Rilevamento Phishing URL

Analisi tramite Machine Learning: Pipeline dei Dati, Hyperparameter Tuning, Classificazione e Stress Test

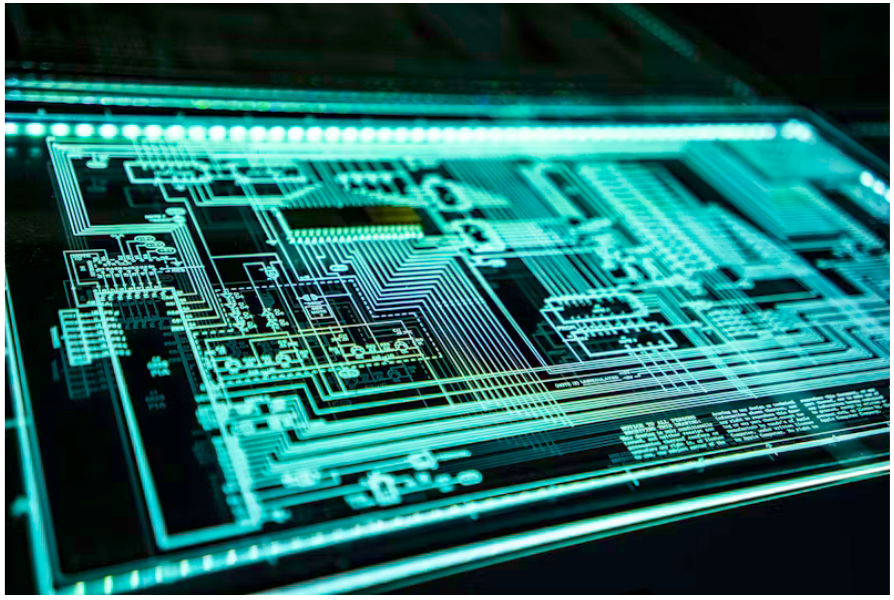
Il Task e la Sfida del Phishing

Definizione del Problema

Il **Phishing** è una tecnica fraudolenta di social engineering in cui un attaccante inganna l'utente per estorcere informazioni sensibili (credenziali, dati bancari) mascherandosi da entità affidabile.

Task di Machine Learning: Il task consiste nell'addestrare e confrontare due classificatori in grado di discriminare i siti legittimi (Classe 0) dai tentativi di phishing (Classe 1). La classificazione risulterà quindi di tipo binario.

Il Dataset: L'analisi si basa su un dataset strutturato in cui gli URL originali sono stati vettorializzati in 89 *feature* (caratteristiche metriche). Queste feature sono divise in 5 macro-categorie, che vanno dall'analisi strutturale della URL al ranking della pagina (es. AlexaRank e PageRank).



Preparazione Dati: Concetti Chiave

Target Encoding

È il processo di traduzione logica delle etichette testuali ("legitimate", "phishing") in un formato numerico binario discreto (0 e 1). Questa traduzione è un prerequisito fondamentale, poiché i modelli di ottimizzazione matematica richiedono esclusivamente tensori numerici in ingresso.

Data Splitting (80/20)

La divisione del dataset previene il memorizzamento a memoria (overfitting). Usiamo l'80% dei dati per l'addestramento e teniamo "nascosto" il 20% per il test finale. Viene applicata la **stratificazione** per assicurare che la proporzione esatta di siti sicuri/phishing rimanga inalterata in entrambi i set.

Feature Segregation

Consiste nel separare programmaticamente le 89 feature originarie in vettori categorici (es. presenza/assenza di un protocollo) e puramente numerici continui. Questo isolamento permette di applicare trasformazioni statistiche avanzate esclusivamente dove l'algoritmo ne beneficia realmente.

Feature Engineering: Distribuzione delle feature

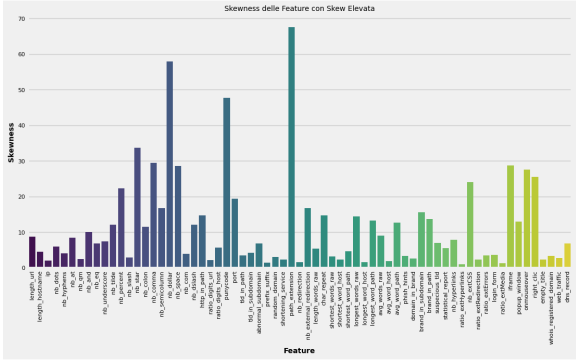
Skewness & Magnitudo

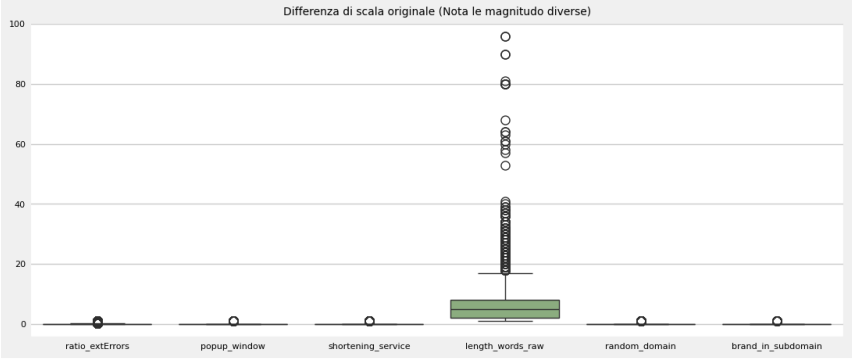
Calcolo della Skewness: La skewness misura il grado di asimmetria della distribuzione di una feature. Valori elevati indicano una forte asimmetria.

Calcolo della Magnitudo: La magnitudo è una misura della grandezza relativa di una feature. Per ogni feature, calcoliamo la magnitudo come il rapporto tra la media e la deviazione standard, per identificare quelle con variazione significativa.

Perchè calcolare questi due valori?

Le feature con skewness elevata e magnitudo significativa possono distorcere l'apprendimento del modello, poiché tendono a dominare la funzione di perdita durante l'ottimizzazione. Identificarle permette di applicare trasformazioni mirate (es. log-transform) per stabilizzare la varianza e migliorare la performance del modello.





Feature Engineering: Trasformazione

Log-Transform e Z-Score

Trasformazione Logaritmica: Molte feature naturali presentano distribuzioni fortemente asimmetriche con "code lunghe". Applicando il logaritmo naturale, comprimiamo la scala, stabilizzando la varianza e regolarizzando i picchi anomali (outlier).

Standardizzazione (Z-Score): Consiste nel "centrare" i dati rimuovendo la media e scalando per la deviazione standard.

$$z = \frac{x - \mu}{\sigma}$$

Senza questa operazione, i modelli basati sulle distanze geometriche (es. SVM) verrebbero completamente "accecati" da feature con magnitudo numerica maggiore, ignorando segnali deboli ma informativi.

I Modelli di Classificazione

I Modelli Baseline (Lineari)

Stabiliscono il "livello minimo" da superare per certificare l'apprendimento:

- **Dummy Classifier:** Modello ingenuo che prevede sempre la classe più frequente. È la baseline assoluta.
- **Logistic Regression:** Un potente classificatore lineare che stima la probabilità di appartenenza a una classe tramite la funzione sigmoide. Eccellente per misurare la naturale "separabilità lineare" dei dati.

I Modelli Complessi (Non-Lineari)

Architetture progettate per scovare correlazioni latenti e non-lineari:

- **Support Vector Machine (SVM):** Ricerca l'iperpiano ottimo che massimizza il margine di sicurezza tra le due classi tramite una rigida pipeline.
- **Random Forest:** Un metodo *Ensemble* robusto. Costruisce una foresta di alberi decisionali scorrelati e ne aggrega le predizioni tramite "voto". Unico metodo ensemble testato in profondità.

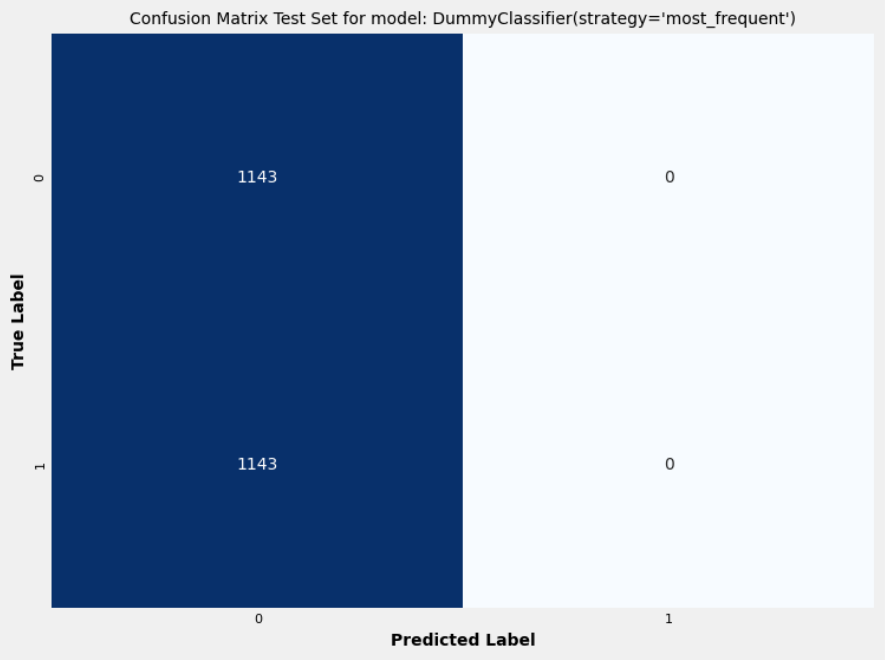
Analisi Baseline: Il Test Set

Interpretare la Confusion Matrix

La valutazione del **Dummy Classifier** sul Test Set evidenzia un fenomeno critico per l'analisi dei dati.

Essendo il dataset di validazione bilanciato, il Dummy (che prevede sempre 'Legittimo') raggiunge un'Accuracy del 50%. Tuttavia, la sua **Recall** sui tentativi di phishing è esattamente dello **0%**, poiché non "alza mai la bandiera" della minaccia.

Questo dimostra inequivocabilmente perché affidarsi alla sola Accuracy è letale in ambito Cybersecurity: un modello con un'elevata precisione apparente potrebbe essere completamente cieco ai vettori di attacco reali.



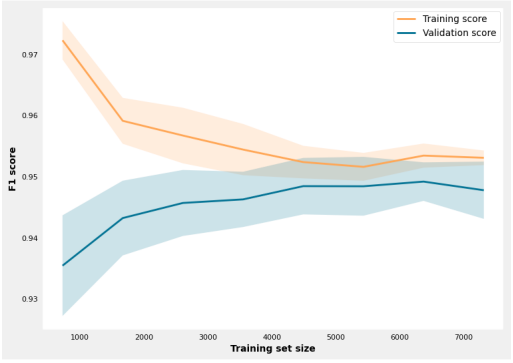
Logistic Regression: Apprendimento

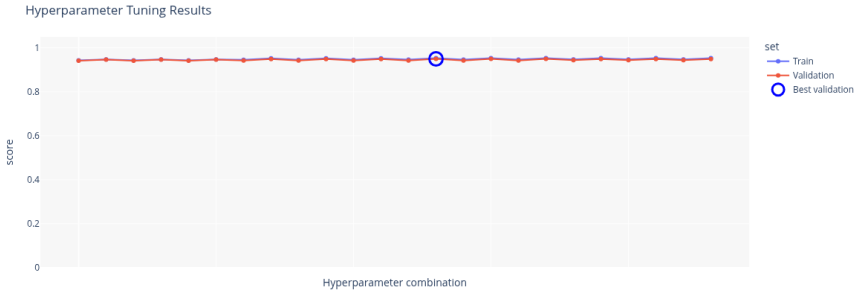
Analisi delle Learning Curves

L'addestramento della Logistic Regression (ottimizzata con `C=1.0` e max_iter=100 tramite GridSearch) ha mostrato una straordinaria solidità algoritmica.

L'analisi visiva delle **Learning Curves** conferma una convergenza stretta tra il punteggio sui dati di Training e quello sui dati di Validazione all'aumentare dei campioni.

Questa convergenza sintomatica indica che il modello **non soffre di Overfitting**. Tuttavia, l'assenza di miglioramenti ulteriori suggerisce la presenza di un lieve *Bias* strutturale: lo spazio vettoriale degli URL non è perfettamente separabile da una semplice retta lineare.





Support Vector Machine (SVM)

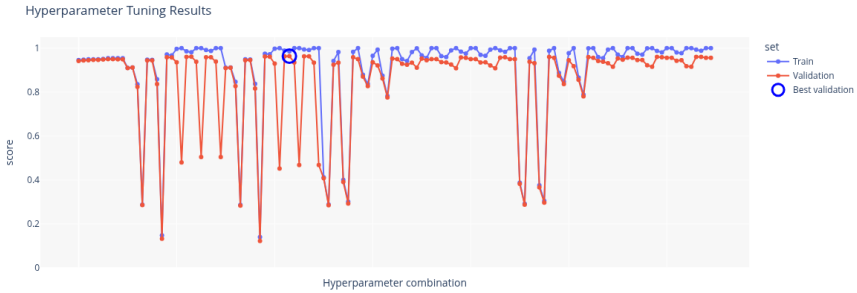
Tuning Iperparametri

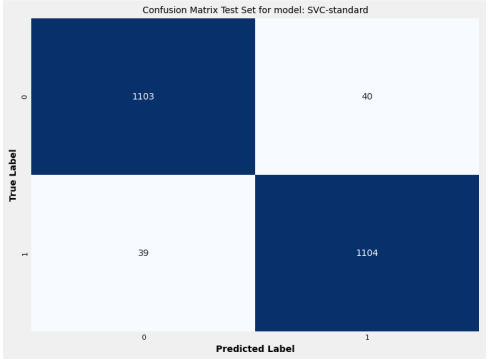
Per superare i limiti lineari, la SVM è stata potenziata introducendo il Kernel **RBF (Radial Basis Function)**. Questo operatore matematico proietta implicitamente le feature in uno spazio a infinite dimensioni, dove divengono linearmente separabili.

Ottimizzazione Iperparametri:

- `C = 5.0`: Permette un margine "morbido", penalizzando severamente gli errori ma tollerando lievi sovrapposizioni per garantire la generalizzazione.
- `Gamma = 0.01`: Definisce l'area di influenza di ogni punto.

Il modello definitivo fa perno su esattamente **1477 Support Vectors** (i punti critici al confine tra legittimo e phishing) per definire la topologia del Decision Boundary.



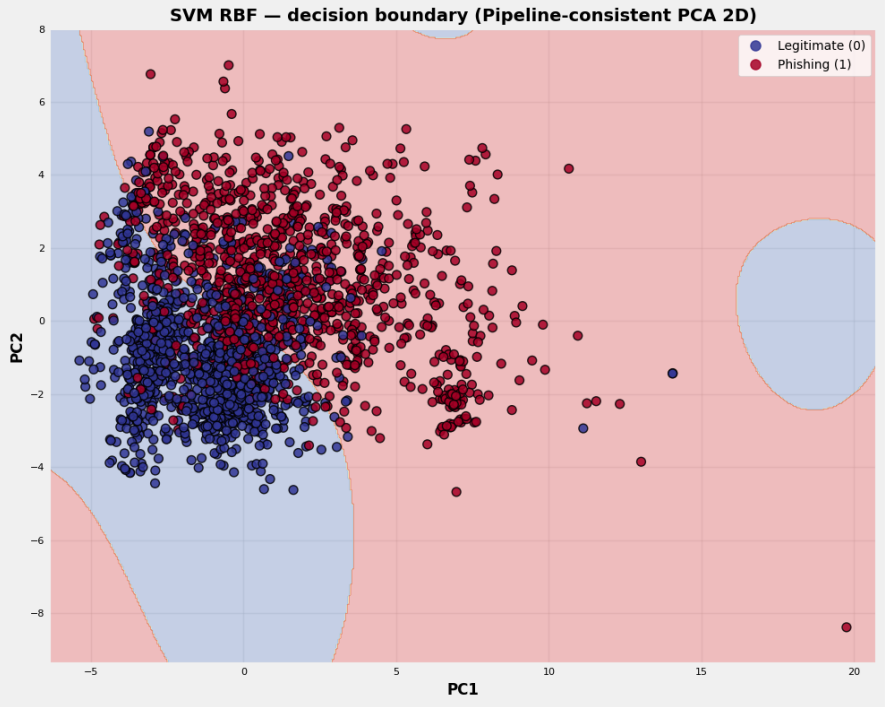


Support Vector Machine (SVM): Decision Boundary

Decision Boundary della SVM RBF

La decision boundary della SVM RBF è stata costruita in modo da massimizzare la separazione tra le classi, utilizzando i parametri ottimizzati: C = 5.0 e Gamma = 0.01.

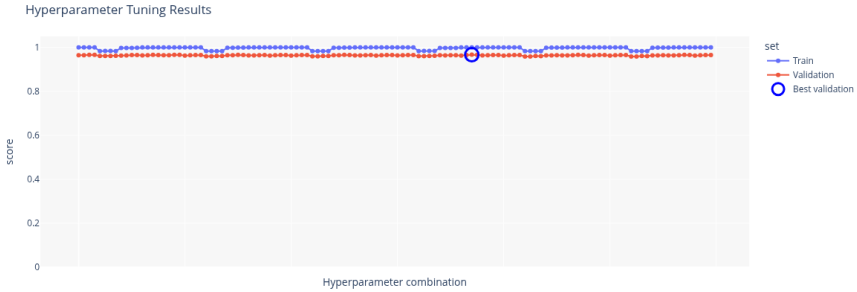
Da questa configurazione, si è ottenuta una decision boundary altamente robusta, in grado di distinguere con precisione tra URL legittimi e phishing, mantenendo un alto livello di generalizzazione.



Tuning: Random Forest (Ensemble)

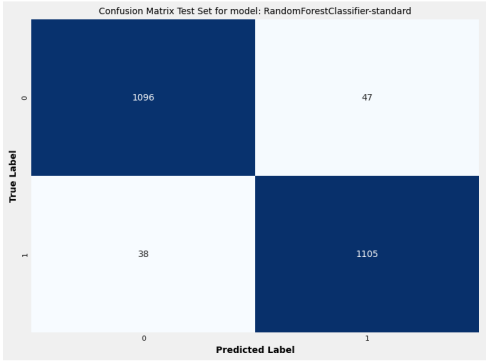
Stress Test sui Dati Grezzi vs Scalati

L'addestramento della Random Forest ha previsto un confronto diretto sulle feature log-trasformate e scalate contro i **dati originali grezzi (raw)**. Fedele alla natura degli alberi decisionali (insensibili alla varianza di scala), il modello ha mostrato performance superiori sui dati non trasformati.



Configurazione GridSearch Vincitrice:

- `n_estimators = 200`: L'aggregazione di 200 alberi indipendenti ha neutralizzato il rischio di overfitting.
- `criterion = 'gini'`: Metrica di impurità preferita per gli split nodali.
- `max_depth = None`: Gli alberi sono cresciuti fino alla purezza assoluta delle foglie.



Questo tuning ha conferito alla Random Forest la capacità di massimizzare categoricamente la **Recall pura**, rendendolo il modello teoricamente più letale contro i Falsi Negativi

Lo Stress Test SVM: Concetto

Obiettivo dell'Analisi di Resilienza

I modelli di Machine Learning tendono fisiologicamente ad "adagiarsi" su poche variabili dominanti trascurando i segnali deboli. Ma cosa accade in uno scenario avverso (Adversarial ML)?

Se un attaccante riuscisse a **mascherare o offuscare** le caratteristiche principali dell'URL (es. nascondendo parole chiave specifiche), il modello crollerebbe drasticamente o saprebbe riadattarsi? Lo Stress Test misura questa esatta tolleranza algoritmica al degrado dei dati.

Metodologia (Permutation Importance)

Per simulare l'attacco, la pipeline segue due step rigorosi:

1. **Calcolo**: Eseguiamo la Permutation Importance sul training set per identificare le variabili chiave per l'SVM. Il calcolo avverrà sia sfruttando la miglior Random Forest sia sfruttando il miglior modello SVM
2. **Rimozione**: Escludiamo dal dataset le feature che hanno ottenuto il punteggio di importanza più elevato.
3. **Rivalutazione**: Riaffiniamo e testiamo nuovamente il modello SVM sullo spazio delle feature ridotto.

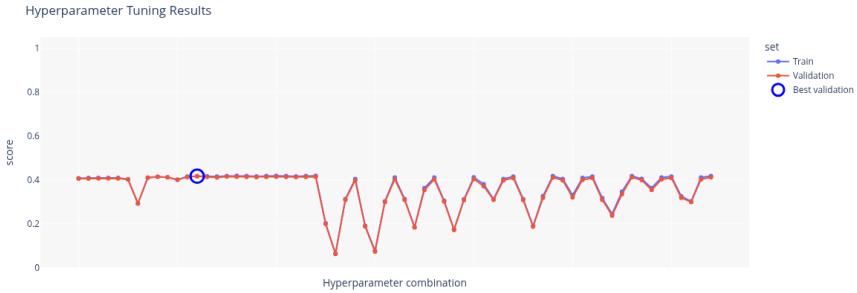
Risultati Stress Test: Feature Importance tramite RF

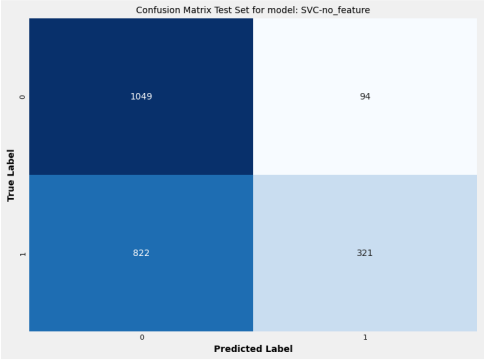
Risultati della Valutazione

La Random Forest, con la sua capacità di scovare correlazioni latenti, ha identificato le feature più informative per la classificazione. Rimuovendo queste feature chiave e testando nuovamente la SVM, abbiamo osservato un crollo drastico della performance, confermando la dipendenza critica del modello da queste variabili.

In questo caso, la SVM ha penalizzato drasticamente la sua capacità di riconoscere i siti truffa, evidenziando la sua vulnerabilità alle feature meno informative.

La **Recall è scesa da 0.96 a 0.28**, evidenziando la scarsa robustezza del modello in questo contesto simulato.





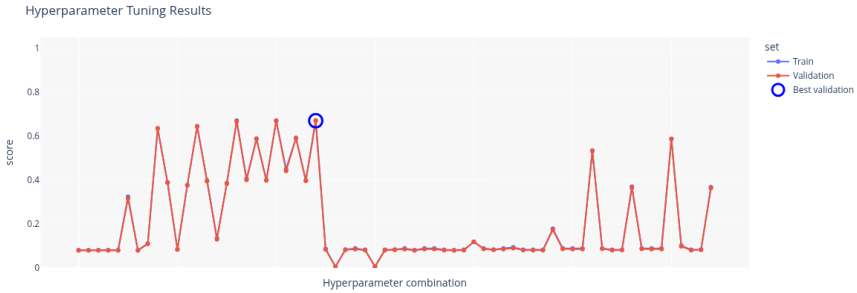
Risultati Stress Test: Feature Importance tramite SVM

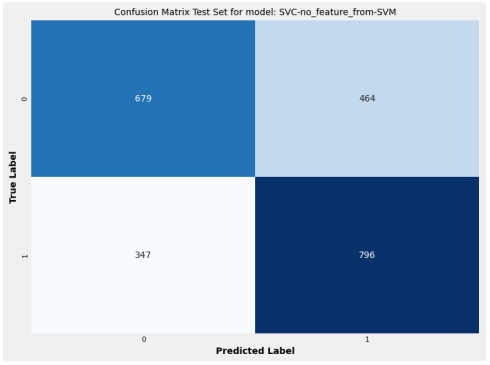
Risultati della Valutazione

La SVM, con la sua capacità di scovare correlazioni latenti, ha identificato le feature più informative per la classificazione. Rimuovendo queste feature chiave e testando nuovamente la SVM, abbiamo osservato un crollo drastico della performance, confermando la dipendenza critica del modello da queste variabili.

In questo caso, la SVM ha dimostrato una maggiore resilienza, riuscendo a mantenere una capacità di riconoscimento dei siti truffa più elevata rispetto al caso precedente.

È stata comunque penalizzata molto la **Precision**, creando quindi un sistema che non garantisce un alto livello di usabilità e user experience.





Le Metriche Definitive

Recall (Sensibilità)

Risponde a: *"Su 100 URL malevoli effettivi, quanti ne ho bloccati?"* In Cybersecurity, un Falso Negativo (sito truffa ignorato) causa danni letali. È la priorità assoluta di ottimizzazione.

Precision (Esattezza)

Risponde a: *"Se scatta l'allarme rosso, quanto è probabile che sia davvero Phishing?"* Una bassa Precision genera Falsi Positivi, bloccando l'accesso dell'utente a siti legittimi e minando l'usabilità del sistema.

F1-Score

È la media armonica tra Recall e Precision. Poiché queste due metriche sono spesso in trade-off (aumentare una fa scendere l'altra), l'F1-Score fornisce l'indicatore matematico definitivo del bilanciamento strutturale del classificatore.

Confronto Definitivo sul Test Set

MODELLO PREDITTIVO	ACCURACY	PRECISION	RECALL	F1-SCORE
Dummy Classifier (Baseline)	50.00%	0.00%	0.00%	0.00%
Logistic Regression (C=1.0)	93.83%	93.72%	93.96%	93.84%
Random Forest (Gini, n=200)	96.28%	95.92%	**96.68%**	96.30%
SVM (RBF, C=5.0)	**96.54%**	**96.50%**	96.59%	**96.55%**

Mentre la Random Forest eccelle di misura nella pura Recall categorizzando il maggior numero assoluto di minacce, la **Support Vector Machine RBF** trionfa nell'F1-Score globale, limitando drasticamente i Falsi Positivi e presentandosi come la soluzione ottimale e più equilibrata per la messa in produzione.