UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

SISTEMI INTELLIGENTI PER LA COMUNICAZIONE DIGITALE

A.A. 2024/2025

DOCENTI: GABRIELLA CASALINO, GIANLUCA ZAZA

A CURA DI: IVONE DANILO - D.IVONE1@STUDENTI.UNIBA.IT
MAT: 735245

1. INTRODUZIONE

L'obiettivo di questo elaborato è presentare in modo dettagliato la progettazione e l'implementazione di una pipeline di **apprendimento supervisionato** per la **classificazione delle malattie cardiache,** utilizzando il dataset *UCI Heart Disease - subset Cleveland,* e discutendo infine i risultati ottenuti. L'interesse è duplice: in primo luogo, desideriamo valutare quali tecniche di **preprocessing** e quali **modelli di classificazione** risultino più efficaci per il problema affrontato, per poi documentare il flusso di lavoro (riproducibile) che include validazione incrociata, ricerca di iperparametri e valutazione tramite test hold-out.

Nel concreto, si vuole predire la **presenza (1) o assenza (0) di malattie cardiache,** binarizzando l'attributo '*num*'. Osserveremo come vi è uno **sbilanciamento moderato** delle classi e per questo motivo, useremo come metrica di riferimento la *F1*, affiancata da *Accuracy, Precision, Recall, ROC-AUC e PR-AUC*.

2. DESCRIZIONE DEL PROBLEMA

Il sopracitato dataset contiene esempi circa le condizioni cliniche di pazienti che sono (o possono essere) affetti da malattie cardiache. Data l'importanza clinica di ridurre i **falsi negativi** (pazienti malati classificati come sani) dovremo porre grande attenzione al **Recall** e al compromesso **Precision-Recall** indicato dalla metrica **F1.**

Il target del dataset '*num*' può essere binarizzato in modo tale che se {'*num*'>0} il paziente verrà etichettato come **affetto da una malattia**, viceversa sarà ritenuto **sano.**

Ipotesi operative:

- I pattern contenuti all'interno dei dati contengono discriminanti sufficienti per distinguere positivi e negativi.

- È essenziale una fase di preprocessing integrata nella **pipeline di addestramento** per evitare il *leakage* per migliorare la stabilità del processo.

3. DESCRIZIONE DEGLI ALGORITMI

In questa sezione analizziamo quali algoritmi di **classificazione supervisionata** sono stati messi a confronto, per affrontare il problema descritto.

3.1. K-NEAREST NEIGHBORS (KNN)

Il KNN, detto anche *lazy learner,* risulta essere semplice da **interpretare e adattare**. Quest'ultimo consente di predire le etichette di classe di nuovi casi usando le istanze di training: le predizioni sono ottenute mediante un ***majority vote*** o tramite la **media** delle classi target.

La procedura richiede la scelta di un ***k,*** numero di "vicini", dopodiché viene calcolata la distanza tra il nuovo punto inserito e i *k* vicini, si considera poi quale è la classe target, dei vicini, che compare più volte (*majority vote*) e si assegna infine il nuovo punto a questa classe. Questo algoritmo risulta **poco interpretabile** ma è **poco scalabile ed è incline all'overfitting.**

3.2. DECISION TREE (DT)

L'albero di decisione permette di creare delle strutture **gerarchiche.** Offre **l'interpretabile e supporta dati numerici e categorici**. Questi risultano però poco stabili e **inclini all'overfitting** se non regolarizzato.

3.3. RANDOM FOREST (RF)

Considerabile come un insieme di alberi decisionali. È tipicamente **robusto** perché è costruito combinando DT, che sono sistemi deboli, per costruire un modello più affidabile. Ottimo quando abbiamo **molti dati** e molti **rumori/outlier** poiché è poco sensibile a questi ultimi, inoltre risulta meno suscettibile **all'overfitting.** Chiaramente risulta meno interpretabile di un singolo albero ed è un algoritmo più **lento.**

4. DATI

Il dataset utilizzato *UCI Machine Learning Repository – Heart Disease, subset Cleveland* è reperibile al seguente link: (https://archive.ics.uci.edu/dataset/45/heart+disease).

Di seguito viene presentata una tabella contenente i 14 attributi più significativi del dataset:

Attributo	Ruolo	Tipo	Significato	Codifica/Note
age	Feature	Integer	Età del paziente	
sex	Feature	Categorical	Sesso del paziente	0 = femmina; 1 = maschio;
cp	Feature	Categorical	Tipo di dolore toracico	1 = angina tipica; 2 = angina atipica; 3 = dolore non anginoso; 4 = asintomatico;
trestbps	Feature	Integer	Pressione a riposo (mmHg)	
chol	Feature	Integer	Colesterolo sierico	
fbs	Feature	Categorical	Glicemia a digiuno	
restecg	Feature	Categorical	ECG a riposo	0 = normale; 1 = anomalia ST-T; 2 = ipertrofia ventricolare sinistra;
thalach	Feature	Integer	Frequenza cardiaca massima raggiunta	**Tipicamente più basso nei positivi**
exang	Feature	Categorical	Angina da sforzo	0 = No; 1 = Si;
oldpeack	Feature	Integer	Depressione ST (servizio vs riposo)	**Tipicamente più alto nei positivi**
slope	Feature	Categorical		
ca	Feature	Integer	N. Vasi coronarici colorati (da una fluoroscopia)	0-3

Attributo	Ruolo	Tipo	Significato	Codifica/Note
thal	Feature	Categorical	Stato talassemia	3 = normale; 6 = difetto fisso; 7 = difetto reversibile;
num	Target	Integer	Diagnosi	0 = assente; 1-4 = presenza di malattia;

Tabella 1: Attributi significativi del dataset

4.1. EDA

Dalla fase di esplorazione dei dati emergono le seguenti informazioni:

4.1.1. DIMENSIONE

Il dataset contiene 303 osservazioni e 15 colonne (303 righe x 15 colonne) incluso il target binarizzato.

4.1.2. TARGET

Come precedentemente descritto utilizziamo l'attributo diagnostico (target) '*num*', in particolare modo l'attributo può assumere valori interi non negativi tra 0 e 4 (inclusi) definiamo quindi:

$$target = \begin{cases} 0 & \text{se num = 0} \\ 1 & textsenum > 0 \end{cases}$$

4.1.3. BILANCIAMENTO

La distribuzione risultante è **164 negativi (num = 0) e 139 positivi (num > 0)** con una prevalenza ≈ 139/303 ≈ 0.459 ≈ 45,9%. Da questo risulta che le classi individuate **sono moderatamente sbilanciate,** per questo motivo adottiamo **F1** come metrica principale (integrata con **PR-AUC** per valutare il trade-off precision/recall).

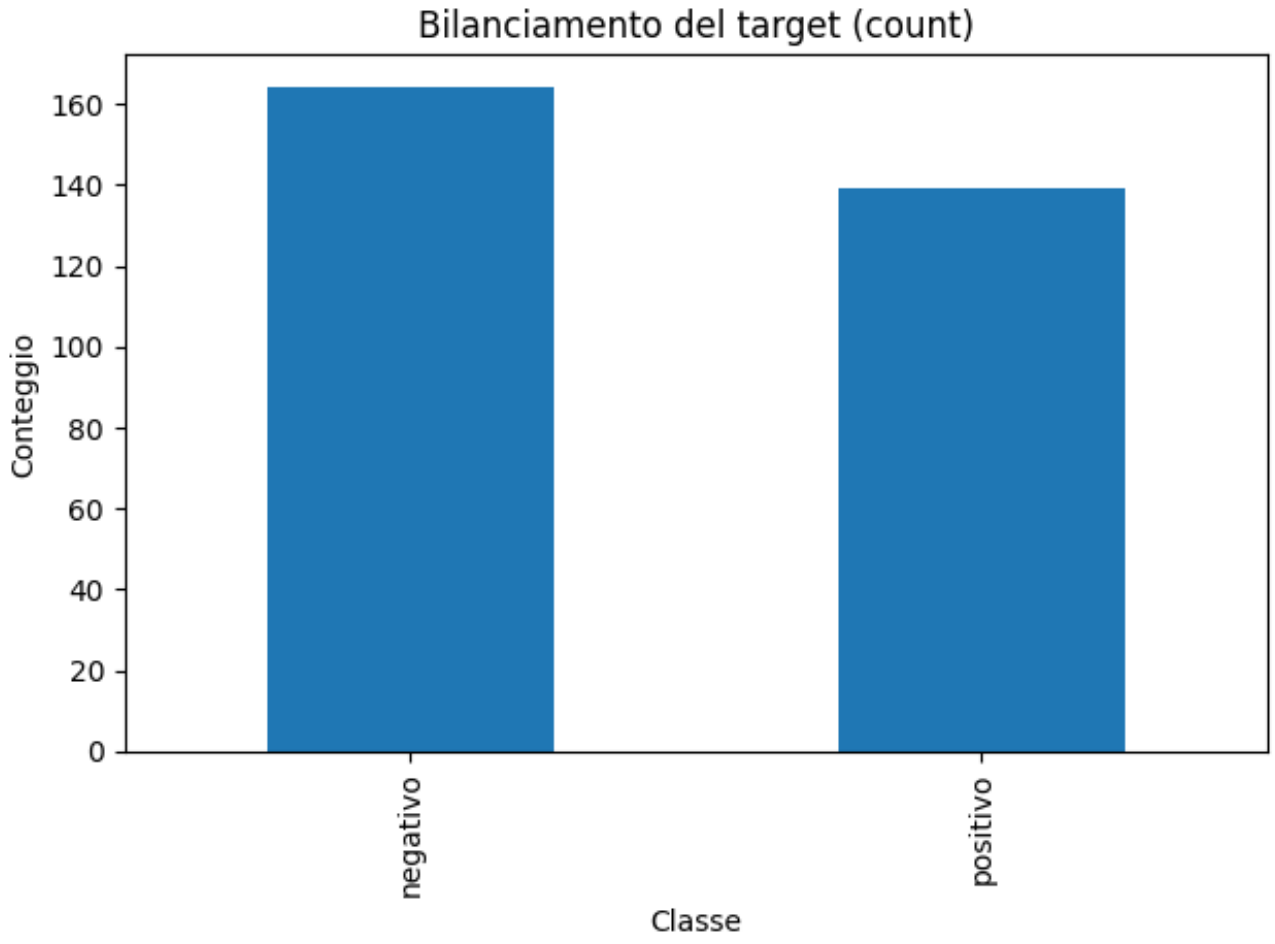

Figura 1: 164 negativi (0), 139 positivi (1), prevalenza 45.9%

4.1.4. DATI MANCANTI

Risultano **pochi e localizzati**, in particolare modo notiamo che l'attributo '***ca***' presenta 4 *missing values* (rappresentante ~1,3%) e che '**thal**' ne presenta solo 2 (rappresentante ~0,7%). Questo consente di adottare una **strategia di imputazione semplice** (mediana per i valori numerici; valore più frequente per gli attributi categorici) evitando cosi di introdurre distorsioni significative. Durante la codifica, gli attributi categorici sono trattati con **One-Hot Encoding** (*handle_unknown = "ignore"*), in questo modo categorie rare o mai osservate nella fase di addestramento vengono gestite in sicurezza.

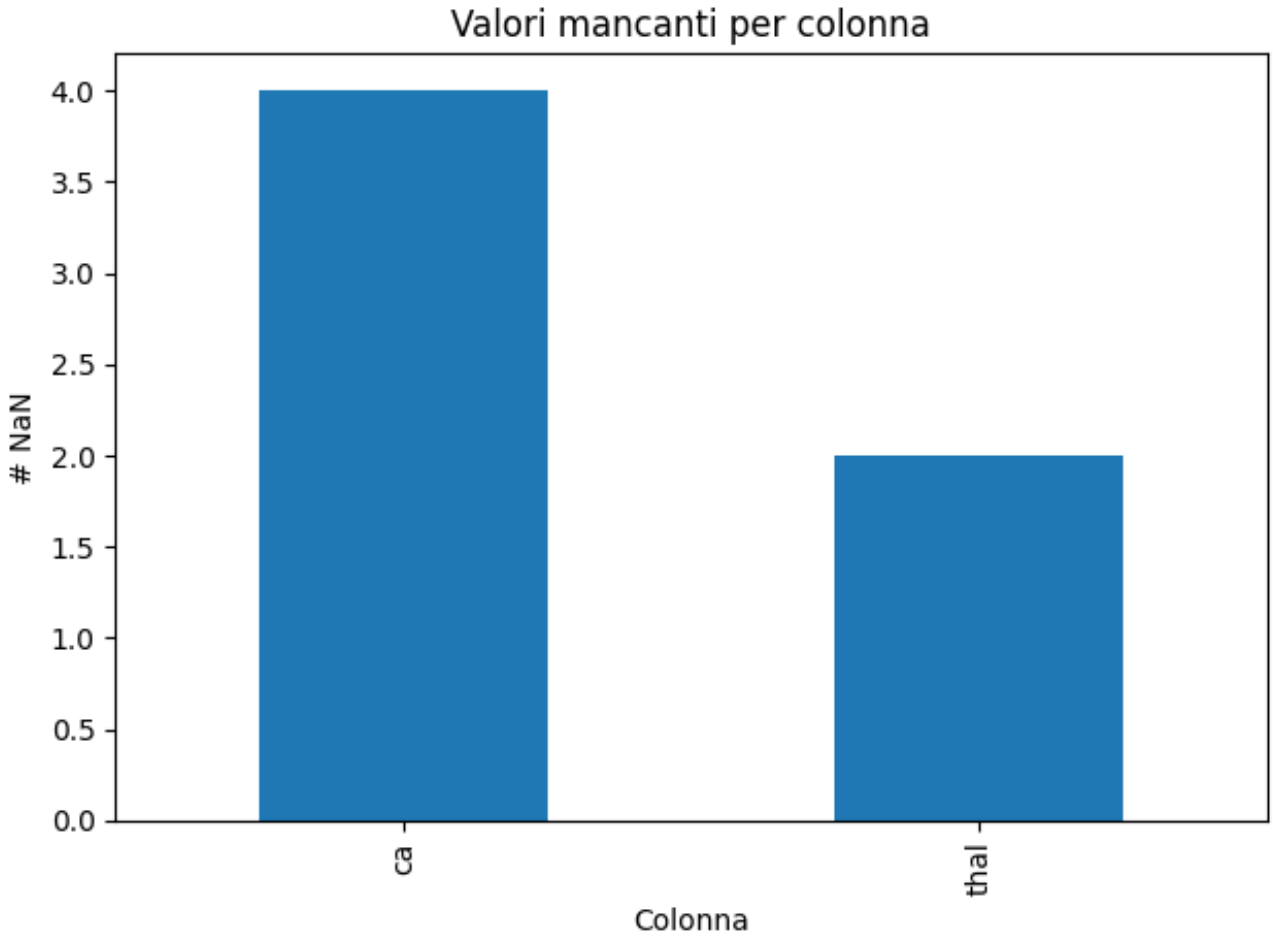

Figura 2: Valori mancanti ca (1,3%), thal(0,7%)

4.1.5. PRE-PROCESSING

Per portare i dati clinici in una forma **coerente, informativa e priva di *leakage***, cosi che il modello apprenda pattern reali e **generalizzi i test**, le trasformazioni sono **incapsulate** in una ***Pipeline scikit-learn*** garantendo che il *fit* delle trasformazioni avvenga esclusivamente sul ***training set*** in ogni *fold* **CV** (*cross-validation*).

I **valori numerici** sono gestiti tramite imputazione mediana (molto robusta per gli outlier):

$$x_j^{imp} = \begin{cases} x_j & \text{se } x_j \text{ è osservato} \\ median(x_j) & \text{altrimenti} \end{cases}$$

Inoltre usiamo uno ***StandardScaler,*** in questo modo otteniamo variabili con **media ~0 e varianza ~1** garantendo in questo modo una distribuzione equa nel misurare le distanze. Questo risulta necessario per applicare l'algoritmo KNN (non ha implicazioni quando usiamo gli algoritmi RF/DT).

Per i **valori categorici** usiamo i valore più frequente (*most frequent),* ogni categoria diventa una colonna binaria e le categorie che non sono state osservate in fase di **addestramento** non causano errori durante il test.

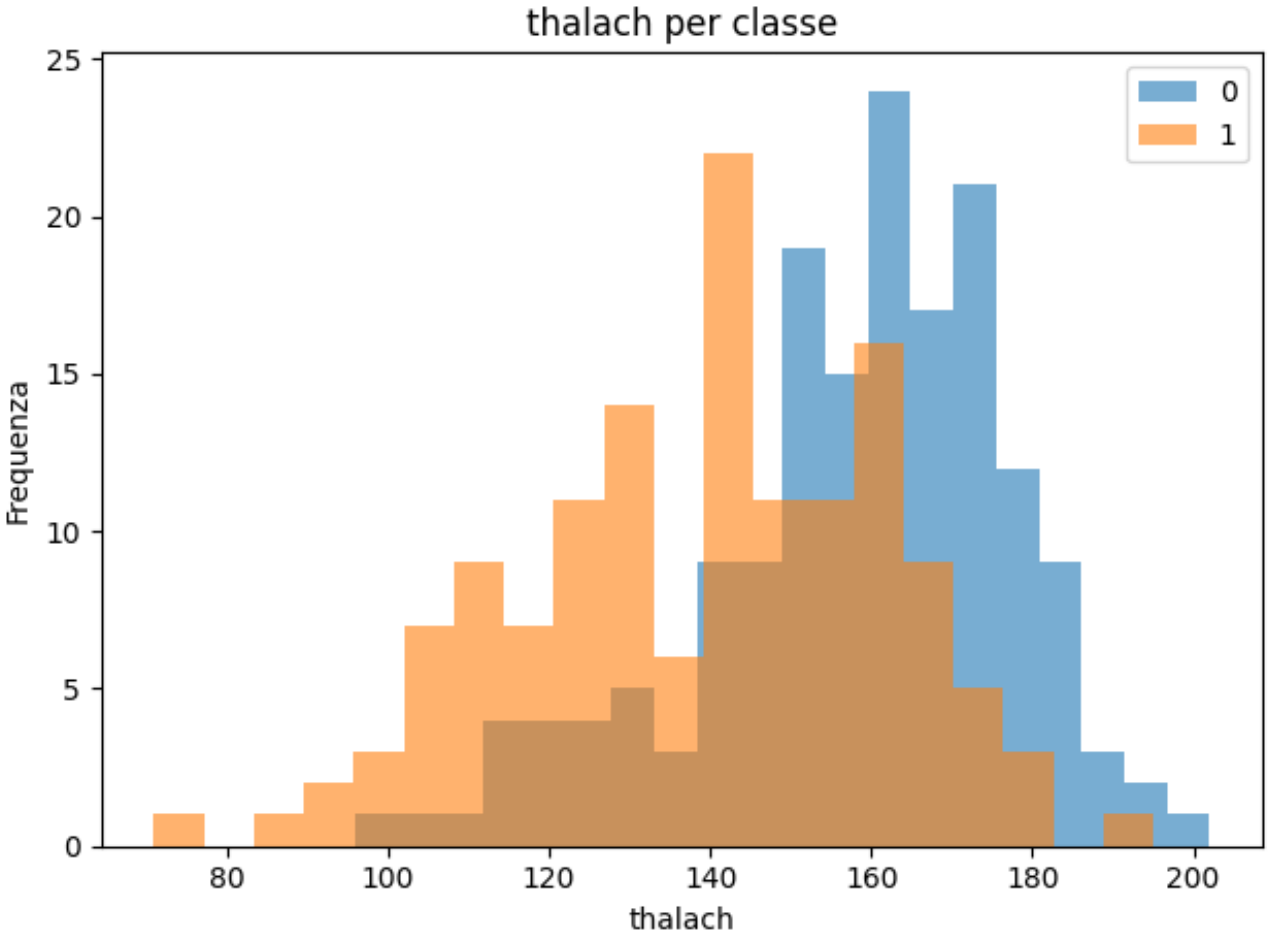

Figura 3: Distribuzione di ‘thalach’ per classe, valori più bassi nei positivi

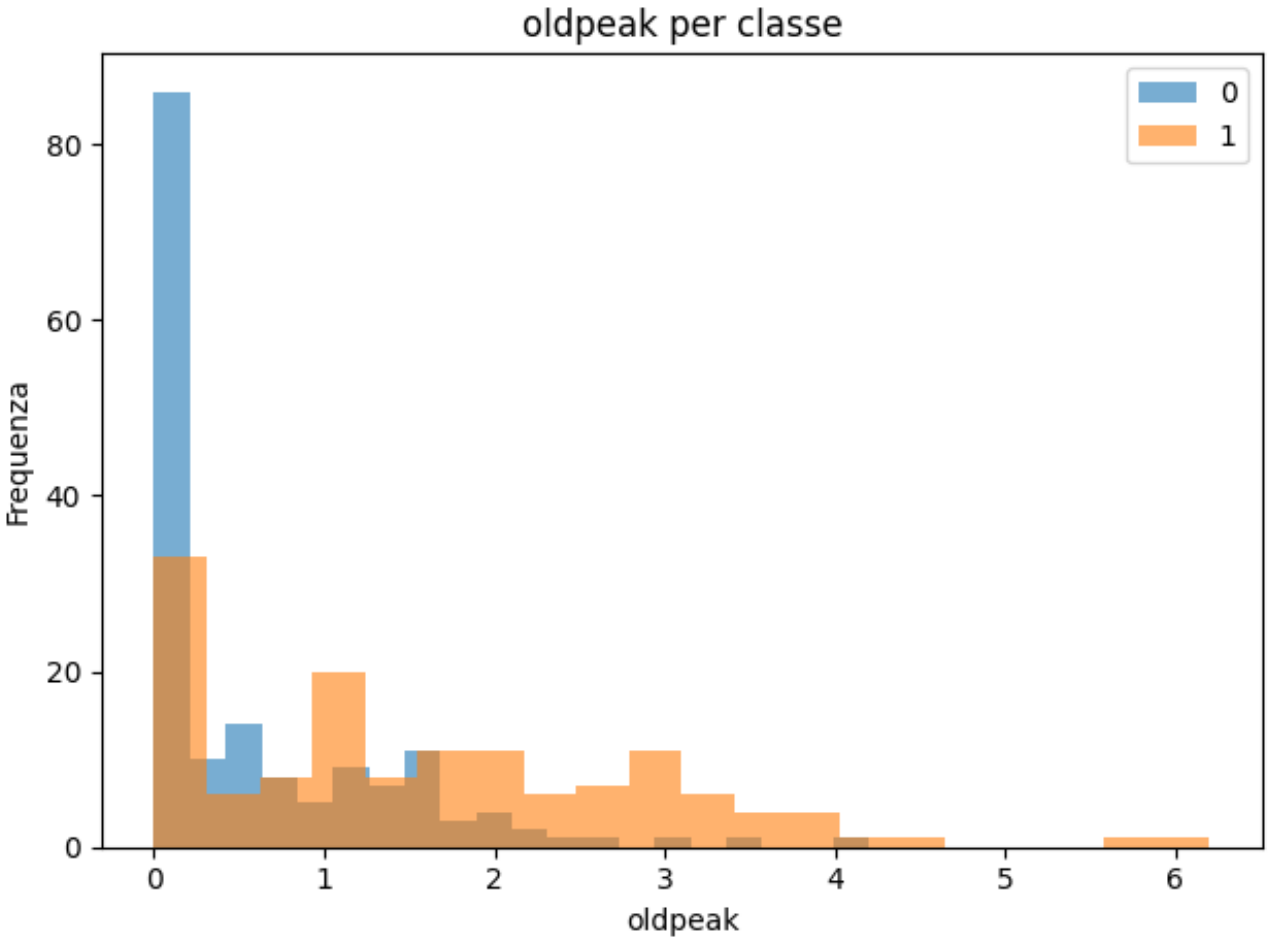

Figura 4: Distribuzione di ‘oldpeak’ per classe, valori più alti nei positivi

4.1.6. FEATURE ENGINEERING

Durante l'EDA sono stati individuati dei pattern tra gli attribuiti del dataset (es. nei positivi abbiamo valori alti dell'attributo '*oldpeak*' e bassi per '*thalach*').

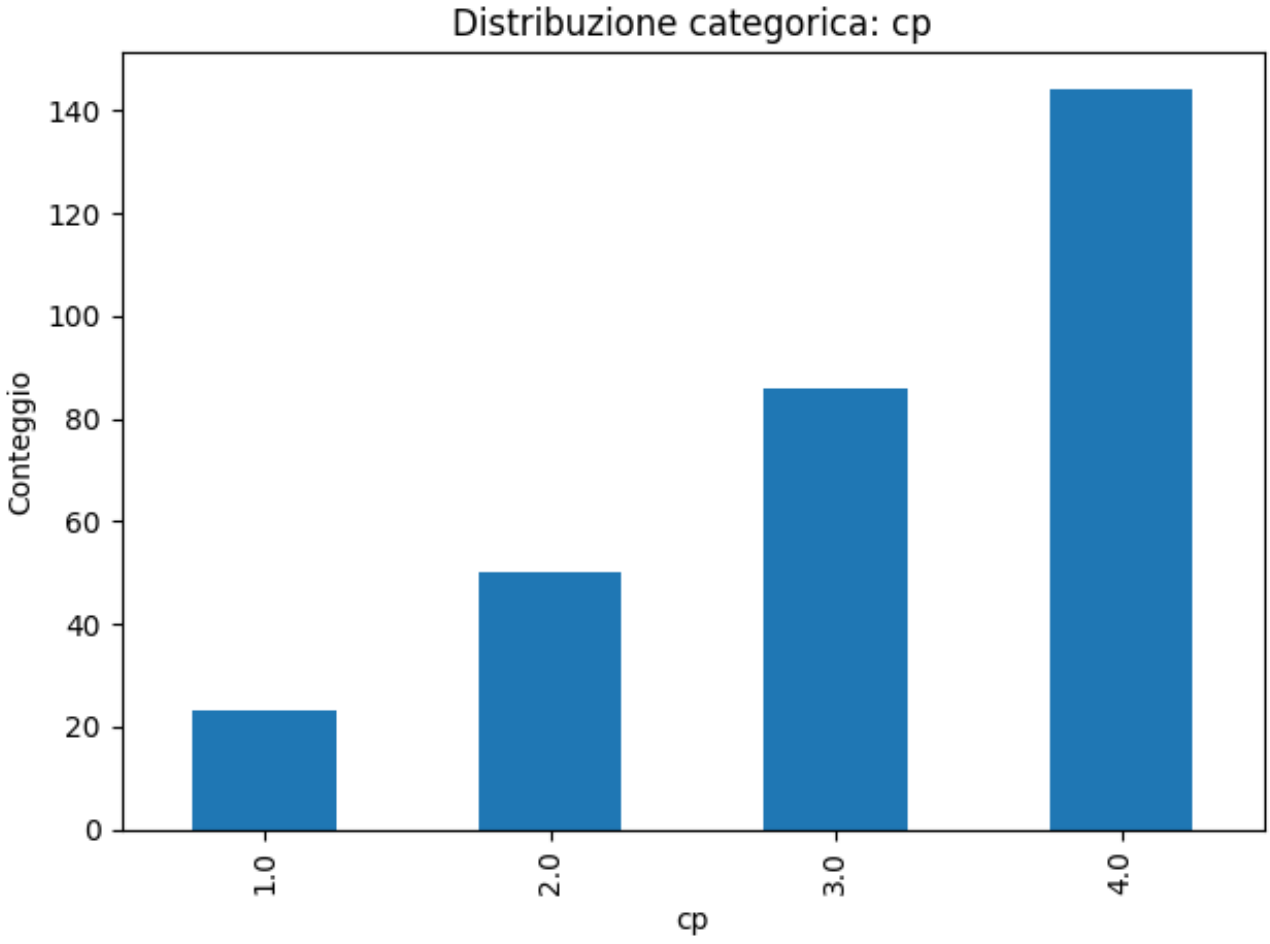

Figura 5: *cp* percentuale di positivi per categoria

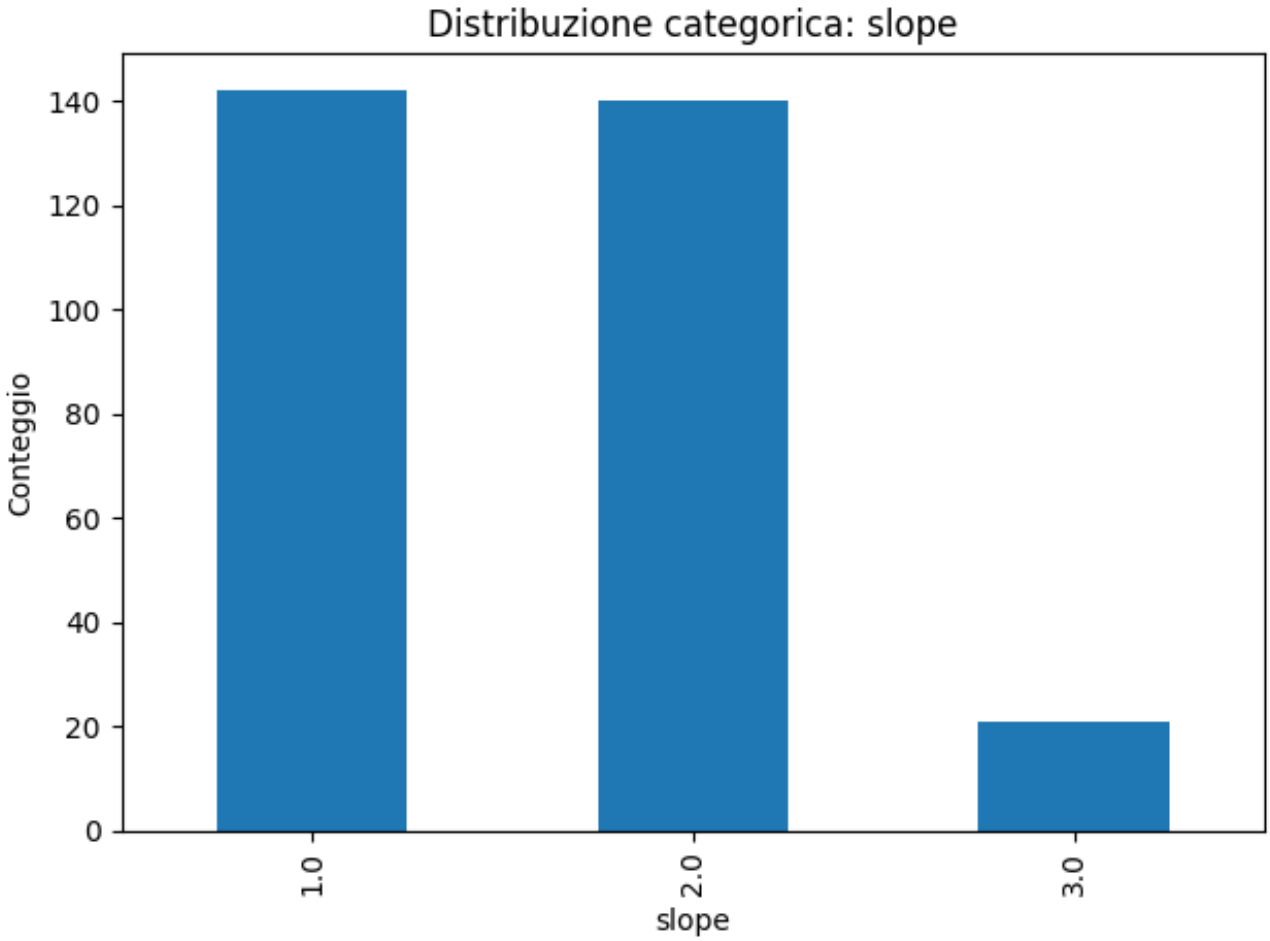

Figura 6: *slope* percentuale di positivi per categoria

Per questo motivo si rende necessario estrarre informazioni cliniche utili partendo dai dati originali:

- **Riserva di frequenza cardiaca** (normalizzata per età): la frequenza massima decresce con l'avanzare dell'età, normalizzando l'attributo '*thalach*' possiamo comparare pazienti di età diverse:

$$\text{hr-reverse} = \frac{thalach}{220 - age}$$

- **Colesterolo normalizzato per età**: Il colesterolo può avere un impatto diverso in base alla fascia di età considerata, per questo introduciamo un piccolo aggiustamento:

$$\text{chol-age} = \frac{chol}{age}$$

- **Interazione ischemica x sforzo**: L'ischemia da sforzo ('oldpeak') può essere più marcato quando il paziente presente **angina da sforzo** ('exang'):

$$\text{oldpeak-exang} = oldpeak * exang$$

- **Interazione ischemica x morfologia ST**: il tratto '*slope*' (valori {1: crescente, 2: piatta, 3: decrescente}) modula l'interpretazione clinica di '*oldpeak*':

$$\text{oldpeak-exang} = oldpeak * slopecode$$

- **Pressione/età:** indice usato per "pesare" la pressione a riposo in funzione dell'età:

$$\text{bp-age} = \frac{trestbps}{age}$$

- ***Binning* di *'age'*** (opzionale: il calco è disattivatile nel codice tramite *add_age_bin = True*)***:*** la relazione età-rischio può non essere lineare, per questo motivo introduciamo una soglia che possiamo interpretare mediante il *binning*.

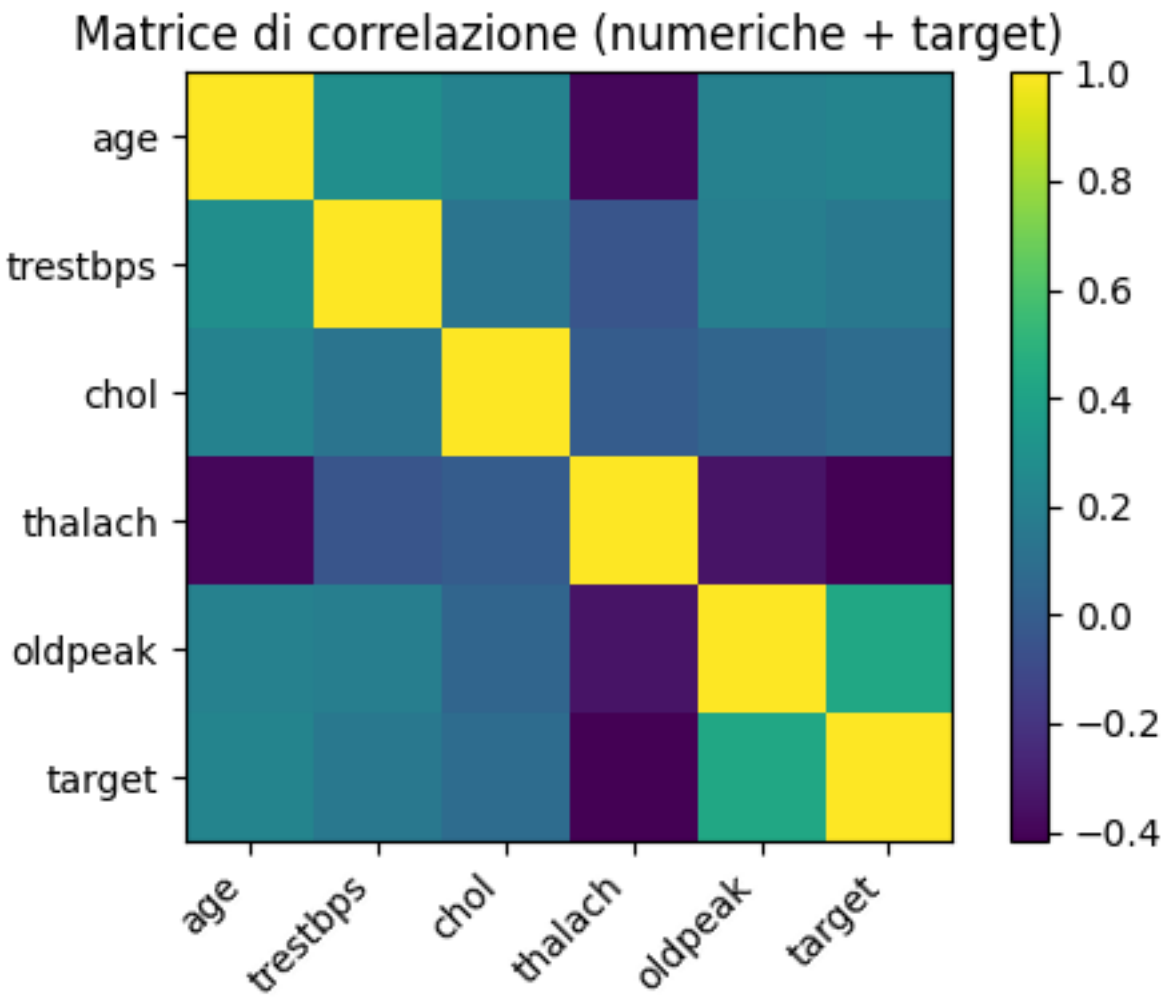

Figura 7: correlazioni *oldpeak (+)* e *thalach (-)* legate al target

5. METODO DI VALUTAZIONE

Il confronto **corretto e riproducibile** dei tre algoritmi utilizzati per risolvere il problema sopra descritto (KNN, Decision Tree e Random Forest) risulta centrare. Come abbiamo potuto constatare le classi sono moderatamente sbilanciate, questo ci porta a scegliere come **metrica principale la F1** che penalizza i falsi positivi e i falsi negativi. Affianco alla metrica principale, usiamo **ROC-AUC e PR-AUC** per valutare la qualità dell'ordinamento delle probabilità prodotte dai modelli. Infine come supporto per la lettura dei risultati usiamo: ***l'accuracy*, la *precision* e il *recall***.

Il dataset viene suddiviso in uno **split stratificato 80/20.** L'80% è utilizzato in fase di training e validazione del modello, viene applicato un ***cross-validation* a 5 *fold*** con *shuffle*. Mediante poi la GridSearchCV (storing = F1), ciascun algoritmo esplora una griglia di iperparametri scegliendo la configurazione che massimizza la **F1 media in CV**. Per ogni algoritmo otterremo un '***best estimator***' già addestrato sull'intero training (usando sempre GridSearchCV con *refit = True*). Per il restante 20%, basandoci su ciascun '*best estimator*', calcoliamo: *F1, Accuracy, Precision, Recall* (imposetermo una soglia standard pari a 0.5) infine tramite questi indicatori tracceremo le **curve ROC** e **Precision-Recall** con le rispettive aree (ROC-AUC e PR-AUC). Tra le misure vi è un indice **"best-F1"** che massimizza la F1 lungo la curva PR, questo va letto con cautela perché è usato per ottimizzare il test non per il confronto dei modelli. È importante sottolineare come il test non influenza in alcun modo il tuning degli iperparametri o la scelta del modello.

Gli artefatti del processo (cv_*.cvs, metrics_*.json, confutino Matrix ecc..) vengono memorizzati nella cartella *out/*.

6. RISULTATI OTTENUTI

Di seguito riportiamo le prestazioni dei tre algoritmi considerati:

Hold-out test

Modello	CV best F1	Test F1 @0.5	Test Accuracy	ROC-AUC	PR-AUC	Test F1 @best
Random Forest	0,792	0,912	0,918	0,960	0,950	0,929
KNN	0,799	0,909	0,918	0,945	0,899	0,909
Decision tree	0,742	0,772	0,787	0,83	0,77	0,815

Tabella 2: i dati vengono arrotondanti alla terza cifra decimale per migliorare la lettura

Dalle misure ottenute possiamo giungere alla conclusione che l'algoritmo **Random Forest (RF)** è il migliore. In base alla soglia 0.5 notiamo come lo score F1 del RF è il più alto tra i modelli considerati, cosi come nelle metriche ROC-AUC e PR-AUC. Questo indica che la RF ordina bene i campioni per rischio. Segue subito il **KNN**, esso ha uno score F1 molto vicino al RF, grazie anche allo **scaling** delle numeriche, resta pero leggermente indietro in AUC (opera bene con le soglie fisse ma l'ordinamento probabilistico è meno robusto rispetto al RF). Infine il **decision tree** risulta meno poco attendibile rispetto ai precedenti algoritmi, nonostante sia apprezzabile per l'interpretabilità.

Di seguito sono forniti i grafici delle misure ricavate dall'algoritmo RF:

Note: I grafici relativi agli algoritmi KNN e DT sono presenti nella cartella *out/figures*

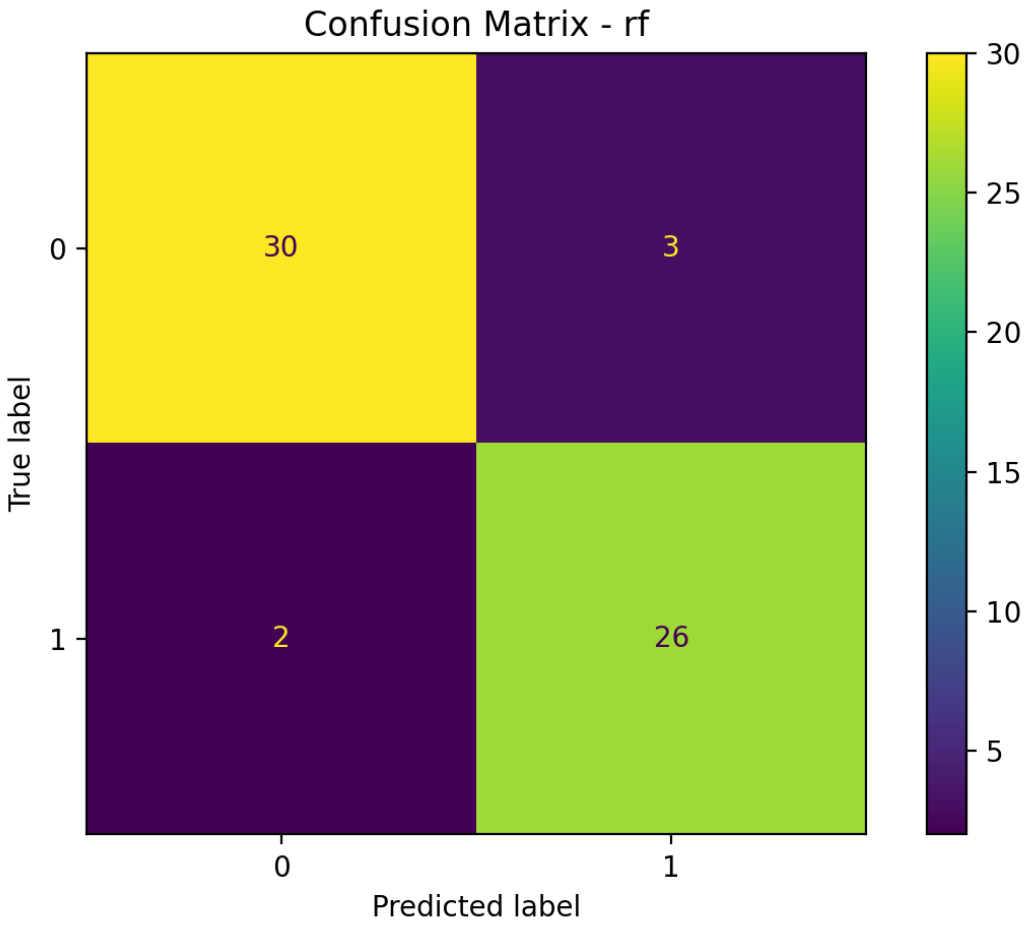

Figura 8: Confusion Matrix (RF, soglia 0.5).

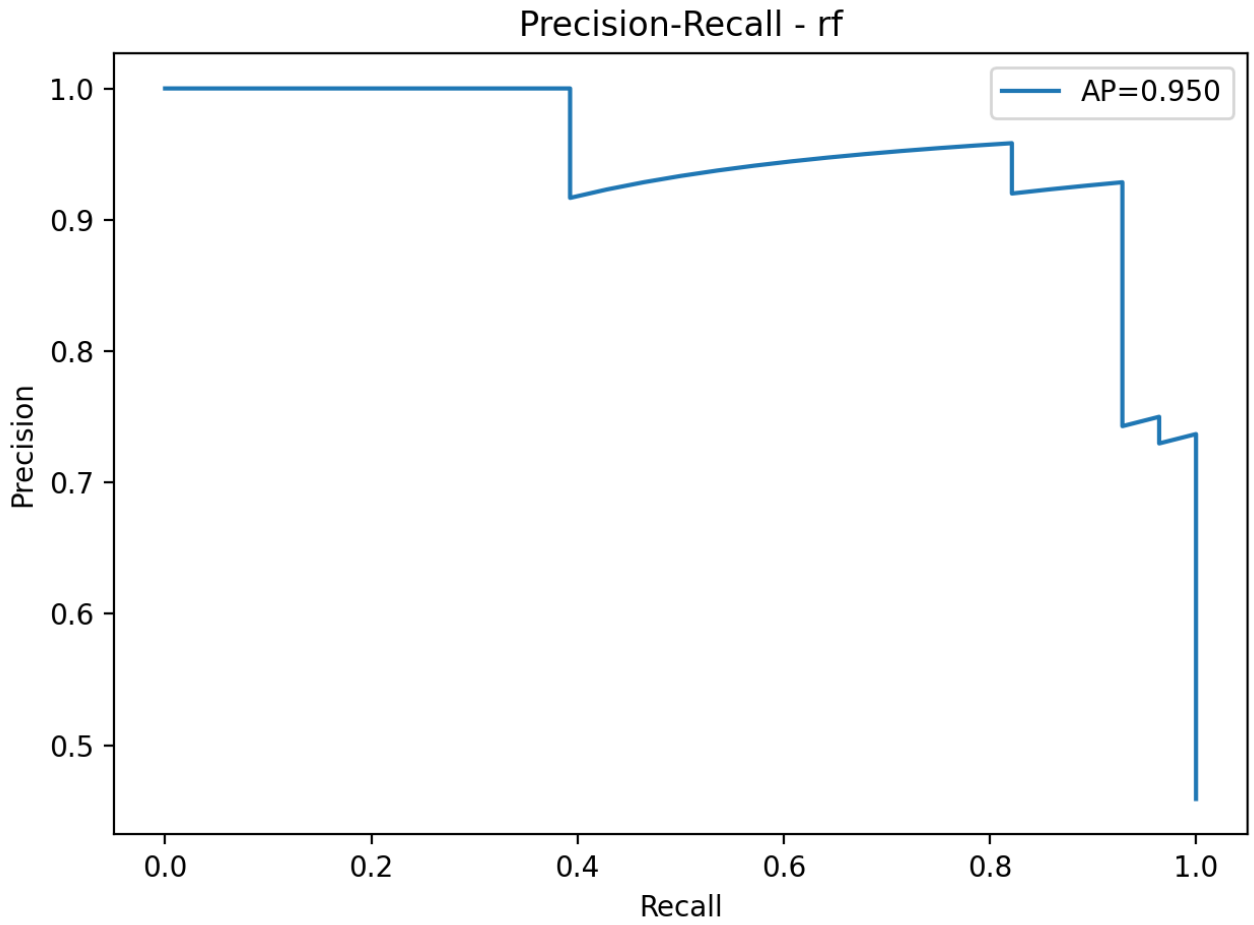

Figura 9: Curva PR (RF): PR-AUC ≈ 0.95.

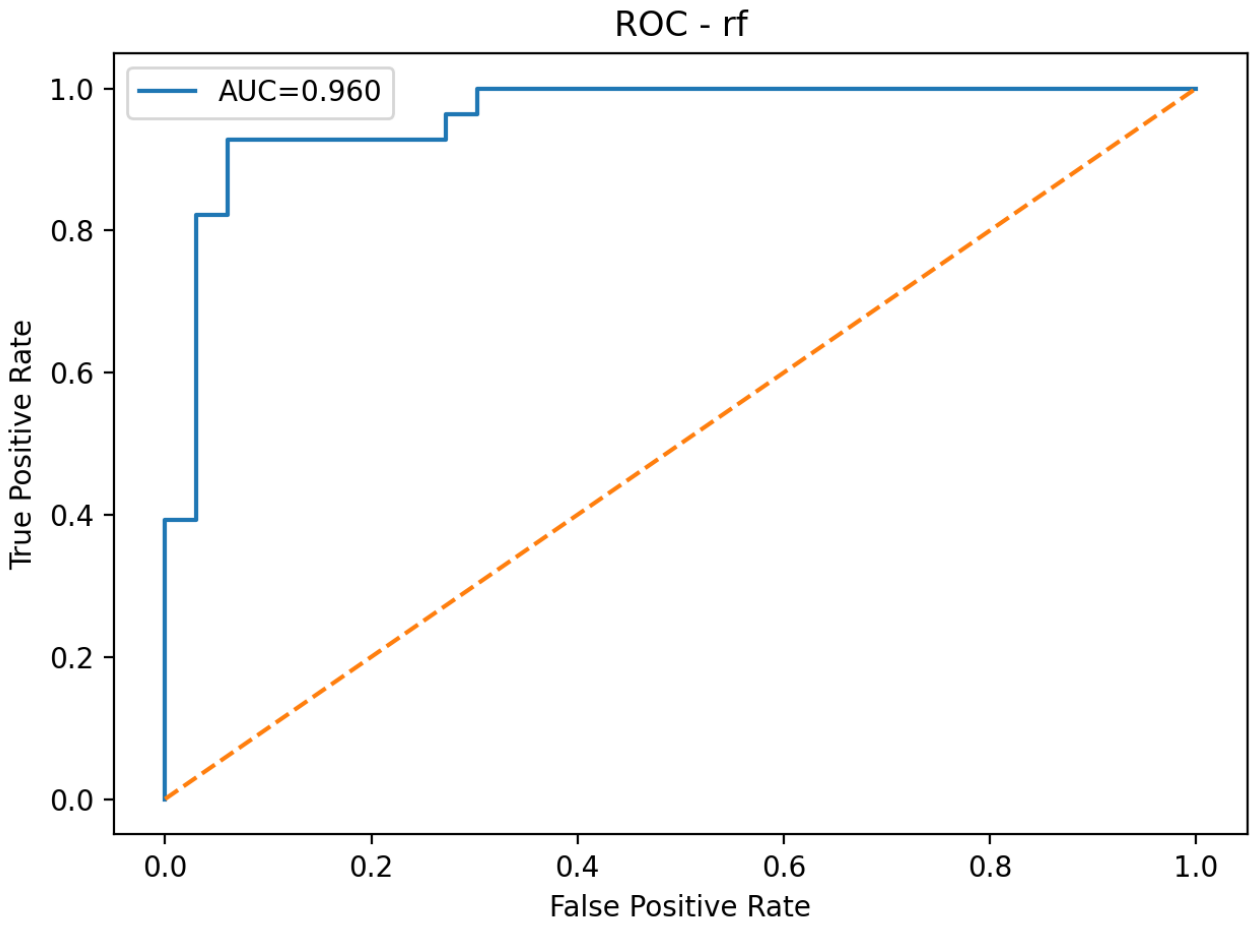

Figura 10: Curva ROC (RF): AUC ≈ 0.96.

7. CONCLUSIONI

Abbiamo costruito una pipeline per classificare la presenza o assenza di malattie cardiache (Cleveland), con **imputazione, scaling, One-Hot e feature engineering**. Abbiamo constatato che esiste un moderato sbilanciamento delle classi e per questo motivo abbiamo usato come metrica principale la F1. Dai risultati ottenuti la **Random Forrest** risulta, tra gli algoritmi presi in considerazione, il migliore da utilizzare.

7.1. LIMITI

- **Dimensione del dataset** contenuta (303 righe): le stime possono fluttuare con lo split; piccole differenze tra KNN e RF devono essere valutate con cautela.

- **Generalizzabilità**: i risultati valgono per il subset Cleveland, non sono stati eseguiti ulteriori test sugli altri subset; questo limita le conclusioni cliniche.

7.2. POSSIBILI ESTENSIONI

1. **Per garantire la robustezza dei risultati**, si raccomanda di ripetere lo split con un numero maggiore di seed (media±std) o di utilizzare il bootstrap sui test per l'Intervallo di Confidenza (IC).

2. **La soglia operativa** dovrebbe essere determinata in base al valore di val/CV, tenendo conto dei costi clinici. In particolare, si raccomanda di privilegiare la soglia Recall qualora l'obiettivo primario sia la riduzione dei falsi negativi.

3. **Calibrazione**: Platt/Isotonic, Brier score e calibration curves.

4. **Interpretabilità**: Permutation/SHAP, PDP/ICE su oldpeak, thalach, cp, ca, thal.

5. **Modellazione**: grid più ampia per RF (es. min_samples_leaf, max_features), o confronto con Gradient Boosting(XGBoost/LightGBM).

6. **Validazione esterna**: provare su altri subset UCI (Hungarian, Long Beach, Switzerland) o dati indipendenti.