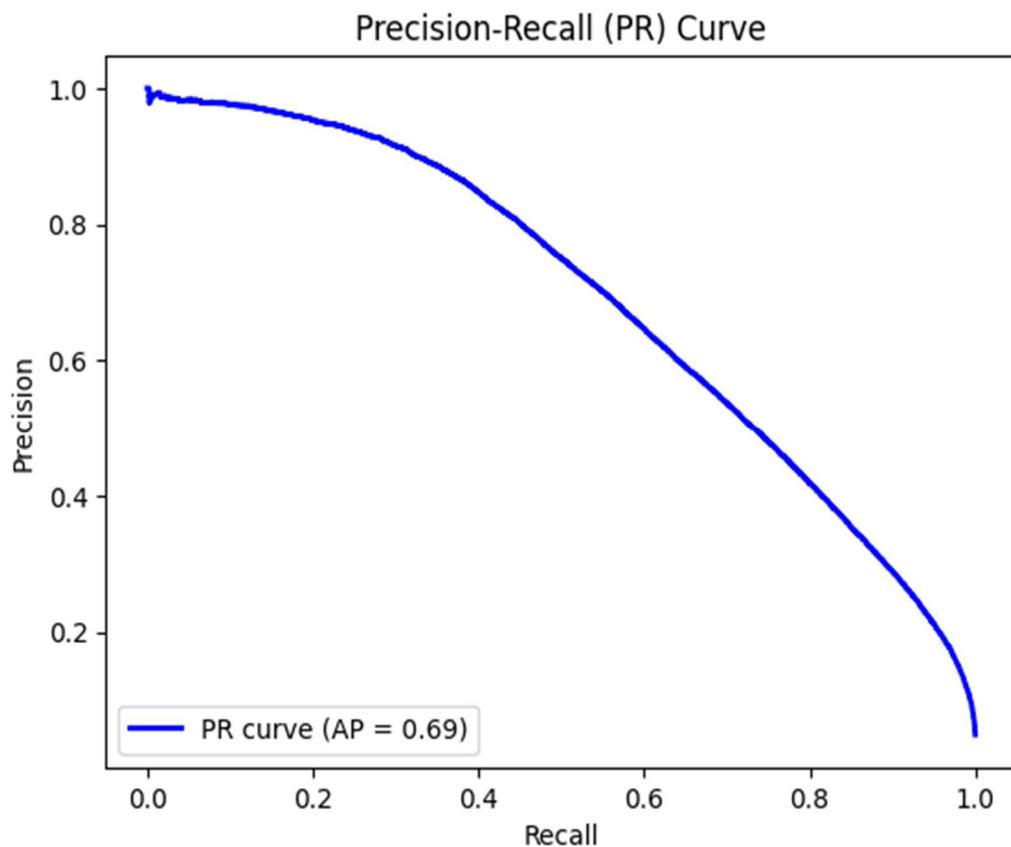


Grafico 31 curva Precision-Recall del modello Ibrido

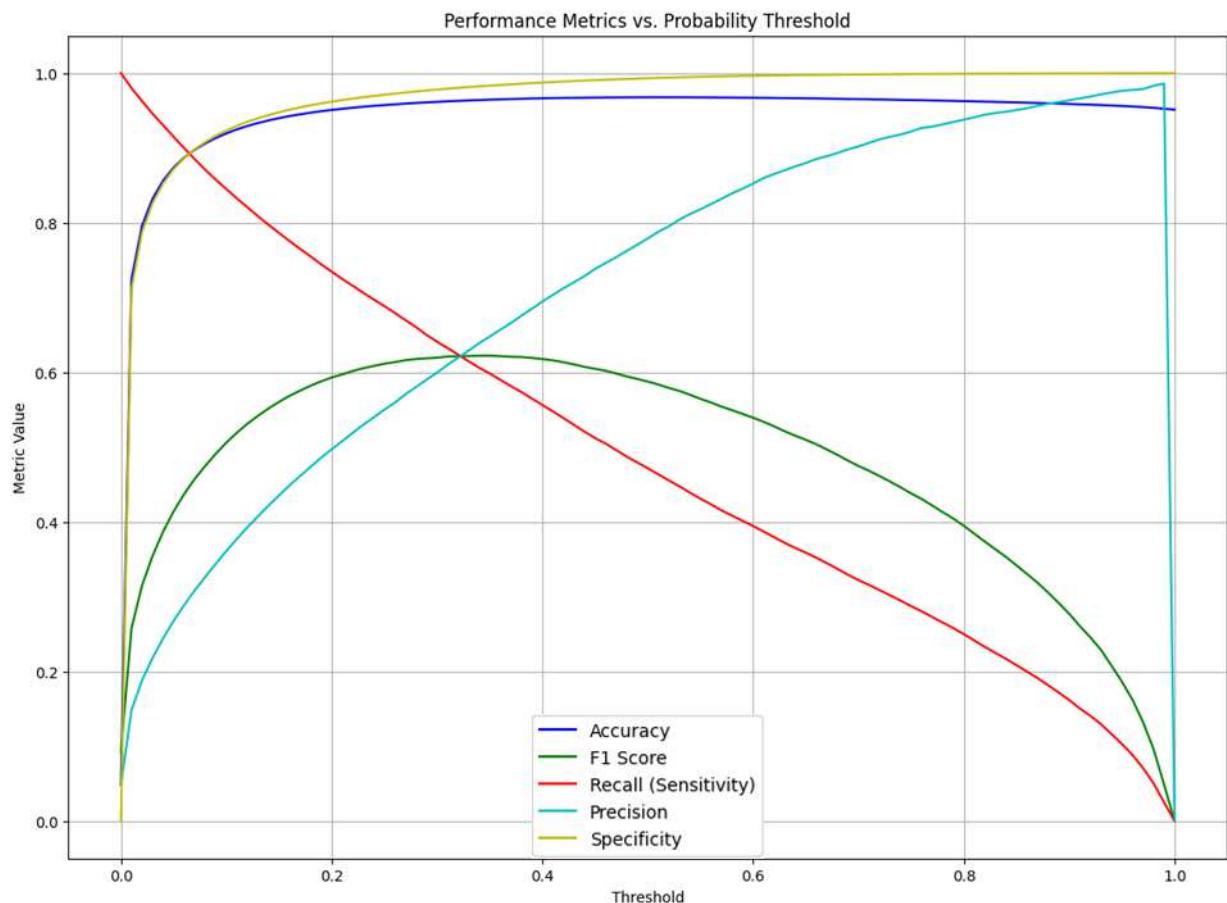


Grafico 32 Grafico Threshold vs Metrics del modello Irido

L'analisi del threshold di classificazione del modello Irido mostra che un possibile abbassamento della soglia (ad esempio 0.27-0.28) potrebbe corrispondere a un valore di Recall per la classe “Deceased” intorno al 61-62%, migliorando così la capacità dei modelli di identificare i soggetti a rischio (*Grafico 32*).

5.4 Interpretazione degli Esiti tramite XAI

Di seguito viene presentata una analisi approfondita di 10 osservazioni classificate dal modello del terzo approccio (Bio_Discharge_Summary_BERT), eseguita mediante LIME, osservando in particolare come il modello interpreti il testo delle diagnosi. È importante sottolineare che, in ciascun testo, la prima parte rappresenta la diagnosi principale – ovvero il primo insieme di parole che definisce in modo compiuto la patologia – mentre le parole successive potrebbero indicare eventuali diagnosi concomitanti, che forniscono ulteriori informazioni sul quadro clinico. L'analisi locale con LIME evidenzia come il modello pre-addestrato attribuisce pesi differenti alla diagnosi principale o alle concomitanze, determinando così la classificazione complessiva del soggetto.

Commenti Dettagliati per le:

Osservazione 1 VP (vero positivo) Classe Reale: Deceased

Classe Prevista: Deceased

Testo Pulito: *malignant neoplasm breast female unspecified malignant neoplasm liver secondary malignant neoplasm bone bone marrow jaundice unspecified newborn cachexia*
(La parte iniziale "malignant neoplasm breast female unspecified" viene interpretata come la diagnosi principale, mentre il resto del testo – che menziona "malignant neoplasm liver secondary malignant neoplasm bone bone marrow jaundice unspecified newborn cachexia" – rappresenta diagnosi concomitanti.)

Classe	Prediction Probabilities
SURVIVED	0,39
DECEASED	0,61

Tabella 11 Prediction probabilities della 1° osservazione

Parola	Valore Importanza LIME	Tipo Contributo
CACHEXIA	0,4	POSITIVO
MALIGNANT	0,09	POSITIVO
BREAST	0,08	POSITIVO
LIVER	0,06	POSITIVO
NEOPLASM	0,04	POSITIVO
JAUNDICE	0,04	NEGATIVO
UNSPECIFIED	0,03	POSITIVO

Tabella 12 Valore importanza LIME della 1° osservazione

Commento: LIME mostra che il modello ha pesato fortemente sia la diagnosi principale – indicata dalle parole legate a neoplasia mammaria – sia le diagnosi concomitanti, in particolare “cachexia”. Le feature “cachexia” e “malignant” giocano un ruolo decisivo nella previsione, evidenziando che la presenza di condizioni oncologiche gravi, associata a segni clinici di deterioramento, contribuisce in modo significativo alla classificazione del soggetto come "Deceased". La spiegazione locale dimostra come il modello attribuisca un forte peso positivo a queste parole, rafforzando la credibilità della decisione e spiegando la corretta classificazione.

Osservazione 2 VP (vero positivo) Classe Reale: Deceased

Classe Prevista: Deceased

Testo Pulito: *malignant neoplasm bronchus lung unspecified secondary malignant neoplasm brain spinal cord cachexia (la diagnosi principale è rappresentata da "malignant neoplasm bronchus lung unspecified", mentre le informazioni aggiuntive, come "secondary malignant neoplasm brain spinal cord cachexia", fungono da diagnosi concomitanti)*

Classe	Prediction Probabilities
SURVIVED	0,44
DECEASED	0,56

Tabella 13 Prediction probabilities della 2° osservazione

Parola	Valore Importanza LIME	Tipo Contributo
CACHEXIA	0,44	POSITIVO
MALIGNANT	0,09	POSITIVO
UNSPECIFIED	0,04	POSITIVO
NEOPLASM	0,02	POSITIVO
SECONDARY	0,01	NEGATIVO
BRAIN	0,01	POSITIVO
LUNG	0,01	POSITIVO

Tabella 14 Valore importanza LIME della 2° osservazione

Commento: LIME evidenzia come il termine “cachexia” e il riferimento al termine “malignant”, presente nelle diagnosi principali e concomitanti, rafforzino l’impatto della diagnosi principale, portando a una corretta identificazione del soggetto come “Deceased”. Inoltre, la spiegazione locale rivela una forte influenza delle parole “cachexia” e “malignant”, in particolare in relazione alle neoplasie che interessano sia il sistema respiratorio che quello neurologico, sottolineando come il modello interpreti la presenza di patologie multiple e gravi come un chiaro indicatore di rischio elevato di mortalità.

Osservazione 3 VP (vero positivo) Classe Reale: Deceased

Classe Prevista: Deceased

Testo Pulito: *specified complications procedures elsewhere classified perforation intestine severe sepsis septic shock* (La prima parte del testo – "specified complications procedures elsewhere classified perforation intestine" – costituisce il nucleo della diagnosi principale, mentre "severe sepsis septic shock" rappresenta condizioni concomitanti critiche.)

Classe	Prediction Probabilities
SURVIVED	0,41
DECEASED	0,59

Tabella 15 Prediction probabilities della 3° osservazione

Parola	Valore Importanza LIME	Tipo Contributo
SHOCK	0,4	POSITIVO
PROCEDURES	0,18	NEGATIVO
SEVERE	0,13	POSITIVO
SEPSIS	0,09	POSITIVO
SEPTIC	0,05	POSITIVO
COMPLICATIONS	0,05	NEGATIVO
PERFORATION	0,05	POSITIVO

Tabella 16 Valore importanza LIME della 3° osservazione

Commento: Le spiegazioni LIME evidenziano come le parole "sepsis" e "septic shock" abbiano un impatto determinante nella previsione del decesso. Questi termini, presenti nella parte concomitante, vengono considerati dal modello come indicatori di complicanze gravi e, quando integrati con la diagnosi principale, contribuiscono in modo significativo all'accuratezza della previsione. L'interpretazione locale offerta da LIME mostra chiaramente che il modello associa tali condizioni critiche a un elevato rischio di mortalità, confermando così la coerenza tra la gravità clinica osservata e la decisione predittiva.

Osservazione 4 VP (vero positivo) Classe Reale: Deceased

Classe Prevista: Deceased

Testo Pulito: *malignant neoplasm parts bronchus lung acute chronic respiratory failure cardiac arrest* (In questo caso, "malignant neoplasm parts bronchus lung" definisce la diagnosi principale, mentre "acute chronic respiratory failure cardiac arrest" rappresenta ulteriori condizioni concomitanti.)

Classe	Prediction Probabilities
SURVIVED	0,01
DECEASED	0,99

Tabella 17 Prediction probabilities della 4° osservazione

Parola	Valore Importanza LIME	Tipo Contributo
ARREST	0,71	POSITIVO
FAILURE	0,08	POSITIVO
ACUTE	0,07	POSITIVO
MALIGNANT	0,07	POSITIVO
NEOPLASM	0,04	POSITIVO
CARDIAC	0,04	POSITIVO
RESPIRATORY	0,04	POSITIVO

Tabella 18 Valore importanza LIME della 4° osservazione

Commento: Le spiegazioni LIME evidenziano come i termini "acute" e "respiratory failure", insieme a "cardiac arrest", rivestano un ruolo decisivo nelle diagnosi concomitanti. Queste feature, quando integrate con la diagnosi principale di "malignant neoplasm", permettono al modello di valutare in maniera integrata diverse condizioni cliniche gravi. In particolare, l'interazione tra una neoplasia maligna e le condizioni di insufficienza respiratoria acuta e arresto cardiaco porta a una forte associazione con un elevato rischio di mortalità, rendendo la previsione coerente con la complessità del quadro clinico.

Osservazione 5 FP (falso positivo) Classe Reale: Survived

Classe Prevista: Deceased

Testo Pulito: *unspecified septicemia coma malignant neoplasm main bronchus*
(La diagnosi principale potrebbe essere identificata dalla prima parte “unspecified septicemia coma”, mentre “malignant neoplasm main bronchus” appare come informazione concomitante.)

Classe	Prediction Probabilities
SURVIVED	0,34
DECEASED	0,66

Tabella 19 Prediction probabilities della 5° osservazione

Parola	Valore Importanza LIME	Tipo Contributo
COMA	0,43	POSITIVO
SEPTICEMIA	0,11	POSITIVO
MALIGNANT	0,07	POSITIVO
UNSPECIFIED	0,07	POSITIVO
BRONCHUS	0,06	POSITIVO
NEOPLASM	0,03	POSITIVO
MAIN	0	NEUTRO

Tabella 20 Valore importanza LIME della 5° osservazione

Commento: In questa osservazione, LIME evidenzia che le parole “septicemia” e “coma” hanno avuto un’influenza predominante, portando il modello a sovrastimare il rischio di mortalità. Nonostante la presenza della diagnosi concomitante legata a neoplasia, il soggetto è sopravvissuto, segnalando un errore nella classificazione (un falso positivo). La spiegazione locale suggerisce che, sebbene queste condizioni siano potenzialmente gravi, altri fattori protettivi o attenuanti non sono stati sufficientemente considerati dal modello. Questa osservazione mostra però che il modello anche quando missclassifica indica una situazione molto grave di salute che potrebbe portare alla morte.

Osservazione 6 FP (falso positivo) Classe Reale: Survived

Classe Prevista: Deceased

Testo Pulito: *malignant neoplasm abdomen malignant neoplasm ascending colon calculus gallbladder acute cholecystitis obstruction defibrillation syndrome* (*La diagnosi principale può essere individuata nelle prime parole “malignant neoplasm abdomen malignant neoplasm ascending colon”, mentre il resto del testo – che include “calculus gallbladder acute cholecystitis obstruction defibrillation syndrome” – rappresenta diagnosi concomitanti.*)

Classe	Prediction Probabilities
SURVIVED	0,37
DECEASED	0,63

Tabella 21 Prediction probabilities della 6° osservazione

Parola	Valore Importanza LIME	Tipo Contributo
DEFIBRINATION	0,5	POSITIVO
MALIGNANT	0,1	POSITIVO
CALCULUS	0,07	NEGATIVO
SYNDROME	0,06	POSITIVO
NEOPLASM	0,05	POSITIVO
GALLBLADDER	0,04	POSITIVO
ACUTE	0,03	POSITIVO

Tabella 22 Valore importanza LIME della 6° osservazione

Commento: In questa osservazione, le spiegazioni LIME mettono in luce che le feature “malignant neoplasm” e “defibrillation syndrome” hanno avuto un peso ELEVATO nel processo decisionale del modello. Tale sovrastima ha portato alla classificazione errata del soggetto come "Deceased", nonostante il paziente sia sopravvissuto. Questo suggerisce che il contributo relativo tra la diagnosi principale e le condizioni concomitanti debba essere ricalibrato, in modo da evitare una sovrastima del rischio e migliorare la precisione predittiva del modello.

Osservazione 7 VP (vero positivo) Classe Reale: Deceased

Classe Prevista: Deceased

Testo Pulito: *malignant neoplasm brain unspecified severe sepsis acute kidney failure unspecified septic shock acidosis* (*La parte iniziale "malignant neoplasm brain unspecified" è considerata la diagnosi principale, mentre "severe sepsis acute kidney failure unspecified septic shock acidosis" fornisce dettagli sulle condizioni concomitanti.*)

Classe	Prediction Probabilities
SURVIVED	0,03
DECEASED	0,97

Tabella 23 Prediction probabilities della 7° osservazione

Parola	Valore Importanza LIME	Tipo Contributo
SHOCK	0,25	POSITIVO
ACIDOSIS	0,07	POSITIVO
UNSPECIFIED	0,07	POSITIVO
MALIGNANT	0,06	POSITIVO
BRAIN	0,06	POSITIVO
SEVERE	0,05	POSITIVO
SEPTIC	0,05	POSITIVO

Tabella 24 Valore importanza LIME della 7° osservazione

Commento: Le spiegazioni LIME evidenziano come le condizioni concomitanti, in particolare “sepsis” e “septic shock”, abbiano un impatto determinante nella previsione di mortalità. Integrando queste informazioni con la diagnosi principale, il modello realizza una classificazione che rispecchia fedelmente la gravità clinica del paziente. La spiegazione locale sottolinea che tali feature rappresentano il contributo più elevato, confermando così la validità e l’accuratezza del modello anche in contesti clinicamente critici.

Osservazione 8 VP (vero positivo) Classe Reale: Deceased

Classe Prevista: Deceased

Testo Pulito: *malignant neoplasm bronchus lung unspecified secondary malignant neoplasm brain spinal cord secondary unspecified malignant neoplasm intrathoracic lymph nodes cachexia cardiac arrest (Qui, la diagnosi principale si identifica nella prima parte "malignant neoplasm bronchus lung unspecified", mentre le diagnosi concomitanti – che includono ulteriori neoplasie, "cachexia" e "cardiac arrest" – arricchiscono il profilo clinico.)*

Classe	Prediction Probabilities
SURVIVED	0,01
DECEASED	0,99

Tabella 25 Prediction probabilities della 8° osservazione

Parola	Valore Importanza LIME	Tipo Contributo
ARREST	0,43	POSITIVO
CACHEXIA	0,24	POSITIVO
CARDIAC	0,1	POSITIVO
MALIGNANT	0,04	POSITIVO
SPINAL	0,03	POSITIVO
BRAIN	0,02	POSITIVO
NEOPLASM	0,02	POSITIVO

Tabella 26 Valore importanza LIME della 8° osservazione

Commento: Le spiegazioni LIME evidenziano come la combinazione di condizioni gravi, sia nella diagnosi principale che nelle concomitanze, come "cachexia" e "cardiac arrest", determini una forte probabilità di mortalità. In particolare, queste feature risultano tra le più influenti, e la presenza di molteplici neoplasie, associata a tali condizioni critiche come l'arresto cardiaco, viene interpretata dal modello come un forte indicatore del rischio. Questo approccio integrato porta a una classificazione corretta, confermando la capacità del modello di valutare adeguatamente il quadro clinico complesso del paziente.

Osservazione 9 FP (falso positivo) Classe Reale: Survived

Classe Prevista: Deceased

Testo Pulito: *acute respiratory failure renal failure unspecified malignant neoplasm stomach unspecified site malignant neoplasm liver secondary cachexia* (*La diagnosi principale è individuabile in “acute respiratory failure” mentre “renal failure unspecified” “malignant neoplasm stomach unspecified site” “malignant neoplasm liver secondary”, “cachexia” funge da concomitanza che rafforza il quadro clinico.*)

Classe	Prediction Probabilities
SURVIVED	0,19
DECEASED	0,81

Tabella 27 Prediction probabilities della 9° osservazione

Parola	Valore Importanza LIME	Tipo Contributo
CACHEXIA	0,31	POSITIVO
RESPIRATORY	0,14	POSITIVO
FAILURE	0,14	POSITIVO
ACUTE	0,12	POSITIVO
LIVER	0,05	POSITIVO
RENAL	0,03	POSITIVO
UNSPECIFIED	0,03	POSITIVO

Tabella 28 Valore importanza LIME della 9° osservazione

Commento: LIME evidenzia che le feature “respiratory failure” e “cachexia” hanno avuto un peso eccessivo nella decisione del modello, portando a una sovrastima del rischio di mortalità e a una classificazione errata. Questo suggerisce che il modello potrebbe non aver contestualizzato adeguatamente la relazione tra la diagnosi principale e le concomitanze, trascurando altri segnali clinici rilevanti. Di conseguenza, potrebbe essere necessaria una revisione del bilanciamento delle feature per migliorare l’accuratezza predittiva.

Osservazione 10 FP (falso positivo) Classe Reale: Survived

Classe Prevista: Deceased

Testo Pulito: *malignant neoplasm upper lobe bronchus lung cachexia acute respiratory* *failure*

(*La diagnosi principale, identificata da "malignant neoplasm upper lobe bronchus lung", viene integrata dalle informazioni concomitanti "cachexia acute respiratory failure".*)

Classe	Prediction Probabilities
SURVIVED	0,28
DECEASED	0,72

Tabella 29 Prediction probabilities della 10° osservazione

Parola	Valore Importanza LIME	Tipo Contributo
CACHEXIA	0,45	POSITIVO
FAILURE	0,16	POSITIVO
ACUTE	0,11	POSITIVO
MALIGNANT	0,11	POSITIVO
UPPER	0,05	NEGATIVO
LUNG	0,04	POSITIVO
LOBE	0,04	NEGATIVO

Tabella 30 Valore importanza LIME della 10° osservazione

Commento: LIME evidenzia che le feature “cachexia” e “respiratory failure” hanno avuto un impatto predominante sulla decisione del modello, portando a una classificazione errata. Questo suggerisce che il modello potrebbe attribuire un peso eccessivo a queste condizioni, che aumenta la presenza di falsi positivi. La spiegazione locale indica la necessità di una migliore calibrazione tra l’influenza della diagnosi principale e quella delle diagnosi concomitanti, al fine di migliorare l’accuratezza predittiva.

5.5 Interpretazione dei Risultati

Nel grafico 33 sono state riportate i risultati principali del nostro studio. I risultati evidenziano come l'integrazione di fonti dati eterogenee, unitamente all'impiego di modelli avanzati di NLP e ML, costituisca una strategia efficace per la classificazione del rischio clinico. In particolare, l'approccio ibrido, che combina dati strutturati e non strutturati, ha dimostrato prestazioni superiori, riuscendo a superare le limitazioni insite nell'utilizzo di tecniche isolate. Tale combinazione permette di ottenere una visione completa e accurata del profilo clinico del paziente, elemento essenziale per l'identificazione tempestiva dei soggetti a rischio di mortalità.

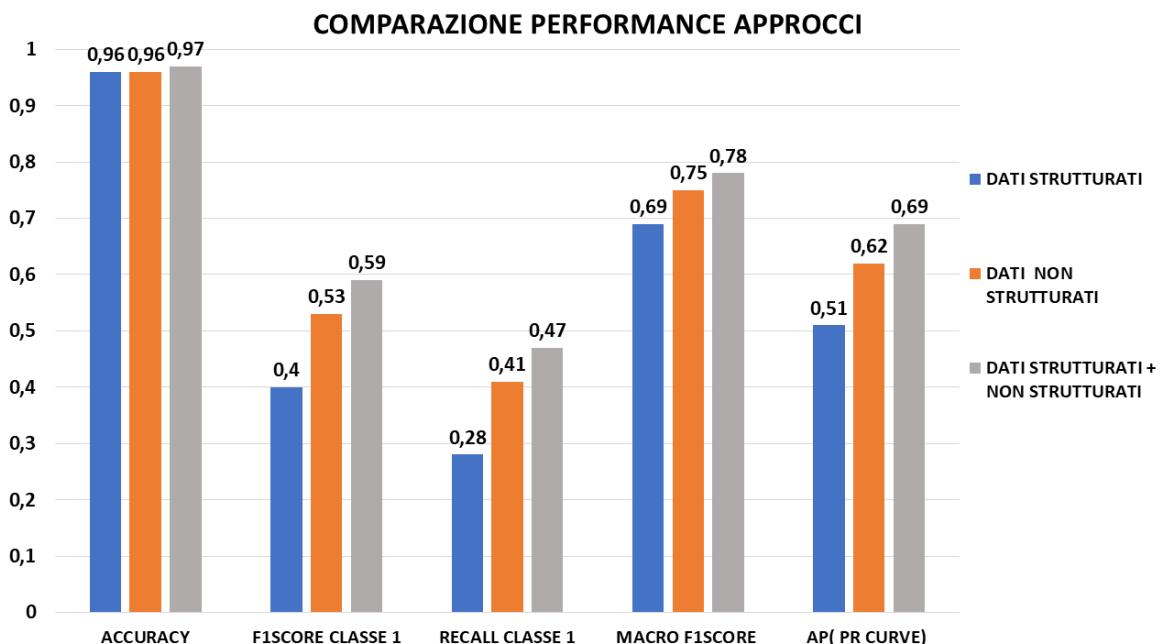


Grafico 33 confronto performance dei diversi approcci

In sintesi, i principali risultati ottenuti dai diversi modelli possono essere schematizzati come segue:

- L'applicazione di modelli sui dati strutturati (XGBoost e Random Forest) ha evidenziato: un'elevata accuratezza complessiva, ma la capacità di riconoscere correttamente i soggetti potenzialmente a rischio di mortalità si è rivelata insufficiente.
- L'utilizzo di rappresentazioni testuali tradizionali (Bag-of-Words e TF-IDF) ha permesso una semplice trasformazione del testo in vettori numerici, ottenendo performance leggermente migliori rispetto ai soli dati strutturati, la rappresentazione testuale, seppur priva di contesto semantico avanzato, apporta un beneficio nella capacità di separare i

soggetti a rischio. Tuttavia, la perdita delle relazioni contestuali e della complessità semantica limita la capacità discriminante dei modelli NLP classici in un contesto clinico.

- L'impiego del modello pre-addestrato per dati non strutturati (Bio_Discharge_Summary_BERT) ha rappresentato un notevole passo avanti: grazie alla sua capacità di cogliere le relazioni contestuali e le sfumature semantiche del linguaggio medico, si è registrato un miglioramento significativo nelle metriche. Questo risultato conferma il potenziale dei modelli basati su Transformer per l'analisi dei testi clinici, riducendo falsi positivi e negativi e migliorando la discriminazione tra le classi.
- L'approccio ibrido (dati strutturati + dati non strutturati), che combina le informazioni anagrafiche, demografiche e mediche con quelle testuali delle diagnosi, ha prodotto il miglior modello complessivo. L'integrazione delle diverse tipologie di dati ha permesso di rappresentare in maniera più completa il quadro clinico, affinando la soglia decisionale e mitigando l'effetto negativo dello sbilanciamento.

Un'analisi approfondita tramite LIME ha offerto una visione trasparente del funzionamento del modello black-box, rivelando come le diverse componenti testuali della diagnosi influenzino la classificazione. Il modello attribuisce un peso significativo sia alla diagnosi principale, sia alle diagnosi concomitanti e la combinazione di questi elementi insieme a specifici termini critici risulta determinante per la decisione finale. Tali termini, che rappresentano condizioni cliniche gravi, risultano essere forti indicatori del rischio di mortalità. LIME ha consentito di visualizzare chiaramente come la presenza di tali condizioni aumenti la probabilità di classificare un soggetto come "Deceased", confermando la coerenza del modello con la letteratura clinica. Inoltre, la capacità di LIME di decomporre il contributo di ciascun termine ha permesso sia di individuare quali parti della diagnosi (principale o concomitante) abbiano un impatto maggiore sulla decisione, sia di identificare eventuali bias, come la tendenza del modello a sovrastimare il peso di alcune diagnosi concomitanti, suggerendo la necessità di ulteriori ricalibrazioni. Le osservazioni che hanno portato a falsi positivi, hanno evidenziato un'eccessiva influenza delle diagnosi concomitanti rispetto a quella della diagnosi principale, evidenziato la necessità di integrare ulteriori variabili cliniche e di ribilanciare l'impatto tra diagnosi principale e concomitanti, per migliorare ulteriormente la precisione del modello.

5.6 Punti di Forza

L'uso combinato di informazioni strutturate e non strutturate ha sfruttato sinergie informative, consentendo una rappresentazione più completa del quadro clinico, migliorando la capacità discriminante del modello.

L'analisi condotta su un vasto dataset proveniente dalle SDO italiane garantisce robustezza e generalizzabilità dei risultati, offrendo spunti preziosi per applicazioni cliniche reali. L'utilizzo di 3.5 milioni di osservazioni ha garantito una solida base statistica, mitigando parzialmente i rischi di overfitting.

Inoltre, l'adozione di modelli pre-addestrati basati su Transformer (come Bio_Discharge_Summary_BERT) ha evidenziato come una sofisticata rappresentazione semantica possa incrementare significativamente le performance.

Infine, l'implementazione di Metodi di Explainable AI tramite LIME, ha permesso di esplorare come le diverse componenti testuali della diagnosi influenzino la corretta classificazione dei soggetti a rischio.

5.7 Limitazioni dello Studio

I modelli più sofisticati, quali Bio_Discharge_Summary_BERT e architetture ibride, richiedono risorse computazionali significative, il che può renderne l'adozione impraticabile in contesti con risorse limitate.

In particolare, il modello pre-addestrato (Bio_Discharge_Summary_BERT) mostra talvolta una perdita di informazioni contestuali e difficoltà nell'interpretare correttamente le diagnosi e le relazioni tra le diagnosi concomitanti, compromettendo la capacità di discriminare efficacemente i casi critici. Queste problematiche potrebbero essere riconducibili a bias nella valutazione, con un'eccessiva attribuzione del peso ad alcune diagnosi concomitanti e a determinati termini critici, suggerendo la necessità di una calibrazione più accurata.

Nonostante l'utilizzo di LIME, i modelli basati su DL rimangono meno trasparenti e difficilmente interpretabili e rendono necessaria una maggiore attenzione nell'analisi delle spiegazioni.

Inoltre, lo sbilanciamento del dataset, con una marcata disparità tra la classe dei sopravvissuti e quella dei deceduti limita la capacità predittiva dei modelli, compromettendone la generalizzabilità nonostante l'elevato volume di dati disponibili.

Infine, poiché i dati provengono da un singolo contesto nazionale, ci sono potenziali fattori confondenti non direttamente controllabili; pertanto, la validazione su ulteriori dataset, anche di carattere internazionale, potrebbe essere utile e necessaria per confermare l'efficacia predittiva del modello.

6. Discussione

I risultati evidenziano come l'integrazione di fonti dati eterogenee, unitamente all'impiego di modelli avanzati di NLP e ML, costituisca una strategia efficace per la classificazione del rischio clinico. In particolare, l'approccio ibrido, che combina dati strutturati e non strutturati, ha dimostrato prestazioni superiori, riuscendo a superare le limitazioni insite nell'utilizzo di tecniche isolate. Tale combinazione permette di ottenere una visione completa e accurata del profilo clinico del paziente, elemento essenziale per l'identificazione tempestiva dei soggetti a rischio di mortalità.

È fondamentale evidenziare che la scelta tra modelli complessi o meno deve essere guidata dall'obiettivo di ricerca. Se il raggiungimento di un determinato scopo richiede un impegno computazionale elevato, questo aspetto va valutato con attenzione, poiché un modello troppo oneroso potrebbe risultare impraticabile in contesti operativi reali. Di conseguenza, la selezione del modello deve tener conto delle peculiarità del contesto applicativo e delle limitazioni economiche e infrastrutturali, garantendo così un equilibrio ottimale tra prestazioni elevate e sostenibilità computazionale.

L'analisi delle spiegazioni tramite LIME ha ulteriormente confermato la validità del modello pre-addestrato, evidenziando come il contributo delle diagnosi principali e concomitanti sia in linea con la letteratura clinica.

Le prospettive di ricerca futura si articolano in diversi ambiti:

- **Integrazione di Variabili Cliniche Addizionali:** Ampliare il modello integrando ulteriori variabili cliniche, quali test di laboratorio, referti di immagini diagnostiche e biomarcatori, per potenziare la capacità predittiva. Ad esempio implementando le SDO con altri dati dei pazienti, archiviati in testo non strutturato e/o dataset isolati, potrebbe migliorare la capacità di prevedere i risultati dei pazienti come dimostrato da alcuni studi recenti (35).
- **Tecniche di Bilanciamento del Dataset:** Sperimentare metodi avanzati di oversampling, undersampling o tecniche di generazione sintetica dei dati per migliorare la rappresentazione della classe “Deceased”.
- **Ottimizzazione delle Soglie Decisionali:** Implementare strategie di tuning del threshold per ridurre i falsi positivi e falsi negativi, affinando ulteriormente il processo decisionale.

- Evoluzione dei Modelli NLP: Valutare l’impiego di modelli transformer di nuova generazione e tecniche di data augmentation per arricchire le rappresentazioni semantiche dei testi clinici.
- Sviluppo di Metodi di Explainable AI: Potenziare l’utilizzo di strumenti di interpretabilità (oltre a LIME, come SHAP) per rendere il modello più trasparente e favorire l’adozione clinica dei sistemi predittivi.
- Validazione clinica: Collaborare con medici per testare l’impatto del modello nelle decisioni reali, misurando metriche orientate all’outcome (es. riduzione mortalità).
- Riduzione dei Costi Computazionali: Esplorare tecniche di compressione o quantizzazione dei modelli per renderli più efficienti senza compromettere le performance.

In sintesi, la tesi ha dimostrato come la combinazione di tecniche tradizionali e avanzate di ML e NLP, applicate a dati sanitari complessi e provenienti da fonti big data, possa portare allo sviluppo di modelli di classificazione robusti ed efficaci. Sebbene siano emerse limitazioni legate allo sbilanciamento del dataset e ai costi computazionali, l’approccio ibrido, che integra informazioni strutturate e non strutturate, risulta il più promettente per l’identificazione dei soggetti a rischio di mortalità, nonostante le sfide poste dallo sbilanciamento del dataset e dalla complessità dei dati clinici.

L’ulteriore affinamento delle tecniche di bilanciamento, la calibrazione dei modelli e l’adozione di metodologie di interpretabilità avanzate rappresentano direzioni essenziali per tradurre questi risultati in strumenti clinici affidabili e utili per il supporto alle decisioni terapeutiche.

Questo lavoro oltre ad aumentare la fiducia nell’intelligenza artificiale in ambito medico, fornisce un contributo significativo per migliorare l’accuratezza e la trasparenza dei sistemi di intelligenza artificiale applicati alla medicina clinica offrendo spunti preziosi per future ricerche nel campo dell’analisi predittiva in ambito sanitario, supportando anche un utilizzo più efficace delle risorse ospedaliere. I risultati di questo studio forniscono indicazioni strategiche per un uso ottimizzato dei Big Data sanitari, con importanti ricadute sulla personalizzazione delle terapie e sull’efficienza nella gestione del sistema sanitario.

7. Bibliografia

1. *HealthTech360 – Intelligenza Artificiale per la Cartella Clinica Elettronica: Benefici, Casi d’Uso e Soluzioni* <https://www.healthtech360.it/salute-digitale/intelligenza-artificiale/intelligenza-artificiale-per-la-cartella-clinica-elettronica-benefici-casi-duso-e-soluzioni/>
2. *AI4Business – Natural Language Processing: Nuove Applicazioni nella Sanità* <https://www.ai4business.it/intelligenza-artificiale/ai-manager/natural-language-processing-nuove-applicazioni-nella-sanita/>
3. ICD-9-CM labels <https://www.cms.gov/medicare/coding-billing/icd-10-codes/icd-9-cm-diagnosis-procedure-codes-abbreviated-and-full-code-titles>
4. Il manuale ICD9CM
https://www.salute.gov.it/imgs/C_17_pubblicazioni_2251_allegato.pdf
5. Busnatu Ş, Niculescu AG, Bolocan A, Petrescu GED, Păduraru DN, Năstasă I, Lupușoru M, Geantă M, Andronic O, Grumezescu AM, Martins H. *Clinical Applications of Artificial Intelligence-An Updated Overview*. J Clin Med. 2022 Apr 18;11(8):2265. <https://doi.org/10.3390/jcm11082265>
6. Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P. S., & He, L. *A Survey on Text Classification: From Traditional to Deep Learning* <https://doi.org/10.1145/3495162>
7. Hugging Face – Synthetic Medical Data and Models Collection
<https://huggingface.co/collections/hf4h/synthetic-medical-data-and-models-64f9bf3446f3f06f5abdb770>
8. Hugging Face – Clinical Language Models Collection
<https://huggingface.co/collections/hf4h/clinical-language-models-64f9c1cd0cedc04f3caca264>
9. Hugging Face – Biomedical Language Models Collection
<https://huggingface.co/collections/hf4h/biomedical-language-models-64f9c1e740eb5daaeb828615>
10. Elixhauser A, Steiner C, Harris DR, Coffey RM. *Comorbidity measures for use with administrative data*. Med Care. 1998 Jan;36(1):8-27. <https://doi.org/10.1097/00005650-199801000-00004>

11. HCUP – Comorbidity Tool <https://hcup-us.ahrq.gov/toolssoftware/comorbidity/comorbidity.jsp#download>
12. MCHP – Concept View (conceptID: 1436) <http://mchp-appserv.cpe.umanitoba.ca/viewConcept.php?conceptID=1436>
13. Naviden – Introduction to XAI (GitHub) <https://github.com/Naviden/Introduction-to-XAI>
14. PNRR Salute – Home Assistenza Ospedaliera
<https://www.pnrr.salute.gov.it/portale/assistenzaOspedaliera/homeAssistenzaOspedaliera.jsp>
15. PNRR Salute – Dettaglio Contenuti Assistenza Ospedaliera (id=6133)
<https://www.pnrr.salute.gov.it/portale/assistenzaOspedaliera/dettaglioContenutiAssistenzaOspedaliera.jsp?lingua=italiano&id=6133&area=ricoveriOspedalieri&menu=rilevazione>
16. PNRR Salute – Dettaglio Contenuti Assistenza Ospedaliera (id=1237)
<https://www.pnrr.salute.gov.it/portale/assistenzaOspedaliera/dettaglioContenutiAssistenzaOspedaliera.jsp?lingua=italiano&id=1237&area=ricoveriOspedalieri&menu=vuoto>
17. PNRR Salute – Dettaglio Contenuti Assistenza Ospedaliera (id=1232)
<https://www.pnrr.salute.gov.it/portale/assistenzaOspedaliera/dettaglioContenutiAssistenzaOspedaliera.jsp?lingua=italiano&id=1232&area=ricoveriOspedalieri&menu=rilevazione>
18. Lam BD, Chrysafi P, Chiasakul T, Khosla H, Karagkouni D, McNichol M, Adamski A, Reyes N, Abe K, Mantha S, Vlachos IS, Zwicker JI, Patell R. *Machine learning natural language processing for identifying venous thromboembolism: systematic review and meta-analysis*. Blood Adv. 2024 Jun 25;8(12):2991-3000.
[https://doi.org/10.1182/bloodadvances.2023012200.](https://doi.org/10.1182/bloodadvances.2023012200)
19. Mikolov Tomas, Chen Ken, Corrado Greg, Dean Jeffrey *Efficient Estimation of Word Representations in Vector Space* Jan 2013 <https://doi.org/10.48550/arXiv.1301.3781>
20. Jeffrey Pennington, Richard Socher, Christopher D. Manning *GloVe: Global Vectors for Word Representation* Feb 2019
<https://doi.org/10.48550/arXiv.1902.11004>
21. Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov
<https://github.com/facebookresearch/fastText>

22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. *Attention Is All You Need* <https://doi.org/10.48550/arXiv.1706.03762>
23. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K.
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
 Google AI Language <https://doi.org/10.48550/arXiv.1810.04805>
24. Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jin, D., Naumann, T., & McDermott, P. *Publicly Available Clinical BERT Embeddings*. Harvard-MIT; MIT CSAIL; Microsoft Research <https://doi.org/10.48550/arXiv.1904.03323>
25. Shreeraj – Explainable AI for Communicable Disease Prediction: A Breakthrough in Healthcare Technology (Medium) <https://medium.com/@shreeraj260405/explainable-ai-for-communicable-disease-prediction-a-breakthrough-in-healthcare-technology-662d66efcdb3>
26. Kaggle – Sentiment Analysis BERT XAI
<https://www.kaggle.com/code/tahmidmir/sentiment-analysis-bert-xai#5.-Fine-Tuning-BERT>
27. Rajistics – Explaining Predictions from Transformer Models (Medium)
<https://medium.com/@rajistics/explaining-predictions-from-transformer-models-55ab9c6cab24>
28. Yashvaant Lakham – Decoding the Black Box (Medium)
<https://yashvaantlakham73.medium.com/decoding-the-black-box-3e608417a1e1>
29. Social Science Data Lab, Uni. di Mannheim – BERT & Explainable AI
 Disponibile all’indirizzo: <https://www.mzes.uni-mannheim.de/socialsciencedatalab/article/bert-explainable-ai/>
30. SHAP – Text Examples (Read the Docs)
https://shap.readthedocs.io/en/latest/text_examples.html
31. Ding, J.-E., Nguyen Minh Thao, P., Peng, W.-C., Wang, J.-Z., Chug, C.-C., Hsieh, M.-C., Tseng, Y.-C., Chen, L., Luo, D., Wu, C., Wang, C.-T., Chen, P.-F., Liu, F., & Hung, F.-M. *Large Language Multimodal Models For 5-Year Chronic Disease Cohort Prediction Using Ehr Data* . March 2024 <https://doi.org/10.48550/arXiv.2403.04785>

32. Lyu, W., Dong, X., Wong, R., Zheng, S., Abell-Hart, K., Wang, F., & Chen, C. *A Multimodal Transformer: Fusing Clinical Notes with Structured EHR Data for Interpretable In-Hospital Mortality Prediction*. Stony Brook University, Stony Brook, NY, USA Sept 2022 <https://doi.org/10.48550/arXiv.2208.10240>
33. Xu, K., Lam, M., Pang, J., Gao, X., Band, C., Mathur, P., Papay, F., Khanna, A. K., Cywinski, J. B., Maheshwari, K., Xie, P., & Xing, E. P. *Multimodal Machine Learning for Automated ICD Coding*. Petuum Inc, Pittsburgh, PA, USA; Cleveland Clinic, Cleveland, OH, USA <https://doi.org/10.48550/arXiv.1810.13348>
34. Goh, K.H., Wang, L., Yeow, A.Y.K. et al. *Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare*. Nat Commun 12, 711 (2021). <https://doi.org/10.1038/s41467-021-20910-4>
35. Jee J, Fong C, Pichotta K, Tran TN, et al. MSK Cancer Data Science Initiative Group. *Automated real-world data integration improves cancer outcome prediction*. Nature. 2024 Dec;636(8043):728-736 <https://doi.org/10.1038/s41586-024-08167-5>