DEVELHOPE

Machine Learning:

Predizione del fallimento aziendale

Team:
Cataldo Ferrara
Federico D'Ubaldi
Giuseppe Ponzo
Viviana Di Maio

18 Giugno, 2025

Obiettivo

Prevedere con accuratezza se un'**azienda** andrà in **bancarotta** utilizzando **tecniche** di **classificazione avanzate**

Focus → ridurre i **Falsi Negativi**

Pipeline del progetto

1.
EDA

2.
Preprocessing con gestione outlier

3.
Bilanciamento Classi

4.
Modellazione con 3 algoritmi

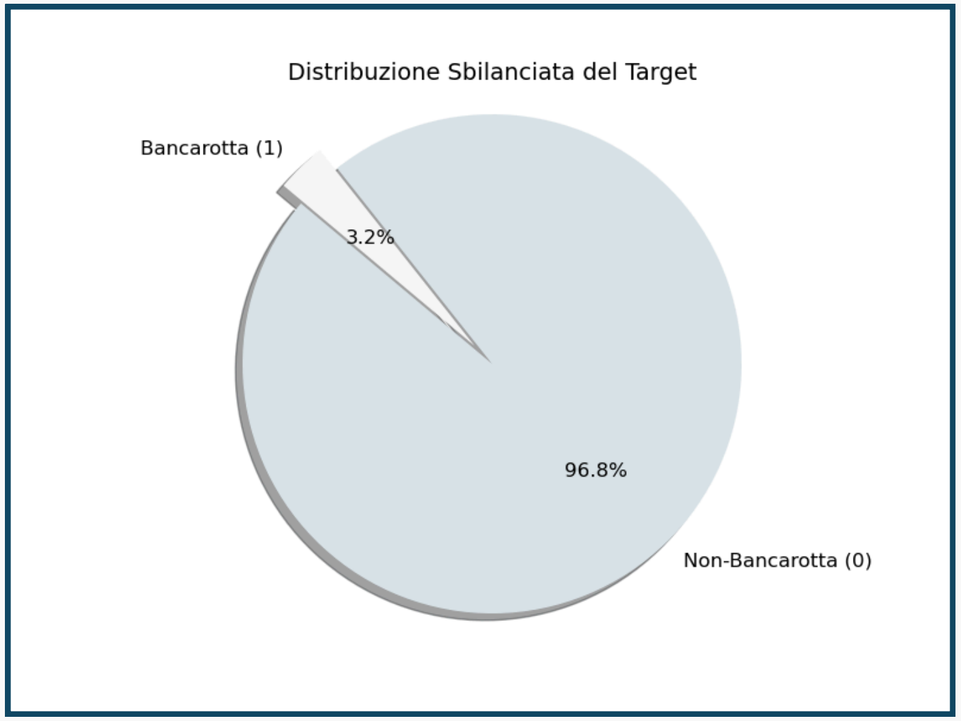
5.
Modellazione con 5 algoritmi

6.
Tuning soglie e iperparametri

7.
Costruzione Ensemble Voting Classifier

Dataset utilizzato

Il dataset, costituito da dati economico-finanziari, presenta un class imbalance nella variabile **target**



Conseguenze dello sbilanciamento

Elevato **rischio** per i modelli di classificare tutte le osservazioni nella Classe dominante (0)

Gestione dello sbilanciamento

Esistono **varie tecniche** per rimediare a questo problema, come:

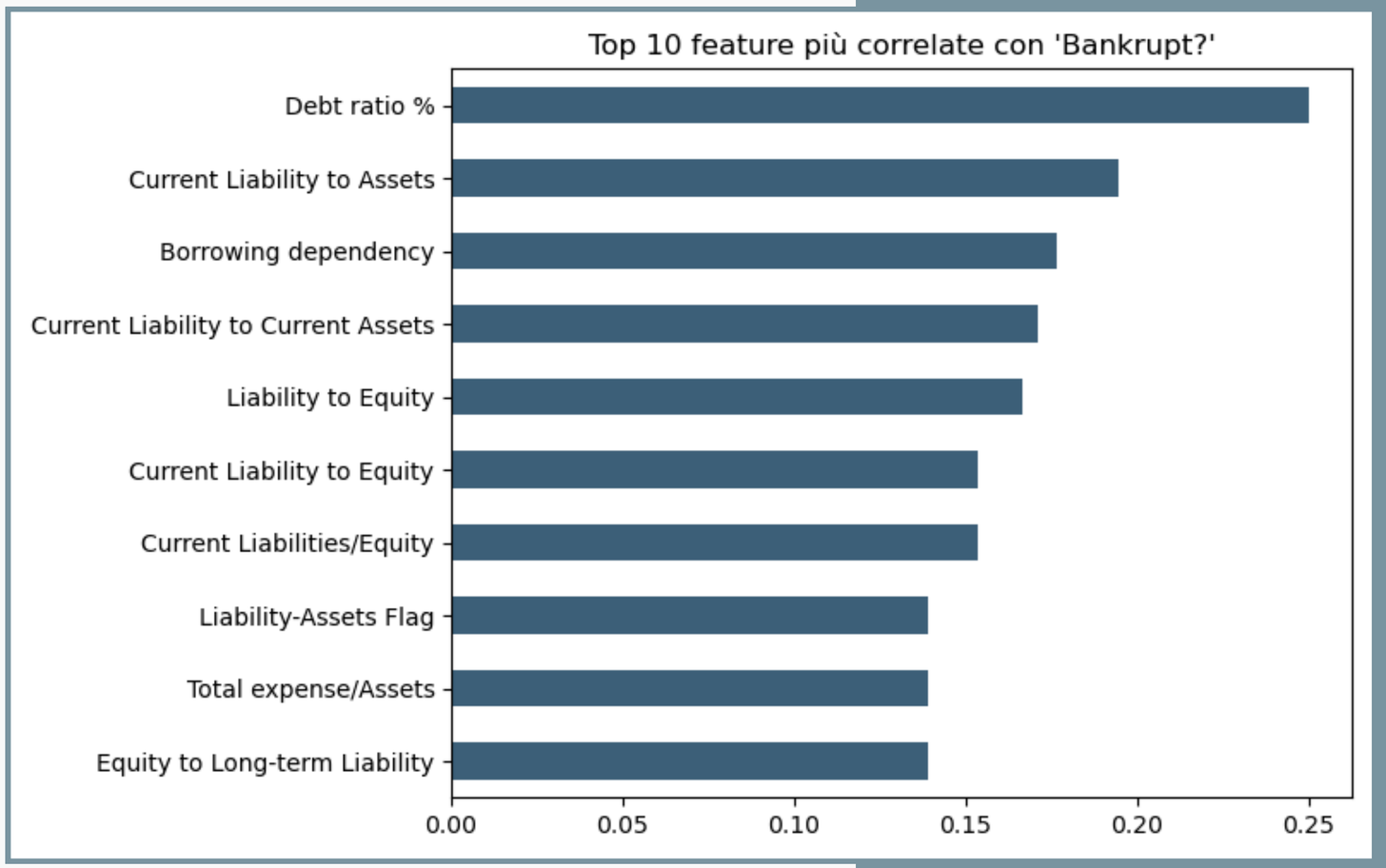
- Oversampling
- Undersampling
- Tecniche ibride
- Class weight
- Thresholds

Analisi Esplorativa

Durante l'**EDA** è stato osservato:

- Nessun **valore mancante**
- Tutte le **colonne** sono **numeriche**
- **Outlier** estremi in alcune variabili
- **Forti correlazioni** tra alcune feature

Sono state identificate le **10 variabili** più rilevanti per la predizione del **target**, selezionate sulla base delle **correlazioni** più elevate con la variabile '**Bankrupt?**'



Logistic Regression

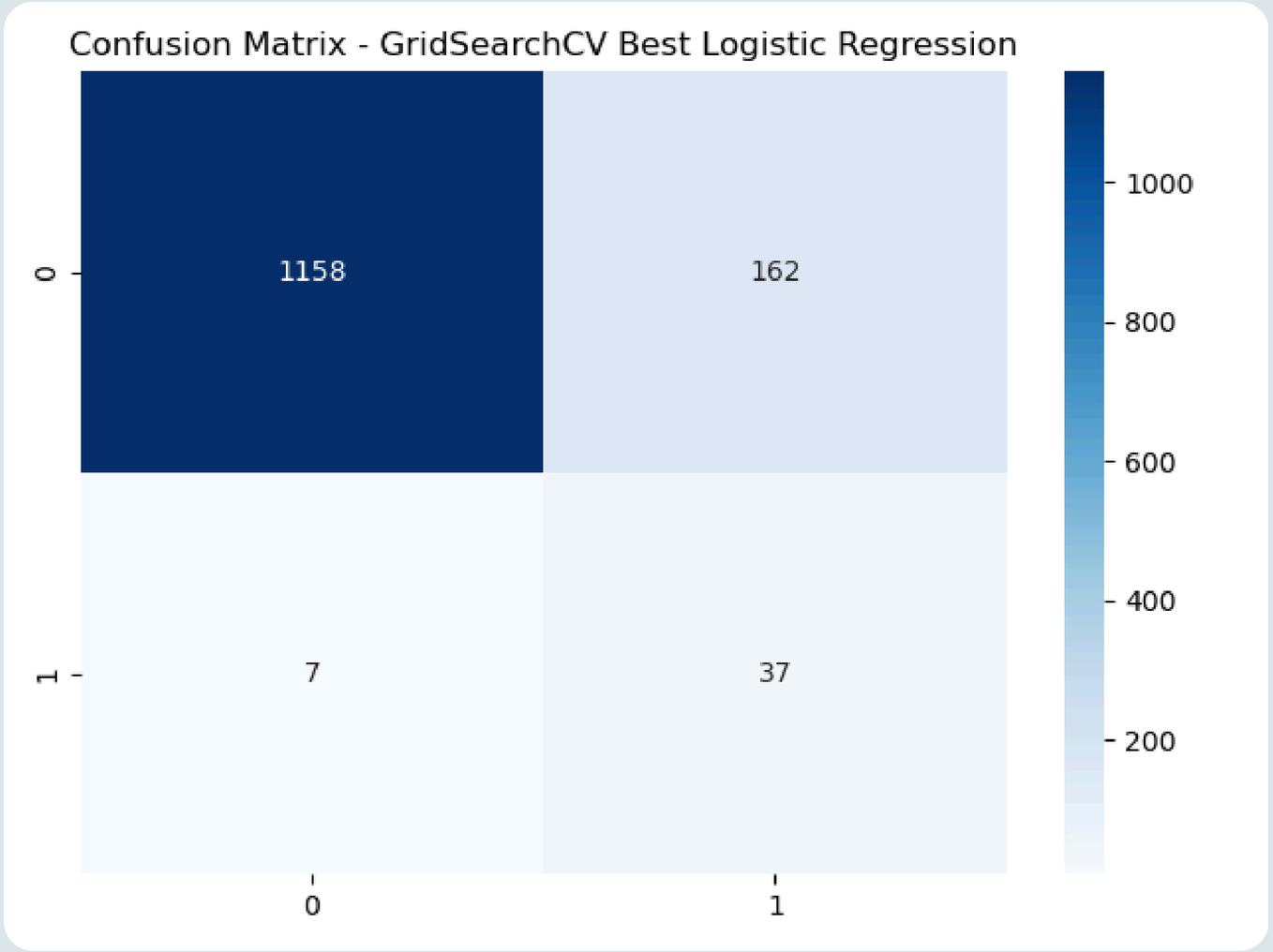
Il primo modello testato è stato la **Logistic Regression** in varie configurazioni:

- La versione base del modello è stata pessima sul riconoscere i fallimenti

Quindi è stato fatto il tuning con:

- **SMOTE** → per gestire lo sbilanciamento
- **StandardScaler** → per normalizzare le feature
- **GridSearchCV → p**er ottimizzare gli iperparametri tramite cross-validation

Metrica	Classe 0	Classe 1
Precision	0.99	0.19
Recall	0.88	0.84
F1-Score	0.93	0.30
Support (n. esempi)	1320	44



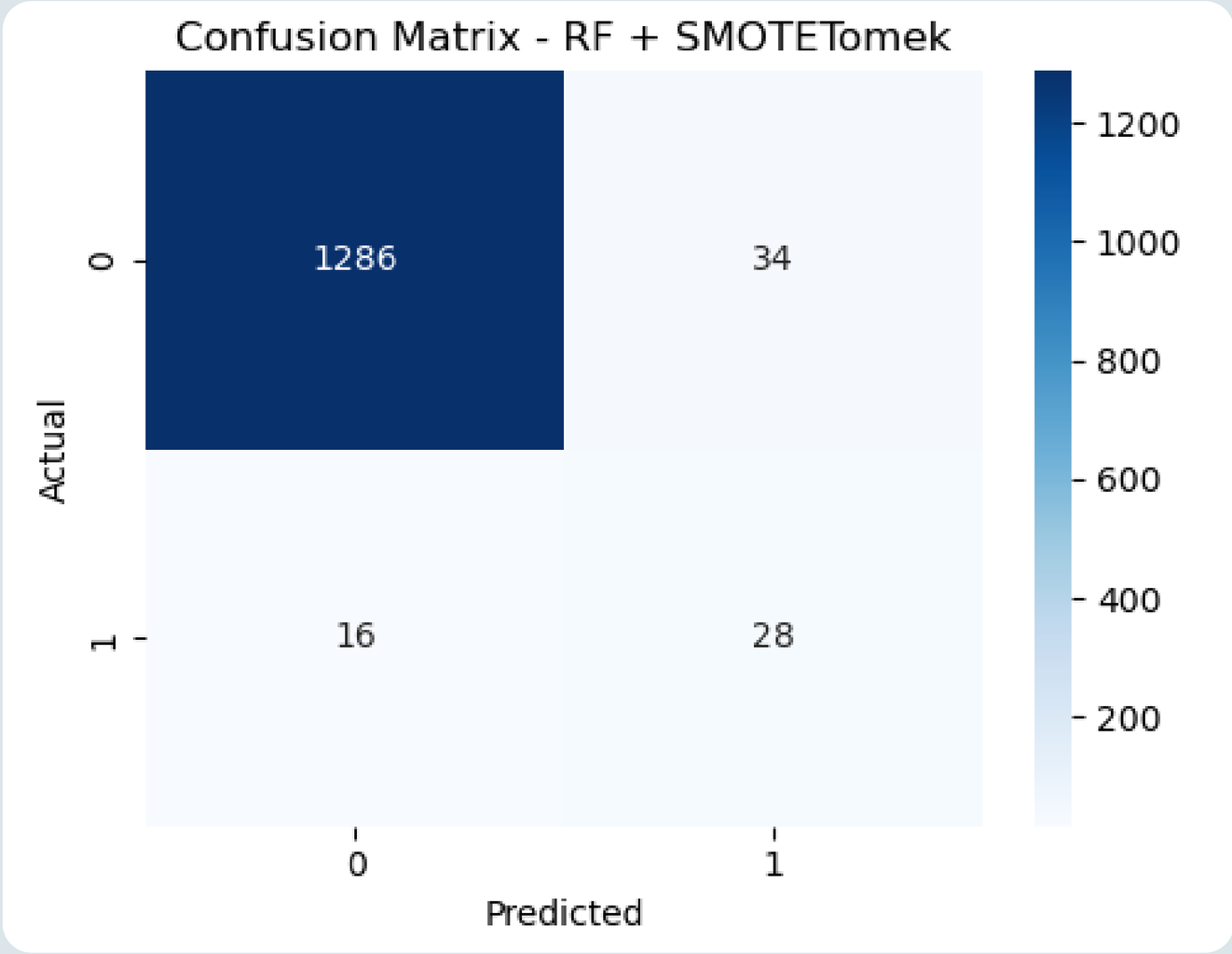
Random Forest

Il secondo modello testato è stato **Random Forest**, noto per la sua robustezza e capacità di gestire dataset complessi.

Abbiamo effettuato una fase di ottimizzazione con:

- **SMOTETomek** → combina l'oversampling sintetico (SMOTE) con un'operazione di pulizia dei dati (Tomek Links) per rimuovere esempi ambigui
- **StandardScaler** → per garantire uniformità tra le feature
- **GridSearchCV** → per trovare la configurazione ottimale degli iperparametri

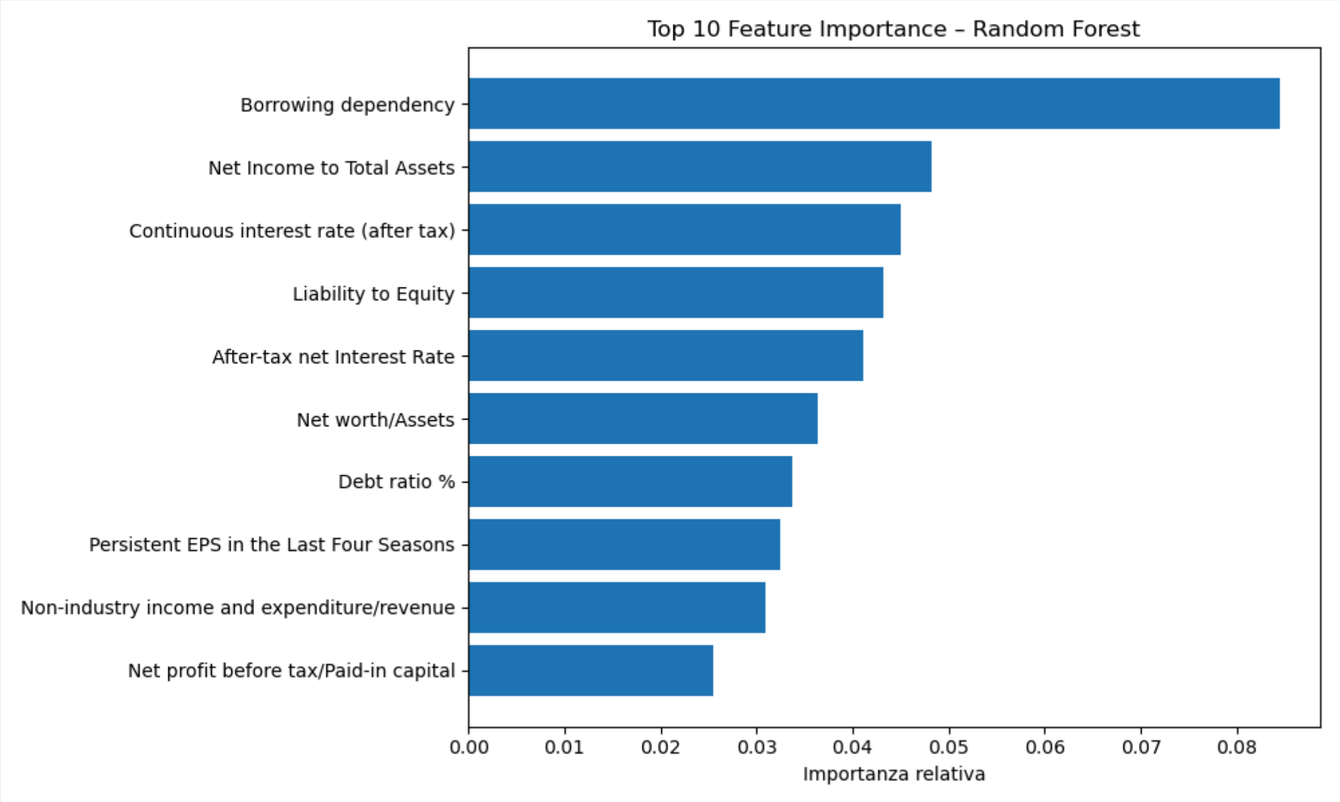
Metrica	Classe 0	Classe 1
Precision	0.99	0.45
Recall	0.97	0.64
F1-Score	0.98	0.53
Support (n. esempi)	1320	44



Feature Importance - Random Forest

Il grafico mostra le 10 variabili più influenti nella classificazione delle aziende a rischio di bancarotta, secondo il modello Random Forest.
Ogni barra rappresenta il contributo medio della feature alle decisioni del modello: più la barra è lunga, maggiore è stato il suo peso nella classificazione.
Le variabili più rilevanti includono:
- **Borrowing dependency** → misura quanto l'azienda dipende dai finanziamenti esterni; un valore elevato può indicare vulnerabilità finanziaria.
- **Net Income to Total Assets** → indica la redditività rispetto agli asset; valori bassi segnalano scarsa efficienza operativa.
- **Continuous interest rate (after tax)** e **After-tax net Interest Rate** → legate ai costi del debito; valori alti suggeriscono un peso significativo degli oneri finanziari.
- **Liability to Equity** e **Debt ratio %** → misurano la leva finanziaria; più è alta, più l'azienda è esposta al rischio.
- **Persistent EPS in the Last Four Seasons** → utile per valutare la stabilità degli utili nel tempo.

Questa analisi è utile per identificare quali aspetti aziendali pesano maggiormente nella valutazione del rischio finanziario da parte del modello.

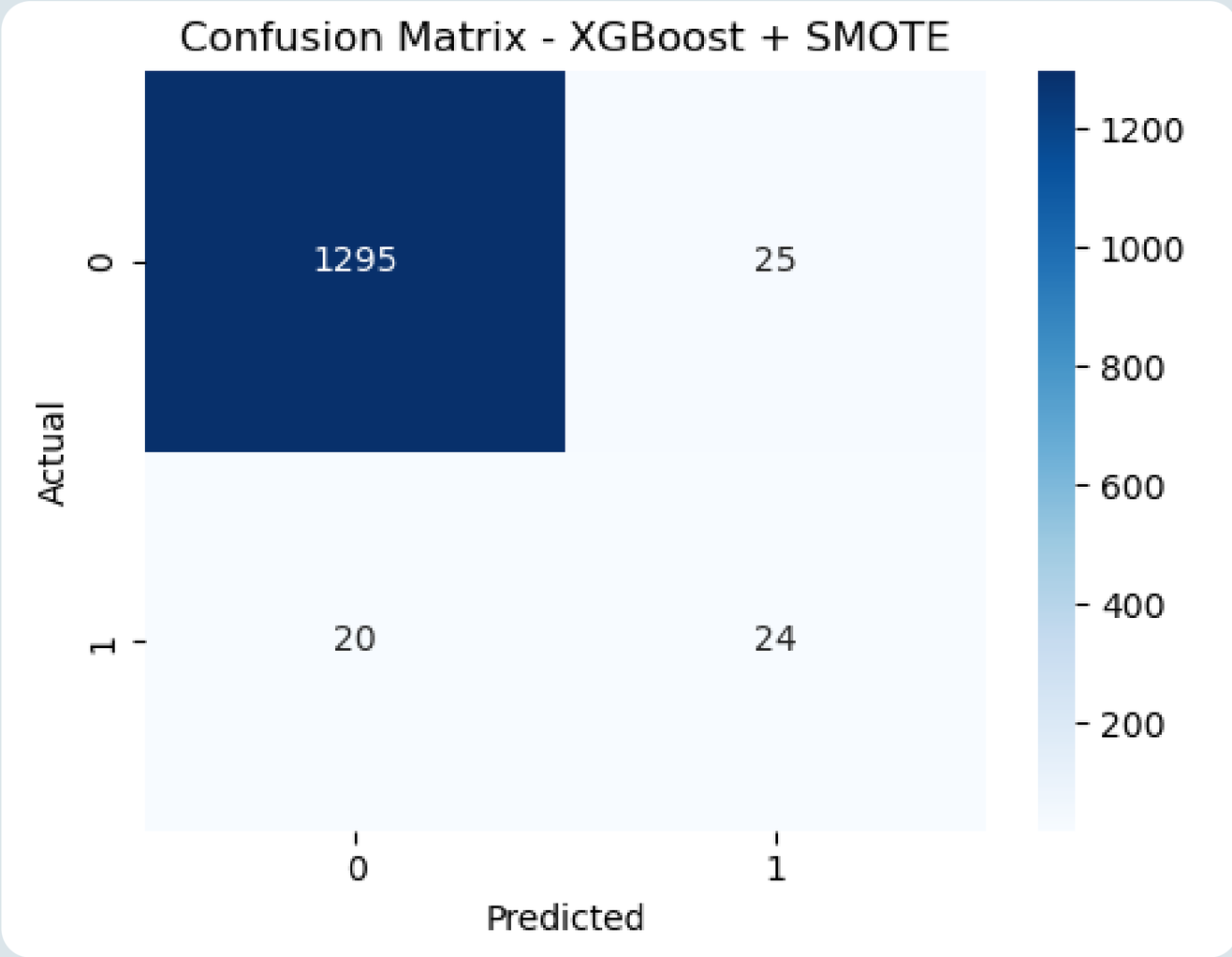


XGBoost

XGBoost → terzo modello testato, apprezzato per la sua capacità di gestire feature non lineari in modo molto efficace. È stata effettuata una fase di **ottimizzazione** con:

- **SMOTE** → gestisce lo sbilanciamento generando esempi sintetici della Classe minoritaria (1)
- **StandardScaler** → standardizza tutte le feature (media = 0, σ = 1)
- **GridSearchCV** → ricerca della combinazione ottimale di iperparametri del modello

Metrica	Classe 0	Classe 1
Precision	0.98	0.49
Recall	0.98	0.55
F1-Score	0.98	0.52
Support (n. esempi)	1320	44



Honorable mention

Modelli testati ma esclusi dall’ensemble finale

Support Vector Machine (SVM)

- **Kernel lineare** e **Kernel RBF**
- Tempi di training → **elevati**
- Performance su Classe 1 → **deboli**

Multi-Layer Perceptron (MLP)

- **Buone prestazioni** con SMOTE
- **Poco** interpretabile
- Performance **instabili**

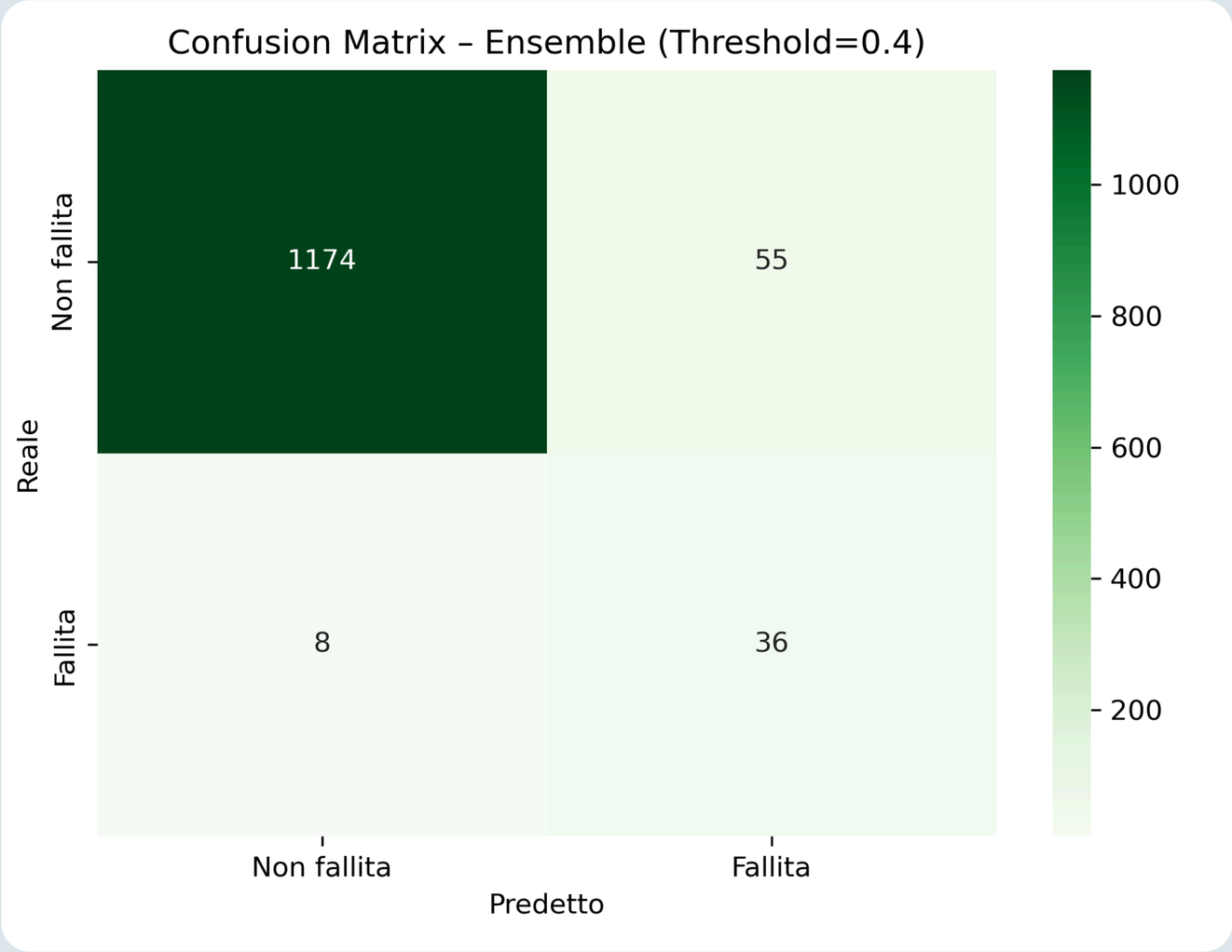
Ensemble Model

Ensemble Voting Classifier → modello finale sviluppato. Combina i risultati di **Logistic Regression**, **Random Forest** e **XGBoost**.

È stata effettuata una fase di **ottimizzazione** attraverso:

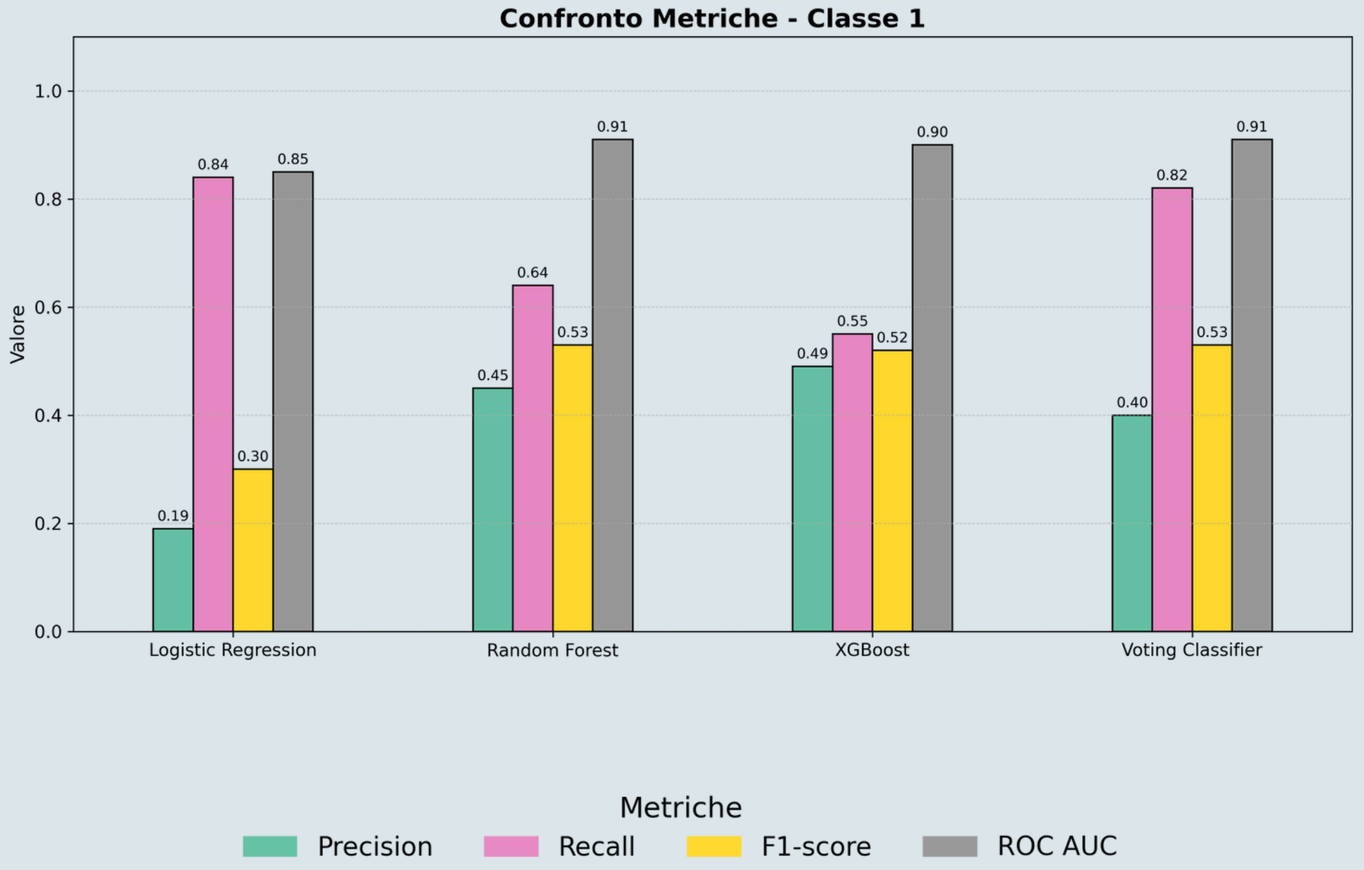
- **Rimozione outlier** solo dalla Classe dominante (0)
- **SMOTEENN** → bilanciamento sintetico combinato con pulizia dei dati ambigui
- **StandardScaler** → standardizza tutte le feature (media = 0, σ = 1)
- **GridSearchCV** → ricerca dei migliori iperparametri per ciascun modello base (LR, RF, XGB)
- **Class_Weight →** Random Forest ha un peso maggiore

Metrica	Classe 0	Classe 1
Precision	0.99	0.40
Recall	0.96	0.82
F1-Score	0.97	0.53
Support (n. esempi)	1229	44



Confronto metriche Classe 1

- **Logistic Regression** → baseline interpretabile, ma meno performante
- **Random Forest** e **XGBoost** → buon equilibrio tra Precision, Recall e F1-Score per la Classe minoritaria
- **Voting Classifier** → sintesi robusta



Raccomandazione operativa

- **Modelli consigliati** → **Random Forest** e **XGBoost**, performance migliori per rilevare casi di bancarotta
- **Voting Classifier** → ideale per massimizzare Recall e stabilità
- Focus su **riduzione FN** → preferire modelli più sensibili alla Classe 1

Modello	Pro	Contro
Logistic Regression	Interpretabile, veloce, baseline chiara	Performance inferiori su classe minoritaria
Random Forest	Ottima capacità predittiva, gestisce bene outlier e feature complesse	Meno interpretabile, può causare overfitting se non ottimizzato
XGBoost	Alta accuratezza, gestisce bene non linearità e feature complesse	Più complesso da configurare, maggiore tempo di addestramento
Voting Classifier	Bilancia punti di forza dei singoli modelli, stabile	Richiede tuning accurato dei pesi e modelli base

Conclusioni

Risultati principali:

- Ensemble Voting Classifier (LR, RF, XGB) → **ROC AUC > 0.90**
- SMOTEENN e rimozione outlier (Classe 0) → **miglior bilanciamento**
- **Random Forest** e **XGBoost** → modelli più robusti ed efficaci

Criticità:

- Classe 1 molto piccola → rischio **overfitting** e **falsi negativi**
- Modelli complessi (MLP, SVM) con **performance instabili**

Sviluppi futuri:

- Tecniche **cost-sensitive learning** (penalizzare FN)
- **Dati sequenziali** o **temporali** se disponibili
- Validazione su **nuovi dati reali fuori dal campione**

Grazie per l'attenzione