

# **Identificatore di transazioni fraudolente**

Progetto di Machine Learning

Sabato Iaquino  
0512123029  
a.a. 2025-2026

# Indice

1. Introduzione al problema	3
1.1.Definizione del problema	3
1.2.Sviluppo del progetto	3
2. Dataset	4
2.1.Descrizione del dataset	4
2.2.Analisi del dataset	4
2.3.Pre-processing e feature engineering	7
3. Scelte progettuali	9
3.1.Modelli addestrati	9
4. Analisi delle prestazioni	10
4.1.Metriche di valutazione	10
4.2.Dataset di train e test	10
4.3.Performance dei modelli	10
5. Considerazioni finali	12
5.1.Considerazioni sulle performance	12
5.2.Sviluppi futuri	12

# **1. Introduzione al problema**

## **1.1.Definizione del problema**

La diffusione dei pagamenti digitali ha reso le transazioni online sempre più comode ma anche più esposte ai truffatori finanziari: le frodi finanziarie svolte tramite transazioni con carta di credito rappresentano un problema significativo per le istituzioni finanziarie e i loro clienti, causando perdite, danni e sfiducia.

L'obiettivo del progetto è la realizzazione di un sistema di individuazione automatica di transazioni potenzialmente fraudolente, classificando ogni transazione come legittima o fraudolenta tramite un modello di Machine Learning, riducendo le perdite economiche e i falsi allarmi, allo stesso momento aumentando il livello di sicurezza dei sistemi bancari digitali.

## **1.2.Sviluppo del progetto**

Il progetto si articolerà in quattro fasi:

1. Analisi esplorativa dei dati a disposizione, pre-processing e feature engineering, nel capitolo 2;
2. Scelte progettuali dello sviluppo, dei criteri e della costruzione del modello, nel capitolo 3;
3. Analisi delle prestazioni ottenute dal modello addestrato, nel capitolo 4;
4. Considerazioni finali sui risultati dello studio e possibili futuri sviluppi, nel capitolo 5.

## 2. Dataset

### 2.1. Descrizione del dataset

Il progetto è stato sviluppato con l'uso del dataset **Credit Card Fraud Detection** di Machine Learning Group of Université Libre de Bruxelles, disponibile su Kaggle (<https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>).

È una raccolta di transazioni con carta di credito europee svolta a settembre del 2013 durante una ricerca congiunta tra Wordline e l'Université Libre de Bruxelles sull'ottenimento dei big data e il rilevamento di frodi.

### 2.2. Analisi del dataset

Il dataset è formato in totale da 284,807 campioni e da 31 feature numeriche:

- Time: tempo passato tra la prima transazione del dataset e ogni altra transazione, in numero di secondi;
- V1-V28: 28 diverse feature che rappresentano tutte le informazioni personali censurate tramite trasformazione PCA<sup>1</sup>;
- Amount: quantità di denaro trasferito con la transazione;
- Class: classificazione dell'attività, le attività con Class=0 sono legittime mentre quelle con Class=1 sono fraudolente.

---

<sup>1</sup> Trasformazione Principal Component Analysis (PCA): tecnica di riduzione della dimensionalità che trasforma dati complessi e correlati in un set più piccolo di variabili non correlate, chiamate componenti principali.

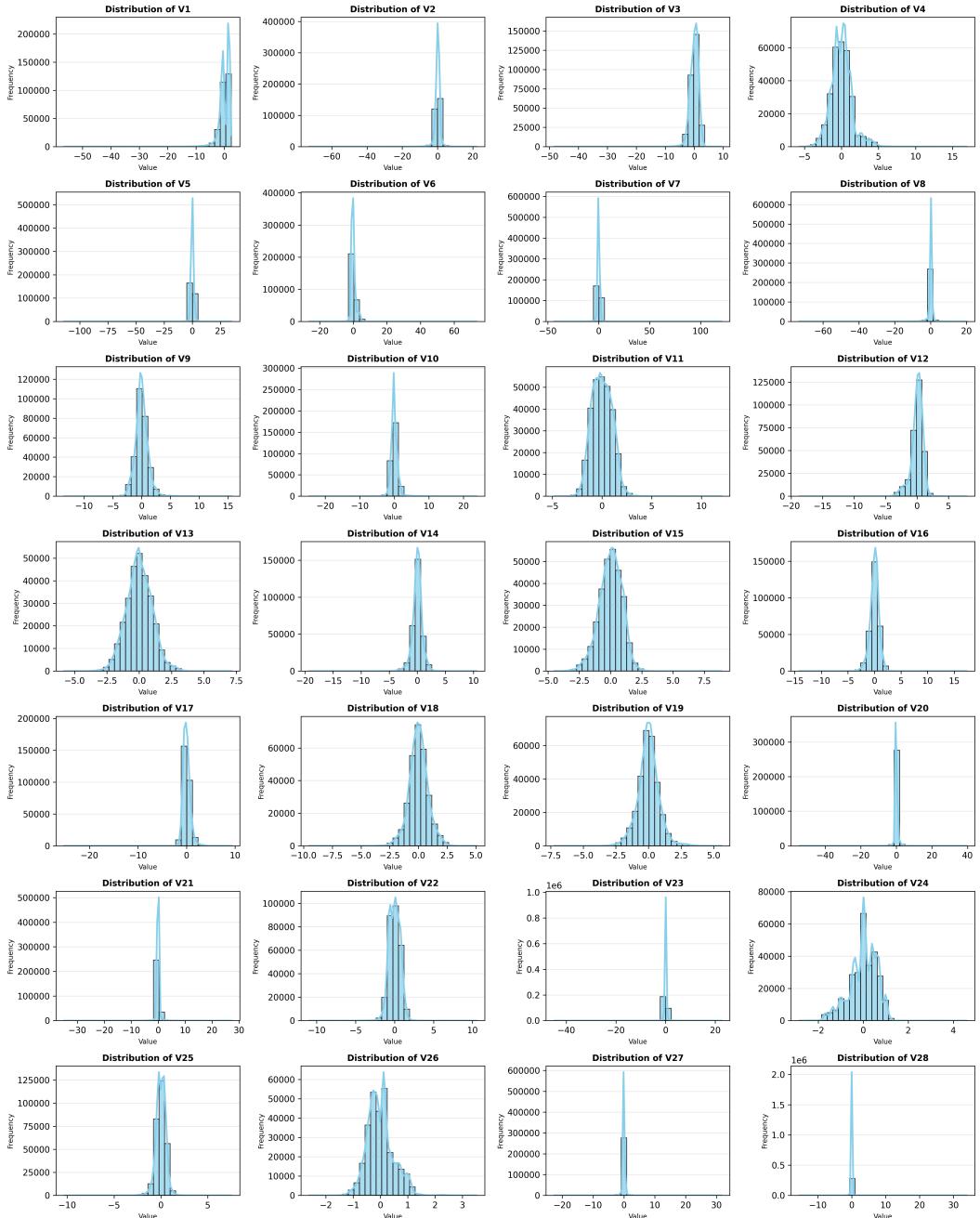


Fig. 1 - Distribuzione feature V1-V28

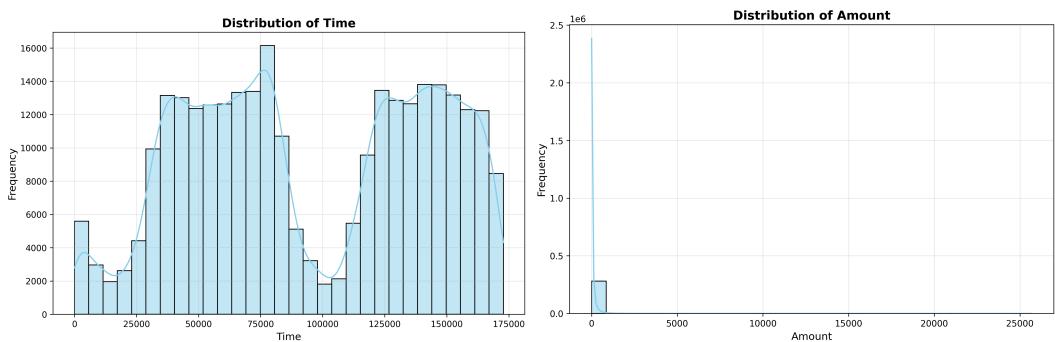


Fig. 2 - Distribuzione feature "Time"

Fig. 3 - Distribuzione feature "Amount"

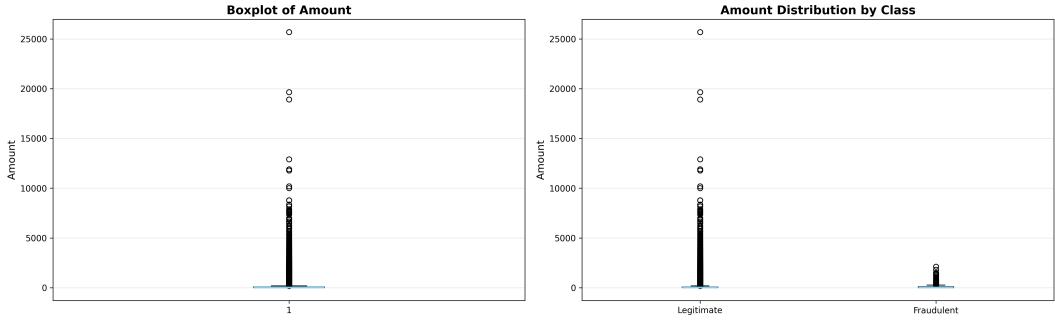


Fig. 4 - Boxplot feature “Amount”

Fig. 5 - Boxplot feature “Amount” per classe

Feature	Type	Missing	Statistics
Time	Numeric	0 (0.0%)	Mean: 94.813.86 Std: 47.488.15 Min: 0.00   Max: 172,792.00
V1	Numeric	0 (0.0%)	Mean: 0.00 Std: 1.96 Min: -56.41   Max: 2.45
V2	Numeric	0 (0.0%)	Mean: 0.00 Std: 1.65 Min: -72.72   Max: 22.06
V3	Numeric	0 (0.0%)	Mean: -0.00 Std: 1.00 Min: -48.33   Max: 9.38
V4	Numeric	0 (0.0%)	Mean: 0.00 Std: 1.42 Min: -5.68   Max: 16.88
V5	Numeric	0 (0.0%)	Mean: 0.00 Std: 1.38 Min: -113.74   Max: 34.80
V6	Numeric	0 (0.0%)	Mean: 0.00 Std: 1.33 Min: -26.16   Max: 73.30
V7	Numeric	0 (0.0%)	Mean: 0.00 Std: 1.24 Min: -43.56   Max: 120.59
V8	Numeric	0 (0.0%)	Mean: 0.00 Std: 1.19 Min: -73.22   Max: 20.01
V9	Numeric	0 (0.0%)	Mean: -0.00 Std: 1.10 Min: -13.43   Max: 15.59
V10	Numeric	0 (0.0%)	Mean: 0.00 Std: 1.09 Min: -10.59   Max: 23.75
V11	Numeric	0 (0.0%)	Mean: 0.00 Std: 1.02 Min: -4.80   Max: 12.02
V12	Numeric	0 (0.0%)	Mean: -0.00 Std: 1.00 Min: -18.68   Max: 7.85
V13	Numeric	0 (0.0%)	Mean: 0.00 Std: 1.00 Min: -5.79   Max: 7.13
V14	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.99 Min: -19.21   Max: 10.53
V15	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.92 Min: -4.50   Max: 8.88
V16	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.88 Min: -14.13   Max: 17.32
V17	Numeric	0 (0.0%)	Mean: -0.00 Std: 0.85 Min: -25.16   Max: 9.25
V18	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.84 Min: -9.50   Max: 5.04
V19	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.81 Min: -7.21   Max: 5.59
V20	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.77 Min: -54.50   Max: 39.42
V21	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.73 Min: -31.83   Max: 27.20
V22	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.73 Min: -10.93   Max: 10.50
V23	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.62 Min: -44.81   Max: 22.53
V24	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.61 Min: -2.84   Max: 4.58
V25	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.60 Min: -10.30   Max: 7.52
V26	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.48 Min: -2.60   Max: 3.52
V27	Numeric	0 (0.0%)	Mean: -0.00 Std: 0.40 Min: -22.57   Max: 31.61
V28	Numeric	0 (0.0%)	Mean: -0.00 Std: 0.33 Min: -18.43   Max: 33.85
Amount	Numeric	0 (0.0%)	Mean: 0.00 Std: 250.12 Min: 0.00   Max: 25,691.16
Class	Numeric	0 (0.0%)	Mean: 0.00 Std: 0.04 Min: 0.00   Max: 1.00

Total Samples: 284,807

Fig. 6 - Statistiche feature

Il dataset è privo di missing values e le variabili categoriche sono già state trasformate in variabili numeriche (ovvero V1-V28), tuttavia il dataset è estremamente sbilanciato, su 284,807 campioni solo 492 sono frodi:

Class	Samples	Percentage
Legitimate	284,315	99.83%
Fraudulent	492	0.17%

**Total Samples: 284,807**

Fig. 7 - Bilanciamento della feature “Class”

## 2.3.Pre-processing e feature engineering

Sono state svolte diverse operazioni di pre-processing e feature engineering sul dataset, affinché si abbia una struttura e una qualità dell’insieme dei campioni migliore:

1. Rimozione dei duplicati: il dataset originale comprende alcuni duplicati che vengono individuati e rimossi, tramite le funzioni pandas.duplicated() e pandas.drop\_duplicates();
2. Gestione degli outliers: nella feature Amount c’è un numero molto alto di outliers (vedi fig. 4 e fig. 5), per gestirli è stato utilizzato il metodo IQR<sup>2</sup>;

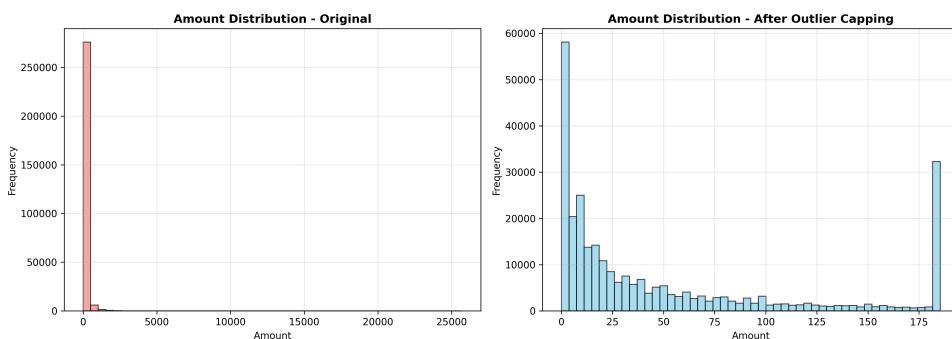


Fig. 8 - Distribuzione feature “Amount” prima e dopo la gestione degli outliers

3. Log transformation: la feature Amount ha una forte asimmetria nella sua distribuzione, pertanto viene svolta una trasformazione logaritmica sulla feature tramite numpy.log1p(), per evitare di causare il calcolo di log(0);

<sup>2</sup>Interquartile Range (IQR): differenza tra il terzo e il primo quartile, ovvero l’ampiezza della fascia di valori che contiene la metà centrale della distribuzione dei valori osservati, riportando gli outliers entro i limiti del secondo e terzo quartile.

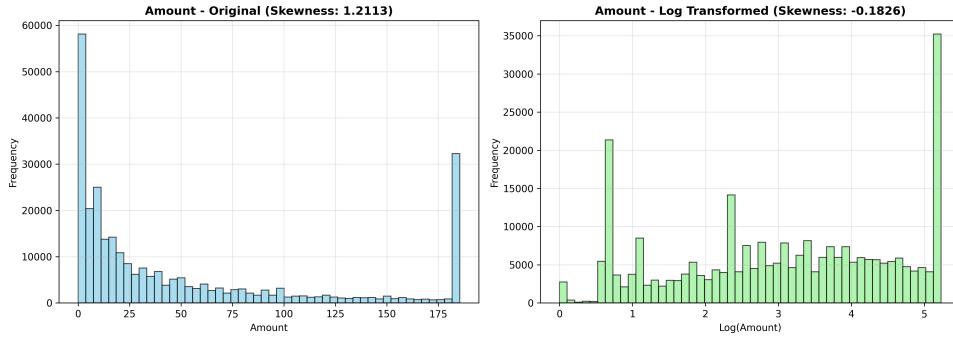


Fig. 9 - Distribuzione feature “Amount” prima e dopo la log transformation

4. Splitting: viene svolto lo split del dataset in due dataset, uno di train (`X_train.csv`, `y_train.csv`) e uno di test (`X_test.csv`, `y_test.csv`), utilizzando la regola di Pareto e verificando che tutti i dataset abbiano campioni di entrambe le tipologie utilizzando il campionamento stratificato;
5. Bilanciamento delle classi: le classi vengono bilanciate tramite SMOTE<sup>3</sup>, le percentuali testate sono 5%, 10% e 20%, ma i risultati migliori sono stati ottenuti con una percentuale del 10%, mentre con 5% e 20% si verificano casi, rispettivamente, di undersampling e oversampling, causando un aumento del numero di falsi negativi e falsi positivi che compromette l’efficacia del modello addestrato;

È stato deciso di non fare operazioni di scaling o normalizzazione sulle feature V1-V28, in quanto, essendo l’output di trasformazioni PCA, sono già normalizzate e non serve andare ad operare ulteriormente.

---

<sup>3</sup> Synthetic Minority Over-sampling Technique (SMOTE): tecnica di pre-elaborazione dati usata per risolvere lo sbilanciamento delle classi, creando campioni sintetici per la classe minoritaria tramite interpolazione tra i punti dati esistenti.

### 3. Scelte progettuali

#### 3.1. Modelli addestrati

Per lo sviluppo è stato deciso di effettuare due addestramenti:

- Un classificatore probabilistico Gaussian Naïve Bayes, tramite la funzione di libreria `sklearn.naive_bayes.GaussianNB`, utilizzato come baseline per il confronto e la validazione delle performance. Sono stati utilizzati gli iperparametri di default della libreria;
- Un ensemble Random Forest, tramite la funzione di libreria `sklearn.ensemble.RandomForestClassifier`, è il modello principale da utilizzare per effettuare le predizioni sulle transazioni fraudolente. Per selezionare il migliore sottoinsieme di iperparametri è stato svolto hyperparameter tuning con K-Fold Cross Validation, con K=5, utilizzando `sklearn.model_selection.GridSearchCV()` per valutare tutte le possibili combinazioni, usando come metrica di valutazione F1-Score. Le combinazioni di iperparametri sono state effettuate con i seguenti valori:

- ▶ `n_estimators` numero complessivo di alberi: [50, 100, 150]
- ▶ `max_depth` profondità massima degli alberi: [10, 15, 20]
- ▶ `min_sample_split` numero minimo di campioni per svolgere uno split: [5, 10, 15]
- ▶ `min_sample_leaf` numero minimo di campioni che ogni foglia deve avere: [2, 5, 10]

I migliori iperparametri trovati dalla K-Fold Cross Validation sono:

- ▶ `n_estimators=100`
- ▶ `max_depth=20`
- ▶ `min_sample_split=10`
- ▶ `min_sample_leaf=2`

È stato inoltre deciso di applicare degli iperparametri fissi:

- ▶ `class_weight=[‘balanced’]` per il peso delle classi
- ▶ `random_state=42` per la riproducibilità;
- ▶ `n_jobs=-1` per parallelizzare la computazione e usare tutti i core della CPU a sua disposizione;

# 4. Analisi delle prestazioni

## 4.1. Metriche di valutazione

Le metriche di valutazione utilizzate per i modelli addestrati sono:

- Accuracy, percentuale della precisione generale del modello;
- Precision, percentuale delle predizioni positive corrette rispetto al totale delle predizioni positive;
- Recall: percentuale delle predizioni positive corrette rispetto al totale dei campioni positivi;
- F1-Score: media armonica tra le medie di precisione e recall;
- ROC-AUC<sup>4</sup>: area sotto la curva che traccia il tasso di veri positivi contro il tasso di falsi positivi.

## 4.2. Dataset di train e test

Dalle operazioni di pre-processing e feature engineering sono risultati come dataset di train e di test:

- X\_train.csv e y\_train.csv: in totale 249,262 campioni, di cui 226,602 classificati come legittimi e 22,660 classificati come frodi;
- X\_test.csv e y\_test.csv: in totale 62,316 campioni, di cui 56,651 classificati come legittimi e 5,665 classificati come frodi;

## 4.3. Performance dei modelli

Entrambi i modelli sono stati addestrati sui dataset di train e provati sui dataset di test descritti precedentemente, ottenendo performance pari a:

### Gaussian Naïve Bayes

- Accuracy: 96.72%
- Precision: 90.48%
- Recall: 71.46%
- F1-Score: 79.85
- ROC-AUC: 96.96%

### Random Forest [Best parameters]

- Accuracy: 98.92%
- Precision: 99.84%
- Recall: 88.24%
- F1-Score: 93.68%
- ROC-AUC: 98.30%

---

<sup>4</sup> Area Under the Receiver Operating Characteristic Curve

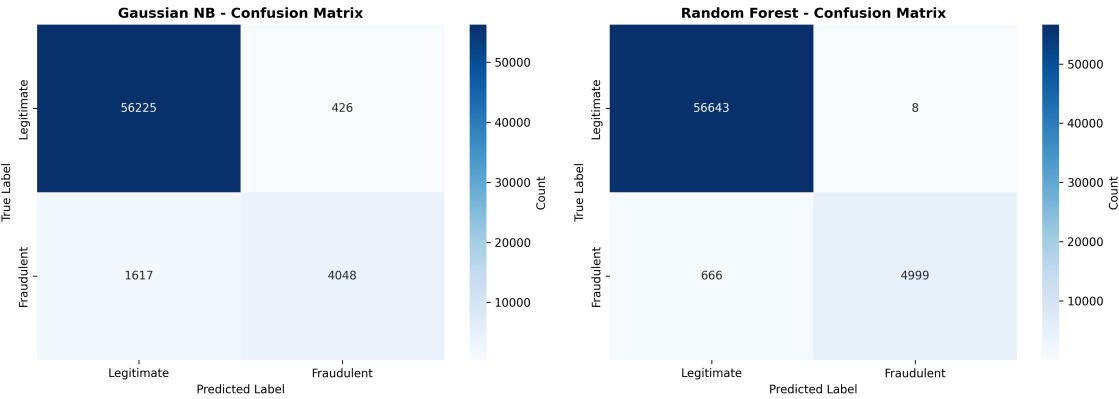


Fig. 10 - Confusion matrices dei modelli addestrati

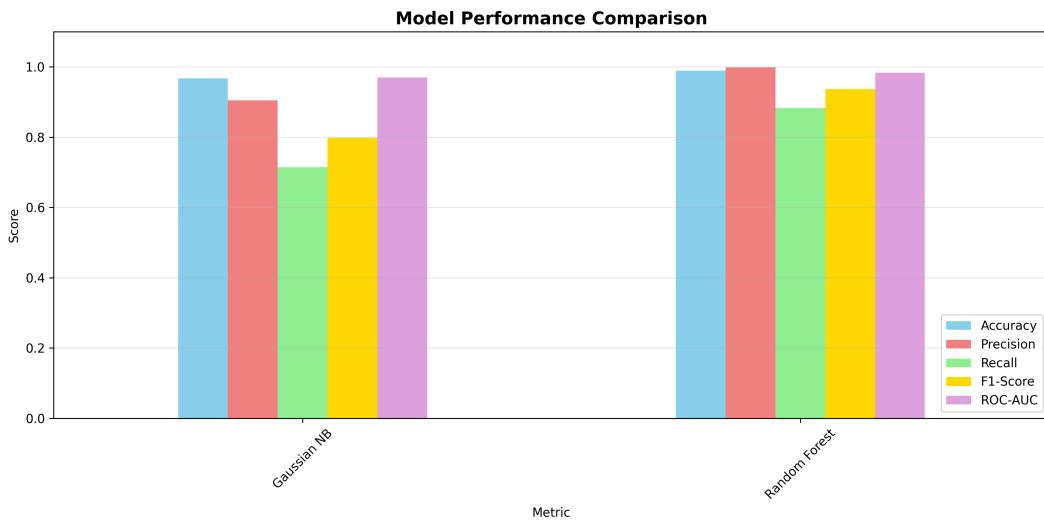


Fig. 11 - Istogramma di comparazione delle performance dei modelli addestrati

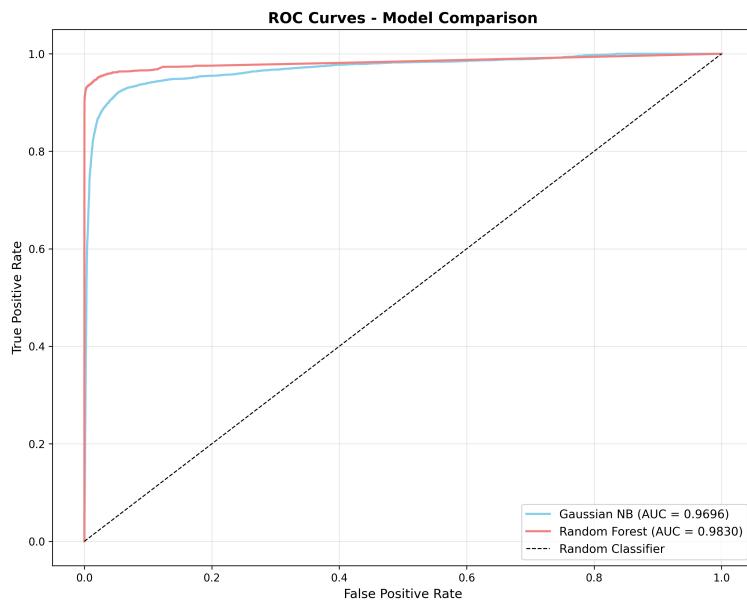


Fig. 12 - Curve ROC dei modelli addestrati

# **5. Considerazioni finali**

## **5.1. Considerazioni sulle performance**

Valutando le prestazioni del modello Gaussian Naïve Bayes e del modello Random Forest è possibile notare come un’ensemble possa aiutare drasticamente il sistema ad effettuare predizioni più accurate, soprattutto in situazioni più delicate (dataset con dati censurati per privacy, situazioni che richiedono minimo tasso di falsi positivi/falsi negativi).

Tuttavia, la Random Forest è solo uno degli ensemble che permette di migliorare le performance di classificazione in scenari simili che non offre performance ottimali sotto alcuni aspetti, soprattutto per quanto riguarda il tasso di falsi positivi. Si potrebbero usare ensemble con tecniche di boosting per migliorare le performance, in primis la recall, come XGBoost<sup>5</sup>, inoltre svolgere addestramenti con sistemi di Deep Learning o librerie con supporto nativo per l’accelerazione GPU, in quanto Scikit-Learn non offre nativamente soluzioni che sfruttino la GPU del sistema ma basa l’intero addestramento e uso del modello sulla CPU.

## **5.2. Sviluppi futuri**

Le tipologie di frodi bancarie sono molteplici, ma ulteriori verifiche le si possono fare anche sulla feature “time” che, in questo studio, non è stata utilizzata.

Si possono sviluppare soluzioni algoritmiche ad-hoc che verificano il tempo entro il quale avvengono specifici bonifici dello stesso importo allo stesso destinatario da diversi mittenti, per valutare l’eventuale fraudolenza del conto bancario di destinazione, oppure verificare la cadenza di trasferimenti di piccolo importo per verificare la presenza in atto delle truffe a microtransazioni periodiche.

---

<sup>5</sup> eXtreme Gradient Boosting: tecnica di ensemble learning che combina modelli considerati deboli per creare un unico modello accurato considerato forte, usando tecniche di boosting dove il modello successivo impara dagli errori del modello precedente.