Relazione di Lab.Machine Learning. Classificazione: Pacchetti benigni e maligni

Valeria Nardoni

April 2024

1 Obiettivo

L'obiettivo del progetto consiste nell'addestrare un modello utilizzando tecniche di Machine Learning al fine di classificare correttamente i flussi di pacchetti come maligni o benigni.

2 Studio del dataset

Il dataset denominato NF-UQ-NIDS [1] contiene records relativi a flussi di rete dove sono presenti anche attacchi informatici. Il dataset è completo, privo di dati mancanti, e comprende complessivamente 46 colonne. Per scopi di analisi, sono state eliminate le chiavi che potrebbero influenzare eccessivamente il processo di classificazione dei pacchetti come maligni o benigni, così come le colonne non numeriche. Il numero di colonne utilizzate come feature per la classificazione è di 38. Sono stati considerati 70000 records, tuttavia, a seguito della rimozione di alcune chiavi, è probabile che siano state generate diverse righe duplicate nel dataset. Utilizzando la funzione **df.duplicated().sum()**, sono state individuate e identificate 21947 univoche. Al fine di mitigare il potenziale impatto negativo su eventuali modelli di classificazione utilizzati, le righe duplicate sono state rimosse dal dataset. Il dataset considerato è composto da 9887 elementi con Label = 0, indicando pacchetti benigni, e 12060 elementi con Label = 1, indicando pacchetti maligni.

3 Classificazione

Il dataset viene diviso in set di dati per addestramento e test utilizzando la funzione **train_test_split** del modulo **sklearn.model_selection** di Python. Viene creato un DataFrame "X" contenente tutte le colonne del dataset a eccezione della colonna 'Label', che rappresenta le caratteristiche o predittori utilizzati per l'addestramento del modello.Viene creata una serie y contenente solo

[1] https://www.kaggle.com/datasets/aryashah2k/nfuqnidsv2-network-intrusion-detection-dataset/data

la colonna 'Label', che rappresenta le etichette da predire. Il dataset è diviso in quattro parti:

- **X_train:** É l'80% delle caratteristiche utilizzate per l'addestramento del modello.
- **X_test:** É il 20% delle caratteristiche utilizzate per valutare le prestazioni del modello.
- **y_train:** Sono le etichette corrispondenti ai dati di addestramento.
- **y_test:** Sono le etichette corrispondenti ai dati di test.

3.1 Support Vector Classifier (SVC)

Metrica	**Valore**
Cross-validation accuracy	0.549
Test accuracy	0.548
Precision	0.548
Recall	1.00
F1-score	0.708
AUC-ROC	0.499

Tabella 1: Support Vector Classifier, kernel = poly

In base ai valori trovati in 1 l'accuratezza media ottenuta è circa 0.55, ovvero il modello ha una capacità predittiva aleatoria. Anche la Test accuracy, la percentuale di predizioni corrette rispetto al totale delle predizioni fatte sul set di test, ha un valore coerente con l'accuratezza calcolata sui fold. l'analisi più approfondita delle metriche di precisione, richiamo e F1-score rivela che il modello mostra una maggiore difficoltà nel rilevare correttamente i pacchetti maligni presenti nei dati di test, anche il valore basso dell'AUC-ROC suggerisce che il modello non ha capacità discriminante, è essenzialmente casuale nella sua capacità di fare predizioni. Tali affermazioni sono visibili anche nella seguente fig.1.

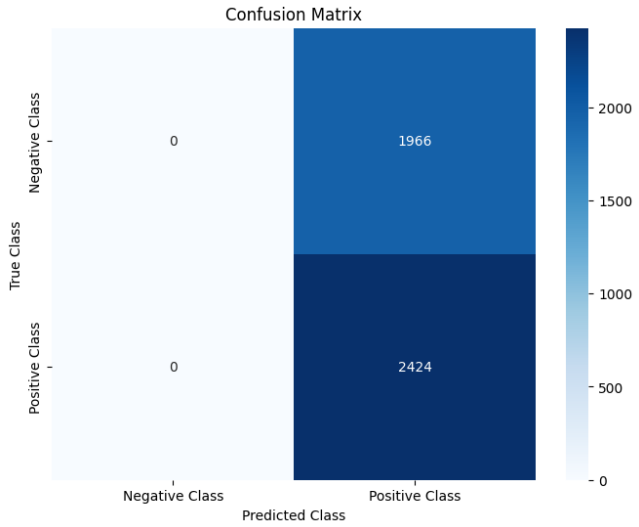


Figura 1: Matrice di confusione:SVC

I valori riportati nella tabella 1 e nella fig.1 si riferiscono al caso standard, dove sono stati utilizzati i parametri di default C = 1 e gamma = 'scale'. Tuttavia, a causa dei risultati non soddisfacenti, è stato eseguito un tuning dei parametri al fine di individuare dei risultati più ottimali. È emerso che il valore ottimale per ottenere i migliori risultati è gamma = 0.0001.

Hyperparameter	Value
gamma	0.0001

Tabella 2: Miglior iperparametro trovato

Metrica	Valore
Cross-validation accuracy	0.816
Test accuracy	0.825
Precisione	0.988
Recall	0.691
F1-score	0.813
AUC-ROC	0.912

Tabella 3: Risultati delle metriche di valutazione

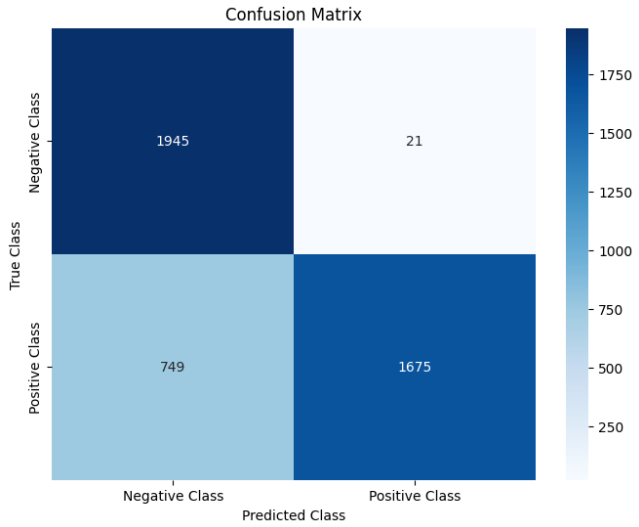


Figura 2: Matrice di confusione:SVC,tuning

Dalla tabella 10 e dalla Figura 2, è evidente come le metriche migliorino significativamente dopo l'ottimizzazione dei parametri. Prima del tuning, il modello aveva una capacità predittiva casuale, ma con l'adeguamento dell'iperparametro, la sua capacità di predizione e classificazione è notevolmente migliorata. I valori ottenuti nelle metriche riflettono una performance accettabile del modello.

3.2 LogisticRegression

Metrica	Valore
Cross-validation accuracy	0.743
Test accuracy	0.769
Precision	0.944
Recall	0.620
F1-score	0.748
AUC-ROC	0.838

Tabella 4: Logistic Regression

I risultati dell'analisi predittiva utilizzando il modello di regressione logistica mostrano prestazioni buone nel distinguere tra classi di pacchetti di rete benigni e maligni. Durante la cross-validation, il modello ha raggiunto un'accuratezza media del 74.34%, mentre sull'insieme di test l'accuratezza è stata leggermente più alta, pari al 76.99%. La precisione del modello è notevolmente elevata, raggiungendo il 94.41%, tuttavia, va notato che la recall è leggermente più bassa, attestandosi al 62.00%, il che suggerisce che il modello potrebbe avere difficoltà

a individuare tutti i pacchetti benigni presenti nel dataset. L'F1-score, che rappresenta una combinazione di precisione e richiamo, è del 74.85%, indicando un buon equilibrio tra le due metriche. Infine, l'AUC-ROC, che misura la capacità discriminante del modello, è pari a 0.84, indica una buona capacità del modello di distinguere correttamente tra pacchetti benigni e maligni. Queste affermazioni sono evidenziate anche nella seguente fig.3.

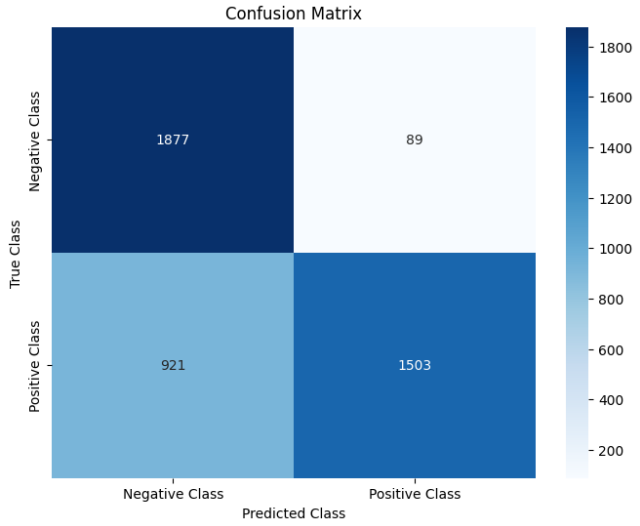


Figura 3: Matrice di confusione:LogisticRegression

3.3 Random Forest

Metrica	Valore
Cross-validation accuracy	0.977
Test accuracy	0.976
Precision	0.984
Recall	0.974
F1-score	0.978
AUC-ROC	0.994

Tabella 5: Random Forest

I risultati ottenuti utilizzando il modello Random Forest indicano un'eccellente capacità di distinguere tra pacchetti di rete benigni e maligni. Durante la cross-validation, il modello ha raggiunto un'accuratezza media del 97.76%, confermata anche sui dati di test con un'accuratezza del 97.68%. La precisione del modello è notevolmente alta, attestandosi al 98.42%. Inoltre, il modello ha un alto valore di recall, pari al 97.36%, suggerendo che riesce a individuare la maggior parte dei pacchetti benigni presenti nel dataset. L'F1-score, che

rappresenta una combinazione di precisione e recall, è del 97.88%, indicando un eccellente equilibrio tra le due metriche. Infine, l'AUC-ROC, che misura la capacità discriminante del modello, è pari a 0.99, indicando una performance eccezionale nel distinguere correttamente tra pacchetti benigni e maligni. La matrice di confusione del seguente modello e rappresentata dalla fig.4.

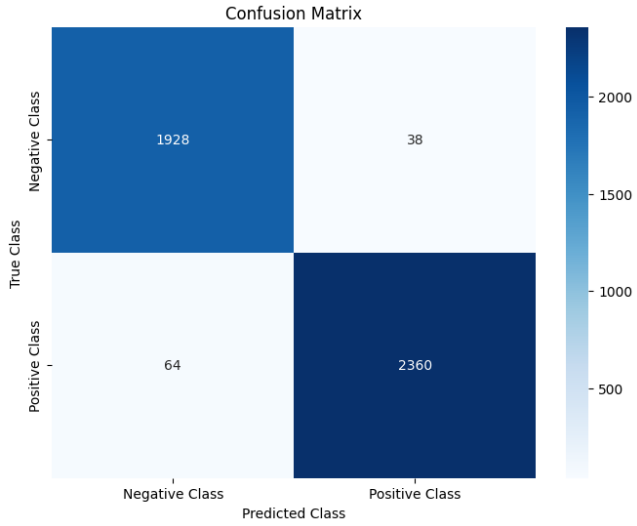


Figura 4: Matrice di confusione:RandomForest

Anche se i risultati attuali sono già soddisfacenti, è stato ritenuto opportuno eseguire il tuning dei parametri poiché può contribuire a creare modelli più robusti che generalizzano meglio su dati non osservati.

Hyperparameters	Value
max_depth	10
n_estimators	200

Tabella 6: Miglior iperparametri trovati

Metrica	Valore
Cross-validation accuracy	0.978
Test accuracy	0.978
Precisione	0.983
Recall	0.977
F1-score	0.980
AUC-ROC	0.996

Tabella 7: Risultati delle metriche di valutazione

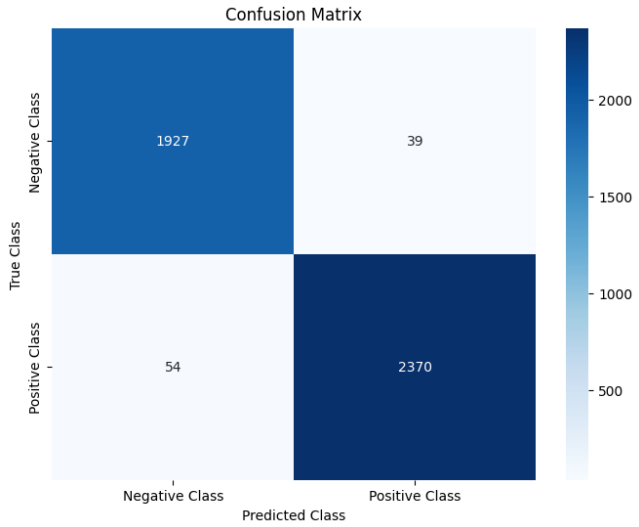


Figura 5: Matrice di confusione:RandomForest, tuning

Come osserviamo nella fig.7 aumenta lievemente il numero di classificazioni positive corrette e diminuisce di una sola unità il numero di classificazioni negative corrette. Anche le metriche migliorano leggermente.

3.4 K-Nearest Neighbors

Metrica	Valore
Cross-validation accuracy	0.944
Test accuracy	0.949
Precision	0.953
Recall	0.955
F1-score	0.954
AUC-ROC	0.978

Tabella 8: K-Nearest Neighbors

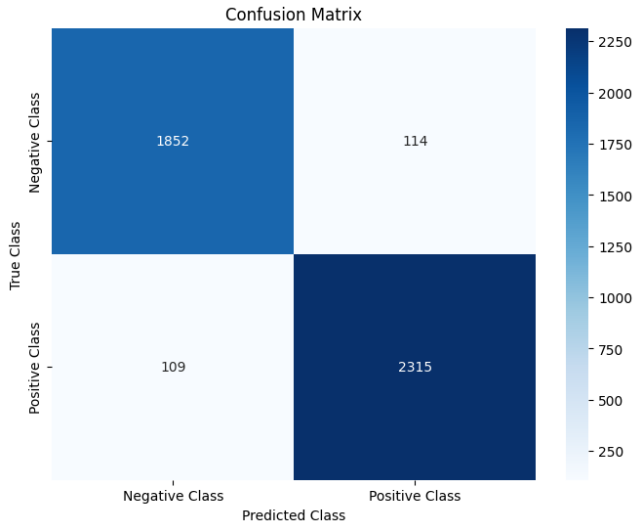


Figura 6: Matrice di confusione:K-Nearest Neighbors

I risultati ottenuti dall'analisi predittiva utilizzando il modello K-Nearest Neighbors sono incoraggianti. Durante la cross-validation, il modello ha dimostrato una buona capacità di generalizzazione, con un'accuratezza media del 94.41%. Tale accuratezza è stata confermata anche sui dati di test, con un'accuratezza del 94.92%. La precisione del modello è risultata essere del 95.31%. La recall del modello è del 95.50%. L'F1-score è del 95.40%, indicando un buon equilibrio tra le due metriche. Infine, l'AUC-ROC è pari a 0.978, confermando l'ottima capacità del modello di distinguere correttamente tra pacchetti benigni e maligni. Complessivamente, i risultati indicano che il modello K-Nearest Neighbors ha prestazioni solide nella classificazione dei pacchetti di rete in benigni e maligni.Nonostante le ottime prestazioni è stato ritenuto utile fare tuning, come parte del processo di miglioramento del modello e per garantire una migliore generalizzazione.

Hyperparameter	Value
n_neighbors	3

Tabella 9: Miglior iperparametro trovato

Metrica	Valore
Cross-validation accuracy	0.945
Test accuracy	0.946
Precisione	0.952
Recall	0.950
F1-score	0.954
AUC-ROC	0.972

Tabella 10: Risultati delle metriche di valutazione

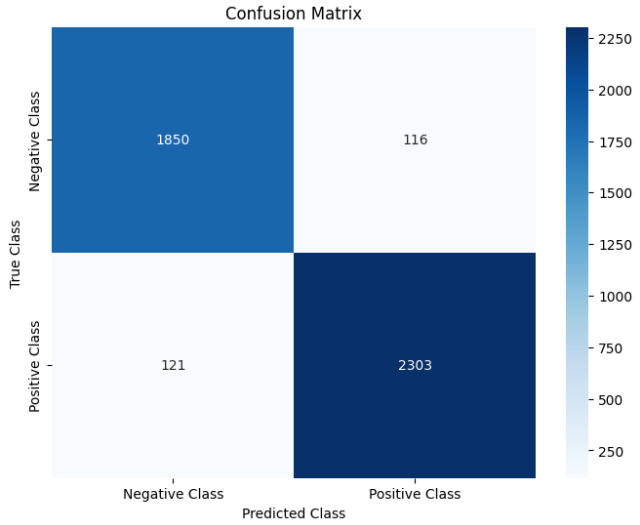
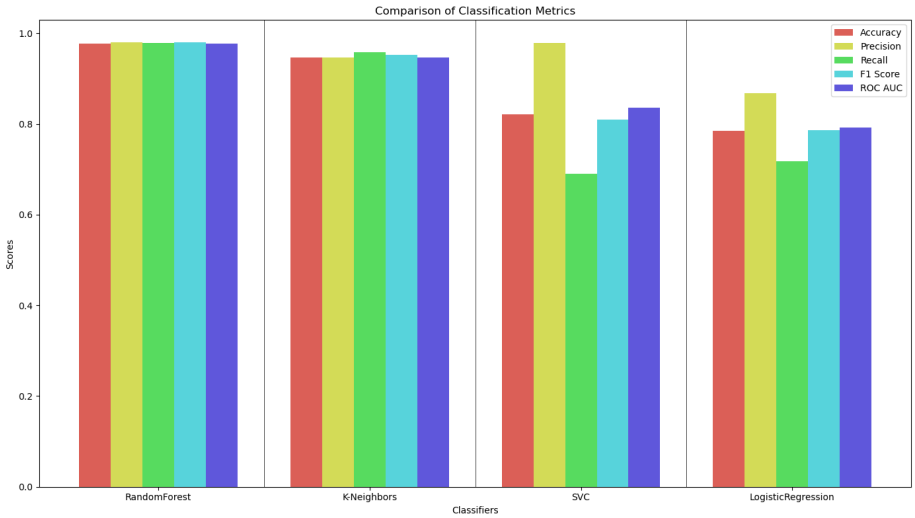


Figura 7: Matrice di confusione: K-Nearest Neighbors, tuning

. Anche dopo il tuning, le metriche rimangono sostanzialmente stabili, senza cambiamenti significativi. Nonostante ciò, i risultati rimangono soddisfacenti.

3.5 Confronto dei modelli



Model	**AUC**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
RandomForest	0.977511	0.977677	0.980042	0.979227	0.979634
K-Neighbors	0.945944	0.947153	0.945879	0.958455	0.952125
SVC	0.835706	0.821640	0.978210	0.690071	0.809257
LogisticRegression	0.792684	0.785421	0.868276	0.717491	0.785714

Tabella 11: Metriche per i diversi modelli

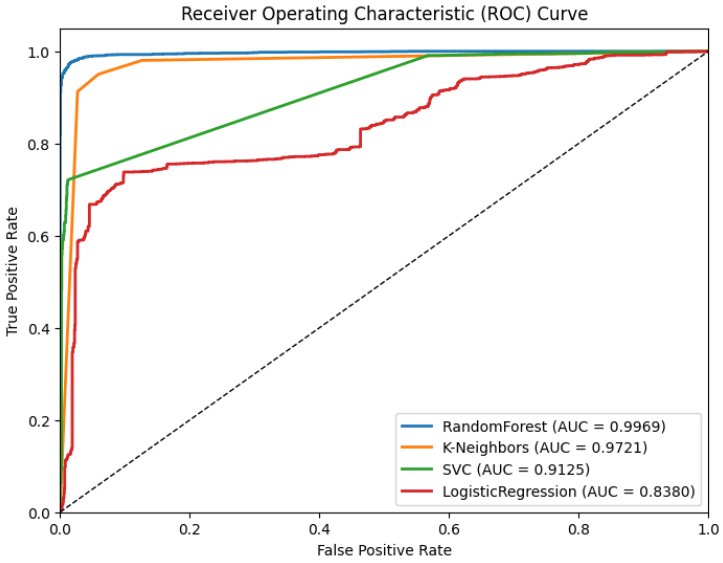


Figura 8: Curva ROC

.

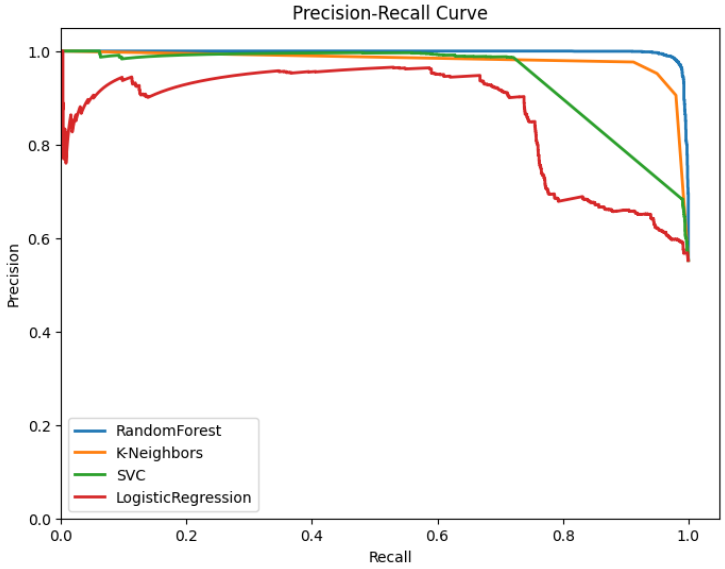


Figura 9: Curva Precision-Recall

Nelle figure 8 e 9 sono state rappresentate curve ROC e Precision-Recall per i modelli utilizzati. Nella figura 8, sull'asse x è rappresentato il "False Positive

Rate", che indica la proporzione di falsi positivi rispetto al totale dei negativi reali, mentre sull'asse y è rappresentato il "True Positive Rate", che rappresenta la proporzione di veri positivi rispetto al totale dei positivi reali. Un modello che classifica in modo perfetto avrà un'area sotto la curva (AUC) pari a 1, mentre un modello casuale avrà un'area di circa 0.5. Come è possibile osservare, tutti i modelli presentati hanno valori elevati di AUC, indicando una buona capacità di classificazione. Il secondo grafico fig.9 indica la curva precision-recal, Un'alta precisione e un alto richiamo indicano che il modello è in grado di identificare correttamente la maggior parte delle istanze positive e di minimizzare gli errori di classificazione.In entrambi grafici, come potevamo aspettarci dai risultati precedentemente ottenuti, osserviamo che il RandomForest ottiene i risultati migliori, seguito da K-Neighbors, SVC e LogisticRegression. Questi risultati sono in linea con la teoria. La LogisticRegression, sebbene sia una scelta valida nei problemi di classificazione binaria, potrebbe non funzionare bene se la relazione tra le variabili indipendenti e dipendenti non è lineare. Inoltre, potrebbe non adattarsi ai dati in modo flessibile come altri modelli più complessi come il Random Forest.

4 Conclusioni

Le tecniche usate mostrano come il Machine Learning possa rivelarsi uno strumento potente e promettente per il rilevamento delle minacce informatiche nelle reti. La sicurezza informatica, in un mondo sempre più interconnesso assume un ruolo cruciale. Utilizzare modelli di Machine Learning, potrebbe essere una buona strategia per individuare attività sospette o malevole con elevata precisione e tempestività.