

# Reti neurali e Algoritmi: diversi modelli, diverse prestazioni

Luca Evangelisti, Gianluca Milani, Alessandro Manucci

## Algoritmi e Reti Neurali ai raggi-x

Il confronto di algoritmi di machine learning tradizionali e di reti neurali è fondamentale per valutare l'efficacia dell'intelligenza artificiale nella classificazione di immagini, poiché l'applicazione di questi modelli a problemi concreti potrebbe consentire soluzioni più rapide ed efficienti rispetto a quelle attuali.

In questo studio, analizziamo le prestazioni di tre reti neurali (VGG-16, DenseNet169 e Multi-Layer Perceptron) e tre algoritmi di machine learning tradizionali (Support Vector Machine, K-Nearest Neighbors e Decision Tree) applicati alla classificazione di immagini radiografiche.

I modelli sono stati valutati secondo metriche standard come accuratezza, precisione e analisi delle matrici di confusione. L'analisi comparativa ha inoltre permesso di identificare i punti di forza e le limitazioni di ciascun modello, fornendo indicazioni utili per la scelta dell'algoritmo più appropriato in base a diverse situazioni.

## Motivazione e obiettivi dello studio

Negli ultimi anni, il mondo del machine learning ha visto un rapido sviluppo, con un crescente interesse nel confronto tra diversi approcci di machine learning e di reti neurali. In particolare, la scelta su quali siano i modelli migliori per compiti di classificazione rimane un tema dibattuto, influenzato da fattori quali la capacità di generalizzazione, l'efficienza computazionale e la qualità dei dati.

Questo studio si propone di analizzare in modo sistematico le prestazioni di diversi algoritmi di machine learning e reti neurali nel contesto della classificazione di immagini. Utilizzando il dataset MURA come banco di prova, è stato possibile confrontare i modelli in tutte le loro sfaccettature. È stato deciso di utilizzare proprio questo dataset per diverse ragioni.

Innanzitutto, è una delle più ampie raccolte di immagini radiografiche etichettate disponibili pubblicamente, il che consente di testare i modelli su un dataset realistico e diversificato. Inoltre, la variabilità nelle immagini fornisce un contesto ideale per valutare la capacità di generalizzazione degli algoritmi, mettendo alla prova sia modelli tradizionali di machine learning che reti neurali profonde. Un ulteriore aspetto di interesse è il tema stesso del dataset: l'identificazione di anomalie ossee. Questo problema rappresenta una sfida complessa nell'ambito della classificazione di immagini, poiché le anomalie possono presentare grande variabilità nelle loro caratteristiche visive. Analizzare come diversi algoritmi affrontano questa difficoltà permette di ottenere una comprensione più approfondita dei punti di forza e dei limiti di ciascun approccio.

L'articolo è strutturato nel seguente modo: la prossima sezione presenta lo stato dell'arte sull'analisi comparativa tra algoritmi di machine learning tradizionali e reti neurali. Successivamente, vengono descritti il dataset e la metodologia adottata per la preparazione dei dati e l'addestramento dei modelli. Infine, discutiamo i risultati ottenuti, i fattori che li influenzano e consigliamo una possibile direzione futura per proseguire questo studio.

## **Studi precedenti su confronti tra algoritmi**

L'analisi comparativa di algoritmi di machine learning tradizionali e di reti neurali è un argomento di crescente interesse nella comunità scientifica.

Diversi studi hanno evidenziato le prestazioni di algoritmi come Support Vector Machine (SVM), k-Nearest Neighbors (KNN) e Decision Tree (DT) e di reti neurali come Multi-Layer Perceptron (MLP), VGG-16 e DenseNet169, evidenziando vantaggi e limiti di ciascun approccio.

## **Algoritmi**

Gli algoritmi di machine learning sono stati ampiamente utilizzati per la classificazione e il riconoscimento di pattern nei dati. Qui sotto troviamo alcuni studi relativi ai 3 algoritmi (SVM, KNN, Decision tree):

### **1. Support Vector Machine (SVM) - Shin et al., 2016**

Nel 2016, Shin et al. hanno utilizzato le SVM per classificare attività di aritmetica mentale rispetto a uno stato di base, ottenendo performance intorno al 77% con occhi aperti e al 75% con occhi chiusi. Questo studio evidenzia l'efficacia delle SVM nel gestire dati ad alta dimensionalità e nella classificazione di pattern complessi, grazie alla loro capacità di trovare iperpiani che separano le classi con il massimo margine.

### **2. K-Nearest Neighbors (KNN) - Liu et al., 2018**

Nel 2018, Liu et al. hanno proposto un metodo chiamato kTree per migliorare la classificazione kNN. Questo approccio mira a determinare valori ottimali di k per ciascun campione di test, affrontando così le problematiche legate alla scelta del numero di vicini e alla scalabilità. Il metodo kTree prevede una fase di addestramento in cui si apprendono i valori ottimali di k per i campioni di training tramite un nuovo modello di ricostruzione sparsa, seguita dalla costruzione di un albero decisionale (kTree) utilizzando i campioni di training e i valori di k appresi. Durante la fase di test, il kTree fornisce rapidamente il valore ottimale di k per ciascun campione, migliorando l'accuratezza della classificazione rispetto ai metodi kNN tradizionali che utilizzano un k fisso per tutti i campioni.

### **3. Decision Tree - Safavian & Landgrebe, 1991**

Nel 1991, Safavian e Landgrebe hanno pubblicato un'ampia rassegna sulle metodologie dei classificatori ad albero decisionale. Lo studio discute i vantaggi dei DTC, come la capacità di scomporre processi decisionali complessi in una serie di decisioni più semplici, rendendo il modello più interpretabile. Tuttavia, evidenzia anche le sfide legate al rischio di overfitting,

specialmente con dataset complessi. Per mitigare questo problema, gli autori suggeriscono tecniche come la potatura dell'albero e l'uso di criteri di divisione appropriati.

## Reti neurali

L'avanzamento delle reti neurali ha portato a un significativo miglioramento nelle prestazioni di classificazione.

1. **MLP**, sebbene efficace per compiti di classificazione generica, mostra limiti nel trattamento di dati complessi come immagini, non sfruttando la struttura spaziale.  
*L'MLP non sfrutta la struttura spaziale dei dati perché tratta ogni input come indipendente dagli altri. In altre parole, un MLP considera ogni pixel di un'immagine come un'unità separata, senza tener conto delle relazioni spaziali tra i pixel vicini.*
2. **VGG-16** ha introdotto un'architettura basata su una struttura profondamente convoluzionale con strati più uniformi e piccoli filtri  $3 \times 3$ , che migliorano l'estrazione delle caratteristiche mantenendo una maggiore semplicità strutturale (Simonyan & Zisserman, 2014).
3. **DenseNet169** grazie alla connettività densa tra i livelli, favorisce un migliore flusso di informazioni e il riutilizzo delle caratteristiche, risultando particolarmente efficace in molteplici contesti di classificazione (Huang et al., 2017).

Nonostante numerosi studi abbiano esaminato le prestazioni di singoli modelli, mancano confronti sistematici di algoritmi di machine learning e di reti neurali applicati a un medesimo problema. Questo studio mira a colmare tale lacuna attraverso un confronto dettagliato di algoritmi di machine learning e di reti neurali, analizzando le differenze in termini di accuratezza, efficienza computazionale e capacità di generalizzazione applicando tutti i modelli presi in considerazione al dataset MURA X-rays.

L'obiettivo è fornire una comprensione più chiara delle condizioni in cui ciascun modello risulta più vantaggioso, contribuendo alla scelta ottimale del metodo per specifiche applicazioni di classificazione.

## Metodologia

Per avere una migliore comprensione delle scelte e del funzionamento dei vari modelli è necessario prima di tutto osservare la struttura e il contenuto del dataset utilizzato. Successivamente andremo ad analizzare le varie implementazioni degli algoritmi.

## Descrizione del dataset

Il dataset MURA MX-rays (Musculoskeletal Radiographs) è un ampio dataset di radiografie muscoloscheletriche progettato per la ricerca nell'ambito della diagnosi assistita da intelligenza artificiale. Contiene immagini etichettate con l'obiettivo di distinguere tra studi radiografici normali e anormali, rendendolo una risorsa preziosa per sviluppare e valutare algoritmi di classificazione.

## Struttura del dataset

1. **Cartella train**
  - Contiene i dati di addestramento utilizzati per allenare i modelli.
  - Strutturata in sottocartelle che rappresentano diversi studi radiologici delle diverse parti del corpo, contenenti a loro volta cartelle nominate con i numeri identificativi dei pazienti aventi al loro interno immagini in formato .png delle lastre eseguite.
2. **Cartella valid**
  - Contiene i dati di validazione per valutare le prestazioni del modello durante il training.
  - Strutturata in modo analogo alla cartella train, con un file CSV associato, `valid_labeled_studies.csv`, per le etichette.
3. **File CSV**
  - `train_image_paths.csv` e `valid_image_paths.csv`: Contengono i percorsi delle immagini per le cartelle train e valid.
  - `train_labeled_studies.csv` e `valid_labeled_studies.csv`: Collegano ogni studio (o paziente o gruppo di immagini) alla rispettiva etichetta:
    - 1: Studio anormale.
    - 0: Studio normale.

## Caricamento e preprocessing del dataset

Il caricamento e la preparazione dei dati sono stati eseguiti attraverso un approccio strutturato che include cache per migliorare l'efficienza. Di seguito i passaggi principali:

1. **Preprocessing delle immagini**
  - Ogni immagine è stata caricata in scala di grigi utilizzando OpenCV (`cv2.imread`) per ridurre la complessità computazionale.
  - Ridimensionamento uniforme delle immagini a 64x64 pixel.
  - Normalizzazione dei pixel nel range [0, 1] dividendo ogni valore per 255.
  - Appiattimento delle immagini in vettori unidimensionali per facilitarne l'elaborazione.
2. **Cache dei dati**
  - I dati preprocessati sono stati salvati in file di cache (`train_data_cache.pkl` e `valid_data_cache.pkl`) per evitare di rielaborarli ad ogni esecuzione.
  - Se i file di cache sono disponibili, i dati vengono caricati direttamente dalla cache, riducendo significativamente i tempi di caricamento.
3. **Suddivisione dei dati**
  - I dati di addestramento (`X_train`, `y_train`) e validazione (`X_valid`, `y_valid`) sono stati estratti e strutturati utilizzando i file CSV di etichette e le immagini corrispondenti.
  - Ogni immagine è associata a un'etichetta binaria (0 o 1) che rappresenta la normalità o l'anomalia dello studio.

## Creazione di un subset bilanciato

Per ridurre il carico computazionale e bilanciare le classi, è stato creato un subset dei dati di training:

### 1. Divisione per classe

- Sono stati individuati gli indici delle immagini appartenenti a ciascuna classe (normale e anormale) nel set di training.

### 2. Campionamento bilanciato

- Sono stati selezionati 12048 o 2048 campioni dalla classe normale e 11916 o 1916 campioni dalla classe anormale, seguendo la distribuzione delle etichette nel set di validazione.
- Gli indici dei campioni vengono poi combinati e mescolati per evitare un ordine sistematico.

### 3. Dataset finale

- Il subset bilanciato è stato denominato X\_train\_small e y\_train\_small:
  - Dimensione: 24000 o 4000 campioni totali.
  - Distribuzione: Bilanciata rispetto al set di validazione.

## Scelta degli algoritmi

Ora che è chiara la struttura del dataset possiamo iniziare a vedere quali algoritmi sono stati scelti, per quale motivo e come funzionano

### Decision Tree (DT)

Come primo algoritmo per l'analisi è stato scelto di utilizzare Decision Tree (DT), un modello di machine learning che organizza il processo decisionale in una struttura ad albero, composta da nodi decisionali e rami che rappresentano le possibili scelte basate sulle caratteristiche dei dati. Ogni decisione porta a una suddivisione dei dati fino a raggiungere una previsione finale nei nodi foglia.

Una delle ragioni principali per questa scelta è la sua interpretabile rappresentazione grafica, che semplifica l'identificazione dei criteri utilizzati dal modello per separare le classi. Questo aspetto è particolarmente importante in contesti dove è essenziale comprendere e verificare le previsioni del sistema, garantendo un'analisi trasparente e verificabile.

Un altro motivo per scegliere il Decision Tree per il dataset MURA è la sua capacità di funzionare bene anche con dataset sbilanciati, poiché può adattarsi alle distribuzioni delle classi durante la costruzione dell'albero. Questo lo rende utile in contesti come quello medico, dove alcune condizioni potrebbero essere meno rappresentate rispetto ad altre.

Rispetto ad altri modelli più complessi, il Decision Tree richiede meno risorse computazionali sia in fase di addestramento sia in fase di predizione. Questo lo rende adatto per studi preliminari su dataset di grandi dimensioni, dove l'obiettivo è ottenere una prima analisi rapida ed efficace senza incorrere in costi computazionali elevati.

Nel contesto specifico di questo studio, è stata utilizzata una profondità massima di 10 (max\_depth=10) per bilanciare accuratezza e generalizzazione. Questa scelta ha permesso di limitare la complessità del modello, evitando sovraadattamenti, e di mantenere un compromesso efficace tra prestazioni e efficienza computazionale.

L'uso contenuto delle risorse computazionali rende il Decision Tree una soluzione particolarmente vantaggiosa per scenari in cui è necessario testare diverse configurazioni di parametri o confrontare più modelli senza compromettere l'efficienza del sistema. Questo aspetto si rivela essenziale soprattutto in ambiti come la diagnostica medica assistita, dove la rapidità nella generazione delle previsioni può influenzare direttamente i processi decisionali clinici.

Tuttavia, è importante evidenziare alcune limitazioni associate all'uso dei Decision Tree:

#### 1. **Overfitting**

Il Decision Tree può sovradattarsi ai dati di addestramento, specialmente se la profondità dell'albero non viene limitata. Per mitigare questo rischio, in questo studio è stata fissata una profondità massima di 10.

#### 2. **Prestazioni inferiori rispetto a modelli ensemble**

I Decision Tree tendono ad avere prestazioni inferiori rispetto a metodi ensemble, come Random Forest o Gradient Boosting, che sfruttano più alberi per migliorare accuratezza e robustezza.

## K-Nearest Neighbors (KNN)

Come secondo algoritmo è stato scelto KNN (K-Nearest Neighbors) che consiste in un metodo di classificazione e regressione non parametrico, il che significa che non assume una distribuzione specifica dei dati, ma basa le sue previsioni sulla somiglianza tra i punti nello spazio delle caratteristiche. Il principio alla base di KNN è che un dato sconosciuto viene classificato in base ai suoi K vicini più prossimi nel set di addestramento, utilizzando metriche di distanza come la distanza euclidea, Manhattan o coseno.

Una volta selezionato il valore di K, l'algoritmo calcola la distanza tra il nuovo punto e tutti i punti noti, individuando i K più vicini. Se il problema è di classificazione, la classe assegnata è quella più frequente tra i vicini, mentre nel caso della regressione, il valore predetto è la media o la mediana dei valori dei vicini. Questo approccio rende KNN semplice ed efficace, soprattutto quando le classi sono ben separate e i dati sono ben bilanciati.

La decisione di utilizzare KNN è stata guidata principalmente dai seguenti fattori:

#### 1. **Non parametricità:**

Il KNN non richiede ipotesi sulla distribuzione dei dati, risultando efficace per dataset complessi e altamente variabili come MURA.

#### 2. **Sensibilità del KNN alla struttura dei dati ad alta dimensionalità:**

Il KNN si adatta bene ai dati ad alta dimensionalità del dataset MURA, in cui le immagini mediche preprocessate vengono rappresentate come vettori di migliaia di feature. Grazie al calcolo delle distanze euclideanee, riesce a riconoscere pattern locali,

come bordi e anomalie, associandoli correttamente alle classi.

**3. Capacità del KNN di sfruttare dataset bilanciati e sottocampionati:**

Il KNN si dimostra efficace su dataset bilanciati come MURA, sfruttando la distribuzione uniforme delle classi per migliorare l'accuratezza ed evitare il predominio di una classe. Grazie alla normalizzazione e al ridimensionamento dei dati, classifica le immagini senza necessitare di modelli complessi. Inoltre, l'uso di un dataset compatto riduce i tempi computazionali, rendendolo una scelta pratica rispetto a metodi più sofisticati.

**4. Robustezza alle variazioni locali:**

Il KNN è robusto alle variazioni locali poiché basa la classificazione sui vicini più prossimi, limitando l'influenza di outlier e anomalie. Considerando più vicini, un singolo dato anomalo incide poco sulla decisione finale, rendendolo meno sensibile al rumore e alle variazioni nei dati. Nel dataset MURA, questo aiuta a gestire artefatti o variazioni di illuminazione nelle immagini mediche, migliorando l'affidabilità delle predizioni.

Tuttavia, è importante evidenziare alcuni svantaggi relativi all'uso del KNN:

1. **Elevata complessità computazionale:** L'algoritmo richiede il calcolo delle distanze con tutti i campioni del dataset, risultando poco efficiente per dataset di grandi dimensioni, anche con sottocampionatura.
2. **Sensibilità alla scelta della metrica:** La performance dipende dalla metrica di distanza utilizzata (nel caso specifico, la distanza euclidea) e dalla scala delle feature.
3. **Limitata interpretabilità:** Il KNN non fornisce spiegazioni dirette per le sue predizioni, rappresentando un limite nel contesto medico, dove la spiegabilità è fondamentale.

## **Support Vector Machine (SVM)**

Come ultimo algoritmo è stato scelto di utilizzare il Support Vector Machine (SVM), un algoritmo di classificazione noto per la sua efficacia nel gestire dati non lineari grazie all'uso di kernel appropriati. Il dataset MURA presenta una sfida particolare in quanto le immagini grezze sono importate dal dataset in uno spazio ad alta dimensionalità (64x64 pixel per immagine, ovvero 4096 caratteristiche per ogni esempio).

L'SVM (Support Vector Machine) è un algoritmo di classificazione che si basa sulla ricerca del margine di separazione massimo tra le classi. L'idea alla base è quella di trovare un iperpiano ottimale che separi i punti appartenenti a classi diverse, massimizzando la distanza tra i punti più vicini delle due classi (detti vettori di supporto) e il margine stesso. Se i dati non sono linearmente separabili, l'SVM utilizza una trasformazione non lineare attraverso i kernel, che proiettano i dati in uno spazio di dimensione superiore dove la separazione diventa possibile. Il kernel RBF (Radial Basis Function) è particolarmente utile per dati complessi come le immagini radiografiche, poiché permette di modellare separazioni non lineari in modo più efficace rispetto a un semplice iperpiano lineare.

Inizialmente, il modello SVM è stato addestrato sui dati grezzi, senza alcuna riduzione dimensionale. In questa fase, i dati delle immagini sono stati "appiattiti" (flattened) in vettori unidimensionali di dimensione 4096 (64x64), e successivamente il modello è stato addestrato utilizzando il kernel radiale (RBF) con un parametro di regolarizzazione C elevato, pari a 10.000, per ottimizzare la separazione tra le classi presenti nelle immagini.

Successivamente, per consentire un'approssimativa rappresentazione grafica, è stato riaddestrato il modello applicando però una riduzione dimensionale tramite la PCA (Principal Component Analysis). Questa è stata applicata sia sui dati di addestramento che su quelli di validazione per ridurre la dimensionalità a due componenti principali. Questo approccio ha permesso non solo di ridurre la complessità del modello e di accelerare i tempi di addestramento, ma anche di ottenere una rappresentazione 2D utile per visualizzare graficamente i risultati, come i vettori di supporto e i margini di separazione.

La decisione di utilizzare l'SVM è stata guidata principalmente dai seguenti fattori:

**1. Efficienza su dataset bilanciati e binari:**

Il dataset MURA è stato trattato come un problema di classificazione binaria, con etichette corrispondenti a "presenza" o "assenza" di anomalie nelle radiografie. In questo contesto, l'SVM si è rivelato un'opzione ideale grazie alla sua capacità di creare un margine di separazione massimo tra le due classi. L'SVM è stato addestrato su un **subset bilanciato** del dataset, con il campionamento equo di immagini appartenenti a ciascuna classe per evitare il problema del bilanciamento dei dati.

**2. Efficacia del kernel RBF (Radial Basis Function):**

L'SVM è stato configurato con il kernel RBF, che consente di gestire separazioni non lineari tra le classi. Questo è particolarmente utile per dati complessi come le immagini radiografiche, dove la separazione tra le classi potrebbe non essere semplice e lineare. Il kernel RBF è stato scelto per la sua capacità di trasformare lo spazio originale delle feature in uno spazio ad alta dimensione, permettendo una separazione più precisa.

Sono presenti, inoltre, anche svantaggi legati all'uso di questo modello come:

**1. Sensibilità ai Parametri:**

L'accuratezza dell'SVM è fortemente influenzata dalla scelta dei parametri, come il valore di C e il parametro gamma del kernel RBF. La corretta ottimizzazione di questi parametri è essenziale per ottenere buone prestazioni, ma può richiedere una ricerca su griglia o metodi di ottimizzazione più complessi, aggiungendo un ulteriore livello di complessità al processo.

**2. Perdita di informazioni usando la PCA:**

La riduzione dimensionale tramite PCA implica una perdita di informazioni. Le componenti meno significative (che vengono scartate) potrebbero contenere informazioni utili per la classificazione, riducendo la capacità del modello di distinguere tra classi.

**3. Grafico non sempre rappresentativo della complessità del problema:**

La rappresentazione grafica a 2D, pur essendo utile per l'interpretazione, non riesce

a catturare completamente la complessità della separazione tra le classi nel **spazio originale ad alta dimensione**. Pertanto, pur essendo efficace per visualizzare la separazione nel caso di un modello a bassa dimensione, potrebbe non riflettere esattamente il comportamento del modello nello spazio di alta dimensione.

## Scelta della reti neurali

Terminati gli algoritmi, passiamo ora alle reti neurali che verranno affrontate nello stesso modo

### Multi Layer Perceptron (MLP)

Come prima rete neurale è stata scelta la MLP (Multilayer Perceptron) che è un modello appartenente alla categoria delle reti feedforward, in cui i dati fluiscono in una sola direzione, dall'input all'output, senza cicli o connessioni ricorrenti. Questo tipo di rete è ampiamente utilizzato in ambiti come la classificazione, la regressione e il riconoscimento di pattern grazie alla sua capacità di apprendere relazioni complesse tra i dati.

Un MLP è composto da tre strati principali. Lo strato di input riceve i dati grezzi e rappresenta ciascuna variabile o caratteristica del problema. Gli strati nascosti elaborano le informazioni applicando trasformazioni non lineari tramite funzioni di attivazione, permettendo alla rete di modellare relazioni complesse nei dati. Infine, lo strato di output produce il risultato finale, come una classe assegnata nel caso della classificazione o un valore numerico nella regressione.

Il processo di apprendimento dell'MLP avviene in due fasi. Durante l'addestramento, la rete utilizza l'algoritmo di retropropagazione per ottimizzare i pesi delle connessioni tra i neuroni, minimizzando l'errore tra le previsioni e i valori reali attraverso un aggiornamento iterativo. Una volta completato l'addestramento, il modello è in grado di eseguire inferenze su nuovi dati, applicando le trasformazioni apprese per generare previsioni accurate.

Esistono diverse motivazioni della scelta di questa rete:

1. **Adattabilità a immagini di bassa risoluzione:** La rete MLP è in grado di lavorare efficacemente con immagini di bassa risoluzione (64x64 pixel), come quelle importate dal dataset MURA. Ridimensionare le immagini in questo formato riduce i tempi di elaborazione, mantenendo comunque un buon livello di prestazione nel riconoscimento delle caratteristiche rilevanti, senza richiedere risorse computazionali eccessive.
2. **Robustezza nella classificazione binaria:** Essendo il dataset MURA un problema di classificazione binaria (malattia vs. non malattia), l'MLP si adatta bene a questa tipologia di task. La rete neurale, grazie alla sua struttura multilivello, è in grado di catturare relazioni complesse tra le caratteristiche delle immagini, facilitando una separazione chiara tra le due classi.
3. **Capacità di generalizzare sui dati di validazione:** Nonostante le immagini nel dataset MURA possano variare notevolmente tra i vari esami, la rete MLP è capace di generalizzare grazie alla sua capacità di apprendere rappresentazioni astratte

attraverso i suoi strati nascosti. Questo le permette di adattarsi a variazioni locali nelle immagini senza perdere precisione.

Esistono però anche alcuni svantaggi nell'utilizzo della rete MLP per il dataset MURA:

1. **Limitata capacità di estrarre caratteristiche spaziali:** L'MLP non è progettato per sfruttare le caratteristiche spaziali delle immagini, come le convoluzioni in una rete CNN (Convolutional Neural Network). Nel caso del dataset MURA, dove le immagini possono contenere dettagli localizzati importanti (come anomalie nelle radiografie), la rete MLP potrebbe non essere altrettanto efficiente nell'identificare queste caratteristiche rispetto a modelli specializzati come le CNN.
2. **Overfitting con dataset di piccole dimensioni:** Anche se l'uso di un subset bilanciato aiuta a mitigare l'overfitting, l'MLP può comunque essere soggetto a questo problema, soprattutto con un numero limitato di esempi di addestramento. Il modello potrebbe adattarsi troppo ai dati di training, perdendo la capacità di generalizzare correttamente sui dati di validazione.
3. **Alto costo computazionale durante l'addestramento:** L'addestramento di una rete MLP con molteplici strati nascosti e un grande numero di neuroni richiede una notevole quantità di tempo e risorse computazionali. Per il dataset MURA, che può includere migliaia di immagini, ciò potrebbe comportare tempi di addestramento lunghi, specialmente se si utilizzano risorse hardware limitate.

## Passaggio alle reti neurali pre-addestrate

Notando le basse prestazioni ottenute con un Multi-Layer Perceptron (MLP) utilizzando reti neurali non pre-addestrate, si è deciso di passare ad un approccio basato sullo sfruttamento di reti pre-addestrate. In particolare, sono state selezionate DenseNet169, VGG-16 e ResNet50 per la classificazione del dataset MURA.

Dopo aver eseguito una serie di test comparativi su queste architetture, valutando metriche quali accuratezza, F1-score e Recall, sono stati identificati i due modelli con le migliori prestazioni.

Modello	Accuracy	Precisione (0)	Precisione (1)	Recall (0)	Recall (1)	F1 - score (0)	F1- score (1)
DenseNet169	67.31%	70%	65%	65%	70%	67%	67%
VGG-16	64.59%	71%	60%	55%	75%	62%	67%
ResNet50	56.08%	58%	54%	54%	58%	56%	56%

DenseNet169 e VGG-16 sono stati quindi successivamente analizzati in modo più approfondito per comprendere meglio il loro comportamento sulla classificazione delle immagini radiografiche e ottimizzarne ulteriormente le prestazioni.

## Visual Geometry Group 16-layer CNN (VGG-16)

La rete VGG-16 è stata scelta per il dataset MURA grazie alla sua architettura semplice ma efficace, basata su una profondità consistente e su strati convoluzionali di piccole dimensioni (3x3). Questa struttura consente di apprendere caratteristiche gerarchiche, catturando pattern locali nelle immagini mediche e permettendo una rappresentazione robusta delle anomalie diagnostiche. L'architettura di VGG-16 è composta da blocchi di convoluzione seguiti da livelli di pooling, che riducono progressivamente la dimensione spaziale mantenendo le informazioni più rilevanti, facilitando così la classificazione.

VGG-16 si distingue per la sua capacità di estrazione delle caratteristiche, risultando particolarmente utile per dataset come MURA, in cui è essenziale individuare strutture e anomalie ossee in immagini a toni di grigio. L'uso di pesi pre-addestrati su ImageNet favorisce il trasferimento di conoscenze, migliorando la capacità della rete di riconoscere pattern generali utili anche nel contesto medico, riducendo i tempi di addestramento e migliorando le prestazioni del modello.

Tra i suoi punti di forza troviamo:

- **Struttura profonda e gerarchica:** Consente l'apprendimento di rappresentazioni multi-livello, essenziali per catturare dettagli locali e globali nelle immagini mediche.
- **Facilità di implementazione:** La sua architettura sequenziale rende VGG-16 intuitiva da ottimizzare e integrare in pipeline di deep learning.
- **Robustezza grazie al pre-addestramento:** Il modello beneficia dei pesi appresi su ImageNet, migliorando la generalizzazione su dataset più specifici come MURA.

Tuttavia, rispetto a modelli più recenti come DenseNet169, VGG-16 presenta alcune limitazioni. La sua architettura, pur essendo efficace, è meno efficiente nell'uso dei parametri, comportando un elevato numero di pesi e richiedendo quindi una maggiore capacità computazionale e memoria. Inoltre, l'assenza di connessioni dense tra i layer può limitare il flusso delle informazioni, rendendo l'addestramento meno efficiente rispetto a reti con strutture più avanzate.

Nonostante questi svantaggi, VGG-16 rimane una scelta solida per il dataset MURA, garantendo una buona capacità di generalizzazione e una struttura facilmente adattabile a immagini di input a bassa risoluzione (64x64). In studi futuri, verrà valutato l'impatto di un incremento della risoluzione delle immagini a 128x128 per verificare se una maggiore quantità di dettagli visivi possa migliorare ulteriormente la capacità del modello di rilevare anomalie ossee con maggiore

## Densely Connected CN 169-layer (DenseNet169)

La rete DenseNet169 è stata scelta per il dataset MURA grazie alla sua architettura innovativa, in cui ogni layer è connesso a tutti i successivi. Questa struttura favorisce un flusso continuo di informazioni e gradienti attraverso la rete, riducendo il rischio di vanishing gradient, ovvero quando i gradienti diventano troppo piccoli, bloccando l'apprendimento negli strati profondi di una rete neurale, e migliorando la capacità di apprendimento. DenseNet169 si distingue per la sua abilità nell'estrazione di caratteristiche rilevanti, riuscendo a catturare dettagli fini e strutture complesse, essenziali per analizzare immagini mediche come quelle

del dataset MURA. Inoltre, utilizza i parametri in modo efficiente, limitando il rischio di overfitting e garantendo prestazioni elevate anche in contesti complessi.

Tra i suoi punti di forza troviamo:

1. **Capacità di estrazione di caratteristiche:** L'architettura densa cattura dettagli fini e strutture complesse, fondamentali per la diagnosi medica.
2. **Efficienza nell'uso dei parametri:** Utilizza i parametri in modo più efficace rispetto ad altre reti comparabili, riducendo il rischio di overfitting.

Il pre-addestramento, effettuato su ImageNet, permette a DenseNet169 di trasferire conoscenze generali apprese su un dataset di immagini, facilitando il riconoscimento di caratteristiche utili per il task specifico. Questo migliora la performance e riduce i tempi di addestramento su MURA, poiché il modello parte già da una base solida.

Successivamente, condurremo un'analisi sperimentale, per entrambi le reti neurali utilizzate e già addestrate, aumentando la dimensione delle immagini a 128x128, al fine di valutare se un incremento della risoluzione possa migliorare la capacità del modello di estrarre caratteristiche rilevanti e affinare la diagnosi. Questo studio ci permetterà di comprendere se una risoluzione maggiore possa offrire un vantaggio in termini di accuratezza e generalizzazione rispetto alla configurazione attuale (64x64).

Tra gli svantaggi di DenseNet169 è presente la richiesta di maggiori risorse computazionali per il fine-tuning, ovvero il processo di adattamento di un modello pre-addestrato su un nuovo dataset, ottimizzandone i pesi per un compito specifico. Questa esigenza deriva dalla sua maggiore profondità e complessità architettonica. Inoltre, il preprocessing delle immagini, che include il ridimensionamento e la conversione a tre canali, può comportare una lieve perdita di informazioni.

## Risultati ottenuti

Osserviamo ora i risultati prodotti dalle applicazioni dei vari modelli, analizzandoli e confrontandoli tra di loro

## Algoritmi - Analisi comparativa dei risultati

Per valutare le prestazioni dei modelli SVM, KNN e Decision Tree applicati al dataset MURA (MSK X-rays), sono state utilizzate diverse metriche di valutazione, tra cui l'**accuracy**, la **precision**, il **recall** e il **F1-score** per entrambe le classi (0 e 1). Le prestazioni dei modelli sono riportate nella Tabella, che evidenzia i valori di accuracy, precisione, recall e F1-score per ogni modello e classe.

La tabella riporta i risultati dei tre algoritmi

Modello	Accuracy	Precisione (0)	Precisione (1)	Recall (0)	Recall (1)	F1- score (0)	F1- score (1)

<b>Decision Tree</b>	57.21%	59%	55%	56%	58%	58%	57%
<b>KNN</b>	57.08%	59%	55%	59%	55%	59%	55%
<b>SVM</b>	56.15%	58%	54%	60%	52%	59%	53%

#### - **Decision Tree (Miglior accuracy: 57.21%)**

Il Decision Tree ha ottenuto la migliore accuracy (57.21%) tra i modelli testati, con una precisione bilanciata per entrambe le classi. Tuttavia, il modello presenta alcune limitazioni: il recall per la classe 0 (56%) è inferiore rispetto agli altri modelli, suggerendo che il modello ha più difficoltà a identificare correttamente gli esempi della classe negativa.

#### - **KNN (Accuracy: 57.08%)**

Il KNN ha ottenuto un'accuracy di 57.08%, con performance complessive simili a quelle del Decision Tree. La precisione per entrambe le classi è bilanciata, ma la classe 1 ha mostrato un leggero vantaggio nella precisione rispetto alla classe 0. Un aspetto critico del KNN è la scelta del parametro k, che, se non ottimizzato, può influire significativamente sulle performance. Inoltre, il KNN è computazionalmente più costoso rispetto al Decision Tree, soprattutto per dataset di grandi dimensioni.

#### - **SVM (Accuracy: 56.15%)**

Il SVM ha ottenuto l'accuracy più bassa tra i tre modelli, pari al 56.15%. Sebbene il SVM abbia una precisione più alta per la classe 0 rispetto al KNN e al Decision Tree, il suo recall per la classe 1 (52%) è significativamente inferiore, suggerendo che il modello ha più difficoltà a classificare correttamente gli esempi della classe positiva. Inoltre, il SVM può risultare computazionalmente più costoso rispetto agli altri modelli, specialmente in scenari con grandi quantità di dati.

I tre modelli (SVM, KNN e Decision Tree) presentano vantaggi e svantaggi a seconda delle specifiche esigenze di applicazione.

### Miglior modello per performance complessive

1. Il **Decision Tree** ha ottenuto il miglior valore di accuracy (57.21%), ma presenta un recall più basso per la classe 0, indicando che potrebbe confondere esempi negativi con positivi.
2. Il **KNN** ha mostrato performance simili a quelle del Decision Tree, ma può essere migliorato con una selezione più accurata del parametro k.
3. Il **SVM** ha ottenuto l'accuracy più bassa (56.15%), ma potrebbe trarre beneficio da un'ottimizzazione più approfondita dei parametri del kernel.

### Scelta del modello in base alle esigenze

- a. **Decision Tree** è la scelta ideale se si desidera un modello interpretabile e veloce.
- b. **KNN** può essere una buona alternativa, soprattutto se il tuning del parametro k è ottimizzato.
- c. **SVM** potrebbe essere preferito in scenari con risorse computazionali elevate, ma solo se adeguatamente ottimizzato tramite il kernel e la normalizzazione delle feature.

## Reti neurali - Analisi comparativa dei risultati

Per valutare le prestazioni delle reti neurali VGG-16, DenseNet169 e MLP applicate al dataset MURA (MSK X-rays), sono state impiegate le stesse metriche utilizzate per gli algoritmi

I risultati delle tre reti sono riportati nella tabella seguente:

Modello	Accuracy	Precisione (0)	Precisione (1)	Recall (0)	Recall (1)	F1 - score (0)	F1- score (1)
DenseNet169	67.31%	70%	65%	65%	70%	67%	67%
VGG-16	64.59%	71%	60%	55%	75%	62%	67%
MLP	55.74%	60%	53%	45%	68%	51%	59%

### - MLP (Accuracy: 55.74%)

MLP si colloca in una posizione intermedia, con un'accuracy di 55.74%. Mostra una buona precisione per la classe 0 (60%) e il miglior recall per la classe 1 (68%), suggerendo una maggiore capacità di identificare esempi positivi rispetto agli altri modelli. Tuttavia, il suo recall per la classe 0 è relativamente basso (45%), indicando che una parte significativa degli esempi negativi viene erroneamente classificata come positiva.

### - VGG-16 (Accuracy: 64.59%)

VGG-16 ha ottenuto un'accuracy del 64.59%, posizionandosi tra DenseNet169 e MLP. Questo modello si distingue per la sua elevata precisione nella classe 0 (71%), indicando una buona capacità di classificare correttamente gli esempi negativi. Tuttavia, il recall per la classe 0 è piuttosto basso (55%), il che significa che molti esempi negativi vengono erroneamente classificati come positivi. D'altra parte, il recall per la classe 1 è il più alto tra i modelli testati (75%), suggerendo un'ottima capacità di identificare gli esempi positivi. Complessivamente, i valori di F1-score indicano che VGG-16 rappresenta un buon compromesso tra precisione e recall, con un leggero sbilanciamento verso la classe positiva.

### - DenseNet169 (Miglior accuracy: 67.94%)

DenseNet169 ha ottenuto la migliore accuracy (67.94%) tra i modelli testati, grazie a un buon bilanciamento tra precisione e recall. I suoi valori di F1-score sono i più elevati, suggerendo una capacità predittiva superiore rispetto agli altri modelli. Tuttavia, il recall per la classe 1 (64%) è inferiore rispetto a quello per la classe 0 (71%), indicando una leggera difficoltà nel riconoscere gli esempi positivi.

Si può dire quindi che DenseNet169 si configura come il modello più efficace per la classificazione delle immagini MURA, grazie alla sua accuracy più elevata e al buon equilibrio tra precisione e recall. Tuttavia, il costo computazionale deve essere considerato prima dell'implementazione su larga scala. VGG-16, invece, offre un buon compromesso tra precisione e recall, con una particolare capacità di riconoscere gli esempi positivi grazie all'alto valore di recall per la classe 1. Risulta quindi una scelta valida nei contesti in cui l'identificazione dei positivi è prioritaria, pur richiedendo meno risorse rispetto a DenseNet169. Infine, MLP che rispetto agli altri 2 modelli ha prestazioni nettamente inferiori.

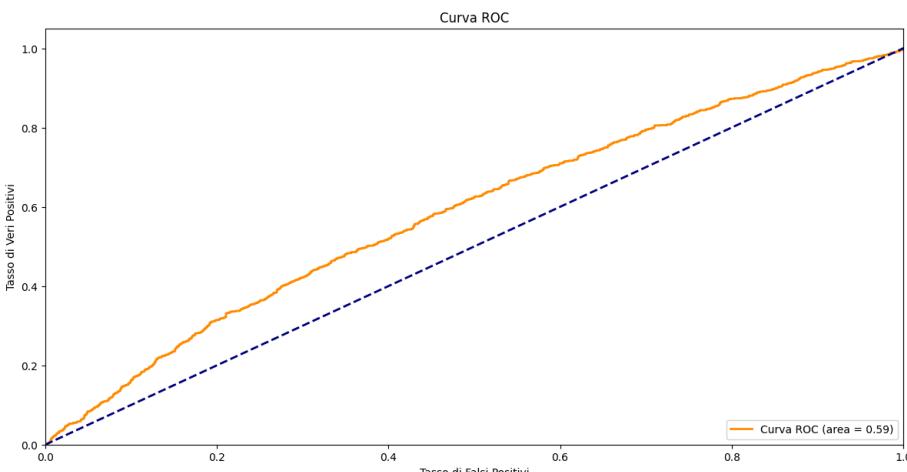
## Matrici di confusione, grafi e estrazione immagini

Passiamo ora all'analisi di nuovi dati relativi ai risultati esaminati finora, con l'obiettivo di approfondire ulteriormente la comprensione delle prestazioni dei vari modelli. Per facilitare la consultazione e garantire una presentazione chiara e strutturata, i dati sono stati organizzati in tabelle, rendendo l'interpretazione più intuitiva e agevole.

Nello specifico andremo a vedere per ogni modello:

- **Matrice di confusione:** tabella che riassume le prestazioni di un modello di classificazione, confrontando le predizioni con i valori reali ed è per calcolare metriche come precisione, recall e F1-score, fornendo un'analisi dettagliata degli errori del modello. Ogni riga rappresenta le istanze della classe reale, mentre ogni colonna indica le previsioni del modello. I principali elementi sono:
  - **True Positives (TP):** esempi correttamente classificati come positivi.
  - **True Negatives (TN):** esempi correttamente classificati come negativi.
  - **False Positives (FP):** esempi negativi erroneamente classificati come positivi.
  - **False Negatives (FN):** esempi positivi erroneamente classificati come negativi.
- **Grafico della curva ROC:** È un grafico che mostra la capacità di un modello di classificazione nel distinguere tra classi, tracciando il tasso di veri positivi (TPR, True Positive Rate) rispetto al tasso di falsi positivi (FPR, False Positive Rate) a vari livelli di soglia. Il grafico aiuta a valutare la performance del modello in modo complessivo, indipendentemente dalla scelta di una soglia specifica. L'asse X rappresenta il tasso di falsi positivi, mentre l'asse Y indica il tasso di veri positivi. L'area sotto la curva (AUC) è un indicatore di quanto il modello sia in grado di discriminare tra le classi, con un valore AUC più alto che indica una maggiore capacità discriminativa. Una curva più vicina all'angolo superiore sinistro riflette una migliore capacità predittiva del modello
- **Estrazione immagini con predizione e valore:** elenco di immagine dove ognuna mostra la 'Label' (etichetta reale) e la 'Pred' (predizione del modello). Le predizioni corrette coincidono con l'etichetta reale, mentre gli errori di classificazione evidenziano le limitazioni del modello. Questa rappresentazione aiuta a evidenziare visivamente la precisione e l'accuratezza del modello nelle varie classi.
- **Altro:** altro materiale possibile e variabile da modello a modello

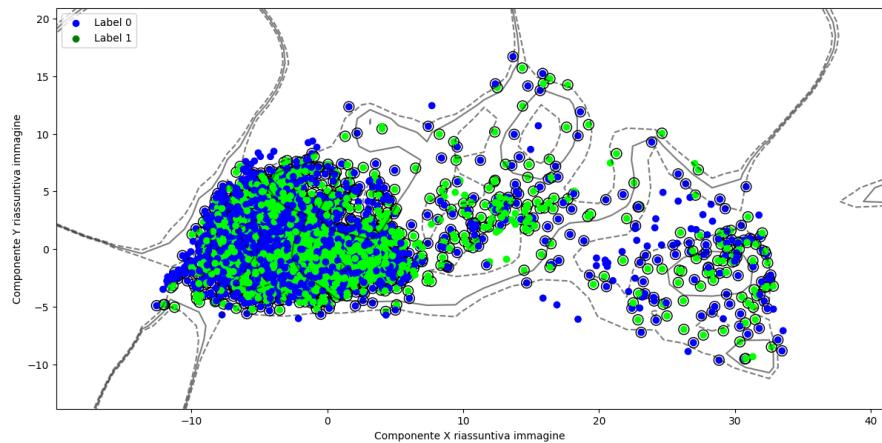
## SVM (Support Vector Machine)

CONFUSION MATRIX	ANALISI
[[988 679] [740 790]]	<ul style="list-style-type: none"><li>• <b>988</b> → (True Negatives, TN): Il modello ha correttamente classificato 988 immagini come appartenenti alla classe 0 (negativo).</li><li>• <b>679</b> → (False Positives, FP): Il modello ha classificato erroneamente 679 immagini della classe 0 come classe 1.</li><li>• <b>740</b> → (False Negatives, FN): Il modello ha classificato erroneamente 740 immagini della classe 1 come classe 0.</li><li>• <b>790</b> → Vero Positivi (True Positives, TP): Il modello ha correttamente classificato 790 immagini come appartenenti alla classe 1 (positivo).</li></ul> <p>La matrice di confusione mostra che il modello SVM ha un'accuratezza relativamente bassa (55.6%), il che indica che il modello sta classificando correttamente poco più della metà dei campioni.</p>
CURVA ROC	
 <p>The figure shows an ROC curve titled "Curva ROC". The vertical axis is labeled "Tasso di Veri Positivi" (True Positive Rate) and ranges from 0.0 to 1.0. The horizontal axis is labeled "Tasso di Falsi Positivi" (False Positive Rate) and ranges from 0.0 to 1.0. A dashed diagonal line represents a random classifier. An orange line represents the ROC curve for the SVM model, which starts at (0,0) and ends at (1,1), passing above the diagonal line. A legend indicates "Curva ROC (area = 0.59)".</p> <p>L'area sotto la curva (AUC) è pari a 0.59. In questo caso, il valore AUC suggerisce una performance modesta del modello nel distinguere tra le classi.</p>	

## ESTRAZIONE IMMAGINI CON PREDIZIONE E VALORE

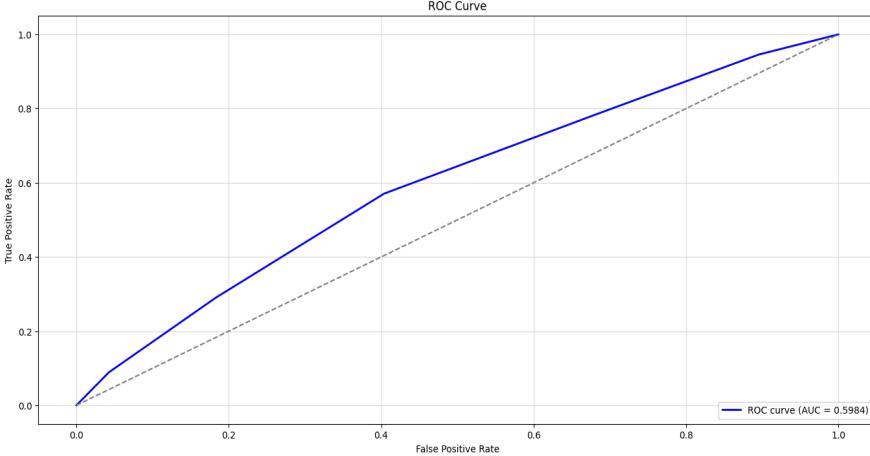
Label: 0 Pred: 0	Label: 0 Pred: 1	Label: 1 Pred: 1	Label: 0 Pred: 1	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 1 Pred: 0	Label: 0 Pred: 1	Label: 1 Pred: 1	Label: 1 Pred: 1
Label: 1 Pred: 1	Label: 1 Pred: 0	Label: 1 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 1 Pred: 0	Label: 1 Pred: 0	Label: 0 Pred: 0	Label: 1 Pred: 1

## RAPPRESENTAZIONE GRAFICA SVM



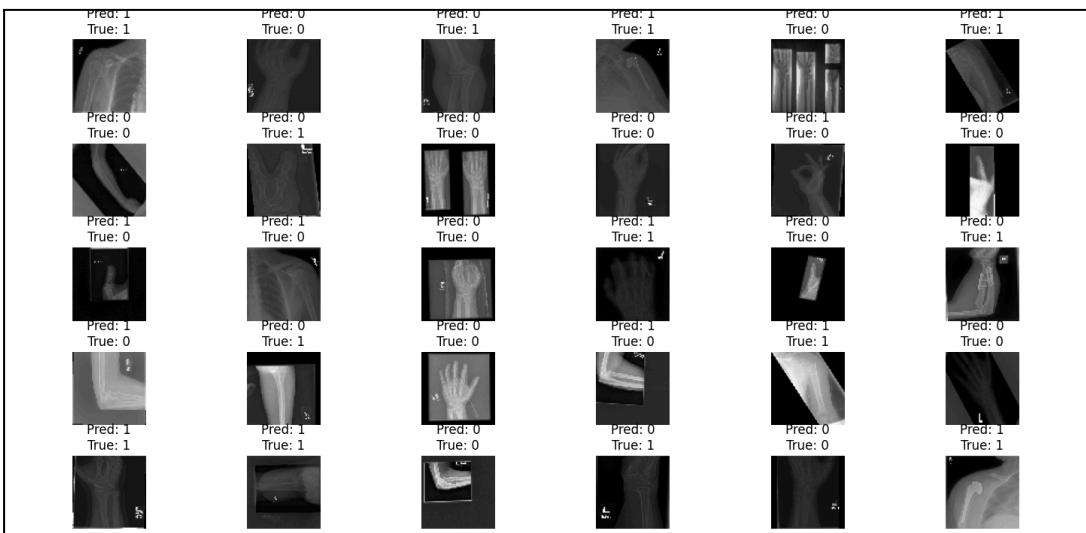
Rappresentazione grafica in 2D delle immagini ridotte dimensionalmente con PCA. I punti blu e verdi rappresentano rispettivamente le etichette reali 0 e 1, mentre i cerchi indicano i vettori di supporto utilizzati dal modello SVM per definire i margini di separazione tra le classi. Le linee tratteggiate mostrano i confini decisionali del modello. Questa visualizzazione evidenzia la capacità del modello di separare le due classi e identificare le regioni di incertezza.

## KNN (K-Nearest Neighbors)

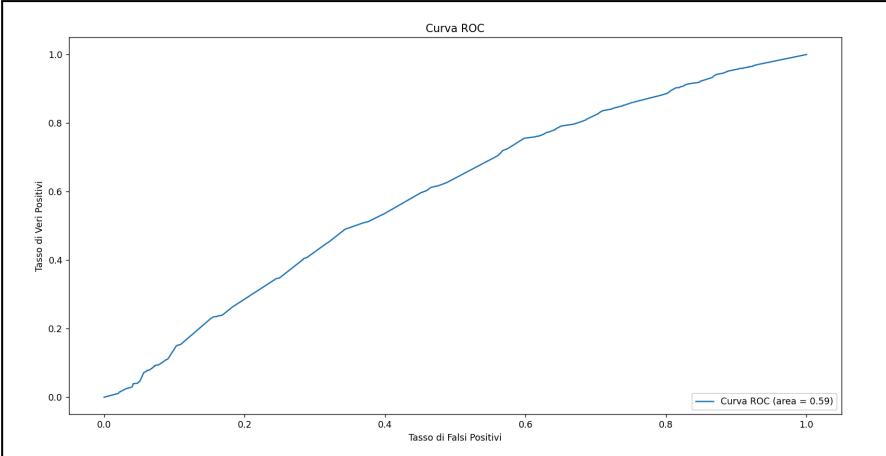
CONFUSION MATRIX	ANALISI
<p><b>Confusion Matrix:</b></p> <p><b>[[994 673]</b></p> <p><b>[657 873]]</b></p>	<ul style="list-style-type: none"><li><b>994 (TN)</b> → Il modello ha correttamente classificato 994 immagini come appartenenti alla classe 0 (negativo).</li><li><b>673 (FP)</b> → Il modello ha classificato erroneamente 673 immagini della classe 0 come classe 1.</li><li><b>657 (FN)</b> → Il modello ha classificato erroneamente 657 immagini della classe 1 come classe 0.</li><li><b>873 (TP)</b> → Il modello ha correttamente classificato 873 immagini come appartenenti alla classe 1 (positivo).</li></ul> <p>La matrice di confusione mostra che il modello KNN ha un'accuratezza relativamente bassa (58.4%) e tende a confondere le classi con un numero simile di FP e FN.</p>
<b>CURVA ROC</b>	
 <p>The figure shows an ROC Curve with the following characteristics:</p> <ul style="list-style-type: none"><li>The x-axis is labeled "False Positive Rate" and ranges from 0.0 to 1.0.</li><li>The y-axis is labeled "True Positive Rate" and ranges from 0.0 to 1.0.</li><li>A diagonal dashed line represents a random classifier (AUC = 0.5).</li><li>A solid blue line represents the ROC curve for the KNN model.</li><li>The area under the curve is labeled "ROC curve (AUC = 0.5984)".</li></ul>	

Il valore AUC di 0.5984 indica una capacità predittiva leggermente superiore al caso casuale (AUC = 0.5), suggerendo che il modello KNN fatica a distinguere efficacemente tra le classi.

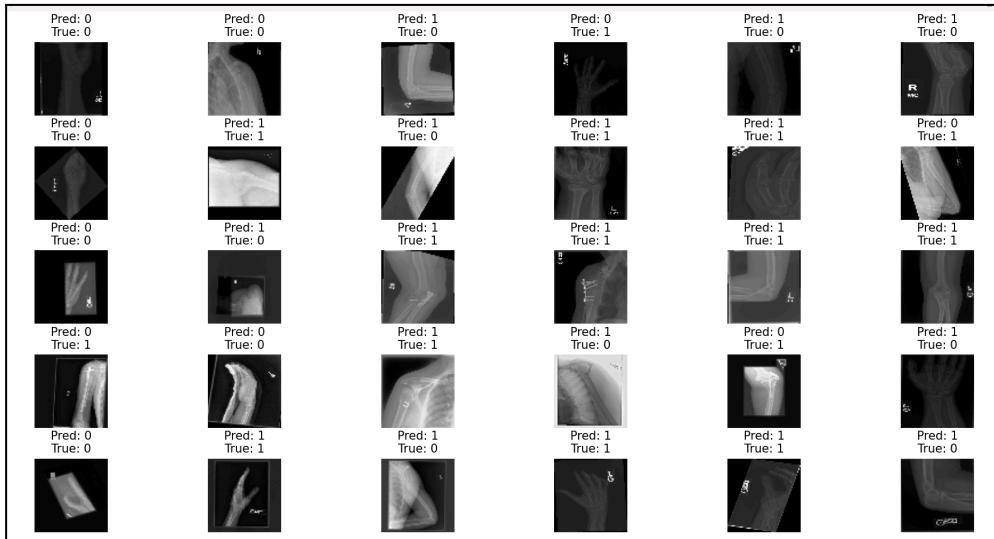
## ESTRAZIONE IMMAGINI CON PREDIZIONE E VALORE



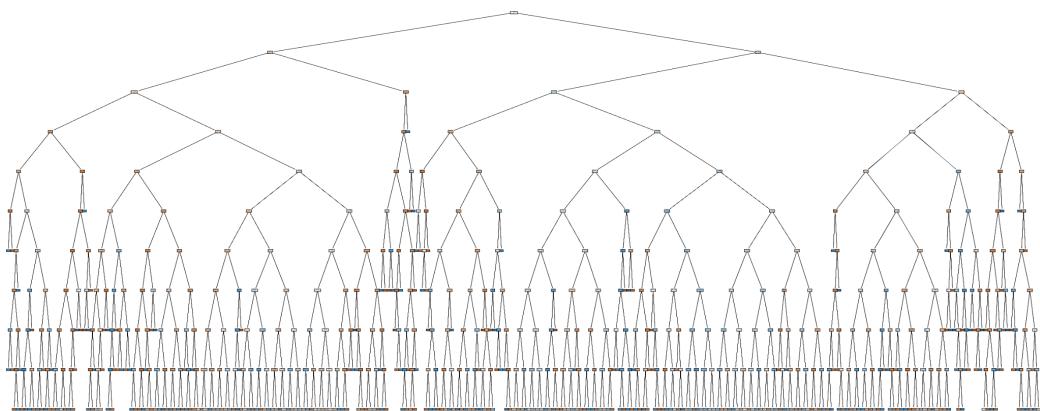
## DT (Decision Tree)

CONFUSION MATRIX	ANALISI
<p><b>Confusion Matrix:</b></p> <p><b>[[875 792]</b></p> <p><b>[559 971]]</b></p>	<ul style="list-style-type: none"><li>• <b>875 (TN)</b> → Il modello ha correttamente classificato 875 immagini come appartenenti alla classe 0 (negativo).</li><li>• <b>792 (FP)</b> → Il modello ha classificato erroneamente 792 immagini della classe 0 come classe 1.</li><li>• <b>559 (FN)</b> → Il modello ha classificato erroneamente 559 immagini della classe 1 come classe 0.</li><li>• <b>971 (TP)</b> → Il modello ha correttamente classificato 971 immagini come appartenenti alla classe 1 (positivo).</li></ul> <p>La matrice di confusione mostra che il modello Decision tree ha un accuracy del 57.7% e Sebbene il modello stia facendo alcune previsioni corrette, ci sono molte classificazioni errate, con un buon numero di False Positives e False Negatives.</p>
CURVA ROC	
 <p>L'area sotto la curva (AUC), pari a 0.59, evidenzia una capacità predittiva leggermente più alta del caso casuale.</p>	

## ESTRAZIONE IMMAGINI CON PREDIZIONE E VALORE

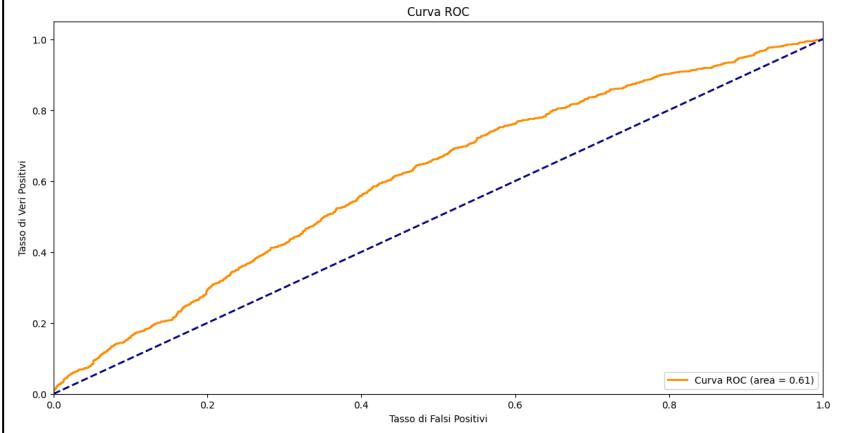


## RAPPRESENTAZIONE ALBERO



Questo Decision Tree rappresenta un modello di machine learning che suddivide i dati attraverso condizioni gerarchiche sulle feature, visualizzate nei nodi decisionali. Ogni nodo interno indica una regola basata su una feature specifica, mentre i rami rappresentano i risultati delle condizioni. I nodi foglia, colorati, indicano le classi finali o la probabilità delle classi, riflettendo la distribuzione dei dati.

## MLP (Multi Layer Perceptron)

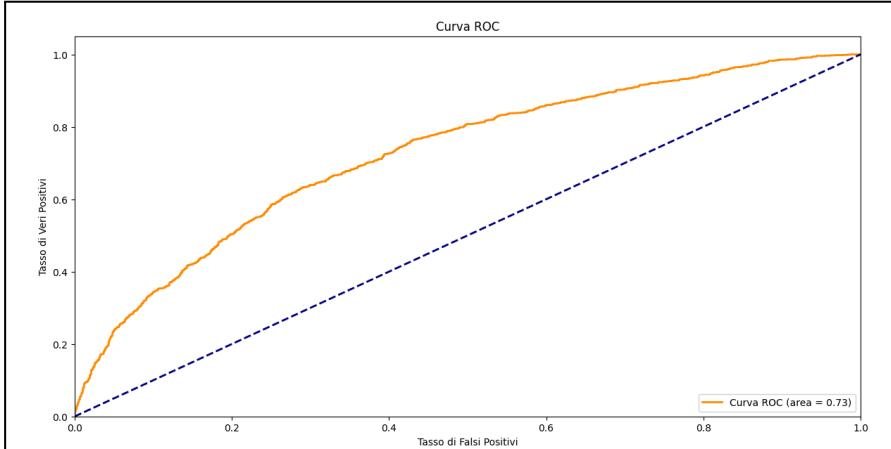
CONFUSION MATRIX	ANALISI
<p><b>Confusion Matrix:</b></p> <p><b>[[1035 632]</b></p> <p><b>[ 721 809]]</b></p>	<ul style="list-style-type: none"><li>• <b>1035 (TN)</b> → Il modello ha correttamente classificato 1035 immagini come appartenenti alla classe 0 (negativo).</li><li>• <b>632 (FP)</b> → Il modello ha classificato erroneamente 632 immagini della classe 0 come classe 1.</li><li>• <b>721 (FN)</b> → Il modello ha classificato erroneamente 721 immagini della classe 1 come classe 0.</li><li>• <b>809 (TP)</b> → Il modello ha correttamente classificato 809 immagini come appartenenti alla classe 1 (positivo).</li></ul> <p>La matrice di confusione mostra che la rete neurale ha un accuracy del 57.6% e Sebbene il modello stia facendo alcune previsioni corrette, ci sono molte classificazioni errate, con un buon numero di False Positives e False Negatives.</p>
<b>CURVA ROC</b>	
 <p>The figure shows an ROC curve titled "Curva ROC". The x-axis is labeled "Tasso di Falsi Positivi" (False Positive Rate) and ranges from 0.0 to 1.0. The y-axis is labeled "Tasso di Veri Positivi" (True Positive Rate) and ranges from 0.0 to 1.0. A dashed blue diagonal line represents a random classifier. An orange solid line represents the ROC curve for the model, starting at (0,0) and ending at (1,1). The area under the curve is indicated as 0.61. The curve is relatively flat and stays close to the diagonal line, indicating low discrimination power.</p>	

Un AUC di 0.61 suggerisce che il modello ha una capacità di discriminazione tra le classi leggermente basso.

## ESTRAZIONE IMMAGINI CON PREDIZIONE E VALORE

Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 1 Pred: 1	Label: 0 Pred: 1	Label: 1 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 1 Pred: 0	Label: 1 Pred: 0	Label: 0 Pred: 1
									
Label: 0 Pred: 1	Label: 0 Pred: 0	Label: 0 Pred: 1	Label: 1 Pred: 1	Label: 0 Pred: 0	Label: 1 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 1 Pred: 0	Label: 1 Pred: 0
									
Label: 1 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 1	Label: 0 Pred: 1	Label: 1 Pred: 0	Label: 1 Pred: 1	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 1 Pred: 1
									

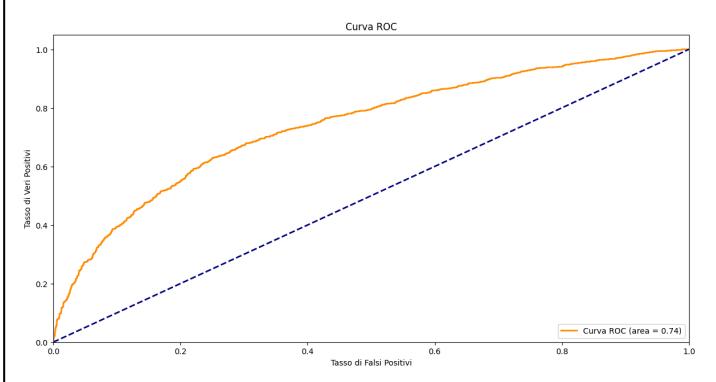
## VGG-16 (Visual Geometry Group 16-layer CNN)

CONFUSION MATRIX	ANALISI
<b>Confusion Matrix:</b> $\begin{bmatrix} 912 & 755 \\ 377 & 1153 \end{bmatrix}$	<ul style="list-style-type: none"> <li><b>912 (TN)</b> → Il modello ha correttamente classificato 912 immagini come appartenenti alla classe 0 (negativo).</li> <li><b>755 (FP)</b> → Il modello ha classificato erroneamente 755 immagini della classe 0 come classe 1.</li> <li><b>377 (FN)</b> → Il modello ha classificato erroneamente 377 immagini della classe 1 come classe 0.</li> <li><b>1153 (TP)</b> → Il modello ha correttamente classificato 1153 immagini come appartenenti alla classe 1 (positivo).</li> </ul> <p>La matrice di confusione mostra che la rete neurale ha un accuracy del 64.59%, questo indica che il modello è abbastanza preciso nel classificare correttamente le immagini. Sebbene l'accuratezza totale sembri buona, la precisione della classe 1 non è particolarmente alta, il che implica che ci sono molte predizioni false positive.</p>
<b>CURVA ROC</b>	
 <p>Ha un'AUC pari a 0.73, indicando una discreta capacità di discriminare tra classi positive e negative. La curva si posiziona al di sopra della diagonale tratteggiata dimostrando che il modello offre prestazioni migliori rispetto al caso.</p>	

## ESTRAZIONE IMMAGINI CON PREDIZIONE E VALORE

Label: 1 Pred: 1	Label: 1 Pred: 1	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 1 Pred: 1	Label: 1 Pred: 1	Label: 0 Pred: 0	Label: 1 Pred: 1	Label: 0 Pred: 0	Label: 1 Pred: 0
									
									
Label: 0 Pred: 1	Label: 0 Pred: 0	Label: 0 Pred: 1	Label: 1 Pred: 1	Label: 1 Pred: 1	Label: 0 Pred: 1	Label: 1 Pred: 1	Label: 0 Pred: 0	Label: 1 Pred: 1	Label: 0 Pred: 1
									
Label: 0 Pred: 1	Label: 1 Pred: 1	Label: 1 Pred: 1	Label: 1 Pred: 0	Label: 1 Pred: 1	Label: 0 Pred: 1	Label: 1 Pred: 1	Label: 0 Pred: 0	Label: 1 Pred: 1	Label: 1 Pred: 1
									

## DenseNet169 (Densely Connected CN 169-layer)

CONFUSION MATRIX	ANALISI
<p><b>Confusion Matrix:</b></p> <p><b>[[1069 598]</b></p> <p><b>[ 433 1097]]</b></p>	<ul style="list-style-type: none"><li>• <b>1069 (TN)</b> → Il modello ha correttamente classificato 1069 immagini come appartenenti alla classe 0 (negativo).</li><li>• <b>598 (FP)</b> → Il modello ha classificato erroneamente 598 immagini della classe 0 come classe 1.</li><li>• <b>433 (FN)</b> → Il modello ha classificato erroneamente 433 immagini della classe 1 come classe 0.</li><li>• <b>1097 (TP)</b> → Il modello ha correttamente classificato 1097 immagini come appartenenti alla classe 1 (positivo).</li></ul> <p>La matrice di confusione mostra che la rete neurale ha un accuracy dell <b>67.7%</b>, questo indica che il modello è abbastanza preciso nel classificare correttamente le immagini, tuttavia ha ancora margini di miglioramento, specialmente per quanto riguarda la riduzione degli errori di classificazione (FP e FN).</p>
<b>CURVA ROC</b>	
	
<p>L'area sotto la curva (AUC = 0.74) evidenzia una buona performance di classificazione. La curva si posiziona al di sopra della diagonale tratteggiata della classificazione casuale, dimostrando che il modello offre prestazioni migliori rispetto al caso.</p>	

## ESTRAZIONE IMMAGINI CON PREDIZIONE E VALORE

Label: 1 Pred: 1	Label: 0 Pred: 0	Label: 1 Pred: 1	Label: 1 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 1	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 0
									
Label: 1 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 1	Label: 0 Pred: 0	Label: 1 Pred: 1	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 1 Pred: 1
									
Label: 1 Pred: 0	Label: 0 Pred: 1	Label: 1 Pred: 1	Label: 0 Pred: 0	Label: 1 Pred: 1	Label: 0 Pred: 0	Label: 0 Pred: 0	Label: 0 Pred: 1	Label: 1 Pred: 0	Label: 1 Pred: 1
									

## Discussione risultati e nuove soluzioni

Esaminiamo in modo più critico i risultati ottenuti, ampliando il contesto attraverso il confronto con studi precedenti. Successivamente, effettuando un'ablation study per ogni modello, approfondiamo nel dettaglio i fattori che influenzano l'efficienza, fornendo un'analisi ancora più completa e dettagliata.

### Confronto risultati con studi precedenti

Esistono migliaia di studi che confrontano l'utilizzo di algoritmi di machine learning in vari contesti applicativi. In particolare, per quanto riguarda il dataset **MURA**, è disponibile una classifica dei modelli più efficienti, consultabile al seguente link:

<https://stanfordmlgroup.github.io/competitions/mura/> e di cui riportiamo qui sotto un riassunto.

Posizione	Data	Modello	Kappa
1	30 Nov 2018	base-comb2-xuan-v3 (ensemble) jzhang Availink	0.843
5	08 Nov 2018	base-comb2-jun2 (ensemble)	0.814
10	19 Ago 2018	ianpan (ensemble) RIH 3D Lab	0.774
15	14 Lug 2018	Trs (single model) SCU_MILAB	0.763
20	21 Ago 2018	dn169-Aftrva (single) AliHealth	0.747
25	04 Ott 2018	ExtremityModel (ensemble)	0.734
30	11 Dic 2018	incev3_xy (single) UCB	0.716
35	10 Mar 2019	asa_model_nasnetmo (single) toyohashi	0.702
40	18 Gen 2019	first-attempt-kmle (ensemble) kmle	0.696
45	23 Dic 2018	{EnglebertDGC} (single model) UCLouvain	0.684
50	09 Lug 2018	- (dati non disponibili)	0.664
55	05 Nov 2018	DenseNet (single model) Rology	0.659

60	30 Giu 2018	Densenet121 (single model) Personal	0.629
65	26 Lug 2018	Bhaukali_v1.0 (single model) IIT BHU, Varanasi	0.581
70	6 Lug 2018	DenseNet (single model) Zhou	0.518

L'efficienza di questi modelli viene misurata utilizzando il coefficiente Kappa, un parametro di valutazione che si distingue leggermente dall'accuracy.

Il Kappa (Cohen's Kappa) è una misura statistica che quantifica l'accordo tra due valutatori (o un valutatore e un modello) correggendo per la possibilità che tale accordo sia avvenuto per caso. A differenza dell'accuracy, che indica semplicemente la percentuale di previsioni corrette rispetto al totale, il Kappa tiene conto della probabilità che le predizioni corrette siano avvenute per puro caso, ed è pertanto più affidabile, specialmente in situazioni con classi sbilanciate. Il valore del Kappa varia tra -1 e 1, dove:

- **1** indica un accordo perfetto,
- **0** indica un accordo pari a quello che ci si aspetterebbe per caso,
- valori negativi indicano un accordo peggiore di quanto ci si aspetterebbe per caso.

Pertanto, l'uso del Kappa permette di ottenere una valutazione più accurata e robusta delle prestazioni del modello, soprattutto quando le classi nel dataset non sono distribuite in modo equilibrato.

Per andare a collocare il nostro studio in questa classifica e comprendere quanto possa essere valido abbiamo calcolato il valore Kappa del modello DensNet169 addestrato con immagini 128x128, che è l'algoritmo tra tutti con accuracy, equilibrio tra precisione e recall per entrambe le classi maggiori. Il valore Kappa di 0.521 indica che il nostro studio ha ottenuto risultati accettabili, posizionandosi alla 69° posizione.

## Ablation Study

L'**Ablation Study** è una metodologia sperimentale utilizzata per analizzare l'impatto di specifici componenti di un modello di machine learning sulle sue prestazioni. Rimuovendo o modificando selettivamente alcune parti, è possibile comprendere quali elementi contribuiscono maggiormente all'accuratezza e alla generalizzazione del modello.

In questa parte dello studio, verranno testate diverse varianti dei modelli di riferimento e confrontate tra di loro.

## DT (Decision Tree)

Per il Decision Tree verranno valutate le variazioni della profondità massima dell'albero (ridotta a 5 e aumentata a 20) per comprendere il trade-off tra capacità predittiva e rischio di overfitting. Inoltre, verrà sostituito il criterio di suddivisione da Gini a Entropy, per esaminare come la misura dell'impurità influenzi la separabilità delle classi.

Configurazione	Accuracy	Precisione (0)	Precisione (1)	Recall (0)	Recall (1)	F1-score (0)	F1-score (1)
Decision Tree	56.74%	60%	54%	51%	63%	55%	58%
DT max_depth = 5	55.83%	60%	53%	48%	65%	53%	58%
DT max_depth = 20	56.08%	59%	54%	51%	61%	55%	57%
DT criterion = entropy	57.08%	62%	54%	45%	71%	52%	61%

Il modello Decision Tree nella sua configurazione originale (profondità massima 10, criterio di suddivisione Gini) ha ottenuto un'accuracy del 56.74%, con una leggera preferenza nella precisione per la classe 0 (60%) rispetto alla classe 1 (54%). Il recall mostra una discrepanza tra le due classi, con il modello che identifica meglio gli esempi della classe 1 (63%) rispetto alla classe 0 (51%), suggerendo una tendenza a classificare più esempi come appartenenti alla classe 1.

La riduzione della profondità massima a 5 ha portato a una leggera diminuzione dell'accuracy (-0.91%), con un calo del recall della classe 0 (48%) e un incremento del recall della classe 1 (65%). Questo indica che l'albero più superficiale ha meno capacità di modellare relazioni complesse, riducendo la capacità predittiva su alcuni esempi.

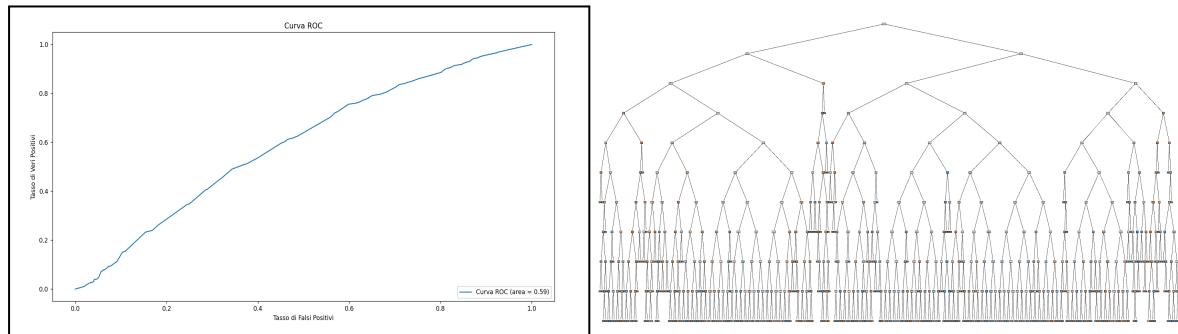
Aumentando la profondità a 20, le prestazioni rimangono pressoché invariate rispetto alla configurazione originale, suggerendo che una maggiore profondità non porta necessariamente a un miglioramento significativo, probabilmente perché il modello inizia a sovradattarsi senza un effettivo guadagno in generalizzazione.

Infine, la modifica del criterio di suddivisione da Gini a Entropy ha avuto un impatto più marcato, aumentando l'accuracy al 57.08% e migliorando il recall della classe 1 al 71%. Tuttavia, il recall della classe 0 è sceso al 45%, indicando che il modello ha acquisito una maggiore sensibilità verso la classe 1 a discapito della classe 0.

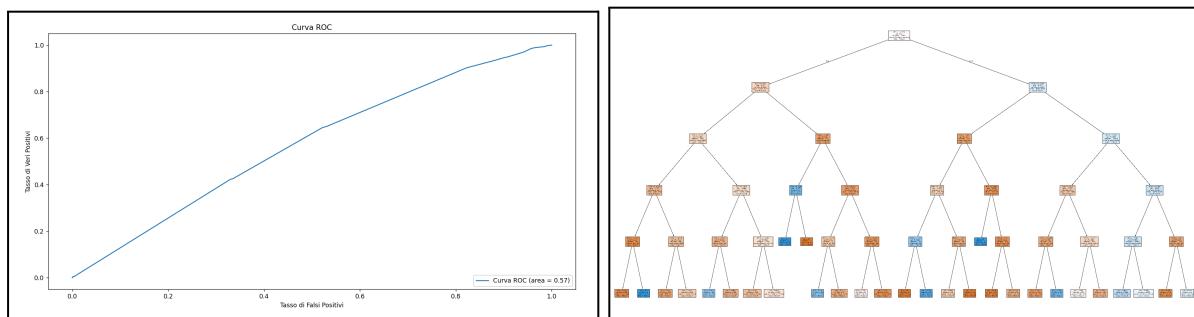
Nel complesso, i risultati suggeriscono che la profondità dell'albero ha un impatto limitato sulla performance, mentre il criterio di suddivisione può influenzare significativamente la

bilanciatura tra recall delle classi, rendendo il modello più o meno propenso a identificare una classe rispetto all'altra.

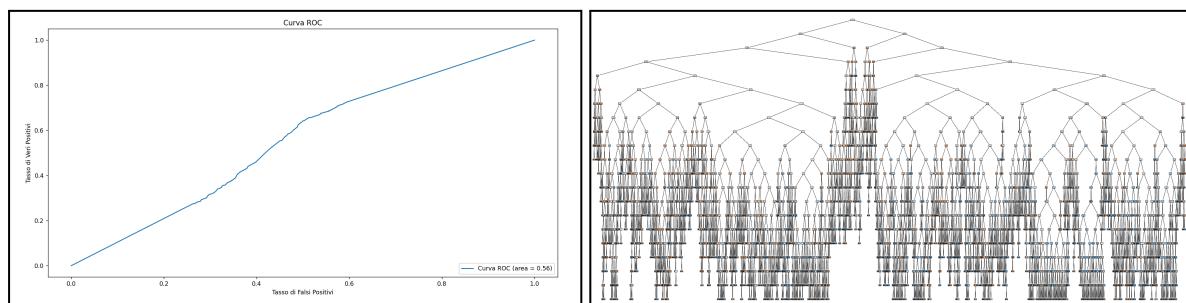
## Decision Tree



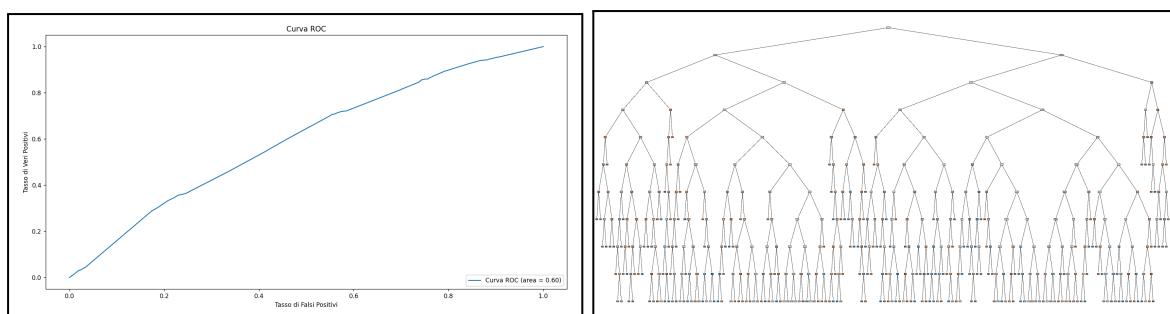
## DT max\_depth = 5



## DT max\_depth = 20



## DT criterion = entropy



## SVM (Support Vector Machine)

Il modello utilizzato fino ad ora del SVM implementa un kernel RBF con C=1000. Le modifiche testate includono: l'assenza di normalizzazione per valutare l'importanza della scalatura dei dati, un kernel lineare per confrontare la capacità di separazione rispetto a funzioni non lineari, e una riduzione del parametro di regolarizzazione (C=1) per esaminare l'impatto della penalizzazione sui margini della classificazione.

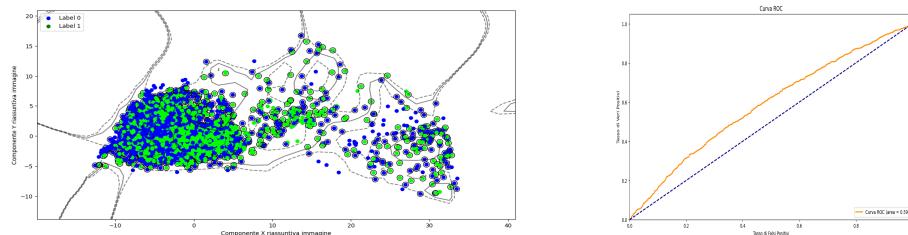
Modello	Accuracy	Precisione (0)	Precisione (1)	Recall (0)	Recall (1)	F1-score (0)	F1-score (1)
<b>SVM</b>	56.15%	58%	54%	60%	52%	59%	53%
<b>SVM (no normalizzazione)</b>	56.87%	59%	55%	59%	54%	59%	55%
<b>SVM (kernel linear)</b>	51.33%	53%	49%	52%	51%	53%	50%
<b>SVM (C = ?)</b>	58.96%	60%	58%	63%	54%	62%	56%

L'analisi dei risultati ottenuti applicando il modello SVM al dataset MURA mostra come la scelta dei parametri influenzi significativamente le prestazioni del modello. L'accuratezza più alta (58.96%) si ottiene con SVM con kernel RBF e parametro C=1, suggerendo che un valore più basso di C riduce l'overfitting e migliora la generalizzazione del modello.

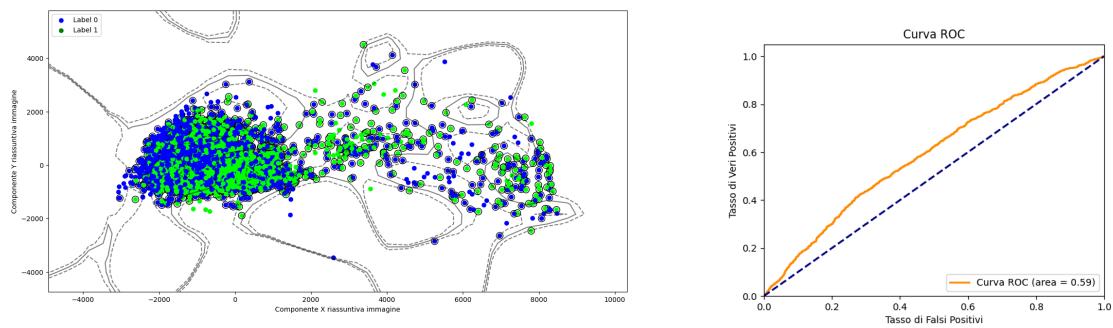
L'uso di un kernel lineare porta a un calo significativo dell'accuratezza (51.33%), indicando che la separabilità dei dati non è ben rappresentata da un'iperpiano lineare e che il kernel RBF è più adatto per questo dataset. Inoltre, la normalizzazione dei dati sembra avere un impatto limitato, con un leggero miglioramento della precisione e del recall rispetto alla versione senza normalizzazione.

Un aspetto chiave è il compromesso tra precisione e recall nelle due classi: mentre un C più basso aumenta il recall per la classe 0, la precisione della classe 1 migliora con valori più alti di C. Questo suggerisce che la scelta del parametro C ha un impatto rilevante sull'equilibrio tra bias e varianza, influenzando il bilanciamento tra le due classi. In sintesi, il miglior compromesso tra accuratezza e performance bilanciate si ottiene con C=1 e kernel RBF.

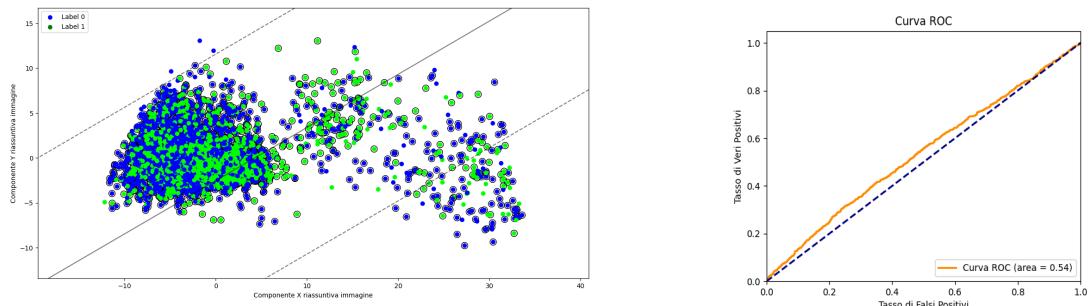
## SVM



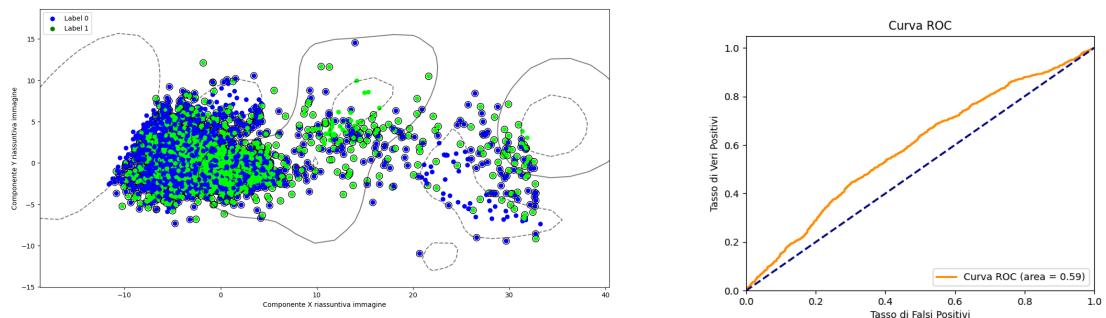
## SVM (no normalizzazione)



## SVM (kernel linear)



## SVM (C=1)



## KNN (K-Nearest Neighbors)

Nel tentativo di ottimizzare il modello KNN vengono valutati gli effetti della normalizzazione dei dati, del numero di vicini k e della metrica di distanza utilizzata. La normalizzazione influenza sulle distanze tra i punti e potrebbe avere un ruolo nella stabilità del modello. La variazione di k consente di osservare come il numero di vicini considerati influenzino la capacità del modello di generalizzare sulle classi. Infine, il confronto tra la metrica euclidea e quella di Chebyshev permette di verificare quale funzione di distanza sia più adatta per questo specifico task.

Modello	Accuracy	Precisione (0)	Precisione (1)	Recall (0)	Recall (1)	F1-score (0)	F1-score (1)
KNN	57.08%	59%	55%	59%	55%	59%	55%
KNN (no normalizzazione)	58.24%	60%	56%	60%	56%	60%	56%
KNN (k=10)	58.71%	59%	58%	67%	49%	63%	53%
metrica (chebyshev)	48.61%	51%	46%	53%	43%	52%	45%

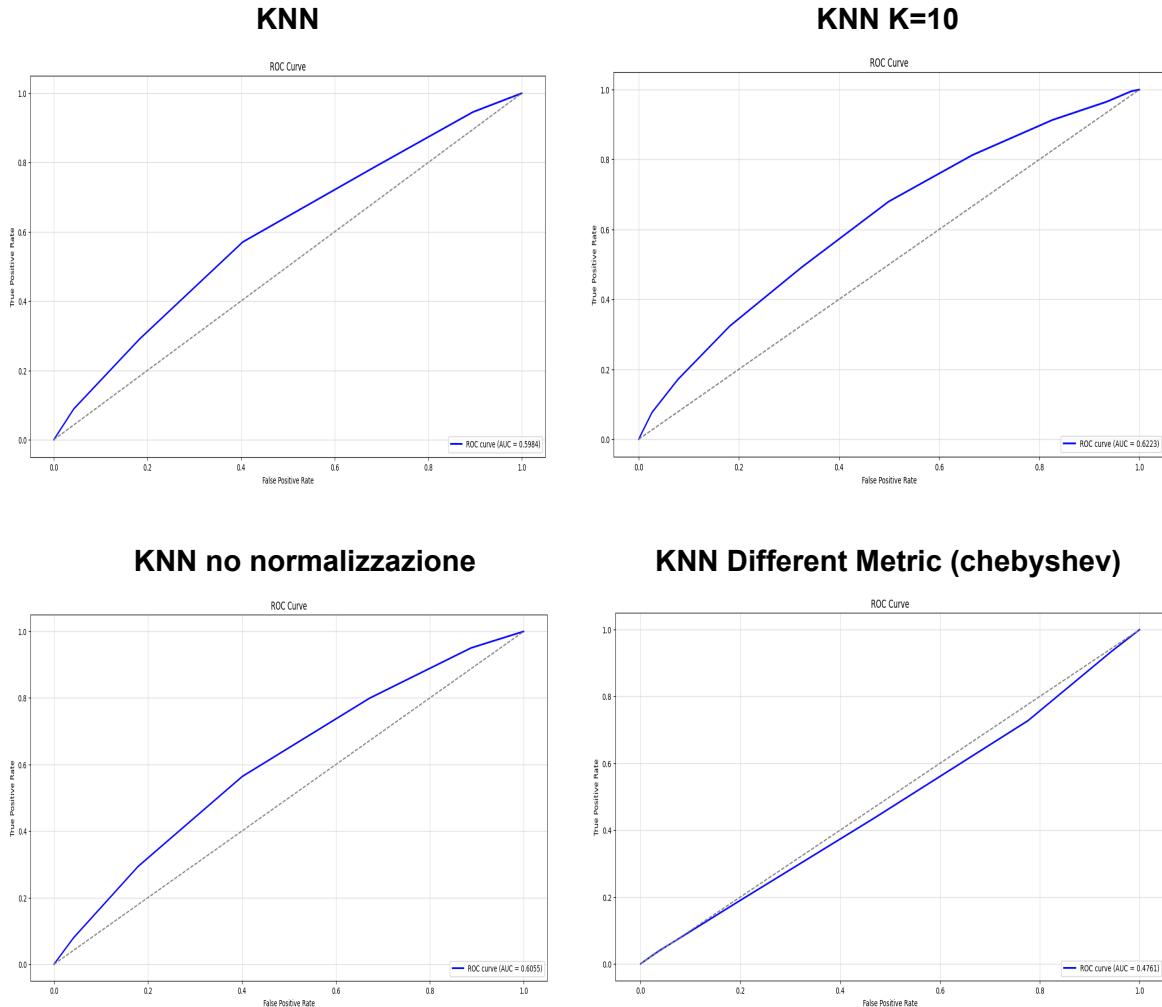
La tabella riassume i risultati ottenuti applicando il modello K-Nearest Neighbors (KNN) al dataset MURA: MSK X-rays, con diverse configurazioni dei parametri. La versione originale del modello, con normalizzazione dei dati, k=5 e distanza euclidea, ha ottenuto un'accuracy del 57.08%, con prestazioni bilanciate tra le due classi (F1-score del 59% per la classe 0 e 55% per la classe 1).

Rimuovendo la normalizzazione dei dati, l'accuracy è aumentata leggermente al 58.24%, suggerendo che la normalizzazione potrebbe non essere strettamente necessaria per questo dataset. Infatti, KNN può essere influenzato dalle trasformazioni dei dati quando la distanza euclidea è utilizzata come metrica, ma in questo caso l'effetto sembra minimo.

Aumentando il numero di vicini a k=10, l'accuracy è salita al 58.71%, e si nota un miglioramento significativo nel recall della classe 0 (67%), a scapito della classe 1, che ha visto un calo del recall al 49%. Questo indica che un valore più alto di k tende a favorire la classe dominante, riducendo la sensibilità verso le minoranze nel dataset.

Infine, cambiando la metrica da euclidea a Chebyshev, l'accuracy è drasticamente calata al 48.61%, con un impatto negativo su tutte le metriche. Questo dimostra come la scelta della funzione di distanza sia cruciale per il modello KNN: la metrica Chebyshev, che considera solo la dimensione con la massima differenza tra punti, sembra meno adatta per il dataset MURA rispetto alla distanza euclidea, che tiene conto delle variazioni su tutte le feature.

In conclusione, l'accuracy del modello KNN può essere migliorata ottimizzando i parametri di  $k$  e della funzione di distanza, mentre la normalizzazione dei dati non ha avuto un impatto rilevante.



## MLP (Multi Layer Perceptron)

Nell'ablation study del Multi Layer Perceptron, vengono testate due variazioni: la riduzione del numero di neuroni nei layer nascosti (da [128, 64] a [64, 32]), per valutare l'impatto sulla capacità di rappresentazione del modello, e la sostituzione dell'ottimizzatore Adam con Stochastic Gradient Descent (SGD), per esaminare come un differente schema di aggiornamento dei pesi influenzi la convergenza e la generalizzazione.

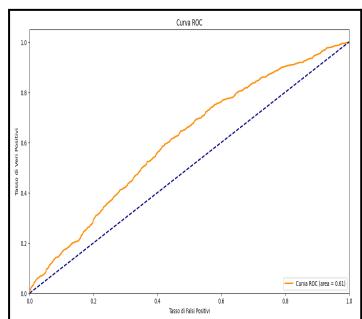
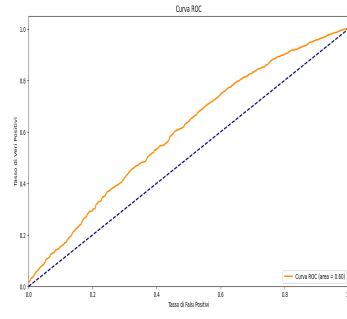
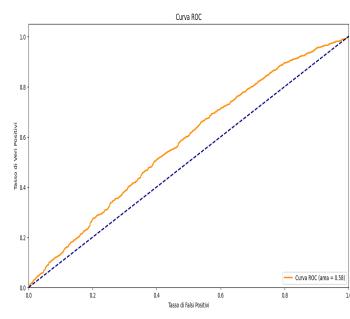
Modello	Accuracy	Precisione (0)	Precisione (1)	Recall (0)	Recall (1)	F1-score (0)	F1-score (1)
MLP	55.74%	60%	53%	45%	68%	51%	59%
Layer nascosti diminuiti (64, 32)	54.8%	60%	52%	39%	72%	48%	60%
funzione di ottimizzazione (SGD)	56.83%	59%	55%	59%	55%	59%	55%

Inizialmente, con la configurazione originale del modello, che prevedeva layer nascosti di dimensioni (128, 64) e l'ottimizzatore Adam, il modello ha raggiunto un'accuratezza del 55.74%, con una precisione della classe 0 pari al 60% e una precisione della classe 1 del 53%. Tuttavia, le metriche di recall e F1-score sono risultate sbilanciate, con la classe 0 che ha registrato un recall del 45% e la classe 1 un recall del 68%. La valutazione della classe 1 ha ottenuto i migliori risultati, con un F1-score del 59%, il che indica una maggiore capacità del modello nel rilevare correttamente i campioni di questa classe.

Successivamente, modificando la struttura della rete, riducendo i layer nascosti a (64, 32), si è osservata una lieve diminuzione dell'accuratezza, che è scesa al 54.8%. La precisione per la classe 0 è rimasta invariata al 60%, ma il recall della classe 0 ha subito una notevole riduzione al 39%, mentre il recall della classe 1 è aumentato al 72%, con un miglioramento significativo dell'F1-score della classe 1 al 60%. Questa configurazione ha dimostrato che la riduzione del numero di neuroni nei layer nascosti può portare a un miglioramento del rilevamento della classe 1, ma a costo di una performance inferiore per la classe 0.

Infine, utilizzando la funzione di ottimizzazione SGD al posto di Adam, si è ottenuto un miglioramento globale delle performance, con un'accuratezza del 56.83%, la più alta tra tutte le configurazioni. La precisione della classe 0 è scesa leggermente al 59%, mentre la precisione della classe 1 è aumentata al 55%. Entrambi i valori di recall sono migliorati, con la classe 0 che ha raggiunto il 59% e la classe 1 che ha mantenuto un buon valore di 55%. L'F1-score per la classe 0 è salito al 59%, suggerendo una performance più equilibrata rispetto alle configurazioni precedenti.

In sintesi, i risultati indicano che l'ottimizzatore Adam inizialmente favorisce una performance più elevata sulla classe 1, mentre l'uso di SGD ha portato a un miglioramento complessivo, soprattutto per la classe 0. La modifica della struttura della rete, con la riduzione del numero di neuroni nei layer nascosti, ha comportato un trade-off tra le prestazioni nelle due classi, con un miglioramento della classe 1 ma una riduzione significativa nella classe 0.

**MLP****MLP ottimizzatore SGD****MLP Layer nascosti (64,32)**

## VGG-16 (Visual Geometry Group 16-layer CNN)

Il modello base VGG-16 utilizzato comprende layer dropout e ottimizzatore Adam. La prima modifica effettuata per testare gli elementi influenti nelle prestazioni del modello consiste nell'eliminazione del layer di dropout presente nel Fully Connected Layer, al fine di analizzare l'effetto della regolarizzazione sulla capacità del modello di generalizzare ai dati di validazione. La seconda prevede la sostituzione dell'ottimizzatore Adam con Stochastic Gradient Descent (SGD) per verificare come una diversa strategia di aggiornamento dei pesi influenzi la stabilità e la convergenza dell'addestramento. I risultati di queste varianti vengono riportati nella tabella seguente e confrontati tra di loro e con il modello di riferimento.

Modello	Accuracy	Precisione (0)	Precisione (1)	Recall (0)	Recall (1)	F1-score (0)	F1-score (1)
<b>VGG-16</b>	64.59%	71%	60%	55%	75%	62%	67%
<b>VGG-16 (eliminazione layer dropout)</b>	67.38%	69%	66%	69%	66%	69%	66%
<b>VGG-16 (sostituzione ADAM con SDG)</b>	55.30%	55%	59%	86%	22%	67%	32%

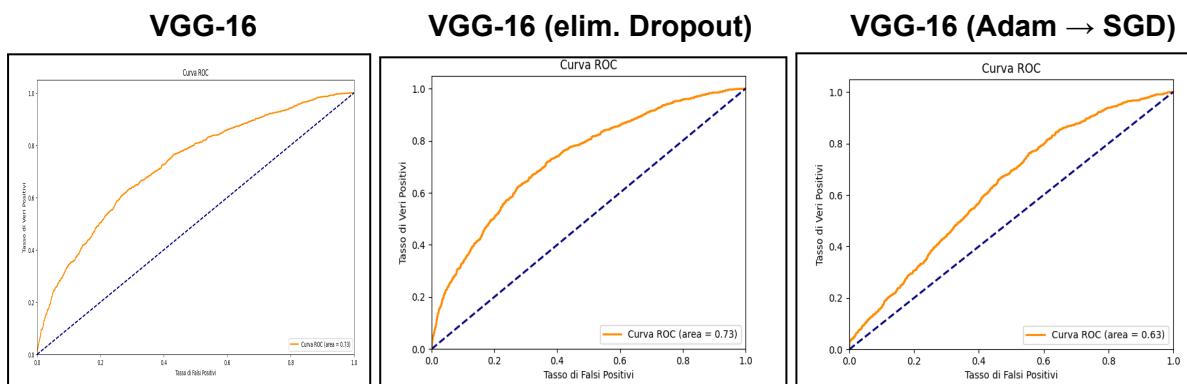
Il modello VGG-16 nella sua configurazione originale (con Dropout e ottimizzatore Adam) ha ottenuto un'accuracy del 64.59%, con una discreta capacità discriminativa tra le due classi. In particolare, la precisione sulla classe 0 è risultata più alta (71%) rispetto alla classe 1 (60%), mentre il recall ha mostrato uno sbilanciamento inverso, con un valore maggiore per la classe 1 (75%) rispetto alla classe 0 (55%). Questo indica che il modello, pur

riconoscendo bene gli esempi della classe 1, presenta una tendenza a commettere falsi positivi.

Con la rimozione del layer di Dropout abbia portato a un incremento della accuracy (+2.79%), suggerendo che il modello, senza regolarizzazione, riesca a sfruttare meglio la capacità espressiva dei suoi livelli densi. Tuttavia, questa variazione potrebbe implicare un maggior rischio di overfitting, in quanto l'aumento della performance sul training set non garantisce necessariamente una migliore generalizzazione.

La sostituzione dell'ottimizzatore Adam con SGD ha invece avuto un impatto negativo sulle prestazioni del modello, con una significativa riduzione dell'accuracy (-9.29%) e un drastico calo del recall della classe 1 (22%), indicando una difficoltà nel riconoscere le immagini di questa categoria. Ciò è probabilmente dovuto al fatto che Adam, grazie al suo adattamento dinamico del tasso di apprendimento, permette una convergenza più rapida e stabile, mentre SGD, senza un tuning ottimale del learning rate, può risultare meno efficace in reti profonde come la VGG-16.

Nel complesso, questi risultati evidenziano come la scelta dell'ottimizzatore e l'uso del dropout influenzino in modo significativo le prestazioni del modello, con il primo che gioca un ruolo chiave nella stabilità dell'addestramento e il secondo nella capacità di generalizzazione.



## DenseNet169 (Densely Connected CN 169-layer)

Per condurre lo studio sulla DenseNet169 sono state costruite tre configurazioni della rete neurale ciascuna con una delle seguenti modifiche:

- **Baseline Model (DenseNet169 originale)**
  - Include il modello pre-addestrato DenseNet169 con 512 neuroni nel Fully Connected Layer e un Dropout del 50%.
- **Modifica 1: Riduzione del Fully Connected Layer**
  - Il numero di neuroni nel layer denso è stato ridotto da 512 → 256, per valutare se una rete più compatta mantiene prestazioni elevate con minori parametri.
- **Modifica 2: Aumento del Dropout**
  - Il tasso di Dropout è stato aumentato da 0.5 → 0.7, per testare se una maggiore regolarizzazione riduce l'overfitting migliorando la generalizzazione.

Modello	Accuracy	Precision (0)	Precision (1)	Recall (0)	Recall (1)	F1-Score (0)	F1-Score (1)
<b>DenseNet169</b>	69.00%	69%	69%	74%	64%	71%	66%
<b>DN169 con FC Layer = 256</b>	67.91%	70%	66%	68%	67%	69%	67%
<b>DN169 con Dropout = 0.7</b>	67.66%	68%	67%	71%	64%	70%	65%

I risultati ottenuti applicando la rete DenseNet169 al dataset MURA: MSK X-rays evidenziano variazioni di performance significative a seguito delle modifiche strutturali effettuate, in particolare la riduzione del numero di neuroni nel Fully Connected Layer e l'aumento del Dropout.

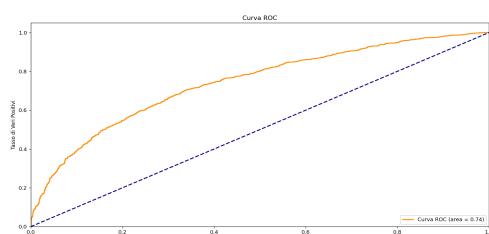
La configurazione originale del modello, ovvero la Baseline, ha raggiunto un'accuratezza del 69.00%, con precisione bilanciata al 69% per entrambe le classi e un recall del 74% per la classe 0 e del 64% per la classe 1.

Riducendo il numero di neuroni nel Fully Connected Layer da 512 a 256, l'accuratezza è rimasta quasi invariata al 67.91%, con valori di precisione e recall stabili o leggermente inferiori, mentre l'F1-score si è mantenuto simile al Baseline (69% per la classe 0 e 67% per la classe 1). Questa modifica non ha compromesso significativamente le prestazioni, rendendo però la rete più leggera e rapida.

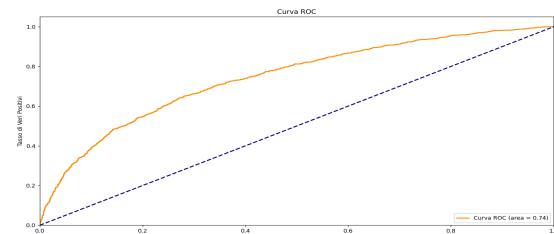
Aumentando il Dropout da 0.5 a 0.7, l'accuratezza è scesa al 67.66%, con una riduzione della precisione per la classe 0 (68%) e un recall migliorato per la stessa classe (71%), mentre il recall della classe 1 è sceso al 64%. L'F1-score della classe 0 è aumentato al 70%, ma quello della classe 1 è leggermente calato al 65%, evidenziando una maggiore generalizzazione a scapito delle prestazioni su una classe.

In sintesi, la Baseline DenseNet169 offre il miglior bilanciamento tra accuratezza e performance sulle classi, ma è computazionalmente più onerosa. La riduzione del Fully Connected Layer è ideale per mantenere prestazioni simili con un modello più leggero, mentre l'aumento del Dropout riduce l'overfitting, migliorando la generalizzazione a discapito di una leggera perdita di accuratezza complessiva.

**DenseNet169 riduzione del FCL**



**DenseNet169 'aumento del dropout'**



## Utilizzo immagini 128x128 per i modelli pre-addestrati

Per tentare di aumentare l'accuracy, sono stati utilizzati i due modelli pre-addestrati con immagini di dimensioni 128x128 invece di 64x64, in modo da fornire una maggiore quantità di informazioni visive al modello e migliorare la capacità di apprendere dettagli più fini nelle immagini radiografiche. L'aumento della risoluzione consente ai modelli di sfruttare strutture più complesse e potenzialmente migliorare le prestazioni di classificazione.

Qui di seguito, confrontiamo i risultati ottenuti con le due risoluzioni per analizzare l'impatto di questo cambiamento sulle performance dei diversi modelli.

Modello	Accuracy	Precision (0)	Precision (1)	Recall (0)	Recall (1)	F1-score (0)	F1-score (1)
DenseNet169 (64x64)	67.31%	70%	65%	65%	70%	67%	67%
DenseNet129 (128x128)	70.94%	72%	70%	73%	68%	73%	69%
VGG-16 (64x64)	64.59%	71%	60%	55%	75%	62%	67%
VGG-16 (128x128)	63.87%	64%	64%	71%	57%	67%	60%

Il confronto tra i modelli DenseNet e VGG-16, sia a risoluzione 64x64 che 128x128, evidenzia alcune differenze significative nelle prestazioni. Per quanto riguarda l'accuratezza, DenseNet169 (a risoluzione 128x128) ottiene la performance migliore con un valore di 70.94%, seguito da DenseNet169 (64x64) con il 67.31%. Al contrario, i modelli VGG-16 ottengono valori inferiori, con il modello a risoluzione 64x64 che raggiunge un'accuratezza del 64.59% e quello a risoluzione 128x128 che si attesta al 63.87%.

Le metriche di precisione mostrano che DenseNet169, sia a 64x64 che a 128x128, ha una precisione migliore nella classe 0 (rispettivamente 70% e 72%) rispetto a VGG-16, che ottiene valori inferiori (71% e 64% a 64x64 e 128x128, rispettivamente). Tuttavia, per quanto riguarda la precisione nella classe 1, DenseNet169 supera VGG-16 in entrambi i casi. La performance di recall per la classe 0 vede DenseNet169 con valori superiori, ma è interessante notare che VGG-16 a 128x128 ottiene un recall della classe 0 del 71%, superiore al 65% di DenseNet169 (64x64). Al contrario, DenseNet169 (128x128) mostra una maggiore sensibilità nella classe 1 con un recall del 73%, che è superiore a quello di VGG-16.

Infine, il F1-score, che combina precisione e recall, evidenzia come DenseNet169 a 128x128 mantenga la performance migliore nella classe 1 (F1-score del 69%), con un leggero vantaggio rispetto agli altri modelli, mentre i modelli VGG-16 mostrano una leggera difficoltà nell'equilibrare precisione e recall.

Nel complesso, i risultati suggeriscono che DenseNet169, in particolare a risoluzione maggiore, fornisce prestazioni superiori rispetto a VGG-16 in diverse metriche, soprattutto per quanto riguarda l'accuratezza e la sensibilità nella classe 1.

## Potenza di calcolo limitata e future evoluzioni dello studio

L'adozione di modelli avanzati come DenseNet169 e VGG-16 implica l'utilizzo di risorse computazionali significative, necessitando di macchine ad alta potenza di calcolo.

Sebbene l'uso di grandi capacità computazionali possa teoricamente migliorare l'efficienza e la precisione dei modelli, è importante sottolineare che tale approccio non è sempre sostenibile, in particolare in contesti con limitate risorse hardware. Tuttavia, il nostro studio si è concentrato sull'applicazione di questi algoritmi a un dataset molto grande e complesso come il MURA, che richiede una gestione accurata dei dati e una notevole potenza di calcolo. In scenari in cui vengono utilizzati dataset di dimensioni più contenute e meno complessi, è possibile che i modelli di machine learning utilizzati in questo studio possano offrire risultati migliori in termini di efficienza computazionale e classificazione, riducendo la necessità di risorse avanzate senza compromettere la qualità dei risultati.

Inoltre, lo studio potrebbe evolversi in future direzioni, esplorando l'uso di nuovi dataset, magari più piccoli e meno complessi, per esaminare come questi impattino sulle prestazioni degli algoritmi. Tale approccio permetterebbe di confrontare i risultati ottenuti con quelli presentati in questo studio, consentendo di valutare ulteriormente l'efficacia degli algoritmi su una varietà di dati e condizioni. In questo contesto, la scelta dell'algoritmo e della dimensione del dataset rimarrebbe un fattore cruciale per ottimizzare le prestazioni del modello in base alle risorse disponibili, guidando eventuali miglioramenti futuri nella classificazione automatizzata delle immagini mediche.

## Conclusioni

Questo studio ha fornito un'analisi dettagliata sulle prestazioni di diversi algoritmi di machine learning e reti neurali applicati alla classificazione di immagini radiografiche del dataset MURA. I risultati evidenziano la superiorità delle reti neurali rispetto agli algoritmi tradizionali, con DenseNet169 che emerge come il modello più efficace, raggiungendo un'accuratezza del 67,31%. Tuttavia, anche VGG-16 ha dimostrato un buon compromesso tra precisione e recall, risultando una scelta interessante in contesti con limitate risorse computazionali.

Dall'analisi comparativa, emerge che gli algoritmi di machine learning tradizionali, come Decision Tree, KNN e SVM, soffrono di prestazioni inferiori rispetto alle reti neurali, con valori di accuratezza inferiori al 58%. Questi modelli, sebbene interpretabili e meno onerosi dal punto di vista computazionale, mostrano limiti nella gestione di dati complessi come le immagini mediche, specialmente nella rilevazione di strutture intricate.

Lo studio contribuisce significativamente alla comprensione delle differenze tra diversi metodi tradizionali di machine learning e diverse reti neurali nel contesto della classificazione delle immagini. Il confronto con studi precedenti suggerisce che, sebbene le

reti neurali utilizzate offrono buone prestazioni, l'utilizzo di modelli nuovi e più specifici potrebbe migliorare molto le prestazioni

*Per future ricerche, si consiglia di:*

- *Esplorare l'integrazione di tecniche di ensemble learning per combinare i punti di forza di algoritmi tradizionali e reti neurali;*
- *Valutare l'efficacia di modelli di deep learning più recenti rispetto agli attuali benchmark;*
- *Investigare l'applicabilità di questi modelli in contesti più ampi, mantenendo sempre un focus sull'equilibrio tra accuratezza e interpretabilità.*

Questi sviluppi potrebbero contribuire a migliorare la comprensione dell'efficacia relativa di diversi approcci di classificazione, con implicazioni anche per il settore medico, dove la scelta del modello ottimale può influenzare l'efficienza diagnostica.

## Fonti

MURA Dataset: <https://stanfordmlgroup.github.io/competitions/mura/>

