Università degli studi di Bari Aldo Moro

SISTEMA DI MONITORAGGIO PER LA RILEVAZIONE DI UN ATTACCO CARDIACO

Gruppo di lavoro: Barile Rossana
Matricola: 738058
Email: r.barile@studenti.uniba.it

INDICE

INTRODUZIONE

Il progetto si focalizza sulla diagnosi degli attacchi di cuore, un tema importante nella medicina moderna. Un attacco di cuore, o infarto miocardico, si verifica quando il flusso sanguigno verso una parte del cuore è bloccato, spesso a causa di un coagulo nelle arterie coronarie. Questa interruzione può portare alla morte delle cellule cardiache per mancanza di ossigeno, rendendo essenziale una diagnosi tempestiva e accurata.

Per affrontare questo problema, utilizzeremo una varietà di algoritmi di apprendimento automatico, tra cui K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree e Random Forest. Questi algoritmi permetteranno di analizzare i dati clinici e i fattori di rischio associati, contribuendo a identificare rapidamente i pazienti a rischio di attacco di cuore. Inoltre, implementeremo tecniche di visualizzazione per rappresentare graficamente i risultati delle diagnosi, facilitando una comprensione chiara e immediata delle informazioni ottenute.

Questo approccio integrato non solo esplorerà diverse metodologie di analisi, ma valuterà anche l'efficacia di ciascun algoritmo nel fornire diagnosi affidabili, contribuendo così a migliorare le pratiche cliniche e a salvare vite umane.

ELENCO ARGOMENTI DI INTERESSE

Creazione di una base di conoscenza:

- Sviluppo di una base di dati per analizzare le informazioni presenti nel dataset.
- Obiettivo di identificare il rischio di attacco cardiaco nei pazienti.

Utilizzo di query:

- Esecuzione di specifiche query per valutare i parametri clinici dei pazienti.

Algoritmi di apprendimento supervisionato impiegati:

- K-Nearest Neighbors (KNN): utilizzato per la classificazione basata sulla vicinanza dei dati.
- Naive Bayes: algoritmo probabilistico per la classificazione dei pazienti.
- Decision Tree: modello di classificazione che utilizza un approccio a struttura ad albero per le decisioni.
- Random Forest: combinazione di più alberi decisionali per migliorare l'accuratezza delle previsioni.

ANALISI DEL PROBLEMA

l sistema di diagnosi di un attacco di cuore è un problema di ottimizzazione che si occupa di determinare il percorso migliore per identificare e trattare i sintomi di un attacco cardiaco. In questo contesto, si può considerare un insieme di indicatori di rischio e segni clinici, con lo scopo di trovare il metodo diagnostico più efficace per valutare la condizione del paziente.

DATI UTILIZZATI

Prima di iniziare a lavorare sul sistema di diagnosi per gli attacchi di cuore, è stato fondamentale condurre un'analisi approfondita dei dati presenti nel dataset, dal nome "cuore", (preso dal seguente sito https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset). È importante comprendere alcuni aspetti chiave del dataset prima di proseguire con l'analisi. Questo è composto dalle seguenti variabili:

- age: Età del paziente.
- sex: Genere del paziente (1 = maschio, 0 = femmina).
- cp: Tipo di dolore toracico (0 = angina tipica, 1 = angina atipica, 2 = dolore non correlato all'angina, 3 = asintomatico).
- TRTBPS: Pressione sanguigna a riposo (in mmHg).

- CHOL: Livello di colesterolo in mg/dl, misurato tramite un sensore BMI.
- fbs: Zucchero nel sangue a digiuno (> 120 mg/dl) (1 = vero, 0 = falso).
- restecg: Risultati dell'elettrocardiogramma a riposo (0 = normale, 1 = anomalia dell'onda ST-T, 2 = ipertrofia del ventricolo sinistro).
- thalachh: Frequenza cardiaca massima raggiunta.
- exng: Angina pectoris indotta dall'esercizio (1 = sì, 0 = no).
- old peak: Picco precedente.
- slp: Pendenza dell'elettrocardiogramma.
- caa: Numero di grandi vasi.
- thall: Risultato del test Thallium (0.3).
- output: Variabile target.

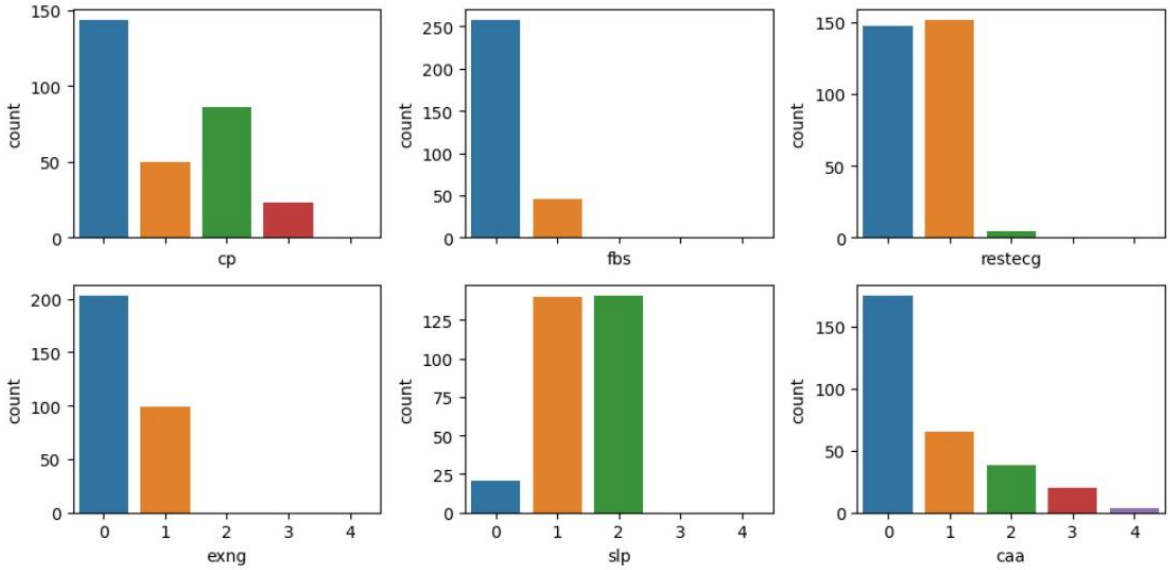
Queste informazioni sono essenziali per una corretta interpretazione e analisi dei dati al fine di sviluppare un sistema diagnostico efficace.

STRUMENTI AGGIUNTIVI

Questo codice permette di visualizzare la distribuzione di sei variabili categoriali relative a caratteristiche cliniche presenti nel dataset. Utilizzando i countplot, è possibile osservare rapidamente la frequenza di ciascuna categoria per ogni variabile, ad esempio il numero di persone con un certo tipo di dolore toracico o con glicemia alta.

L'inserimento di questi grafici risulta particolarmente utile nell'analisi esplorativa dei dati (EDA), poiché consente di comprendere meglio come sono distribuiti i valori delle variabili categoriali. Queste informazioni possono rivelarsi preziose per individuare eventuali squilibri nelle classi o tendenze nei dati, elementi che

potrebbero influenzare sia l'accuratezza di un modello di machine learning, sia l'interpretazione dei risultati.



KNOWLEDGE BASE O BASE DI CONOSCENZA

Una base di conoscenza è un costrutto fondamentale per la rappresentazione della conoscenza in un formato formale e logico. Essa rappresenta un insieme di informazioni che possono essere interrogate, manipolate e inferite, costituendo la spina dorsale dei sistemi di intelligenza artificiale e dei programmi di ragionamento automatico. La sua struttura è composta principalmente da fatti e regole, che lavorano insieme per consentire il ragionamento e la deduzione di nuove informazioni.

COMPOSIZIONE DELLA BASE DI CONOSCENZA

I fatti sono le unità di informazione di base all'interno della base di conoscenza. Rappresentano affermazioni concrete o verità accettate riguardo il dominio in questione. I fatti non necessitano di ulteriori prove per essere considerati veri; la loro validità è assunta come predefinita. Queste affermazioni possono riguardare entità, relazioni o proprietà specifiche.

Le regole, d'altra parte, sono affermazioni condizionali che stabiliscono relazioni tra i fatti esistenti nella base di conoscenza. Esse definiscono come alcune conclusioni possano essere dedotte da altre affermazioni. Una regola è composta da una parte antecedente, nota come "corpo", che specifica le condizioni necessarie affinché la conclusione, o "testa", possa essere considerata vera. Le regole consentono di formulare ragionamenti più complessi, trasformando la base di conoscenza in un sistema in grado di fornire nuove informazioni a partire da quelle già presenti.

FUNZIONAMENTO DELLA BASE DI CONOSCENZA

Il funzionamento di una base di conoscenza in Prolog è caratterizzato da diversi processi logici che permettono l'interazione con l'utente e la manipolazione delle informazioni. Gli utenti possono interagire con la base di conoscenza ponendo query o domande. Queste query possono essere formulate per verificare l'esistenza di un certo fatto, esplorare le relazioni tra diverse entità o inferire informazioni non esplicitamente presenti nella base di conoscenza. Prolog elabora queste query e restituisce risposte basate sulla conoscenza codificata.

In situazioni in cui non è disponibile una risposta diretta a una query, Prolog utilizza il processo di inferenza per dedurre nuove informazioni. Questo processo si basa sull'applicazione delle regole alla luce dei fatti esistenti. Se una regola è soddisfatta, Prolog può inferire conclusioni aggiuntive, estendendo così la propria base di conoscenza e rispondendo a query più complesse.

Un altro aspetto fondamentale del funzionamento di Prolog è il backtracking. Questo meccanismo consente al sistema di esplorare diverse possibilità di soluzione per una query. Se Prolog trova che una possibile via non porta a una conclusione valida, è in grado di tornare indietro e provare un'altra opzione, garantendo così una ricerca esaustiva delle risposte possibili. Il backtracking rende Prolog particolarmente potente nella gestione di problemi complessi e nella ricerca di soluzioni ottimali.

Prolog utilizza il concetto di unificazione, un processo che consente di confrontare e combinare termini all'interno delle query e delle regole. La unificazione è essenziale

per stabilire corrispondenze tra variabili e valori specifici, permettendo a Prolog di adattare le regole e i fatti in base ai dati forniti dall'utente. Questo processo è alla base della capacità di Prolog di inferire nuove conoscenze e di rispondere in modo pertinente alle domande poste.

La base di conoscenza in è un costrutto essenziale che consente di rappresentare, organizzare e inferire informazioni in modo logico e coerente. Attraverso l'uso di fatti e regole, Prolog si configura come uno strumento potente per il ragionamento automatico, in grado di gestire la complessità delle relazioni all'interno di un dominio specifico. La sua capacità di inferenza, backtracking e unificazione rende Prolog un linguaggio di programmazione particolarmente adatto per applicazioni in intelligenza artificiale, dove la rappresentazione della conoscenza e il ragionamento logico sono fondamentali.

DECISIONI DI PROGETTO

Nel progetto sono state formulate delle normative per la creazione della base di conoscenza. Grazie a queste normative, l'utente ha la possibilità di elaborare query per interrogare il sistema. Di seguito, sono presentate e illustrate in dettaglio le regole redatte in Prolog:

REGOLA 1

```
% Regola 1: Identificare se un paziente è a rischio di attacco cardiaco
paziente_a_rischio(Eta, Colesterolo, AnginaEsercizio, TipoDolore) :-
    Eta > 50,
    Colesterolo > 240,
    AnginaEsercizio = 1,
    TipoDolore > 1.
```

Questa regola Prolog ha lo scopo di determinare se un paziente presenta un rischio elevato di attacco cardiaco sulla base di specifiche variabili cliniche, incluse età, livelli di colesterolo, angina durante l'esercizio fisico e tipo di dolore. L'approccio logico consente un'elaborazione rapida e automatizzata delle informazioni cliniche.

La regola utilizza una forma di logica proposizionale per valutare se tutte le condizioni specificate sono vere. Se tutte le condizioni sono soddisfatte, il predicato "paziente_a_rischio" restituisce "true", indicando che il paziente è considerato a rischio di attacco cardiaco. Questo approccio permette una facile estensione e modifica della logica, facilitando l'adattamento alle linee guida cliniche emergenti o alle nuove evidenze scientifiche.

- Predicato: paziente_a_rischio. Il predicato accetta quattro parametri di input:
    - Età: rappresenta l'età del paziente (numerico).
    - Colesterolo: rappresenta il livello di colesterolo nel sangue (numerico).
    - AnginaEsercizio: rappresenta un flag binario (1 o 0) per indicare la presenza di angina durante l'esercizio.
    - TipoDolore: un valore numerico che classifica l'intensità del dolore.
- Condizioni di Attivazione:
    - Eta > 50: Questa condizione verifica che l'età del paziente sia superiore a 50 anni.
    - Colesterolo > 240: Questa condizione verifica che il livello di colesterolo sia superiore a 240 mg/dL.
    - AnginaEsercizio = 1: Questa condizione verifica che il paziente abbia riportato angina durante l'esercizio fisico.
    - TipoDolore > 1: Questa condizione verifica che il tipo di dolore avvertito dal paziente sia di intensità moderata o elevata.

REGOLA 2

```
% Regola 2: Tipo di attacco basato su pressione e picco ST
tipo_attacco(Eta, Colesterolo, AnginaEsercizio, TipoDolore, Pressione, PiccoPrecedente, alto) :-
    paziente_a_rischio(Eta, Colesterolo, AnginaEsercizio, TipoDolore),
    Pressione > 160,
    PiccoPrecedente > 2.5.

tipo_attacco(Eta, Colesterolo, AnginaEsercizio, TipoDolore, Pressione, PiccoPrecedente, medio) :-
    paziente_a_rischio(Eta, Colesterolo, AnginaEsercizio, TipoDolore),
    Pressione > 140, Pressione =< 160,
    PiccoPrecedente =< 2.5.

tipo_attacco(Eta, Colesterolo, AnginaEsercizio, TipoDolore, Pressione, PiccoPrecedente, basso) :-
    paziente_a_rischio(Eta, Colesterolo, AnginaEsercizio, TipoDolore),
    Pressione =< 140.
```

La prima regola classifica un paziente come ad alto rischio di attacco cardiaco se soddisfa determinate condizioni. Innanzitutto, il paziente deve essere identificato come a rischio, attraverso la chiamata al predicato “paziente_a_rischio”. Inoltre, la pressione sanguigna deve essere superiore a 160 mmHg e il valore di picco ST precedente deve superare 2.5. L'inclusione di soglie specifiche per pressione sanguigna e picco ST consente di identificare pazienti con un rischio elevato in modo preciso, basandosi su dati clinici oggettivi.

- Condizioni: Questa regola classifica un paziente come ad alto rischio di attacco cardiaco se soddisfa le seguenti condizioni:

  1. Il paziente è identificato come a rischio (paziente_a_rischio).
  2. La pressione sanguigna è superiore a 160 mmHg.
  3. Il valore di picco ST precedente è superiore a 2.5.

L'inclusione di soglie specifiche per pressione sanguigna e picco ST consente di identificare pazienti con un rischio elevato in modo preciso, basandosi su dati clinici oggettivi.

La seconda regola classifica un paziente come a rischio medio, sempre previa identificazione come paziente a rischio. In questo caso, la pressione sanguigna deve essere compresa tra 140 mmHg e 160 mmHg (inclusi), e il valore di picco ST precedente deve essere minore o uguale a 2.5. La definizione di un range di

pressione sanguigna specifico e un limite per il picco ST permette una categorizzazione più raffinata, garantendo una risposta adeguata per i pazienti non gravemente compromessi.

- Condizioni: questa regola classifica un paziente come a rischio medio se soddisfa le seguenti condizioni:
    1. Il paziente è identificato come a rischio (paziente_a_rischio).
    2. La pressione sanguigna è compresa tra 140 mmHg e 160 mmHg (inclusi).
    3. Il valore di picco ST precedente è minore o uguale a 2.5.

La definizione di un range di pressione sanguigna specifico e un limite per il picco ST permette una categorizzazione più raffinata, garantendo una risposta adeguata per i pazienti non gravemente compromessi.

La terza regola classifica un paziente come a rischio basso, richiedendo che il paziente sia identificato come a rischio. Inoltre, la pressione sanguigna deve essere minore o uguale a 140 mmHg. Stabilendo una soglia di pressione sanguigna ben definita, questa regola fornisce un chiaro segnale per identificare pazienti che, sebbene a rischio, presentano valori clinici che indicano una minore probabilità di attacco cardiaco imminente.

- Condizioni: Questa regola classifica un paziente come a rischio basso se soddisfa le seguenti condizioni:
    1. Il paziente è identificato come a rischio (paziente_a_rischio).
    2. La pressione sanguigna è minore o uguale a 140 mmHg.

Stabilendo una soglia di pressione sanguigna ben definita, questa regola fornisce un chiaro segnale per identificare pazienti che, sebbene a rischio, presentano valori clinici che indicano una minore probabilità di attacco cardiaco imminente.

REGOLE 3

```
% Regola 3: Età media dei pazienti
eta_media(ListaEta, EtaMedia) :-
    sumlist(ListaEta, Somma),
    length(ListaEta, NumeroPazienti),
    EtaMedia is Somma / NumeroPazienti.
```

La regola Prolog “eta_media” calcola l'età media dei pazienti a partire dalla lista dell’età presente nel dataset. Accetta due argomenti: “ListaEta”, che è una lista di numeri rappresentanti le età dei pazienti, e “EtaMedia”, un numero reale che rappresenta l'età media calcolata.

Utilizza il predicato built-in “sumList” per calcolare la somma di tutti gli elementi nella lista “ListaEta”„assegnando il risultato alla variabile “Somma”. Usa il predicato built-in “lenght” per determinare il numero di elementi nella lista “ListaEta”„ assegnando il risultato alla variabile “NumeroPazienti”. Successivamente calcola l'età media dividendo la somma delle età per il numero di pazienti e assegna il risultato alla variabile “EtaMedia” utilizzando l'operatore “is”

REGOLA 4

```
% Regola 4: Determinare se un paziente può avere un attacco cardiaco
puo_avere_attacco_cardiaco_prob(Eta, TipoDolore, AnginaEsercizio, Pendenza,
                                NumeroVasi, RisultatoThallium, FrequenzaCardiaca,
                                PiccoPrecedente, PuoAvereAttacco) :-
    (
        (TipoDolore == 0 -> ValoreCondizione1 = 0; ValoreCondizione1 = 1),
        (AnginaEsercizio == 1 -> ValoreCondizione2 = 0; ValoreCondizione2 = 1),
        ((Pendenza == 0; Pendenza == 1) -> ValoreCondizione3 = 0; ValoreCondizione3 = 1),
        (NumeroVasi =< 3 -> ValoreCondizione4 = 0; ValoreCondizione4 = 1),
        (RisultatoThallium == 3 -> ValoreCondizione5 = 0; ValoreCondizione5 = 1),
        ((Eta > 50; Eta < 30) -> ValoreCondizione6 = 0; ValoreCondizione6 = 1),
        (FrequenzaCardiaca < 130 -> ValoreCondizione7 = 0; ValoreCondizione7 = 1),
        (PiccoPrecedente > 2.0 -> ValoreCondizione8 = 0; ValoreCondizione8 = 1)
    ),
    SommaCondizioni is ValoreCondizione1 + ValoreCondizione2 + ValoreCondizione3 +
    ValoreCondizione4 + ValoreCondizione5 + ValoreCondizione6 + ValoreCondizione7 +
    ValoreCondizione8,
    (SommaCondizioni > 4 -> PuoAvereAttacco = si; PuoAvereAttacco = no).
```

Questa regola Prolog ha lo scopo di determinare se un paziente può essere a rischio di attacco cardiaco, basandosi su una serie di parametri clinici. La funzione riceve in ingresso diversi valori relativi alla condizione fisica del paziente, come l'età, il tipo di dolore toracico, la presenza di angina durante l'esercizio fisico, il numero di vasi

sanguigni interessati, il risultato di un esame con tallio, la frequenza cardiaca massima e la pressione sanguigna in un determinato momento.

Il comportamento della funzione si basa sull'applicazione di otto condizioni che controllano ciascuna una specifica caratteristica del paziente. Per ogni condizione, viene effettuato un controllo logico che determina se il parametro corrispondente suggerisce un rischio maggiore o minore di attacco cardiaco. Se una determinata caratteristica si ritiene non influente o riduce il rischio, viene assegnato il valore 0 a una variabile temporanea, mentre se aumenta il rischio, viene assegnato il valore 1.

Ogni controllo segue una struttura semplice di tipo "if-then-else". Per esempio, se il tipo di dolore al petto è uguale a 0 (il che potrebbe indicare assenza di dolore), la variabile associata a questa condizione assume il valore 0; altrimenti, assume il valore 1. Lo stesso vale per le altre condizioni: l'angina durante l'esercizio fisico, la pendenza del tratto ST nell'ECG, il numero di vasi sanguigni e così via. Ogni parametro è trattato individualmente, e alla fine tutti i valori temporanei ottenuti vengono sommati.

Il risultato di questa somma rappresenta un indice di rischio. Se il totale delle condizioni che indicano rischio è superiore a 4, allora la funzione conclude che il paziente è a rischio di avere un attacco cardiaco, restituendo come risultato il valore "sì". Se, invece, la somma è 4 o inferiore, la funzione restituisce "no", indicando che il paziente ha un rischio basso.

REGOLA 5

```
% Regola 5: Condizione cardiovascolare grave
condizione_grave(Colesterolo, NumeroVasi, FrequenzaCardiaca) :-
    Colesterolo > 300,
    NumeroVasi >= 2,
    FrequenzaCardiaca < 100.
```

Questa regola Prolog che descrive una condizione cardiovascolare grave è progettata per identificare situazioni di rischio sulla base di tre parametri clinici: il livello di colesterolo, il numero di vasi sanguigni interessati e la frequenza cardiaca.

Questa regola funziona come un filtro logico, verificando se i valori forniti per questi parametri superano o rientrano in soglie specifiche.

Nello specifico, la regola stabilisce che una condizione è considerata grave quando:

1. Il colesterolo supera i 300 mg/dL, indicando un livello particolarmente elevato di colesterolo nel sangue.
2. Sono coinvolti almeno due vasi sanguigni (indicati dal parametro NumeroVasi), il che suggerisce un'estensione significativa del problema cardiovascolare.
3. La frequenza cardiaca è inferiore a 100 battiti per minuto, che può riflettere una ridotta risposta del cuore sotto sforzo o in situazioni di rischio.

Quando questi tre criteri sono soddisfatti contemporaneamente, il sistema Prolog considera vera la regola e conclude che il paziente si trova in una condizione cardiovascolare grave.

Questa logica permette al sistema di classificare automaticamente le condizioni cardiovascolari, fornendo un meccanismo utile per evidenziare i casi di rischio elevato. In un contesto medico, potrebbe essere integrata in un sistema di supporto alle decisioni per agevolare la valutazione dei pazienti sulla base di dati clinici essenziali.

APPRENDIMENTO SUPERVISIONATO

L'apprendimento supervisionato è una tecnica di machine learning in cui un algortimo viene addestrato su dati etichettati, cioè su input associati a output noti. L'obiettivo è che il modello impari a mappare correttamente gli input agli output, così da poter fare previsioni su nuovi dati non visti. Il processo di divide in due fasi: addestramento, dove il modello impara dai dati, e test, dove viene valutata la sua capacità di generalizzare. Le sfide principali includono evitare l'verfitting e scegliere modelli e caratteristiche adeguate per ottenere buone prestazioni su dati nuovi.

SOMMARIO

Il progetto si concentra sulla diagnosi di possibili problemi cardiaci di un paziente, utilizzando come variabile di riferimento quella dell'output, che rappresenta il target da predire a partire dalle altre caratteristiche del paziente.

I modelli che sono stati scelti e valutati sono i seguenti: K-Nearest Neighbors (KNN), Random Forest, Decision Tree e Naive Bayes, utilizzando per tutti Scikit-learn.ù

DECISIONI DI PROGETTO

Inizialmente, i diversi modelli sono stati testati senza applicazione di parametri specifici, con l'obiettivo di ottenere una panoramica preliminare dei loro comportamenti. Questa fase esplorativa ci ha permesso di valutare le prestazioni allo stato iniziale dei modelli, fornendo una base di confronto per eventuale miglioramenti futuri. I risultati ottenuti in questi primo step sono i seguenti.

```
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.86      0.80        14
           1       0.87      0.76      0.81        17

    accuracy                           0.81        31
   macro avg       0.81      0.81      0.81        31
weighted avg       0.81      0.81      0.81        31

---------------------------------------
```

```
KNN Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.93      0.87        14
           1       0.93      0.82      0.88        17

    accuracy                           0.87        31
   macro avg       0.87      0.88      0.87        31
weighted avg       0.88      0.87      0.87        31

---------------------------------------
Naive Bayes Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90        14
           1       0.94      0.88      0.91        17

    accuracy                           0.90        31
   macro avg       0.90      0.91      0.90        31
weighted avg       0.91      0.90      0.90        31

---------------------------------------
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       0.58      0.79      0.67        14
           1       0.75      0.53      0.62        17

    accuracy                           0.65        31
   macro avg       0.66      0.66      0.64        31
weighted avg       0.67      0.65      0.64        31
```

È stato creato un report in cui ho analizzato le metriche di valutazione per ciascun modello. In particolare, ho approfondito tre indicatori fondamentali: precision, recall e f1-score. Queste metriche sono essenziali per comprendere le prestazioni dei vari modelli. È stato realizzato un report complessivo sui diversi metodi, al fine di confrontare l'accuratezza iniziale.

- PRECISION: la precision è una metrica che valuta l'accuratezza delle predizioni positive di un modello. In altre parole, indica la proporzione di instaze correttamente classificate come positive rispetto al totale delle istante classificate positive. Risulta particolarmente utile quando il costo è particolarmente elevato. La precision si calcola con la seguente formula:

$$Precision = \frac{Vero\ Positivo\ (TP)}{Vero\ Positivo(TP) +\ Falso\ Positivo(FP)}$$

- RECALL: la recall misura la capacità del modello di identificare tutte le istanze positive nel dataset. Indica la porzione di istanze positive correttamente classificate rispetto al totale delle istanze realemente positive. Risulta molto utile quando il costo di un falso negativo è elevato. La recall si calcola con la seguente formula:

$$Recall = \frac{Vero\ Positivo(TP)}{Vero\ Positivo(TP) + Falso\ Negativo(FN)}$$

- F1-SCORE: l'F1 Score è una misura che combina sia la precision che la recall in un unico valore. È utile in situazioni in cui si desidera il bilanciamento tra le due metriche. L'F1 Score è particolarmente rilevante quando di ha una distribuzione sbilanciata delle classi, poiché fornisce un'indicazione più compleya delle performance del modello rispetto alla sola precision o alla sola recall. La formula dell'F1 Score è la seguente:

$$F1 = 2 * \frac{Precisiom * Recall}{Precision + Recall}$$

Le decisioni cliniche sono fortemente influenzate dalle metriche di valutazione come la precision e la recall. Un modello che presenta valori elevati in entrambe le metriche risulta più affidabile e, di conseguenza, può offrire un valido supporto ai medici nella diagnosi degli attacchi cardiaci.

Queste metriche non solo riflettono l'affidabilità del modello, ma forniscono anche prezioni informazioni sulle caratteristiche del dataset utilizzato. Per esempio, se un modello mostra una precisione elevata ma una recall bassa, ciò potrebbe indicare la necessità di un'analisi più approfondita del dataset stesso. Potrebbe esserci, infatti, uno squilibrio tra le classi di dati oppure alcuni sintomi potrebbero non essere adeguatamente rappresentati, compromettendo la qualità della diagnosi.

Durante il processo di sviluppo del modello, il monitoraggio continuo di queste metriche è importante per ottimizzare gli algoritmi di apprendimento automatico. Questo approccio non solo migliora le prestazioni predittive del modello, ma contribuisce anche a garantire diagnosi più accurate e tempestive.

Inoltre, identificare correttamente i pazienti a rischio di attacco di cuore non è solo una questione di diagnosi, ma può anche avere un impatto significativo sulle strategie di trattamento. Un'identificazione precisa consente l'implementazione di misure preventive, migliorando così la prognosi e riducendo il numero di eventi avversi. In questo modo, le metriche di precisione e recall si dimostrano fondamentali non solo nella fase di analisi, ma anche nella pratica quotidiana.

RANDOM FOREST

Il modello Random Forest è utilizzato per la classificazione di un dataset,e in questo si tratta di un sistema di monitoraggio per gli attacchi cardiaci. La sua finalità principale è addestrare il modello su diverse suddivisioni dei dati e valutare le prestazioni attraverso più iterazioni (o run), al fine di analizzare la robustezza e la stabilità del modello. Viene prestata particolare attenzione a metriche fondamentali come l'accuratezza e l'F2-score, per comprendere l'efficacia del modello in termini di precisione complessiva e capacità di ridurre i falsi negativi.

L'obiettivo è valutare il modello su diverse partizioni del dataset per comprendere la sua stabilità. A tal fine, vengono eseguite 10 iterazioni. Durante ogni run, i dati vengono suddivisi casualmente in un training set (90%) e un test set (10%) utilizzando il metodo "train_test_split()", con "random_state=i" che cambia ad ogni iterazione per garantire suddivisioni differenti.

In ogni iterazione, si costruisce un modello Random Forest, che è un algoritmo di apprendimento supervisionato basato su un insieme di alberi decisionali. Questo approccio ensemble combina i risultati di diversi alberi per migliorare la capacità predittiva e ridurre il rischio di overfitting. Il modello viene addestrato con i dati di training e successivamente testato su quelli di test. Le predizioni vengono quindi confrontate con i valori effettivi del test set per calcolare l'accuratezza: la proporzione di predizioni corrette rispetto al totale.

Nel modello si calcolano la media e la deviazione standard per l'accuratezza. La media fornisce una misura sintetica della performance complessiva del modello,

mentre la deviazione standard consente di valutare la stabilità delle performance tra i vari run. Una bassa deviazione standard indica che il modello è coerente nelle diverse suddivisioni dei dati, mentre una deviazione standard elevata può suggerire una maggiore variabilità.

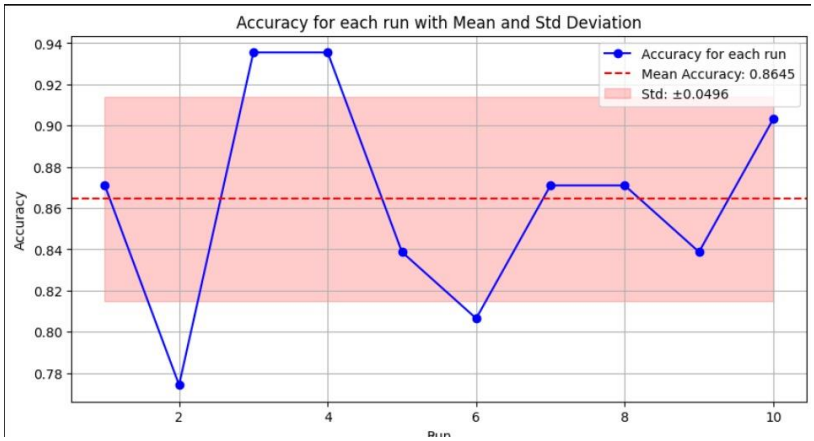
Dopo l'ultimo run, il modello viene valutato sul test set finale, e viene generato un classification report. Questo report fornisce ulteriori dettagli sulle prestazioni del modello, tra cui precision**,** recall e F1-scor**e** per ciascuna classe. Questi dettagli permettono di avere una visione più ampia delle capacità del modello, oltre la semplice accuratezza, e aiutano a comprendere meglio come si comporta nei confronti delle diverse classi di output.

```
Mean Accuracy over 10 runs: 0.8645
Standard Deviation of Accuracy: 0.0496

Classification Report (last run):
              precision    recall  f1-score   support

           0       0.89      0.80      0.84        10
           1       0.91      0.95      0.93        21

    accuracy                           0.90        31
   macro avg       0.90      0.88      0.89        31
weighted avg       0.90      0.90      0.90        31
```



KNN (K-NEAREST NEIGHBORS)

L’obiettivo del K-Nearest Neighbors (KNN) è addestrare e valutare il modello su un dataset riguardante problemi cardiaci attraverso una serie di esecuzioni (o run) ripetute. L'idea alla base è esaminare le prestazioni del modello su diverse

suddivisioni dei dati e misurare due metriche chiave: l'accuratezza e l'F2-score, per analizzare la robustezza e la stabilità del modello in vari contesti di test.

Il KNN è un algoritmo di classificazione che classifica un campione sulla base della vicinanza ai suoi vicini più prossimi (k-nearest neighbors). In pratica, dato un nuovo campione, il modello calcola la distanza tra il campione e i campioni di training, assegnando l'etichetta della classe più comune tra i "k" vicini più prossimi. Questo lo rende particolarmente utile in scenari con dati ben separati ma sensibili al rumore nei dati.

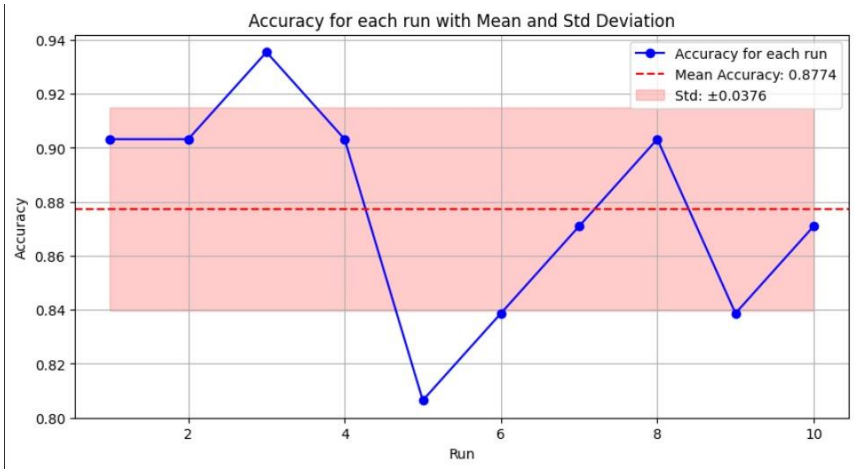
Anche per questo modello, così come per il Random Forest, è stata fatta una valutazione utilizzando e calcolando la media e deviazione standard tramite dieci run

```
Mean Accuracy over 10 runs: 0.8774
Standard Deviation of Accuracy: 0.0376

Classification Report (last run):
              precision    recall  f1-score   support

           0       0.80      0.80      0.80        10
           1       0.90      0.90      0.90        21

    accuracy                           0.87        31
   macro avg       0.85      0.85      0.85        31
weighted avg       0.87      0.87      0.87        31
```



Questo grafico rappresenta l'accuratezza per ciascuna delle 10 esecuzioni. Ogni punto sulla linea rappresenta l'accuratezza ottenuta in quel run specifico, mentre una linea tratteggiata orizzontale indica la media dell'accuratezza complessiva.

L'area ombreggiata rappresenta la deviazione standard**,** mostrando l'intervallo entro il quale l'accuratezza tende a oscillare. Questo consente di visualizzare la stabilità del modello: una variazione limitata indica che il modello è abbastanza robusto rispetto alla suddivisione dei dati.

NAIVE BAYES

L'obiettivo di Naive Bayes Gaussiano (GaussianNB) è valutare le prestazioni del modello su un dataset relativo a problemi cardiaci attraverso un processo iterativo su 10 esecuzioni (run). L'approccio scelto consente di osservare le variazioni delle metriche di prestazione, come l'accuratezza e l'F2-score, su più suddivisioni dei dati di training e testing, garantendo una valutazione robusta delle capacità predittive del modello.

Il modello Naive Bayes Gaussiano è un classificatore probabilistico basato sul teorema di Bayes con l'assunzione che le feature siano indipendenti tra loro, e che seguano una distribuzione gaussiana (normale). È particolarmente efficiente su dataset con caratteristiche numeriche e può essere utilizzato per problemi di classificazione binaria o multiclasse. La semplicità computazionale lo rende veloce da addestrare, sebbene l'ipotesi di indipendenza delle feature sia spesso una semplificazione eccessiva.

Per ciascun run, si calcola l'accuratezza: la percentuale di predizioni corrette rispetto al totale dei dati di test. L'accuratezza è utile come indicatore generale delle prestazioni, ma può risultare fuorviante in dataset sbilanciati (dove una classe è molto più rappresentata dell'altra).

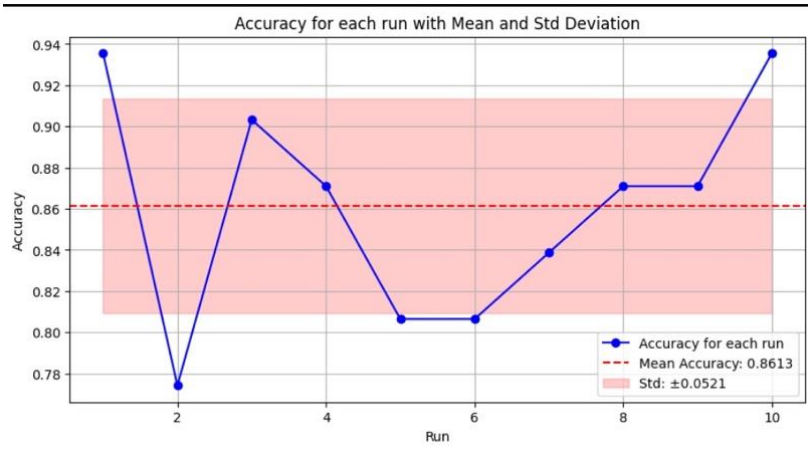
Al termine delle dici iterazioni, viene calcolata la media e deviazione standard dell'accuratezza: la media rappresenta l'accuratezza complessiva del modello nei diversi run, mentre la deviazione standard indica la variabilità delle prestazioni. Una deviazione standard bassa suggerisce che il modello è stabile e che le sue prestazioni non variano molto tra le diverse suddivisioni dei dati.

```
Mean Accuracy over 10 runs: 0.8613
Standard Deviation of Accuracy: 0.0521

Classification Report (last run):
              precision    recall  f1-score   support

           0       0.83      1.00      0.91        10
           1       1.00      0.90      0.95        21

    accuracy                           0.94        31
   macro avg       0.92      0.95      0.93        31
weighted avg       0.95      0.94      0.94        31
```



Il grafico mostra l'accuratezza ottenuta in ciascuno dei 10 run, con una linea orizzontale che rappresenta la media dell'accuratezza complessiva. L'area ombreggiata attorno alla media rappresenta l'intervallo di una deviazione standard, fornendo un'idea della variabilità tra le diverse esecuzioni. Questo tipo di grafico è utile per valutare la stabilità del modello: se i punti sono distribuiti vicino alla media, significa che il modello è stabile.

DECISION TREE

L'albero decisionale (Decision Tree) è utilizzato per valutare la stabilità e l'affidabilità del modello attraverso una serie di esecuzioni (run), durante le quali vengono calcolate metriche di prestazione come l'accuratezza. Questa metodologia permette di avere una visione complessiva delle capacità predittive del modello, riducendo l'incertezza legata a una singola suddivisione del dataset.

L'albero decisionale è un modello di classificazione non parametriche che divide iterativamente i dati in base a determinate condizioni sui valori delle feature. Ogni nodo rappresenta una condizione che separa i dati, e le foglie finali indicano la classe predetta. Questo modello è particolarmente utile per la sua interpretabilità: le decisioni prese dall'albero possono essere rappresentate in forma di regole comprensibili.

Per ogni iterazione si calcola l'accuratezza, ovvero la proporzione di predizioni corrette rispetto al totale dei campioni del test set. L'accuratezza fornisce una misura generale della bontà delle predizioni del modello, ma può essere poco rappresentativa nei casi di classi sbilanciate.

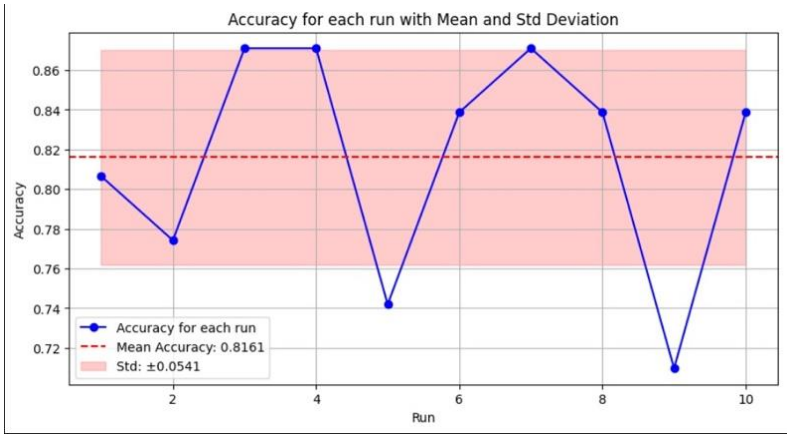
Al termine delle dieci iterazioni, il codice calcola le media e deviazione standard dell'accuratezza**:** La media dell'accuratezza fornisce una stima del rendimento medio del modello, mentre la deviazione standard indica la stabilità o variabilità dei risultati tra i vari run. Una bassa deviazione standard è indice di un modello stabile che offre prestazioni simili su diverse suddivisioni dei dati.

```
Mean Accuracy over 10 runs: 0.8161
Standard Deviation of Accuracy: 0.0541

Classification Report (last run):
              precision    recall  f1-score   support

           0       0.78      0.70      0.74        10
           1       0.86      0.90      0.88        21

    accuracy                           0.84        31
   macro avg       0.82      0.80      0.81        31
weighted avg       0.84      0.84      0.84        31
```



Mostra l'accuratezza ottenuta in ciascun run, con una linea rossa che rappresenta la media dell'accuratezza complessiva. L'area ombreggiata attorno alla media rappresenta una deviazione standard, fornendo un'indicazione della variabilità tra i run.

Questo grafico è utile per comprendere quanto sia stabile il modello: se i valori sono vicini alla linea della media, il modello si comporta in modo costante.

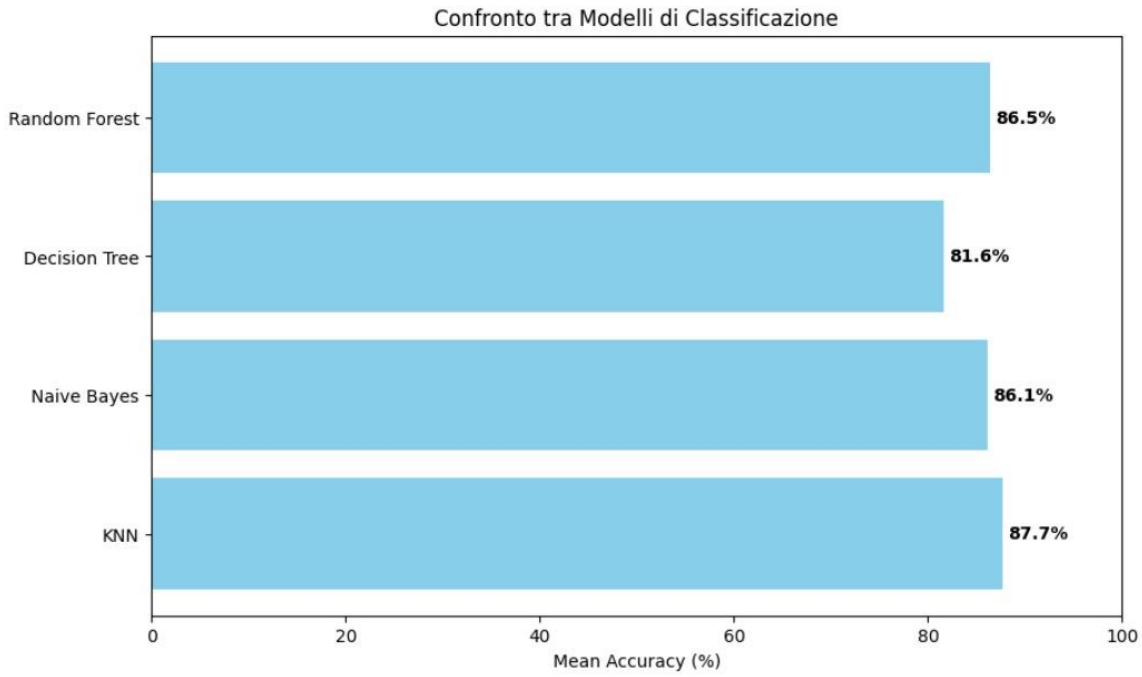
VALUTAZIONI FINALI

Il grafico finale fornisce una rappresentazione chiara delle performance dei quattro modelli di classificazione analizzati, consentendo un confronto diretto delle loro accuratezze. Tra i modelli considerati, K-Nearest Neighbors (KNN) emerge come quello con la performance migliore, raggiungendo un'accuratezza media del 87,74%. Questo lo distingue come il modello più efficace nel contesto analizzato.

Naive Bayes e Random Forest presentano risultati simili tra loro, con accuratezze medie rispettivamente dell'86,13% e dell'86,45%. Anche se leggermente inferiori rispetto a KNN, le loro prestazioni restano comunque competitive e suggeriscono che entrambi i modelli potrebbero rappresentare valide alternative a seconda del contesto o di altri fattori specifici, come la complessità computazionale o l'interpretabilità.

Il Decision Tree, invece, si colloca come il modello con la performance più bassa, con un'accuratezza media del 81,61%. Questo lo rende meno adatto rispetto agli altri modelli presi in considerazione, almeno in termini di accuratezza sul dataset utilizzato.

Questo tipo di rappresentazione grafica è particolarmente utile per facilitare il processo decisionale, poiché permette di identificare immediatamente quale modello offre le migliori prestazioni e quali invece risultano più simili tra loro. In tal modo, si ottengono informazioni essenziali per guidare la scelta del modello da implementare in produzione o da esplorare ulteriormente in analisi successive.



Un confronto tra le accuratezze iniziali e finali dei modelli ha rivelato risultati interessanti. Il modello k-Nearest Neighbors (kNN) ha mostrato un miglioramento modesto, con un incremento dell'accuratezza finale dello 0,6%. Al contrario, il modello Naive Bayes ha registrato una diminuzione dell'accuratezza, passando dal 90,32% all'86,10%, con un calo significativo del 4,22%.

Il Decision Tree, invece, ha evidenziato il miglioramento più consistente, con un aumento dell'accuratezza pari al 17,08%. Anche il modello Random Forest ha ottenuto risultati positivi, con un incremento dell'accuratezza del 5,85%. In sintesi, il Decision Tree ha dimostrato il miglior progresso tra tutti i modelli analizzati, mentre il Naive Bayes è stato l'unico modello a registrare un calo delle prestazioni.