

Stroke Prediction

Machine Learning Project by Roberto Fiorenza

27/02/2025



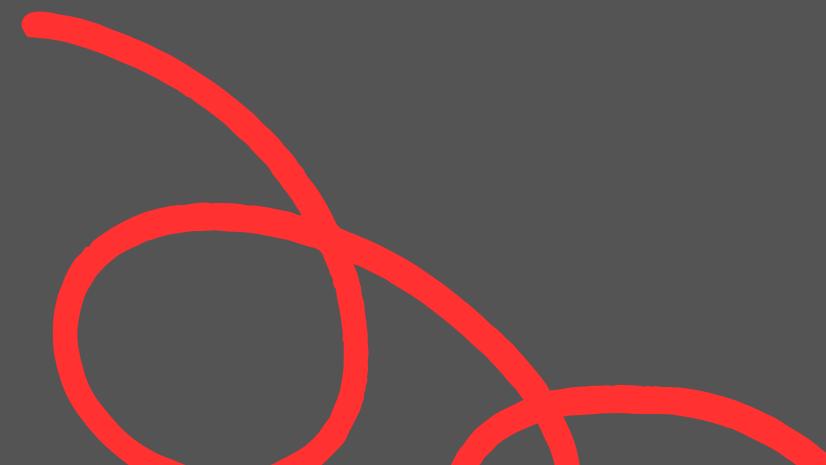


Indice

- 1) Dataset
- 2) Data Analysis
- 3) Data Preparation
- 4) Feature Engineering
- 5) Dataset Balancing
- 6) Modeling



1) Introduzione al Dataset



Introduzione al Dataset

Informazioni: Questo dataset fornisce informazioni cruciali per prevedere il rischio di ictus in base a fattori demografici e clinici. Contiene dati su oltre 5.000 pazienti, inclusi età, ipertensione, malattie cardiache, BMI, livelli di glucosio e abitudini di vita. Un'analisi di questi dati può aiutare a identificare i principali fattori di rischio e migliorare la prevenzione attraverso modelli di machine learning.

Fonte: <https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset>

Dataset caricato correttamente!

	<code>id</code>	<code>gender</code>	<code>age</code>	<code>hypertension</code>	<code>heart_disease</code>	<code>ever_married</code>	<code>work_type</code>	<code>Residence_type</code>	<code>avg_glucose_level</code>	<code>bmi</code>	<code>smoking_status</code>	<code>stroke</code>
<code>0</code>	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
<code>1</code>	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
<code>2</code>	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
<code>3</code>	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
<code>4</code>	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1



Variabili numeriche

- **age:** Età del paziente in anni.
- **avg_glucose_level:** Livello medio di glucosio nel sangue (mg/dL).
- **bmi:** Indice di Massa Corporea (rapporto tra peso e altezza).

Variabili categoriche

- **gender:** Genere del paziente (Male, Female, Other).
- **ever_married:** Indica se il paziente si è mai sposato (Yes, No).
- **work_type:** Tipo di occupazione (Private, Self-employed, Govt_job, Children, Never_worked).
- **Residence_type:** Tipo di residenza (Urban, Rural).
- **smoking_status:** Stato di fumatore (formerly smoked, never smoked, smokes, unknown).
- **hypertension:** Indica se il paziente soffre di ipertensione
- **heart_disease:** Indica se il paziente ha malattie cardiache
- **stroke:** Variabile target, indica se il paziente ha avuto un ictus

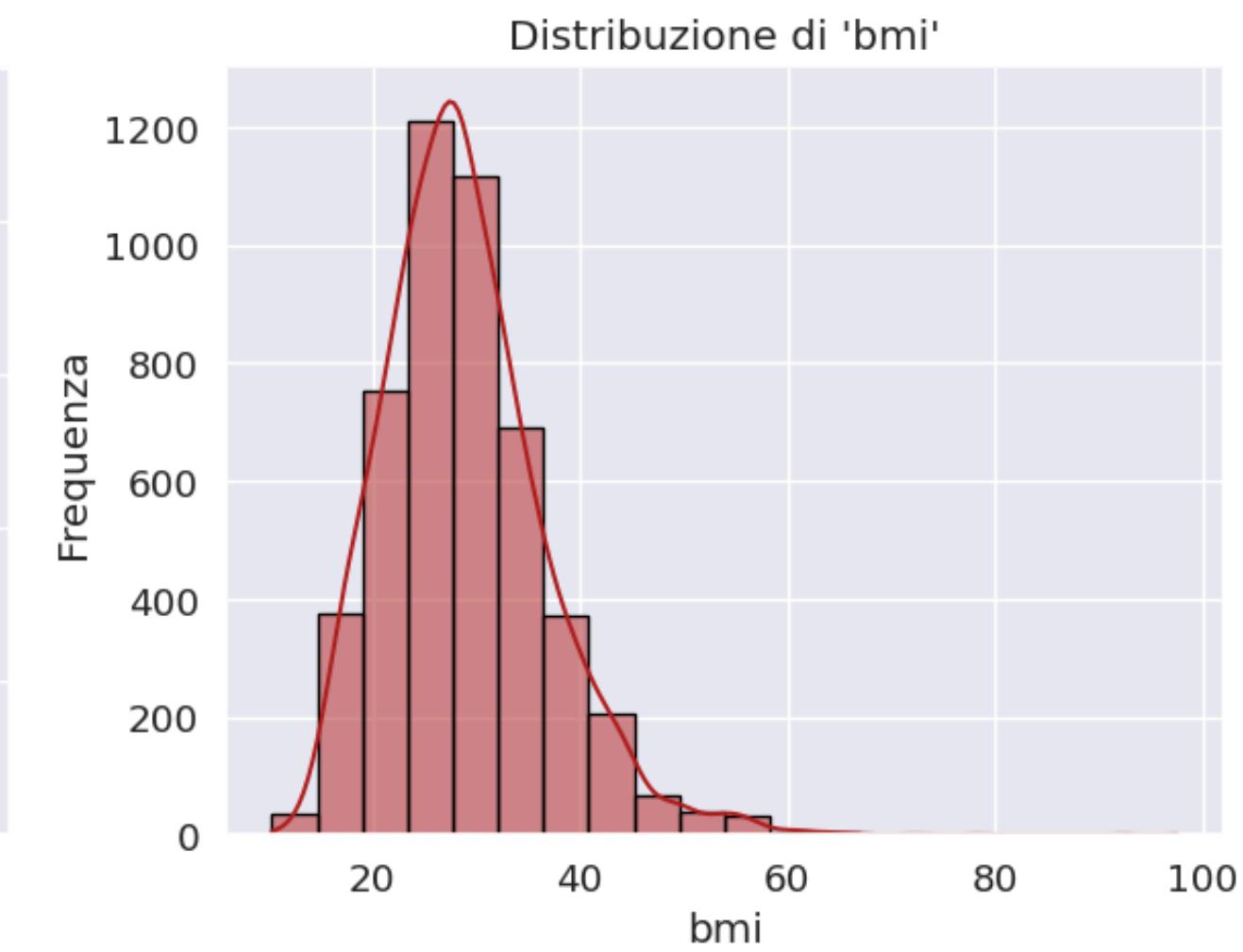
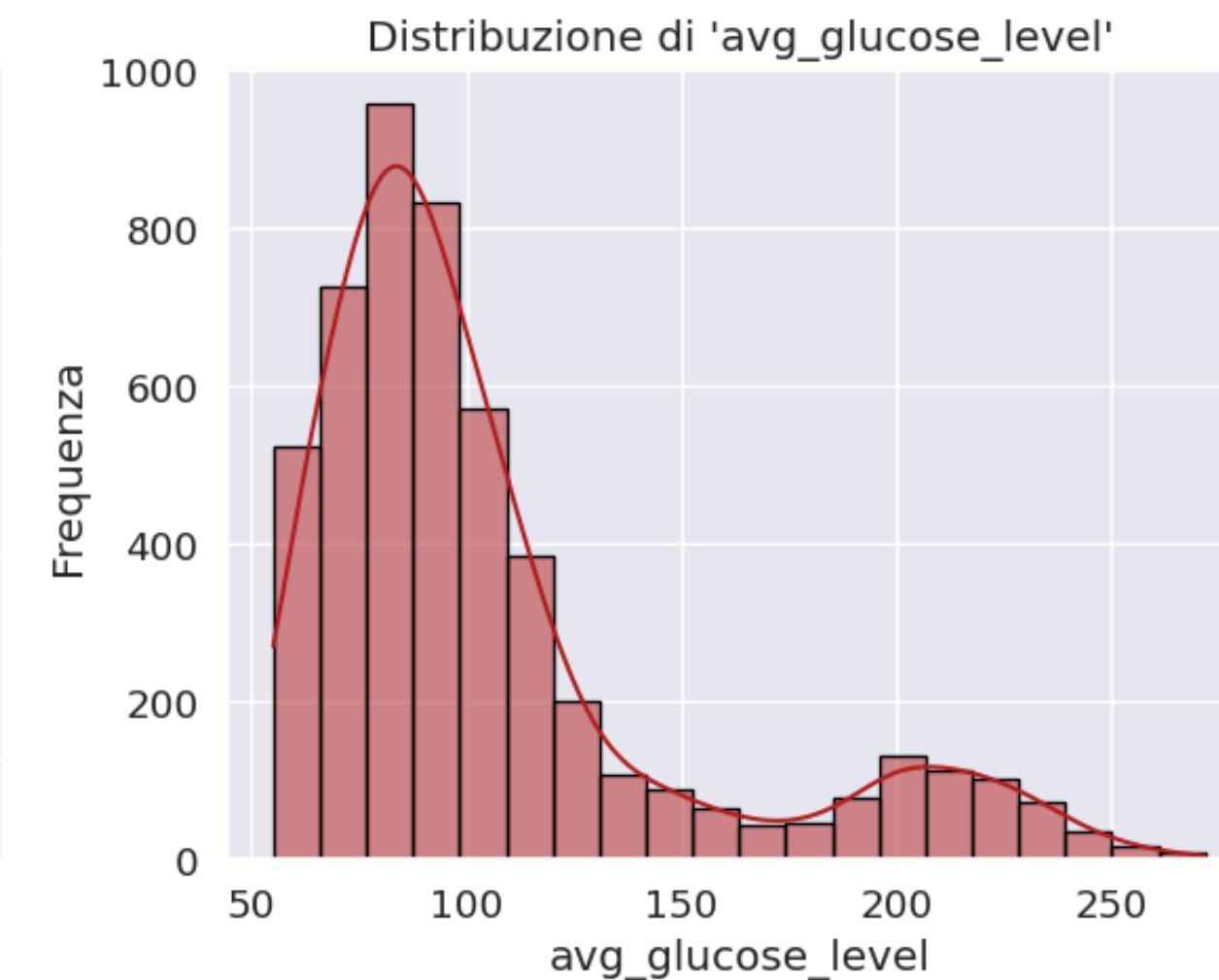
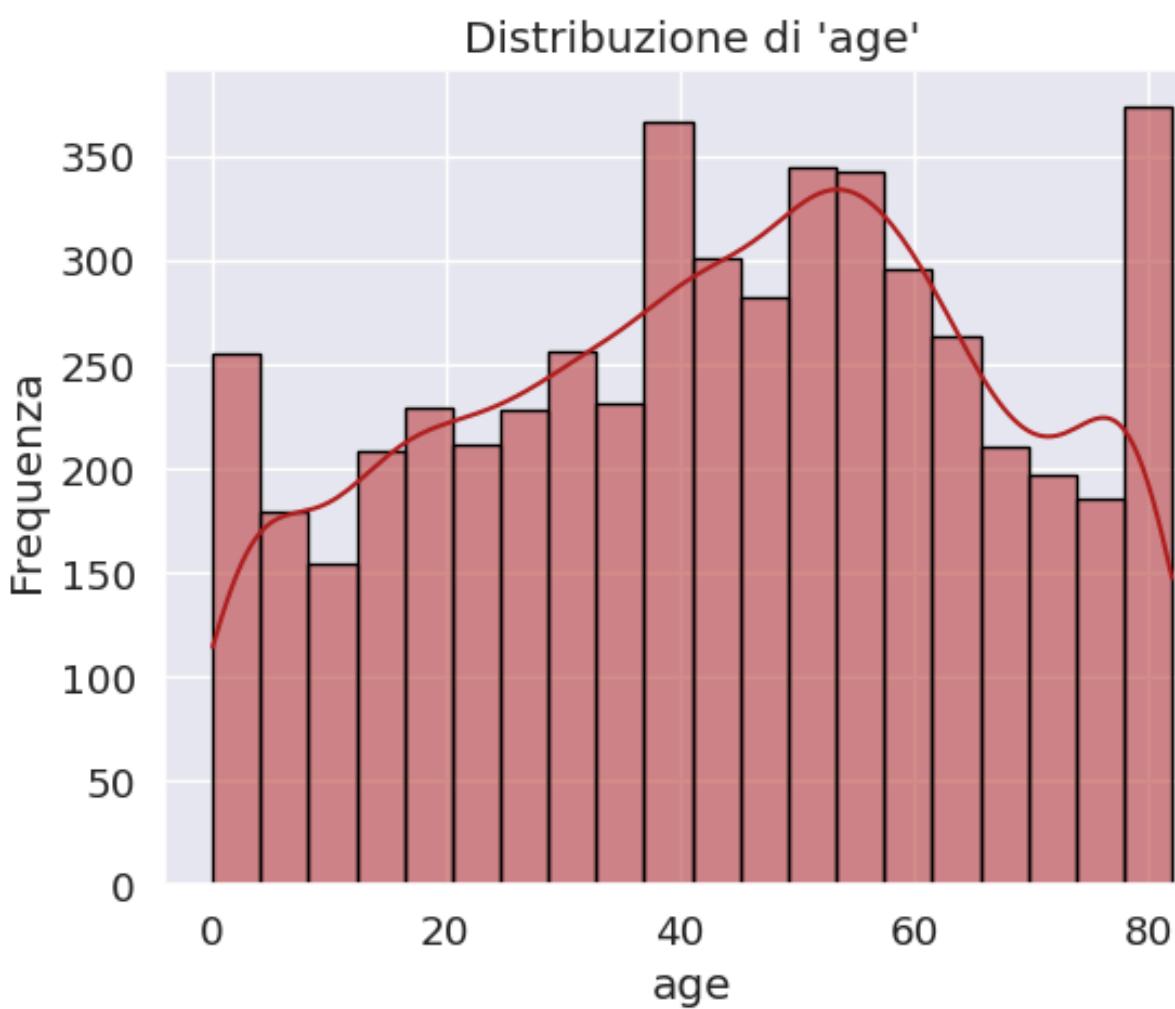


2) Data Analysis

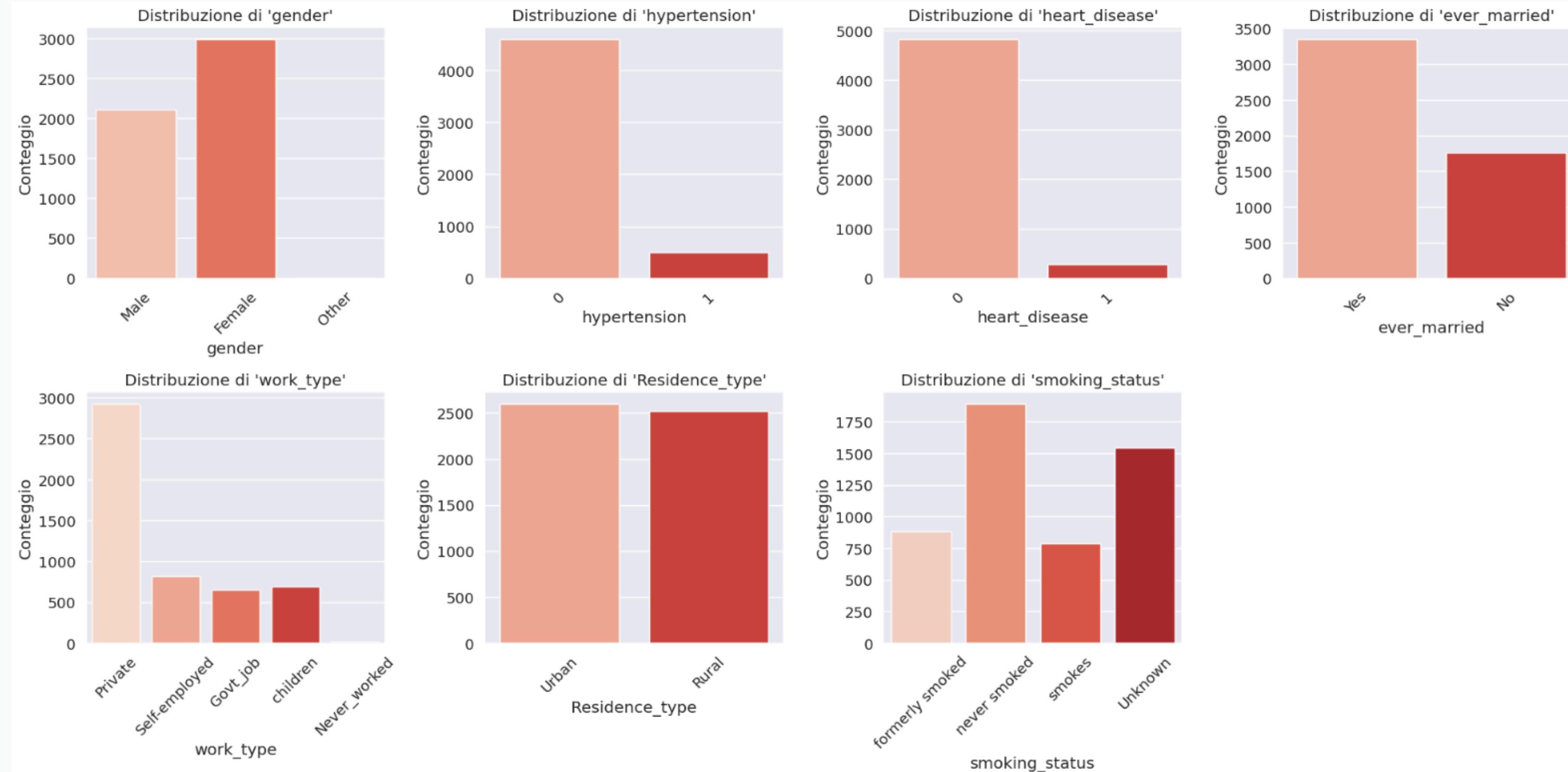


Analisi del dataset:

Istogrammi variabili numeriche

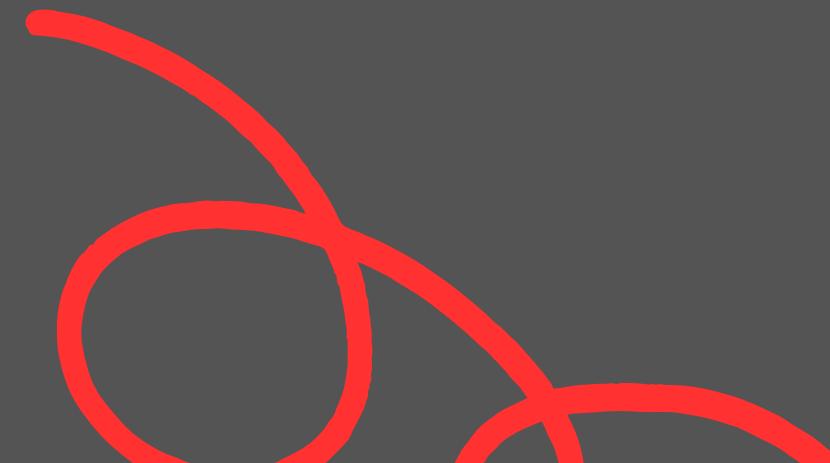


Analisi del dataset: Istogrammi variabili categoriche

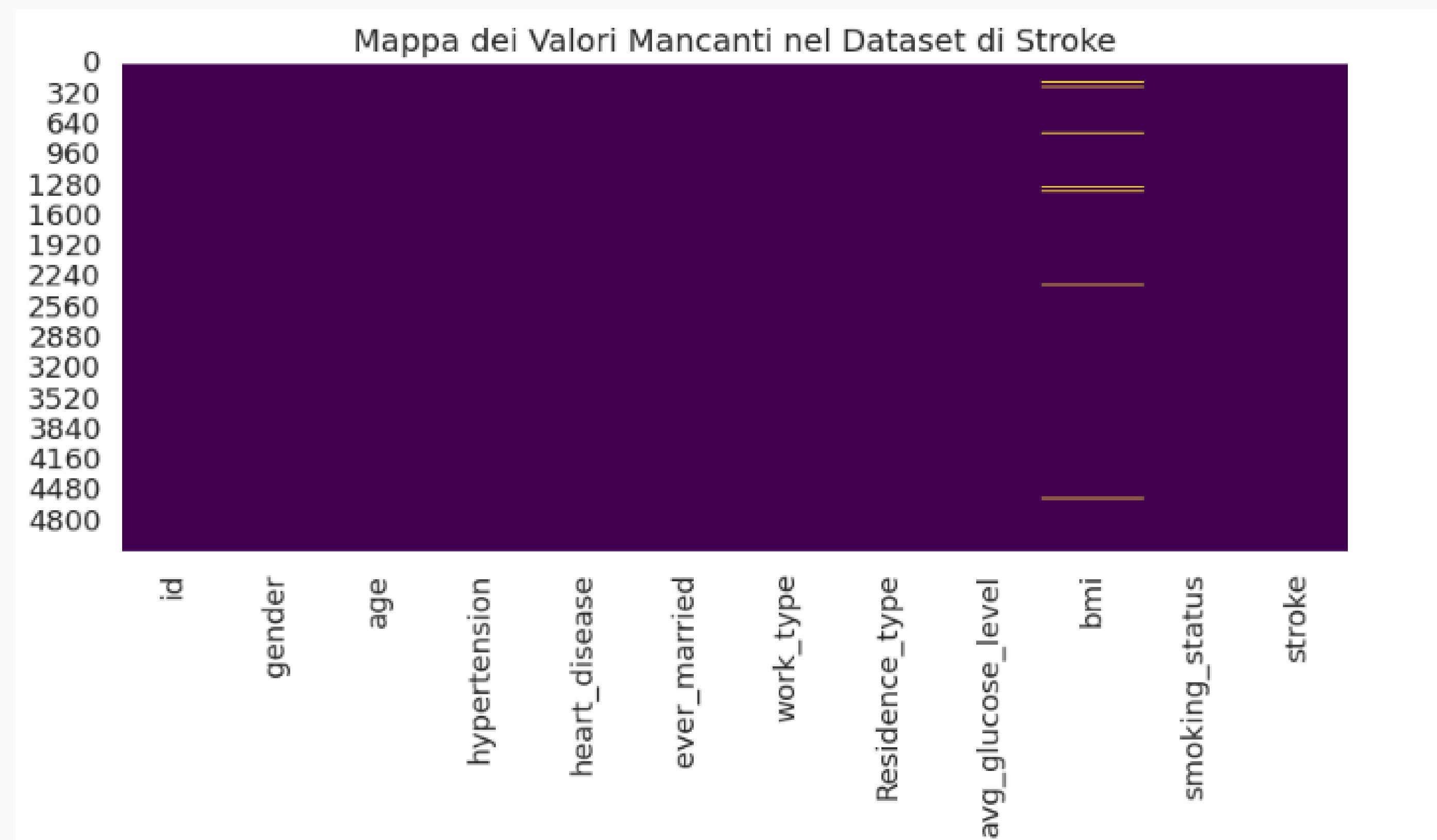




3) Data Preparation



Gestione dei dati mancanti: Heatmap

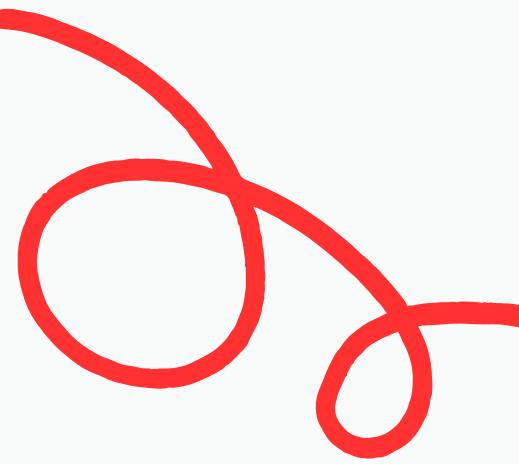




Gestione dei dati mancanti: Heatmap

- L'heatmap dei valori mancanti evidenzia che solo la variabile **BMI** presenta dati nulli.
- Per garantire la qualità del dataset, i valori mancanti di **BMI** vengono sostituiti con la **mediana** della distribuzione.
- La scelta della mediana è preferibile alla media per ridurre l'influenza di valori anomali ed evitare distorsioni nei dati.

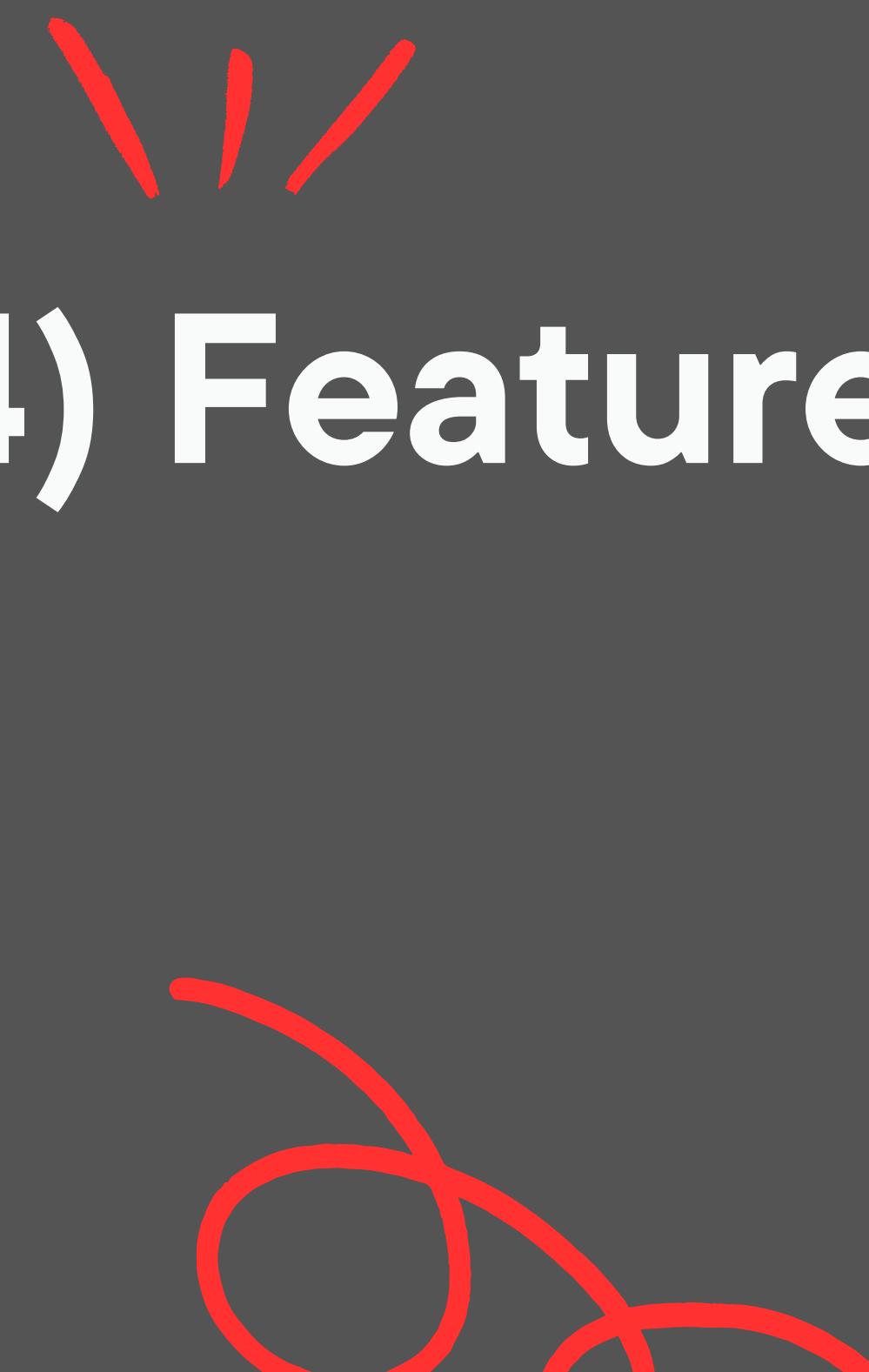
Dopo il preprocessing, il dataset non contiene più valori nulli, garantendo la coerenza delle analisi e dei modelli predittivi.



Rimozioni di feature

La variabile **id** è stata rimossa perché non fornisce informazioni utili per la previsione dell'ictus. Si tratta di un identificativo univoco assegnato a ciascun paziente, privo di correlazione con le altre feature del dataset. Mantenere questa colonna nei dati potrebbe introdurre rumore nel modello senza apportare alcun beneficio predittivo. Inoltre, l'id non è una caratteristica replicabile su nuovi dati, rendendolo inutile per la generalizzazione del modello. La sua eliminazione semplifica il dataset e riduce la dimensionalità senza perdita di informazioni rilevanti.

```
# Rimuovo la colonna 'id'
if 'id' in df.columns:
    df.drop(columns=['id'], inplace=True)
```



4) Feature Engineering

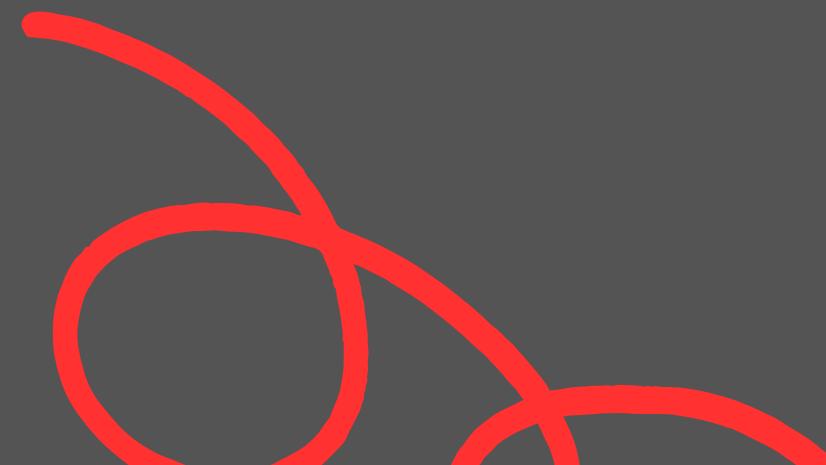


Utilizzo di One-Hot Encoding

- Il dataset contiene variabili categoriche come gender, work_type, smoking_status, che devono essere convertite in formato numerico per essere utilizzate dai modelli di machine learning.
- È stato applicato **One-Hot Encoding**, che trasforma ogni categoria in una serie di colonne binarie, rendendo le informazioni interpretabili dai modelli.
- Questo metodo evita di introdurre relazioni ordinali errate tra le categorie, cosa che potrebbe accadere con una semplice codifica numerica (es. Label Encoding).
- Il risultato è un dataset numerico pronto per l'addestramento del modello, senza perdere informazioni sulle categorie originali.

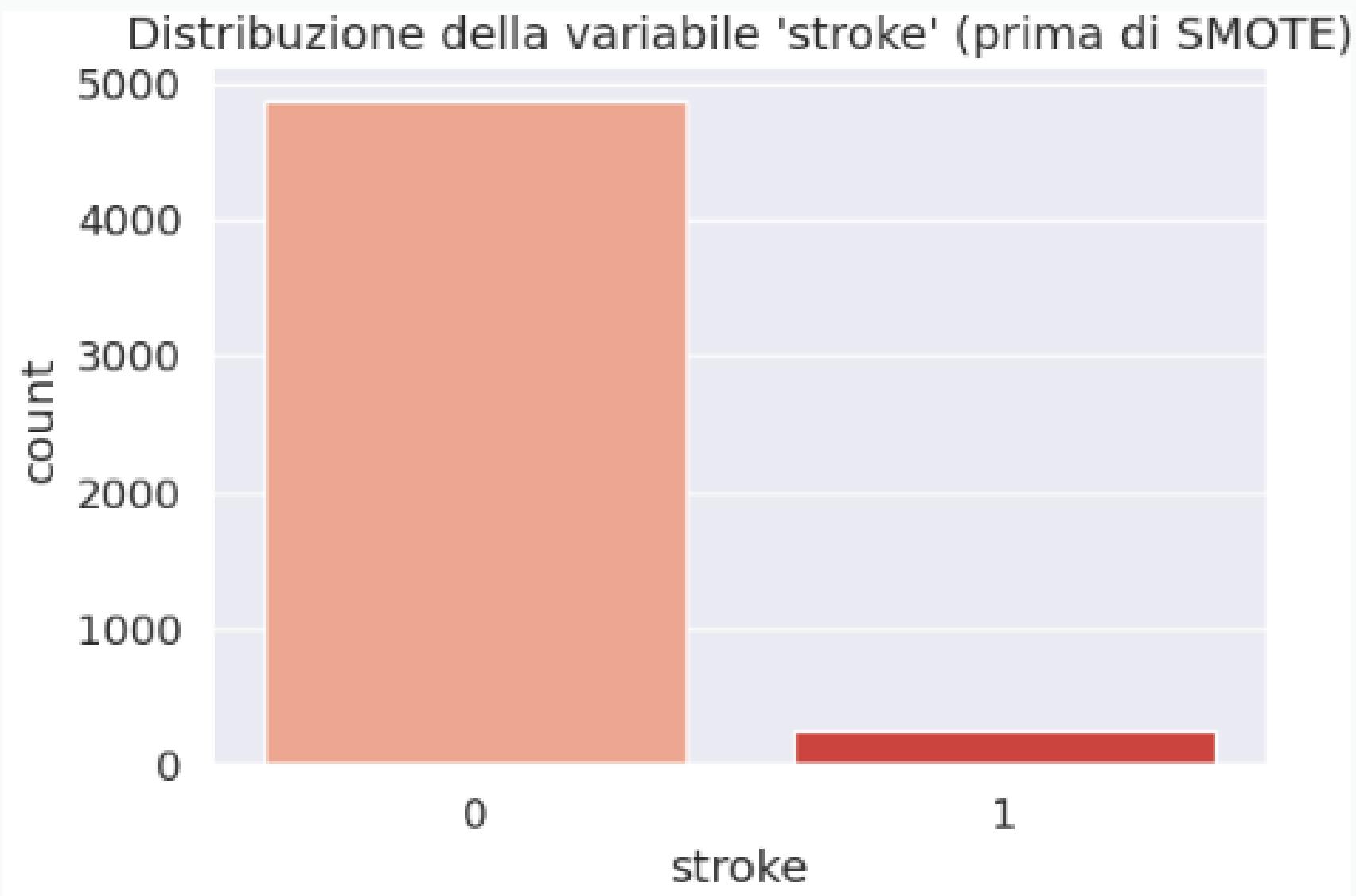


5) Data Balancing



Bilanciamento dei dati: utilizzo di SMOTE

Nel dataset originale, la variabile target stroke è fortemente sbilanciata, con un numero molto maggiore di pazienti senza ictus rispetto a quelli che ne hanno avuto uno. Questo squilibrio può portare i modelli di machine learning a imparare una forte preferenza per la classe dominante, riducendo la capacità di identificare correttamente i casi di ictus. Per affrontare questo problema, è stato applicato **SMOTE** (Synthetic Minority Over-sampling Technique), una tecnica di oversampling che genera nuovi esempi sintetici della classe minoritaria invece di replicare semplicemente i dati esistenti. Questo approccio aiuta il modello a imparare meglio i pattern legati ai casi di ictus, migliorando la capacità di generalizzazione senza introdurre overfitting dovuto alla duplicazione dei dati.



5) Modeling



Modeling: Scelta dei modelli

Decision Tree Classifier

- Funziona suddividendo lo spazio delle feature in regioni omogenee tramite una serie di decisioni binarie, costruendo una struttura ad albero dove ogni nodo rappresenta una condizione sulle feature.
- Modello flessibile, in grado di catturare relazioni non lineari nei dati.

Random Forest Classifier

- Combina molteplici alberi decisionali addestrati su sottocampioni casuali del dataset e aggrega le loro predizioni per ottenere un risultato più robusto e generalizzabile.
- Utilizzato per migliorare la stabilità e ridurre l'overfitting rispetto a un singolo albero decisionale.

Logistic Regression

- Modello statistico che utilizza la funzione sigmoide per calcolare la probabilità di appartenenza a una classe, assegnando una soglia (es. 0.5) per determinare la classificazione binaria.
- Interpretabilità elevata e adatto per problemi in cui le feature hanno relazioni lineari con la variabile target.

K-Nearest Neighbors (KNN)

- Scelto per testare un approccio non parametrico, che classifica in base alla vicinanza ai dati di training.
- Funziona classificando un nuovo punto in base alla maggioranza delle classi tra i K punti più vicini nel dataset, misurando la distanza (es. Euclidea) tra le osservazioni.

XGBoost Classifier

- Modello avanzato basato su gradient boosting, noto per la sua elevata accuratezza in problemi di classificazione.
- Può essere molto costoso in termini di tempo di addestramento e risorse computazionali, specialmente su dataset grandi



Metriche di performance: definizioni

Per valutare le prestazioni dei modelli di machine learning nel classificare correttamente i casi di ictus, sono state utilizzate diverse metriche:

 **Accuracy** misura la proporzione di predizioni corrette sul totale, ma può risultare fuorviante in dataset sbilanciati, dove la classe maggioritaria domina i risultati.

 **Precision** indica la percentuale di predizioni positive che sono effettivamente corrette, ovvero quanto il modello evita falsi positivi.

 **Recall** misura la capacità di catturare tutti i veri positivi, indicando la sensibilità del modello nel riconoscere i casi di ictus.

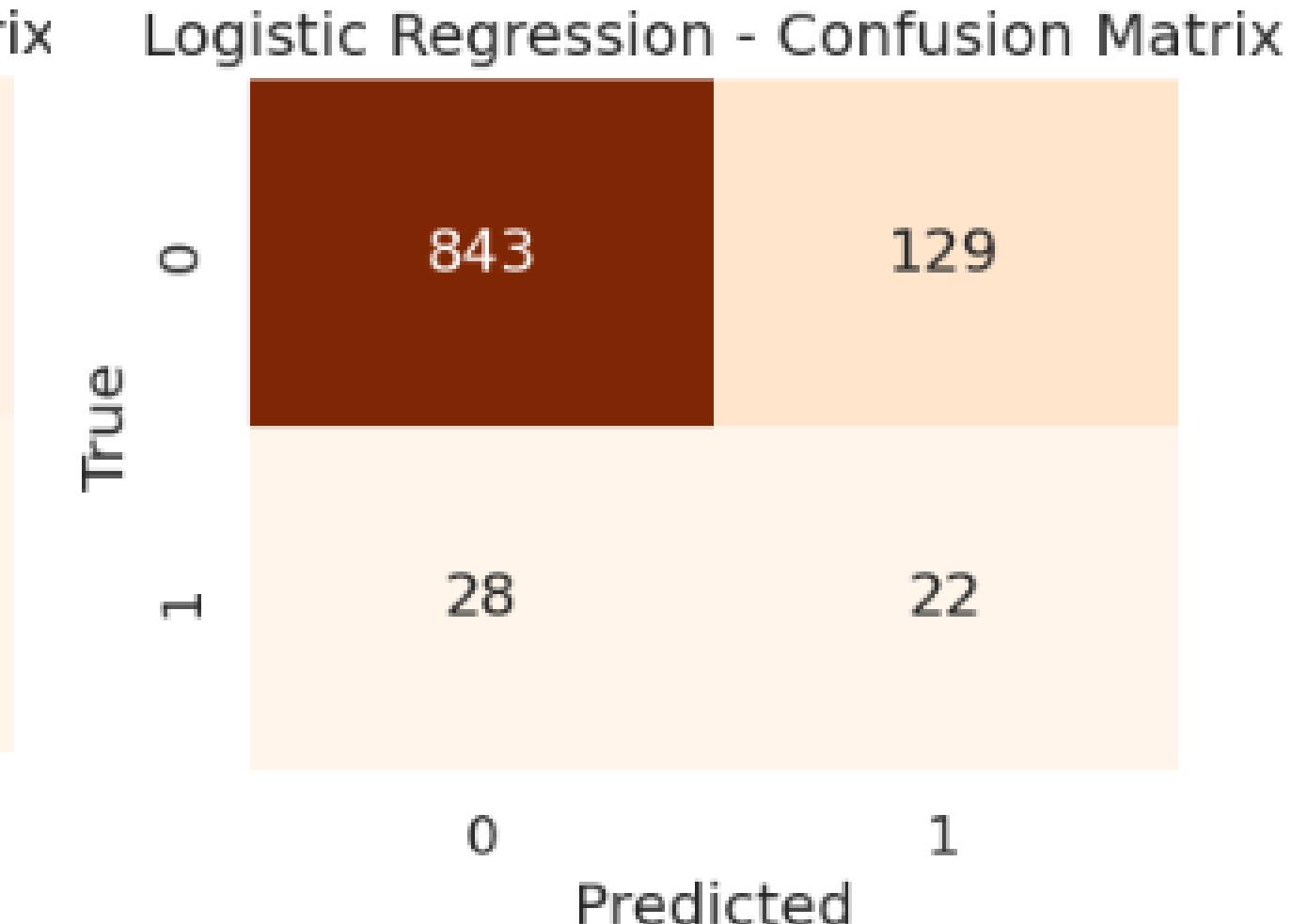
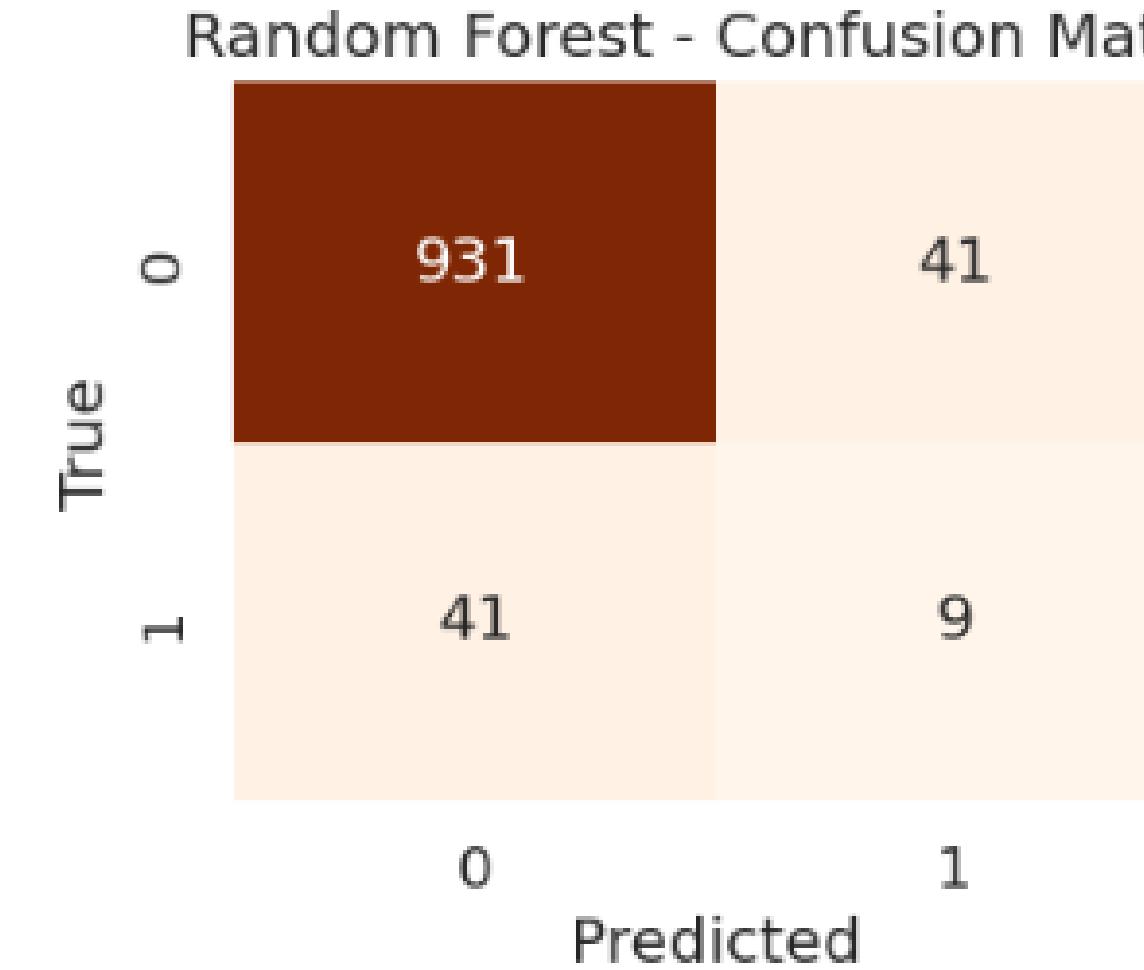
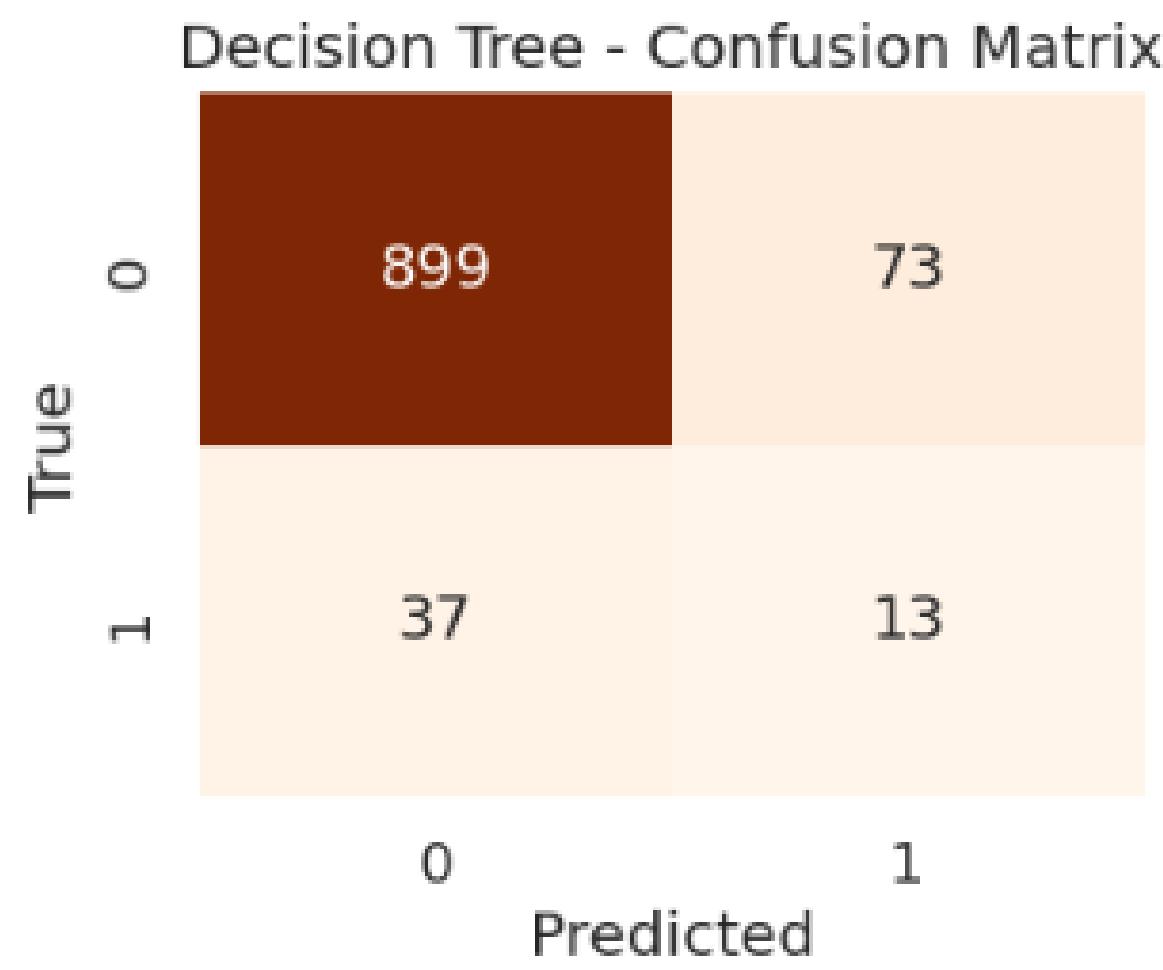
 **F1-score** è la media armonica tra precision e recall e viene utilizzato quando è necessario un compromesso tra le due metriche, particolarmente utile nei dataset sbilanciati.

Metriche di performance: Confusion Matrix

--- Decision Tree ---
Accuracy: 0.892
Precision: 0.921
Recall: 0.892
F1 Score: 0.906

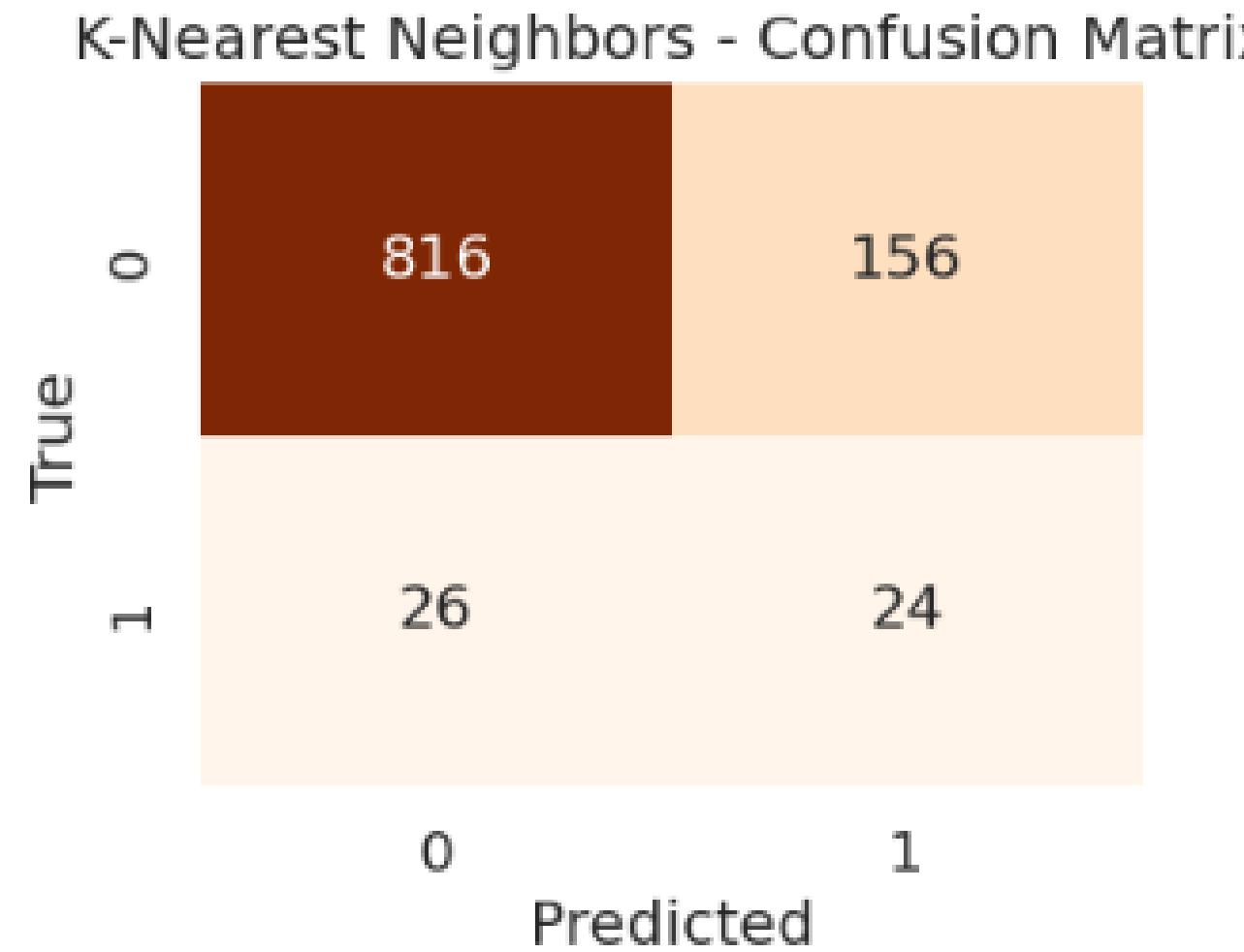
--- Random Forest ---
Accuracy: 0.920
Precision: 0.920
Recall: 0.920
F1 Score: 0.920

--- Logistic Regression ---
Accuracy: 0.846
Precision: 0.928
Recall: 0.846
F1 Score: 0.881

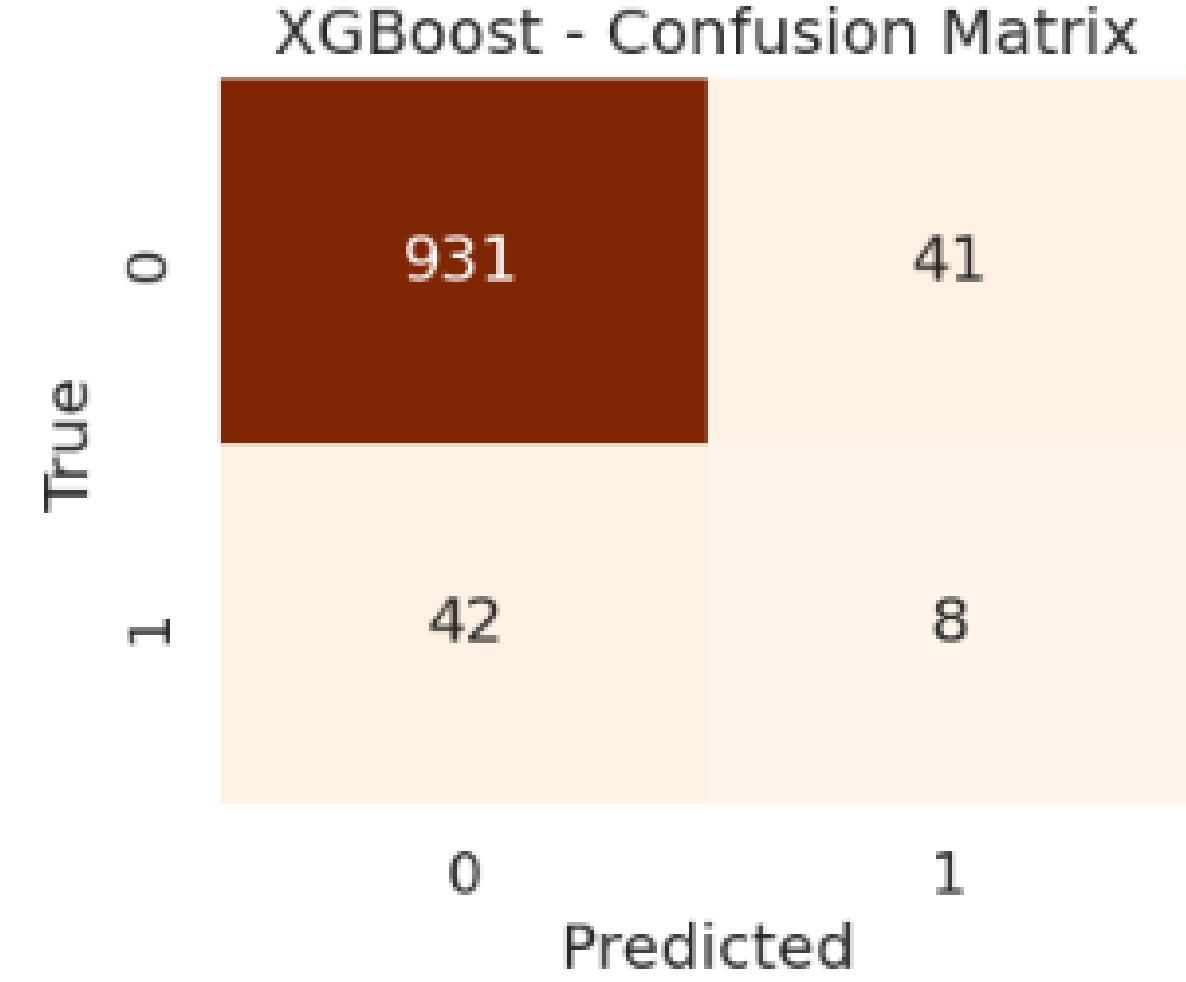


Metriche di performance: Confusion Matrix

--- K-Nearest Neighbors ---
Accuracy: 0.822
Precision: 0.928
Recall: 0.822
F1 Score: 0.866

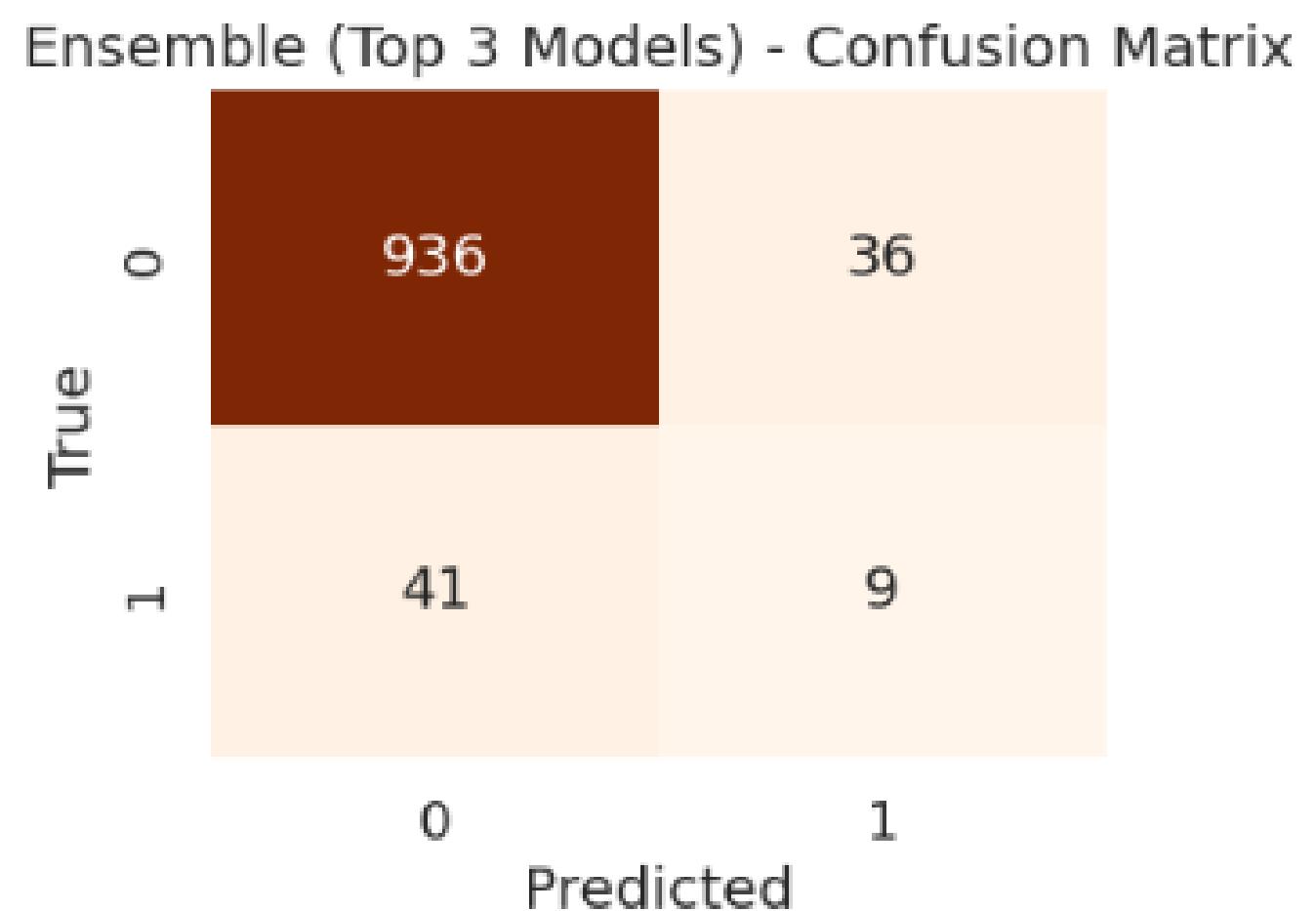


--- XGBoost ---
Accuracy: 0.919
Precision: 0.918
Recall: 0.919
F1 Score: 0.918



Metriche di performance: Confusion Matrix

```
Modelli selezionati per l'ensemble:  
Random Forest: 0.920  
XGBoost: 0.919  
Decision Tree: 0.892  
  
== Ensemble Majority Vote Performance (Top 3 Modelli) ==  
Accuracy: 0.925  
Precision: 0.921  
Recall: 0.925  
F1 Score: 0.923
```

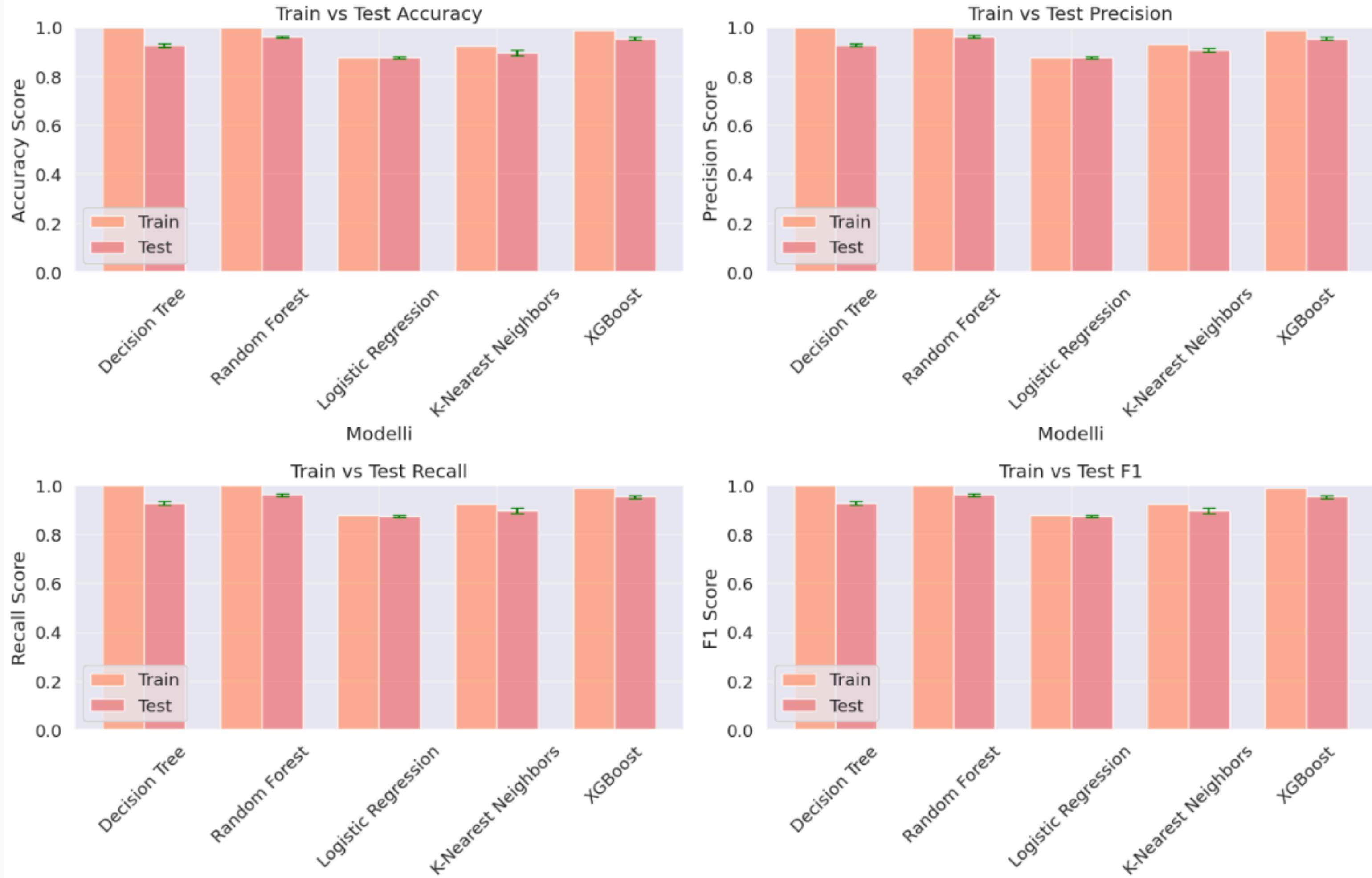


Modeling: K-fold e verifica di overfitting

La **K-Fold Cross-Validation** è una tecnica di validazione che suddivide il dataset in K sottoinsiemi (o folds) di dimensioni uguali. Il modello viene addestrato K volte, utilizzando in ciascuna iterazione K-1 folds per il training e il fold rimanente per il test. Questo processo viene ripetuto fino a che ogni fold non è stato utilizzato esattamente una volta come test set. Alla fine, le metriche di valutazione vengono mediante su tutte le iterazioni, fornendo una stima più affidabile della performance del modello.

Nel nostro progetto, la K-Fold Cross-Validation è stata utilizzata per ridurre il rischio di overfitting e garantire che il modello generalizzasse bene su dati non visti. Rispetto a una semplice suddivisione **Train/Test (80/20)**, la cross-validation fornisce una valutazione più robusta, evitando che le prestazioni dipendano troppo dalla scelta specifica dei dati di training e test. Questo è particolarmente utile quando si lavora con dataset relativamente piccoli e sbilanciati come quello dell'ictus, dove ogni campione è prezioso per migliorare l'accuratezza del modello.

Modeling: K-fold e verifica di overfitting



Risultati Decision Tree:
Test Accuracy: 0.926 (± 0.007)
Test Precision: 0.927 (± 0.007)
Test Recall: 0.926 (± 0.007)
Test F1: 0.926 (± 0.007)

Differenze Train-Test (possibile overfitting se > 0.1):
Accuracy Diff: 0.074
F1 Diff: 0.074

Validazione del modello: Random Forest

Risultati Random Forest:
Test Accuracy: 0.960 (± 0.006)
Test Precision: 0.960 (± 0.006)
Test Recall: 0.960 (± 0.006)
Test F1: 0.960 (± 0.006)

Differenze Train-Test (possibile overfitting se > 0.1):
Accuracy Diff: 0.040
F1 Diff: 0.040

Validazione del modello: Logistic Regression

Risultati Logistic Regression:
Test Accuracy: 0.874 (± 0.004)
Test Precision: 0.875 (± 0.004)
Test Recall: 0.874 (± 0.004)
Test F1: 0.874 (± 0.004)

Differenze Train-Test (possibile overfitting se > 0.1):
Accuracy Diff: 0.001
F1 Diff: 0.001

Validazione del modello: K-Nearest Neighbors

Risultati K-Nearest Neighbors:
Test Accuracy: 0.895 (± 0.011)
Test Precision: 0.908 (± 0.008)
Test Recall: 0.895 (± 0.011)
Test F1: 0.895 (± 0.012)

Differenze Train-Test (possibile overfitting se > 0.1):
Accuracy Diff: 0.026
F1 Diff: 0.026

Validazione del modello: XGBoost

Risultati XGBoost:
Test Accuracy: 0.953 (± 0.006)
Test Precision: 0.953 (± 0.005)
Test Recall: 0.953 (± 0.006)
Test F1: 0.953 (± 0.006)

Differenze Train-Test (possibile overfitting se > 0.1):
Accuracy Diff: 0.034
F1 Diff: 0.034

Conclusioni e punti di miglioramento

L'analisi ha sviluppato un sistema predittivo per l'ictus utilizzando un approccio metodologico completo e rigoroso. Il progetto ha affrontato lo sbilanciamento delle classi tramite SMOTE, implementato e confrontato diversi algoritmi di machine learning, e migliorato le performance attraverso un ensemble di modelli. La cross-validation ha confermato la robustezza dell'approccio, permettendo di identificare e prevenire problemi di overfitting.

Punti di miglioramento

- **Hyperparameter tuning:** Il codice utilizza prevalentemente i parametri di default per i modelli. Un processo di ottimizzazione degli iperparametri (es. GridSearchCV o RandomizedSearchCV) potrebbe migliorare significativamente le performance.
- **Analisi delle correlazioni:** Sarebbe utile aggiungere un'analisi delle correlazioni tra le variabili per identificare dipendenze e potenziali ridondanze.
- **Analisi di specifici sottogruppi:** Potrebbe essere interessante analizzare le performance del modello per diversi sottogruppi demografici o clinici, permettendo di identificare se il sistema predittivo funziona uniformemente su diverse popolazioni.