

# Phishing URL Detection con CART



# Indice

## 01 Data

Quali dati stiamo usando?

## 02 Mission

Qual è il nostro obiettivo?

## 03 Data Visualization

Come visualizziamo i dati di cui disponiamo?

## 04 Data Preparation

Come prepariamo i dati per l'analisi?

## 05 Algorithm

Come elaboriamo i dati?

## 06 Results

Quali risultati abbiamo ottenuto?

## 07 Valuation

I risultati ottenuti sono accurati e validi?

01

## Il Phishing

Il phishing è una tecnica di attacco informatico progettata per indurre l'utente a **rivelare informazioni sensibili** come credenziali di accesso, dati bancari o informazioni personali.

Gli attaccanti sfruttano **comunicazioni apparentemente legittime** con l'obiettivo di **manipolare** la vittima e portarla ad agire in modo **inconsapevole**.

Si tratta di una delle minacce più diffuse e pericolose, perché punta direttamente sul **fattore umano**, aggirando anche le difese tecniche più robuste.



01

## Il Dataset: PhiUSIIL Phishing URL (Website)

il **PhiUSIIL Phishing URL Dataset** preso da **UC Irvine Machine Learning Repository**, è un dataset che comprende **235.795 istanze**, (ognuno con **54 features** reali/categoriche/intere) di **URLs legittimi e di phishing** (**le due classi sono bilanciate**).  
Le feature vengono estratte dal **codice sorgente delle pagine web** e dall'**URL** stesso.

15.94,66755.39,0,0,0,
9.12,42826.99,0,0,0,
35.64,50656.8,0,0,0,
15.94,67905.07,0,0,0,
115.94,66938.9,0,0,0,
192.49,86421.04,0,0,0,
72798.5,0,0,0,0,

# 02 Mission del progetto

Cosa vogliamo ottenere dai dati in nostro possesso?

## Visualizzazione

Effettuare un'**analisi visiva** dei dati del dataset attraverso grafici che ci danno una visione generale sugli URL presenti.

In particolare, ci avvaliamo di:

- **Teoria della Gestalt** per strutturare **grafici intuitivi**, che consentano una **lettura immediata** e naturale delle informazioni.
- **Data Ink Ratio** di Tufte per **massimizzare** la quantità di **informazione utile** e **ridurre** gli elementi grafici **non necessari**.

## Classificazione

Addestrare un **albero decisionale (CART)** che **classifica** un URL usando **feature** presenti distinguendo in:

- **URL legittimo**, un indirizzo web che porta a un sito vero, **sicuro**, ufficiale e con comportamenti normali.
- **URL malevolo**, un indirizzo creato apposta per **ingannare**, rubare dati o infettare chi ci clicca.

## Validazione

Verificare la **correttezza** del **modello** confrontando le sue **predizioni** con i **dati reali** valutando specialmente:

- **L'accuratezza** del modello sui dati di test.
- La **matrice di confusione** per misurare **errori** e **correttezza** nelle due classi.
- **ROC** e **AUC** per capire quanto bene il modello **distingue** URL legittimi da URL malevoli.

## 03

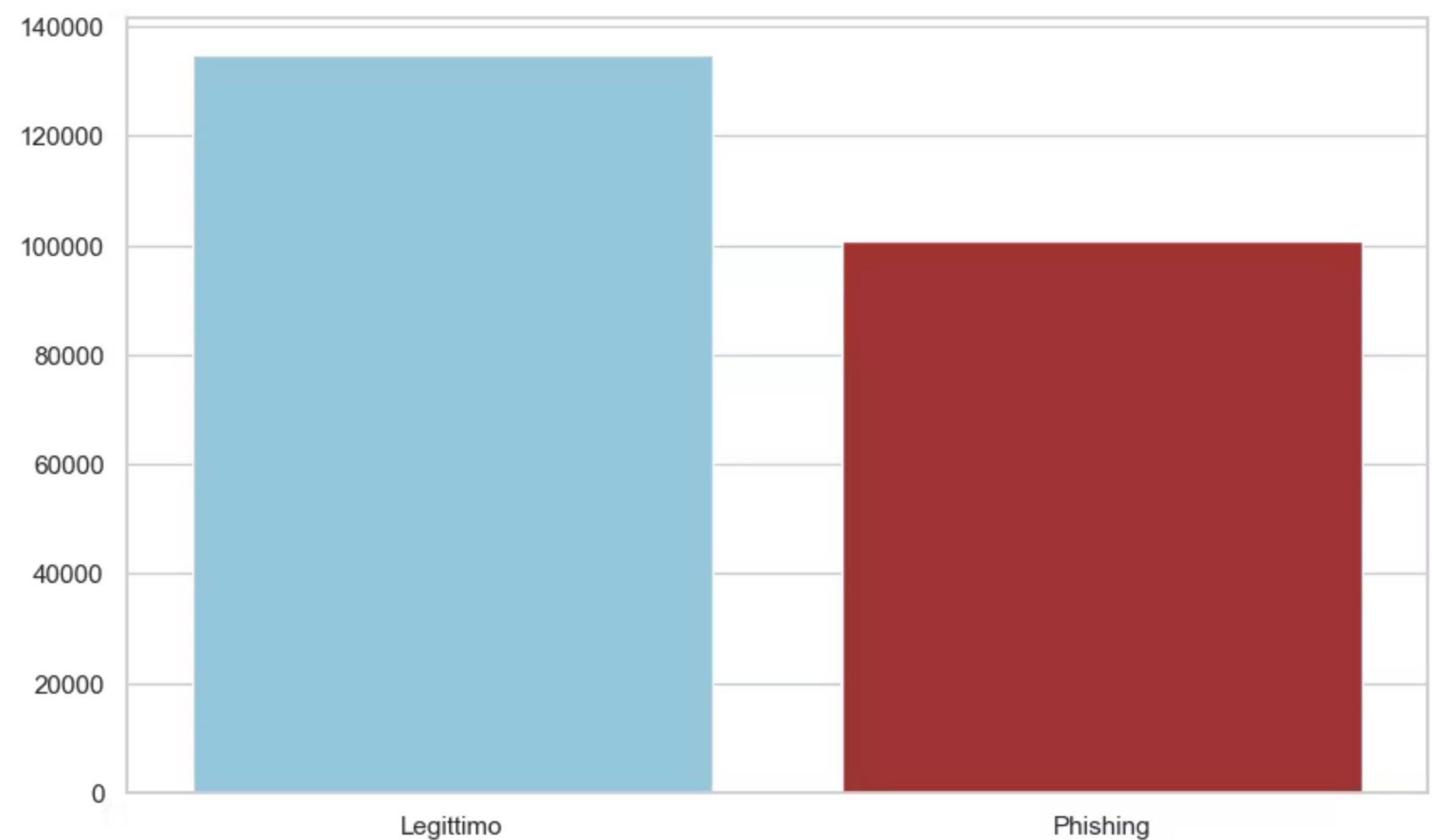
# PhiUSIIL Phishing URL (Website)

Ogni istanza del dataset corrisponde a un URL e alla relativa pagina web che è stata analizzata.

Nel complesso, il dataset include **134.850 URL**

**leggitti** e **100.945 URL malevoli**, offrendo un grosso volume per l'analisi comparativa (vengono distinti con valore della label a 0/1 in base alla loro natura).

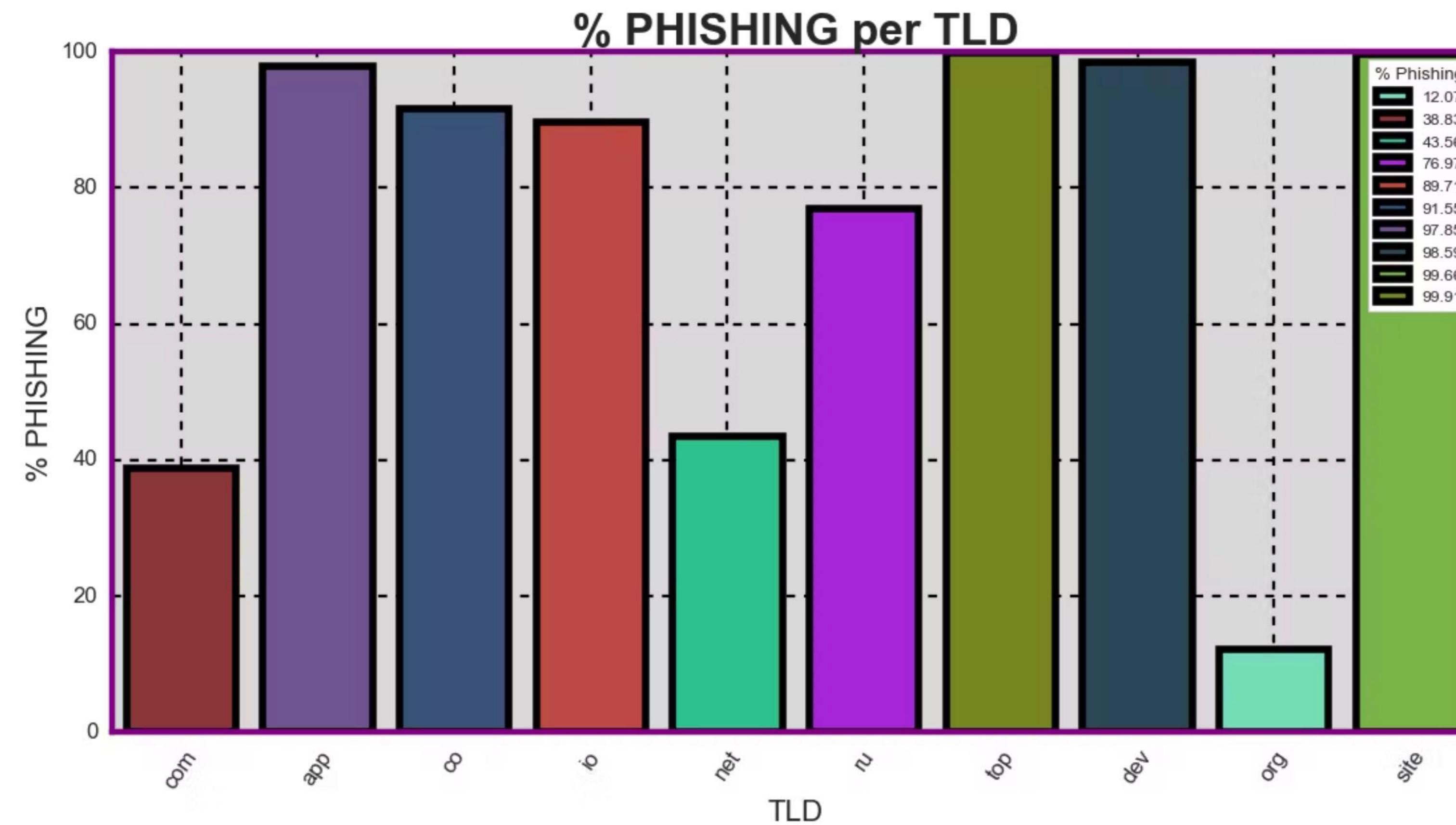
Le feature associate agli URL presentano valori di diversa natura (**reali, categorici e interi**) questo ci permette di effettuare una valutazione completa delle caratteristiche utili all'identificazione di un URL legittimo o meno.



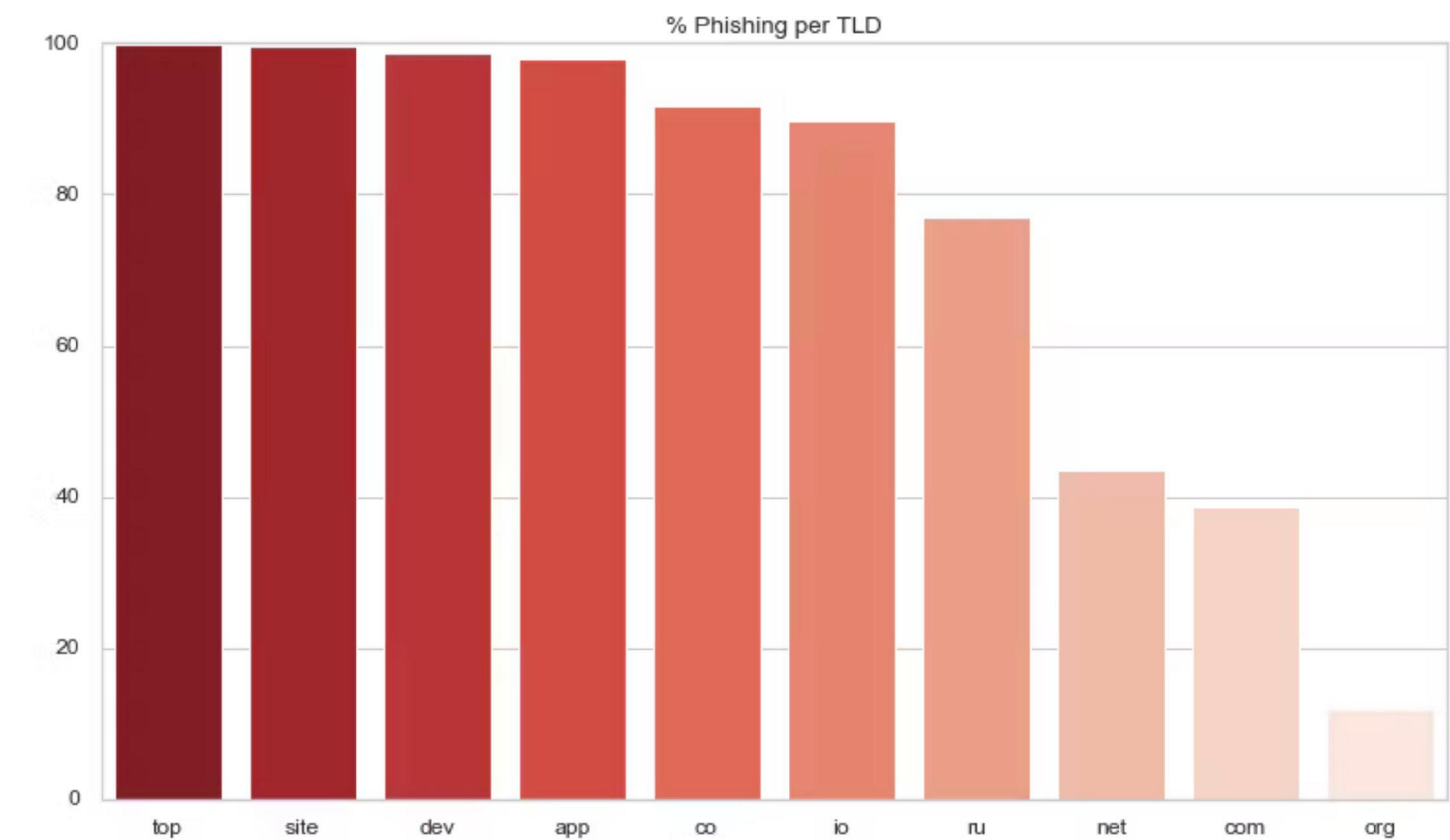
# 03

## Il data ink ratio in azione

Basso data-ink ratio



Data-ink ratio elevato



03

## Il Data-Ink Ratio

Il **Data-Ink Ratio** è un principio di visualizzazione introdotto da Edward Tufte che misura quanto dell'inchiostro usato in un grafico è effettivamente **dedicato ai dati**, e non a elementi decorativi.

Per Tufte un buon grafico deve:

- **Massimizzare l'inchiostro informativo** (data-ink);
- **Minimizzare** tutto il resto.

L'obiettivo è ottenere visualizzazioni **pulite, essenziali e ad alta leggibilità**, dove l'attenzione dell'utente è **concentrata sui contenuti** e non sul "rumore visivo".



## 03

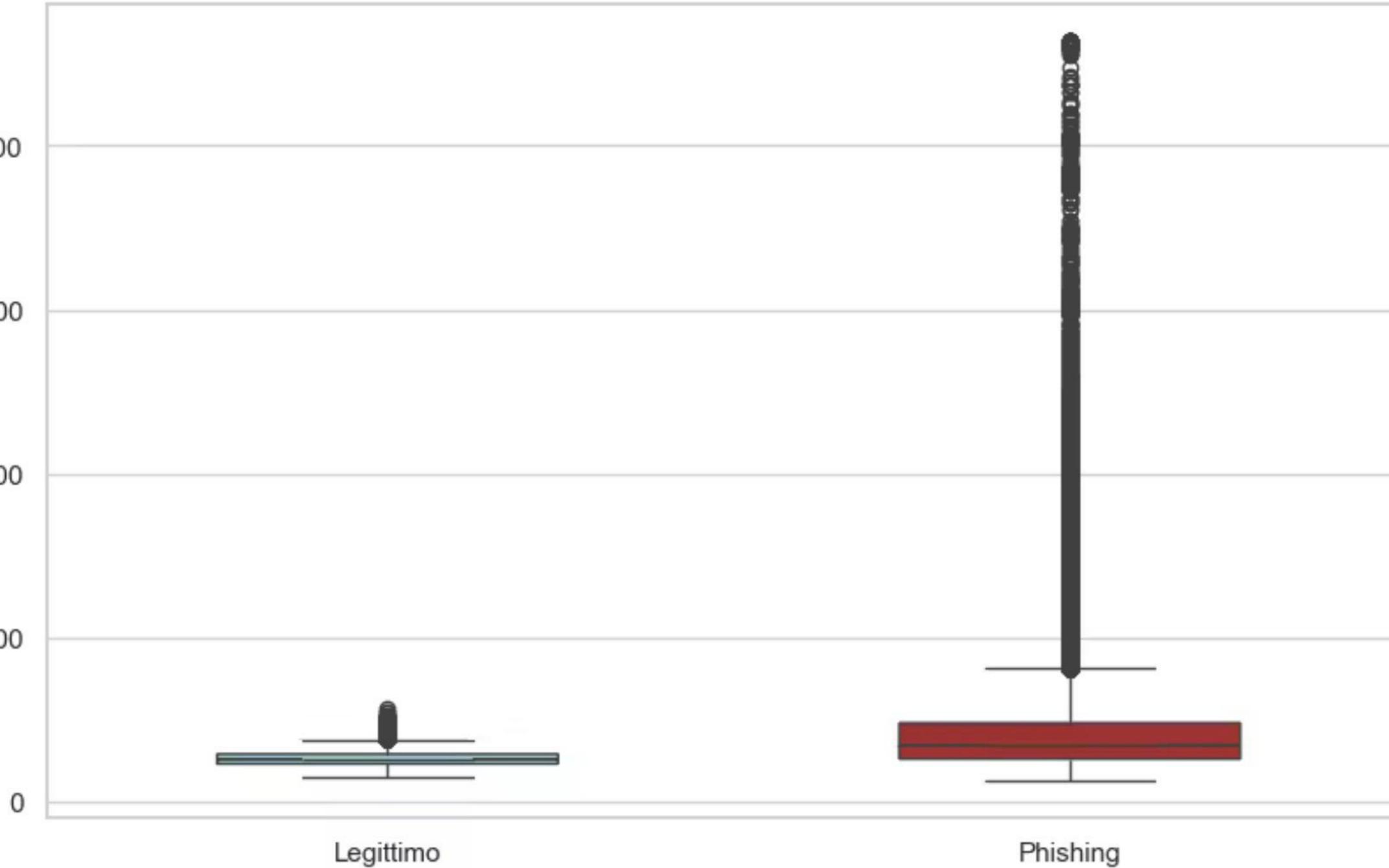
# Box plots URLLength vs Classe

```
q999 = data["URLLength"].quantile(0.999)  
data = data[data["URLLength"] <= q999]
```

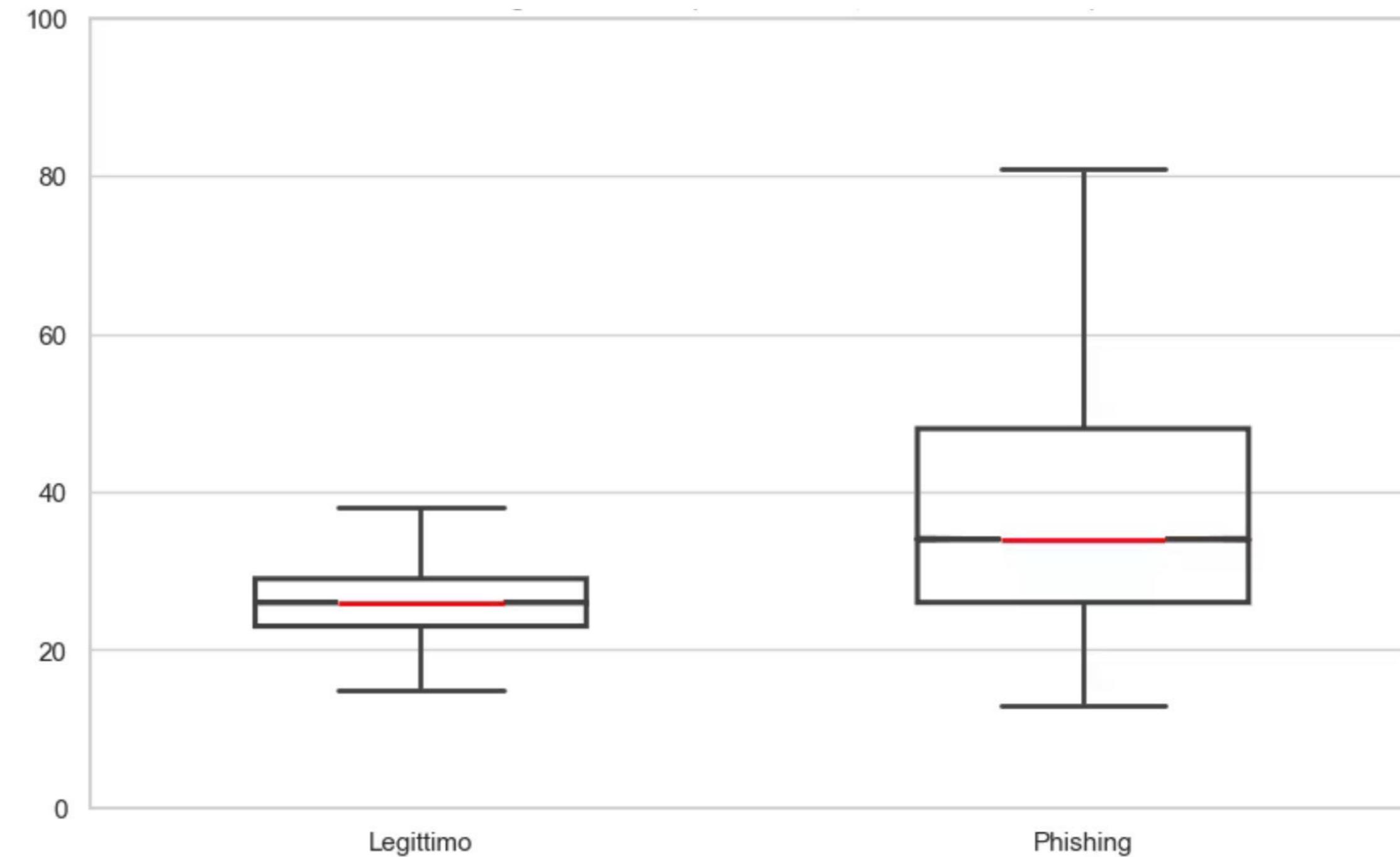
Sono stati costruiti **due box plot** per analizzare la distribuzione di **URLLength** nelle classi **phishing** e **legittimi**.

Prima della visualizzazione sono stati rimossi solo a scopo grafico gli **outlier estremi oltre il quantile q999 (99.9%)**, che includevano URL anche **> 3000 caratteri**, per migliorare la **leggibilità** dei grafici. Gli outlier sono comunque stati **mantenuti nell'addestramento del modello CART**.

**Visualizzazione completa**



**Zoom 0-100 con notch ben visibile**

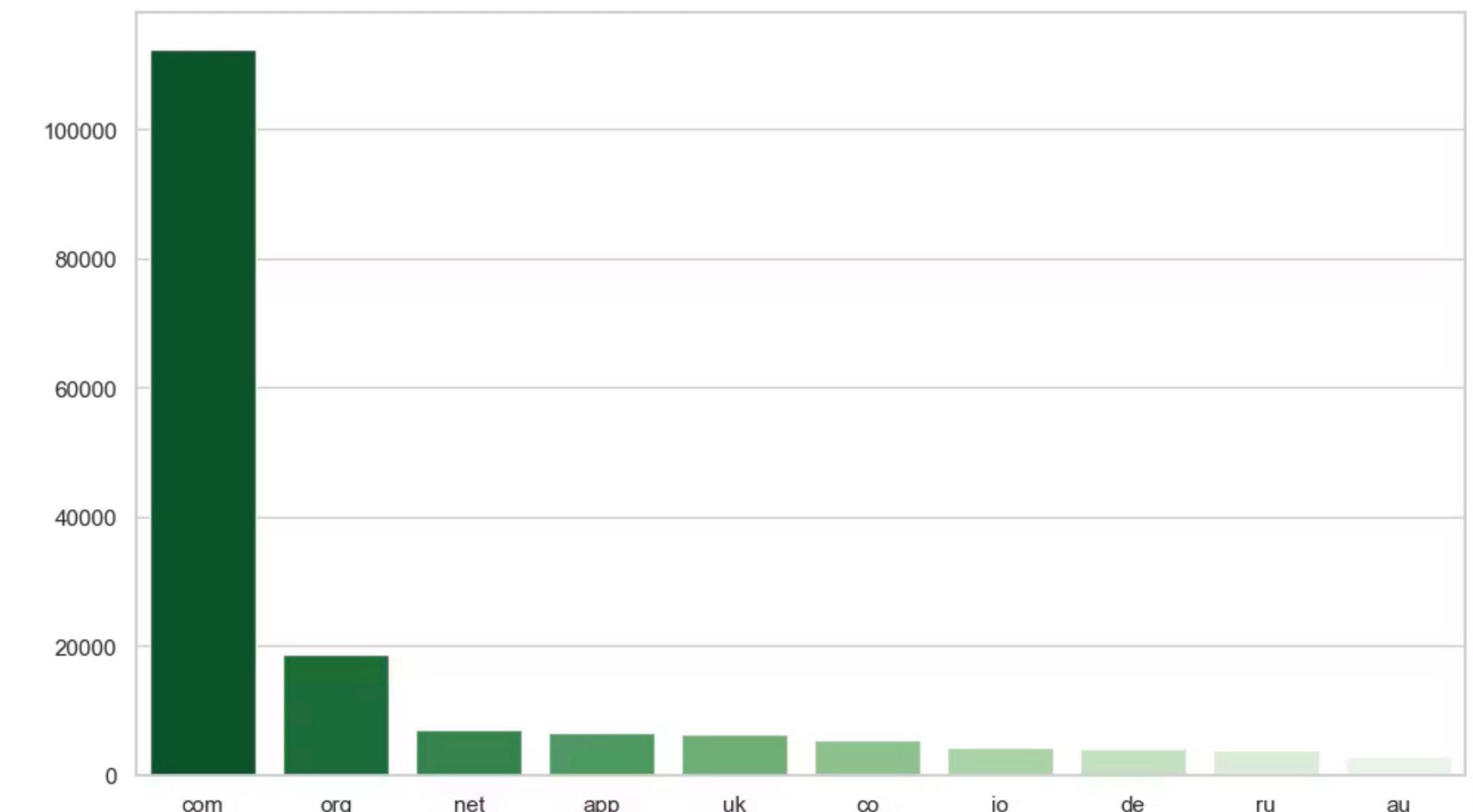


## 03

# Analisi dei TLD

Abbiamo analizzato la distribuzione dei TLD e le differenze tra URL legittimi e di phishing, evidenziando forti squilibri e pattern poco realistici. Alcuni TLD risultavano **sbilanciati** al 100% verso **una sola classe**, portando il CART a prendere decisioni basate quasi esclusivamente sull'estensione del dominio, **falsando il processo di apprendimento**. Per evitare un modello “che bara”, il TLD è stato quindi escluso dalle feature utilizzate per l’addestramento.

**TLD più frequenti nel dataset**

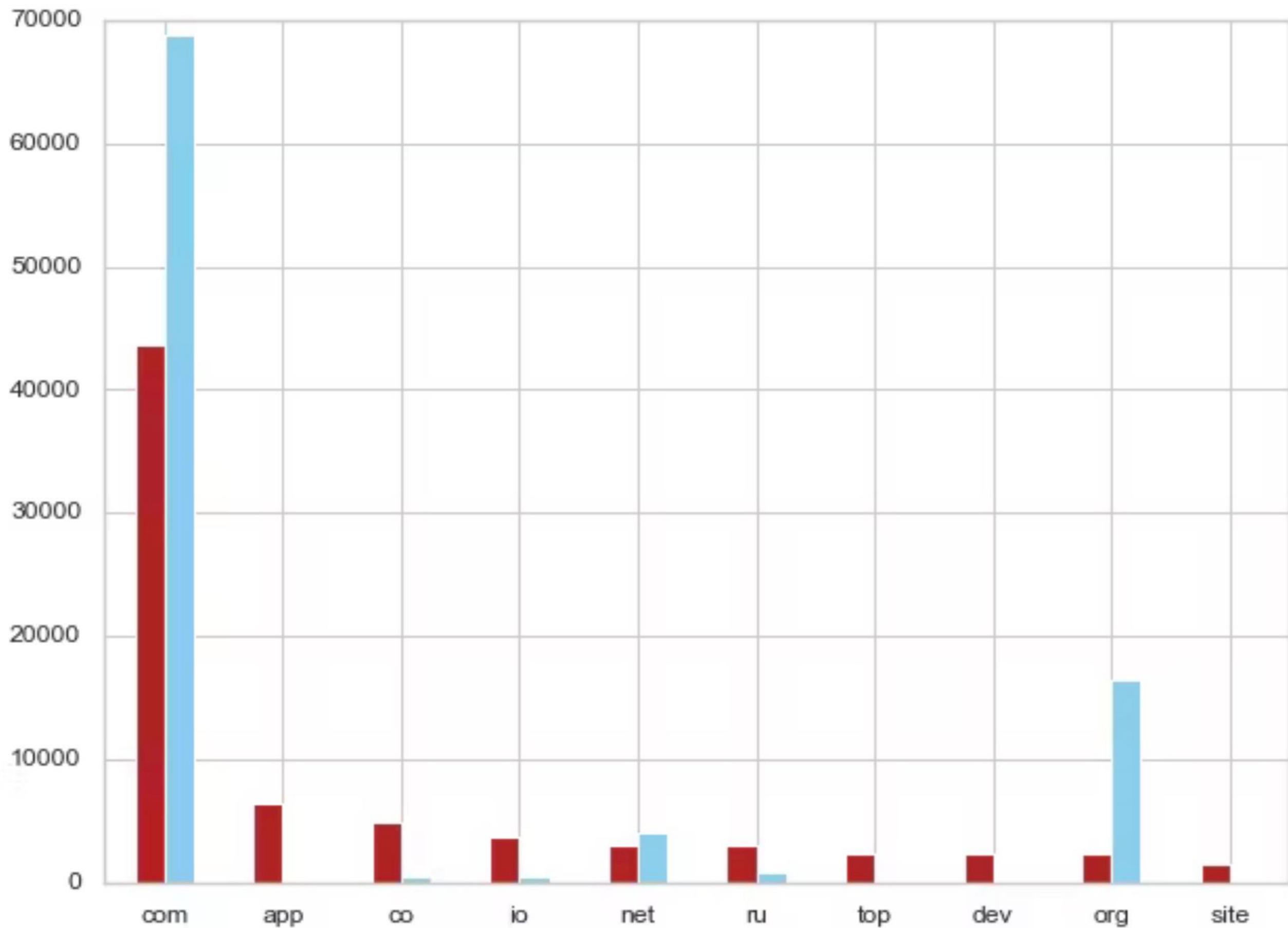


## 03

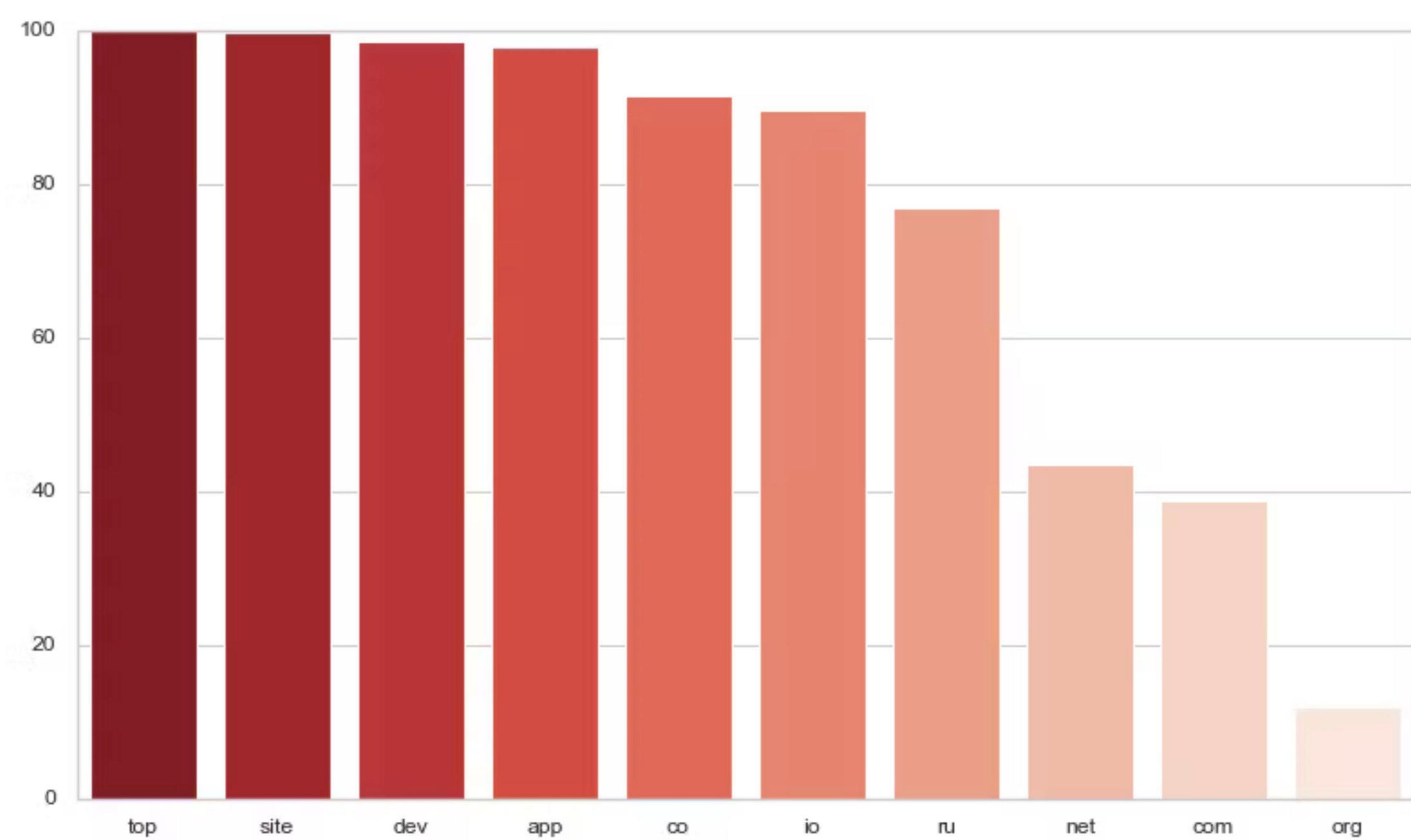
# Analisi dei TLD

Analisi della top 10 dei TLD con maggior numero di URL phishing

**Confronto tra URL legittimi e di phishing**

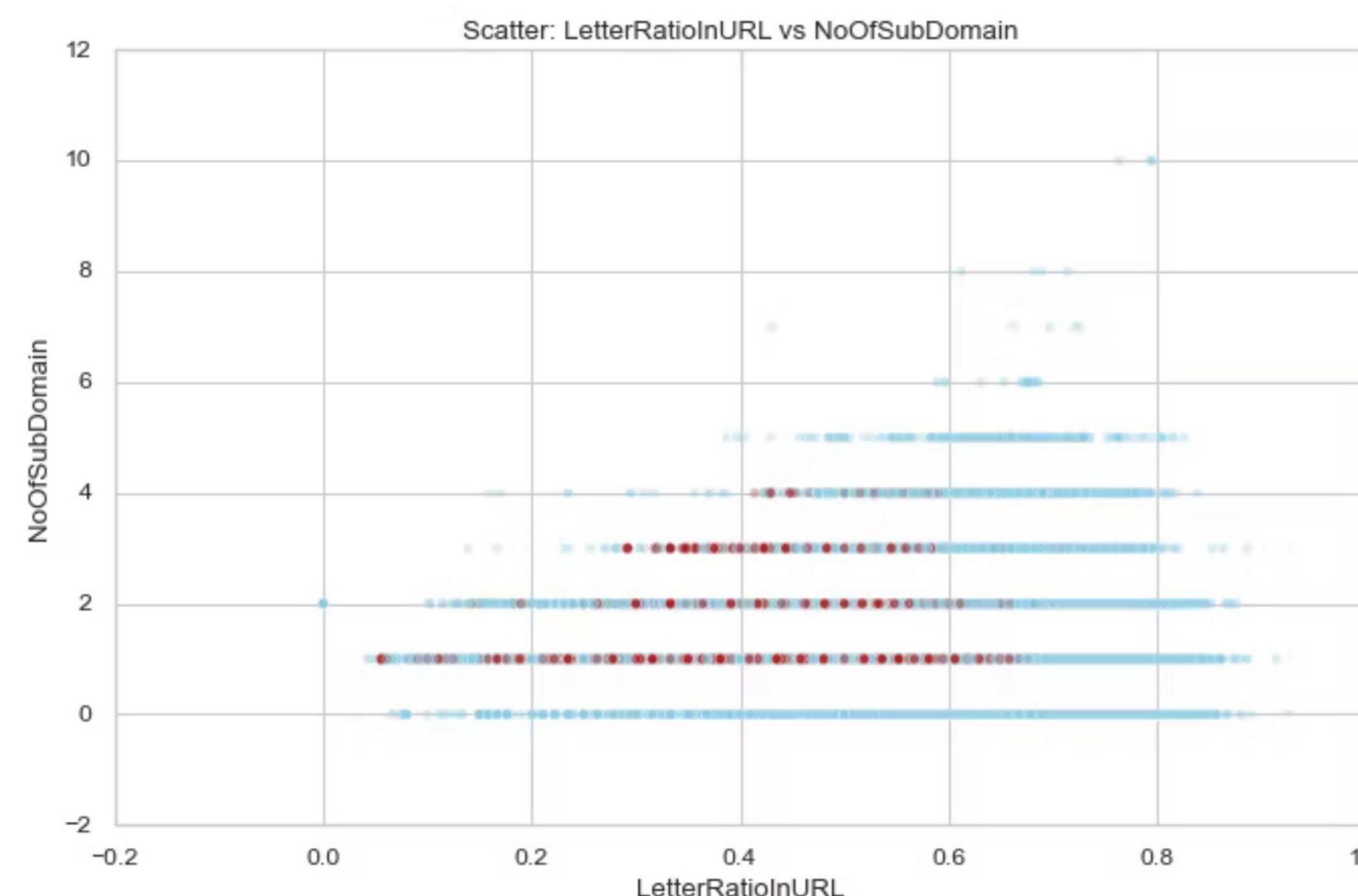


**Percentuale di phishing per TLD**



## 03

# Scatter plot LetterRatioInURL vs NoOfSubDomain



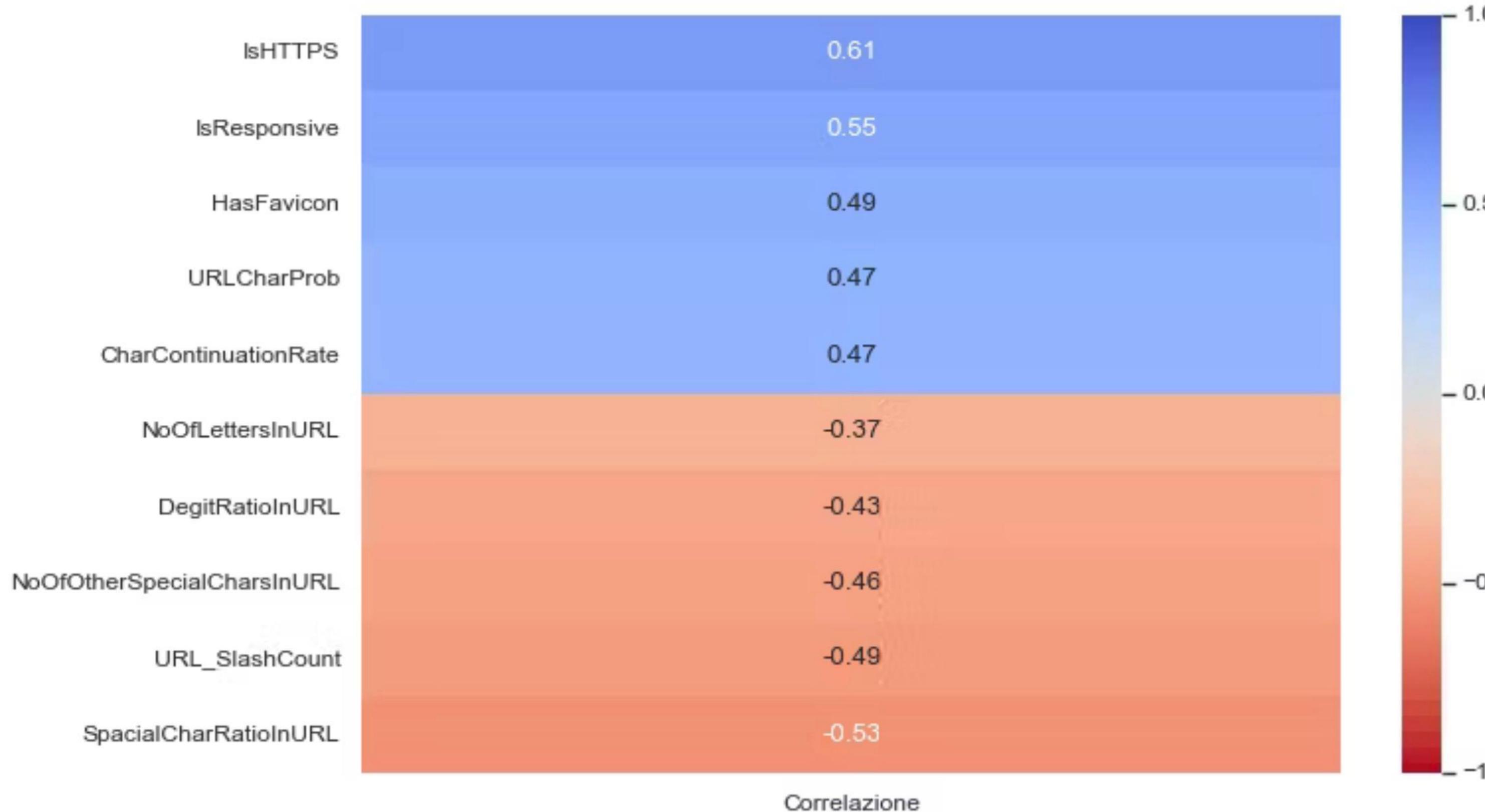
Il grafico mette in relazione la **percentuale di lettere** nell'URL (asse X) e il **numero di sottodomini** (asse Y), distinguendo tra URL legittimi e phishing.

Possiamo osservare che:

- I siti di phishing tendono ad avere pochi sottodomini, mostrando URL molto semplici.
- I siti legittimi spesso hanno più sottodomini, perché appartengono a infrastrutture complesse, questo rende **NoOfSubDomain** una feature molto **informativa**.
- La percentuale di lettere (**LetterRatioInURL**) **non separa** bene le due **classi**: phishing e legittimi sono **mescolati** lungo l'asse X. Per questo CART la utilizza solo se **combinata con altre feature**.

## 03

# Feature Importance correlata alla classe di output



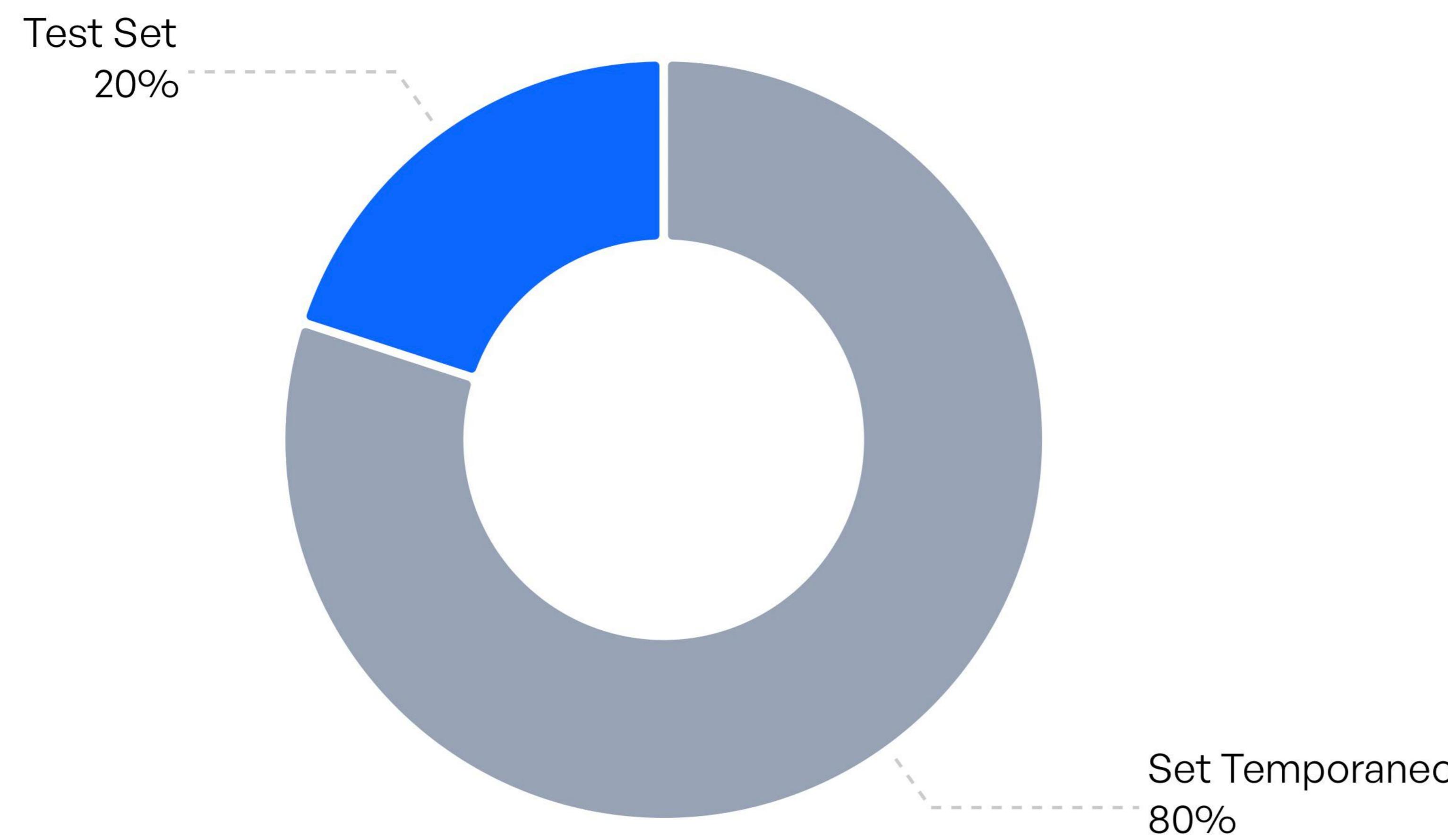
Questo grafico mostra le 5 feature con maggiore correlazione positiva e negativa rispetto alla classe di output.

- **Top 5 feature correlate con URL legittimi (correlazione positiva):** queste variabili descrivono comportamenti tipici dei siti legit, indicando un ambiente più stabile, curato e coerente con uno standard legitimo.
- **Top 5 feature correlate con URL di phishing (correlazione negativa):** queste feature sono fortemente indicative di contenuti sospetti, nonché segnali tipici di URL manipolati, più caotici e meno controllati.

## 04 Preparazione dei dati

Suddivisione iniziale in due sub-set

```
# 20% diventa di test
X_train_full, X_test, y_train_full, y_test = train_test_split(
    X, y, test_size=0.20, stratify=y, random_state=42
)
```



La divisione è avvenuta in **due passaggi**:

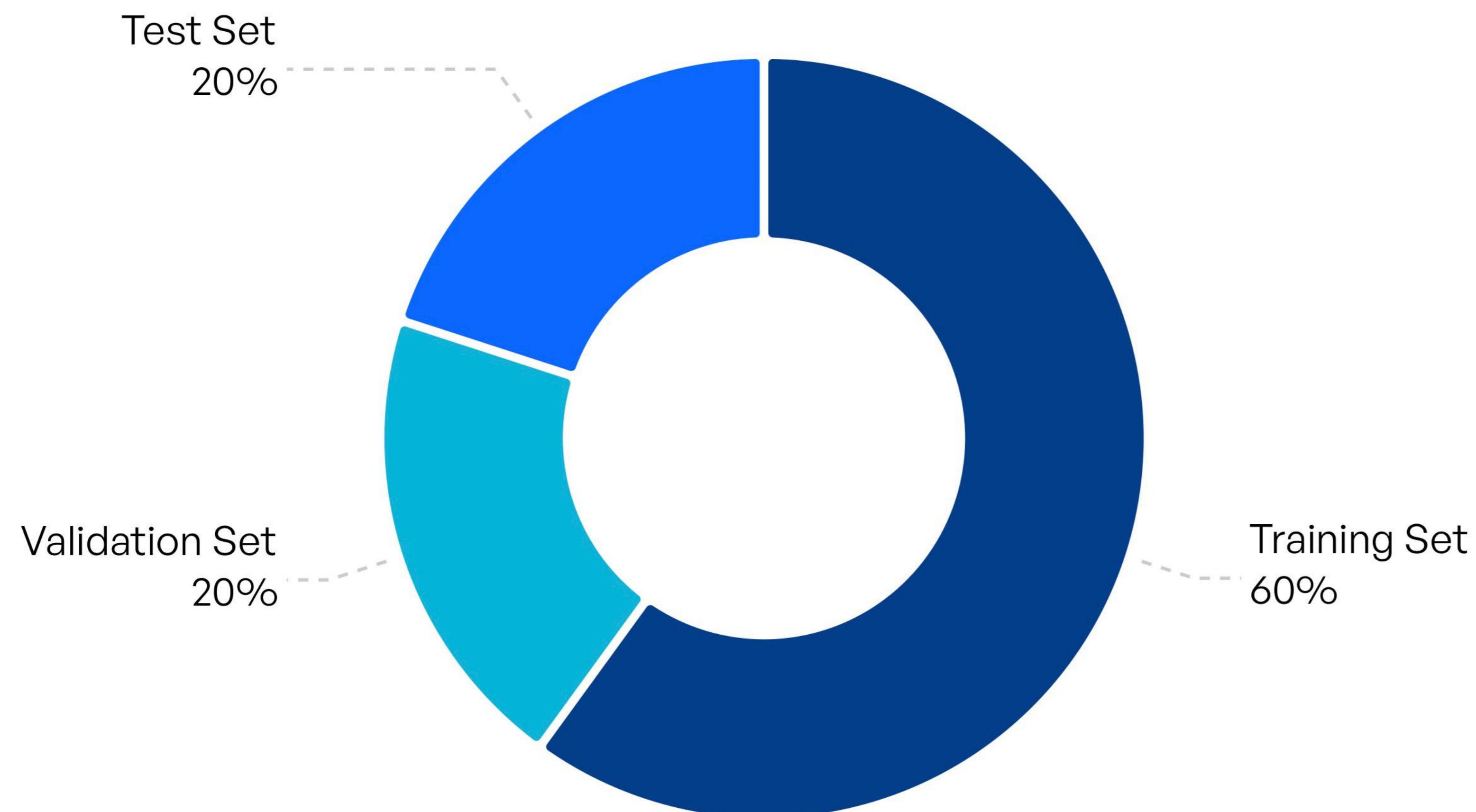
1. Abbiamo isolato il **20%** del totale come **test set**
2. Abbiamo diviso l'**80%** rimanente con una proporzione **75% / 25%** per ottenere il **60%** totale per il **training** e il **20%** totale per la **validation**.

Per garantire l'affidabilità del progetto abbiamo usato:

- **Stratificazione** (con **stratify=y**): mantiene la **stessa proporzione** di classi (legittimi/phishing) in tutti e tre i set per **evitare sbilanciamenti**.
- **Riproducibilità** (con **random\_state=42**): garantisce la **ripetibilità**.

# 04 Preparazione dei dati

Suddivisione in tre sub-set finali



```
# Dell'80% rimanente si effettua una divisione 75-25 per ottenere il 60% di training e il 20% di validation
X_train, X_val, y_train, y_val = train_test_split(
    X_train_full, y_train_full, test_size=0.25, stratify=y_train_full, random_state=42
)
```

- **60% - Training set**

Viene usato per addestrare il modello. È la parte principale del dataset e serve al CART per imparare le relazioni tra le feature (lunghezza URL, HTTPS, TLD, ecc) e la classe (legit oppure phishing). Qui il modello impara gli schemi e costruisce l'albero.

- **20% - Validation set**

Viene usato durante la fase di sviluppo per valutare il modello mentre viene costruito, aiuta a scegliere le features più significative e controlla se il modello sta lavorando bene o se sta andando in overfitting.

- **20% - Test set**

Viene usato nella parte finale per misurare le prestazioni reali del modello su dati mai visti e fornire le metriche di valutazione.

## 04

### Features rimosse

Abbiamo rimosso alcune feature **non essenziali** e potenzialmente **fuorvianti**, incluse quelle soggette a **leakage** o direttamente **collegate alla label**, per evitare che il modello CART possa **derivare il risultato** in modo artificiale anziché apprendere pattern reali.

URL	Domain	label
TLD	Title	HasDescription
DomainTitleMatchScore	URLTitleMatchScore	HasPasswordField
HasExternalFormSubmit	HasSubmitButton	HasHiddenFields
NoOfImage	HasSocialNet	HasCopyrightInfo
LineOfCode	LargestLineLength	NoOfImage
NoOfCSS	NoOfJS	NoOfSelfRef
NoOfEmptyRef	URLSimilarityIndex	

## 04

# Features principali selezionate

Abbiamo selezionato un insieme di **feature rilevanti** per la fase di classificazione, al fine di **migliorare l'efficacia** del modello, **ridurre la complessità** del dataset. Inoltre sono state create appositamente delle **nuove features** che tornano utili al CART per la **classificazione** degli URL.

```
# 4. CREAZIONE DI FEATURES

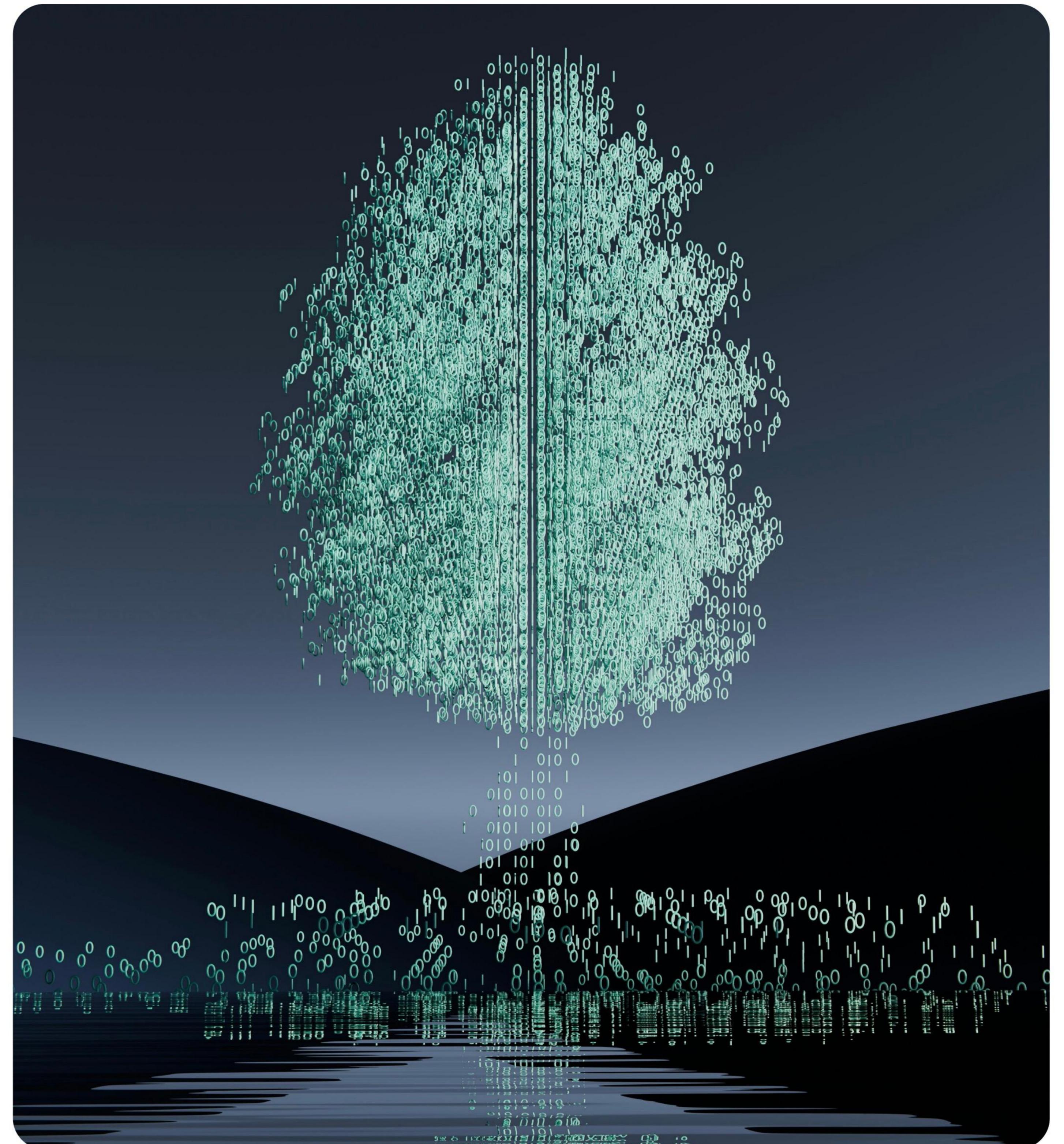
df = data.copy()

# Calcolo della lunghezza del dominio
df["DomainLength"] = df["Domain"].apply(lambda x: len(str(x)))
# Calcolo del numero di punti nell'url
df["URL_DotCount"] = df["URL"].apply(lambda x: str(x).count("."))
# Calcolo del numero di slash nell'url (profondità di esso)
df["URL_SlashCount"] = df["URL"].apply(lambda x: str(x).count("/"))
# Calcolo del rapporto tra lettere e numeri nell'url
df["LetterDigitRatio"] = df["NoOfLettersInURL"] / (df["NoOfDigitsInURL"] + 1)
```

URLLength	NoOfOtherSpecialCharsInURL	IsDomainIP
CharContinuationRate	TLDLegitimateProb	URLCharProb
TLDLength	NoOfSubDomain	HasObfuscation
NoOfObfuscatedChar	ObfuscationRatio	NoOfLettersInURL
LetterRatioInURL	NoOfDigitsInURL	DigitRatioInURL
NoOfEqualsInURL	NoOfQMarkInURL	NoOfAmpersandInURL
HasTitle	SpacialCharRatioInURL	IsHTTPS
NoOfURLRedirect	NoOfSelfRedirect	NoOfPopup
NoOfFrame	Bank	Pay
Crypto	IsResponsive	URL_DotCount
LetterDigitRatio	DomainLength	URL_SlashCount

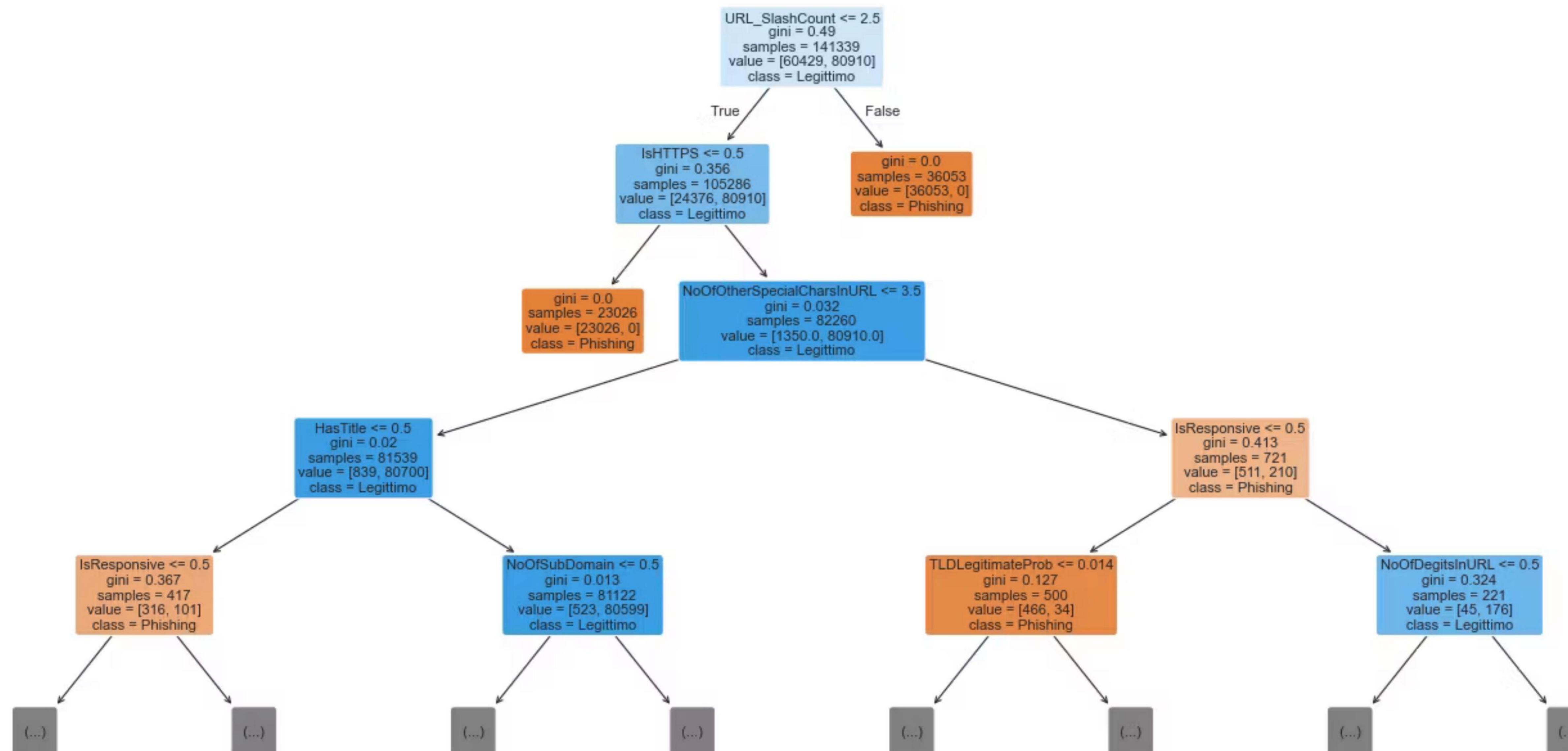
## Elaborazione del dataset via CART

L'algoritmo **CART** costruisce un **albero decisionale** che suddivide progressivamente il dataset in **gruppi** sempre più **omogenei**. A ogni split valuta le **feature** e seleziona quella che migliora maggiormente la **separazione** tra URL legittimi e malevoli, usando metriche come l'**indice di Gini**. Il risultato è una struttura che mette in evidenza le caratteristiche più rilevanti nella rilevazione di comportamenti sospetti.



# 06 L'albero decisionale CART

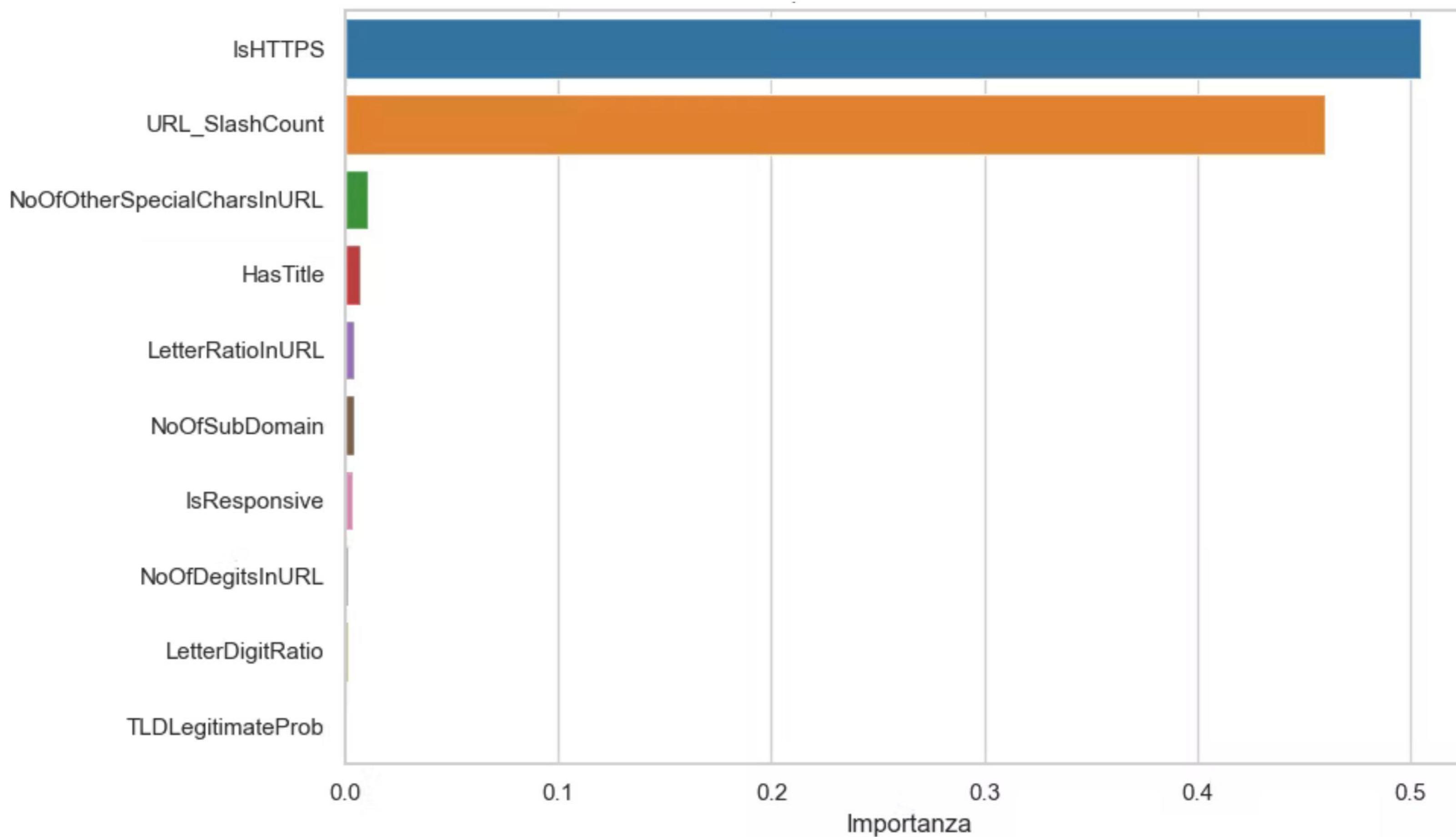
I primi 4 livelli del CART ottenuto attraverso l'algoritmo implementato



# 06 Feature importance

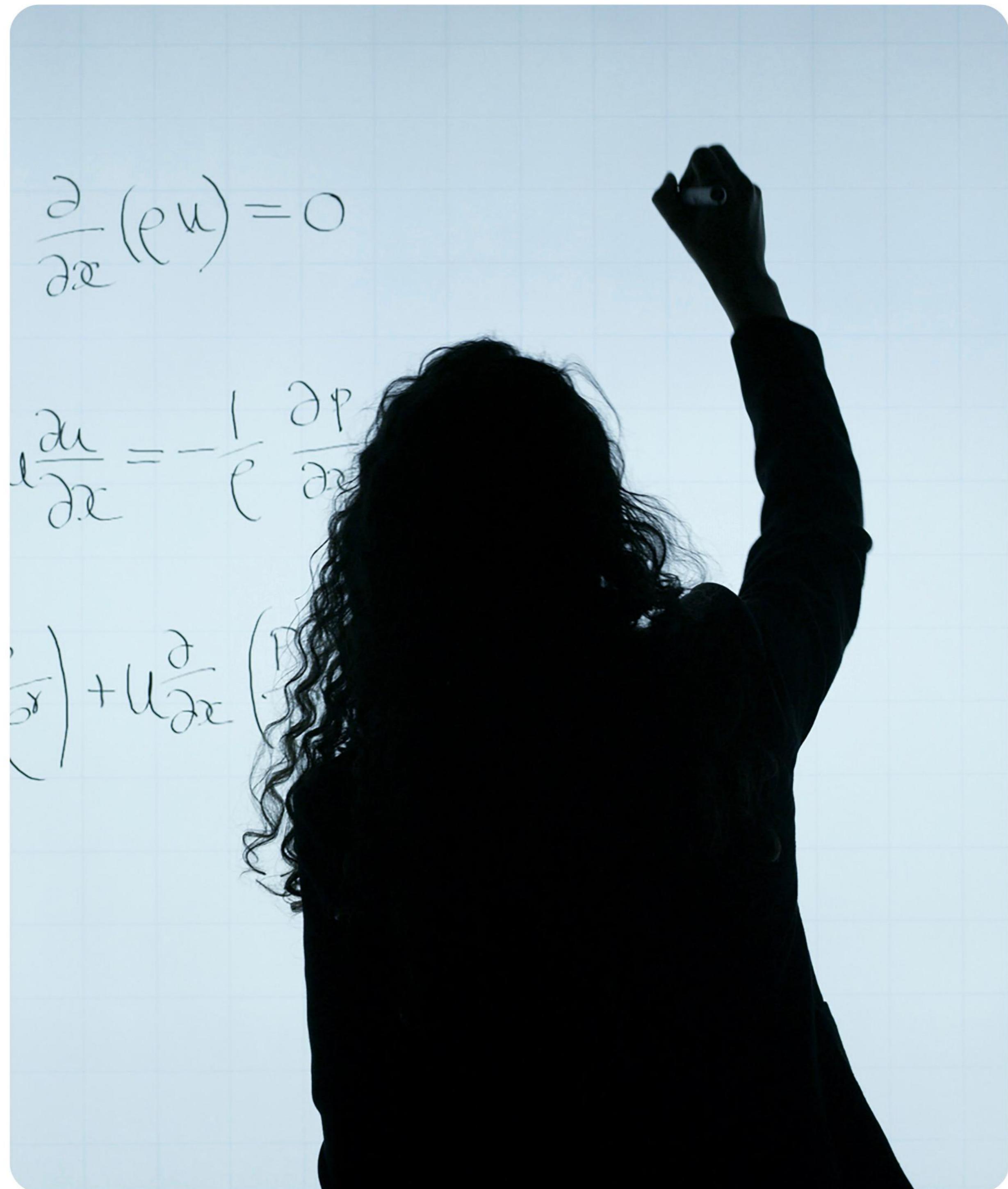
L'importanza delle features presenti nel dataset per l'albero decisionale

L'analisi della feature importance mostra che il modello si basa principalmente su **caratteristiche strutturali** dell'URL, come l'utilizzo di **HTTPS** e il numero di **slash**, mentre le altre feature contribuiscono in misura molto più ridotta. Questo conferma che il CART individua **pattern semplici ma altamente discriminanti**, privilegiando segnali immediatamente legati al comportamento tipico degli URL di phishing.



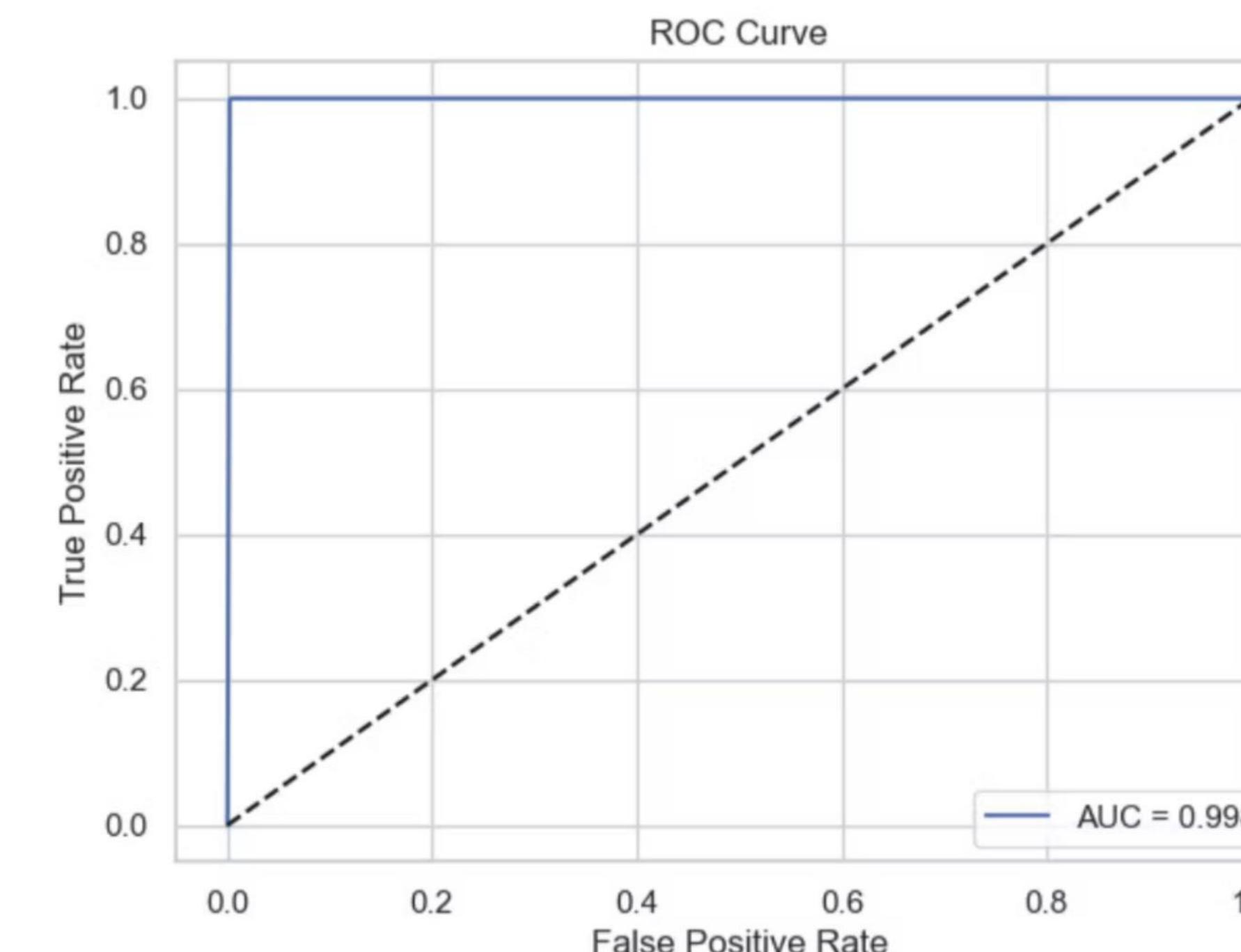
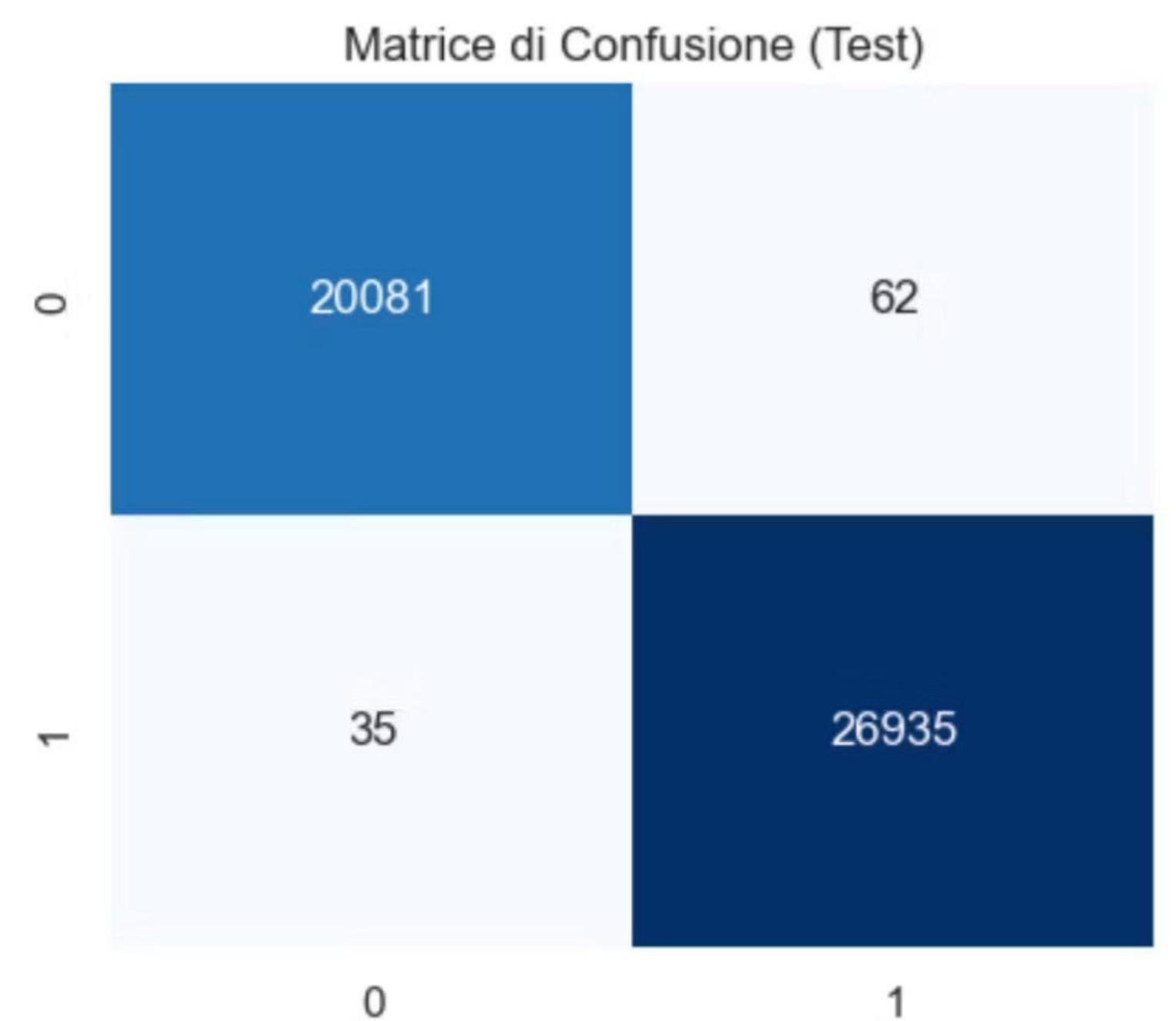
# Metriche di valutazione

- **Accuracy:** rappresenta la **percentuale complessiva di previsioni corrette** e fornisce una prima indicazione delle performance del modello.
- **Precision:** misura l'**affidabilità delle previsioni di phishing**, indicando quanti dei casi segnalati come phishing lo sono realmente.
- **Recall:** valuta la **capacità** del modello **di identificare tutti i veri casi di phishing**, riducendo il rischio di mancarne alcuni.
- **F1-Score:** combina **precision e recall** in un'unica metrica bilanciata.
- **Confusion Matrix:** offre una **visione dettagliata delle classi correttamente predette e degli errori** (TP, TN, FP, FN), permettendo un'analisi più completa del comportamento del modello.



# 07 Valutazione

Analisi dell'accuratezza del modello mediante matrice di confusione, curva ROC e classification report



Accuracy VALIDATION: 0.9982382781822426  
Accuracy TEST: 0.9979411202852716

Classification Report (TEST):				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	20143
1	1.00	1.00	1.00	26970
accuracy			1.00	47113
macro avg	1.00	1.00	1.00	47113
weighted avg	1.00	1.00	1.00	47113

# Fonti

Per questo progetto abbiamo fatto riferimento a un dataset esterno e articoli accademici relativi alle tematiche trattate.

**Prasad, A., & Chandra, S. (2023).  
PhiUSIIL: A diverse security  
profile empowered phishing URL  
detection framework based on  
similarity index and incremental  
learning. Computers & Security,  
103545. doi:  
[https://doi.org/10.1016/j.cose.2023.  
103545](https://doi.org/10.1016/j.cose.2023.103545)**

Dataset consultato

<HTTPS://TINYURL.COM/4BDAFPRK>

**McGurgan, K., Fedoroksay, E.,  
Sutton, T. M., & Herbert, A. M.  
(2021). Graph Design: The Data-ink  
Ratio and Expert Users.  
In VISIGRAPP (3: IVAPP) (pp. 188-  
194).**

Nozioni ed esperimenti su data-ink ratio

<HTTPS://TINYURL.COM/Y76VVAZM>

**A. Karim, M. Shahroz, K. Mustofa,  
S. B. Belhaouari and S. R. K. Joga,  
"Phishing Detection System  
Through Hybrid Machine Learning  
Based on URL," in IEEE Access,  
vol. 11, pp. 36805-36822, 2023, doi:  
<10.1109/ACCESS.2023.3252366>**

Approccio ibrido per il rilevamento di URL  
di phishing

<HTTPS://TINYURL.COM/39WNFAT4>