
ANALISI DATI PHISHING

**Applicazione di metodologie statistiche e di
machine learning per lo studio degli URL
legati al phishing**

Giuseppe Napolitano Nicola Pagliara
Matricole: **0522501961 0512201413**

Contents

1	Dataset originario	5
1.1	Introduzione e descrizione	5
1.2	Analisi descrittiva dataset originario	6
1.2.1	Boxplot	7
1.2.2	Boxplot ad intaglio	7
1.2.3	Iistogrammi	8
1.2.4	Diagrammi di Pareto	9
1.2.5	QQPlot	9
1.3	Selezione delle caratteristiche	10
1.3.1	Eliminazione delle feature derivate	10
1.3.2	Filtraggio per varianza	10
1.3.3	Filtraggio per Correlazione	10
1.3.4	Filtraggio tramite interpretabilità	12
2	Analisi del Dataset Filtrato	18
3	Rappresentazione grafica	19
3.1	Boxplot	19
3.2	Boxplot ad intaglio	19
3.3	Iistogrammi	20
3.4	Diagramma di Pareto	20
3.5	Q-QPlot	21
4	Studio delle distribuzioni	21
4.1	Media, moda e mediana	22
4.2	Varianza e deviazione standard	23
4.2.1	Coefficiente di Variazione	23
4.3	Kurtosis e Skewness	24
5	Clustering	25
5.1	K-Nearest Neighbour	25
5.2	Addestramento del Modello	26
5.3	Risultati degli Esperimenti	26
6	Test del Chi quadrato bilaterale	28
6.1	Cos'è il test del chi quadrato	28
6.2	Risultati del Dataset filtrato	29
7	Modello linguistico	31
7.1	LLM Research Question	31
7.2	Generazione del dataset sintetico tramite LLM	31
7.2.1	Analisi del dataset sintetico con primo prompt	32
7.2.2	Statistica descrittiva univariata dataset sintetico Prompt1	32
7.2.3	BoxPlot Dataset Sintetico Prompt1	32
7.2.4	BoxPlot ad intaglio Dataset Sintetico Prompt1	33

7.2.5	Iistogrammi Dataset Sintetico Prompt1	33
7.2.6	Diagrammi di Pareto Dataset sintetico prompt1	34
7.2.7	QQPlot	34
7.2.8	Model Performance Dataset sintetico prompt1	34
7.2.9	KNN	34
7.2.10	Analisi distribuzione Dataset sintetico prompt1	35
7.2.11	Media, moda, mediana	35
7.2.12	Coeff. di Variazioni del Dataset Sintetico del primo prompt	36
7.2.13	Skewness e Kurtosis del Dataset Sintetico del primo prompt	38
7.2.14	Test del Chi Quadrato Dataset Prompt1	39
7.2.15	Test della normale su Dataset sintetico prompt1	39
7.2.16	Test binomiale su Dataset sintetico prompt1	39
7.2.17	Test Poisson su Dataset sintetico prompt1	40
7.3	Analisi del dataset sintetico finale	41
7.3.1	Statistica descrittiva univariata per Dataset sintetico finale	41
7.3.2	BoxPlot	41
7.3.3	Boxplot ad Intaglio	41
7.3.4	Iistogrammi	42
7.3.5	Diagrammi di Pareto	42
7.3.6	QQPlot	43
7.3.7	Model Performance Dataset Sintetico Finale	43
7.3.8	Analisi distribuzione Dataset sintetico finale	44
7.3.9	Media, moda, mediana	45
7.3.10	Coeff. di Variazioni del Dataset Sintetico finale	46
7.3.11	Skewness e Kurtosis del Dataset Sintetico completo	46
7.3.12	Test del Chi Quadrato Dataset Sintetico Completo	47
7.3.13	Test della normale	47
7.3.14	Test binomiale	48
7.3.15	Test di Poisson	49
8	Conclusioni e sviluppi futuri	49
1	Librerie R utilizzate	51

Introduzione

Negli ultimi anni le tecnologie digitali hanno subito una rapida evoluzione, e con essa sono aumentate anche le minacce informatiche, in particolare gli attacchi di Phishing. Il Phishing è una particolare tipologia di truffa realizzata sulla rete che sfrutta collegamenti (link) che rimandano a siti web fasulli che ingannano l'utente, rubando i suoi dati personali. L'obiettivo principale è quello di distinguere un sito reale da uno fasullo, rendendo la navigazione sul web sicura per gli utenti. I criminali informatici spesso utilizzano tecniche di **spoofing** creando un sottodomini che assomiglia a un sito web legittimo per ingannare gli utenti. **Top-Level Domains**(TLDs), gli ultimi segmenti di un dominio, sono ampiamente utilizzati dai malintenzionati per creare un domini con TLD simili ad estensioni note e affidabili. La crescita esponenziale degli attacchi di phishing tramite URL è diventata sempre più evidente, ed è per questo che il gruppo di Lavoro Anti-Phishing ha raccolto negli ultimi quindici anni un elevato numero di siti phishing. Lo scoppio della pandemia COVID-19 ha portato a un enorme aumento degli attacchi con URL di phishing, poiché i criminali informatici sfruttano la paura, l'urgenza medica e l'aumento dell'attività online da parte degli utenti come punti di forza per i loro attacchi. Il Machine Learning è risultato uno strumento fondamentale contro i vari cyber-attacchi, poiché ha l'abilità di analizzare una grande quantità di dati e di identificare tramite pattern gli URL illegittimi. Bisogna, però, aggiornare e migliorare costantemente il modello poiché gli attacchi di phishing si evolono e cambiano col passare del tempo, ed un modello obsoleto non riuscirebbe a scovare tutti gli url illegittimi. L'approccio proposto per il rilevamento degli URL phishing si basa sull'apprendimento incrementale. L'apprendimento incrementale è una tecnica di machine learning in cui un modello può apprendere continuamente da nuovi dati, riducendo la necessità di un riaddestramento completo e minimizzando l'interruzione causata da quest'ultimo. L'obiettivo dell'articolo di ricerca mira a sviluppare, implementare e valutare un framework scalabile e adattabile per il rilevamento di URL di phishing, che sfrutta l'indice di similarità e le tecniche di apprendimento incrementale per contrastare efficacemente gli attacchi di phishing. Per la costruzione del dataset sono state raccolte informazioni da **phishtank.com** e informazioni DNS da **WHOIS**. Negli ultimi anni sono stati utilizzate diverse metodologie per scovare gli URL malevoli, ed i ricercatori hanno utilizzato cinque modelli di machine learning, ovvero **Random Forest** (RF), **Decision Tree** (DT), **LightGBM**, **Logistic Regression** (LR) e **Support Vector Machine** (SVM) per costruire un modello di machine learning che fosse all'avanguardia. Nel loro esperimento, basato sull'accuratezza della classificazione e sull'analisi della curva di validazione, **LightGBM** ha superato gli altri modelli.

1 Dataset originario

1.1 Introduzione e descrizione

La versione originaria del dataset utilizzato per le analisi dei successivi capitoli di questo elaborato è stato costruito utilizzando una combinazione di URL legittimi ed URL di Phishing. I dati sono stati raccolti in questo modo:

- **URL legittimi:** ottenuti dall'Open PageRank Initiative, una piattaforma che fornisce elenchi di siti web legittimi ed affidabili
- **URL Phishing:** ottenuti da database di sicurezza informatica quali PhishTank, OpenPhish e MalwareWorld.

Per ogni osservazione sono state estratte delle variabili significative, fondamentali per l'individuazione di siti di phishing. Per ogni osservazione sono stati individuati i seguenti dati:

- **Caratteristiche estratte dagli URL**
 - **TLD** (Top Level Domain): Specifica l'estensione del dominio, ad esempio .com, .org, .net.
 - **URLLength:** Lunghezza complessiva dell'URL, espressa in numero di caratteri. URL più lunghi sono statisticamente associati a phishing.
 - **IsDomainIP:** Variabile binaria che indica se il dominio è espresso come indirizzo IP anziché come nome di dominio
 - **NoOfSubDomain:** Conta il numero di sottodomini
 - **NoOfObfuscatedChar:** Numero di caratteri offuscati o codificati
 - **IsHTTPS:** Variabile binaria che segnala l'utilizzo del protocollo HTTPS
 - **No. of digits, equal, qmark, amp:** un elevato numero di '=' , '?' può significare che l'url sia phishing.
- **Caratteristiche HTML**
 - **LargestLineLength:** Lunghezza massima di una riga nel codice HTML
 - **HasTitle:** Variabile binaria che indica la presenza di un titolo nella pagina HTML
 - **HasFavicon:** Variabile binaria che segnala la presenza di un'icona di favicon
 - **IsResponsive:** Variabile binaria che valuta la responsività del design della pagina web
 - **NoOfURLRedirect:** Conta il numero di reindirizzamenti presenti nel codice HTML
 - **HasDescription:** Indica la presenza di una descrizione meta
 - **NoOfPopup eNoOfFrame:** Numero di finestre pop-up o frame incorporati nella pagina
 - **HasExternal FormSubmit:** Variabile binaria che indica se i form HTML inviano dati a domini esterni

- **HasCopyrightInfo** e **HasSocialNet**: Variabile che indica se il sito presenta informazioni di copyright o social network.
- **HasPasswordField** e **HasSubmitButton**: Variabile binaria che segnala la presenza di campi password nel HTML
- **HasHiddenFields**: Variabile che indica la presenza di campi nascosti che possono rubare dati sensibili all'oscuro dell'utente.
- **Bank, Pay e Crypto**: Indicatori della presenza di termini finanziari o relativi a criptovalute
- **NoOfImage**: Numero totale di immagini nella pagina
- **NoOfJS**: Numero di script JavaScript inclusi. Un eccessivo utilizzo può indicare comportamenti sospetti
- **NoOfSelfRef**, **NoOfEmptyRef** e **NoOfExternalRef**: variabile che conta i riferimenti ipertestuali interni, vuoti o esterni

- **Feature Derivate**

- **CharContinuationRate**: variabile che misura la presenza di sequenze omogenee di caratteri alfabetici, numerici o simbolici
- **URLTitleMatchScore**: Variabile che valuta la corrispondenza tra l'URL ed il titolo della pagina.
- **URLCharProb**: Probabilità cumulativa basata sulla frequenza di caratteri alfabetici e numerici in URL legittimi rispetto a quelli Phishing. La probabilità è calcolata secondo questa formula:

$$\text{URLCharProb} = \sum_{i=0}^n \frac{\text{prob}(\text{URL}[\text{char}_t])}{n}$$

- **TLDLegitimateProb**: Calcolo della probabilità che un determinato TLD sia associato a siti web legittimi

- **Colonna Label**: Variabile target binaria che distingue gli URL legittimi (1) da quelli Phishing (0).

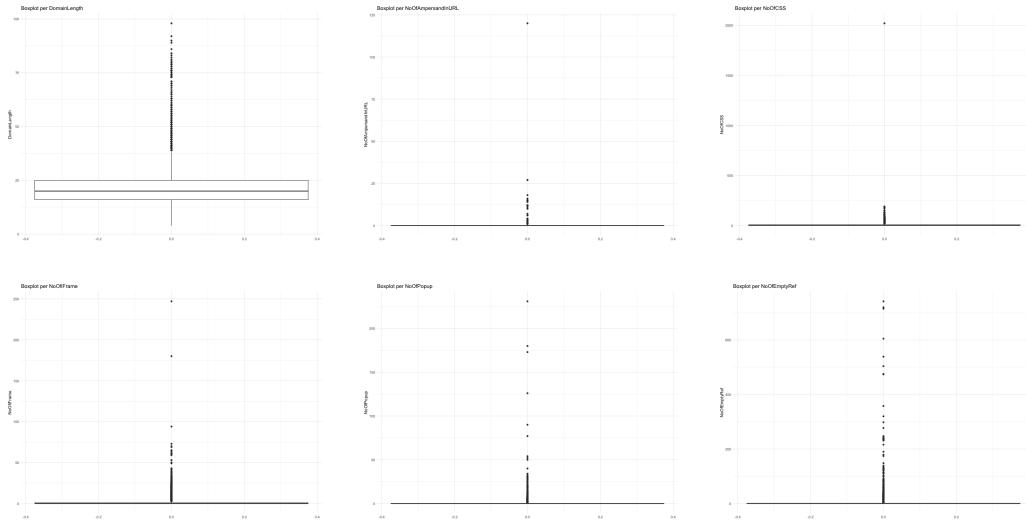
Effettuando una lettura approfondita del paper e visualizzando il dataset a disposizione, siamo giunti alla conclusione che alcune delle colonne non erano state riportate nella documentazione. Il dataset presenta complessivamente un numero di **235.795** osservazioni e **56** colonne.

1.2 Analisi descrittiva dataset originario

Come primo passo andremo ad effettuare l'analisi descrittiva sul dataset iniziale descritto nel capitolo precedente.

1.2.1 Boxplot

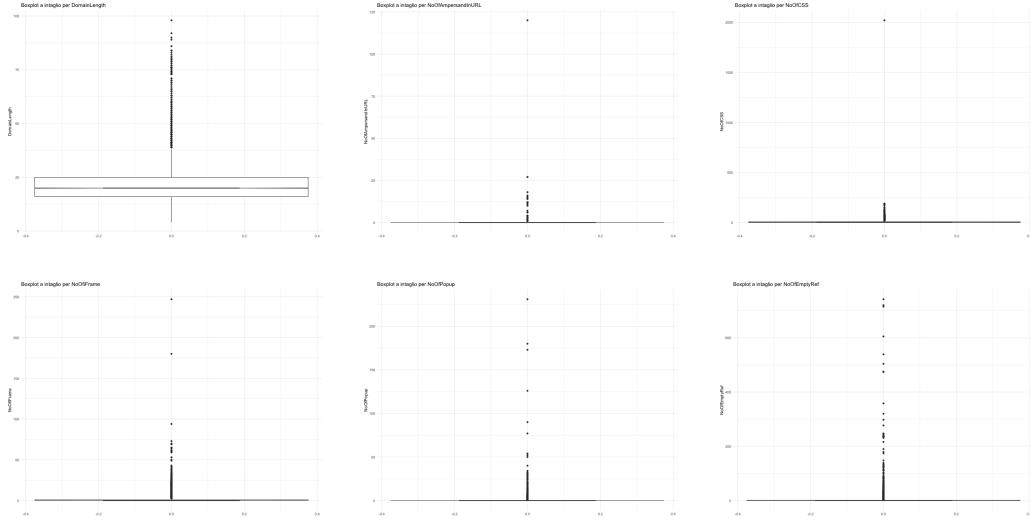
Per prima cosa mostreremo i boxplot ricavati; quest'ultimi rappresentano i dati compresi tra il primo quartile (**Q1**) e il terzo quartile (**Q3**), con al centro il secondo quartile (**Q2**) che divide perfettamente a metà. L'estremo del "baffo" inferiore rappresenta il valore più piccolo tra le osservazioni, mentre l'estremo superiore rappresenta il valore più grande. Qui sotto riportiamo i valori ricavati:



Da questi grafici possiamo notare che la distribuzione della lunghezza del dominio è fortemente asimmetrica, indicando la presenza di numerosi outlier, dimostrando che i dati potrebbero non essere distribuiti in un modo normale.

1.2.2 Boxplot ad intaglio

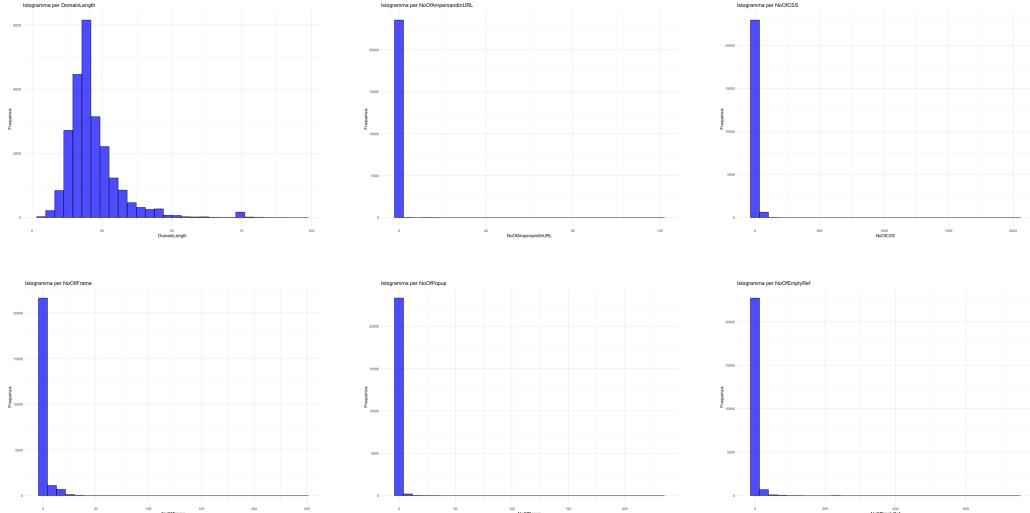
Il secondo approccio utilizzato è stato quello dell'utilizzo dei boxplot ad intaglio; quest'ultimi sono utilizzati per visualizzare la distribuzione dei dati, ma forniscono un'informazione aggiuntiva sulla significatività statistica delle differenze tra mediane di gruppi. Le tacche si restringono attorno alla mediana, e se quest'ultime non si sovrappongono vuol dire che c'è una forte evidenza che le mediane dei gruppi sono significativamente differenti.



Il range interquartile nei grafici qui sopra raffigurati è relativamente stretto, dimostrando che la maggior parte dei domini ha una lunghezza compresa tra il primo ed il terzo quartile. Gli outlier sono numerosi e molto distanti dal valore mediano, indicando lunghezze eccessivamente elevate.

1.2.3 Istogrammi

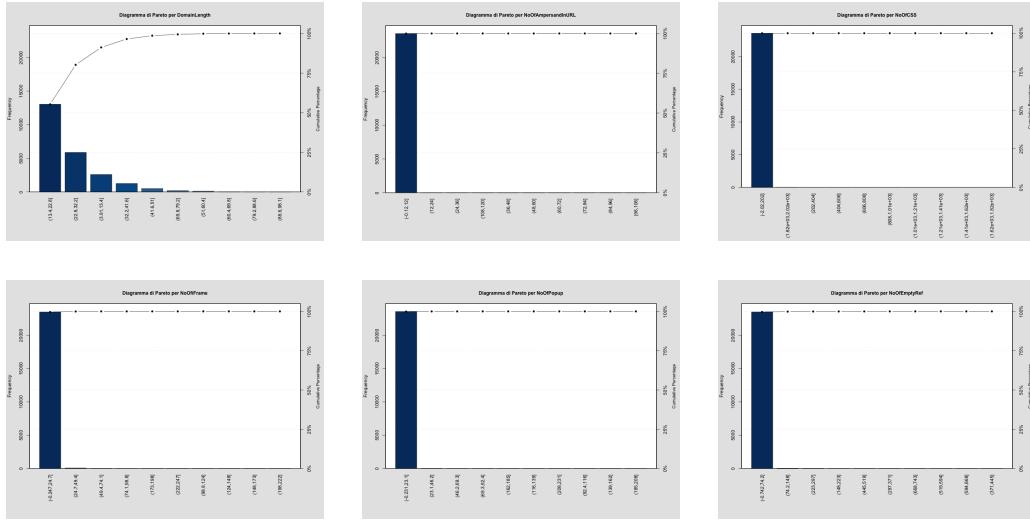
In seguito abbiamo effettuato la rappresentazione dei dati tramite istogrammi. Gli istogrammi si utilizzano per variabili quantitative e sono una particolare rappresentazione grafica di una distribuzione di frequenza in classi. Gli istogrammi ottenuti sono stati i seguenti:



Osservando gli istogrammi del dataset iniziale possiamo notare che la maggior parte di essi presentano una distribuzione asimmetrica a destra indicando un valore di Skewness positivo; la maggior parte dei dati è concentrata in un range ristretto, ma allo stesso tempo sono presenti valori eccezionalmente elevati.

1.2.4 Diagrammi di Pareto

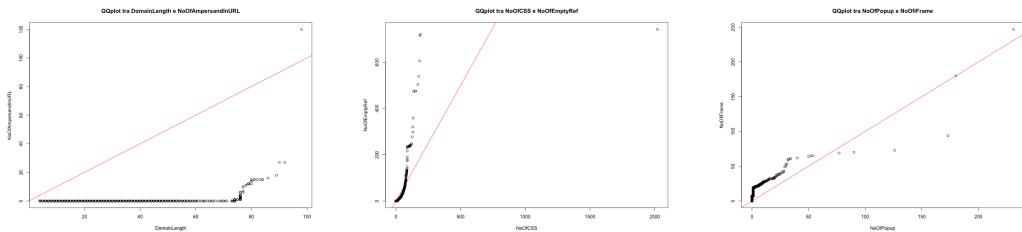
Il diagramma di **Pareto** prende il nome dall'economista italiano Vilfredo Pareto, ed è fondamentale per l'analisi di un insieme di dati in modo da determinarne le poche variabili che influenzano in modo significativo i risultati finali. Considerando le colonne definite nelle sezioni precedenti sono stati ricavati i seguenti risultati:



Da questi grafici si può subito notare che per tutte le variabili analizzate, pochi valori dominano la distribuzione, mentre gli outlier rappresentano una piccola percentuale ma con valori molto elevati.

1.2.5 QQPlot

Il **QQ Plot** è la rappresentazione grafica dei quantili di una distribuzione. Confronta la distribuzione cumulata della variabile osservata con la distribuzione cumulata della normale. Se la variabile osservata presenta una distribuzione normale, i punti di questa distribuzione congiunta si addensano sulla diagonale che va dal basso verso l'alto e da sinistra verso destra. Applicando queste informazioni al dataset i risultati ottenuti sono stati i seguenti:



In tutti e tre i QQ-Plot è possibile osservare un'elevata deviazione dalla linearità, indicando che le variabili non seguono la stessa distribuzione o che la relazione tra di esse è molto influenzata da outlier.

1.3 Selezione delle caratteristiche

1.3.1 Eliminazione delle feature derivate

Alcune di queste informazioni precedentemente citate sono state tuttavia eliminate, in quanto valori derivati secondo metodologie non riportate nel paper scientifico:

- **CharContinuationRate, URLTitleMatchScore, URLCharProb e TLDLegitimateProb** sono state eliminate poiché feature derivate.

1.3.2 Filtraggio per varianza

Dopo aver effettuato il primo filtraggio eliminando le feature derivate, è stato calcolato il valore della varianza per ogni colonna presente nel dataset. Le operazioni che sono state eseguite sono le seguenti:

- Conversione delle colonne in **fattoriali**.
- Generazione di un **array di varianze**.
- Calcolo della **varianza media** dell'array precedentemente ottenuto pari a: **258163735**.
- Filtraggio delle colonne utilizzando la media come **threshold**

Al termine di questa operazione **LargestLineLength** è stata eliminata dal dataset.

1.3.3 Filtraggio per Correlazione

Dopo aver applicato il filtro per varianza, abbiamo deciso di calcolare la matrice di correlazione sulle colonne rimanenti, la correlazione viene definita dalla seguente equazione:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Il metodo utilizzato dalla sopracitata equazione detta **indice di Pearson** ci mostra come le variabili aleatorie X e Y si influenzano a vicenda notando se crescono negativamente o positivamente oppure sono slegate, applicandoli a tutto il dataset otteniamo la seguente matrice di correlazione:

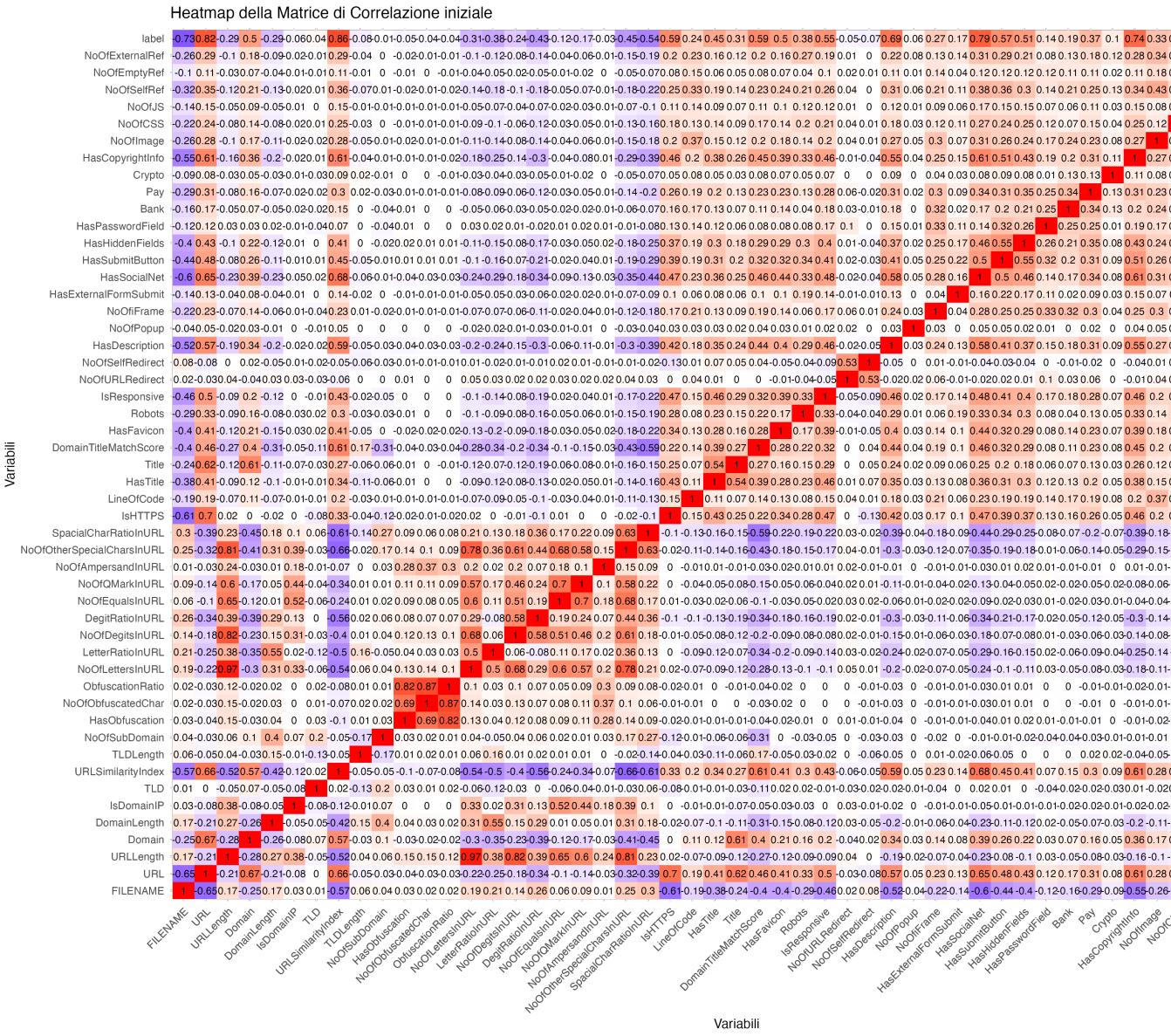


Figure 1: Before filter matrix correlation

Dalla precedente figura 1 vengono applicati i seguenti passi:

1. Calcolo della matrice di correlazione di Pearson per ogni coppia di colonne del dataset.
 2. Definizione del seguente criterio vengono selezionate tutte le coppie fortemente correlate positivamente e negativamente tramite le seguenti soglie 0.60 & -0.50.
 3. Delle coppie selezionate viene scelta una feature è viene rimossa dataset.

Al termine di questa operazione vengono eliminate le seguenti colonne: **DigitRatioInURL**, **Domain**, **DomainTitleMatchScore**, **FILENAME**, **HasCopyrightInfo**, **HasDescription**, **HasObfuscation**, **HasSocialNet**, **IsHTTPS**, **LetterRatioInURL**, **NoOfDigitsInURL**, **NoOfEqualsInURL**, **NoOfExternalRef**, **NoOfLettersInURL**,

NoOfObfuscatedChar, NoOfOtherSpecialCharsInURL, NoOfQMarkInURL, NoOfSelfRef, ObfuscationRatio, SpacialCharRatioInURL, Title, URL, URLLength, URLSimilarityIndex

Inoltre mostriamo la matrice finale delle features rimanenti:

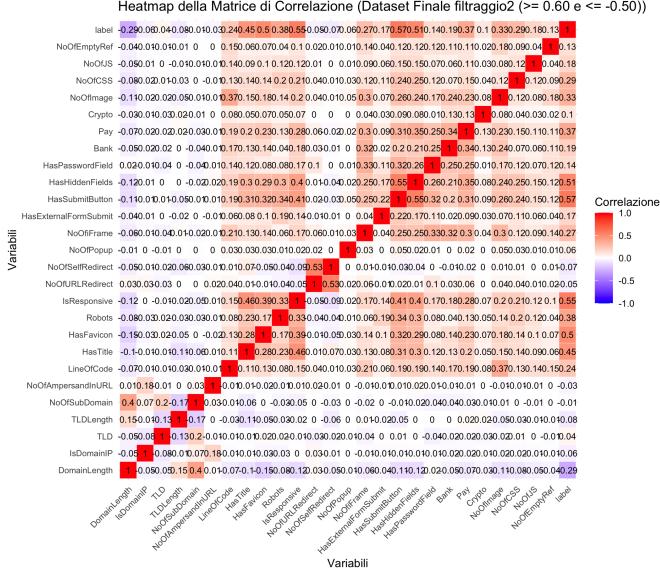


Figure 2: Final correlation matrix

1.3.4 Filtraggio tramite interpretabilità

L'interpretabilità si riferisce al grado in cui un individuo è in grado di prevedere coerentemente l'esito generato dal modello. Una maggiore interpretabilità di un modello di apprendimento automatico facilita la comprensione, da parte di un individuo, delle motivazioni sottese a determinate decisioni o previsioni. Un modello è ritenuto più interpretabile rispetto a un altro qualora le sue decisioni risultino più facilmente comprensibili per un essere umano rispetto a quelle prodotte dall'altro modello. Abbiamo impiegato un metodo **globale** e due **locali**. I metodi globali descrivono il comportamento medio di un modello di apprendimento automatico, mentre in contrapposizione, i metodi locali interpretano i modelli attraverso l'analisi delle singole predizioni. I metodi globali sono generalmente rappresentati come valori attesi derivanti dalla distribuzione dei dati.

Nell'ambito dei metodi globali, utilizziamo la **Permutation feature importance**. Il concetto è relativamente semplice: si valuta l'importanza di una caratteristica misurando l'incremento dell'errore di previsione del modello successivamente alla permutazione dei valori della caratteristica stessa. Una caratteristica è considerata 'importante' se la riorganizzazione dei suoi valori provoca un aumento nell'errore del modello, indicando che il modello ha fatto affidamento su di essa per effettuare la previsione. Al contrario, una caratteristica è considerata 'non importante' se il rimescolamento dei suoi valori non altera l'errore del modello, indicando che il modello non ha utilizzato tale caratteristica nella previsione. La misura dell'importanza della caratteristica di permutazione è stata introdotta da Breiman (2001)43. Sulla base di tale concetto, Fisher, Rudin e Dominici (2018) hanno proposto una versione "model-agnostic" dell'importanza delle caratteristiche, designata "model reliance."

Inoltre, hanno introdotto idee più avanzate sull'importanza delle caratteristiche, quali una versione specifica per il modello che riconosce il fatto che diversi modelli di previsione possono predire efficacemente i dati.

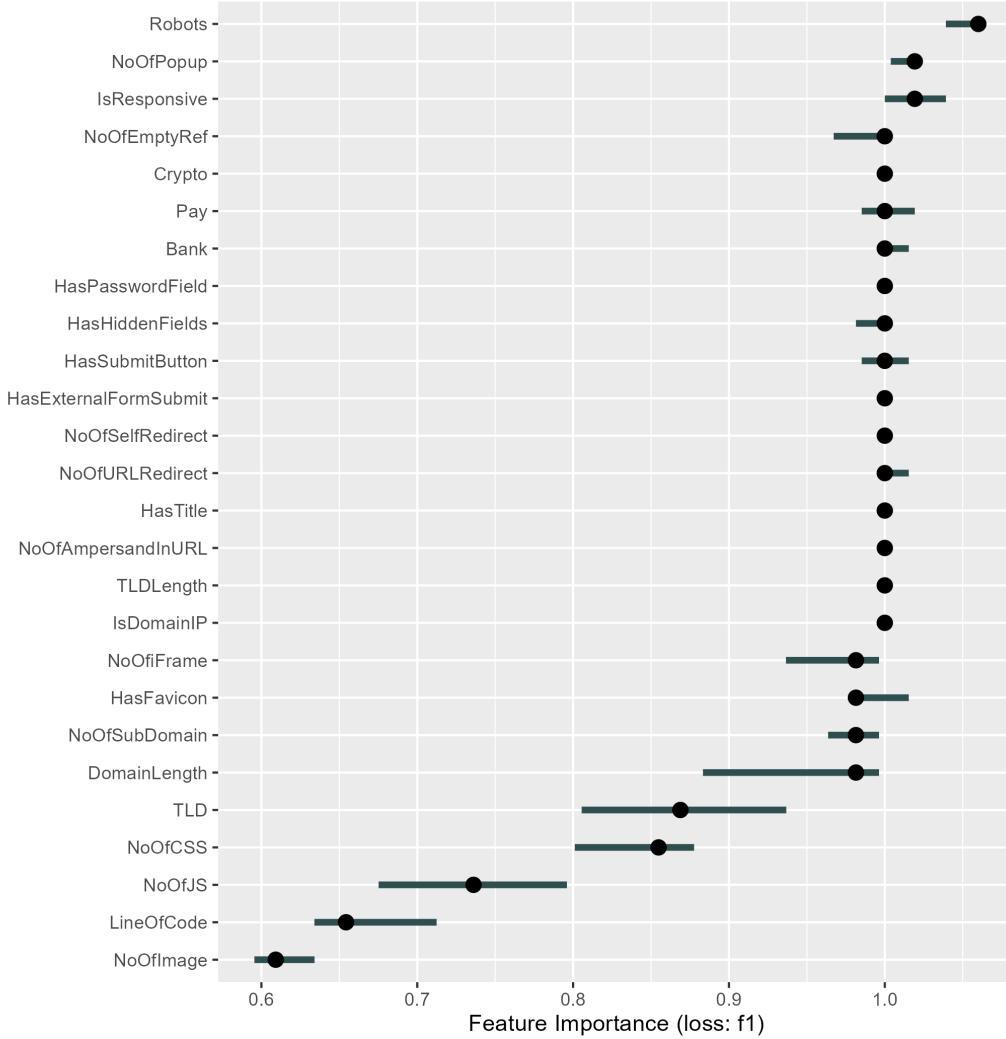


Figure 3: Permutation Feature Importance plot

Dalla figura 3, si possono fare le seguenti osservazioni:

- la metrica scelta per effettuare il calcolo dell'importanza è l'**f1-score**, una metrica resistente agli outlier e fondamentale per il task di classificazione sia binaria che multoclasse.
- La feature su cui il KNN fa maggior affidamento è **robots**, la peggiore è **NoOfImage** con i rispettivi valori quando vengono permutati di 1.2 e 0.63.
- L'85% delle colonne del dataset quando vengono permutate variano di un intervallo tra 0.80 e 1.2 rispetto al metrica adottata, dimostrando l'efficacia dei precedenti filtri.

Infine specifichiamo il criterio di filtraggio usato vengono selezionate come candidate al filtraggio tutte le colonne che quando permutate generavano un valore si FI inferiore a 0.80 in questo caso sono **NoOfJS**, **LineOfCode** e **NoOfImage** si pospone la conferma o l'allungamento di questa lista dopo l'applicazione dei metodi locali. Cominciamo con LIME, una tecnica locale che impiega modelli surrogati locali interpretabili per spiegare le previsioni individuali dei modelli di apprendimento automatico a scatola nera. Diversamente dall'approccio di addestramento di un modello surrogato globale, LIME privilegia la creazione di modelli surrogati locali per chiarire specifiche previsioni. L'idea è piuttosto intuitiva: si tratta di ignorare momentaneamente i dati originali di addestramento e considerare unicamente un modello a scatola nera in cui è possibile inserire dati e ricevere previsioni. Il modello permette esplorazioni libere, consentendo di comprendere le motivazioni delle specifiche previsioni del modello di apprendimento automatico. LIME verifica gli effetti sulle previsioni apportando variazioni ai dati forniti al modello di apprendimento. Esso genera un nuovo insieme di dati costituito da campioni perturbati e dalle corrispettive previsioni del modello a scatola nera. Su questo insieme di dati, LIME addestra un modello interpretabile, ponderato sulla base della vicinanza delle istanze campionate rispetto all'istanza in esame. Il modello interpretabile può essere selezionato tra vari modelli presentati nei capitoli dedicati, come Lasso o un albero decisionale. È fondamentale che il modello appreso approssimi accuratamente le previsioni del modello di apprendimento automatico a livello locale, pur non essendo necessariamente accurato a livello globale. Questo tipo di accuratezza è noto come fedeltà locale.

Osservando la figura 4, notiamo che:

- A differenza del metodo globale LIME evidenzia le top 3 features che hanno avuto effetto sulla predizione sia dal modello a scatola nera sia per il modello locale con cui si cerca di interpretare la previsioni nel nostro caso un regressore lasso.
- Il criterio utilizzato è il seguente viene selezionato una soglia in questo caso 0.35 viene effettuata un differenza tra la previsione del modello a scatola nera ed il lasso sé il risultato è superiore la predizione viene definita negativa ed la feature che ha avuto la maggior magnitudine viene selezionata per essere scartata.

Discutiamo ora un altro approccio locale, i valori di Shapley. Una previsione può essere interpretata considerando ciascun valore caratteristico di un'istanza come un "giocatore" in un gioco in cui il risultato è la previsione stessa. I valori di Shapley, presi dalla teoria dei giochi coalizionali, indicano come dividere equamente il "risultato" tra le varie caratteristiche. L'effetto di ciascuna caratteristica dipende dal prodotto tra il suo peso e il suo valore. Questo metodo è efficace grazie alla linearità del modello, ma nei modelli più complessi è necessaria un'alternativa. Per esempio, LIME utilizza modelli locali per stimare tali effetti. Un'altra soluzione arriva dalla teoria dei giochi cooperativi: Il valore di Shapley, proposto da Shapley nel 1953, è un metodo per distribuire i pagamenti ai giocatori in funzione del loro apporto al guadagno totale. I giocatori cooperano all'interno di una coalizione e ottengono un certo ritorno da tale cooperazione. In questo contesto, il "gioco" si riferisce al compito di previsione per una specifica istanza del dataset, mentre il "guadagno" rappresenta la previsione per questa istanza al netto della media delle previsioni per tutte le istanze. I "giocatori" sono i valori caratteristici dell'istanza che collaborano per ottenere il guadagno, ovvero prevedere un certo valore.

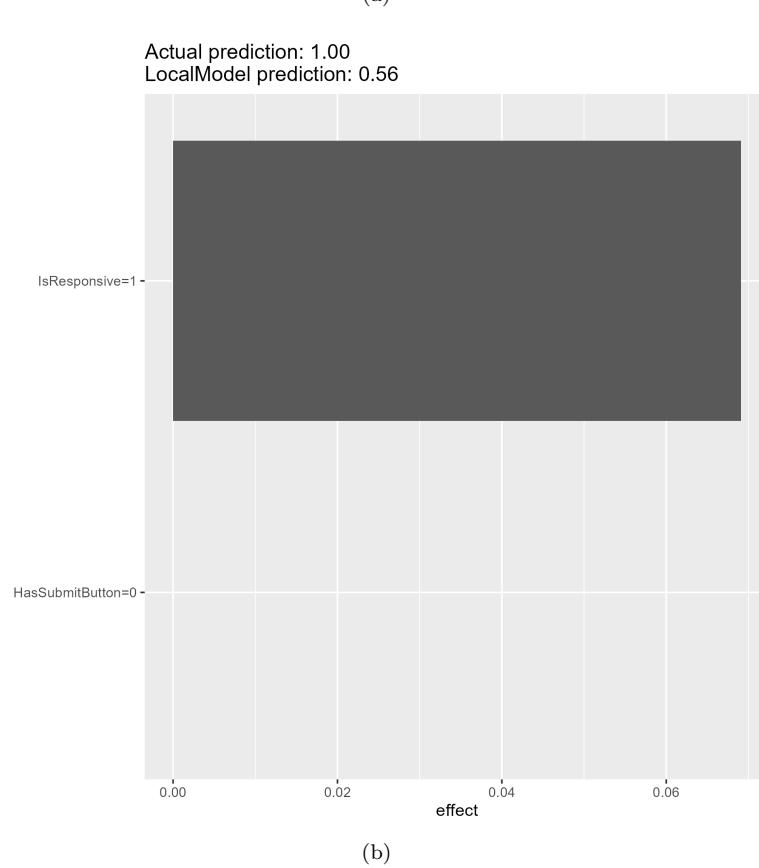
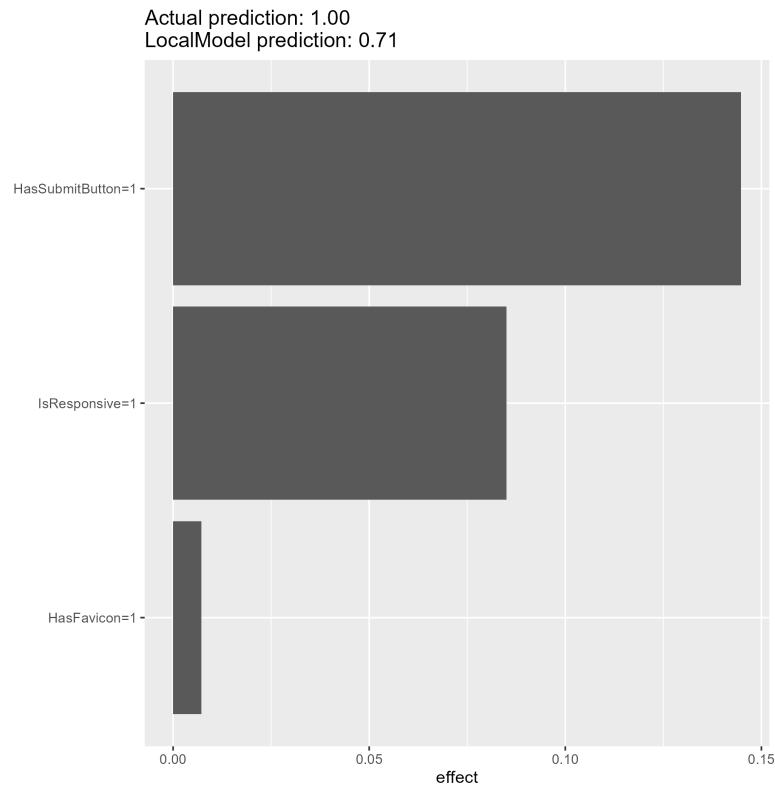
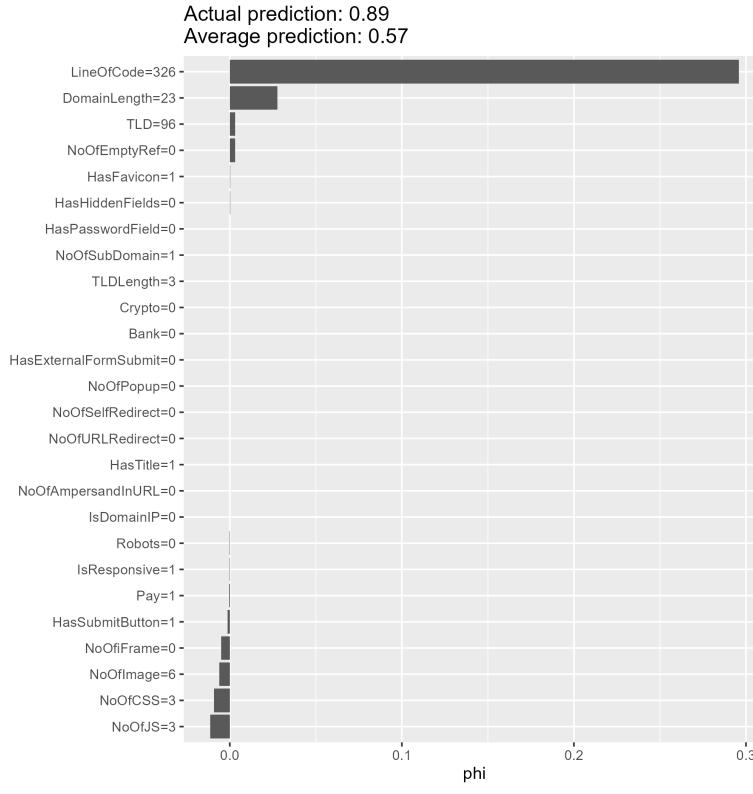
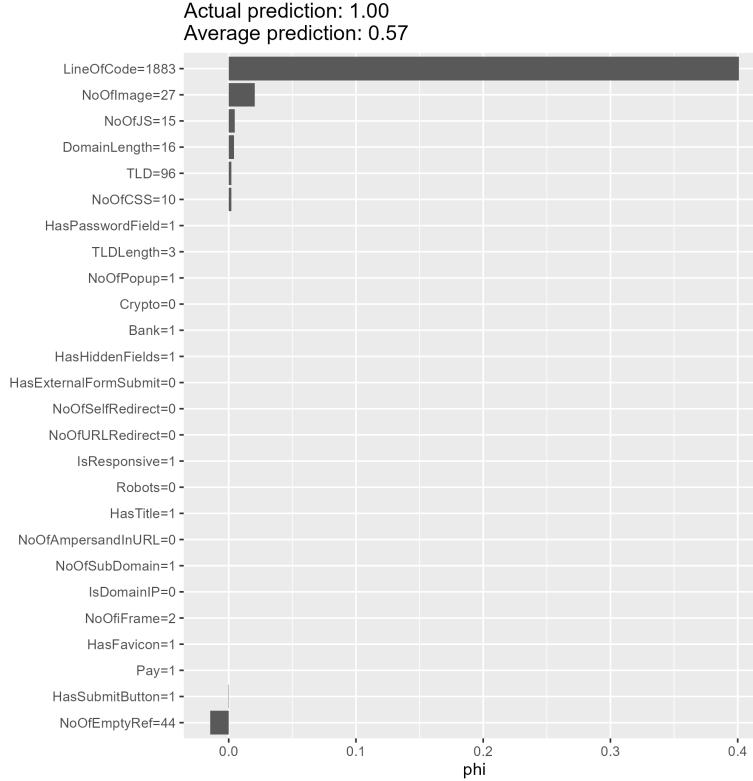


Figure 4: (a) Positive Prediction (b) Negative Prediction



(a)



(b)

Figure 5: (a) Positive Prediction (b) Negative prediction

Analizzando la figura 5, possiamo affermare che

- A differenza di LIME, gli shap value sono valutati su tutti i possibili insiemi di features che hanno contribuito alla predizione viene quantificato con phi.
- Il criterio per la selezione delle colonne da filtrare ha la stessa soglia e metodologia di quella applicata al metodo LIME.

In conclusione vengono filtrati i candidati che hanno avuto riscontro nella maggioranza dei metodi presentati, che in questo caso sono: **NoOfJS**, **LineOfCode**, **NoOfImage**, **IsResponsive**.

2 Analisi del Dataset Filtrato

Al termine dei vari filtri precedentemente citati il dataset ottenuto presenta le seguenti colonne:

- **DomainLength**
- **IsDomainIP**
- **TLD**
- **TLDLength**
- **NoOfSubDomain**
- **NoOfAmpersandInURL**
- **HasTitle**
- **HasFavicon**
- **Robots**
- **NoOfURLRedirect**
- **NoOfSelfRedirect**
- **NoOfPopup**
- **NoOfFrame**
- **HasExternalFormSubmit**
- **HasSubmitButton**
- **HasHiddenFields**
- **HasPasswordField**
- **Bank**
- **Pay**
- **Crypto**
- **NoOfCSS**
- **NoOfEmptyRef**
- **label**

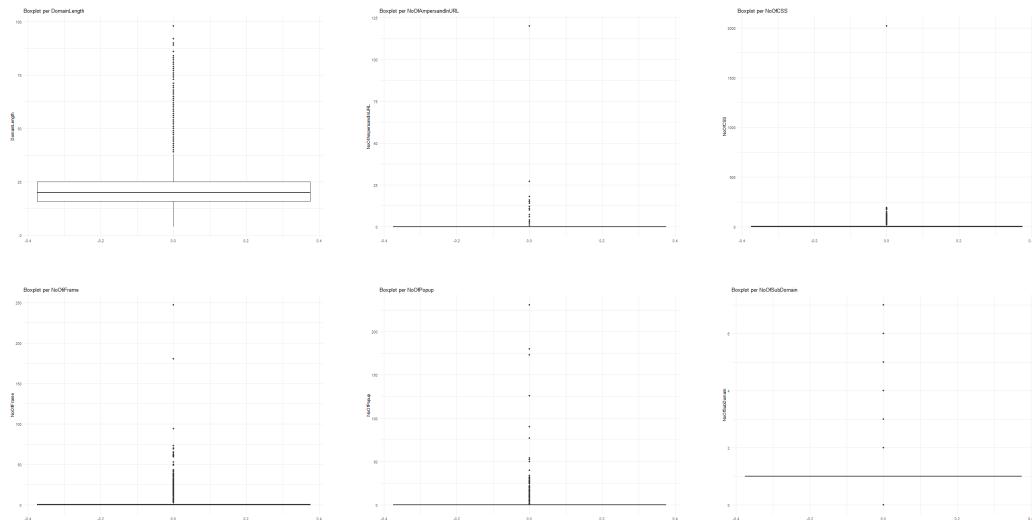
3 Rappresentazione grafica

Successivamente all'ottenimento dei dati riportati nella sezione precedente, è stata applicata della statistica descrittiva che ci consentisse di creare dei grafici, in modo da valutarne le caratteristiche. Sono state scelte sei colonne come esempio per mostrare i risultati ottenuti dopo il filtraggio del dataset ovvero:

- DomainLength
- NoOfAmpersandInURL
- NoOfCSS
- NoOfPopup
- NoOfFrame
- NoOfSubDomain

3.1 Boxplot

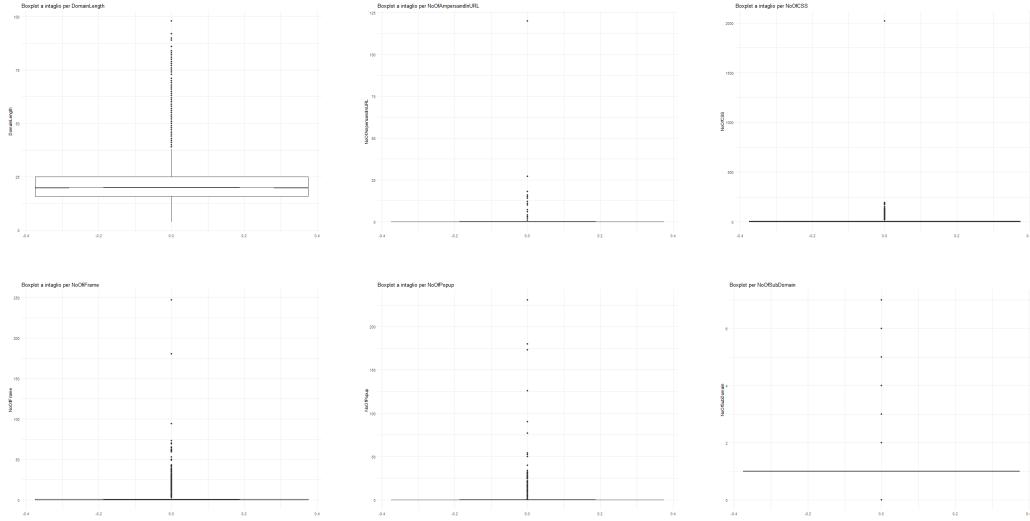
Per prima cosa mostreremo i **boxplot** ricavati; analizzandone le caratteristiche e commentando i risultati ottenuti::



Analizzando questi grafici siamo giunti ad una conclusione che accuma tutte le colonne scelte, ovvero che ognuna di esse presenta un range interquartile basso. Le caratteristiche dei grafici qui sopra riportati sono uguali a quelle precedentemente citate per il dataset iniziale.

3.2 Boxplot ad intaglio

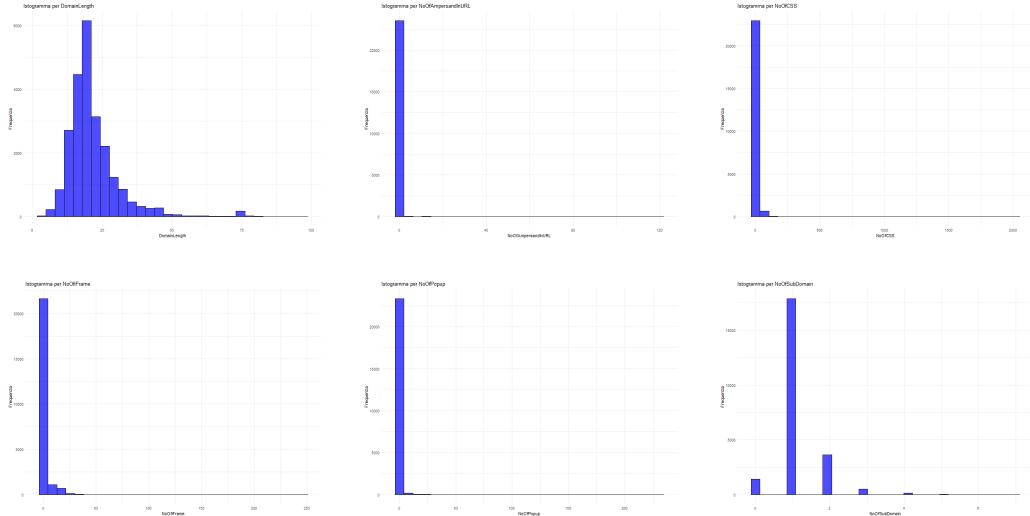
Come secondo approccio mostriamo i risultati ottenuti tramite la rappresentazione dei dati tramite boxplot ad intaglio:



Così come nei grafici visualizzati precedentemente, possiamo notare che tutte le colonne sono accomunate da un range interquartile stretto, ovvero che la maggior parte dei dataset ha la lunghezza compresa tra il primo ed il terzo quartile.

3.3 Istogrammi

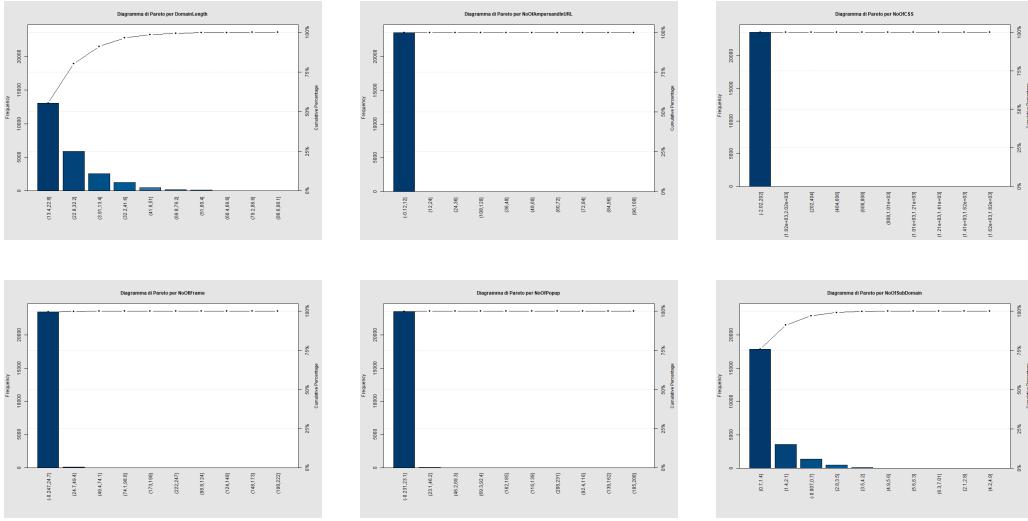
In seguito abbiamo effettuato la rappresentazione dei dati tramite istogrammi ed ottenuto i seguenti risultati:



Analogamente al dataset originario anche qui possiamo notare un'asimetria a destra, con un elevato numero di outlieri.

3.4 Diagramma di Pareto

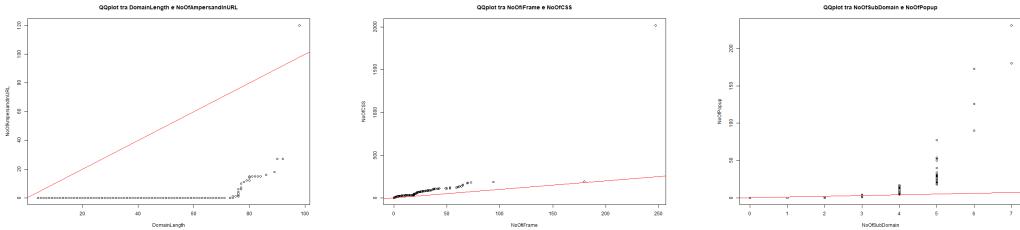
In questa sezione rappresenteremo i grafici ottenuti tramite l'approccio di Pareto. I risultati ottenuti sono stati i seguenti:



In tutti i grafici, si osserva una distribuzione fortemente sbilanciata. I dati confermano il principio di Pareto: una piccola percentuale di valori rappresenta una frazione minima del totale.

3.5 Q-QPlot

I qqplot ottenuti per il dataset filtrato sono i seguenti:



Analogamente al dataset completo troviamo risultati quasi simili; in tutti e tre i QQ-Plot è possibile osservare un'elevata deviazione dalla linearità, indicando che le variabili non seguono la stessa distribuzione o che la relazione tra di esse è molto influenzata da outlier.

4 Studio delle distribuzioni

Dopo aver effettuato questo primo grande filtraggio, uno degli step successivi è stato lo studio delle distribuzioni utilizzando i momenti d'ordine. I momenti d'ordine utilizzati sono i seguenti:

- **media, moda e mediana**
- **varianza e coefficiente di variazione**
- **Skewness**
- **Kurtosis**

In questa sezione analizzeremo nel dettaglio ognuno di questi momenti d'ordine.

4.1 Media, moda e mediana

La media campionaria è semplicemente la media delle variabili di un campione ed è rappresentabile come segue:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Per ogni valore i-esimo è possibile ottenere lo **scarto** della media campionaria che è pari alla differenza tra il valore i-esimo e la media campionaria; ciò indica il grado di scostamento del singolo valore i-esimo dalla media campionaria. La mediana campionaria, invece, bipartisce le osservazioni (dopo aver effettuato un ordinamento crescente) in due gruppi di uguale numerosità, in modo che ricada un egual numero di valori sia a sinistra sia a destra della mediana stessa. Media e Mediana sono fondamentali per la descrizione delle misure di centralità dei dati. La moda campionaria di un insieme di dati, è la modalità a cui è associata la frequenza (assoluta o relativa) più elevata. Se esistono più modalità con frequenza massima, ciascuna di esse è definita come valore modale. Media, moda e mediana sono elementi fondamentali facenti parte dei **momenti d'ordine di primo tipo**. Per effettuare il calcolo di media, moda e mediana è stato creato uno script **R**, che ci ha consentito di ricavare i seguenti valori per ogni colonna del dataset filtrato:

Osservazione	Media	Moda	Mediana
DomainLength	21.57	19	20.00
IsDomainIP	0.002417	0	0.000000
TLD	148.9	96	96.0
TLDLength	2.767	3	3.000
NoOfSubDomain	1.163	1	1.000
NoOfAmpersandInURL	0.02774	0	0.00000
HasTitle	0.8637	1	1.0000
HasFavicon	0.3581	0	0.0000
Robots	0.2681	0	0.0000
NoOfURLRedirect	0.1332	0	0.0000
NoOfSelfRedirect	0.04139	0	0.00000
NoOfPopup	0.1982	0	0.0000
NoOfFrame	1.594	0	0.000
HasExternalFormSubmit	0.04411	0	0.00000
HasSubmitButton	0.4133	0	0.0000
HasHiddenFields	0.3751	0	0.0000
HasPasswordField	0.1026	0	0.0000
Bank	0.1281	0	0.0000
Pay	0.2422	0	0.0000
Crypto	0.02511	0	0.00000
NoOfCSS	6.232	0	2.000
NoOfEmptyRef	2.445	0	0.000

Table 1: Risultati di Media, Moda e Mediana

Dopo un'attenta analisi dei dati sono state eliminate le seguenti colonne in quanto aventi valori pari a **NA**:

- **IsDomainIP**
- **NoOfAmpersandInURL**

- **NoOfSelfRedirect**
- **HasExternalFormSubmit**
- **Crypto**

4.2 Varianza e deviazione standard

Gli indici di posizione non tengono traccia della variabilità dei dati, infatti molte distribuzioni sono diverse tra loro, seppur con media campionaria uguale. Per misurare la variabilità di una distribuzione di frequenze sono fondamentali il calcolo della **varianza campionaria** e la **deviazione standard campionaria**

4.2.1 Coefficiente di Variazione

Il coefficiente di variazione è definito come il rapporto tra la deviazione standard campionaria e il modulo della media campionaria:

$$CV = \frac{s}{|\bar{x}|}$$

Il coefficiente di variazione è un numero puro, poiché non dipende dalle unità di misura utilizzate. Il coefficiente di variazione è un indice di dispersione e nei nostri test è stato utilizzato per visualizzare la dispersione dei dati per ogni colonna presente nel dataset. I valori ottenuti per ogni colonna sono stati i seguenti:

Osservazione	CV
DomainLength	0.4%
TLD	0.6%
TLDLength	0.2%
NoOfSubDomain	0.5%
HasTitle	0.4%
HasFavicon	1.3%
Robots	1.7%
NoOfURLRedirect	2.6%
NoOfPopup	14%
NoOfFrame	3%
HasSubmitButton	1.2%
HasHiddenFields	1.3%
HasPasswordField	3%
Bank	3%
Pay	2%
NoOfCSS	3%
NoOfEmptyRef	6.5%

Table 2: Coefficienti di Variazione

L'analisi dei coefficienti di variazione ha fornito informazioni fondamentali sulla dispersione relativa delle variabili del dataset filtrato. Per interpretare al meglio il coefficiente di variazione bisogna considerare che:

- **CV < 10%**: e indica una bassa variabilità rispetto alla media. I dati sono relativamente stabili.

- **$10\% \leq CV \leq 30\%$** : moderata variabilità
- **$CV > 30\%$** : elevata variabilità, suggerendo che i dati hanno una notevole dispersione rispetto alla media

Nel nostro caso possiamo notare che molte tabelle del dataset presentano un CV inferiore al 10%, dimostrando una buona stabilità.

4.3 Kurtosis e Skewness

Nella teoria di probabilità e statistica un indice che consente di misurare la simmetria di una distribuzione di frequenze è la **Skewness Campionaria**, definita come tale:

$$\gamma_1 = \frac{m_3}{m_2^{3/2}}$$

dove m_3 denota il momento centrato campionario di ordine 3. Nel caso in cui γ_1 ottenga un valore pari a 0 verrà definita **simmetrica**, > 0 sarà asimmetrica positiva, < 0 assimetrica negativa. Un altro indice fondamentale nel calcolo statistico, che consente di misurare la densità dei dati intorno alla media è la curtosi campionaria, quest'ultima definita come tale:

$$\gamma_2 = \beta_2 - 3$$

dove β_2 è pari a:

$$\beta_2 = \frac{m_4}{m_2^2}$$

e rappresenta l'indice di Pearson, con m_2 definito come il momento centrato campionario di ordine 2 e con m_4 il momento centrato campionario di ordine 4. Basandoci su queste informazioni abbiamo, tramite R, abbiam ricavato i valori di Skewness e Kurtosis per ogni colonna presente all'interno del dataset. I valori ottenuti per ogni colonna presente nel dataset sono i seguenti:

Osservazione	Skewness	Kurtosis
DomainLength	2.469991	10.12069
IsDomainIP	0.2026412	408.6517
TLD	1.029455	-0.2530098
TLDLength	1.706009	14.03742
NoOfSubDomain	1.793807	7.308409
NoOfAmpersandInURL	93.07487	11293.04
HasTitle	-2.119355	2.491773
HasFavicon	0.5920928	-1.649496
Robots	1.046908	-0.9040226
NoOfURLRedirect	2.158761	2.66036
NoOfSelfRedirect	4.60438	19.20113
NoOfPopup	52.36099	3489.652
NoOfFrame	12.06871	411.8172
HasExternalFormSubmit	4.44035	17.71746
HasSubmitButton	0.3520434	-1.876145
HasHiddenFields	0.5160846	-1.73373
HasPasswordField	2.618644	4.857502
Bank	2.225804	2.954329
Pay	1.203451	-0.5517298
Crypto	6.070597	34.85362
NoOfCSS	71.00408	8255.962
NoOfEmptyRef	26.08677	923.5378

Table 3: Skewness e Kurtosis

Visualizzando la tabella sono stati estratti i valori di Skewness e Kurtosis, e dopo un'attenta osservazione abbiamo posto come ipotesi statistica quella di considerare le seguenti colonne (aventi tre ordini a favore):

- **HasFavicon**
- **NoOfURLRedirect**
- **Bank**

5 Clustering

5.1 K-Nearest Neighbour

L'algoritmo dei k-nearest neighbors (KNN) è un classificatore supervisionato utilizzato per l'apprendimento non parametrico basato sulla vicinanza tra punti per classificare dati o prevedere la posizione di un punto in un gruppo. È uno dei metodi di classificazione e regressione più semplici e largamente adottati nel machine learning. Pur essendo applicabile sia alla regressione sia alla classificazione, di solito viene usato come algoritmo di classificazione, poiché si fonda sul principio che i punti simili sono situati vicini nello spazio. Il KNN si caratterizza per due elementi chiave: la metrica della distanza e la scelta del K:

- **Distanza:** Per stabilire quali punti dati siano più vicini a un dato punto di interrogazione, occorre calcolare la distanza tra quest'ultimo e gli altri punti. Queste metriche di distanza contribuiscono a costruire i confini decisionali che dividono i punti di interrogazione in varie regioni, nel nostro caso viene scelta la distanza euclidea data dalla seguente formula:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

- **Determinazione del k:** Il valore k nell'algoritmo k-NN determina quanti vicini considerare per stabilire la classificazione di un certo punto di interrogazione. Ad esempio, con $k=1$, l'istanza verrà attribuita alla stessa classe del vicino più prossimo. Definire il k è un processo delicato, perché valori diversi possono provocare overfitting o underfitting. Valori di k più bassi possono portare ad alta varianza e basso bias, mentre valori di k più alti possono risultare in alto bias e bassa varianza. La scelta del k dipende fortemente dai dati di input, poiché un dataset con molti outlier o rumore potrebbe funzionare meglio con k elevati. Generalmente, si consiglia di scegliere un numero dispari per k per evitare errori di classificazione, e le tecniche di cross-validation possono essere utili per determinare il k ottimale per un dataset specifico.

5.2 Addestramento del Modello

La fase di addestramento viene caratterizzato da alcuni passaggi fondamentali, in primis il dataset viene diviso in train e test con un rapporto di 80% - 20%; in seguito vengono effettuate 2 operazioni di preprocessing che sono il **centering** e lo **scaling**. Il primo è fondamentalmente una tecnica in cui la media delle variabili indipendenti viene sottratta da tutti i valori. Ciò significa che tutte le variabili indipendenti hanno media zero. Il secondo è simile ma le variabili predittive vengono divise per la loro deviazione standard. In questo modo i dati avranno una deviazione standard di uno. Trasformando i tuoi dati aiuti il modello a funzionare in modo efficiente. Ad esempio, i modelli basati sulla distanza funzionano bene quando i dati vengono preelaborati e trasformati. Il motivo è che, avendo tutte le funzionalità ridimensionate, il tuo modello diventa più veloce; migliore accuratezza e modello più generalizzato. In conclusione come metodo di addestramento viene scelta la **repeated cross validation** fornisce un modo per migliorare le prestazioni stimate di un modello di apprendimento automatico. Ciò comporta semplicemente la ripetizione della procedura di convalida incrociata più volte e la segnalazione del risultato medio su tutte le pieghe da tutte le esecuzioni. Si prevede che questo risultato medio sia una stima più accurata delle vere prestazioni medie sconosciute sottostanti del modello sul set di dati, come calcolato utilizzando l'errore standard.

5.3 Risultati degli Esperimenti

Osservando la tabella 4 possiamo dire che il modello ha delle performance generali ottime questo viene affermato poichè su tutte le metriche prese in considerazione si ha un valore al di sopra del 80%; oltre questo le altre metriche approfondiscono tali performance confermando la bontà del modello nel nostro caso di phishing sono la specificity che conferma la capacità di riconoscere il phishing, è l'f1-score che ci dice che il modello ha sia una buona precision sia un

alta recall, anche se la sensitivity e la recall stessa che risultano essere le più basse pongono un limite al riconoscimento della classe no phising ciò è dovuto ad uno sbilanciamento del etichette rispetto alla popolazione del dataset. Infine la fig 6 con la roc curve ci mostra un alto rendimento nel discriminare la presenza o no del phishing negli url analizzati, lasciando però un possibilità di miglioramento per quanto riguarda i false positive rate.

Metriche	Valori
Accuracy	94%
Sensitivity	85%
Specificity	97%
Precision	91%
Recall	85%
F1-score	88%

Table 4: Risultati KNN sul dataset filtrato

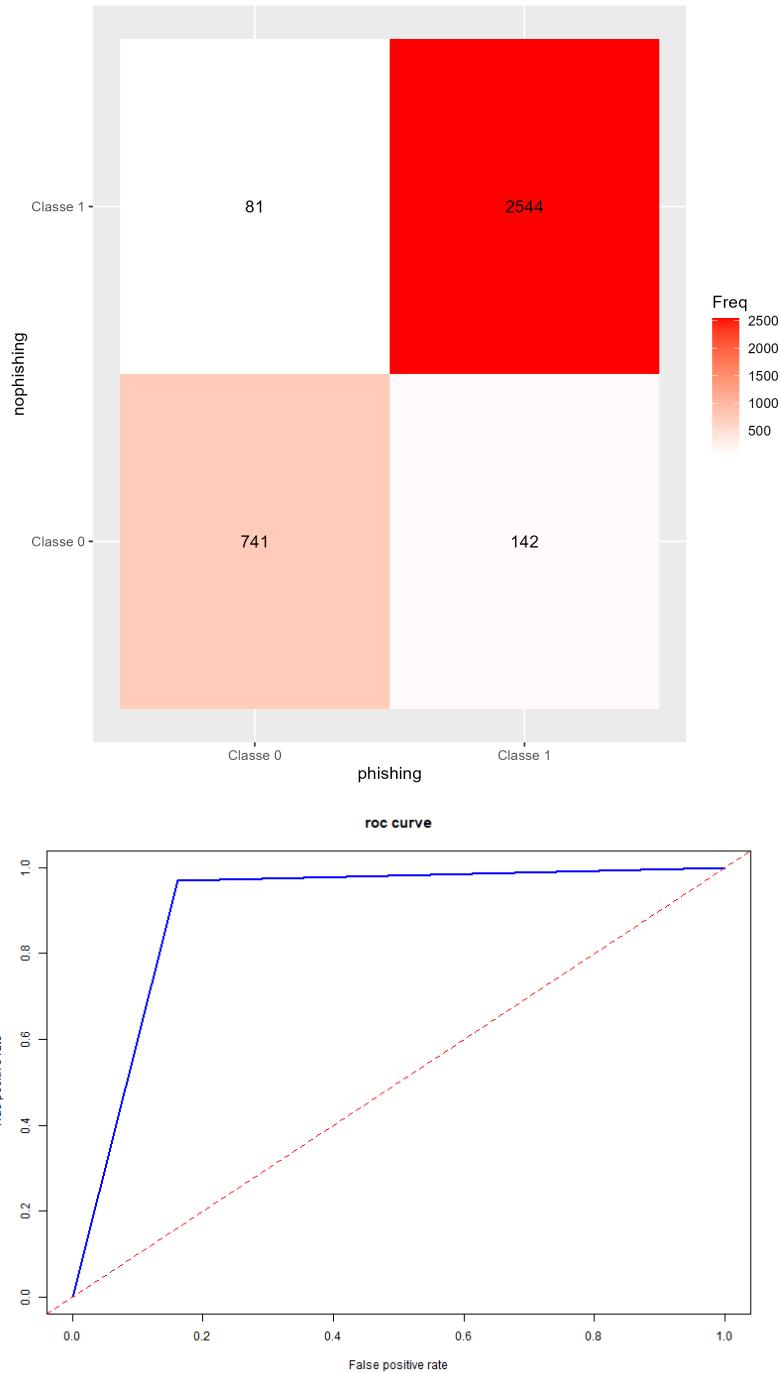


Figure 6: Metriche grafiche per i risultati de KNN

6 Test del Chi quadrato bilaterale

6.1 Cos'è il test del chi quadrato

Con il criterio del chi-quadrato si desidera verificare l'ipotesi che un certa popolazione, descritta da una variabile aleatoria X, sia caratterizzata da una funzione di distribuzione

$FX(x)$, con k parametri non noti da stimare. Denotando con H_0 l'ipotesi soggetta a verifica (ipotesi nulla) e con H_1 l'ipotesi alternativa, il test chi-quadrato con livello di significatività α mira a verificare l'ipotesi nulla. Occorre determinare un test ψ con livello di significatività α che permetta di determinare una regione di accettazione e di rifiuto dell'ipotesi nulla.

6.2 Risultati del Dataset filtrato

Dopo aver introdotto e spiegato il concetto di Chi Quadrato bilaterale, ora andremo a trascrivere i risultati ottenuti effettuando il test **normale** e test **binomiale**. E' stata effettuata una valutazione dei valori del test normale e test binomiale, tramite un'approssimazione a tre cifre. Il primo passaggio che è stato effettuato è stato il calcolo della normale relative alle osservazioni riportare nella tabella sottostante:

Osservazione	Chi2	First	Last	NObs
DomainLength	2837.603	0.05063562	7.377759	2058 5703 4537 3069 2173
TLD	15381.27	0.05063562	7.377759	1284 9398 980 1031 4847
TLDLength	27561.75	0.05063562	7.377759	5618 0 0 11233 689
NoOfSubDomain	37859.56	0.05063562	7.377759	843 13556 0 0 3141
NoOfAmpersandInUrl	69383.47	0.05063562	7.377759	0 0 17462 0 78
NoOfPopup	64913.59	0.05063562	7.377759	0 0 17003 291 246
NoOfFrame	18699.36	0.05063562	7.377759	0 9872 5143 1134 1391
NoOfCSS	13685.22	0.05063562	7.377759	0 8195 6069 1939 1337
NoOfEmptyRef	55272.12	0.05063562	7.377759	0 0 15936 1104 500

Table 5: Risultati Test normale

Abbiamo valutato i valori ricavati ed abbiamo concluso col dire che nessuna delle colonne analizzate ha una distribuzione normale, in alcuni casi ad esempio:

- **NoOfAmpersandInUrl** il valore del chi quadro non è approssimato bene poiché non rispetta la condizione minima del numero di osservazioni minima per ogni intervallo

$$\min(n * p_i) \geq 5$$

con n pari a cinque, ed i che va da 1 a 5

Questo coincide con le osservazioni fatte con gli indici di sintesi ed i momenti d'ordine. Successivamente è stato effettuato il test binomiale sulle osservazioni del dataset aventi valori binari, ed avendo considerato le osservazioni come una sequenza indipendente di Bernoulli, abbiamo ricavato i seguenti dati:

Osservazione	Chi2	First	Last	NObs
IsDomainIP	0.1308448	0.0009820691	5.023886	17490 50
HasFavicon	0.001676062	0.0009820691	5.023886	9562 7978
NoOfURLRedirect	0.004397977	0.0009820691	5.023886	15134 2406
Bank	0.004723647	0.0009820691	5.023886	14772 2768
HasTitle	0.001406997	0.0009820691	5.023886	1387 16153
Robots	0.004541924	0.0009820691	5.023886	11502 6038
NoOfSelfRedirect	0.1195464	0.0009820691	5.023886	16953 587
HasExternalFormSubmit	0.024255	0.0009820691	5.023886	16510 1030
HasSubmitButton	0.001877987	0.0009820691	5.023886	8048 9492
HasHiddenFields	0.007318634	0.0009820691	5.023886	9144 8396
HasPasswordField	0.001637731	0.0009820691	5.023886	15209 2331
Pay	0.01006632	0.0009820691	5.023886	12114 5426
Crypto	5.78862e-05	0.0009820691	5.023886	16961 579

Table 6: Risultati Test Binomiale

Dai dati ricavati abbiamo concluso che la maggior parte dei dati ha esito positivo, tranne **Crypto** che è risultata negativa al test binomiale. Al termine dei due test effettuati siamo giunti alla conclusione che si è giunti al **95%** di confidenza. Per concludere abbiamo applicato come ultimo test, il test di Poisson. La distribuzione di Poisson si utilizza è una distribuzione casuale discreta di media e varianza identiche che si utilizza per calcolare la probabilità che un evento si ripeta esattamente x volte in una certa unità spaziotemporale. Abbiamo deciso di applicare Poisson sui dati che non hanno superato il test della normale ricavando i seguenti risultati:

Osservazione	Chi2	First	Last	NObs
DomainLength	1018.854	0.2157953	9.348404	8 9 12 51 17460
TLD	1.294545e+62	0.2157953	9.348404	4 3 1 1 17531
TLDLength	22484.64	0.2157953	9.348404	5618 11233 513 75 101
NoOfSubDomain	13446.45	0.2157953	9.348404	843 13556 2656 368 117
NoOfAmpersandInUrl	1555702	0.2157953	9.348404	17462 21 13 8 36
NoOfPopup	23004.52	0.2157953	9.348404	16192 811 210 81 246
NoOfFrame	32450.74	0.2157953	9.348404	9872 2252 1950 941 2525
NoOfCSS	17540	0.2157953	9.348404	3289 1934 1709 1263 9345
NoOfEmptyRef	148237.9	0.2157953	9.348404	10348 2111 1183 760 3138

Table 7: Risultati Test Poisson

Osservando i risultati ottenuti dal test di Poisson si può notare che i dati che non avevano passato il test della normale, non hanno nemmeno passato il test di Poisson. Al termine di queste analisi siamo giunti alla conclusione che non è possibile stabilire la popolazioni di questi, e che dovrebbero essere studio di ricerche future.

7 Modello linguistico

7.1 LLM Research Question

Nella fase di scelta del modello linguistico abbiamo selezionato il modello multimodelle GPT-4o con l'obiettivo di rispondere alle seguenti Research Question (RQ):

- **RQ1:** Il dataset sintetico migliora le performance del knn nel contesto del riconoscimento phishing?
- **RQ2:** Le feature generate dai LLM possono essere ricondotte a distribuzioni statistiche note?
- **RQ3:** Il dataset sintetico ha un indice di stabilità maggiore rispetto al dataset filtrato?

7.2 Generazione del dataset sintetico tramite LLM

Per la generazione del dataset sintetico viene utilizzato dapprima il seguente prompt:

Generami un dataset simile per mio progetto di statistica e analisi dati analizzando le proprietà delle varie distribuzioni del dataset di input, crea un file .csv con il risultato da poter scaricare, dividi l'input in 5 parti da 4700 righe alla volta, ricombina poi i risultati in un unico file .csv da scaricare e se durante l'analisi incontri colonne binarie, mantieni queste proprietà, così come i valori interi nell'output

In seguito vengono applicate delle strategie di **prompt engineering** presenti nel cookbook di OpenAI che sono le seguenti, abbiamo reso la richiesta più specifica rispetto al contesto assegnando anche un ruolo al modello, dopodichè abbiamo dato delle linee guida suddividendo il compito di generazione in varie sotto-task dando anche le RQs come obiettivi da eseguire, infine abbiamo fornito degli esempi del dataset sul falsa riga del few-shot learning per migliorare la qualità del output levigando il contesto dell'input. Le seguenti strategie hanno portato ad un secondo prompt:

Sei un assistente utile alla generazione di dati sintetici. Dato il seguente dataset in formato csv, creare un nuovo dataset con la stessa struttura. Il nuovo set di dati dovrebbe:”, 1. Mantenere le proprietà statistiche (ad esempio, media, mediana, modalità, deviazione standard) del dataset originale”, 2. Introdurre leggere variazioni per distinguerlo”, 3. Mantenere il numero di righe e colonne”, 4. Far corrispondere i tipi di dati per ogni colonna (ad esempio, continui, discreti o categorici), 5. Conservare i valori numerici incontrati, ad esempio: se una colonna ha interi genera una colonna di interi, 6. Utilizza varie tecniche statistiche a scelta tua per migliorare il dataset in maniera tale da migliorare le performance di un classificatore KNN, 7. Tale classificatore associa tutte le colonne alla colonna label eseguendo una classificazione binaria, le metriche da migliorare sono: Accuracy, Recall, Precision, F1 Score, True positive rate, false positive rate. 8. Suddividi il dataset di input in parti uguali ognuna contenente 4700 righe e ricombina i risultati in un unico dataset, da poter scaricare 9. Conta il numero di osservazioni duplicate nel dataset di input e mantieni l'esatto numero nel dataset generato.

I risultati sono analizzati nei seguenti sottoparagrafi.

7.2.1 Analisi del dataset sintetico con primo prompt

Come primo passo abbiamo analizzato il primo dataset sintetico, ricavato tramite GPT-4o, utilizzando un prompt semplice e con poche specifiche nella richiesta. Sono stati eseguiti gli stessi passaggi e test effettuati sul dataset originario, in modo da confrontarli e vedere se ci fossero miglioramenti o peggioramenti delle feature. Nelle prossime sezioni analizzeremo il dataset ricavato dal primo prompt.

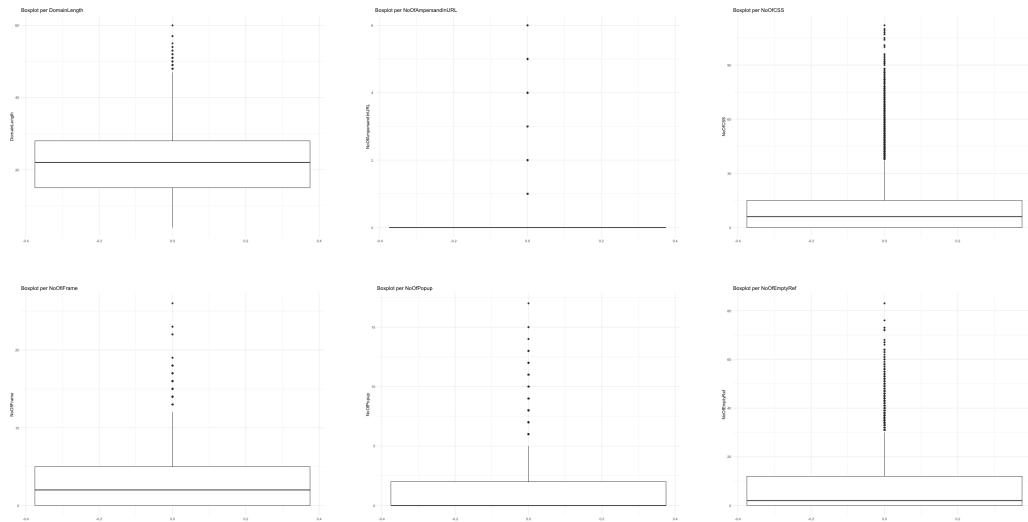
7.2.2 Statistica descrittiva univariata dataset sintetico Prompt1

Per prima cosa, abbiamo effettuato nuovamente della statistica descrittiva univariata, basandoci sempre su boxplot, boxplot ad intaglio, istogrammi, diagrammi di pareto e qqplot. Nelle sezioni successive mostreremo i risultati ottenuti sul dataset del primo prompt. Sono state utilizzate le stesse colonne sia per il dataset sintetico ottenuto dal primo prompt, sia per quello sintetico finale. Le colonne utilizzate sono le seguenti:

- **DomainLength**
- **NoOfAmpersandInURL**
- **NoOfCSS**
- **NoOfEmptyRef**
- **NoOfFrame**
- **NoOfPopup**

7.2.3 BoxPlot Dataset Sintetico Prompt1

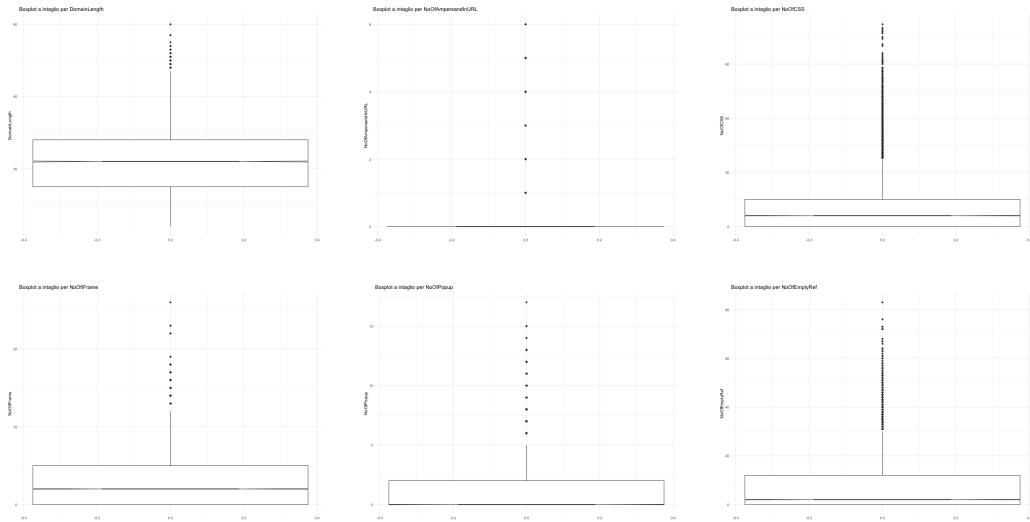
In questo capitolo mostreremo i risultati dei boxplot ottenuti sul dataset del primo prompt. Effettuando gli stessi passaggi abbiamo ottenuti i seguenti boxplot:



Da questi boxplot possiamo già notare che la maggior parte delle variabili analizzate presenta una asimmetria positiva, con una concentrazione di valori bassa ed una coda lunga verso l'alto.

7.2.4 BoxPlot ad intaglio Dataset Sintetico Prompt1

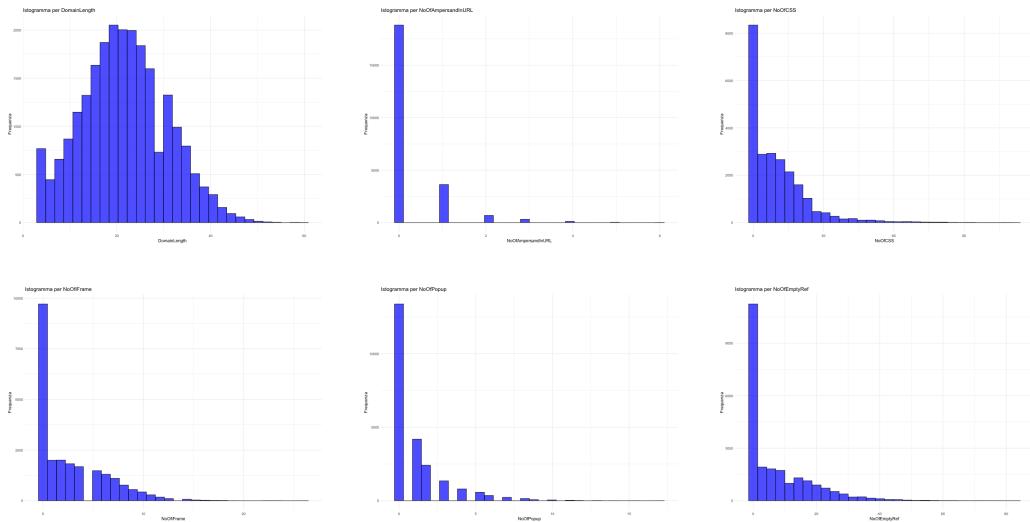
In questa sezione invece analizzeremo i risultati ottenuti nella rappresentazione di boxplot ad intaglio:



Dai grafici possiamo notare che alcune mostrano una distribuzione altamente concentrata come ad esempio la colonna **NoOfAmpersand**, mentre altre evidenziano una dispersione più ampia con mediane ben definite come **NoOfCSS**. Le feature **NoOfEmptyRef** e **NoOfCSS** presentano una densità di outlier maggiore rispetto alle altre.

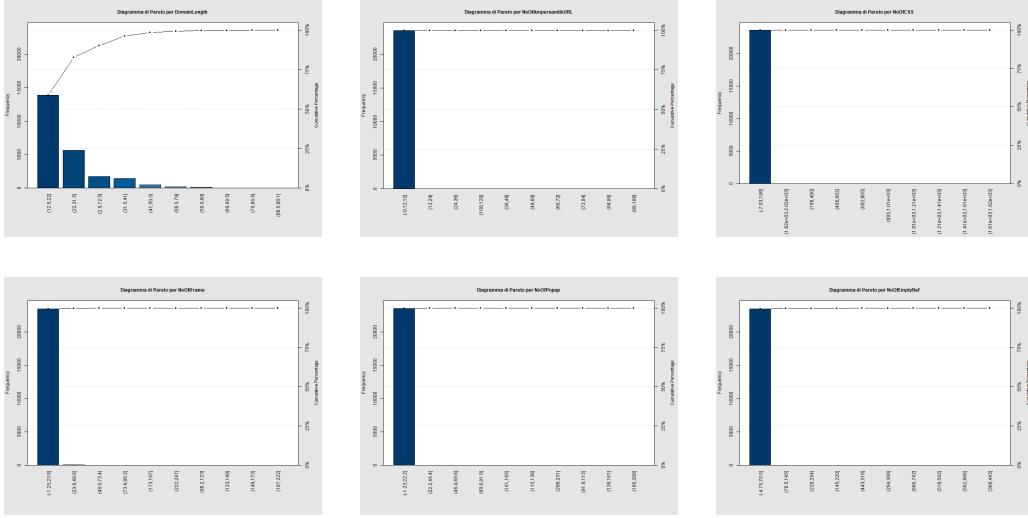
7.2.5 Istogrammi Dataset Sintetico Prompt1

Successivamente abbiamo ricavato gli istogrammi:



La maggior parte delle variabili segue una distribuzione asimmetrica a destra, così come il dataset originario tranne **DomainLength**, che "somiglia" di più ad una normale.

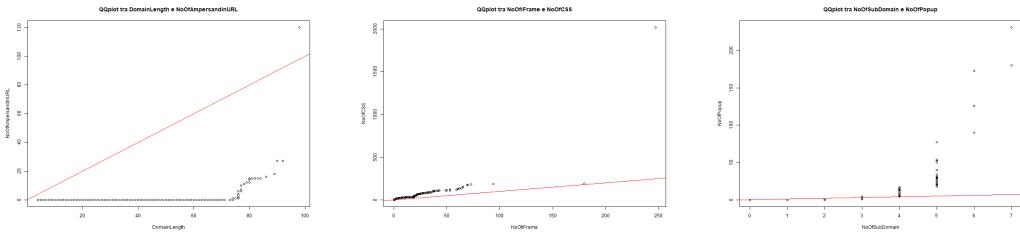
7.2.6 Diagrammi di Pareto Dataset sintetico prompt1



In quasi tutte le variabili analizzate, poche categorie spiegano la maggior parte del fenomeno, seguendo il principio di Pareto. L'unica di quelle mostrare è **DomainLength**.

7.2.7 QQPlot

In seguito riportiamo i qqplot del grafico sintetico generato dal primo prompt:



Nei primi due grafici possiamo notare che le distribuzioni mostrano forti squilibri, indicando anomalie e discrepanze nei dati. Il terzo grafico, invece, presenta una relazione tra le due variabili più stabile in alcuni punti, ma che tende a peggiorare con l'andamento della retta.

7.2.8 Model Performance Dataset sintetico prompt1

7.2.9 KNN

In seguito mostriamo i valori ottenuti effettuando il test del KNN sul dataset sintetico del prompt1:

Metriche	Valori
Accuracy	51%
Sensitivity	34%
Specificity	65%
Precision	44%
Recall	34%
F1-score	38%

Table 8: Risultati KNN dataset sintetico prompt1

Da questa prima analisi dei risultati del **KNN** possiamo notare una differenza sostanziale rispetto al dataset filtrato originario. Il valore della sensitivity, ad esempio, è molto bassa, suggerendo che il modello non è molto bravo a rilevare i positivi (problematica legata al modello sbilanciato, in quanto ricavato da un prompt impreciso). La precisione è anch'essa relativamente bassa (circa 43%), indicando che quando il modello predice il positivo, non è molto affidabile. Su quasi tutte le metriche analizzate, tranne **specificity**, il modello si comporta peggio di un predittore casuale.

7.2.10 Analisi distribuzione Dataset sintetico prompt1

In questo capitolo analizzeremo il dataset sintetico creato dal primo prompt semplice, mettendo in risalto i risultati ottenuti, ovvero: **media**, **moda**, **mediana**, **coefficiente di variazione**, **Skewness** e **Kurtosis**.

7.2.11 Media, moda, mediana

Di seguito riportiamo i risultati ottenuti dal calcolo di media, moda e mediana del dataset sintetico ottenuto dal primo prompt:

Osservazione	Media	Moda	Mediana
DomainLength	21.67	20	22.00
IsDomainIP	0	0	0
TLD	150.3	1	149.0
TLDLength	2.787	3	3.000
NoOfSubDomain	1.162	1	1.000
NoOfAmpersandInURL	0.02771	0	0.00000
HasTitle	0.857	2	1.0000
HasFavicon	0.3903	1	0.0000
Robots	0.3054	0	0.0000
NoOfURLRedirect	0.1415	0	0.0000
NoOfSelfRedirect	0.01022	1	0.00000
NoOfPopup	0.193	0	0.0000
NoOfFrame	2.786	0	2.000
HasExternalFormSubmit	0.0134	0	0.00000
HasSubmitButton	0.4245	0	0.0000
HasHiddenFields	0.3978	0	0.0000
HasPasswordField	0.09703	0	0.0000
Bank	0.135	0	0.0000
Pay	0.2741	0	0.0000
Crypto	0.001187	0	0.00000
NoOfCSS	9.766	0	6.000
NoOfEmptyRef	7.487	0	2.000

Table 9: Risultati di Media, Moda e Mediana

Da questi dati ricavati è possibile notare molte differenze con i valori di media, moda e mediana ricavati dal dataset originario filtrato. Per prima cosa possiamo notare una discrepanza sostanziale nei valori della moda, come ad es. le colonne di **TLD** e **HasTitle**; questa enorme discrepanza potrebbe derivare da un encoding dei dati differente o da una distribuzione dei dati poco coerente rispetto al dataset originale. Colonne come **NoOfEmptyRef** dimostra una media molto diversa, accentuando ancora di più le differenze tra i due dataset. Poche colonne hanno mantenuto una certa stabilità nel procedimento, come ad esempio **DomainLength** o **NoOfSubDomain**. Tutte queste differenze evidenziate sono state causate da un prompt impreciso e poco dettagliato, che hanno portato il modello a generare un dataset con dati poco coerenti tra di loro.

7.2.12 Coeff. di Variazioni del Dataset Sintetico del primo prompt

Di seguito riportiamo i valori del Coefficiente di Variazione per il dataset sintetico ricavato dal primo prompt:

Osservazione	CV
DomainLength	0.4%
TLD	0.5%
TLDLength	0.2%
NoOfSubDomain	0.6%
HasTitle	1.3%
HasFavicon	0.4%
Robots	1.2%
IsResponsive	1.5%
NoOfURLRedirect	1%
NoOfPopup	1.7%
NoOfFrame	1.2%
HasSubmitButton	1.2%
HasHiddenFields	1.2%
HasPasswordField	3.1%
Bank	2.5%
Pay	1.6%
NoOfImage	1.2%
NoOfCSS	1.3%
NoOfJS	1.7%
NoOfEmptyRef	1.4%

Table 10: Coefficienti di Variazione

Da questa tabella possiamo notare delle differenze sostanziali rispetto al dataset originario. Per prima cosa possiamo notare un cambiamento nella colonna **NoOfPopup** che ha avuto una riduzione percentuale di circa -12.3%, così come **NoOfEmptyRef** che è scesa di circa -5.1%, indicando una maggiore stabilità. Al contrario alcune variabili hanno ricevuto un leggero aumento, come ad esempio **HasTitle** o **HasPasswordField**. Da una prima analisi sembrerebbe che il CV sia inferiore per molte variabili, rappresentando un miglioramento nella qualità dei dati; per confermare ciò abbiamo generato nuovamente roc curve confusion matrix e roc curve:

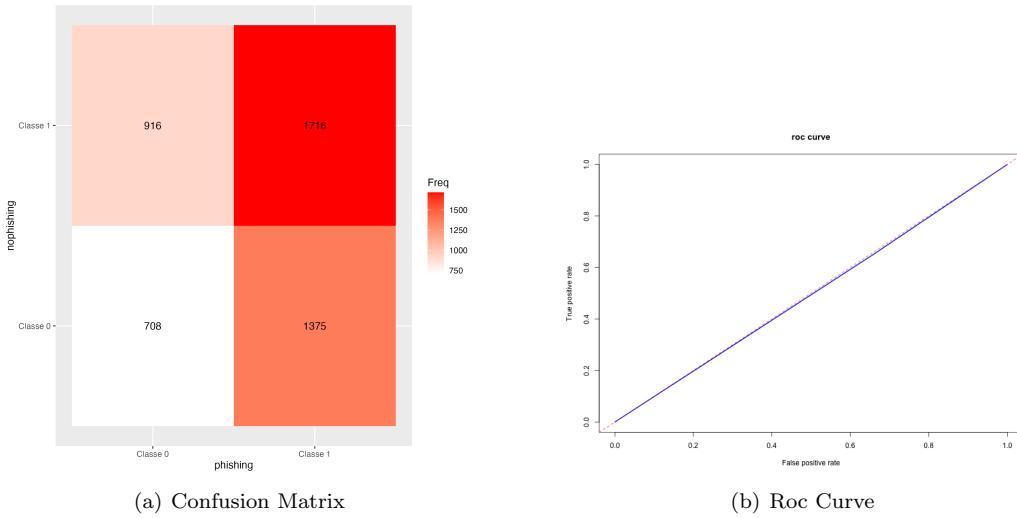


Figure 7: Due immagini affiancate

Dalla **Confusion Matrix**, abbiamo constatato che il modello ha una bassa accuratezza (circa 52%), indicando che le sue predizioni sono poco affidabili. Osservando la tabella, infatti, possiamo notare un elevato numero di falsi negativi (916), quest'ultimo evidenzia che il modello tende a classificare erroneamente siti di phishing come sicuri, aumentando il rischio di esposizione a minacce informatiche. Inoltre, il numero di falsi positivi (1375) è anch'esso elevato, che ci porta a presupporre che il modello identifica molti siti legittimi come pericolosi. L'analisi della **Roc Curve** rafforza le osservazioni precedentemente citate, in la curva si avvicina alla diagonale, suggerendo che il modello ha una capacità discriminante molto limitata, non riuscendo a distinguere efficacemente tra siti phishing e non-phishing.

7.2.13 Skewness e Kurtosis del Dataset Sintetico del primo prompt

In questa sezione invece, riporteremo i dati di Skewness e Kurtosis ricavati:

Osservazione	Skewness	Kurtosis
DomainLength	0.1562948	-0.2734391
TLD	0.162096	-0.4539411
TLDLength	0.2630501	-0.3855822
NoOfSubDomain	0.08895476	-0.174697
NoOfAmpersandInURL	3.138537	12.46491
LineOfCode	2.088554	6.303655
HasTitle	-2.039416	2.15931
HasFavicon	0.4495972	-1.797939
Robots	0.8448289	-1.286319
IsResponsive	-0.4272112	-1.817568
NoOfURLRedirect	2.056827	2.230631
NoOfSelfRedirect	9.738613	92.84452
NoOfPopup	2.381943	6.9813155
NoOfFrame	1.251188	1.26568
HasExternalFormSubmit	8.463139	69.62768
HasSubmitButton	0.3054326	-1.906792
HasHiddenFields	2.225804	2.954329
HasPasswordField	0.4177937	-1.825526
Bank	2.722583	5.412689
Pay	2.136253	2.563686
Crypto	1.012871	-0.9741335
NoOfImage	28.96614	837.0728
NoOfCSS	1.2228	1.156744
NoOfJS	2.234077	7.706833
NoOfEmptyRef	1.674033	3.119043

Table 11: Skewness e Kurtosis

Rispetto alla tabella ottenuta dal dataset filtrato, questa dimostra significativi cambiamenti nelle distribuzioni delle variabili, soprattutto sulle variabili di Skewness e Kurtosis. Ad esempio è possibile notare che:

- **DomainLength** ha avuto una riduzione nel valore di Skewness, da 2.46 a 0.15, e da 10.12 a -0.27 per la Kurtosis
- **NoOfPopup** che prima aveva una skewness pari a 52.36, ora è scesa a 2.38, così come la kurtosis.

7.2.14 Test del Chi Quadrato Dataset Prompt1

7.2.15 Test della normale su Dataset sintetico prompt1

Per prima cosa è stato effettuato, come per il dataset originario, il test della normale sui nuovi dati ricavati, ed il risultato ottenuto è stato il seguente:

Osservazione	Chi2	First	Last	NObs
DomainLength	221.0157	0.05063562	7.377759	5209 4512 4001 5208 4650
TLD	47.29474	0.05063562	7.377759	5059 4634 4560 4478 4849
TLDLength	27618.92	0.05063562	7.377759	7717 0 0 13227 2636
NoOfSubDomain	56322.86	0.05063562	7.377759	0 18819 9 9 4761
NoOfAmpersandInUrl	27463.44	0.05063562	7.377759	3240 0 13588 0 6752
NoOfPopup	22046.29	0.05063562	7.377759	0 13372 4180 2413 3615
NoOfFrame	15760.04	0.05063562	7.377759	0 11705 3827 3161 4887
NoOfCSS	16462.16	0.05063562	7.377759	0 11982 4215 4045 3338
NoOfEmptyRef	21007.24	0.05063562	7.377759	0 13151 3545 2713 4171

Table 12: Risultati Test normale su Dataset sintetico prompt1

Da questa tabella possiamo notare come i test del chi quadro siano molto simili a quelli del dataset originario; ci sono alcuni fallimenti durante l'esecuzione, dimostrando che non c'è stato un improvement da parte del dataset sintetico del primo prompt1. Analizzando i valori ottenuti siamo giunti alla conclusione che quasi nessuna delle colonne passasse il test della normale, come ad esempio **TLDLength**, in quanto il numero di osservazioni attese per l'intervallo non viene rispettato.

7.2.16 Test binomiale su Dataset sintetico prompt1

Nel test della binomiale effettuato sul dataset sintetico abbiamo notato dei cambiamenti rispetto al dataset originario. Nel dataset originario, come già specificato nelle sezioni precedenti mostrava un solo fallimento nella colonna **Crypto**, con una percentuale di confidenza pari a: 95%. I dati che abbiamo ottenuto effettuando il test binomiale sono i seguenti:

Osservazione	Chi2	First	Last	NObs
IsDomainIP	804.6949	0.0009820691	5.023886	0 23580
HasFavicon	0.01084553	0.0009820691	5.023886	14376 9204
NoOfURLRedirect	0.04491974	0.0009820691	5.023886	20243 3337
Bank	3.268503e-05	0.0009820691	5.023886	20397 3183
HasTitle	1.245781e-06	0.0009820691	5.023886	3372 20208
Robots	0.02040865	0.0009820691	5.023886	0.695 0.305
NoOfSelfRedirect	0.1158318	0.0009820691	5.023886	23339 241
HasExternalFormSubmit	0.2957862	0.0009820691	5.023886	23264 316
HasSubmitButton	0.02295066	0.0009820691	5.023886	13570 10010
HasHiddenFields	0. 006036738	0.0009820691	5.023886	14201 9379
HasPasswordField	0. 0002651308	0.0009820691	5.023886	0.903 0.097
Pay	0.0009223499	0.0009820691	5.023886	17117 6463
Crypto	0.0009223499	0.0009820691	5.023886	23552 28

Table 13: Risultati Test Binomiale Prompt1

Da questi valori ricavati, abbiamo analizzato i valori ottenuti, ed abbiamo notato che in questa casistica la colonna **Crypto** passava di poco il test della binomiale. Al contrario colonne come **IsDomainIP**, **Bank** e **HasTitle** che nel dataset filtrato originario passavano il test, ora invece falliscono.

7.2.17 Test Poisson su Dataset sintetico prompt1

Come ultimo step per il dataset generato dal primo prompt abbiamo rieffettuato il test di poisson sulle colonne che non avevano passato il test della normale, in modo da valutarne la popolazione di appartenenza. I valori ottenuti sono stati i seguenti:

Osservazione	Chi2	First	Last	NObs
DomainLength	7191763	0.2157953	9.348404	768 201 245 297 22069
TLD	2.010384e+64	0.2157953	9.348404	599 424 28 22497
TLDLength	22182.37	0.2157953	9.348404	7717 13227 2581 55 0
NoOfSubDomain	7764.135	0.2157953	9.348404	3240 13588 6441 309 0
NoOfAmpersandInUrl	1541.402	0.2157953	9.348404	18819 3626 679 312 32
NoOfPopup	8588.201	0.2157953	9.348404	0 13372 4180 2413 1346 1472
NoOfFrame	52628.46	0.2157953	9.348404	9711 1994 2005 1822 6368
NoOfCSS	43845571	0.2157953	9.348404	7698 642 634 748 131115
NoOfEmptyRef	8511105	0.2157953	9.348404	10611 620 627 644 10429

Table 14: Risultati Test Poisson su Dataset sintetico prompt1

Analogamente al test di Poisson effettuato per il primo dataset, abbiamo riscontrato nuovamente un fallimento su tutte le colonne del dataset sintetico, giungendo alla conclusione che non è possibile definire la popolazione di appartenenza di questi nuovi valori generati tramite il modello linguistico. Possiamo esprimere questi risultati con una confidenza pari al 95%.

7.3 Analisi del dataset sintetico finale

Dopo aver rappresentato i risultati ottenuti dal dataset sintetico ottenuto con il prompt meno efficace, ora mostreremo i risultati ottenuti con il dataset sintetico finale.

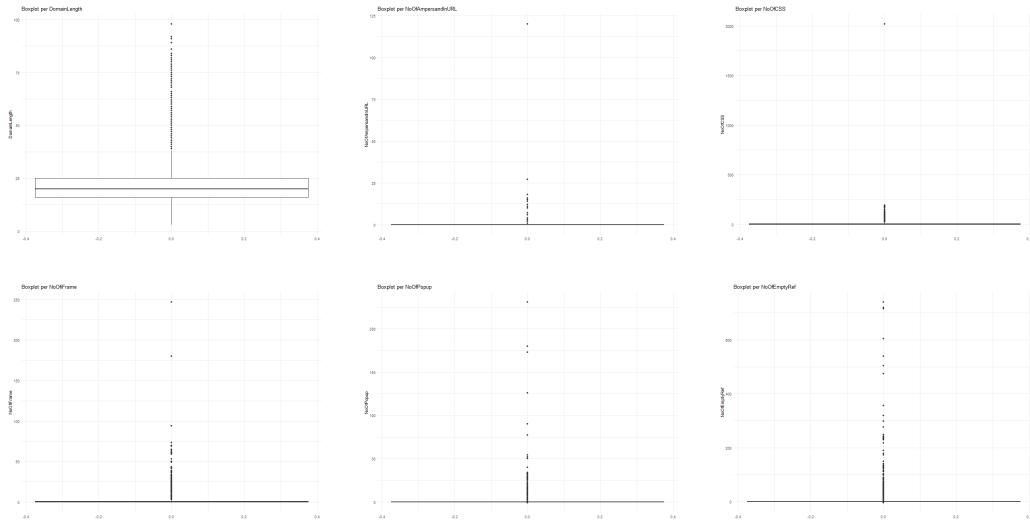
7.3.1 Statistica descrittiva univariata per Dataset sintetico finale

Per prima cosa, come per i dataset precedenti abbiamo applicato nuovamente un approccio alla statistica descrittiva univariata, basandoci sempre su boxplot, boxplot ad intaglio, istogrammi, diagrammi di pareto e qqplot. Le colonne che abbiamo preso come riferimento per le nostre analisi sono le seguenti:

- **DomainLength**
- **NoOfAmpersandInURL**
- **NoOfCSS**
- **NoOfEmptyRef**
- **NoOfFrame**
- **NoOfPopup**

7.3.2 BoxPlot

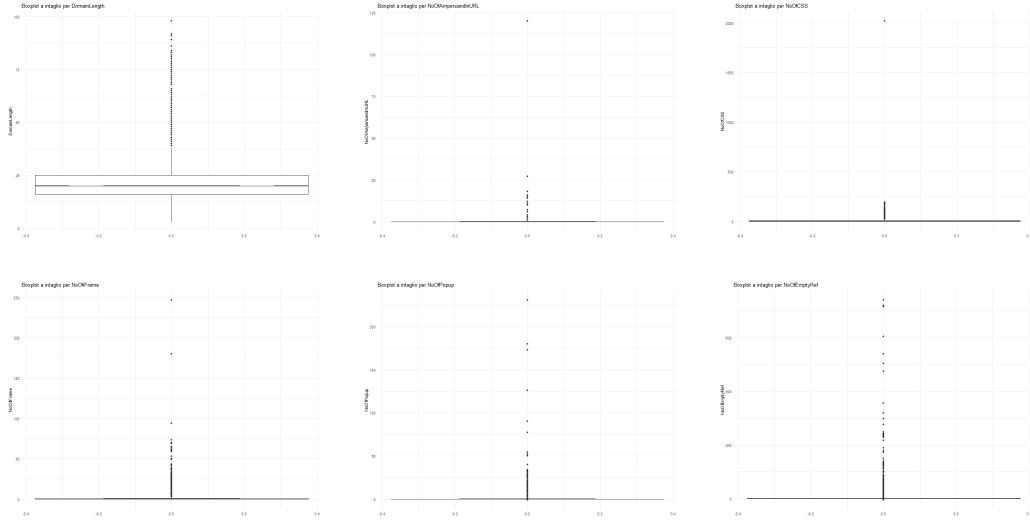
Riportiamo di seguito i boxplot ottenuti dal dataset sintetico finale:



Analizzando i sei boxplot ottenuti qui sopra siamo giunti alla conclusione che vi è una forte presenza di asimmetria nei dati, quest'ultimi concentrati su intervalli relativamente bassi, ma con la presenza di numerosi outlier significativi.

7.3.3 Boxplot ad Intaglio

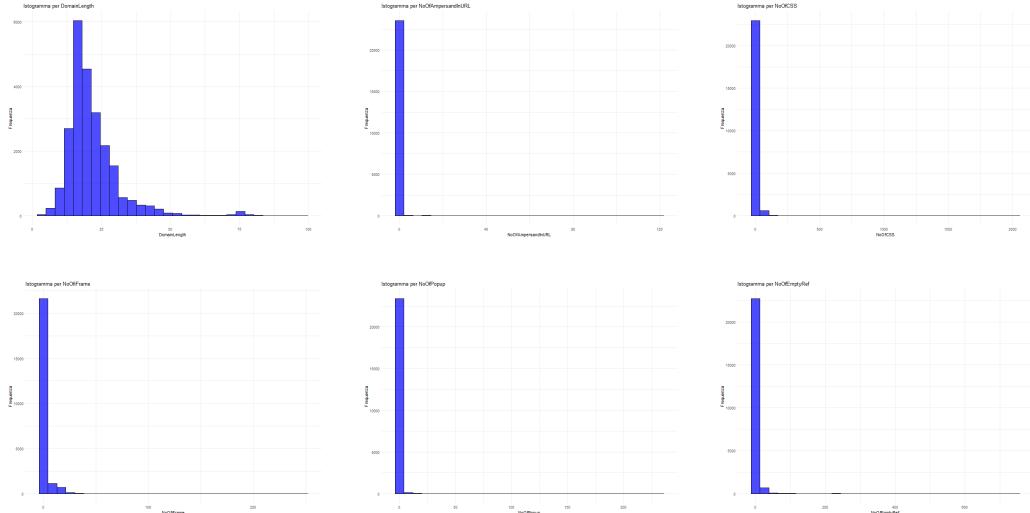
Di seguito riportiamo i boxplot ad intaglio ottenuti per il dataset sintetico finale:



I boxplot ad intaglio evidenziano una forte asimmetria nelle distribuzioni delle variabili analizzate. In tutti i grafici abbiamo osservato che la maggior parte dei dati è concentrata su valori relativamente bassi. Tranne la colonna **DomainLength** l'intaglio non è presente poiché il range interquartile risulta essere troppo basso o nullo.

7.3.4 Istogrammi

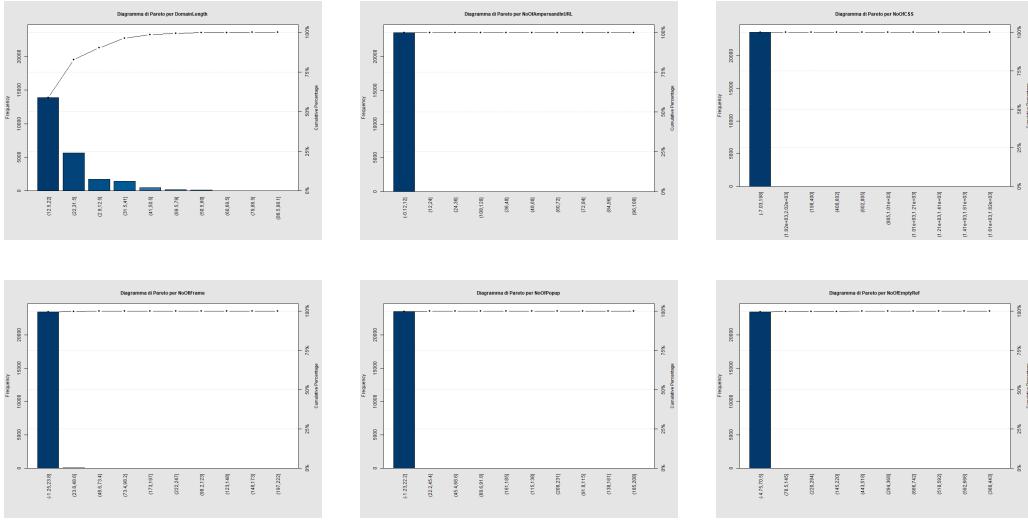
Di seguito riportiamo gli istogrammi ricavati dal dataset sintetico finale:



L'analisi degli istogrammi ha evidenziato un pattern ricorrente tra tutte le variabili: la distribuzione dei dati è fortemente sbilanciata, con la maggior parte delle pagine web che rientrano in valori contenuti, eccetto per qualche eccezione estrema che si distacca significativamente dalla norma.

7.3.5 Diagrammi di Pareto

Di seguito riportiamo i diagrammi di Pareto ottenuti sul dataset finale sintetico:

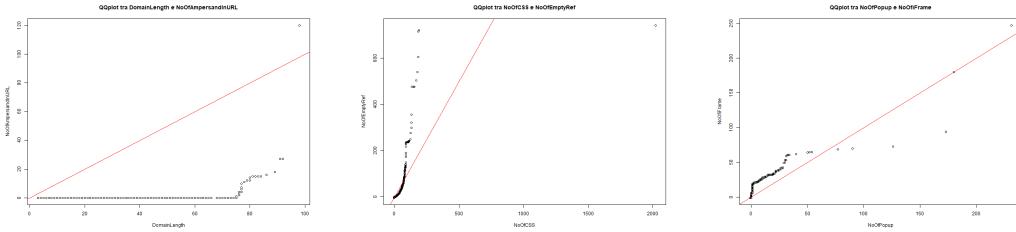


La maggior parte dei dati è concentrata in pochi valori iniziali, mentre una piccola percentuale di osservazioni si discosta nettamente dalla norma, creando una lunga coda di outlier. Da questi grafici siamo giunti alla conclusione che:

- I siti legittimi presentano un livello moderato per quanto riguarda tutte le variabili.
- I siti di phishing, invece, che mostrano valori estremamente elevati, in particolare nell'uso di Popup, CSS etc.

7.3.6 QQPlot

In seguito riportiamo i QQPlot ottenuti dal dataset sintetico completo:



I tre grafici QQPlot mostrano come le variabili esaminate non seguano una distribuzione normale, presentando forti asimmetrie e outlier significativi. Nella maggior parte dei casi le variabili analizzate mostrano una distribuzione regolare fino a un certo punto, per poi divergere fortemente.

7.3.7 Model Performance Dataset Sintetico Finale

In seguito mostriamo i valori ottenuti effettuando il test del KNN sul dataset sintetico finale:

Metriche	Valori
Accuracy	95%
Sensitivity	94%
Specificity	96%
Precision	94%
Recall	94%
F1-score	94%

Table 15: Risultati KNN

Possiamo notare che dal dataset sintetico finale sono stati ottenuti dei miglioramenti per quanto riguarda le metriche del KNN. Per prima cosa possiamo notare un aumento dell'**Accuracy** che ha subito un discreto aumento, evidenziando una migliore accuratezza del modello. Anche la **Precision**, la **Sensitivity** e la **F1-score** hanno subito un discreto aumento, al contrario della **Specificity** che invece ha subito un leggero calo rispetto al modello originale. Successivamente riportiamo i risultati ottenuti dalla confusion matrix e dalla roc curve:

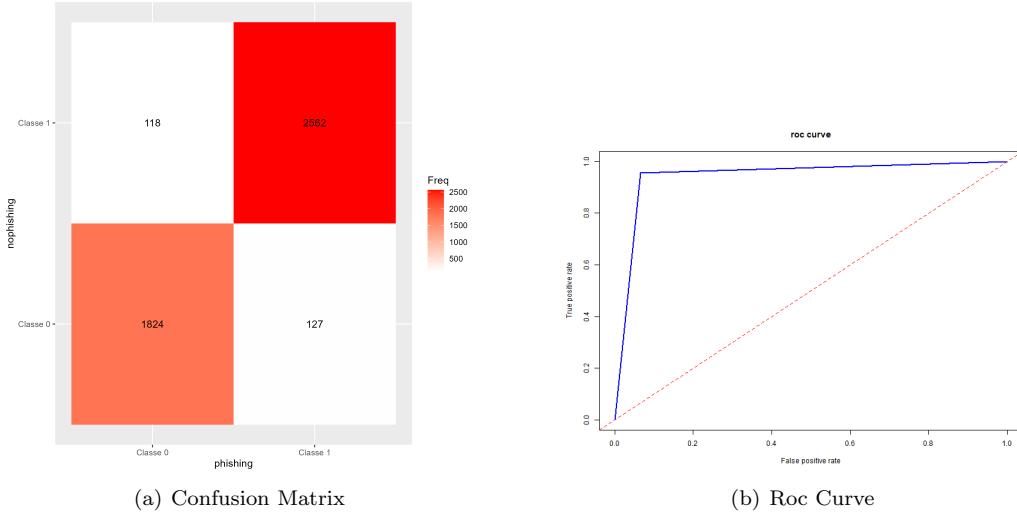


Figure 8: Due immagini affiancate

Dalla **Confusion Matrix** siamo giunti alla conclusione che il modello è in grado di identificare correttamente un elevato numero di siti sicuri, con un basso numero di falsi positivi. Tuttavia, abbiamo osservato un elevato numero di falsi negativi, suggerendo l'incapacità del modello nell'individuare perfettamente siti di phishing. Questo significa che molti siti malevoli vengono classificati erroneamente come sicuri. Al contrario, la **Roc Curve** dimostra che teoricamente il modello potrebbe separare bene le due classi se venisse applicata una soglia di decisione adeguata.

7.3.8 Analisi distribuzione Dataset sintetico finale

Nelle sezioni successive, mostreremo i risultati ottenuti sul dataset sintetico finale, analizzando i valori di media, moda, mediana, Coefficiente di Variazione, Skewness e Kurtosis.

7.3.9 Media, moda, mediana

I valori di media, moda e mediana ottenuti per il dataset sintetico finale sono i seguenti:

Osservazione	Media	Moda	Mediana
DomainLength	21.57	19	20.00
IsDomainIP	0.002417	0	0.000000
TLD	148.9	96	100.0
TLDLength	2.767	3	3.000
NoOfSubDomain	1.163	1	1.000
NoOfAmpersandInURL	0.02774	0	0.00000
HasTitle	0.8637	1	1.0000
HasFavicon	0.3581	0	0.0000
Robots	0.2681	0	0.0000
NoOfURLRedirect	0.1332	0	0.0000
NoOfSelfRedirect	0.04139	0	0.00000
NoOfPopup	0.1982	0	0.0000
NoOfFrame	1.596	0	0.000
HasExternalFormSubmit	0.04411	0	0.00000
HasSubmitButton	0.4133	0	0.0000
HasHiddenFields	0.3751	0	0.0000
HasPasswordField	0.1026	0	0.0000
Bank	0.1281	0	0.0000
Pay	0.2422	0	0.0000
Crypto	0.02511	0	0.00000
NoOfCSS	6.231	0	2.000
NoOfEmptyRef	2.454	0	0.000

Table 16: Risultati di Media, Moda e Mediana

Analizzando i valori ottenuti siamo giunti alla conclusione che i valori di media, moda e mediana non hanno riscontrato dei cambiamenti sostanziali, indicando come il dataset originario e quello sintetico presentino molte similitudini. Sono stati riscontrati dei piccoli cambiamenti per quanto riguarda **NoOfFrame** e **NoOfEmptyRef** che hanno subito un leggero aumento nella colonna della media.

7.3.10 Coeff. di Variazioni del Dataset Sintetico finale

Osservazione	CV
DomainLength	0.4%
IsDomainIP	20%
TLD	0.6%
TLDLength	0.2%
NoOfSubDomain	1.3%
HasTitle	0.4%
HasFavicon	1.3%
Robots	1.7%
NoOfURLRedirect	2.6%
NoOfPopup	14%
NoOfFrame	3%
HasSubmitButton	1.2%
HasHiddenFields	1.3%
HasPasswordField	3%
Bank	2.6%
Pay	1.8%
NoOfCSS	2.7%
NoOfEmptyRef	6.4%

Table 17: Coefficienti di Variazione

Una delle principali variazioni rispetto al dataset originario riguarda la significativa riduzione della variabilità in alcune feature. In particolare, NoOfPopup è passato da 14% a 1.7%, NoOfEmptyRef da 6.5% a 1.4%; ciò significa che queste variabili presentano valori più uniformi e meno dispersi. Anche **Robots** e **HasFavicon** hanno mostrato una riduzione sostanziale, dando come risultato un dataset più stabile.

7.3.11 Skewness e Kurtosis del Dataset Sintetico completo

Di seguito mostriamo i risultati ottenuti per il dataset sintetico finale per quanto riguarda i valori di Skewness e Kurtosis:

Osservazione	Skewness	Kurtosis
DomainLength	2.460146	10.07844
IsDomainIP	20.26412	408.6517
TLD	1.025693	-0.250869
TLDLength	1.706009	14.03742
NoOfSubDomain	1.793807	7.308409
NoOfAmpersandInURL	93.07487	11293.04
HasTitle	-2.119355	2.491773
HasFavicon	0.5920928	-1.649496
Robots	1.046908	-0.9040226
NoOfURLRedirect	2.158761	2.66036
NoOfSelfRedirect	4.60438	19.20113
NoOfPopup	52.33717	3488.422
NoOfFrame	12.03912	410.4492
HasExternalFormSubmit	4.44035	17.71746
HasSubmitButton	0.3520434	-1.876145
HasHiddenFields	0.5160846	-1.73373
HasPasswordField	2.618644	4.857502
Bank	2.225804	2.954329
Pay	1.203451	-0.5517298
Crypto	6.070597	34.85362
NoOfCSS	70.71078	8210.599
NoOfEmptyRef	25.97097	917.8675

Table 18: Risultati di Skewness e Kurtosis

Dalla tabella ricavata abbiamo notato che non ci sono stati cambiamenti sostanziali. Alcuni valori di skewness e kurtosis, però, rimangono un aspetto critico; quest'ultimi sono estremamente lontani dalla media che potrebbero influenzare le prestazioni di eventuali modelli predittivi. Nel complesso, però, abbiamo osservato che il dataset resta quasi del tutto invariato, con leggere modifiche.

7.3.12 Test del Chi Quadrato Dataset Sintetico Completo

Nelle prossime sezioni mostreremo i risultati ottenuti, effettuando il test del chi quadrato sulle seguenti popolazioni: **normale**, **binomiale**, **Poisson**.

7.3.13 Test della normale

Per prima cosa è stato effettuato, come per il dataset originario, il test della normale sui nuovi dati ricavati, ed il risultato ottenuto è stato il seguente:

Osservazione	Chi2	First	Last	NObs
DomainLength	5323.86	0.05063562	7.377759	2603 8725 5045 3908 2874
TLD	22043.47	0.05063562	7.377759	1676 12972 1005 1620 5882
TLDLength	37762.1	0.05063562	7.377759	6845 0 0 15305 1005
NoOfSubDomain	47516.96	0.05063562	7.377759	1361 17528 0 0 4266
NoOfAmpersandInUrl	91763.19	0.05063562	7.377759	0 0 23069 0 86
NoOfPopup	78417.35	0.05063562	7.377759	0 73 21658 1095 329
NoOfFrame	31536.74	0.05063562	7.377759	0 15025 4619 1874 1637
NoOfCSS	17135.11	0.05063562	7.377759	0 9913 8855 2569 1818
NoOfEmptyRef	70214.49	0.05063562	7.377759	0 520 20740 1248 647

Table 19: Risultati Test normale su Dataset sintetico finale

Giudicando i valori ottenuti, siamo giunti alla conclusione che nessuno dei dati presenti nella tabella supera il test della normale, in quanto presenta valori troppo elevati, indicando che i valori si discostano eccessivamente dalla normalità; successivamente andremo ad analizzare le colonne binarie applicando il test della binomiale, ed infine eseguiremo il test di Poisson sui dati qui sopra riportati per vedere un eventuale riscontro positivo.

7.3.14 Test binomiale

I valori ottenuti effettuando il test della binomiale sulle colonne binarie sono stati i seguenti:

Osservazione	Chi2	First	Last	NObs
IsDomainIP	2.472578	0.0009820691	5.023886	23098 57
HasFavicon	0.01084553	0.0009820691	5.023886	14376 9204
NoOfURLRedirect	0.04491974	0.0009820691	5.023886	20243 3337
Bank	3.268503e-05	0.0009820691	5.023886	20397 3183
HasTitle	1.245781e-06	0.0009820691	5.023886	3372 20208
Robots	0.02040865	0.0009820691	5.023886	16378 7202
NoOfSelfRedirect	0.1158318	0.0009820691	5.023886	23339 241
HasExternalFormSubmit	0.2957862	0.0009820691	5.023886	23264 316
HasSubmitButton	0.02295066	0.0009820691	5.023886	13570 10010
HasHiddenFields	0.006036738	0.0009820691	5.023886	14201 9379
HasPasswordField	0.0002651308	0.0009820691	5.023886	21292 2288
Pay	0.0009223499	0.0009820691	5.023886	17117 6463
Crypto	0.829345	0.0009820691	5.023886	23552 28

Table 20: Risultati Test Binomiale Dataset sintetico finale

I valori del chi quadrato sono estremamente bassi quasi per tutte le variabili, il che significa che i dati non si discostano in modo significativo dalla distribuzione binomiale. Ne fanno eccezione le osservazioni **HasTitle** e **Bank**.

7.3.15 Test di Poisson

Osservazione	Chi2	First	Last	NObs
DomainLength	10243.19	0.2157953	9.348404	6 22 24 43 23060
TLD	Inf	0.2157953	9.348404	1 1 1 2 23150
TLDLength	31026.62	0.2157953	9.348404	6845 15305 789 108 108
NoOfSubDomain	16409.56	0.2157953	9.348404	1361 17528 3579 495 192
NoOfAmpersandInUrl	2940076	0.2157953	9.348404	23069 21 14 9 42
NoOfPopup	Inf	0.2157953	9.348404	73 21658 880 215 329
NoOfFrame	Inf	0.2157953	9.348404	262 14763 2603 2016 3511
NoOfCSS	Inf	0.2157953	9.348404	4 16 91 231 22813
NoOfEmptyRef	Inf	0.2157953	9.348404	2 32 486 3489 19146

Table 21: Risultati Test Poisson su Dataset sintetico finale

Tutti i valori di Chi² sono estremamente alti o infiniti (Inf), il che significa che nessuna delle variabili segue una distribuzione di Poisson. Le deviazioni elevate indicano che i dati hanno una varianza molto più grande rispetto alla media, il che è incompatibile con una distribuzione di Poisson, che invece assume che la media e la varianza siano uguali.

8 Conclusioni e sviluppi futuri

In questo progetto abbiamo utilizzato varie metodologie statistiche, di analisi dati e di machine learning per generare un classificatore dei dati sintetici che migliorassero il dataset originale, inoltre rispondiamo qui alle research question della sezione sette.

- **ARQ1:** confrontando le tabelle 15, 4 e 8 affermiamo che la tabella del prompt1 non porta miglioramenti al modello, mentre dopo la fase di prompt engineering eseguita nella sezione 7, la tabella risultante migliora il modello rispetto ai dati originali. Per quanto riguarda le metriche numeriche e grafiche prese in considerazione, nonostante sia presente un forte sbilanciamento nelle etichette legato alla poca presenza della classe NoPhishing rispetto al Phishing, il dataset sintetico presenta una lieve differenza legata alla non presenza di osservazioni duplicate. Per cui il prompt finale permette la creazione di un dataset migliore di quello originariamente filtrato.
- **ARQ2:** Tramite il test del chi quadro affermiamo che non tutte le feature del dataset generato sono state associate ad una distribuzione nota, tale affermazione vale sia per il dataset generato col prompt1, sia quello del prompt finale, sia quello di originariamente filtrato. Abbiamo confrontato con le seguenti distribuzioni: binomiale, normale e Poisson. Osservando le seguenti tabelle 6, 13, 20, 5, 12, 19, 7, 14 e 21 affermiamo che nessuna feature è associata alla distribuzione normale e Poisson, mentre invece abbiamo le seguenti colonne che non sono binomiali: per il **dataset filtrato** abbiamo solo la feature **Crypto**. Per il **prompt1** abbiamo le feature **IsDomainIP**, **Bank** e **HasTitle**. Infine, per il **dataset generato col prompt finale** abbiamo le feature **Bank** e **HasTitle**.
- **ARQ3:** Osservando le seguenti tabelle 2, 10 e 17 abbiamo che la tabella del prompt finale che ha l'instabilità più alta si dimostra la più efficienze nel contesto del clustering

supervisionato relativo al task di classificazione binaria, ciò dimostra che un minimo di instabilità produce dataset di qualità superiore.

Detto ciò aggiungiamo dei suggerimenti per degli sviluppi futuri mirati a migliorare il riconoscimento del phishing:

- Implementare tecniche per minimizzare il false negative rate e migliorare ulteriormente la roc curve, aggiungendo anche l'auc score.
- Sviluppare altri modelli di machine learning come support vector machine, gradient boosting machine e neural network ma poter confrontare con il knn.
- Effettuare tecniche di bilanciamento manuale sulla colonna label per avere una distribuzione più equa delle etichette.
- Utilizzare la PCA effettuando una fase di dimensionally reduction come filtraggio al posto della media per varianza rivelatosi il filtraggio meno efficiente.
- Utilizzare ulteriori llm come gemma2, llama3, phi4, nemotron e qwen2 per verificare la generazione del dataset sintetico.
- Generare un nuovo dataset tramite operatori di combinazione che utilizzino il dataset sintetico e quello filtrato, come ad esempio il prodotto vettoriale valutando in seguito la qualità del risultato.

1 Librerie R utilizzate

- **ggplot2**: utilizzata per la rappresentazione grafica
- **crayon**: utilizzata per "colorare" i risultati nei plot
- **dplyr**: utilizzata per la selezione di colonne nel filtraggio
- **reshape2**: utilizzata per la conversione di un oggetto in dataframe
- **furrr**: utilizzata per consentire l'esecuzione in parallelo
- **polycor**: utilizzata per calcolare le correlazioni in modo misto tra variabili numeriche, ordinali e/o categoriali
- **data.table**: utilizzata per una lettura ottimizzata del dataframe
- **psych**: utilizzata per la creazione di matrici di correlazioni
- **moments**: utilizzata per il calcolo dei valori di Skewness e Kursosis
- **tidyverse**: include ggplot2 ed altre librerie per la rappresentazione grafica
- **DescTools**: utilizzata per il calcolo di media, moda e mediana ed il calcolo del coefficiente di variazione.
- **iml**: utilizzata per il calcolo dei valori di Shapley
- **caret**: Utilizzata per eseguire pre-processing dei dati, addestramento e visualizzazione della confusion matrix per i risultati sui dati di test.
- **glmnet**: Utilizzata per il calcolo dei Shap Values e Kursosis
- **ROCR**: utilizzata per la generazione della ROC Curve
- **httr**: utilizzata per facilitare l'interazione con API web e per l'invio e la gestione delle richieste HTTP.
- **jsonlite**: utilizzata per lavorare con i dati in formato JSON.
- **tidyr**: utilizzata nella statistica descrittiva, per riorganizzare i dati in un formato che ne facilitasse l'analisi
- **gridExtra**: utilizzata per la gestione di layout grafici. Consente di combinare più grafici in un'unica griglia
- **qcc**: utilizzata per costruire i diagrammi di Pareto. È' una forma di visualizzazione utile la distribuzione delle frequenze di un determinato attributo e per applicazioni legate al controllo qualità.