Thyroiding : Documentazione

A.A. 2024/2025

Progetto di Ingegneria della Conoscenza

A cura di:
Giuseppe Giampietro, 774968
g.giampietro3@studenti.uniba.it

Professore del corso:
Nicola Fanizzi

GitHub del progetto: https://github.com/giuseppegmp/ICON24-25

INTRODUZIONE
Il tumore della tiroide ha origine dalla trasformazione di una ghiandola: la tiroide. Essa è posta nel collo appena sotto la cartilagine tiroidea e ha la forma di una farfalla con le due ali ai lati della laringe. Queste due ali costituiscono i lobi della tiroide, mentre la parte centrale che le congiunge è detta istmo.

La tiroide è una ghiandola endocrina: produce gli ormoni tiroidei che rilascia nel circolo sanguigno. Gli ormoni tiroidei regolano, tra le varie cose, il battito cardiaco, la temperatura corporea e soprattutto il metabolismo, ovvero la modalità con cui l'organismo usa e consuma le sostanze nutritive. Nei bambini intervengono anche nello sviluppo fisico e psichico, e la loro carenza determina gravi deficit sia di statura che cognitivi.
Essa produce gli ormoni solo se stimolata a sua volta da un altro ormone, il **TSH**, prodotto e rilasciato da un'altra ghiandola. La struttura degli ormoni tiroidei è caratterizzata dalla presenza di alcuni atomi di iodio, che è quindi un elemento fondamentale per la loro attività.

Il tumore della tiroide è abbastanza diffuso: rappresenta il 3-4% di tutti i tumori umani. Colpisce soprattutto le donne tra i 40 e i 60 anni ed è il più frequente nelle donne in questa fascia d'età. Il numero di casi di tumore della tiroide è molto aumentato negli ultimi decenni. Come confermato dal rapporto "I numeri del cancro 2023", dell'Associazione italiana di oncologia medica (AIOM) e dell'Associazione italiana registri tumori (AIRTUM), è probabile che a tale aumento dei casi contribuiscano i controlli ecografici che in precedenza non si facevano e che hanno portato a diagnosticare tumori indolenti che prima non erano scoperti.

Le donne sono più colpite degli uomini nella proporzione di 4 a 1. Tra i fattori di rischio c'è la carenza di iodio che causa il gozzo, un aumento di volume della tiroide, spesso caratterizzato da numerosi noduli benigni della ghiandola. Il gozzo può predisporre alla trasformazione maligna delle cellule.
Un altro fattore di rischio è l'esposizione a radiazioni ionizzanti: il tumore della tiroide è più comune in persone che sono state trattate per diversi motivi con radioterapia sul collo o che sono state esposte a ricadute di materiale radioattivo (esempi sono esplosioni di bombe nucleari o incidenti nelle centrali atomiche). Infine, è da sapere che si tratta di una tipologia di tumore a carattere ereditario, il che significa che la presenza della malattia in un familiare aumenta la probabilità di svilupparla anche in altri membri della famiglia.

I principali tipi di cancro della tiroide si distinguono in base alle cellule da cui hanno origine e al loro comportamento clinico. In ordine di frequenza si hanno:

- Carcinoma papillare : è la forma più comune (circa 85-90% dei casi). Ha una crescita lenta e generalmente una prognosi favorevole, e tende a diffondersi ai linfonodi del collo.
- Carcinoma follicolare : rappresenta il 10-15% dei casi, si diffonde a distanza (polmoni, ossa) e ha anch'esso una prognosi relativamente buona se trattato precocemente.
- Carcinoma midollare: circa il 3-5% dei casi, origina dalle cellule C della tiroide, produttrici di calcitonina.
- Carcinoma anaplastico : più raro degli altri (1-2%) ed è il più aggressivo, con prognosi sfavorevole.

Il segno più comune del tumore della tiroide è un nodulo isolato all'interno della ghiandola, che si sente con le dita se si tocca il collo in corrispondenza dell'organo. Non tutti i noduli tiroidei nascondono però forme di cancro, anzi nella grande maggioranza dei casi si tratta di forme benigne di crescita ghiandolare. Si stima che solo il 5-10% dei noduli tiroidei sia un tumore maligno. In rari casi, il cancro può manifestarsi all'esordio con una massa che cresce rapidamente e che può coinvolgere estesamente il collo, in corrispondenza sia della tiroide sia dei linfonodi latero-cervicali del collo. Individuato un nodulo, si misurano gli ormoni tiroidei e il TSH, per accertarne il corretto funzionamento.

Per la cura del tumore della tiroide, la chirurgia è il trattamento di prima scelta, in cui si preferisce asportare tutta la ghiandola. Tuttavia, un piccolo carcinoma papillare o follicolare può essere curato con la lobectomia, ossia l'asportazione del solo lato coinvolto. La seconda linea di trattamento consiste nel somministrare iodio radioattivo (I-131), che è una forma radioattiva dello iodio, capace di emettere radiazioni beta e gamma.
Le cellule tiroidee, sane o tumorali, assorbono lo iodio per produrre gli ormoni; dopo la tiroidectomia (rimozione della tiroide), lo iodio radioattivo si concentra nei residui della tiroide e la radiazione emessa distrugge le cellule tiroidee residue tumorali.
Attraverso la realizzazione di questo progetto si potrà essere in grado di **predire** attraverso delle misurazioni diagnostiche, la possibilità che un paziente possa essere soggetto nuovamente ad un tumore della tiroide.

REQUISITI FONDAMENTALI

Il progetto è stato sviluppato utilizzando il linguaggio Python, scelto per la sua semplicità e potenza nell'analisi dei dati e implementazione di algoritmi di apprendimento automatico. Come ambiente di sviluppo è stato utilizzato Visual Studio Code, apprezzato per la sua leggerezza e vasta gamma di estensioni dedicate alla programmazione in Python.

Sono state impiegate diverse librerie, ognuna con un ruolo specifico:

- **Pandas** : per la gestione e manipolazione dei dataset in formato tabellare, oltre che per l'analisi preliminare dei dati;
- **NumPy** : per l'esecuzione di calcoli numerici avanzati e per il supporto a strutture dati multidimensionali;
- **Matplotlib** e **Seaborn** : utilizzate per la creazione di grafici e visualizzazioni, facilitando l'interpretazione dei risultati;
- **Scikit-learn** : per l'applicazione di algoritmi di classificazione e regressione, nonché per la valutazione delle prestazioni tramite metriche standard;
- **XGBoost** : per l'implementazione di modelli di boosting particolarmente efficienti su dataset complessi;
- **PyQt5** : per la creazione di interfacce grafiche desktop, gestendo finestre, pulsanti, caselle di testo, menu e messaggi informativi, facilitando l'interazione con l'utente.

DATASET

Il dataset usato in questo progetto è stato scaricato dal sito www.kaggle.com ed è una versione modificata e migliorata del "Differentiated Thyroid Cancer Recurrence" di Joe Beach Capital.

Il dataset contiene una lista di 383 pazienti, osservati per un periodo minimo di dieci anni in un arco temporale di quindici anni.

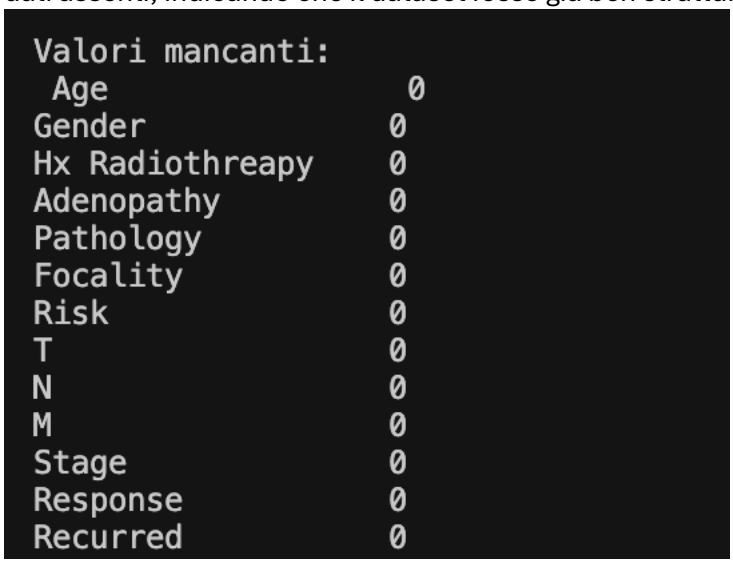
Sono state tenute in considerazione tredici **features** clinico-patologiche per predire il potenziale di recidiva del tumore. Esse sono le seguenti:

- **Age** : età del paziente.
- **Gender** : genere del paziente.
- **Hx Radiotherapy** : presenza di trattamenti radioterapico (sì o no).
- **Adenopathy** : coinvolgimento dei linfonodi (sì o no).
- **Pathology** : tipo di tumore tiroideo (es. micropapillare).
- **Focality** : localizzazione del tumore, uni-focale o multifocale.
- **Risk** : classificazione del rischio (basso, intermedio o alto).
- **T** : dimensione/classificazione del tumore (T1, T2, ecc.)
- **N** : classificazione linfonodale (N0, N1, ecc.).
- **M :** classificazione delle metastasi (M0 = assenti, M1 = presenti).
- **Stage** : stadio complessivo del tumore (stadio I, II, III, IV).
- **Response** : risposta al trattamento. Esito della terapia (es. eccellente, indeterminata, ecc.).
- **Recurred** : indicazione se il tumore è recidivato.

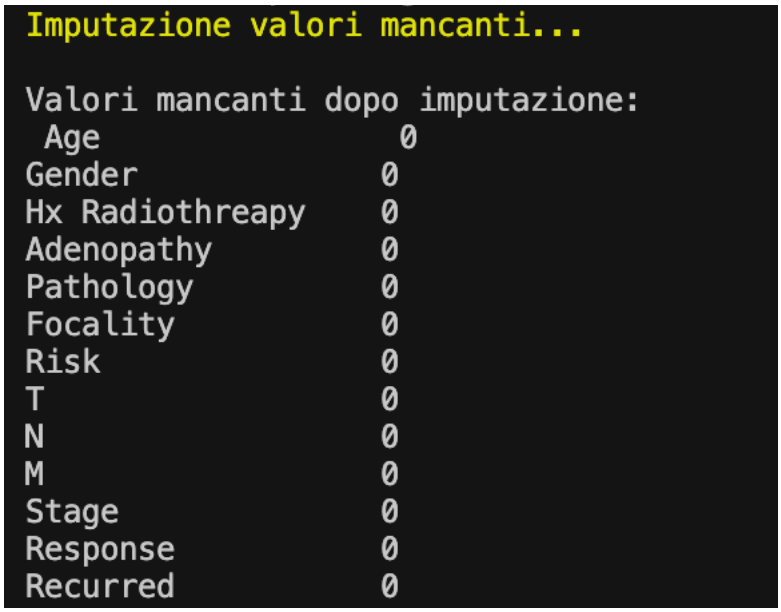
Successivamente si effettua un ragionamento basato su precisi step logici del *preprocessing* e *analisi dei dati*:

1. Pulizia dei dati (gestione dei valori nulli e anomali);
2. Trattamento dei dati mancanti;
3. Controllo e bilanciamento delle classi (*oversampling*).

Durante la fase di pre-elaborazione dei dati è stata condotta una verifica accurata della qualità del dataset, con particolare attenzione alla presenza di valori mancanti. Il controllo iniziale attraverso la creazione di una tabella ha mostrato che tutte le variabili risultavano complete e non presentavano dati assenti, indicando che il dataset fosse già ben strutturato e pronto per l'analisi.



Nonostante ciò, per rendere la pipeline più robusta e in grado di gestire eventuali casi futuri in cui i dati potessero risultare incompleti, è stata comunque predisposta una procedura di imputazione. In questo processo, qualora si fossero riscontrati valori mancanti, le variabili numeriche sarebbero state completate utilizzando la media dei valori disponibili, mentre le variabili categoriali sarebbero state completate usando il valore più frequente (moda). Tale strategia è stata adottata con l'intento di preservare la coerenza interna del dataset e ridurre il rischio che eventuali anomalie potessero compromettere le fasi successive di addestramento dei modelli. Al termine di questa procedura, è stata effettuata una seconda verifica, che ha confermato l'assenza di valori mancanti residui, garantendo così la piena affidabilità del dataset per le successive attività di analisi e modellazione.



Come si evince dalle immagini, i valori sono tutti pari a zero in entrambe le tabelle, poiché in questo caso le feature non contenevano valori nulli già in partenza.

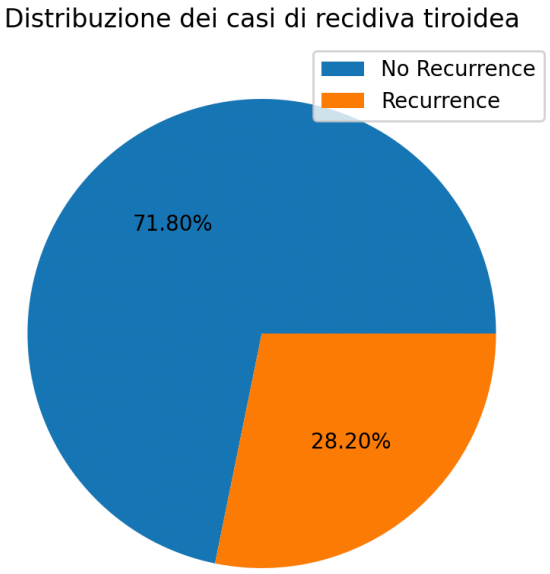
Successivamente si è controllata la distribuzione della variabile target (Recurred), la quale indica se il tumore è recidivato. Inizialmente c'è uno squilibrio, ossia 275 senza ricaduta e 108 con ricaduta.

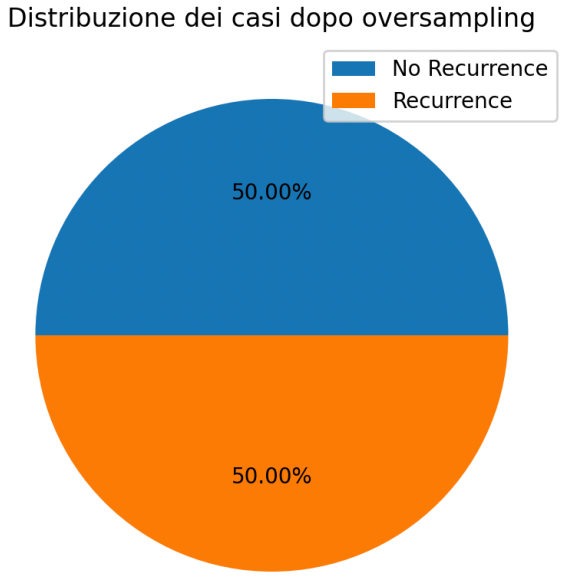
```
Controllo del bilanciamento delle classi
No Recurrence:  275 (% 71.80)
Recurrence:  108 (% 28.20)
```

Per risolvere questa distribuzione sbilanciata della variabile target, si applica l'**oversampling** della classe minoritaria, ossia un processo che consiste nel generare nuovi dati simili alla classe meno rappresentata, con lo scopo di colmare il divario tra le due classi portandole entrambi allo stesso numero di elementi.
Esistono varie tecniche di oversampling, ma quella adottata in questo progetto è quella più semplice, ossia la **Random Oversampling** (ROS), ossia il processo di duplicazione di campioni casuali della classe minoritaria fino a eguagliare la numerosità della classe maggioritaria. Questa tecnica presenta diversi vantaggi rispetto ad altri metodi come la sua facilità di implementazione e la complessità a breve termine a causa di un algoritmo semplice.
Dopo il bilanciamento i valori sono i seguenti:

```
Valori dopo oversampling:
No Recurrence:  275 (% 50.00)
Recurrence:  275 (% 50.00)
```

Per semplicità di visualizzazione dei dati, sono state creati anche dei diagrammi a torta che mostrano il bilanciamento delle classi prima e dopo l'oversampling.





Questo passaggio è fondamentale per contrastare lo squilibrio all'interno di un set di dati. Un set di dati sbilanciato è definito come un set di dati in cui una classe è fortemente sottorappresentata nel set di dati rispetto alla popolazione reale, creando distorsioni non intenzionali.
In breve, con un dataset sbilanciato, i modelli tendono a concentrarsi quasi esclusivamente sulla classe più rappresentata. In questo modo possono raggiungere un'accuratezza apparentemente elevata, ma senza sviluppare una reale capacità predittiva. Il bilanciamento delle classi permette al modello di apprendere in modo equilibrato da entrambe le categorie, migliorando la qualità e affidabilità delle previsioni.

APPRENDIMENTO SUPERVISIONATO
Prima di addestrare i modelli, viene effettuata la standardizzazione dei dati, ossia una tecnica fondamentale per migliorare le prestazioni degli algoritmi di machine learning.
Questo processo rende più facilmente confrontabile una variabile quantitativa con altre variabili quantitative. La trasformazione crea una variabile con media pari a 0 e deviazione standard pari a 1.
È una operazione importante poiché permette di confrontare due valori che hanno magari unità di misure differenti evitando che variabili con scale diverse influenzino sproporzionatamente il modello.
La standardizzazione consiste nel sottrarre la media da ogni valore del vettore e dividere la differenza per la deviazione standard.
Successivamente viene eseguita un'ottimizzazione degli iperparametri, l'ambito che si occupa della scelta di parametri ottimali per un modello di machine learning.
Fare tuning degli iperparametri significa trovare il set ideale per il modello di machine learning. Tra le varie tecniche automatiche per la scelta di iperparametri per questo progetto è stata scelta la **Random Search** in cui viene fissato un numero di combinazioni randomiche. Viene scelto il numero di combinazioni da provare, e in base a questo numero la Random Search andrà a pescare a caso un valore del set di valori per ogni iperparametro. Addestrerà un modello con queste combinazioni e lo valuterà, scegliendo poi il modello con la combinazione migliore di iperparametri.

Attraverso RandomizedSearchCV, oltre alla Random Search, viene eseguita la cross-validation, che viene usata durante l'addestramento del modello. Quello più comunemente utilizzato è il K-fold cross-validation.
In pratica, il dataset viene suddiviso in *k* gruppi (folds). Il modello viene allenato k volte, ogni volta usando un fold come test e i restanti k-1 per il training.
Nel progetto le fold sono fissate a k=5, il che significa che il dataset è diviso in cinque parti, quattro per l'addestramento e una per la validazione. Ad ogni iterazione verranno registrate le prestazioni del modello secondo delle metriche precise.

Come algoritmi di classificazione sono stati scelti i seguenti modelli:

- **Random Forest**
- **K-NN**
- **XGBoost**
- **Decision Tree**

Mentre, per valutare le performance di ciascun classificatore sono state considerate le seguenti metriche:

- **Accuracy** : indica quante volte il modello ha correttamente classificato un item nel nostro dataset rispetto al totale. Percentuale di previsioni corrette.
- **Precision** : rapporto tra le istanze previste correttamente e istanze totali nel dataset. Utile per valutare la correttezza complessiva di un modello. Quantifica la frequenza con cui il modello effettua previsioni corrette su tutte le previsioni effettuate.
- **Recall** : qui si tiene conto dei falsi negativi invece dei falsi positivi.
- **F1-Score** : media armonica di precision e recall, utile in caso di classi sbilanciate. Combina le due metriche in un unico valore. Se lo score è elevato, c'è un buon equilibrio tra P e R, quindi il modello sarà efficace.

Un modello ad alta precisione è conservativo: non riconosce sempre la classe correttamente, ma quando lo fa, si è sicuri che la sua risposta sia corretta.
Uno ad alto richiamo è liberale: riconosce una classe molto più spesso, ma incluse anche molti falsi positivi.
Queste due metriche sono complementari: se aumentiamo una, l'altra deve diminuire. Si tratta del precision/recall trade-off.
Le metriche sono utili per confrontare i diversi modelli e selezionare quello con le migliori performance per un dataset specifico, per esempio, se un modello deve ridurre al minimo i falsi positivi, la Precision diventa una metrica critica per la valutazione; viceversa, per il Recall.

Analizziamo Step-by-Step ciascun algoritmo di classificazione con i relativi risultati delle metriche.

Random Forest

L'algoritmo è un modello *ensemble,* cioè che al proprio interno mette insieme altri modelli più semplici.
È basato sull'addestramento di N alberi decisionali (decision tree), ognuno dei quali effettua una classificazione per ogni esempio. Quando tutti gli alberi (o più precisamente tutta la foresta) hanno classificato l'esempio, si effettua una conta su qual è stata la classe maggiormente stimata e la si assume come predizione della foresta.
Gli iperparametri scelti per questo classificatore sono:

- *n_estimators* : numero di alberi decisionali che verranno creati nella foresta. Più alberi ci sono, più accuratezza ci sarà, ma comporta anche un tempo di calcolo maggiore.
- *Max_depth* : profondità massima di ogni albero. Alberi con profondità maggiore catturano relazioni più complesse nei dati, ma sono anche più inclini all'overfitting (adattamento eccessivo ai dati di addestramento).

- *Min_samples_leaf* : numero minimo di campioni che devono essere presenti in un nodo foglia. Valori più alti possono rendere il modello più robusto.
- *Min_samples_split* : numero minimo di campioni necessari per dividere un nodo interno. Un valore più alto può rendere il modello meno sensibile al rumore nei dati, ma anche a una perdita di informazioni.
- *Max_features*: numero massimo di caratteristiche che verranno considerate quando si cerca la migliore divisione in un nodo.

Questo è il risultato ottenuto:

```
Addestramento modelli supervisionati...
Addestramento Random Forest...
Random Forest - Migliori parametri: {'n_estimators': 300, 'min_samples_split': 2, 'min_samples_leaf': 4, 'max_features': 'log2', 'max_depth': None}
Random Forest - Metriche (CV):
  Accuracy: 0.888
  Precision: 0.883
  Recall: 0.898
  F1-score: 0.889
Random Forest - Deviazione Standard (CV):
  Accuracy: ±0.018
  Precision: ±0.030
  Recall: ±0.042
  F1-score: ±0.019

Classification Report - Random Forest (Test):
                    precision    recall  f1-score   support

      No Recurrence       0.79      0.81      0.80        69
         Recurrence       0.81      0.78      0.79        69

           accuracy                           0.80       138
```

Innanzitutto, la RandomizedSearchCV fa la selezione dei parametri ottimali per la Random Forest. In seguito, elenca le metriche in Cross-Validation, garantendo ottime prestazioni in CV, con particolare forza nel recall (cioè individua bene i casi di recidiva).
Inoltre, vengono elencati i valori della deviazione standard per ogni metrica, la quale informa quanto bisogna fidarsi dei risultati ottenuti.
Infine, con il classification report si vedono i risultati sul test set, cioè dati mai visti in addestramento: classe "no recurrence" (pazienti senza recidiva) hanno precisione 0.79, richiamo 0.81; mentre "recurrence" rispettivamente 0.81 e 0.78 con accuracy finale 0.80, il che significa che il modello non è sbilanciato: predice bene entrambe le classi.

K-NN

L'algoritmo K-NN funziona in base a un principio di somiglianza, tipicamente misurato da una metrica di distanza, generalmente la distanza euclidea, ma ne esistono diverse, a seconda del set di dati.
L'idea è che dato un nuovo punto da classificare, l'algoritmo calcola la distanza tra questo punto e tutti quelli già presenti nel dataset, per poi selezionare i k esempi più vicini e assegna al nuovo punto la classe che compare più frequentemente tra questi.
Sono stati scelti i seguenti iperparametri per questo classificatore:

- *N_neighbors* : numero di vicini da considerare per classificare un nuovo punto
- *Weights* : peso da attribuire ai vicini ("uniform" – tutti i vicini hanno lo stesso peso; "distance" – i vicini più vicini al nuovo punto hanno un peso maggiore, cioè sono più influenti).
- *P* : tipo di distanza da usare ("manhattan", "euclidea", "minkowski").
- *Leaf_size* : numero massimo di punti che un nodo della struttura può contenere.

Questo è il risultato ottenuto:

```
Addestramento k-NN...
k-NN - Migliori parametri: {'weights': 'uniform', 'p': 2, 'n_neighbors': 11, 'leaf_size': 30}
k-NN - Metriche (CV):
  Accuracy: 0.891
  Precision: 0.877
  Recall: 0.912
  F1-score: 0.893
k-NN - Deviazione Standard (CV):
  Accuracy: ±0.020
  Precision: ±0.035
  Recall: ±0.033
  F1-score: ±0.019

Classification Report - k-NN (Test):
                  precision    recall  f1-score   support

      No Recurrence       0.79      0.81      0.80        69
         Recurrence       0.81      0.78      0.79        69

           accuracy                           0.80       138
```

Si notano innanzitutto i parametri ottimali scelti per il k-NN. Poi, vengono elencate le metriche con la Cross-Validation e la sua deviazione standard. Si noti come in CV, il modello si comporti molto bene: alta capacità di riconoscere la recidiva senza perdere troppa precisione.
Sul test set le metriche delle due classi "recurrence" e "no recurrence" sono abbastanza bilanciate, segnale molto importante, specialmente in ambito medico, perché questo vuol dire che il modello non ignora i pazienti con recidiva.

XGBoost

L'XGBoost (eXtreme Gradient Boost) è un algoritmo di apprendimento supervisionato basato su ensemble di alberi decisionali. Funziona costruendo tanti alberi in sequenza, dove ogni nuovo albero corregge gli errori commessi da quelli precedenti. A differenza del bagging (come il Random Forest), il boosting è sequenziale: ogni passo aggiunge un modello che si concentra sugli esempi difficili da classificare.
Quando un modello fa una previsione, c'è una differenza rispetto al valore reale. Questa differenza si chiama errore e viene misurata da una funzione di perdita come la log-loss. L'algoritmo usa dei gradienti, ossia la derivata della funzione di perdita rispetto alla previsione del modello, cioè dice quanto e in che direzione bisogna correggere la previsione per ridurre l'errore. In XGBoost ogni nuovo albero viene addestrato per predire i propri gradienti lasciati dal modello fino a quel punto.
Per questo modello sono stati scelti i seguenti iperparametri:

- *Max_depth* : profondità massima degli alberi
- *N_estimators* : numero di alberi da costruire
- *Learning_rate* : peso dato a ciascun nuovo albero
- *Subsample* : percentuale di dati usata per costruire ogni albero (es. 0.8, ogni albero vede solo 80% dei dati scelti a caso)
- *Colsample_bytree* : percentuale di feature usate per ogni albero.

Il risultato ottenuto per questo modello è il seguente:

```
Addestramento XGBoost...
XGBoost - Migliori parametri: {'subsample': 0.8, 'n_estimators': 200, 'max_depth': 3, 'learning_rate': 0.01, 'colsample_bytree': 0.8}
XGBoost - Metriche (CV):
  Accuracy: 0.898
  Precision: 0.889
  Recall: 0.912
  F1-score: 0.900
XGBoost - Deviazione Standard (CV):
  Accuracy: ±0.016
  Precision: ±0.031
  Recall: ±0.025
  F1-score: ±0.014

Classification Report - XGBoost (Test):
                  precision    recall  f1-score   support

      No Recurrence       0.79      0.81      0.80        69
         Recurrence       0.81      0.78      0.79        69

           accuracy                           0.80       138
```

Il modello con i migliori parametri trovati ha mostrato ottime prestazioni in cross-validation, con un accuracy pari quali al 90% e valori molto bilanciati di precision, recall e F1-score, segno che riesce a riconoscere bene sia i casi di recidiva sia quelli di non recidiva. La deviazione standard molto bassa indica che i risultati sono stabili e poco variabili tra le diverse fold della cross-validation. Tuttavia, quando testato sul set di dati mai visto, l'accuratezza scende all'80%. Questo è normale perché in test viene valutato su dati più difficili.

Decision Tree

Questo è un algoritmo di apprendimento supervisionato con struttura gerarchica ad albero che consiste in un nodo radice, rami, nodi interni e nodi foglia. Si inizia con un nodo radice con rami in uscita alimentando i nodi interni, noti anche come nodi decisionali. Sulla base delle funzionalità disponibili, entrambi i tipi di nodo conducono valutazioni per formare sottoinsiemi omogenei, indicati da nodi foglia. I nodi foglia sono tutti i possibili target nel set di dati.
Per questo modello sono stati scelti i seguenti iperparametri:

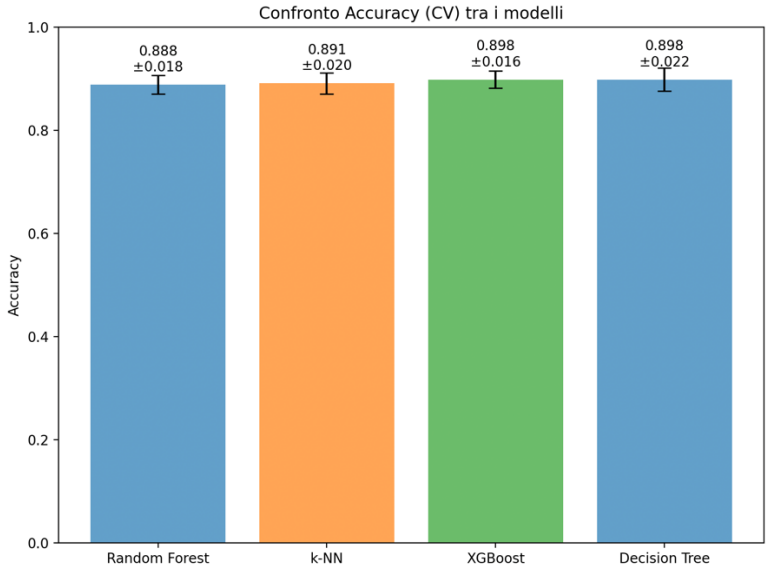
- Max_depth : massima profondità dell'albero.
- Min_sample_split : numero minimo di campioni richiesti per dividere un nodo interno.
- Min_sample_leaf : numero minimo di campioni richiesti per un nodo foglia. Garantisce che ogni nodo foglia abbia un numero minimo di esempi, stabilizzando le predizioni.
- Max_features : numero massimo di feature da considerare per ogni split.
- Criterion : misura usata per valutare la qualità di uno split. ("gini" – misura impurità , "entropy" – misura informativa).

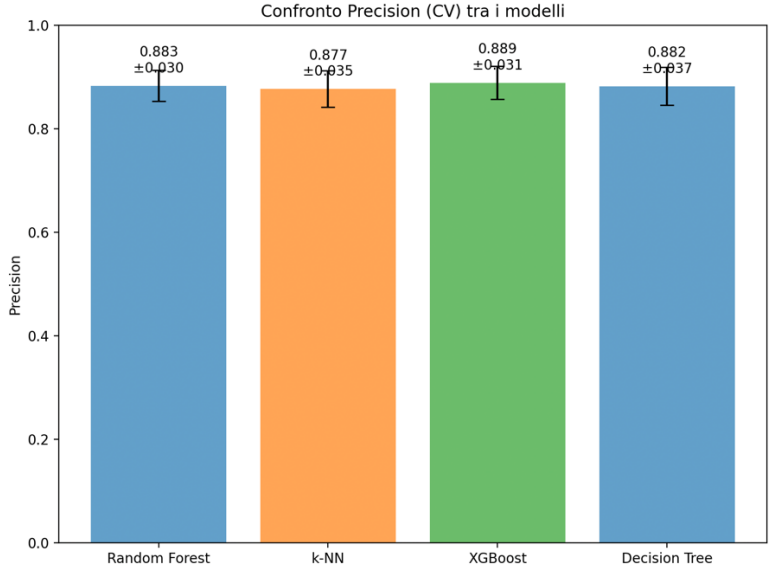
Questo è il risultato ottenuto:

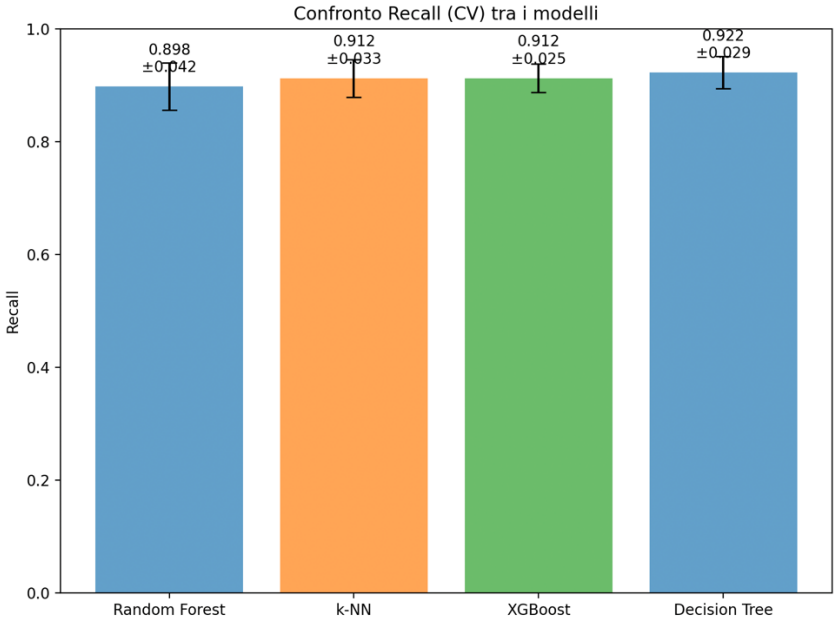
```
Addestramento Decision Tree...
Decision Tree - Migliori parametri: {'min_samples_split': 10, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': 3, 'criterion': 'entropy'}
Decision Tree - Metriche (CV):
  Accuracy: 0.898
  Precision: 0.882
  Recall: 0.922
  F1-score: 0.901
Decision Tree - Deviazione Standard (CV):
  Accuracy: ±0.022
  Precision: ±0.037
  Recall: ±0.029
  F1-score: ±0.020

Classification Report - Decision Tree (Test):
                  precision    recall  f1-score   support

      No Recurrence       0.79      0.80      0.79        69
         Recurrence       0.79      0.78      0.79        69

           accuracy                           0.79       138
```

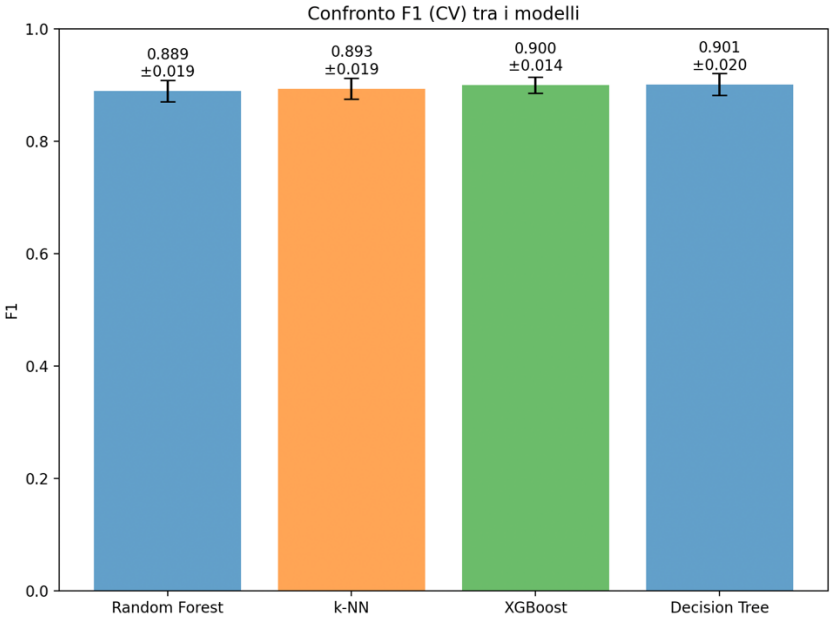
Questo mostra che il modello dopo aver trovato i migliori iperparametri, durante la cross-validation, esso ha raggiunto una accuracy media pari quasi a 0.9, con prestazioni complessive buone e stabilità. Tuttavia, sul test set, le metriche scendono leggermente, suggerendo una certa perdita di generalizzazione rispetto alla cross- validation.

Successivamente sono state create degli istogrammi per ogni metrica rappresentanti i risultati di ogni modello.







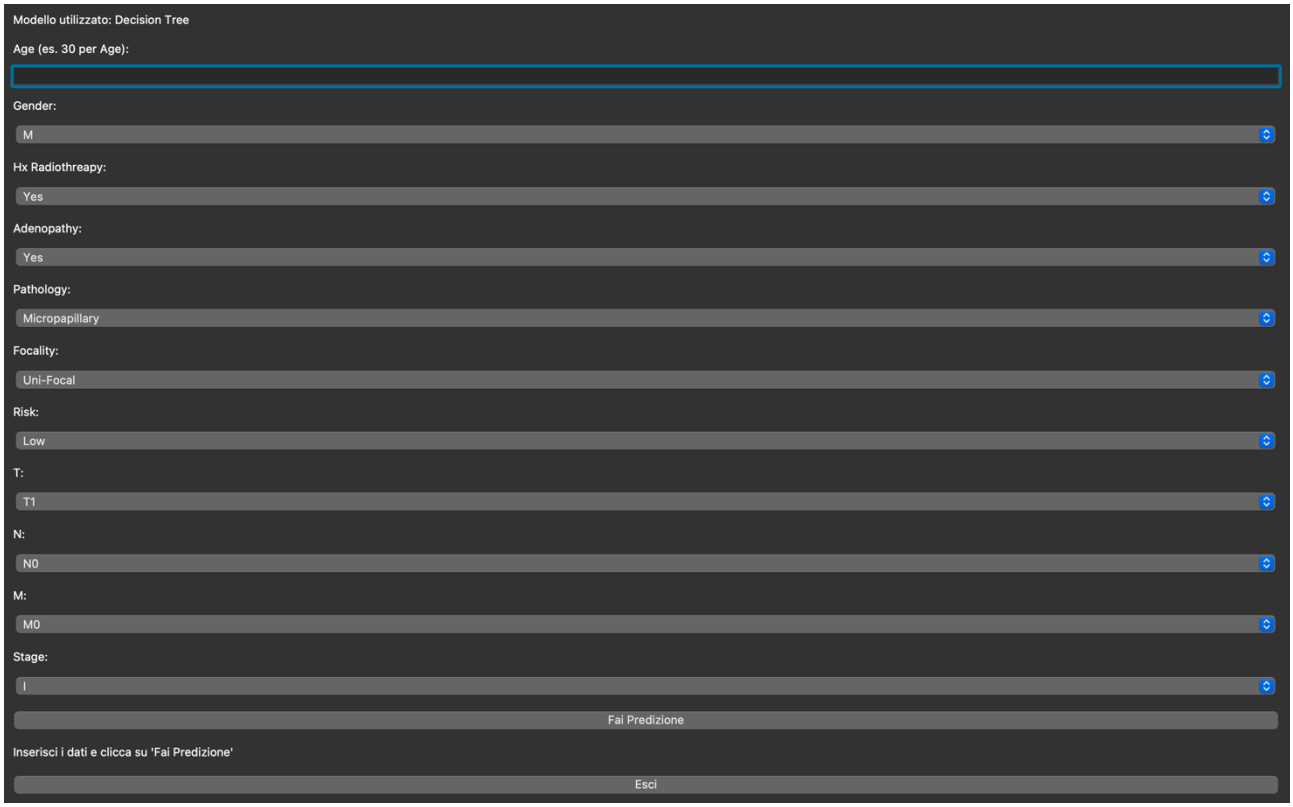


Dai dati analizzati, si evince come il Decision Tree sia il migliore perché tra tutti ha ottenuto il più alto F1-score medio durante la cross-validation (0.901). È una metrica fondamentale se si vuole bilanciare precision e recall, cioè quando sia gli errori di falsi positivi sia quelli di falsi negativi hanno rilevanza. In questo caso il Decision Tree ha mostrato la migliore accuratezza delle predizioni e capacità di catturare entrambe le classi (Recurrence e No Recurrence) durante la CV, anche se sul test set le prestazioni scendono leggermente.

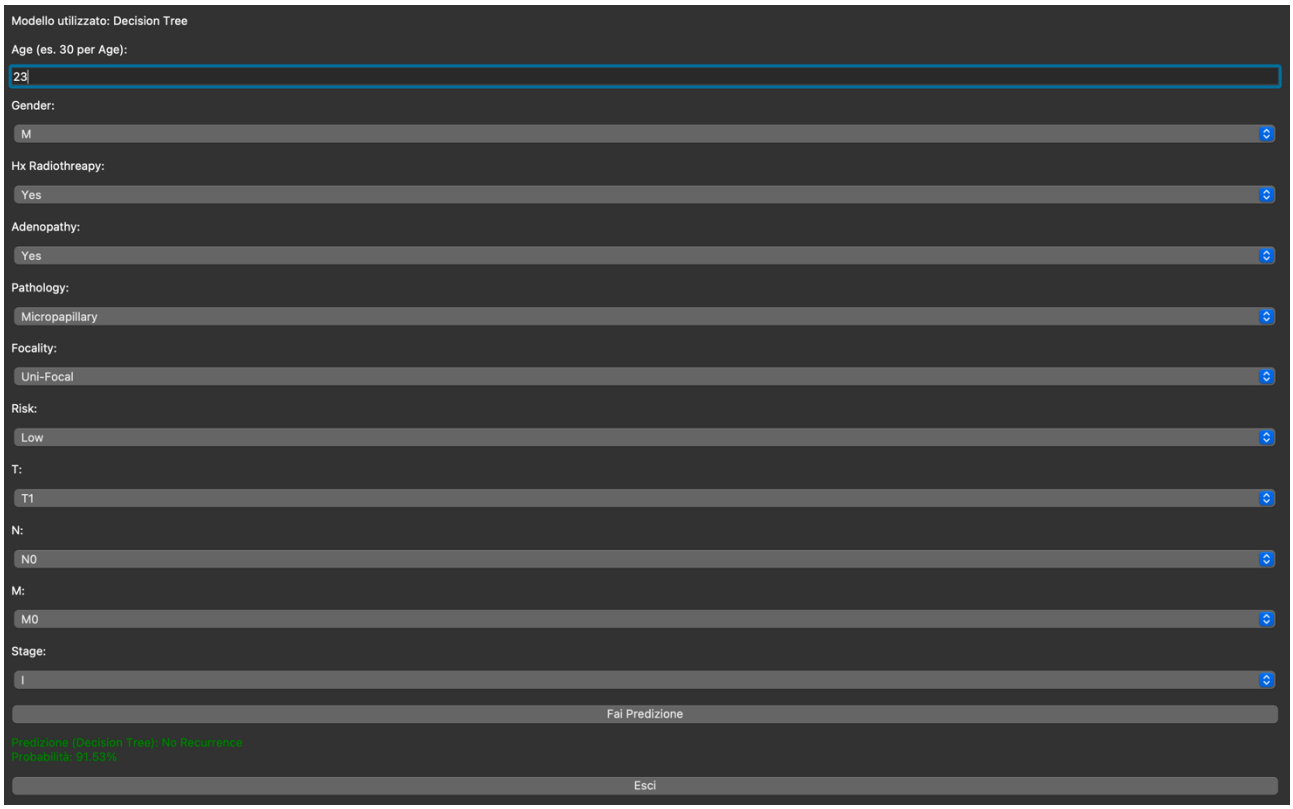
MODALITA' D'USO E INTERFACCIA

Il progetto è stato sviluppato per permettere l'inserimento di dati specifici relativi al tumore della tiroide, seguendo specifiche feature come l'età, genere, caratteristiche patologiche, stadiazione e così via. Questi dati possono essere facilmente inseriti tramite una interfaccia grafica intuitiva, studiata per guidare l'utente nella compilazione dei campi corretti, riducendo errori e rendendo il processo di previsione della recidiva semplice e immediato.

Schermata principale di inserimento dati:



Ci sono solo due tasti presenti per questa interfaccia progettata, ossia "fai predizione", che eseguirà la predizione di recidiva del tumore sul paziente a seconda dei dati inseriti, oppure "esci" che permette di abbandonare il programma in qualsiasi momento.



In questo esempio sono stati inseriti dei dati casuali per verificare il corretto funzionamento del modello. Come si può notare, in fondo alla schermata, dopo aver cliccato sul pulsante “fai predizione”, compare il risultato della predizione, la probabilità di accuratezza e il tipo di modello utilizzato.

CONCLUSIONI E SVILUPPI FUTURI

Questo progetto ha permesso di sviluppare un modello predittivo efficace, che attraverso l’inserimento di parametri clinici essenziali legati al tumore della tiroide, può supportare l’identificazione precoce del rischio di recidiva.

In futuro, il progetto potrebbe essere arricchito da un’interfaccia grafica ancora più avanzata, capace di guidare l’utente nell’inserimento dei dati e nella comprensione del risultato, offrendo anche supporti per diverse lingue.

Inoltre, si potrebbe integrare funzionalità di monitoraggio nel tempo, consentendo di salvare e confrontare le previsioni per analizzare l’evoluzione dello stato clinico del paziente.

RIFERIMENTI BIBLIOGRAFICI

•Descrizione del dataset: https://www.kaggle.com/datasets/moynul75/filtered-thyroid-data

•Materiale scientifico informativo sulla tiroide e tumore della tiroide: https://www.airc.it/cancro/informazioni-tumori/guida-ai-tumori/tumore-della-tiroide