UNIVERSITÀ DEGLI STUDI FIRENZE
Laurea Magistrale in Data Science, Calcolo Scientifico e Intelligenza Artificiale

NAÏVE BAYES
RANDOM FOREST

Un approccio Bayesiano alla classificazione di dataset sbilanciati

Bernardo Busoni, Firenze 26 Giugno 2025

OBBIETTIVO

Nell'ambito dei modelli di classificazione, il modello predittivo di Machine Learning ***Random Forest*** soffre la presenza di classi particolarmente sbilanciate all'interno dei dataset in analisi. In particolare, in presenza di due classi, si riscontra la tendenza del modello a sovrastimare la classe maggioritaria, ottenendo dei valori bassi del parametro di performance ***Recall***. Nei casi di utilizzo di una matrice dei costi, ad esempio per applicazioni mediche, questo può portare ad un costo assolutamente indesiderato a causa dei troppi falsi negativi. In questo progetto si propone un modello che, partendo dal modello *Random Forest*, utilizzi e vi integri un **approccio bayesiano** per risolvere parzialmente il problema.

FREQUENTIST VS BAYESIAN

L’approccio frequentista e quello bayesiano alla statistica differiscono nel modo in cui interpretano la **probabilità** e, di conseguenza, nel modo in cui costruiscono e aggiornano i modelli.

Nell’approccio frequentista, la probabilità è la **frequenza** relativa di un evento in un gran numero di prove, i parametri del modello sono considerati fissi e ignoti e l’incertezza viene gestita solo attraverso la variabilità dei dati osservati.

Nell’approccio bayesiano, la probabilità è il **grado di fiducia** in una ipotesi e i parametri sono variabili aleatorie: si parte da una distribuzione a priori che rappresenta le nostre credenze iniziali, e la si aggiorna con i dati osservati per ottenere una distribuzione a posteriori.

IL PROBLEMA DEL DOTTORE

Supponiamo che a un medico si presenti un paziente il quale presenta dei sintomi che fanno pensare abbia una determinata malattia. Supponiamo che il dottore **assegni una probabilità** di 0.6 che il paziente abbia la malattia.

Successivamente il dottore sottopone il paziente a 5 tamponi dei quali conosciamo la seguente **matrice di confusione**: (frequenze)

Stato \ Previsione	0	1
0	**0.9**	**0.3**
1	**0.1**	**0.7**

*abbiamo anche assunto che i risultati dei tamponi fossero indipendenti condizionatamente allo stato del paziente! (Naïve Bayes assumption)

Dai seguenti risultati dei tamponi:

1 - 0 - 0 - 1 - 0

Cosa può concludere il medico?

Grazie al **teorema di Bayes** e la legge delle probabilità totali*, possiamo sfruttare le probabilità a priori e i risultati osservati per concludere che il paziente è malato con probabilità: **0.73**

NAÏVE BAYES CLASSIFIER

Il modello predittivo Naïve Bayes Classifier è un modello che si fonda sull'approccio Bayesiano. Funziona nel seguente modo:

- Si considera ogni attributo e la classe target come delle variabili aleatorie. I campioni osservati si considerano realizzazioni delle v.a.;
- Si assume che le v.a. degli attributi siano indipendenti condizionatamente alla v.a. target → **Naïve Bayes Assumption**;
- Si stima la probabilità a priori sulla base della distribuzione delle classi nel train set;
- Per ogni unità, si calcola la probabilità che essa appartenga a una determinata classe date le osservate sugli attributi:

$$P(Y \mid X1, \dots, Xn) = \frac{P(X1, \dots, Xn \mid Y) \cdot P(Y)}{P(X1, \dots, Xn)} = \frac{P(X1 \mid Y) \cdot \dots \cdot P(Xn \mid Y) \cdot P(Y)}{P(X1, \dots, Xn)}$$

- Si classifica l'unità nella classe che massimizza questo valore.

NAÏVE BAYES RANDOM FOREST

Il modello proposto è l'**unione** di un modello Naïve Bayes con una classica Random Forest Classifier. Quando il modello viene interrogato su una unità, ogni albero decisionale della foresta fornisce una predizione.

Nel modello Random Forest Standard si classifica l'unità in una classe mediante il **voto di maggioranza** ottenuto.

Nel modello proposto, ogni predizione viene **pesata** condizionatamente all'appartenenza dell'unità ad una determinata classe.

Tenendo conto della distribuzione delle classi nel dataset, l'unità viene classificata nella classe che massimizza le **probabilità a posteriori**.

I DETTAGLI

01 Dividiamo il dataset in tre insiemi disgiunti: **TRAIN SET** - **TEST SET** - **BAYES TEST SET** nelle proporzioni **60** - **20** - **20**. Sul Train Set, calcoliamo la frazione di unità presenti per ogni classe. Queste costituiranno le nostre **probabilità a priori**.

02 Mediante le unità presenti nel Train set, **addestriamo** gli alberi decisionali, ognuno con una random seed differente. Le unità del Test set, invece, le utilizziamo per **stimare** la **frequenza** media con cui ogni albero restituisce **TP, FP, TN, FN**.

03 Infine, le unità del Bayes Test set le utilizziamo per stimare le **performance** del modello proposto e paragonarle a quelle del modello Random Forest Standard. Calcoliamo Accuracy, Precision, Recall e F1 Score oltre alla Matrice di Confusione.

04 In particolare, per ogni unità da predirre, utilizzando la prior e le stime delle frequenze, calcoliamo la **probabilità a posteriori** che l'unità appartenga a una determinata classe: se è maggiore di 0.5, la classifichiamo in quella classe, altrimenti nell'altra.

CONFRONTO

Si è scelto di confrontare il modello proposto (**NBRF**) con la Random Forest Standard (**RFS**) su una serie di dataset **sbilanciati**, sia perché le nostre prior potessero avere un impatto più consistente, sia perché è noto in letteratura che le RFS presentino alcune difficoltà nella classificazione su dataset di questo tipo.

Si è deciso innanzitutto di testare i modelli su **3** dataset di dati sintetici provenienti da 3 ***Data Generating Process*** differenti. Uno di questi è stato riprodotto in 3 proporzioni differenti, complessivamente:

80 – 20; 70 – 30; 50 – 50

Successivamente si sono confrontati i modelli su **7 dataset reali**, ognuno dei quali conteneva una **variabile target bivariata** con classi più o meno sbilanciate. I dataset provengono dalle varie applicazioni, come:

- Telecomunicazioni;
- Crediti Bancari;
- Posta Elettronica;
- Medicina.

DATA GENERATING PROCESS

Dimensione del campione per dataset: 10.000
Sbilanciamento: 70 - 30; 70 - 30; 70 - 30; 80 - 20; 50 - 50

$$X_1 = N(2,1)$$
$$X_2 = N(2,1)$$
$$X_3 = Exp(1)$$
$$X_4 = Exp(1)$$
$$X_5 = Unif[0,4]$$
$$X_6 = Unif[0,X_5]$$
$$X_7 = Ber(0.5)$$

$$Y_0 = X_1 + X_2 \cdot X_7 - 3X_6^2$$
$$Y_1 = X_4 + X_2 \cdot X_3 \cdot X_6 - X_5^3$$

$$X_1 = N(2,1)$$
$$X_2 = N(1,|X_1|)$$
$$X_3 = Exp(1)$$
$$X_4 = Poi(4)$$
$$X_5 = Unif[0,X_4]$$
$$X_6 = Unif[X_5,X_4]$$
$$X_7 = Unif[0,1]$$
$$X_8 = Ber(X_7)$$
$$X_9 = X_8 \cdot N(-5,3) + (1-X_8) \cdot Bin(X_4,X_7)$$
$$X_{10} = N(X_2,X_3)$$
$$X_{11} = Poi(6)$$

$$Y_0 = X_{10}^{|X_3|} + X_6 \cdot X_1 - X_3 \cdot X_4$$
$$Y_1 = X_4^{|X_2|} + X_9 + X_1 \cdot X_2$$

$$X_1 = N(0,1),$$
$$X_2 = LogNormal(0,0.5^2)$$
$$X_3 = Beta(2,5)$$
$$X_4 = Pois(3)$$
$$X_5 = Unif[-3,3]$$
$$X_6 = Ber(0.4)$$
$$X_7 = Bin(X_4, min(0.95, max(0.05, X_3))$$
$$X_8 = N(X_2 X_3, 0.5^2)$$
$$X_9 = Unif[0, X_2 + 1]$$
$$X_{10} = Exp(0.2)$$
$$X_{11} = \chi^2(3)$$
$$X_{12} = X_6 \cdot Lap(0,1)$$
$$X_{13} = Unif[0,1]$$
$$X_{14} = Ber(0.5)$$
$$X_{15} = Poi(1)$$
$$X_{16} = N(0,1)$$

$$Y_0 = \log(1+X_2) + \sqrt{X_{11}} + X_6 X_5^2 - 0.1 \cdot X_{10} + \sin(X_1)$$
$$Y_1 = \sqrt{X_4 + 1} + \cos(X_5) + \frac{X_7}{1+X_2} + X_{12} + \tfrac{1}{2} X_8^2$$

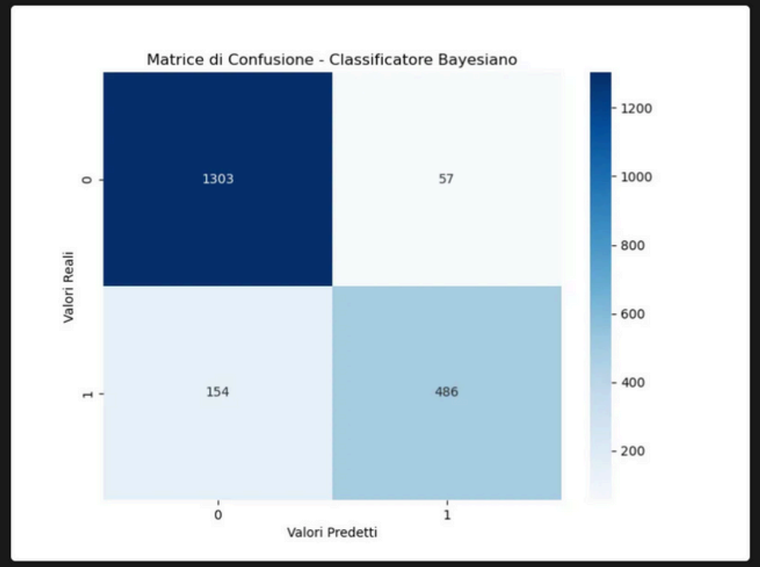
DGP - RISULTATI

1°

70%

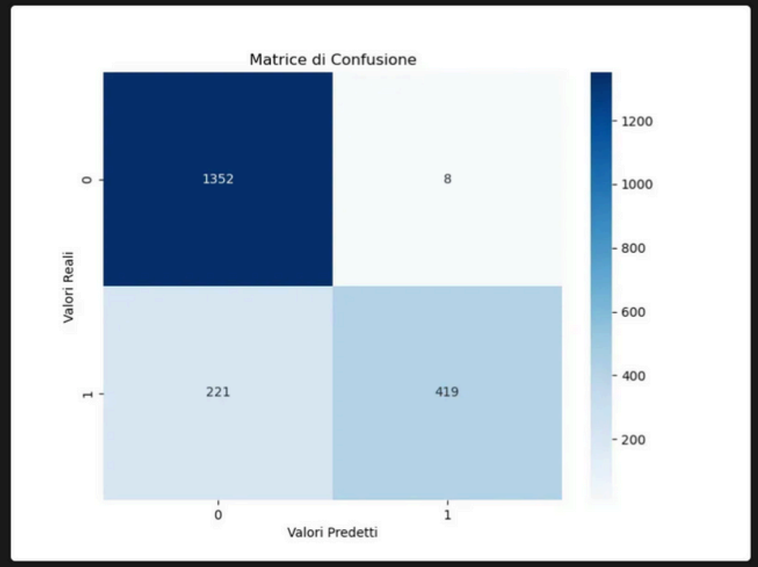
Risultati Naïve Bayes Random Forest:

Accuracy: 0.8945
Precision: 0.8950
Recall: 0.7594
F1 Score: 0.8216



Risultati Standard Random Forest:

Accuracy: 0.8855
Precision: 0.9813
Recall: 0.6547
F1 Score: 0.7854



Naïve Bayes Random Forest | Slide nove - Performance modelli sui DGP

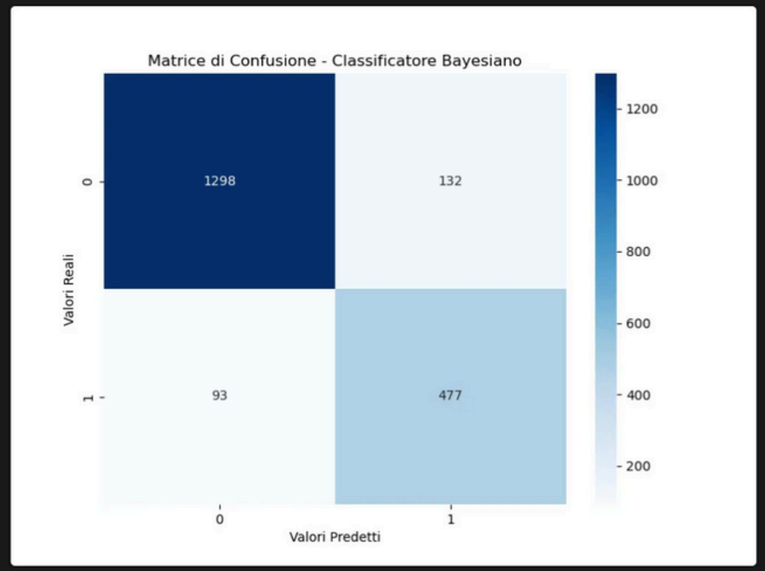
DGP - RISULTATI

2°

70%

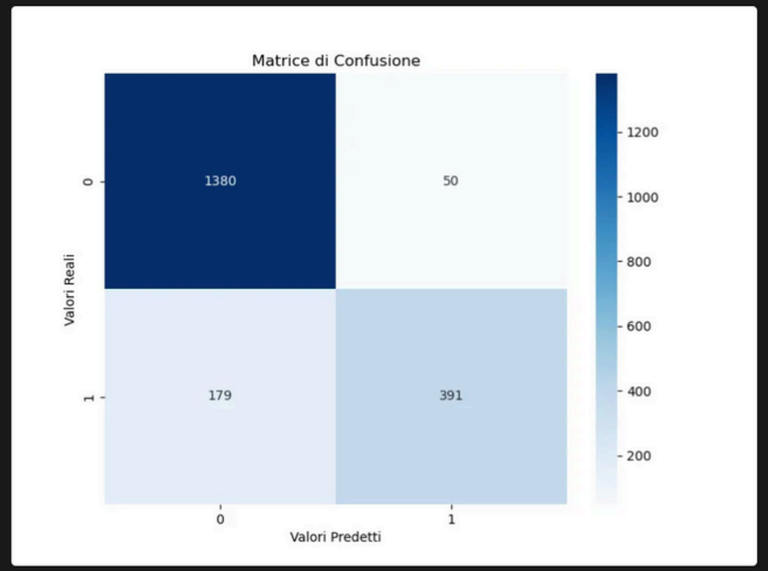
Risultati Naïve Bayes Random Forest:

Accuracy: 0.8875
Precision: 0.7833
Recall: 0.8368
F1 Score: 0.809



Risultati Standard Random Forest:

Accuracy: 0.8855
Precision: 0.8866
Recall: 0.6860
F1 Score: 0.7735



DGP - RISULTATI

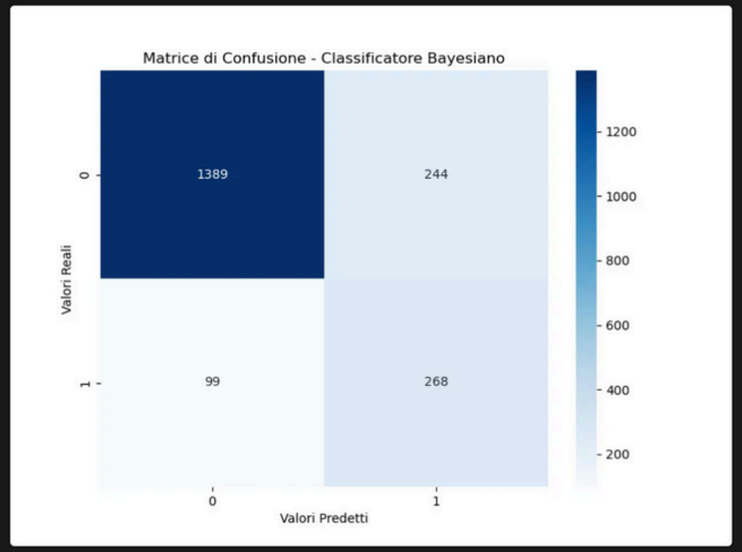
3.1°

80%

Risultati Naïve Bayes Random Forest:

P _Y0 = 0.80

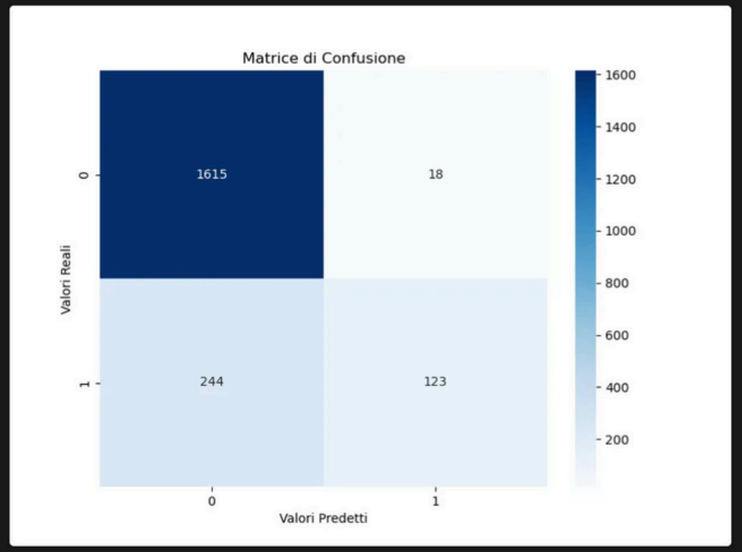
Accuracy: 0.8285
Precision: 0.5234
Recall: 0.7302
F1 Score: 0.6098



Risultati Standard Random Forest:

P_Y0 = 0.80

Accuracy: 0.8690
Precision: 0.8723
Recall: 0.3351
F1 Score: 0.4843



DGP - RISULTATI

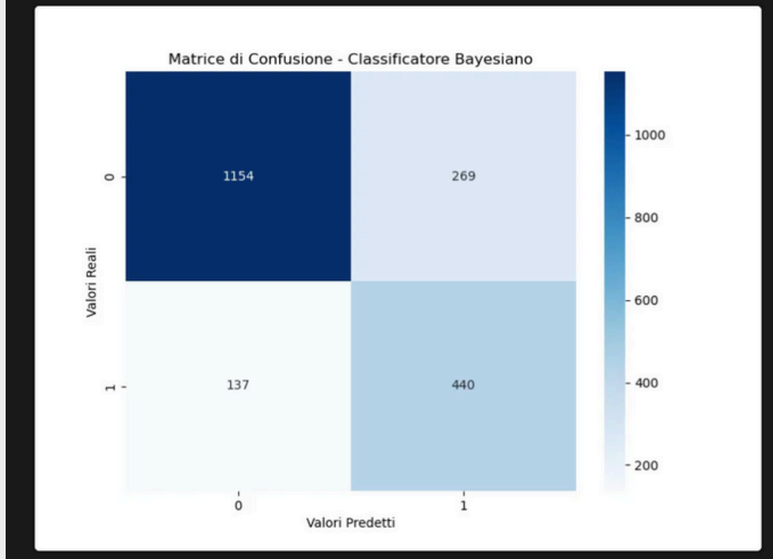
3.2°

70%

Risultati Naïve Bayes Random Forest:

P_Y0 = 0.70

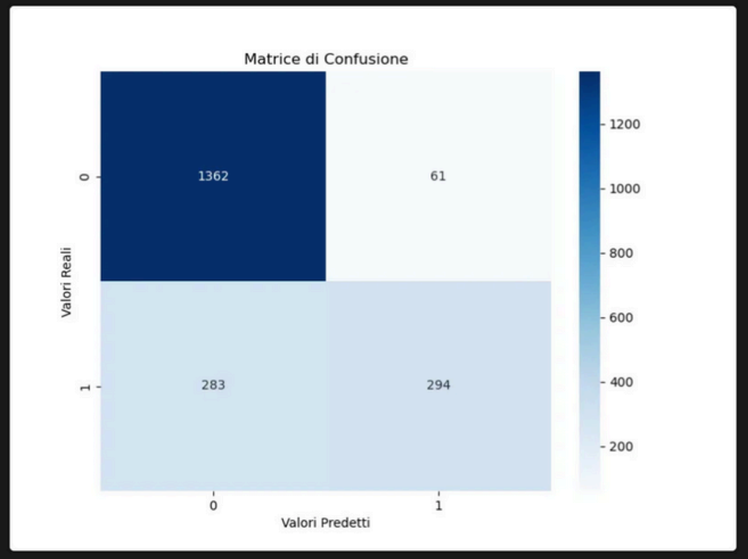
Accuracy: 0.7970
Precision: 0.6206
Recall: 0.7626
F1 Score: 0.6843



Risultati Standard Random Forest:

P_Y0 = 0.70

Accuracy: 0.8280
Precision: 0.8282
Recall: 0.5095
F1 Score: 0.6309



DGP - RISULTATI

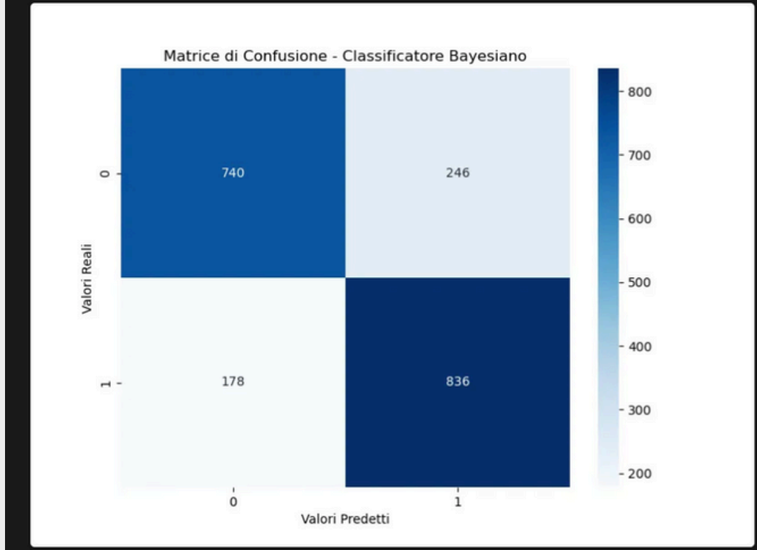
3.3°

50%

Risultati Naïve Bayes Random Forest:

P_Y0 = 0.50

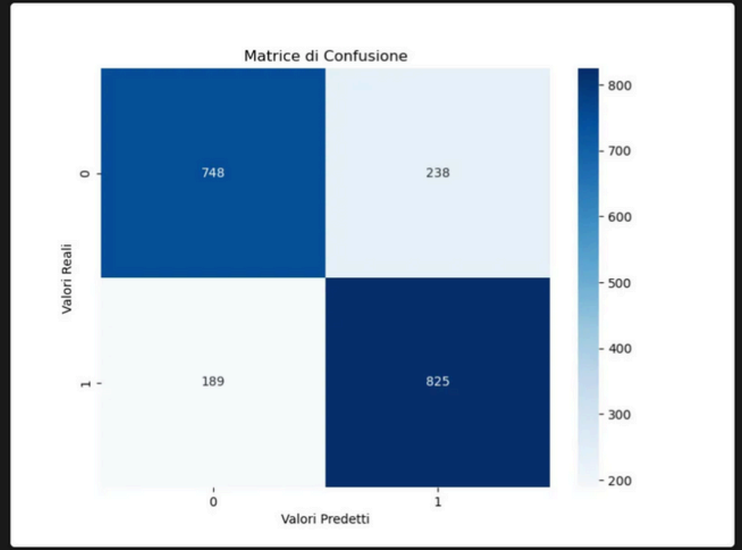
Accuracy: 0.7880
Precision: 0.7726
Recall: 0.8245
F1 Score: 0.7977



Risultati Standard Random Forest:

P_Y0 = 0.50

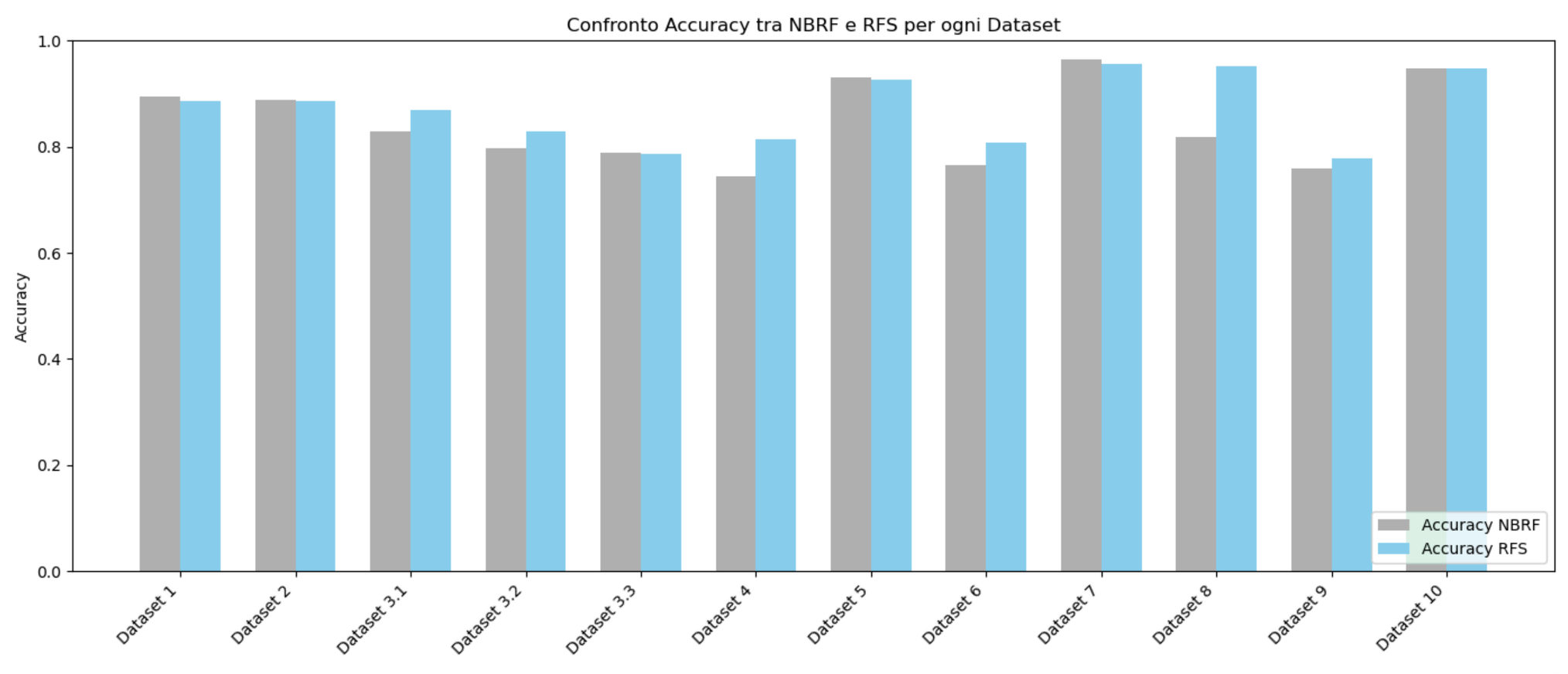
Accuracy: 0.7865
Precision: 0.7761
Recall: 0.8136
F1 Score: 0.7944



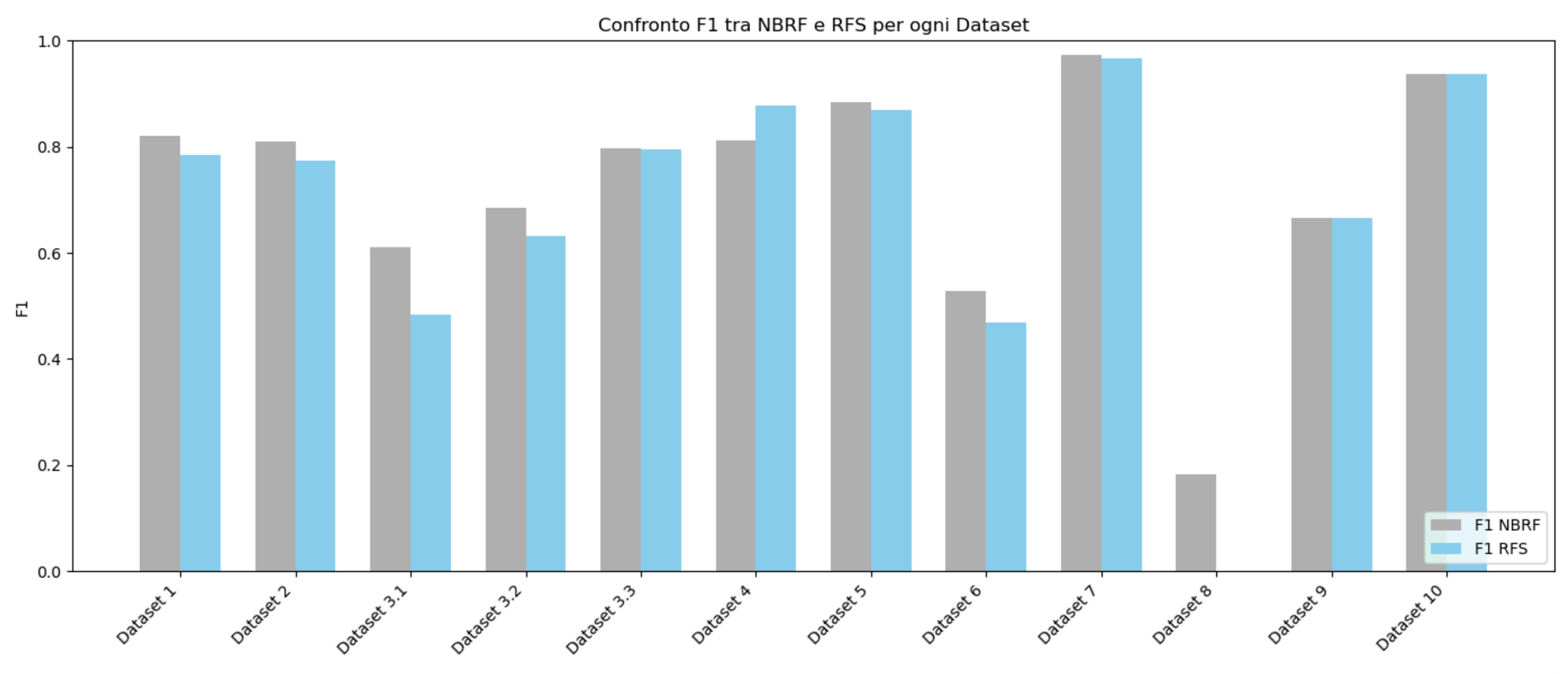
PANORAMICA DATASET

4°	**SouthGermanCredit.csv** • **Dimensione del campione**: 1.000 • **Sbilanciamento**: 70 - 30	8°	**Stroke_Prediction_Dataset.csv** • **Dimensione del campione**: 3.426 • **Sbilanciamento**: 94,7 - 5,3
5°	**Telco_customer_churn.csv** • **Dimensione del campione**: 7.032 • **Sbilanciamento**: 73,5 - 26,5	9°	**diabetes.csv** • **Dimensione del campione**: 768 • **Sbilanciamento**: 65,1 - 34,9
6°	**default_credit_card.csv** • **Dimensione del campione**: 30.000 • **Sbilanciamento**: 77,9 - 22,1	10°	**spambase.csv** • **Dimensione del campione**: 4.601 • **Sbilanciamento**: 60,6 - 39,4
7°	**Breast_Cancer_Wisconsin.csv** • **Dimensione del campione**: 569 • **Sbilanciamento**: 62,7 - 37,3		

PERFORMANCE SUI DATASET



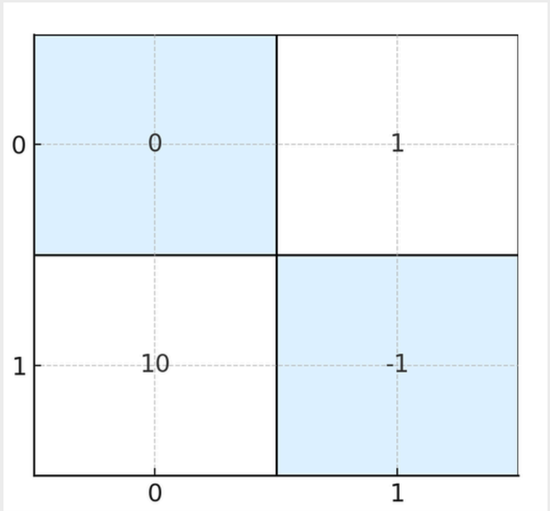
PERFORMANCE SUI DATASET



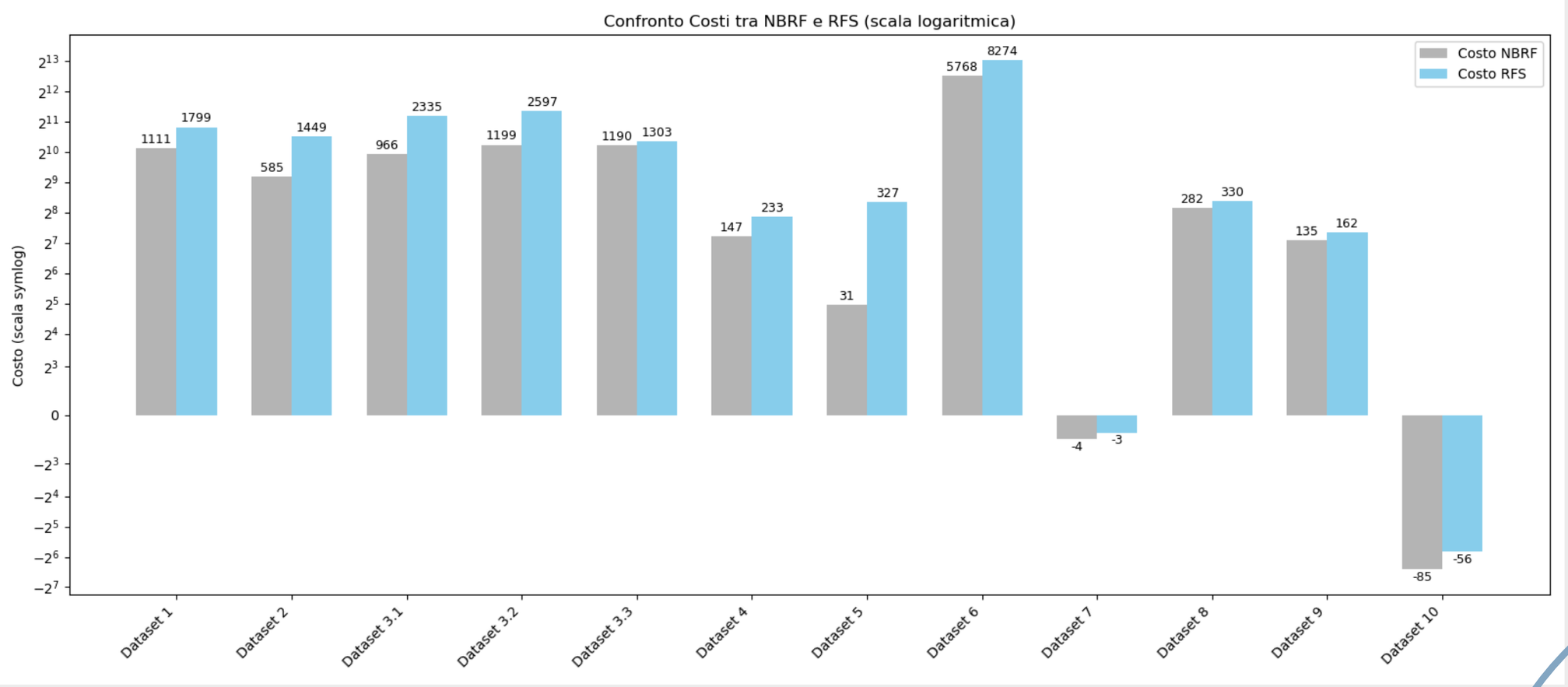
COST MATRIX

La **matrice dei costi**, è uno strumento utilizzato nell'ambito della classificazione per assegnare un costo specifico a ciascun tipo di previsione che il modello può compiere. In pratica, questo consente di **quantificare l'impatto** associato soprattutto agli errori del modello. In molte applicazioni, ad esempio in ambito medico, questo è **fondamentale** poiché l'impatto di un errore del modello può essere molto differente a seconda del tipo sbaglio commesso.

Nel nostro caso, si è scelto di utilizzare la seguente matrice dei costi, la quale assegna un costo maggiore alla classificazione della **classe di minoranza**. In particolare, l'errore più impattante è il **FALSO POSITIVO,** come spesso accade nelle applicazioni.



CONFRONTO DEI COSTI



CRITICITÀ

Come abbiamo visto il modello proposto sembra ottenere nei casi trattati un **minor costo** in termini di impatto degli errori e una maggiore capacità di riconoscere unità appartenenti alla classe di minoranza. Tuttavia si fanno presente le seguenti criticità:

- Il **costo computazionale** dell'algoritmo NBRF è in rapporto molto maggiore rispetto al RFS, il che è dovuto sia al maggior numero di calcoli da svolgere, sia al fatto che è stato implementato in linguaggio Python, contro il RFS implementato tramite i pachetti *scikit_learn*, che sono scritti in C;

- L'algoritmo NBRF utilizza informazioni presenti nel test test oltre a quelle presenti nel train set, in particolare per stimare la frequenza di falsi positivi e falsi negativi degli alberi. Questo gli dà un leggero **vantaggio informativo** rispetto al RFS.

ULTERIORI IMPLEMENTAZIONI

Ulteriori **migliorie** al modello potrebbero essere le seguenti:

- Eseguire una **stima particolare** per ogni albero della Random Forest sulla frequenza di FN e FP, salvare queste da una parte per poi ultilizzarle nella classificazione.
- Utilizzare lo **smoothing di Laplace** nel calcolo delle probabilità a posteriori per evitare zeri e rilassare i parametri. Anche se quest'ultimo problema non si è presentato nei nostri dataset, in esempi con dimensione minore potrebbero accadere problematiche di questo tipo.
- Implementare somme di **log-probabilità** nel calcolo delle probabilità a posteriori per evitare problemi di underflow e instabilità numerica. Anche questo problema non è stato riscontrato nei nostri esempi, tuttavia potrebbe essere un'implementazione desiderabile.
- Unificare i risultati dei due algoritmi per creare un modello ancora più sofisticato e preciso. Questo potrebbe essere fatto ad esempio implementando un metodo di **scelta dinamica** tutte le volte in cui i due modelli non sono d'accordo nella classificazione, magari scegliendo la predizione del modello che si sente più "sicuro" della sua scelta o che in passato ha sbagliato meno per quella particolare istanza.

BIBLIOGRAFIA

I Dataset analizzati:

- https://archive.ics.uci.edu/dataset/573/south%2Bgerman%2Bcredit%2Bupdate
- https://www.kaggle.com/datasets/yeanzc/telco-customer-churn-ibm-dataset
- https://archive.ics.uci.edu/dataset/350/default%2Bof%2Bcredit%2Bcard%2Bclients
- https://archive.ics.uci.edu/dataset/17/breast%2Bcancer%2Bwisconsin%2Bdiagnostic
- https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset?select=healthcare-dataset-stroke-data.csv
- https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database
- https://archive.ics.uci.edu/dataset/94/spambase

Altre fonti:

- Dispense e Slide corso di Statistical Learning Professoressa Gottard;
- Dispense e Slide corso di Data Mining Professoressa Merlini;
- "*Approccio Frequentista e Bayesiano, Due Modi Diversi di Vedere La Stessa Realtà*", Tesi di dottoratto di ricerca di *Manuela Giangreco, A.A 2012 - 2013,* Università degli Studi di Milano;
- Cursor per aiuto nella scrittura del codice;
- ChatGPT per la realizzazione dei grafici.

GRAZIE PER L'ATTENZIONE