Progetto di Machine Learning
Predizione Qualità Mele

Falbo Andrea 887525
Pellegrini Damiano 886261
Tenderini Ruben 879290

Anno Accademico 2024/2025

Indice

Introduzione e Obiettivi

Obiettivo

L'obiettivo del seguente elaborato è la progettazione di un modello di **classificazione** in grado di distinguere tra **mele di buona qualità** e **mele di scarsa qualità**.

Il modello utilizzerà le informazioni fornite dal dataset per poter apprendere quali **caratteristiche** differenziano tale categorie.

Dataset utilizzato: Apple Quality
Fonte: Kaggle
Link al Dataset

Librerie Utilizzate

Per lo sviluppo del modello e l'analisi dei dati, sono state prese in considerazione le seguenti librerie Python:

- **NumPy**
- **Pandas**
- **Matplotlib** e **Seaborn**
- **Scikit-learn**:
  - Metriche di valutazione
  - Preprocessing
  - Modelli di Machine Learning
  - Strumenti avanzati.
  - Creazione di pipeline
  - Ensemble
- **Warnings, random e os**

Design del Dataset

Descrizione del Dataset

Dataset utilizzato: **Apple Quality**.
Numero osservazioni: **4000**

Ogni esempio è descritto dalle seguenti variabili:

- **A_id**: Identificatore univoco per ciascun frutto.
- **Size**: Dimensione del frutto, con valori che variano tra -7.15 e 6.41.
- **Weight**: Peso del frutto, con valori che variano tra -7.15 e 5.79.
- **Sweetness**: Grado di dolcezza del frutto, con valori che variano tra -6.89 e 6.37.
- **Crunchiness**: Texture che indica la croccantezza del frutto, con valori che variano tra -6.06 e 7.62.
- **Juiciness**: Livello di succosità del frutto, con valori che variano tra -5.96 e 7.36.
- **Ripeness**: Stato di maturazione del frutto, con valori che variano tra -5.86 e 7.24.
- **Acidity**: Livello di acidità del frutto, con valori che variano tra -7.01 e 7.4.
- **Quality**: Qualità complessiva del frutto, dove i valori sono "good" per mele di buona qualità, e "bad" per mele di scarsa qualità."

Ipotesi e Assunzioni:

- Bilanciamento delle classi
- Assenza di valori mancanti
- Nessuna variabile categorica

Analisi Esplorativa dei Dati

Considerazioni Iniziali

Esplorazione dei dati:

1. **Visualizzazione Iniziale**
   a. Visualizzati i primi record.
   b. Ottenute informazioni generali
2. **Controllo Valori**
   a. Calcolati i valori distinti per ciascuna colonna.
   b. Rimosso record con valori nulli.
3. **Casting dei Tipi di Dato**
   a. Conversione delle colonne numeriche in float32.
   b. Binarizzazione della variabile target Quality (0 = "bad", 1 = "good").
4. **Bilanciamento delle Classi**
   a. Controllo dell'assunzione di bilanciamento delle classi

Prima

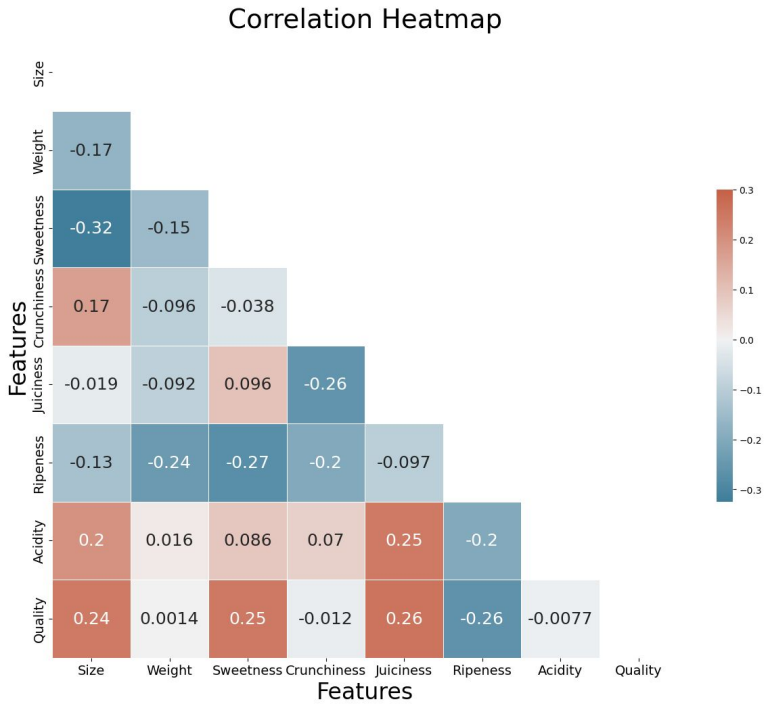
A_id	Size	Weight	Sweetness	Crunchiness	Juiciness	Ripeness	Acidity	Quality
0.0	−3.970	−2.512	5.346	−1.012	1.844	0.330	−0.492	good
1.0	−1.195	−2.839	3.664	1.588	0.853	0.868	−0.723	good
2.0	−0.292	−1.351	−1.738	−0.343	2.839	−0.038	2.622	bad
3.0	−0.657	−2.272	1.325	−0.098	3.638	−3.414	0.791	good
4.0	1.364	−1.297	−0.385	−0.553	3.031	−1.304	0.502	good

Dopo

	Size	Weight	Sweetness	Crunchiness	Juiciness	Ripeness	Acidity	Quality
0	−3.970 048	−2.512 336	5.346 330	−1.012 009	1.844 900	0.329 840	−0.491 590	1
1	−1.195 217	−2.839 257	3.664 059	1.588 232	0.853 286	0.867 530	−0.722 809	1
2	−0.292 024	−1.351 282	−1.738 429	−0.342 616	2.838 635	−0.038 033	2.621 636	0
3	−0.657 196	−2.271 627	1.324 874	−0.097 875	3.637 970	−3.413 761	0.790 723	1
4	1.364 217	−1.296 612	−0.384 658	−0.553 006	3.030 874	−1.303 849	0.501 984	1

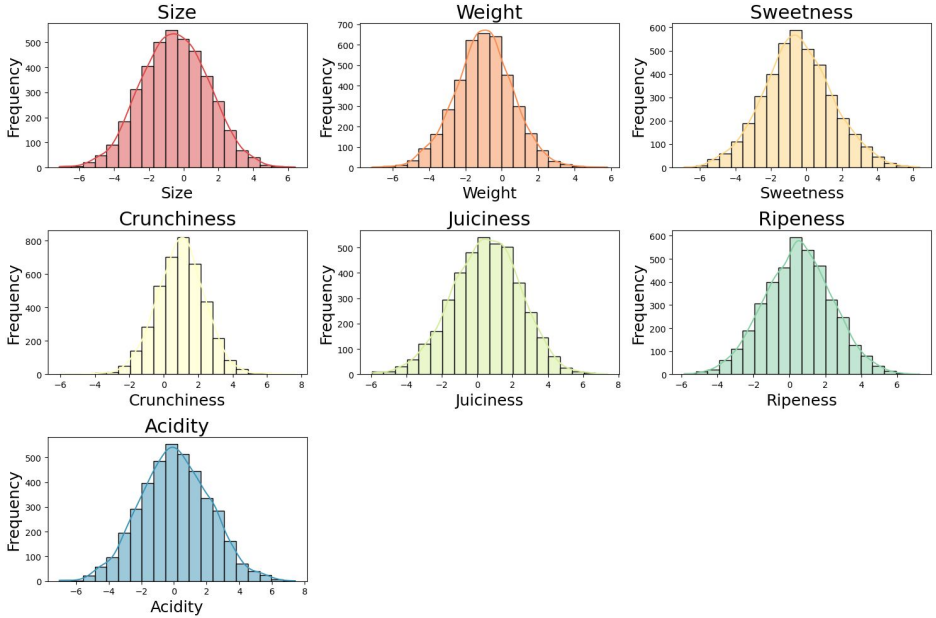
Matrice di Correlazione

Per esplorare le relazioni tra le variabili numeriche, è stata calcolata la **matrice di correlazione**

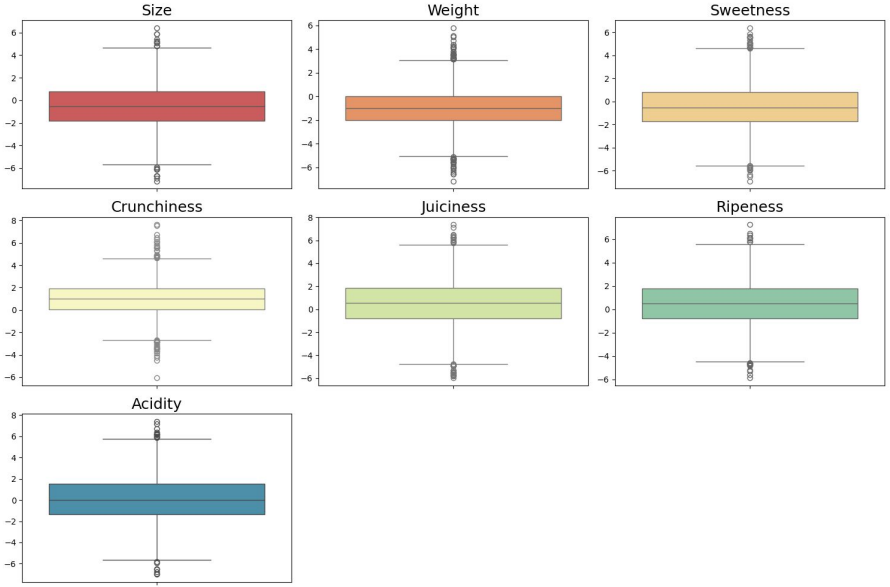


Distribuzioni variabili e Identificazione Outlier

Per analizzare la distribuzione delle variabili numeriche, sono stati creati **istogrammi con densità** (KDE).



Gli **outlier** sono stati identificati tramite i **boxplot**, che mostrano visivamente eventuali valori anomali nelle variabili.



Normalizzazione dei Dati

RobustScaler: Usato per rimuovere gli outlier presenti nel dataset.

StandardScaler: Applicato per ottenere una distribuzione normale (gaussiana) delle variabili.

MinMaxScaler: valutato ma non utilizzato.

Prima

	count	mean	std	min	25%	50%	75%	max
Size	4000.0	-0.503015	1.928058	-7.151703	-1.816765	-0.513703	0.805526	6.406367
Weight	4000.0	-0.989547	1.602507	-7.149848	-2.011770	-0.984737	0.030976	5.790714
Sweetness	4000.0	-0.470479	1.943441	-6.894485	-1.738425	-0.504758	0.801922	6.374916
Crunchiness	4000.0	0.985478	1.402757	-6.055058	0.062764	0.998249	1.894234	7.619852
Juiciness	4000.0	0.512118	1.930287	-5.961897	-0.801286	0.534219	1.835976	7.364403
Ripeness	4000.0	0.498277	1.874426	-5.864599	-0.771677	0.503445	1.766212	7.237837
Acidity	4000.0	0.076877	2.110271	-7.010539	-1.377424	0.022609	1.510493	7.404736
Quality	4000.0	0.501000	0.500062	0.000000	0.000000	1.000000	1.000000	1.000000

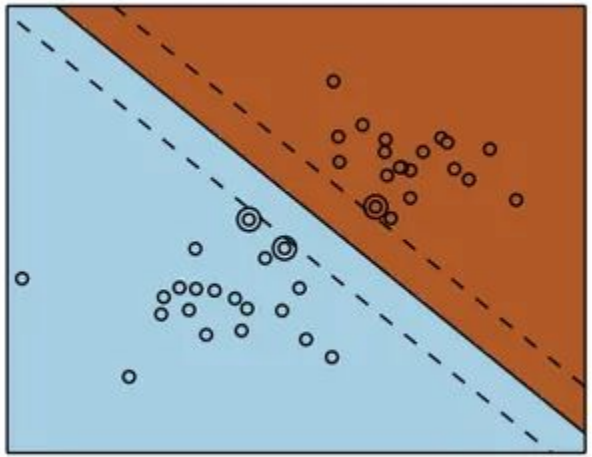
Dopo

	count	mean	std	min	25%	50%	75%	max
Size	4000.0	1.335 144	1.000 125	−3.448 816	−0.681 470	−0.005 544	0.678 768	3.584 043
Weight	4000.0	1.907 349	1.000 125	−3.844 645	−0.637 970	0.003 002	0.636 909	4.231 561
Sweetness	4000.0	−8.821 488	1.000 124	−3.305 895	−0.652 505	−0.017 641	0.654 797	3.522 747
Crunchiness	4000.0	8.583 068	1.000 125	−5.019 697	−0.657 868	0.009 106	0.647 917	4.730 115
Juiciness	4000.0	3.814 697	1.000 125	−3.354 335	−0.680 504	0.011 451	0.685 921	3.550 325
Ripeness	4000.0	1.430 511	1.000 126	−3.394 996	−0.677 601	0.002 757	0.676 523	3.595 980
Acidity	4000.0	−9.536 744	1.000 125	−3.358 955	−0.689 240	−0.025 720	0.679 437	3.472 910
Quality	4000.0	5.010 000	0.500 062	0.000 000	0.000 000	1.000 000	1.000 000	1.000 000

Modelli di Machine Learning

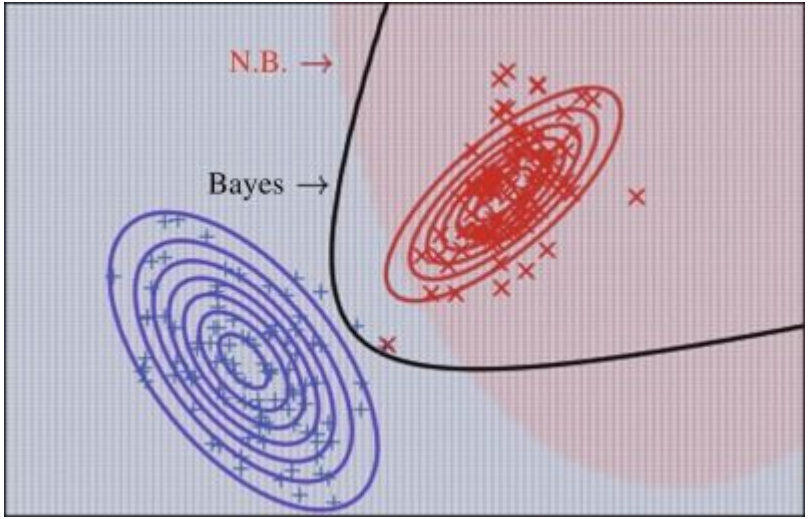
Support Vector Machine

Il modello di **Support Vector Machine** (SVM) si basa sulla ricerca di un iperpiano che separi al meglio le classi in uno spazio ad alta dimensione, cercando di massimizzare il margine tra le due classi.



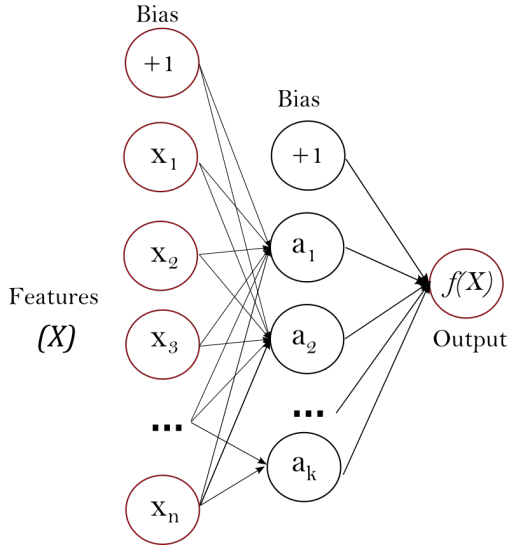
Naive Bayes

Naive Bayes si basa sul teorema di Bayes e sull'assunzione "banale" di indipendenza condizionale tra ogni coppia di features, dato il valore della variabile di classe.



Multilayer Perceptron

Il **Multilayer Perceptron** (MLP) è una rete neurale artificiale che utilizza uno o più strati nascosti per apprendere rappresentazioni complesse dei dati, applicando funzioni di attivazione non lineari.

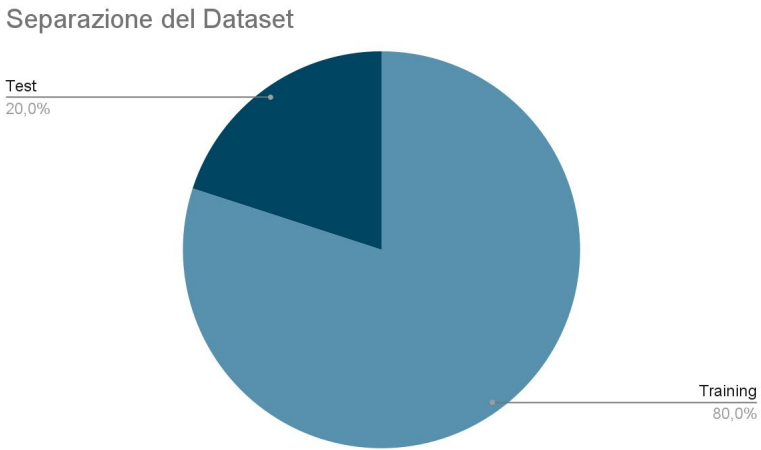


Esperimenti e Validazione

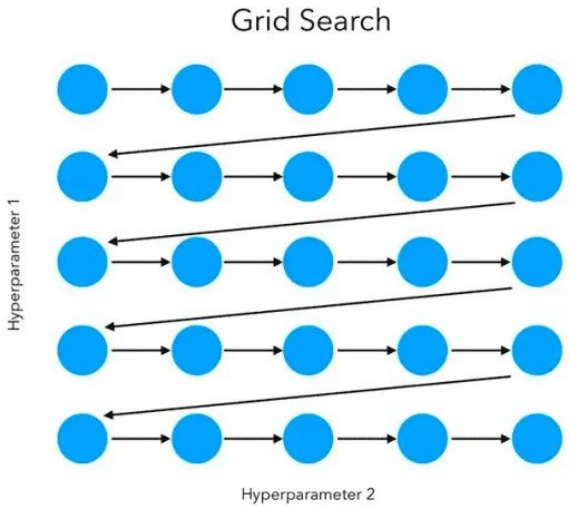
Separazione del Dataset e Grid Search

Separazione del dataset:

- **80%** dei dati assegnati alla fase di **training**
- **20%** dei dati assegnati alla fase di **test**

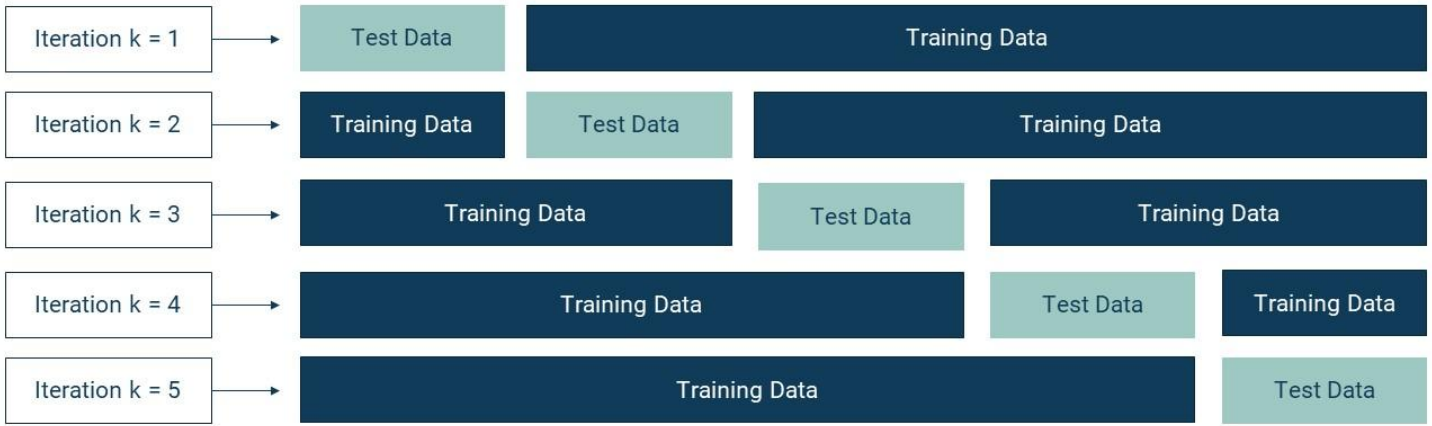


Utilizzo della **grid search** per effettuare ricerca esaustiva e ottimizzare gli iperparametri



Cross Validation

Eseguita **Cross Validation** per valutare le configurazioni dei modelli ottenuti dalla grid search. I dati di training sono stati suddivisi in 5 fold. Ogni fold è stato utilizzato una volta come test set, mentre gli altri 4 fold sono stati usati per addestrare il modello.



Analisi dei Risultati

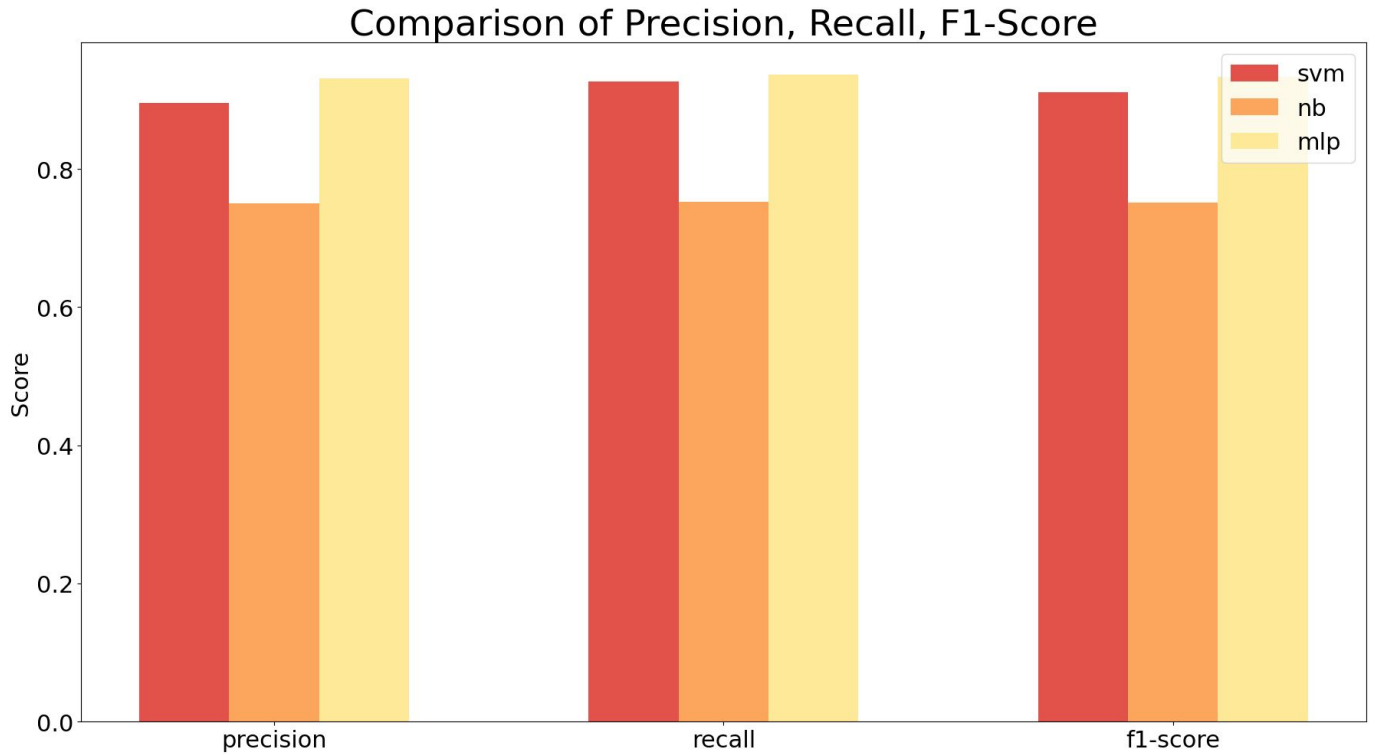
Analisi Training

Eseguita **Cross Validation** per valutare le configurazioni dei modelli ottenuti dalla grid search. I dati di training sono stati suddivisi in 5 fold. Ogni fold è stato utilizzato una volta come test set, mentre gli altri 4 fold sono stati usati per addestrare il modello.

Model	Precision
Support Vector Machine	0.907500
Naive-Bayes	0.751563
Multilayer Perceptron	0.929063

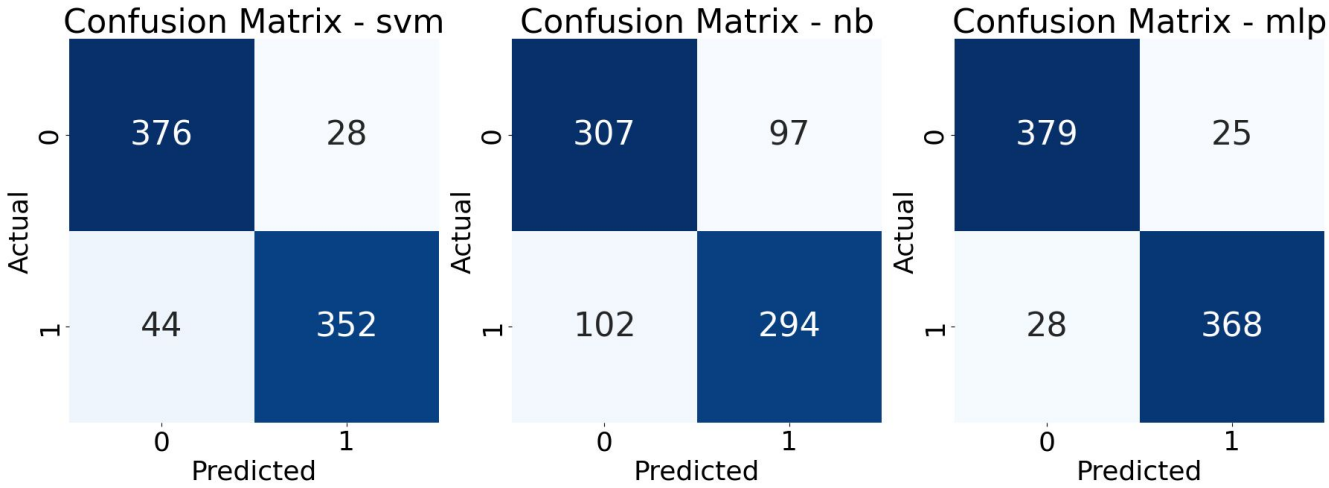
Confronto Metriche

In questo grafico vengono confrontate le prestazioni dei tre modelli in termini di **precision**, **recall** e **F1-score**.



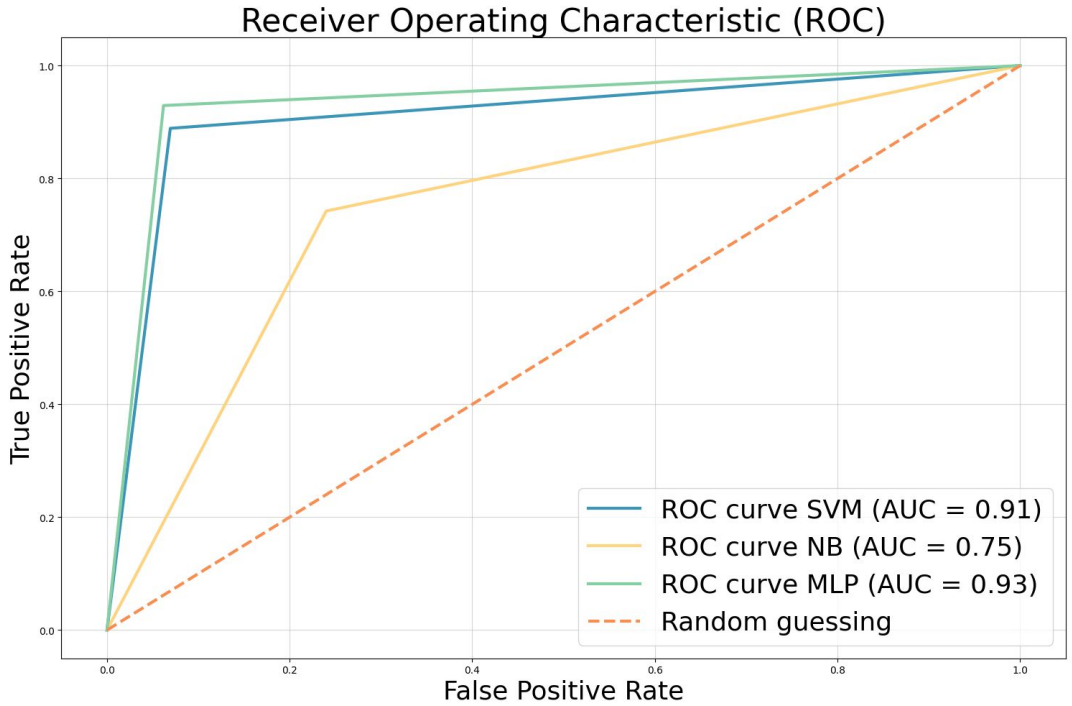
Matrice di Confusione

Le matrici di confusione mostrano la distribuzione degli **errori di classificazione** per ogni modello, indicando come le classi siano state predette erroneamente.



Curva ROC

La curva ROC evidenzia la capacità di **discriminazione tra le classi** per ciascun modello, con il modello MLP che mostra la migliore performance in termini di Area Under the Curve (AUC).



Conclusioni