Università degli Studi di Salerno
Corso di Laurea in Informatica

Stroke Prediction

Progetto di Machine Learning

Alessio Inglese
Matricola: 0512118760

Adriano De Vita
Matricola: 0512117726

Prof. L. Caruccio, Prof. G. Polese

Anno Accademico 2024/2025

Indice

1 Introduzione **3**
1.1 Scopo del progetto 3
1.2 Pipeline 3

2 Esplorazione dei dati (EDA) **3**
2.1 Acquisizione del dataset 3
2.2 Esplorazione del dataset 3
2.3 Insights preliminari 5

3 Pre-processing del dataset **6**
3.1 Data Cleaning 6
3.2 Data Imputation 6
3.3 Data Balancing 7
3.4 Train/Validation/Test split 8
3.5 Feature Selection 8
3.6 Feature Engineering 9

4 Training e validazione del modello **9**
4.1 Training 10
4.1.1 Soft AutoML: regolazione degli iperparametri 10
4.2 Validazione 11

5 Valutazione del modello **11**
5.1 Metriche finali 11
5.2 Considerazioni sulle metriche 14
5.3 Accorgimenti finali 15

6 Repository del Progetto **15**

1 Introduzione

1.1 Scopo del progetto

Questo progetto mira a fornire una base per le predizioni in campo medico, nello specifico nella predizione di infarto cerebrale, volgarmente chiamato *ictus*.

1.2 Pipeline

Per lo svolgimento di questo progetto si è pensato di adoperare una pipeline di esecuzione divisa in *4* steps:

1. Esplorazione dei dati (EDA);
2. Pre-processing del dataset;
3. Training e validazione del modello;
4. Valutazione del modello.

2 Esplorazione dei dati (EDA)

In questa fase andremo a prelevare ed analizzare accuratamente il dataset di *stroke-prediction* utilizzato.

2.1 Acquisizione del dataset

Il dataset utilizzato durante lo sviluppo del progetto è reperibile al seguente link Kaggle: Stroke-Prediction-Dataset. Il dataset è stato dunque prelevato ed inserito all'interno della cartella progettuale al seguente percorso file: */data/raw/stroke-data.csv*.

2.2 Esplorazione del dataset

Il dataset è composto da diverse variabili che rappresentano caratteristiche demografiche, cliniche e comportamentali dei pazienti. Le colonne presenti nel dataset sono le seguenti:

- **gender**: indica il genere del paziente (Male, Female, Other).
- **age**: età del paziente espressa in anni.
- **hypertension**: presenza (1) o assenza (0) di ipertensione.

- **heart_disease**: presenza (1) o assenza (0) di malattie cardiache.
- **ever_married**: indica se il paziente è mai stato sposato (Yes, No).
- **work_type**: tipologia di impiego del paziente (Private, Self--employed, Govt_job, Children, Never_worked).
- **Residence_type**: tipo di residenza del paziente (Urban, Rural).
- **avg_glucose_level**: livello medio di glucosio nel sangue.
- **bmi**: indice di massa corporea (Body Mass Index).
- **smoking_status**: stato di fumatore del paziente (formerly smoked, never smoked, smokes, Unknown).
- **stroke**: variabile target che indica se il paziente ha avuto un ictus (1) o meno (0).

Il dataset contiene sia variabili numeriche (*age*, *avg_glucose_level*, *bmi*) che variabili categoriche (*gender*, *hypertension*, *heart_disease*, *ever_married*, *work_type*, *Residence_type*, *smoking_status*).
Il processo di Exploratory Data Analysis (EDA) è stato poi visualizzato, nello specifico, attraverso l'utilizzo dei seguenti strumenti grafici:

- Barplot per i valori mancanti;
- Istogrammi per mostrare la distribuzione delle variabili numeriche;
- Istogrammi per mostrare la distribuzione delle variabili categoriche;
- Pie-plot per mostrare la distribuzione della variabile target;
- Scatter plots per mostrare gli outliers nelle variabili numeriche;
- Matrici e heatmaps per misurare la correlazione tra gli attributi del dataset;
- Plot ANOVA e Chi-quadro per mostrare l'importanza delle variabili numeriche e categoriche riguardo la variabile target;

Inoltre, un report dell'EDA è stato salvato in *'data/eda/eda_report.txt'*.
L'analisi esplorativa del dataset ha dunque permesso di comprendere la distribuzione delle variabili, identificare eventuali valori mancanti e verificare la presenza di sbilanciamenti nei dati, in particolare per quanto riguarda la classe target (**stroke**).

2.3 Insights preliminari

Una volta terminata l'**EDA**, tramite il report e i plot, abbiamo notato alcune modifiche necessarie da effettuare sul dataset, prima di utilizzarlo per l'addestramento del modello di Machine Learning.

- La variabile *BMI* presenta alcuni valori nulli;
- Sono presenti alcuni outliers nella variabile *avg_glucose_level* e in *BMI*;
- Esiste solo una riga del dataset con la tipologia di *gender* = *other*;
- La colonna target risulta fortemente sbilanciata verso la classe negativa (0), come previsto, poiché i casi di ictus sono, fortunatamente, molto meno frequenti rispetto ai casi negativi. Questo sbilanciamento riflette la reale distribuzione della condizione nella popolazione e rappresenta una sfida significativa, richiedendo tecniche adeguate per gestire il disequilibrio tra le classi.

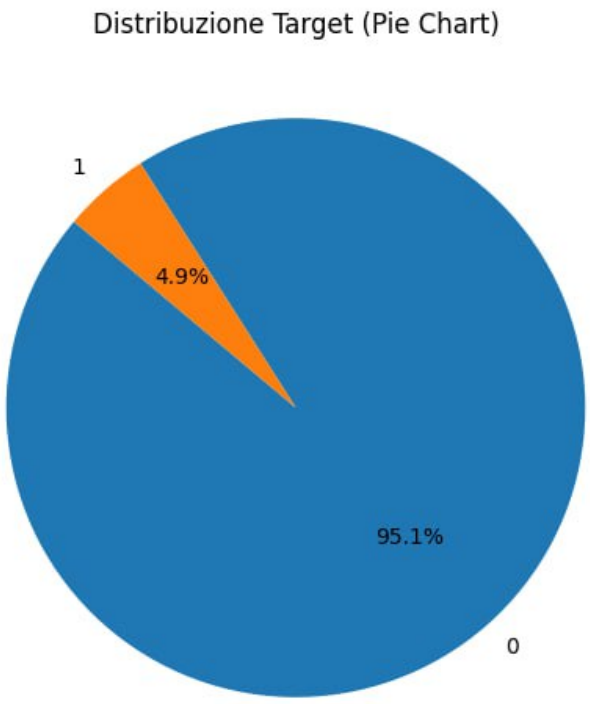

Figura 1: Distribuzione della variabile target, in blu i negativi e in arancio i positivi

3 Pre-processing del dataset

In questa fase andremo dunque ad applicare alcune modifiche suggerite dagli insights preliminari, effettuando pulizia, imputazione, bilanciamento e split del dataset. Dopodichè decideremo in base all'importanza, su quali feature addestrare il modello

3.1 Data Cleaning

Per prima cosa andiamo ad eliminare l'istanza con valore Other nell'attributo gender, trattandosi di una sola istanza non rappresentativa rispetto a tutte le altre classificate come male e female. Andiamo anche ad analizzare gli outliers e decidiamo di mantenerli tali per non distorcere la rappresentatività dei dati poiché essi si dimostrano comunque plausibili dopo alcune ricerche.

3.2 Data Imputation

Andiamo ora ad imputare per **mediana**, i valori nulli appartenenti all'attributo *BMI* del dataset che rappresenta l'indice di massa corporea dei pazienti clinici. E' stata scelta l'imputazione per mediana poichè essa non distorce eccessivamente la realtà dei dati e mantiene la distribuzione in modo resistente agli outliers.

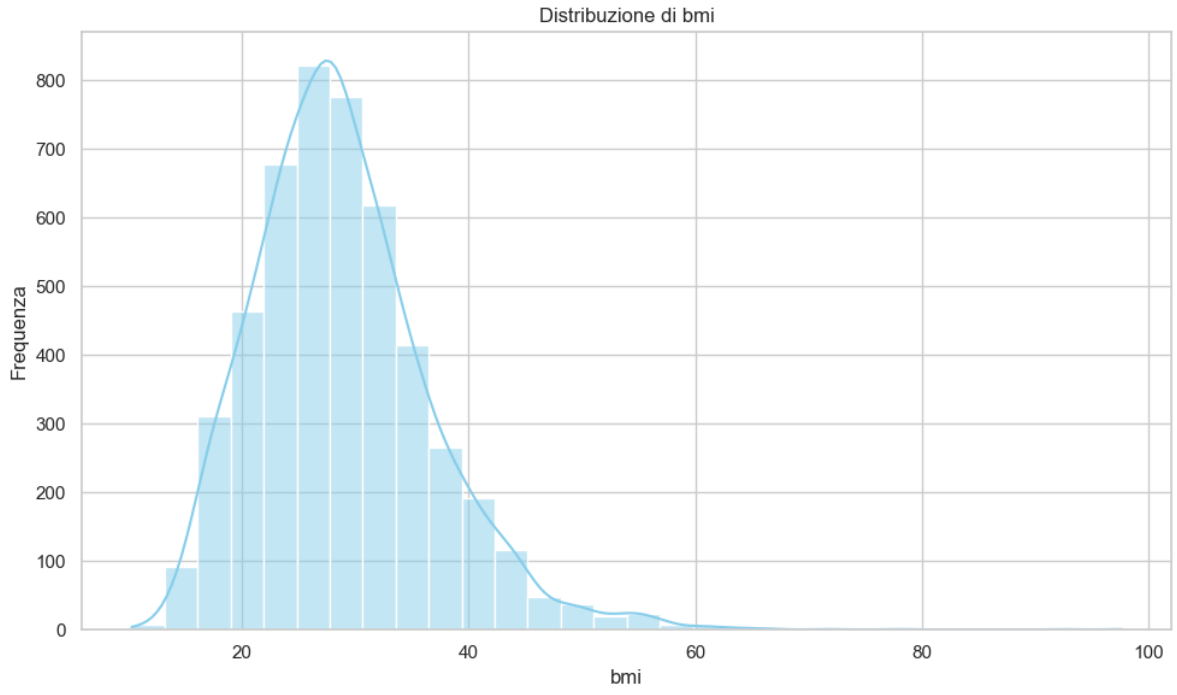

Figura 2: BMI prima dell'imputazione per mediana (presenta valori nulli)

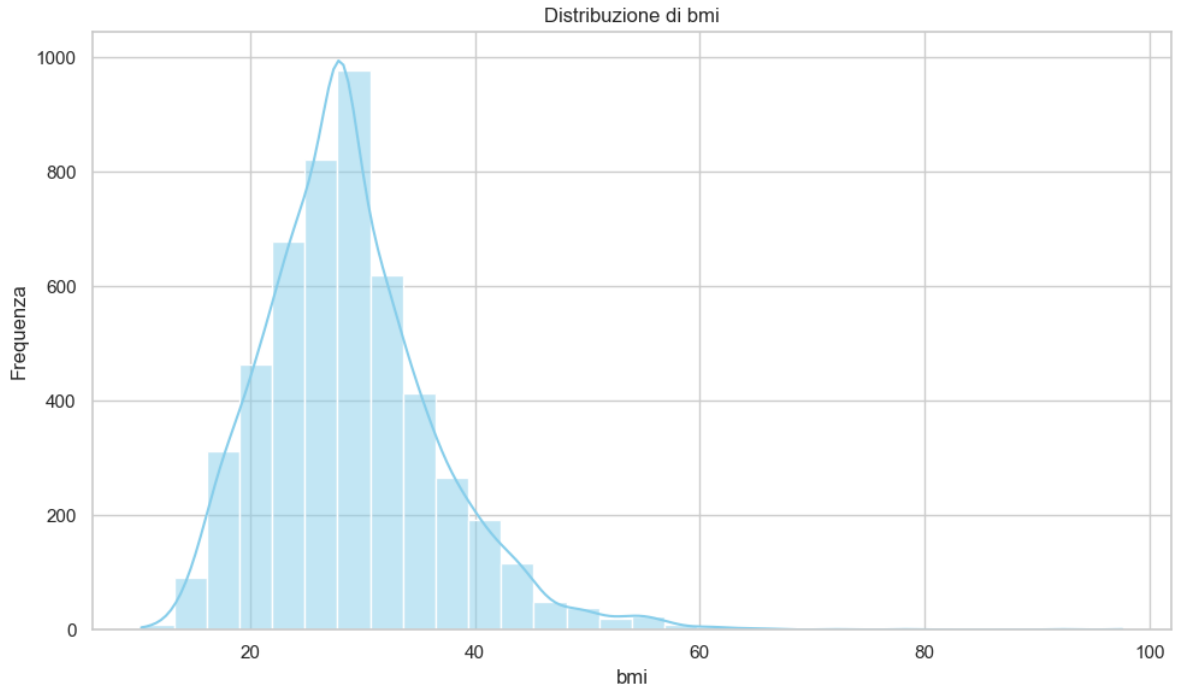

Figura 3: BMI dopo l'imputazione per mediana (non presenta valori nulli)

3.3 Data Balancing

Dall'analisi preliminare emerge che la variabile target è fortemente sbilanciata, con una netta predominanza della classe negativa ($Stroke = 0$). Per mitigare questo squilibrio, utilizziamo la tecnica di oversampling SMOTE-Borderline, che ci consente di aumentare il numero di istanze della classe positiva ($Stroke = 1$) in maniera sintetica senza alterare eccessivamente la distribuzione originale dei dati.
A differenza dello SMOTE standard, che genera nuovi punti interpolando tra i vicini della classe minoritaria in modo casuale, SMOTE-Borderline si concentra sulle istanze della classe minoritaria che si trovano vicino al confine decisionale con la classe maggioritaria. Questo approccio permette di migliorare la rappresentazione dei casi più critici, ovvero quelli difficili da classificare, generando nuove istanze nelle regioni in cui la separazione tra le due classi è meno netta. In questo modo, il modello può apprendere in modo più efficace le caratteristiche distintive dei casi di ictus, migliorando la capacità di generalizzazione e riducendo il rischio di bias a favore della classe maggioritaria. Essendo che la distribuzione originale della classe target è approssimativamente 1:20, il rapporto di SMOTE che andremo ad utilizzare verrà scelto empiricamente e varierà nel range:

$$1 : 15 \leq \text{SMOTE ratio} \leq 1 : 19$$

3.4 Train/Validation/Test split

Per poter addestrare e validare in maniera ottimale e coerente il modello effettuiamo una suddivisione del dataset: il 70% dei dati saranno per l'addestramento del modello e il restante 30% dei dati che il modello non conoscerà, verranno suddivisi al 15% in dati di validazione e il restante 15% in dati di test, su cui il modello predirà se determinate istanze sono o meno **a rischio di ictus**. Per evitare il data leakage, il preprocessing è stato strutturato in modo che tutte le trasformazioni siano calcolate esclusivamente sul training set e poi applicate a validation e test set. In particolare, la suddivisione del dataset avviene prima dell'imputazione dei valori mancanti, della codifica delle variabili categoriche e di qualsiasi altra trasformazione. Infine, nessuna tecnica di oversampling è applicata prima della divisione dei dati, garantendo che la distribuzione della variabile target rimanga inalterata tra i set. Queste accortezze assicurano che il modello non apprenda pattern artificiali derivanti da informazioni future, rendendo la valutazione più affidabile.

3.5 Feature Selection

Rimuoviamo ora dai dati di addestramento, validazione e test le feature con bassa rilevanza. Il criterio di rilevanza utilizzato per la feature selection è il seguente: tutte le feature all'interno del dataset su cui il modello potrà essere addestrato saranno quelle con indice di correlazione è almeno > 2. Nei grafici è stata applicata una trasformazione logaritmica a *p-value* in modo tale da ottenere ciò che è graficamente visibile come indice di correlazione.

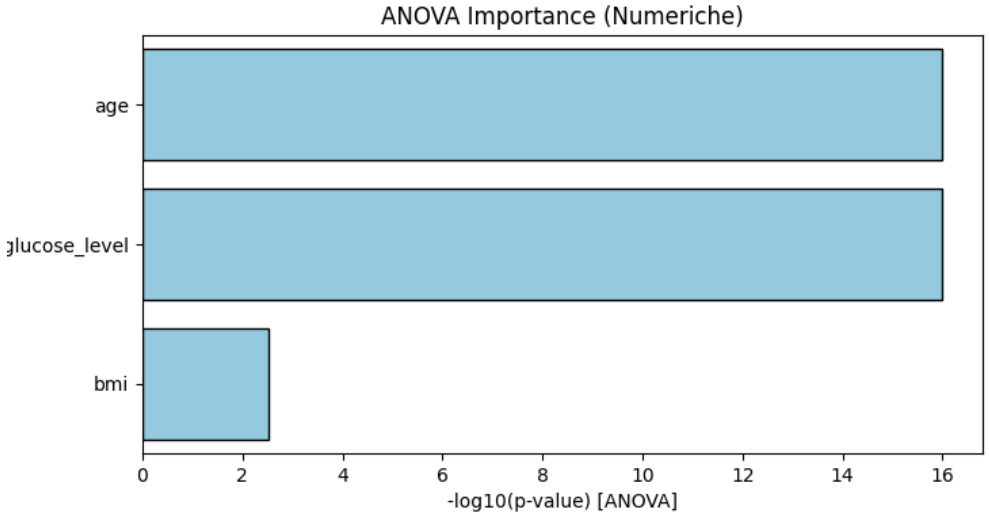

Figura 4: ANOVA plot che mostra l'importanza delle variabili numeriche

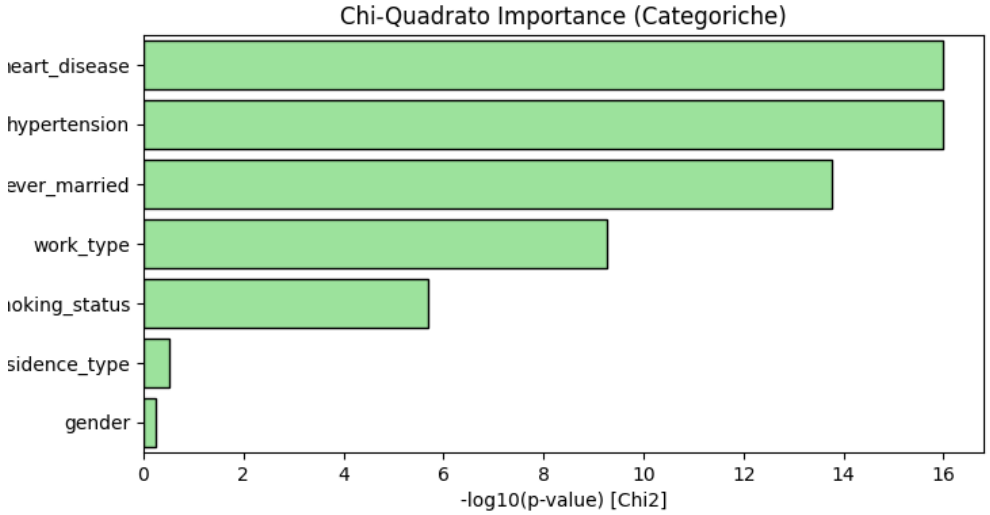

Figura 5: Plot del Chi-Quadrato che mostra l'importanza delle variabili categoriche

Come possiamo notare dai plot, le variabili categoriche *gender* e *residence_type* non superano l'indice di correlazione previsto per la selezione, di conseguenza verranno scartate.

3.6 Feature Engineering

Andiamo poi ad effettuare la binarizzazione della feature *ever_married* in maniera da rendere più interpretabili i dati al modello. Risultato: "Yes" = 1 e "No" = 0. Inoltre effettuiamo l'encoding di tutte le variabili categoriche applicando **Label Encoding**, trasformando così tutte le scelte di categoria in valori numerici ordinali. Abbiamo deciso di adottare il **Label Encoding** in maniera empirica dopo aver testato anche altri approcci come **One Hot Encoder** non ottenendo risultati ottimali.

4 Training e validazione del modello

L'addestramento e la validazione del modello rappresentano fasi cruciali nel processo di sviluppo di un sistema di Machine Learning. In questa sezione verranno descritti i passi adottati per la selezione, la regolazione degli iperparametri e la valutazione del modello, con particolare attenzione alla prevenzione dell'overfitting e alla scelta delle metriche più appropriate per il problema di classificazione degli ictus.

4.1 Training

Per questo studio è stato scelto di utilizzare il **Random Forest Classifier**, un algoritmo di apprendimento supervisionato (*ensemble*) basato su alberi decisionali. La decisione di adottare questa tecnica si basa sulla robustezza del modello rispetto al dataset con variabili miste (numeriche e categoriche), sulla sua capacità di gestire dataset sbilanciati grazie all'attributo *class_weight='balanced'*, e sulla sua interpretabilità, supportata dall'analisi dell'importanza delle feature. Inoltre, diversi studi in letteratura [ea21, ea23, ea24] suggeriscono che Random Forest sia efficace nella predizione di eventi rari come l'ictus. Come citato in precedenza, il dataset è stato suddiviso in tre insiemi:

- **Training set** (70%): utilizzato per l'addestramento del modello e tuning degli iperparametri;
- **Validation set** (15%): utilizzato per la validazione del modello e per la selezione della soglia di decisione ottimale;
- **Test set** (15%): non utilizzato in questa fase, riservato per la valutazione finale.

Questa suddivisione segue un **campionamento stratificato** sulla variabile target (*stroke*) per garantire che la distribuzione delle classi sia preservata in ogni insieme.

4.1.1 Soft AutoML: regolazione degli iperparametri

La regolazione degli iperparametri è stata effettuata tramite **RandomizedSearchCV** con **Stratified K-Fold Cross-Validation** (k=5). Questo approccio consente di esplorare un ampio spazio degli iperparametri riducendo il costo computazionale rispetto a Grid Search. L'ottimizzazione è stata eseguita in 100 iterazioni e rispetto a due metriche:

- **F2-score**: metrica prioritaria, scelta per dare maggiore peso alla *recall*, poiché nel contesto medico è fondamentale minimizzare i falsi negativi (pazienti erroneamente classificati come sani).
- **ROC-AUC**: usata come metrica secondaria per valutare la discriminabilità complessiva del modello.

Lo spazio di ricerca degli iperparametri ha incluso:

- **n_estimators** = [200, 300, 500, 800, 1000]

- **max_depth** = [20, 30, 40, 50]
- **min_samples_split** = [10, 20, 30, 40]
- **min_samples_leaf** = [2, 5, 10]
- **max_features** = [None, "sqrt", "log2", 0.5]
- **criterion** = ["gini", "entropy"]
- **bootstrap** = [True, False]
- **ccp_alpha** = [0.0, 0.0001, 0.001]

L'algoritmo ha individuato la combinazione ottimale di iperparametri, con la quale è stato riaddestrato il modello poi su tutto il training set.

4.2 Validazione

Per valutare le prestazioni del modello e selezionare la soglia di decisione ottimale, è stata utilizzata la curva **Precision-Recall**, con l'obiettivo di massimizzare, anche qui, l'F2-score sul validation set. Il modello, infatti, genera probabilità per ciascuna classe, e la soglia ottimale è stata scelta come quella che massimizza l'F2-score. Inoltre, è stata analizzata l'importanza delle feature, valutando il contributo di ciascuna variabile nella predizione dell'ictus. Questo consente di identificare i fattori di rischio più influenti nel dataset, aumentando l'interpretabilità del modello.

5 Valutazione del modello

Dopo l'addestramento e l'ottimizzazione degli iperparametri, il modello viene valutato su un test set indipendente. Questa fase è cruciale per comprendere le reali prestazioni del modello e la sua capacità di generalizzazione. Sono state utilizzate diverse metriche per garantire una valutazione accurata, con particolare attenzione alla gestione del dataset sbilanciato.

5.1 Metriche finali

Per misurare la qualità delle predizioni del modello, sono state calcolate le seguenti metriche:

- **Accuracy**: misura la percentuale di predizioni corrette rispetto al totale, potrebbe tuttavia risultare fuorviante in caso di sbilanciamento dove i veri negativi sono molti di più dei veri positivi.

- **Precision**: quantifica la proporzione di predizioni positive che sono effettivamente corrette, essenziale per evitare falsi allarmi.
- **Recall**: misura la capacità del modello di identificare tutti i casi positivi, fondamentale in contesti medici dove i falsi negativi sono critici.
- **F1-score**: combina precision e recall in un'unica metrica bilanciata.
- **ROC-AUC**: valuta la capacità del modello di discriminare tra classi, misurando l'area sotto la curva ROC.

Le metriche sono state calcolate utilizzando la soglia di decisione ottimale selezionata nella fase di validazione, basata sull'analisi della Precision-Recall Curve. Inoltre, sono stati generati i seguenti grafici per supportare la valutazione:

- **Matrice di Confusione**: visualizza gli errori di classificazione tra classi positive e negative.

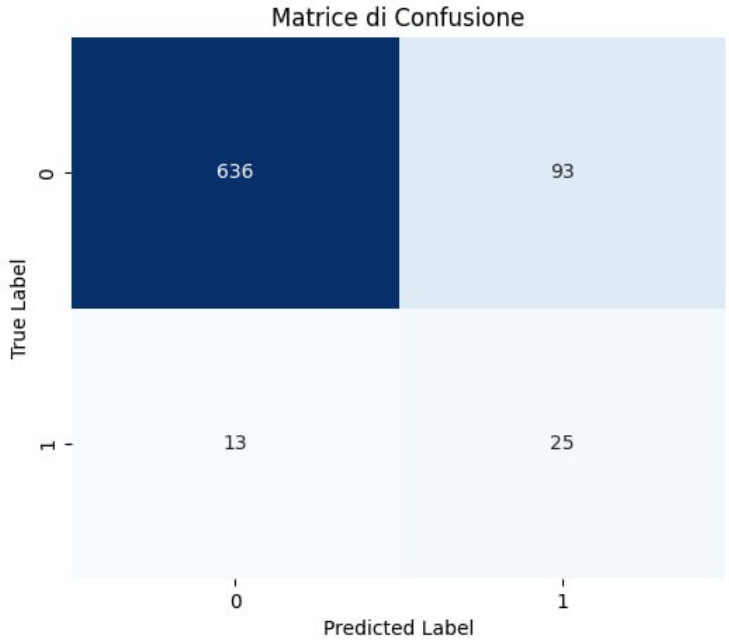

Figura 6: Confusion Matrix

- **Curva ROC**: mostra il compromesso tra True Positive Rate e False Positive Rate.

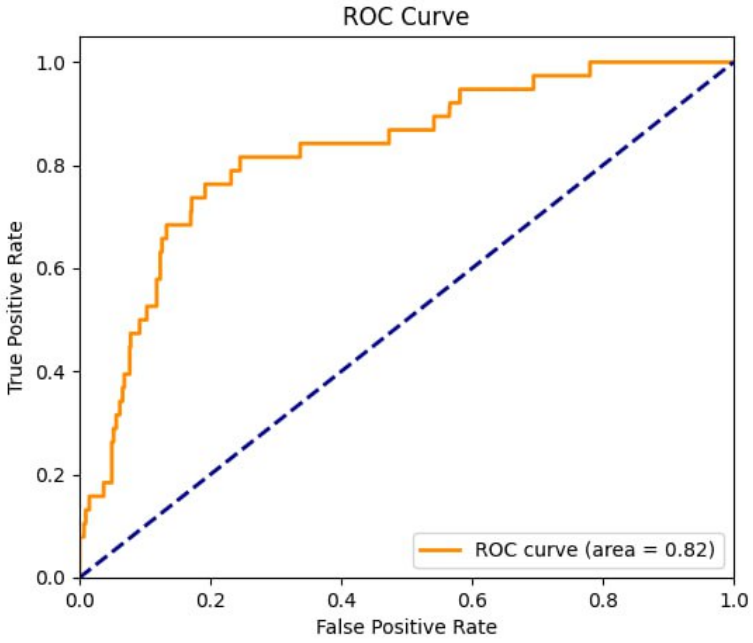

Figura 7: ROC Curve

- **Curva Precision-Recall**: aiuta a valutare il comportamento del modello su dataset sbilanciati.

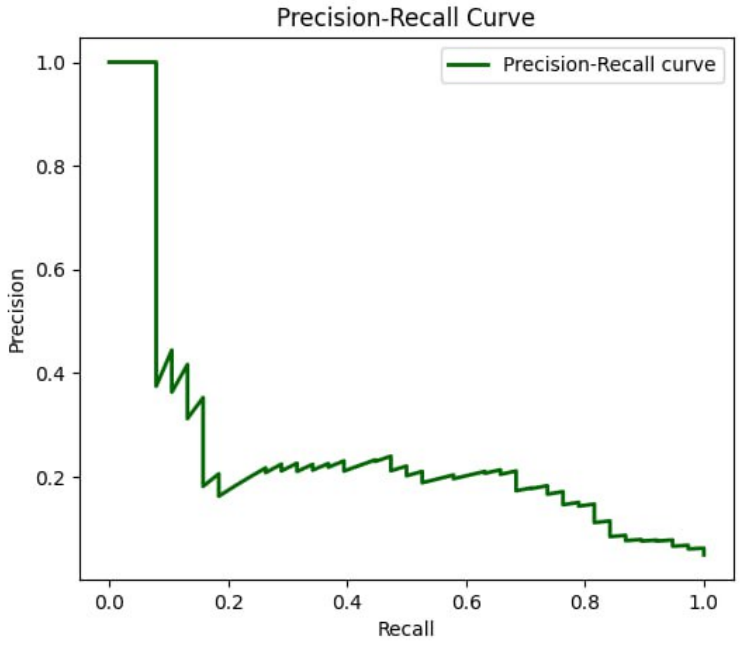

Figura 8: Precision-Recall Curve

- **Importanza delle Feature**: evidenzia le variabili più influenti nelle predizioni.

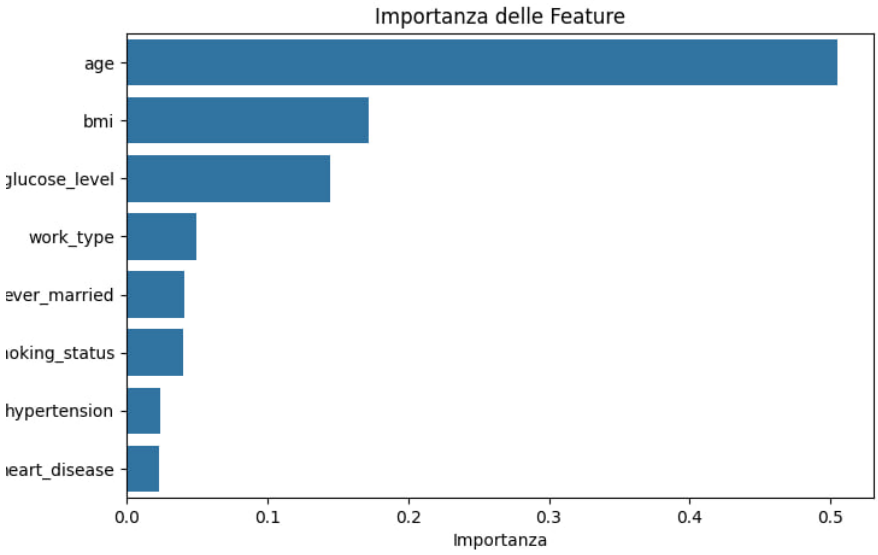

Figura 9: Feature Importance

5.2 Considerazioni sulle metriche

Le metriche finali performate dal modello sul test set sono state:

Listing 1: Metriche finali del modello

```
====== REPORT FINALE DI VALUTAZIONE ======

Accuracy: 0.8618
Precision: 0.2119
Recall: 0.6579
F1-Score: 0.3205
ROC-AUC: 0.8243
Soglia Ottimale: 0.5639
```

L'Accuracy non è stata considerata una metrica principale poiché, essendo la classe *stroke=1* molto meno frequente rispetto alla classe negativa, un modello potrebbe ottenere un'alta accuratezza semplicemente classificando la maggior parte degli esempi come negativi. Per questo motivo, si è preferito analizzare Precision e Recall, con un'attenzione particolare alla Recall, essenziale per ridurre il numero di falsi negativi (ossia pazienti con ictus non identificati dal modello). Dai risultati ottenuti, il modello dimostra una buona capacità discriminante, con un ROC-AUC elevato, indicando che è in grado di differenziare correttamente tra casi positivi e negativi. Tuttavia, la scarsa quantità di dati nel caso della classe positiva rende davvero arduo l'apprendimento dei pattern di classificazione positiva al modello, che di conseguenza, non performa in maniera ottimale. Questo fa capire che

nonostante una ricerca accurata degli iperparametri e un adeguata fase di preprocessing, nonostante l'utilizzo di una tecnica di SMOTEing, il modello fa fatica ad apprendere quando la classe meno rappresentata è quella più critica, soprattutto in contesto medico e pone ancora di più l'attenzione sulla necessità di una maggiore quantità di dati rappresentativi della classe positiva.

5.3 Accorgimenti finali

Dopo una prima fase di sperimentazione, si è deciso di rimuovere SMOTE e di adottare l'opzione *class_weight='balanced'* come iperparametro del Random Forest Classifier per bilanciare la classe target. Questa scelta è stata motivata dai seguenti fattori:

- SMOTE introduce nuovi dati sintetici che, pur bilanciando le classi, possono causare overfitting e alterare la distribuzione naturale dei dati.
- L'uso di *class_weight='balanced'* permette al modello di compensare lo sbilanciamento assegnando pesi maggiori agli esempi della classe meno rappresentata, senza modificare artificialmente il dataset.
- L'analisi delle metriche ha dimostrato che l'utilizzo del solo class weighting garantisce un buon equilibrio tra Recall e Precision senza introdurre distorsioni nel modello.

In sintesi, la valutazione del modello ha confermato la bontà delle scelte effettuate nel preprocessing e nell'addestramento, evidenziando punti di forza e possibili margini di miglioramento per un'eventuale ottimizzazione futura.

6 Repository del Progetto

Di seguito alleghiamo la repository GitHub del progetto descritto: stroke-prediction-ML.

Riferimenti bibliografici

[ea21] Zhu et al. Predicting long-term outcomes in stroke patients using random forest. *Scientific Reports*, 2021.

[ea23] Smith et al. Comparison of machine learning models for stroke prediction. *PMC*, 2023.

[ea24] Johnson et al. Random forest classifiers outperform traditional methods in stroke prediction. *MDPI*, 2024.