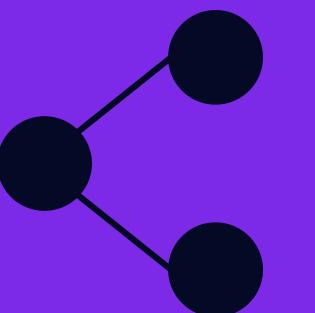
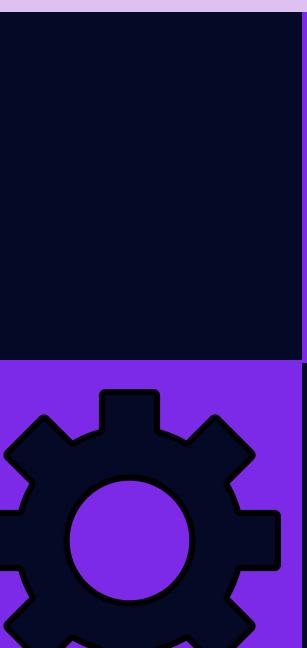
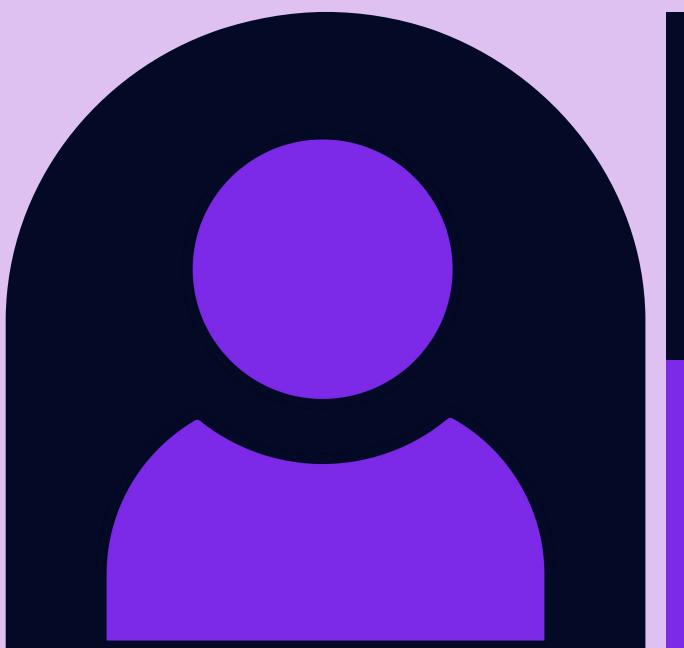


Giuseppe Napolitano: 0522501961

Nicola Pagliara: 0522501413

Analisi dati: Phishing4



Indice dei contenuti



Analisi del dataset

- Modulo filtraggio
- Modulo analisi distribuzione
- Modulo KNN
- Modulo test statistici

Uso LLM per la generazione dati

- Large Language Model (LLM):
Research Question
- LLM: Generazione del Dataset
- Analisi dataset sintetico
- Conclusioni e sviluppi futuri

Introduzione e contesto (1/2)



Definizione del Phishing:

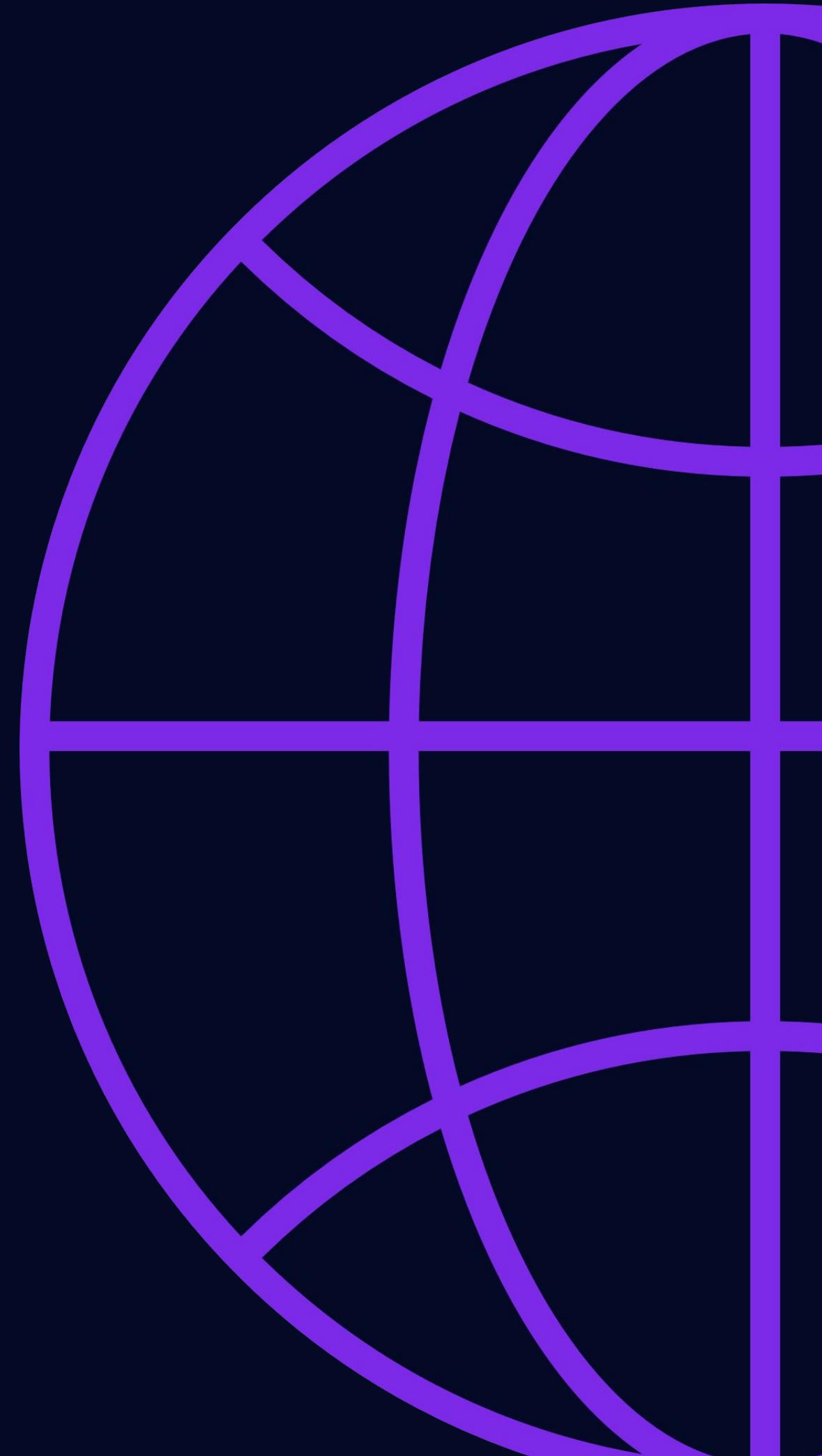
Il **Phishing** è una particolare tipologia di truffa realizzata sulla rete che sfrutta collegamenti (link) che rimandano a siti web fasulli che ingannano l'utente, rubando i suoi dati personali. L'obiettivo principale è quello di distinguere un sito reale da uno fasullo, rendendo la navigazione sul web sicura per gli utenti.



Sintesi del paper: 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. Invece modelli di machine learning, ovvero Random Forest (RF), Decision Tree (DT), LightGBM, Logistic Regression (LR) e Support Vector Machine (SVM)



Spiegazione dataset phishing4: Il dataset analizzato, Phishing4, è stato costruito ricavando i dati di siti legittimi ottenuti dall'**Open PageRank Initiative** e i dati illegittimi ottenuti dai database di sicurezza informatica quali **PhishTank**, **OpenPhish** e **MalwareWorld**.



Introduzione e contesto (2/2)

Il dataset Phishing4 presenta vari dati che sono risultati fondamentali per ricavare informazioni che ci consentissero di effettuare un miglioramento nella ricerca e l'individuazione di siti illegittimi. Dal paper scientifico, ad esempio, abbiamo ricavato alcune delle seguenti informazioni:

- Un numero elevato di **Popup** o **CSS** potrebbe essere un campanello d'allarme; l'eccessiva presenza di questi elementi potrebbe significare che il sito non è legittimo. I criminali informatici utilizzano una grande quantità di popup e css per rubare informazioni importanti dagli utenti, nascondendo link che rubano dati sensibili all'utente.
- Un'elevata quantità di codice **javascript** potrebbe essere un altro campanello d'allarme, poichè quest'ultimo può tramite click o immagini rubare informazioni all'utente, inconsapevole di aver esposto le proprie informazioni personali.
- Elevata presenza di caratteri speciali come "?" "!" sono un altro campanello d'allarme per l'individuazione di siti illegittimi.

Con l'attenta analisi delle informazioni qui sopra riportate, e le altre ottenute tramite la lettura del paper abbiamo iniziato le analisi statistiche sul dataset a disposizione, ottenendo una grande quantità di risultati che analizzeremo con attenzione successivamente.

Modulo 4 Filtraggio



Dopo l'analisi e la presentazione del dataset originario questo modulo si occupa di filtrare le caratteristiche che vengono definite come non rilevati o ridondanti. Qui presentiamo i filtri utilizzati:

- Filtraggio paper
- Filtraggio varianza
- Filtraggio correlazione
- Filtraggio explainable AI

Filtraggio Paper e Varianza

Filtraggio Paper

Come primo step è stata effettuata la lettura del paper scientifico, analizzando i dati e le loro caratteristiche. Dal paper abbiamo notato che quattro osservazioni erano state ricavate tramite dei calcoli a noi sconosciuti. Dopo un'attenta analisi abbiamo deciso di eliminare le seguenti colonne: *CharContinuationRate*, *URLTitleMatchScore*, *URLCharProb* e *TLDLegitimateProb*

Filtraggio per Varianza

Come seconda analisi, invece, abbiamo deciso di effettuare un calcolo della varianza, quest'ultima utilizzata come threshold per filtrare le colonne rimanenti. Come primo step abbiamo trasformato tutte le colonne in valori numerici tramite lo script seguente:

- `filtered_dataset <- filtered_dataset %>%
mutate(across(where(is.factor), as.numeric)).`

Dopodichè abbiamo calcolato la media delle varianze, dividendola per due, in modo da ottenere un valore di threshold

- `variances <- apply(numeric_cols, 2, var, na.rm = TRUE)
mean_variance <- mean(variances, na.rm = TRUE)`
- `threshold_variance <- mean_variance / 2`

In questo modo abbiamo filtrato le colonne che fossero minori del valore ottenuto ed abbiamo eliminato la colonna: *LargestLineLength*.

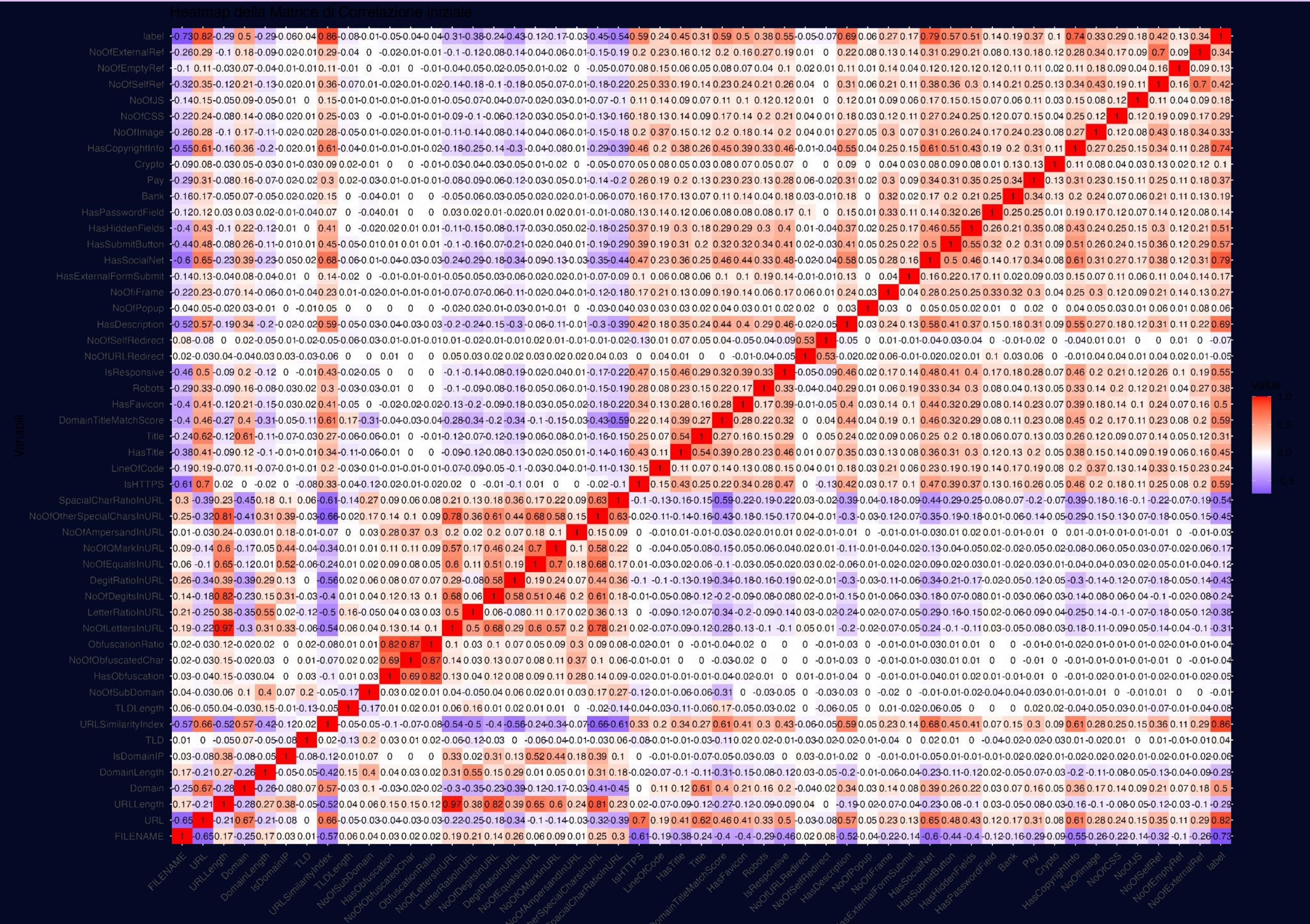
Filtraggio per correlazione (1/3)

Filtraggio per correlazione

Successivamente abbiamo creato una matrice di correlazione che ci consentisse di filtrare e ridurre ancora di più il nostro dataset. Abbiamo calcolato l'indice di correlazione per ogni coppia di colonne presente nel dataset; l'indice di correlazione di due variabili aleatorie X e Y indica quanto quest'ultime si influenzano a vicenda (positivamente, negativamente o completamente slegate tra di loro). Qui sotto riportiamo il codice in R per ricavare la matrice di correlazione, che andremo a visualizzare nella slide successiva:

```
# Funzione per calcolare la matrice di correlazione mista
compute_correlation_matrix <- function(data) {
  cor_matrix <- matrix(NA, ncol(data), ncol(data))
  problematic_pairs <- list()
  for (i in 1:ncol(data)) {
    for (j in i:ncol(data)) {
      tryCatch({
        if (is.numeric(data[[i]]) && is.numeric(data[[j]])) {
          cor_matrix[i, j] <- cor(data[[i]], data[[j]], use = "pairwise.complete.obs")
        } else {
          cor_matrix[i, j] <- mixedCor(data[, c(i, j)], use = "pairwise.complete.obs")$rho[1, 2]
        }
        cor_matrix[j, i] <- cor_matrix[i, j]
      }, error = function(e) {
        message(paste("Could not compute correlation between variables", colnames(data)[i], "and", colnames(data)[j], ":", e$message))
        problematic_pairs <- append(problematic_pairs, list(c(colnames(data)[i], colnames(data)[j])))
        cor_matrix[i, j] <- NA
        cor_matrix[j, i] <- NA
      })
    }
  }
  if (length(problematic_pairs) > 0) {
    message("Problematic variable pairs: ", paste(sapply(problematic_pairs, paste, collapse = " and "), collapse = "; "))
  }
  return(cor_matrix)
}
```

Filtraggio per correlazione (2/3)

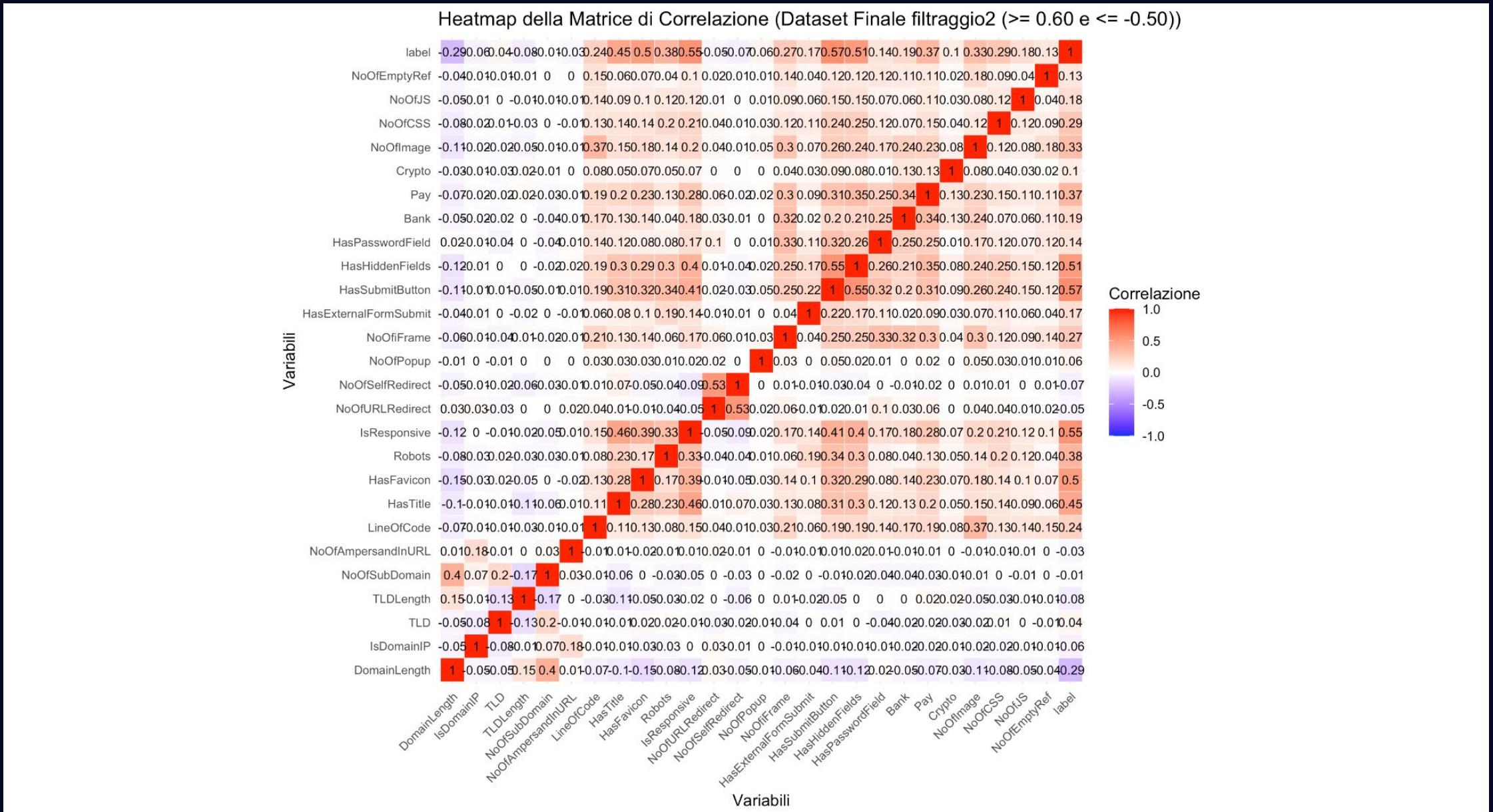


Filtraggio per correlazione

Le colonne che presentavano un valore positivo \geq di 0.60 sono state eliminate.

Le colonne che presentavano
un valore negativo ≤ -0.50
sono state eliminate

Filtraggio per correlazione 3/3

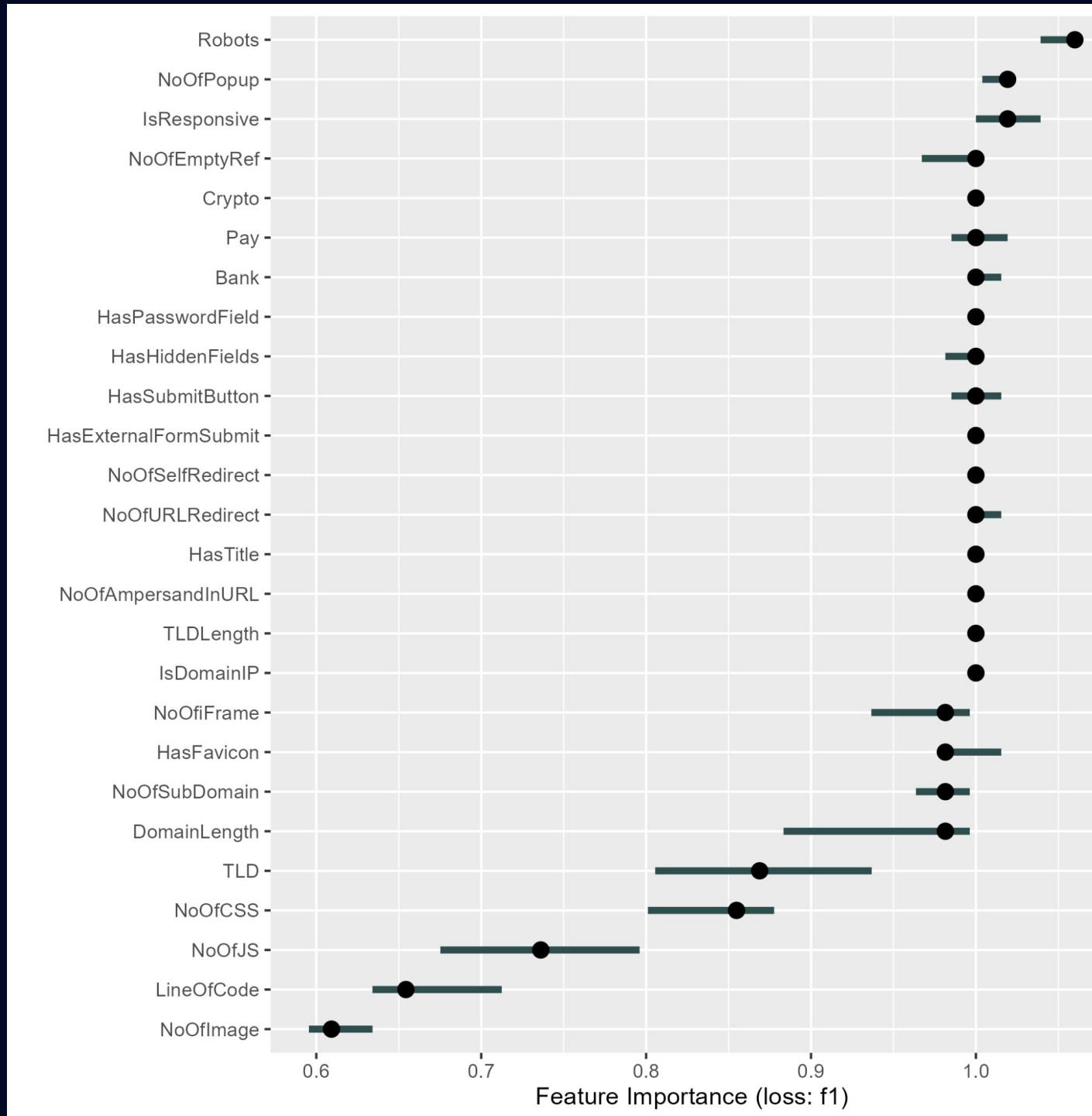


Filtraggio per correlazione

Le colonne eliminate sono state le seguenti

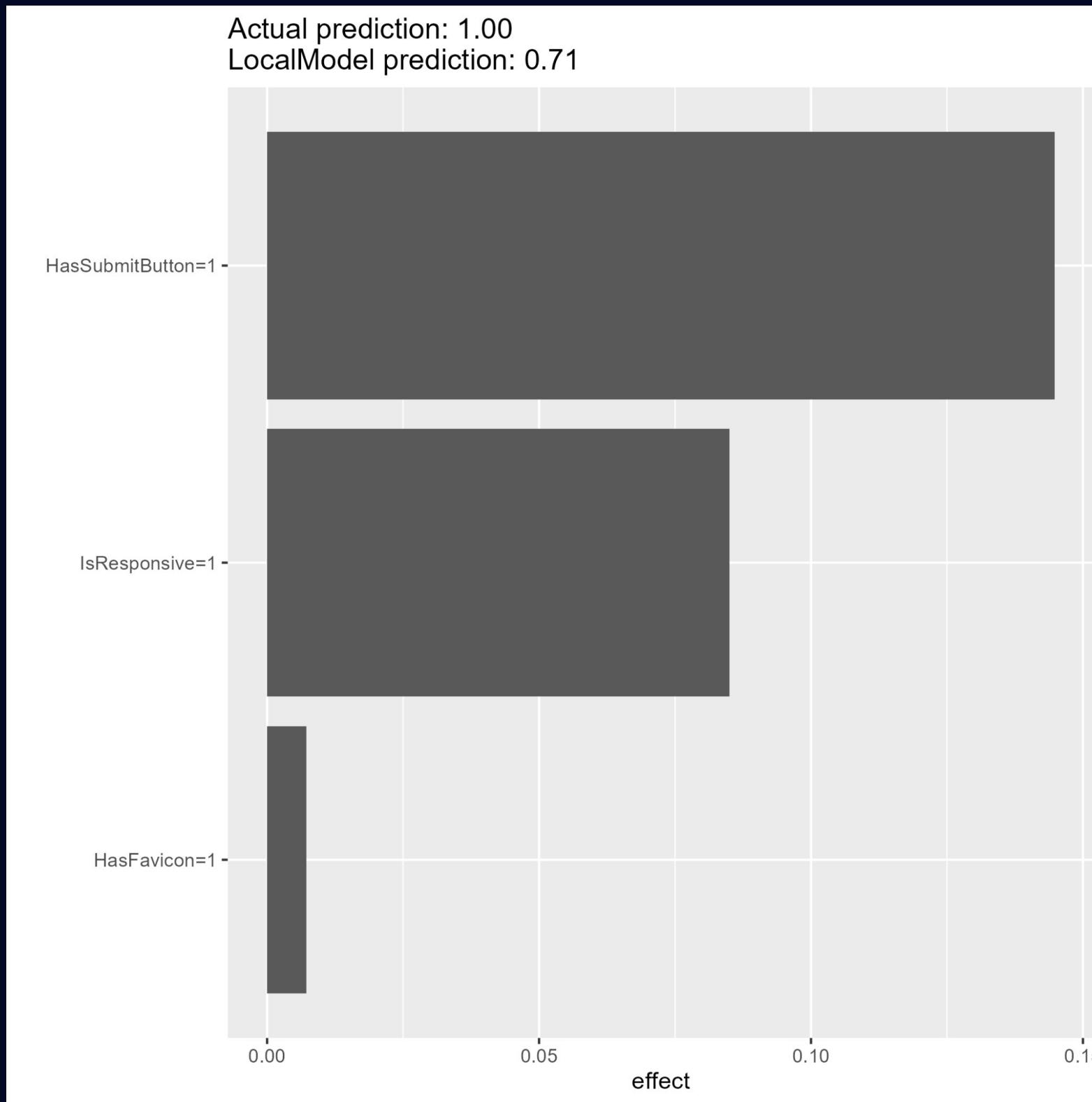
DigitRatioInURL, Domain, DomainTitleMatchScore, FILENAME, HasCopyrightInfo, HasDescription, HasObfuscation, HasSocialNet, IsHTTPS, LetterRatioInURL, NoOfDigitsInURL, NoOfEqualsInURL, NoOfExternalRef, NoOfLettersInURL, NoOfObfuscatedChar, NoOfOtherSpecialCharsInURL, NoOfQMarkInURL, NoOfSelfRef, ObfuscationRatio, SpacialCharRatioInURL, Title, URL, URLLength, URLSimilarityIndex

Filtraggio per Explainable AI (1/3)



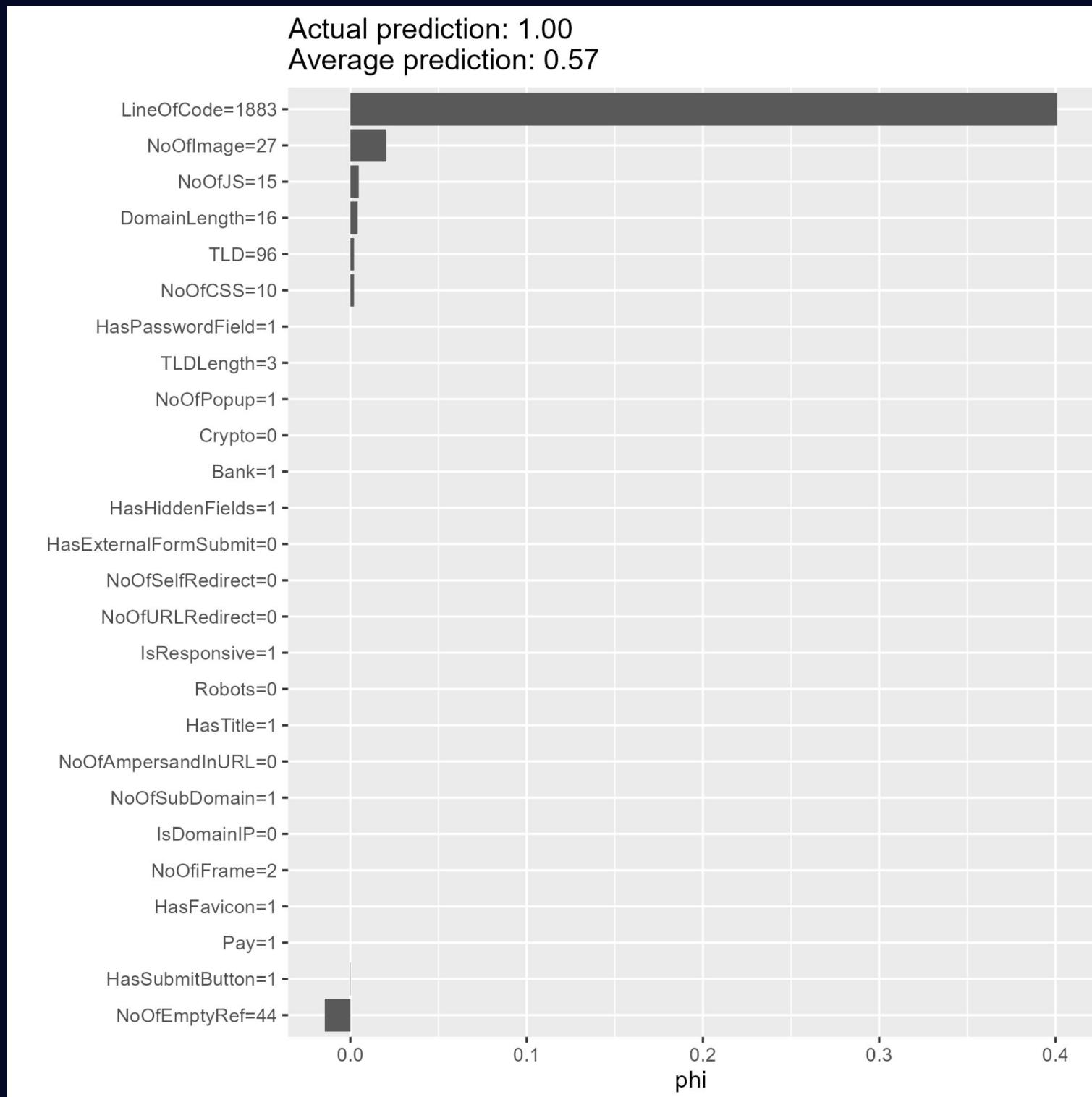
L'Explainable AI viene utilizzata per rendere comprensibile agli umani le decisioni prese dai modelli di Intelligenza Artificiale. Esistono delle tecniche globali e locali, le prime valutano tutte le previsioni in generale (globale), mentre le seconde si basano sulle singole predizioni. Nella tabella a sinistra valutiamo i risultati della f1: 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. Abbiamo valutato tutti i valori avente f1 minore di 0.8. Le colonne candidate per questa tecnica sono: NoOfJS, LineOfCode e NoOfImage.

Filtraggio per Explainable AI (2/3)



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. 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.

Filtraggio per Explainable AI (3/3)



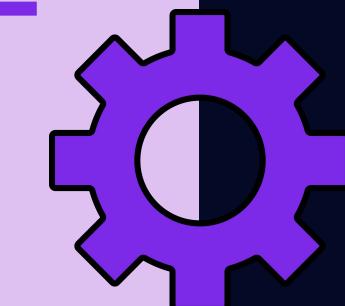
Discussiamo ora un altro approccio locale, i valori di Shapley. Una previsione può essere interpretata considerando ciascun valore caratteristico di un'istanza come un "giocatore" in un gioco in cui il risultato è la previsione stessa. I valori di Shapley, presi dalla teoria dei giochi coalizionali, 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.



Modulo analisi distribuzioni

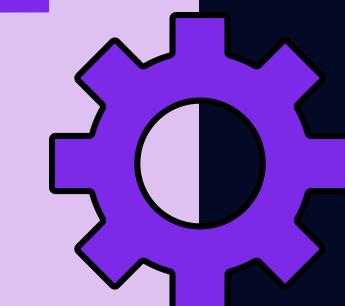
Analisi Indice di sintesi

- Analisi effettuata sul dataset filtrato per ricavare i dati di media, moda, mediana



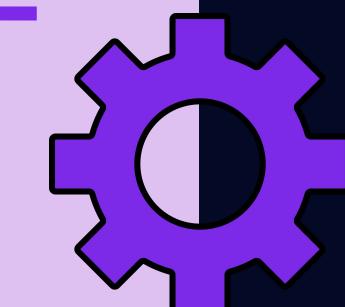
Statistica descrittiva

- Rappresentazione grafica dei valori presenti nel dataset filtrato

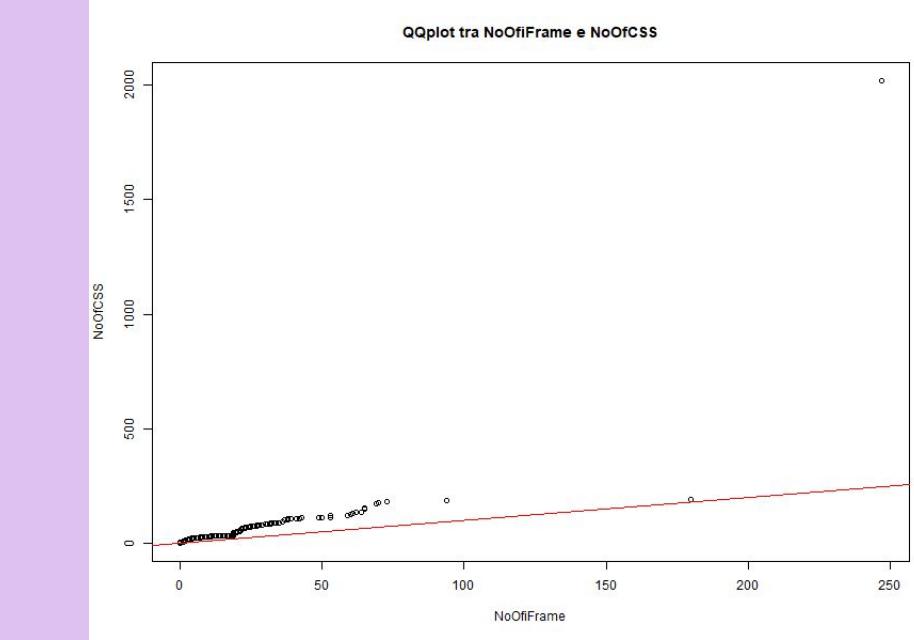
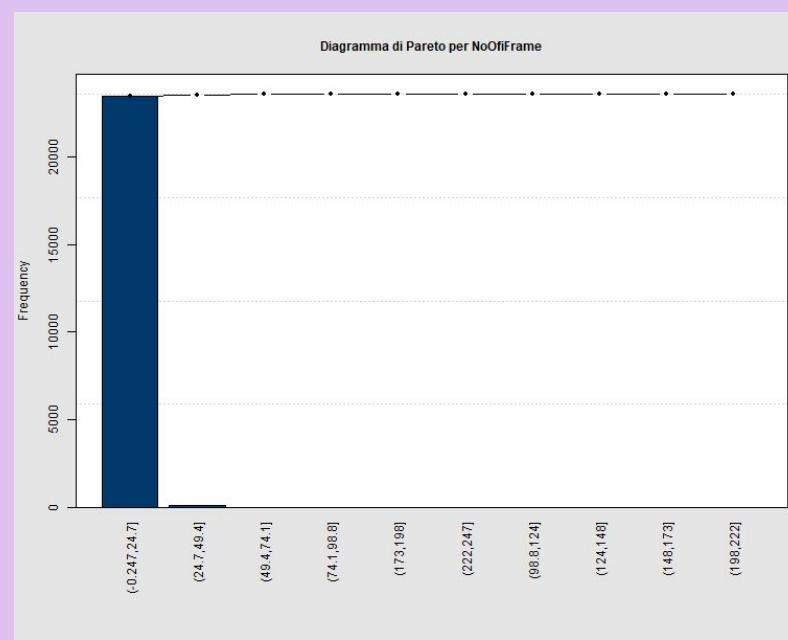
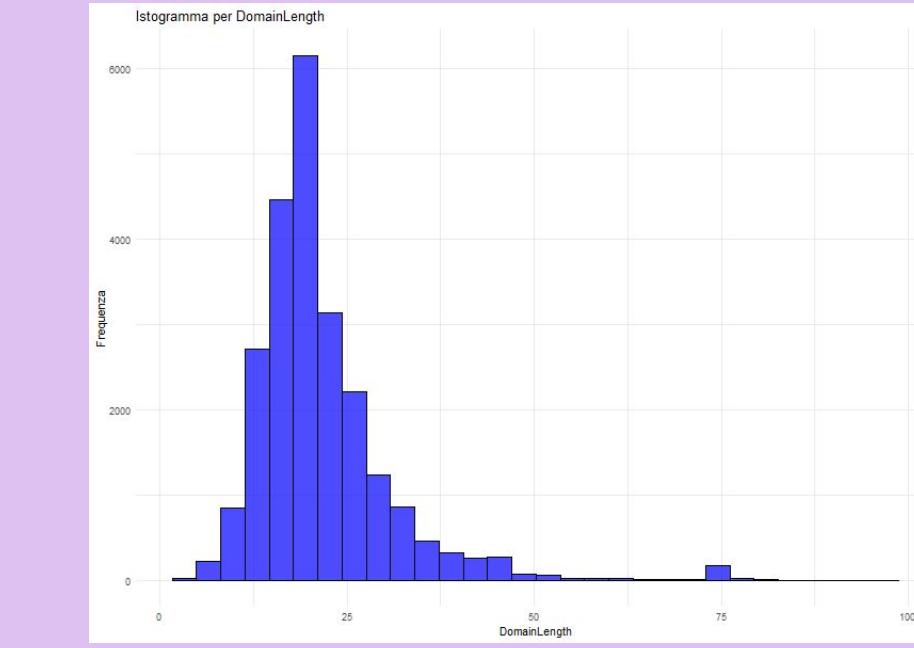
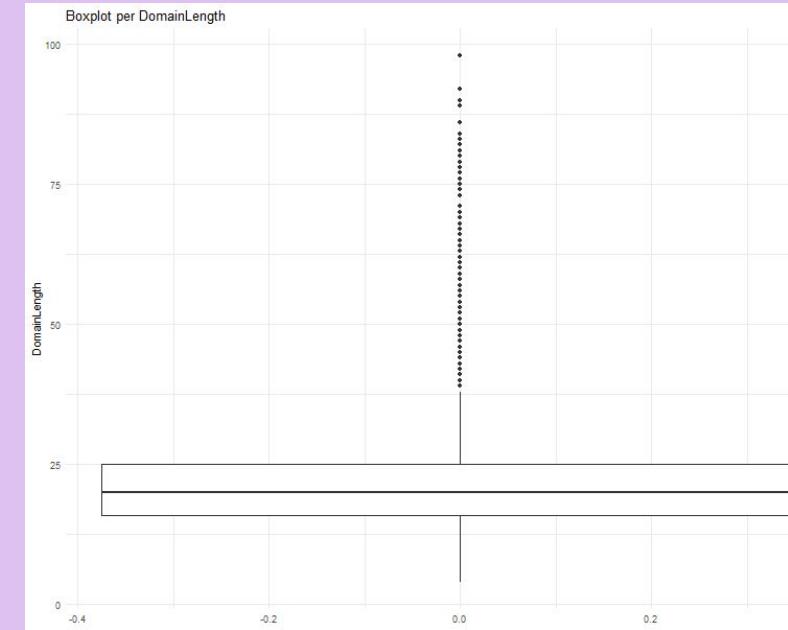
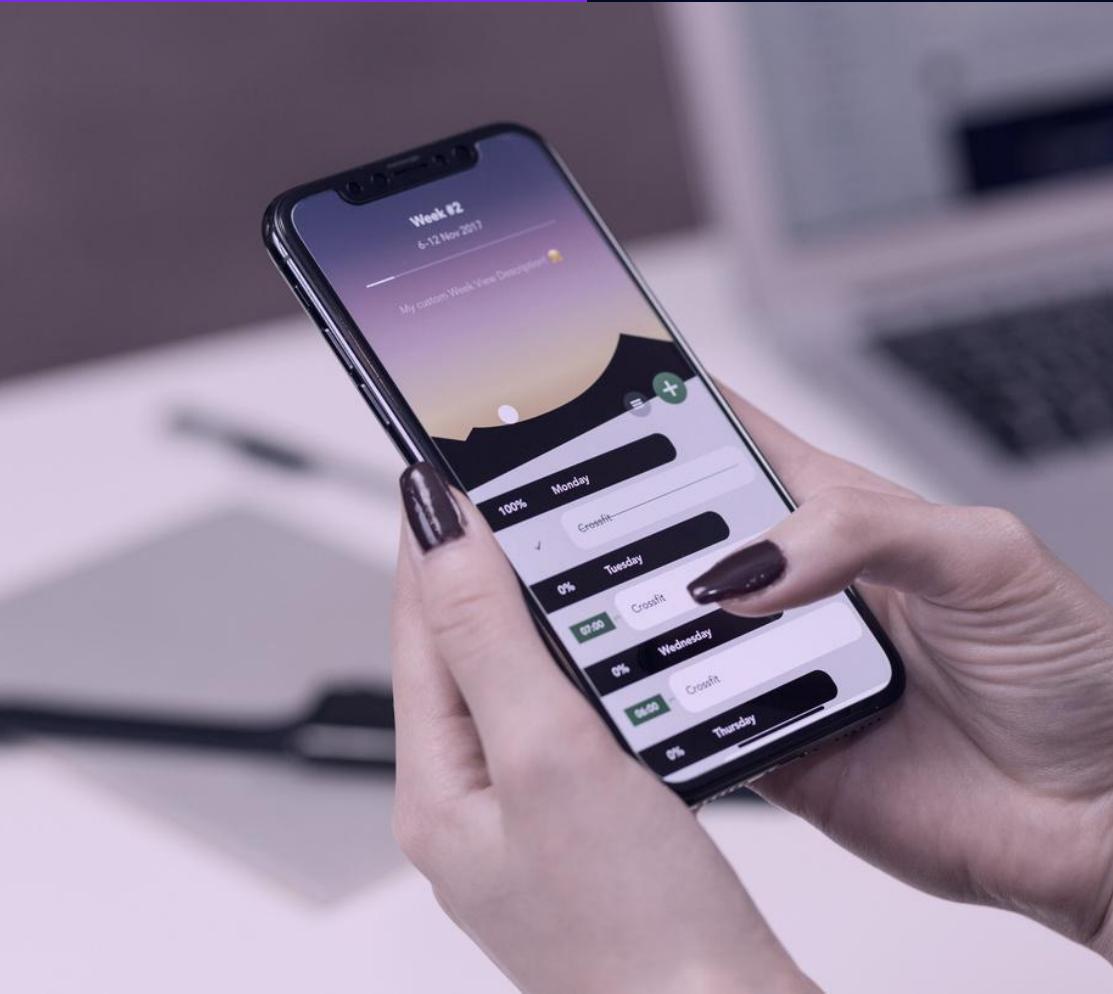
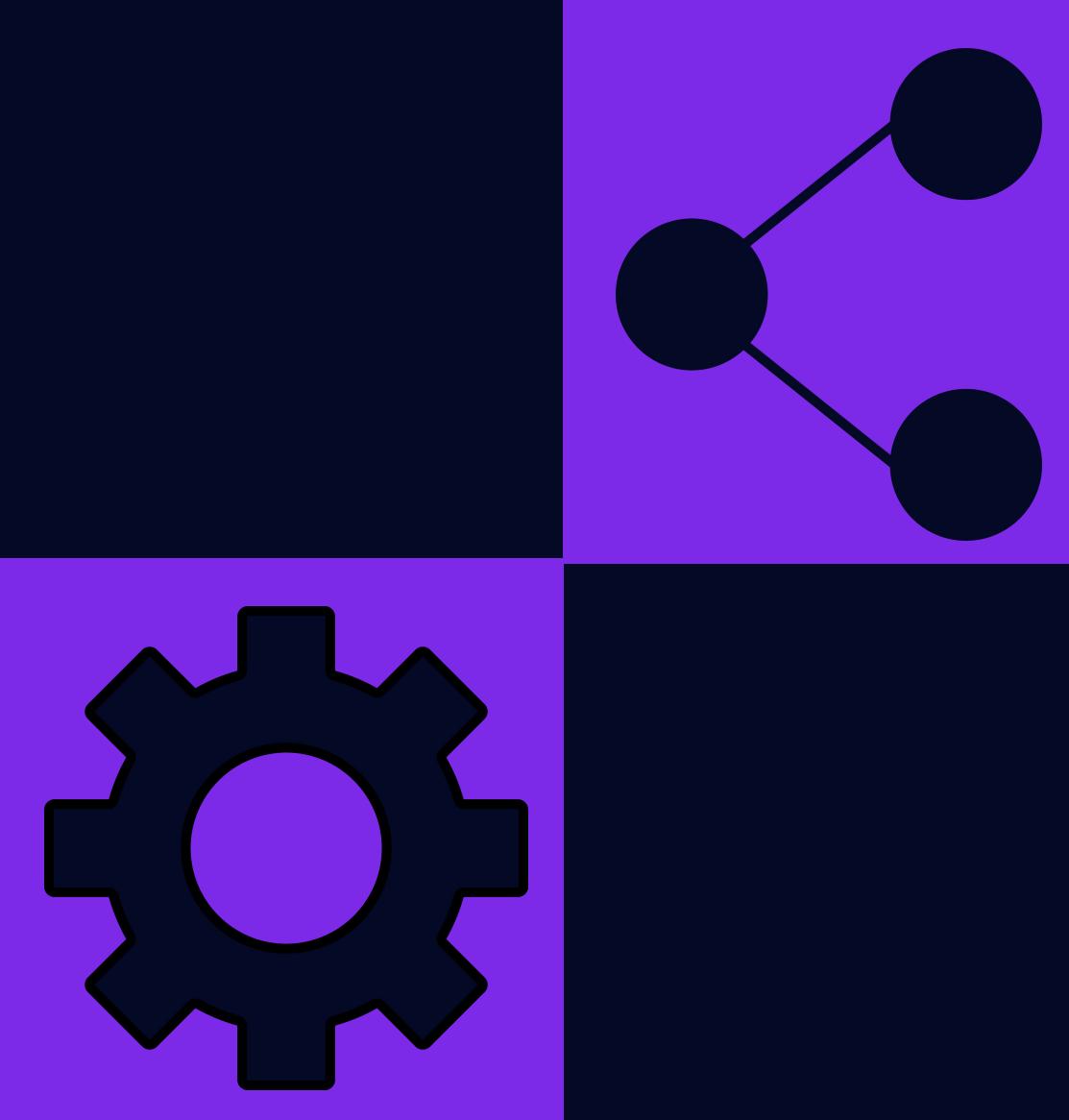


Momenti d'ordine

- Calcolo ed analisi dei valori di Skewness e Kurtosis.



Statistica descrittiva



Indice di sintesi

Media, Moda, Mediana

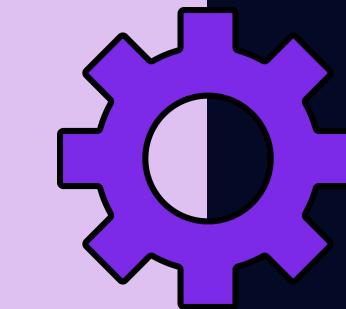
La media campionaria è, banalmente, la sommatoria di tutti gli n valori, divisi per n stesso.

La mediana campionaria, invece, bipartisce le osservazioni in due gruppi di uguale numerosità, in modo che ricada un egual numero di valori sia a sinistra sia a destra della mediana stessa. La moda, infine, 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. Tramite l'utilizzo di R siamo riusciti a ricavare i seguenti valori:

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

Da questa tabella possiamo notare che **TLD** presenta dei valori molto alti in tutte le categorie; il che significa che la maggior parte dei siti presenta un tld con questo valore. Anche **DomainLength** presenta dei valori quasi del tutto bilanciati rispetto alle altre osservazioni che hanno valori pressochè bassi e distaccati tra loro.



Momenti d'ordine

Skewness e Kurtosis

L'indice di Skewness di una distribuzione è un valore che cerca di fornire una misura della sua mancanza di simmetria. L'indice di Skewness può assumere valore > 0 (**Asimmetria Positiva**), $= 0$ (**Simmetria**) oppure < 0 (**Asimmetria negativa**). La Curtosi, invece, indica l'indice di allontanamento dalla normalità distributiva, da cui è possibile visualizzare un maggiore appiattimento o allungamento. Se il valore ottenuto è pari a 0 ci troviamo di fronte ad una **distribuzione normale**, se invece il valore è < 0 avremo una **distribuzione platicurtica**, se invece il valore è > 0 assisteremo ad una **distribuzione leptocurtica**.

Tramite R siamo riusciti a ricavare i seguenti valori dal dataset filtrato:

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

Analizzando la tabella mostrata qui a sinistra siamo giunti alla conclusione che alcune osservazioni come **NoOfAmpersandInURL**, **NoOfCSS** presentano dei valori molto elevati ed una quantità di outlier significativa; al contrario di **HasFavicon**, **Robots**, ad esempio, che presentano indici di Kurtosis minori di 0, indicando una distribuzione platicurtica e più appiattita.

Table 3: Skewness e Kurtosis

Modulo KNN

Definizione KNN

Uso di un modello non parametrico in ambiente di clustering supervisionato.

Fase di Training

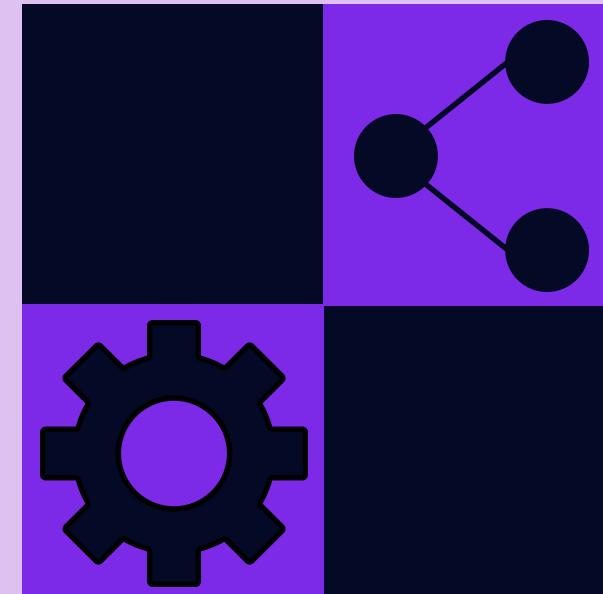
Utilizzo di centring e scaling per preprocessing dei dati, uso della cross validation per l'addestramento.

Evaluation Metrics

Uso di metriche numeriche e grafiche per la valutazione delle performance sul test set.



KNN



Nozione di distanza

Il modello usa questa metrica per raggruppare le osservazioni in cluster ed assegnare a quest'ultimi una label, viene usata la distanza euclidea

Scelta del K

La scelta di tale parametro risulta essere empirica nel nostro caso viene effettuata un'operazione di fine-tuning su un array di valori.

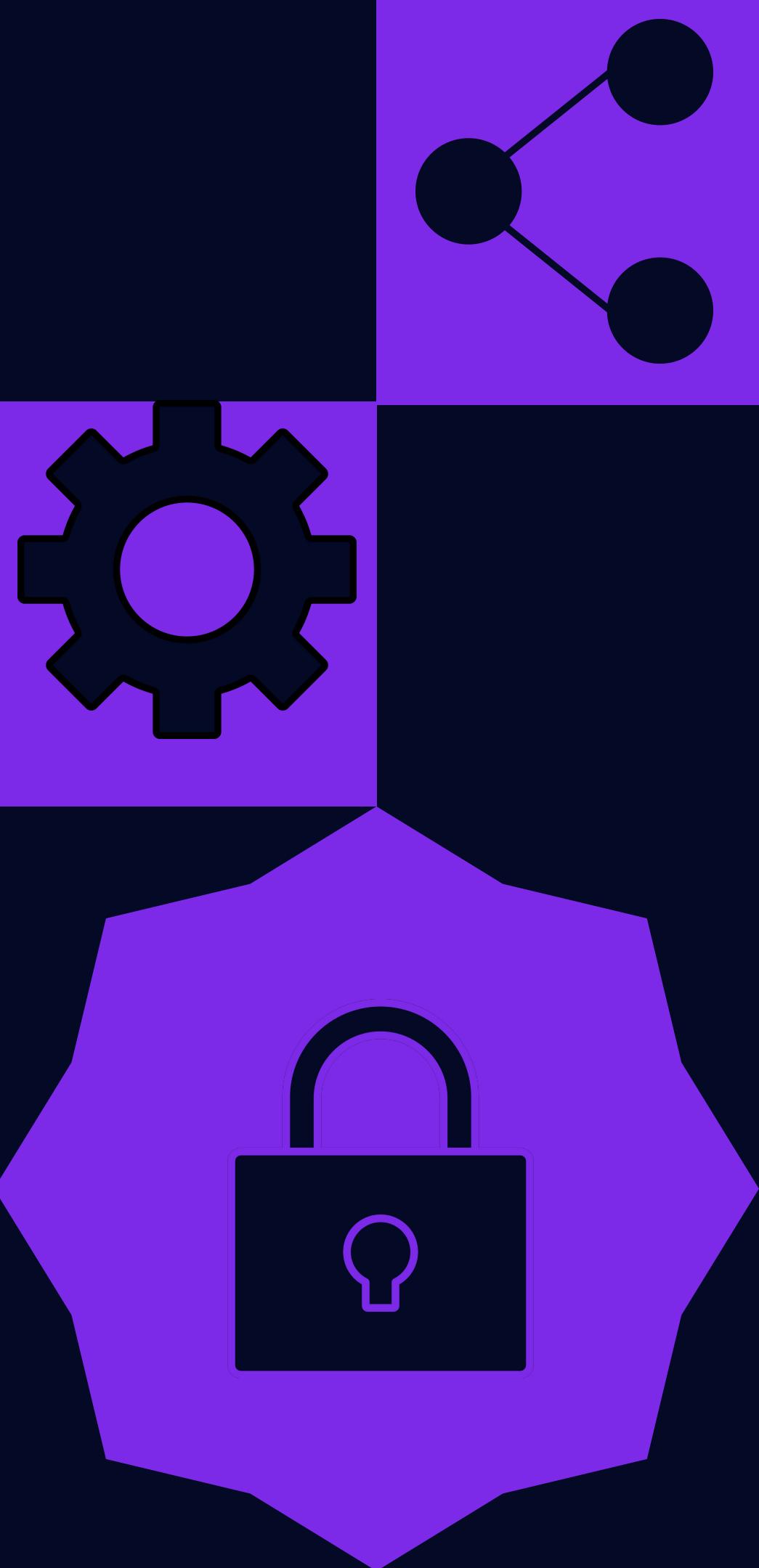
Fase di Training (1/2)

Prima di eseguire l'addestramento vengono effettuati i seguenti passaggi :

- Vengono eliminati tutti i duplicati presenti nei dati.
- Viene effettuato lo splitting del dataset in train e test con ratio 80% - 20%.
- Si applicano tecniche di pre-processing per migliorare la comprensibilità dei dati dati in pasto al modello. (centering e scaling).

In seguito in fase di training viene scelto il metodo di cross validation: **K-Repeated CV**, quest'ultimo funziona così:

- Il dataset di train viene diviso in K folds di cui K-1 viene addestrato il modello mentre l'ultima viene usata per la validazione.
- Il tutto viene ripetuto per n volte, nel nostro caso le ripetizioni sono 4 e i fold 10.



Fase di Training

(2/2)

```
#Carico il csv con i dati, scelto dataset di filtraggio.R
data1 <- read.csv("synthetic_dataset/gpt4o/dataset_sintetico_finale.csv", sep = ",")
data_no_dup1 <- data1 %>% distinct()

#Conversione della colonna label a factor per la classificazione binaria
data_no_dup1$label <- factor(data_no_dup1$label, levels = c("0", "1"))

#Splitting in train e testset
trainIndex1 <- createDataPartition(y = data_no_dup1$label, p = 0.8, list = FALSE)
train1 <- data_no_dup1[trainIndex1, ]
testing1 <- data_no_dup1[-trainIndex1, ]

#Data Pre-processing applicazione scaling piu' centering dei dati
pre_processor_values1 <- preProcess(train1, method = c("center", "scale"))
train_pre_processed1 <- predict(pre_processor_values1, train1)
testing_pre_processed1 <- predict(pre_processor_values1, testing1)

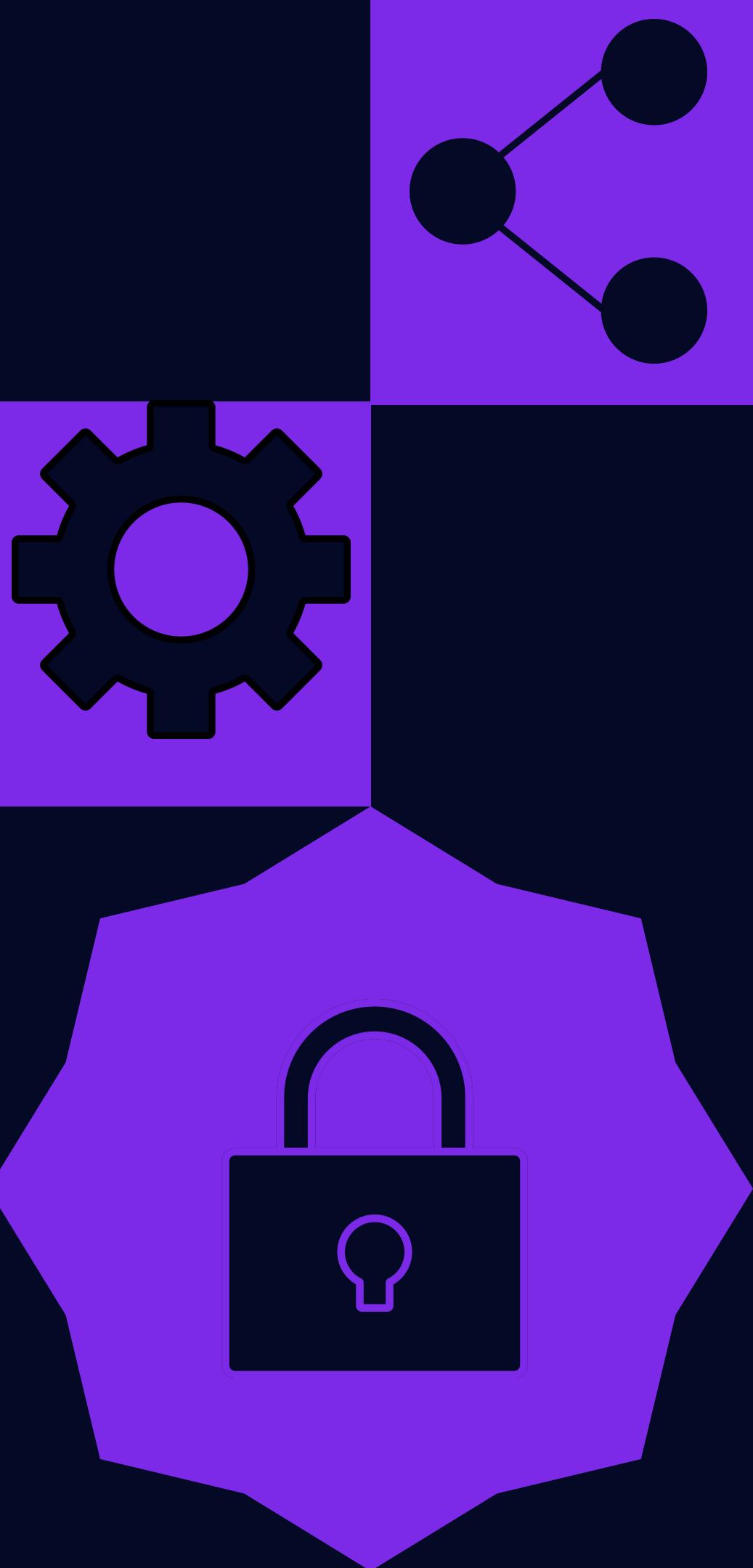
#Train and tuning del modello KNN
trctrl1 <- trainControl(method = "repeatedcv", number = 10, repeats = 4)
knn_model1 <- train(label ~., data = train_pre_processed1, method = "knn", trControl = trctrl1, tuneGrid = data.frame(k = c(3, 5, 7, 9, 11, 13)))
knn_model1

#Generazione delle predizioni del modello addestrato
test_pred1 <- predict(knn_model1, newdata = testing_pre_processed1)

#Valutazione performance su testset tramite metodo di caret confusion matrix
cm1 <- confusionMatrix(test_pred1, testing_pre_processed1$label, mode = "everything")
cm1

#Plotting della confusion Matrix
cm1_to_plot <- as.data.frame(cm1$table)
plot_cm1 <- ggplot(cm1_to_plot, aes(Prediction, Reference, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq)) +
  scale_fill_gradient(low = "white", high = "red") +
  labs(x = "phishing", y = "nophishing") +
  scale_x_discrete(labels = c("Classe 0", "Classe 1")) +
  scale_y_discrete(labels = c("Classe 0", "Classe 1"))

ggsave(paste0("model_performance_plot/", "confusionmatrix_k3_sintetico_finale.png"), plot = plot_cm1)
```

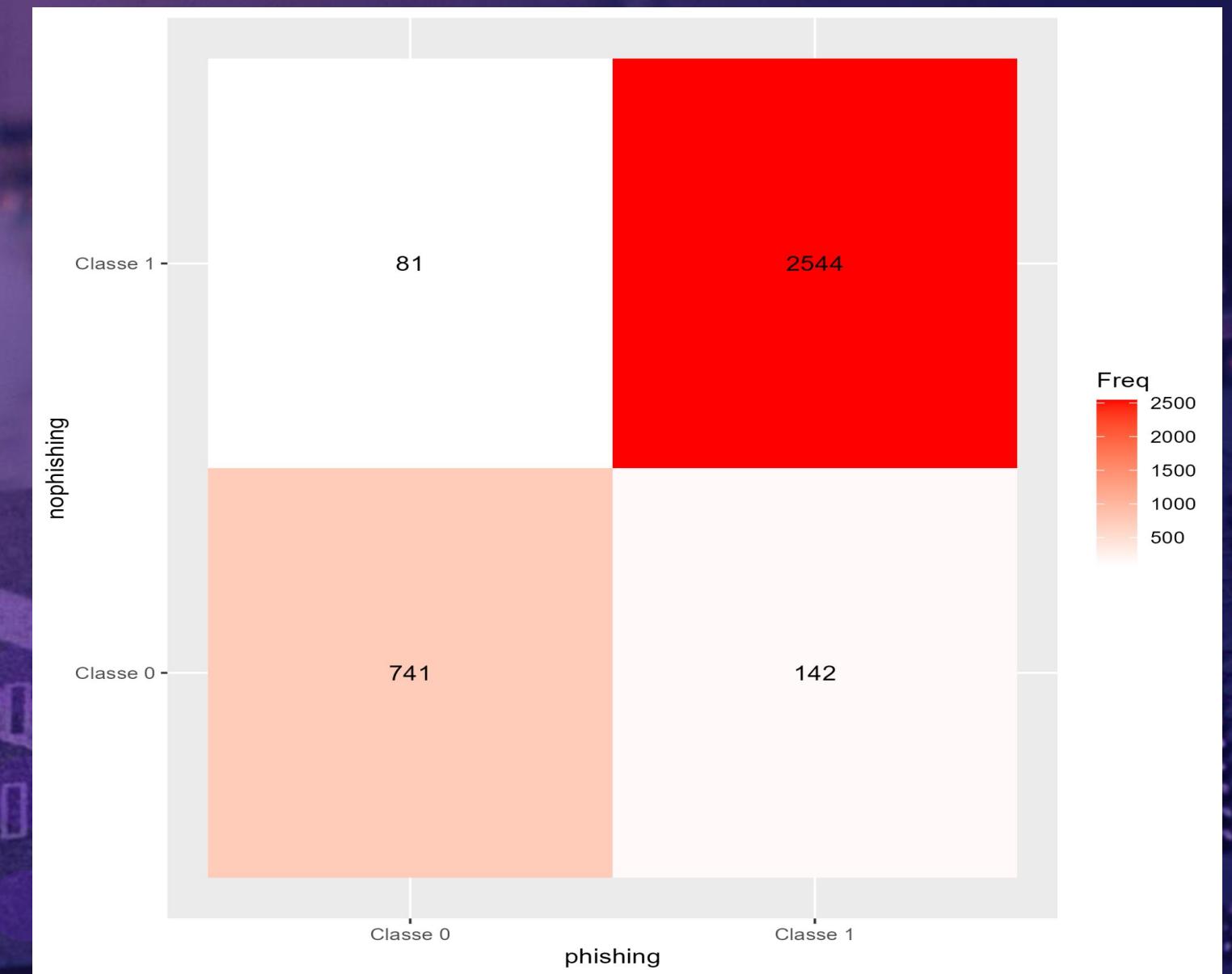
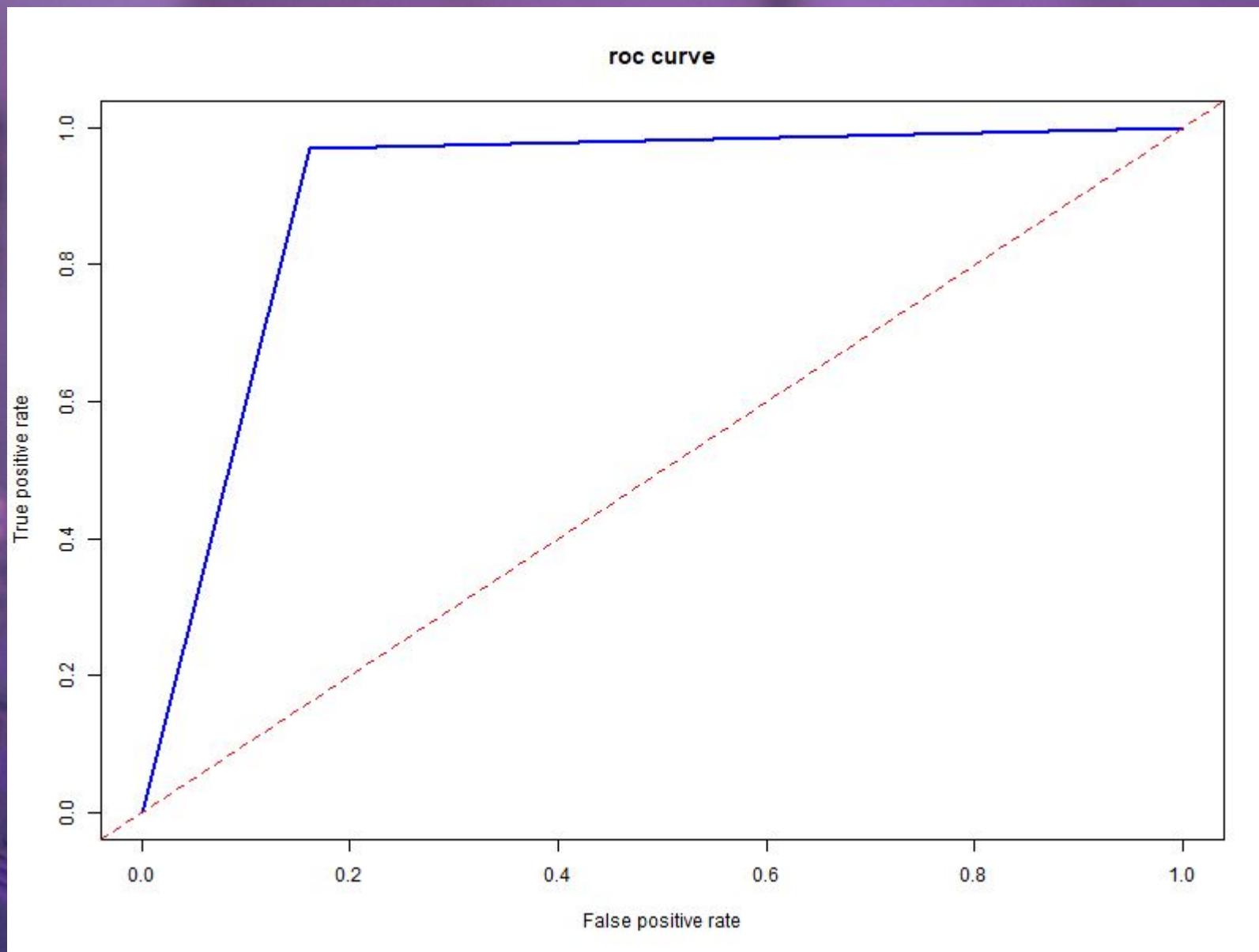


Evaluation metrics (1/2)

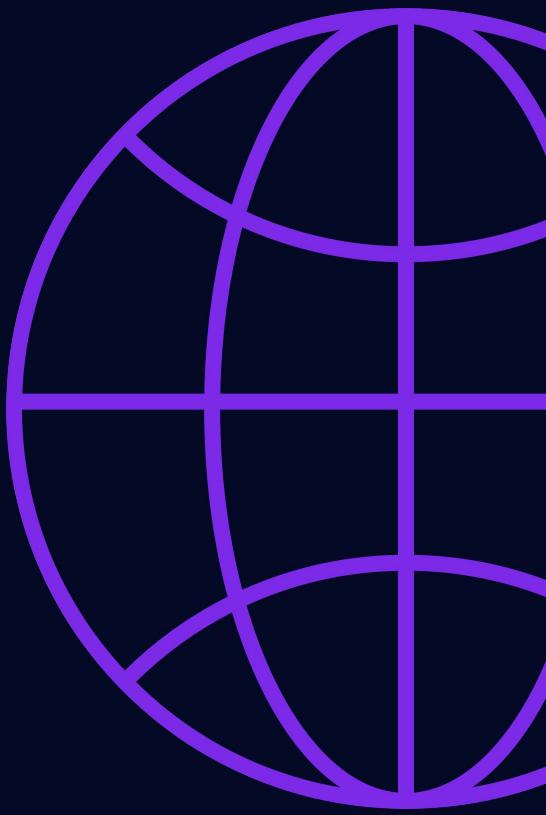
Metriche	Valori
Accuracy	94%
Sensitivity	85%
Specificity	97%
Precision	91%
Recall	85%
F1-score	88%

Le metriche qui usate ci illustrano delle ottime performance generali, ma ci illustrano anche alcuni limiti sulle capacità del modello di individuare la classe legata al phishing mostrano quindi basso false positive rate, aprendo un margine di miglioramento.

Evaluation metrics (2/2)

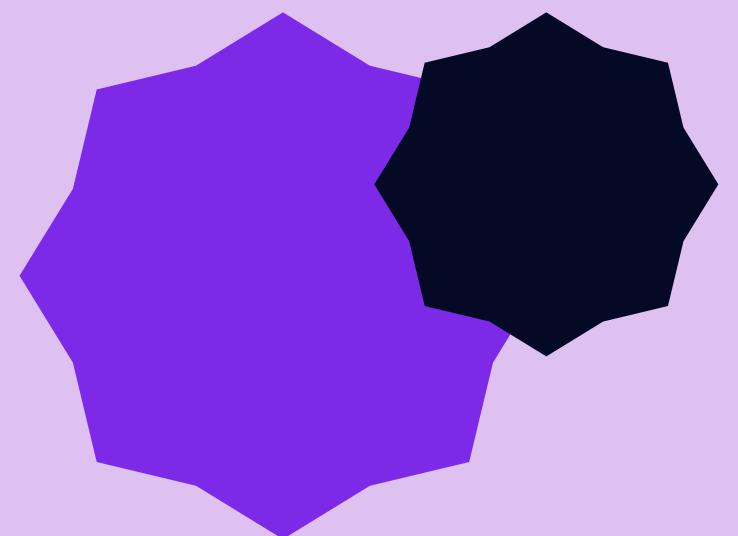


Modulo Test statistici (1/2)



Test utilizzati

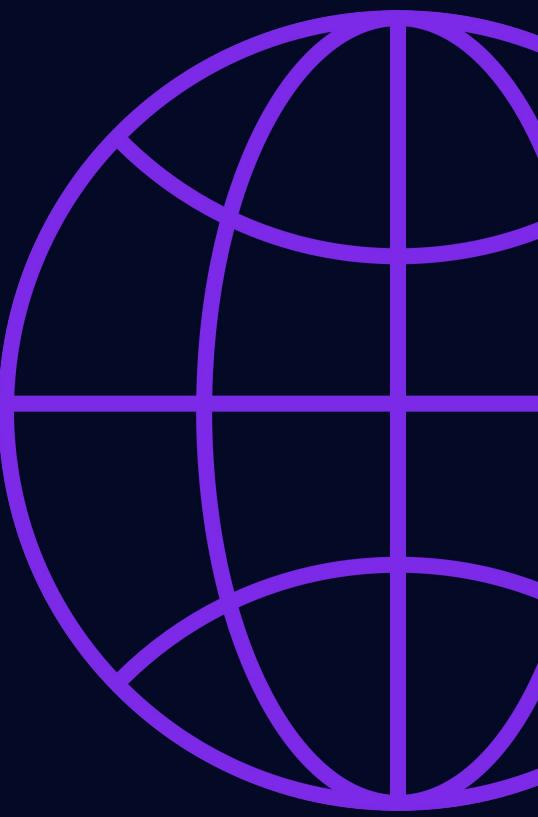
Viene scelto il test del chiquadro bilaterale per confrontare le popolazioni delle datest con distribuzioni note.



Distribuzioni scelte

I test sono stati effettuati sulle seguenti distribuzioni: Normale, binomiale e Poisson

Modulo Test statistici (2/2)



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

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

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 si nota che i mentre i test normale e di poisson sono completamente fallimentari, quello sulla binomiale ha successo su tutte le colonne binarie tranne che per crypto, con confidenza del 95%.

LLM: Research Question



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?

LLM: Generazione del dataset (1/5)

Per eseguire il task di synthetic data generation, viene selezionato i seguenti modelli: ChatGPT-4o e Gemma 2, in seguito viene dato loro questo prompt iniziale:

“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.”

Il dataset prodotto da entrambi i modelli vengono confrontati ed il migliore viene messo a paragone con il dataset originale in questo caso viene scelto quello di GPT-4o.

LLM: Generazione del dataset (2/5)

In seguito viene eseguita una fase di prompt engineering utilizzando alcune strategie presenti nel OpenAI cookbook che includono lo scrivere chiaramente l'obiettivo del prompt, dividere il task in sotto-task tramite linee guida ed l'inclusione di alcuni esempi del dataset d'input, produce il seguente 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.”

Come nel primo caso viene scelto il dataset generato da GPT-4o, nelle slide successive mostriamo lo script dei modelli che esegue la generazione.

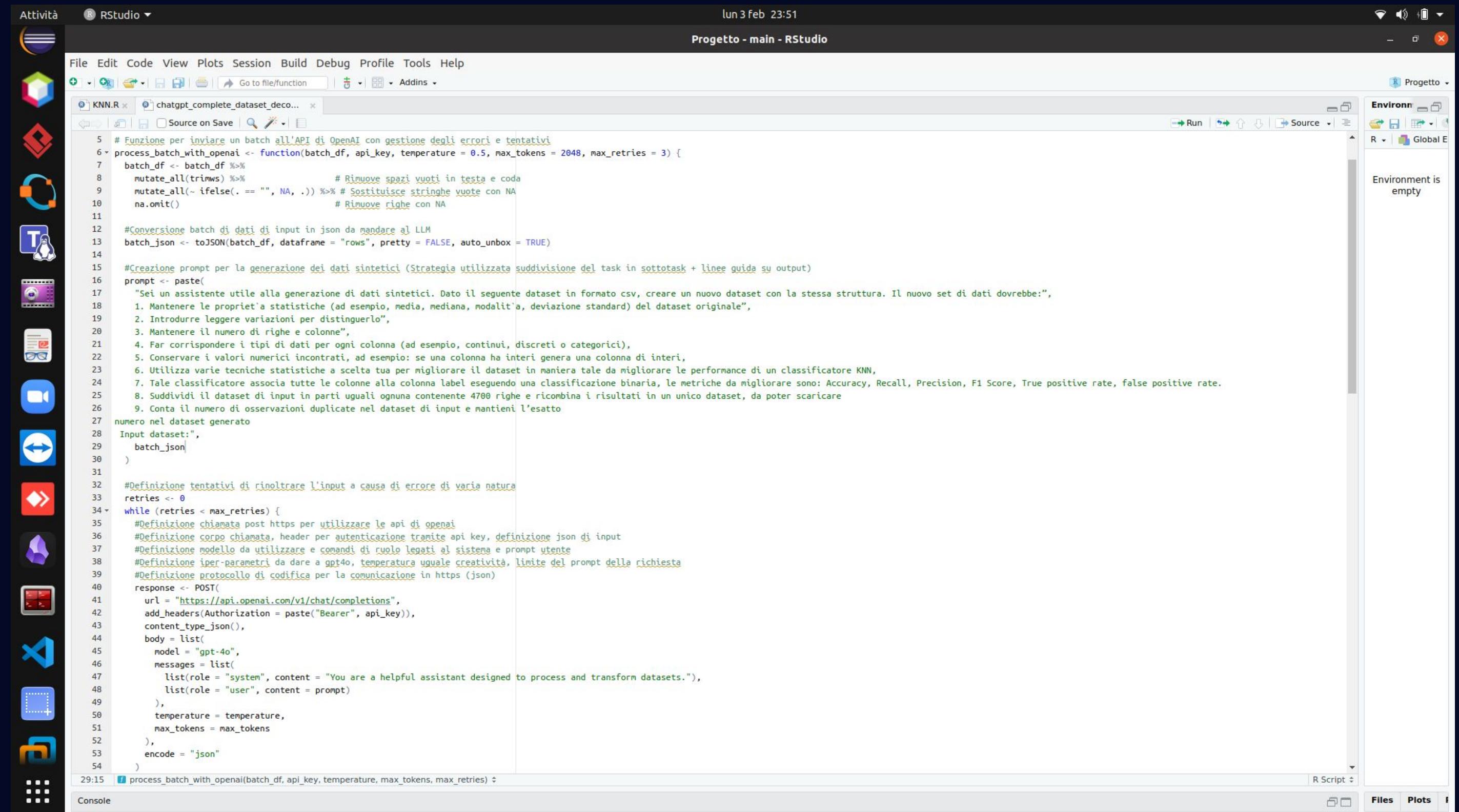
LLM: Generazione del dataset (3/5)

In seguito viene eseguita una fase di prompt engineering utilizzando alcune strategie presenti nel OpenAI cookbook che includono lo scrivere chiaramente l'obiettivo del prompt, dividere il task in sotto-task tramite linee guida ed l'inclusione di alcuni esempi del dataset d'input, produce il seguente 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.”

Come nel primo caso viene scelto il dataset generato da GPT-4o, nelle slide successive mostriamo lo script dei modelli che esegue la generazione.

LLM: Generazione del dataset (4/5)



The screenshot shows the RStudio IDE interface. The title bar reads "Progetto - main - RStudio". The top menu includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The left sidebar has icons for various RStudio features like Attività, RStudio, File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, and Addins. The main workspace contains an R script titled "chatgpt_complete_dataset_deco...". The code is as follows:

```
5 # Funzione per inviare un batch all'API di OpenAI con gestione degli errori e tentativi
6 process_batch_with_openai <- function(batch_df, api_key, temperature = 0.5, max_tokens = 2048, max_retries = 3) {
7   batch_df <- batch_df %>%
8     mutate_all(trimws) %>% # Rimuove spazi vuoti in testa e coda
9     mutate_all(~ ifelse(. == "", NA, .)) %>% # Sostituisce stringhe vuote con NA
10    na.omit() # Rimuove righe con NA
11
12 #Conversione batch di dati di input in json da mandare al LLM
13 batch_json <- toJSON(batch_df, dataframe = "rows", pretty = FALSE, auto_unbox = TRUE)
14
15 #Creazione prompt per la generazione dei dati sintetici (Strategia utilizzata suddivisione del task in sottotask + linee guida su output)
16 prompt <- paste(
17   "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:",
18   1. Mantener le propriet'a statistiche (ad esempio, media, mediana, modalit'a, deviazione standard) del dataset originale",
19   2. Introdurre leggere variazioni per distinguerlo",
20   3. Mantenere il numero di righe e colonne",
21   4. Far corrispondere i tipi di dati per ogni colonna (ad esempio, continui, discreti o categorici),
22   5. Conservare i valori numerici incontrati, ad esempio: se una colonna ha interi genera una colonna di interi,
23   6. Utilizza varie tecniche statistiche a scelta tua per migliorare il dataset in maniera tale da migliorare le performance di un classificatore KNN,
24   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.
25   8. Suddividi il dataset di input in parti uguali ognuna contenente 4700 righe e ricombina i risultati in un unico dataset, da poter scaricare
26   9. Conta il numero di osservazioni duplicate nel dataset di input e mantieni l'esatto
27 numero nel dataset generato
28 Input dataset:",
29   batch_json
30 )
31
32 #Definizione tentativi di rinoltrare l'input a causa di errore di varia natura
33 retries <- 0
34 while (retries < max_retries) {
35   #Definizione chiamata post https per utilizzare le api di openai
36   #Definizione corpo chiamata, header per autenticazione tramite api key, definizione json di input
37   #Definizione modello da utilizzare e comandi di ruolo legati al sistema e prompt utente
38   #Definizione iper-parametri da dare a gpt4o, temperatura uguale creatività, limite del prompt della richiesta
39   #Definizione protocollo di codifica per la comunicazione in https (json)
40   response <- POST(
41     url = "https://api.openai.com/v1/chat/completions",
42     add_headers(Authorization = paste("Bearer", api_key)),
43     content_type_json(),
44     body = list(
45       model = "gpt-4o",
46       messages = list(
47         list(role = "system", content = "You are a helpful assistant designed to process and transform datasets."),
48         list(role = "user", content = prompt)
49       ),
50       temperature = temperature,
51       max_tokens = max_tokens
52     ),
53     encode = "json"
54   )
55 }
```

The right sidebar shows the Environment pane with the message "Environment is empty". The bottom navigation bar includes tabs for R Script, Files, and Plots.

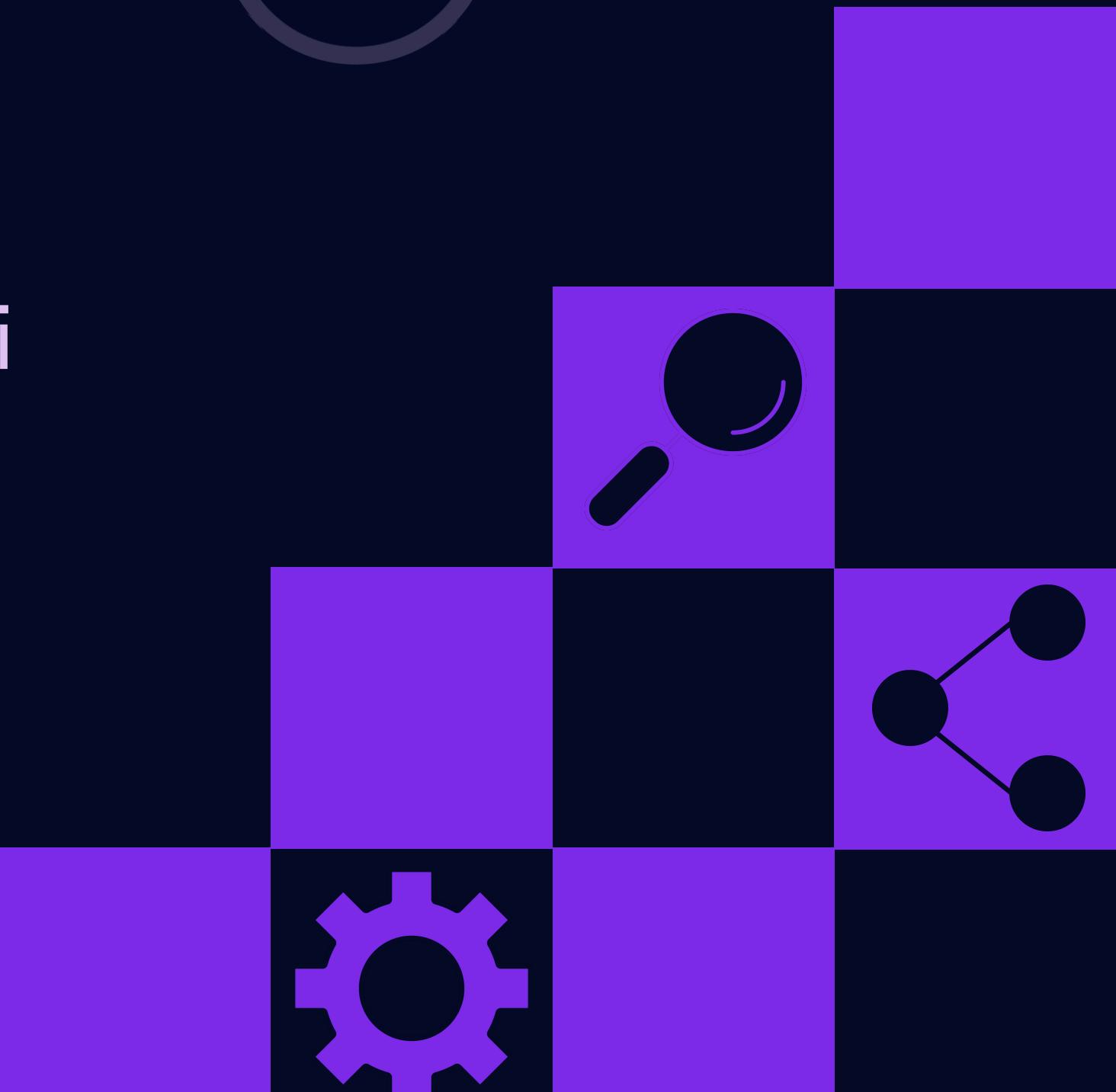
LLM: Generazione del dataset (5/5)

The screenshot shows the RStudio interface with the following components:

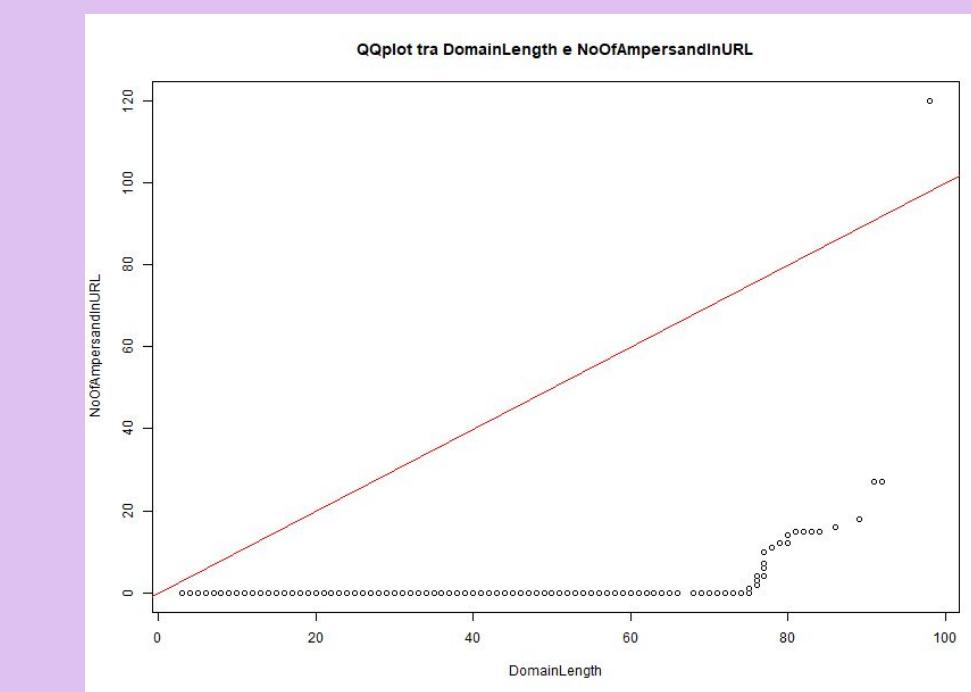
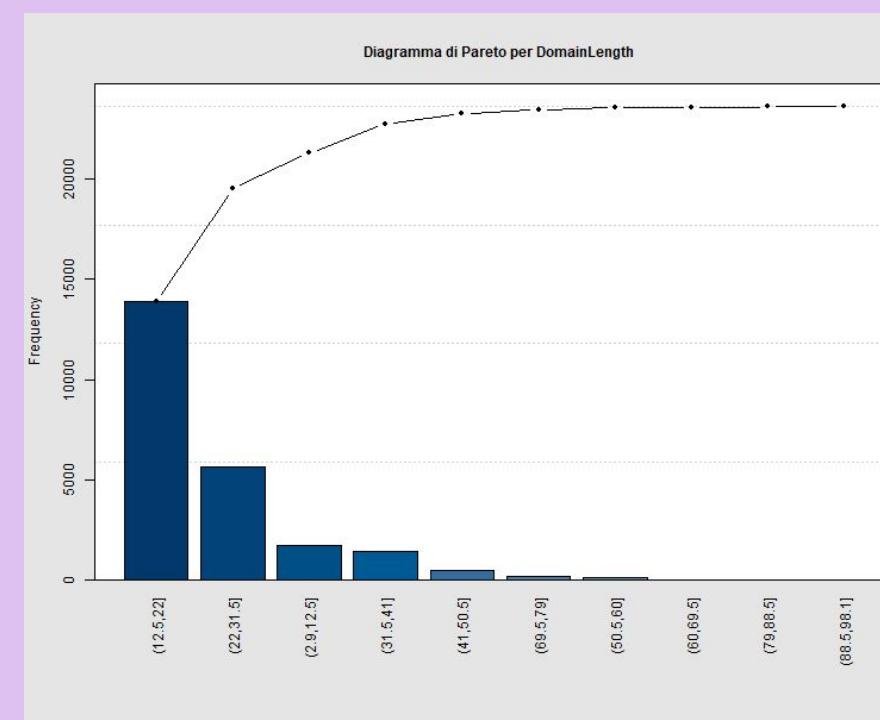
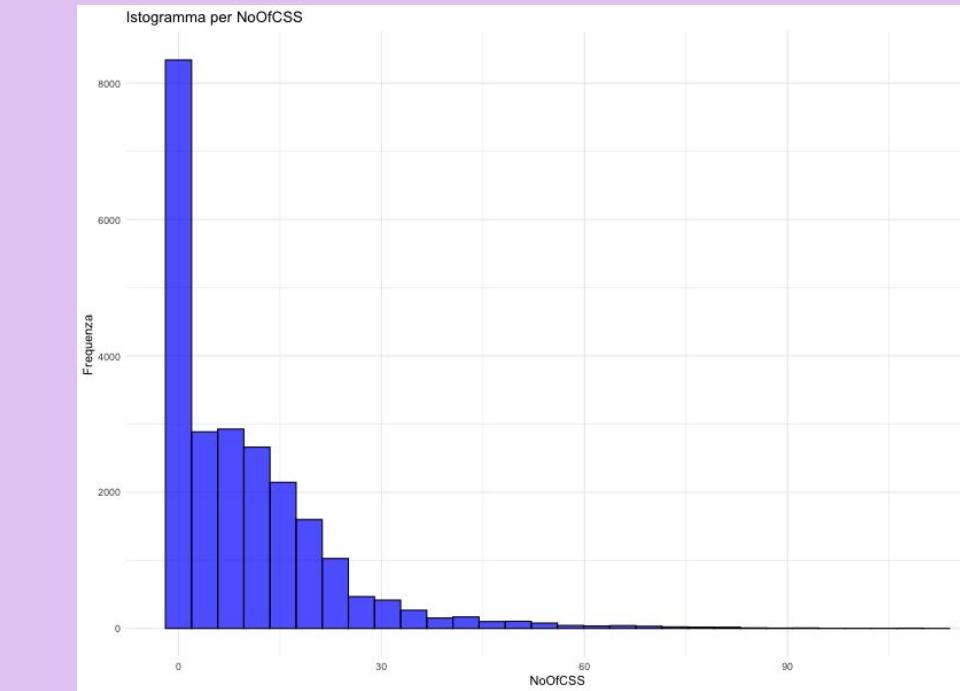
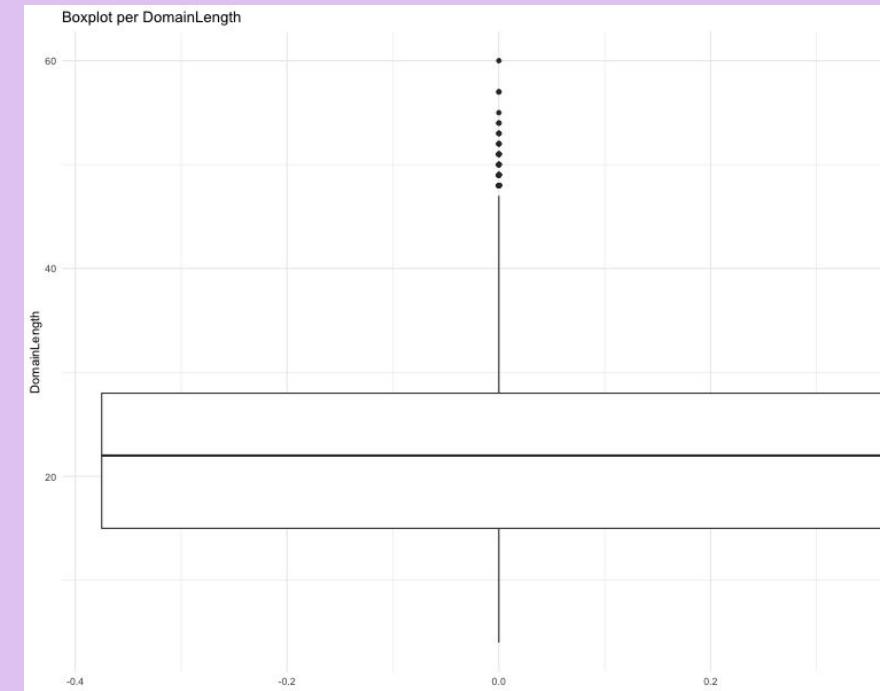
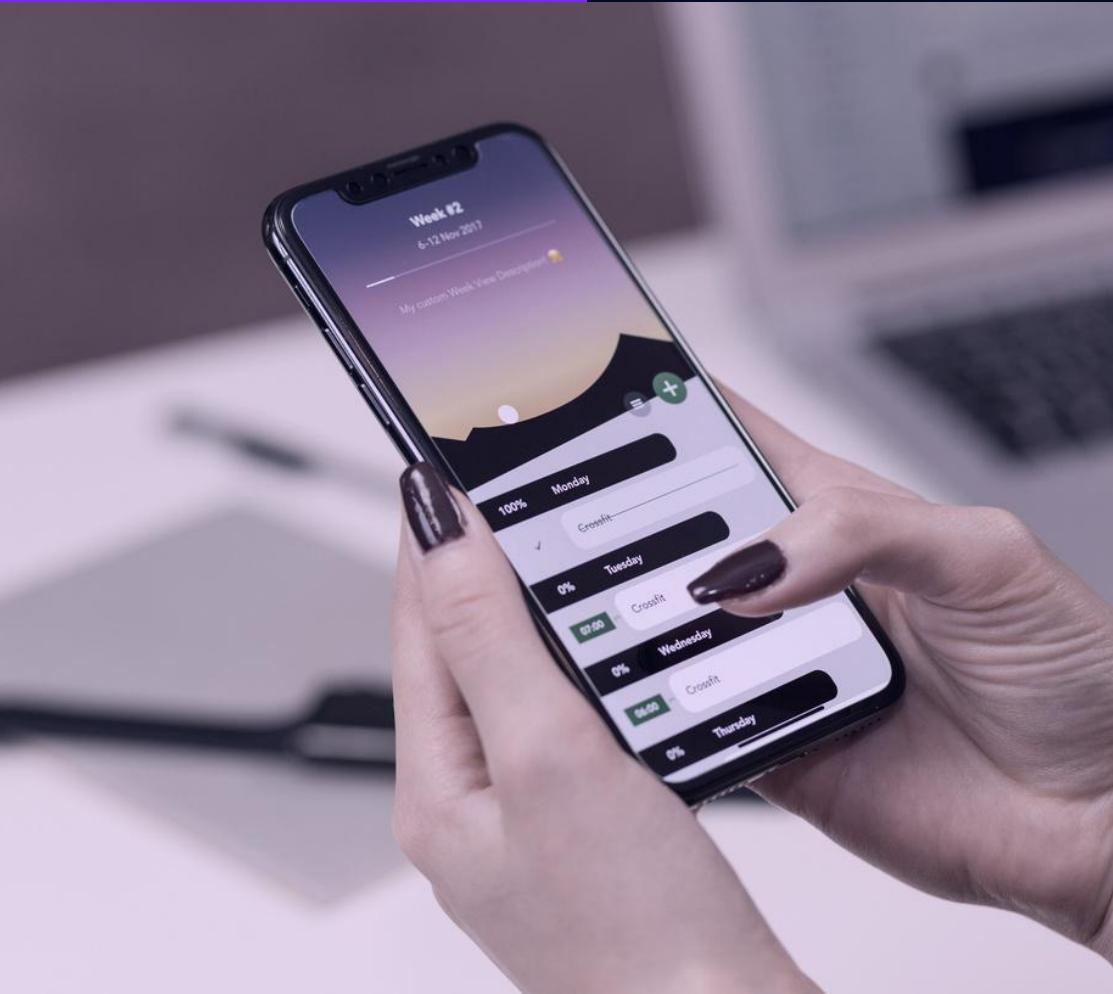
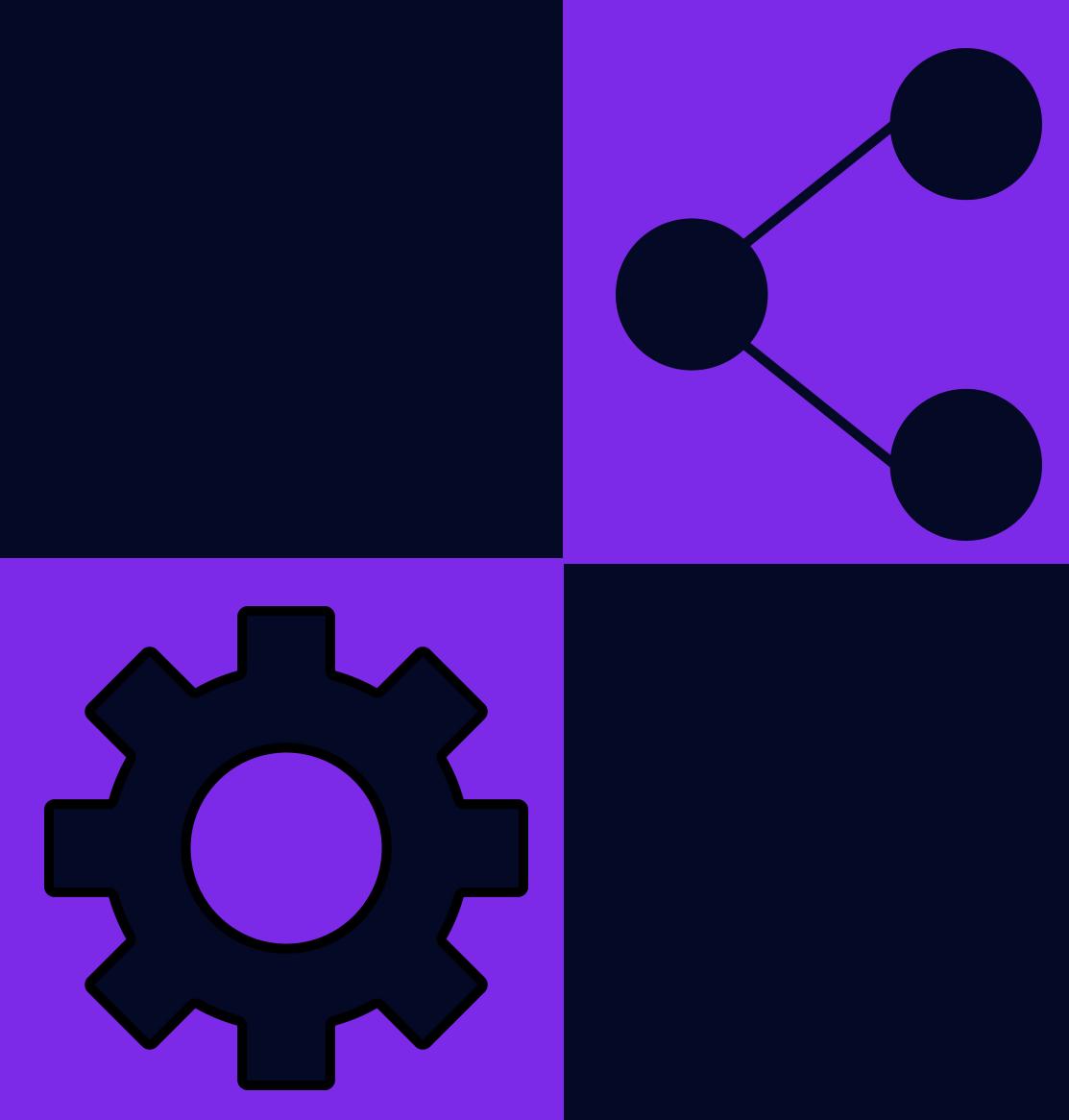
- Left Sidebar:** Icons for various tools like R, Python, Jupyter, and GitHub.
- Top Bar:** Shows "RStudio" and the date/time "lun 3 feb 23:53".
- Project Explorer:** Lists files: KNN.R, chatgpt_complete_dataset_deco..., ollama_gemma2_decompose_try.R.
- Code Editor:** Displays R code for generating a synthetic dataset using the Ollama API. The code includes imports for `httr`, `jsonlite`, and `dplyr`. It defines a function `process_batch_with_ollama` that sends multiple attempts to the API with different temperatures and token limits. The prompt for the API call is detailed, including instructions for handling errors and managing the dataset structure.
- Environment:** Shows "Environment is empty".
- Files:** Shows a list of files in the project directory, including `feature_importance_scoring.png`, `filtraggio2.R`, `finale_filtraggio2.csv`, `Heatmap_Correlazioni`, `iml_plot`, `IMLR`, `intermedio_correlazione.csv`, `intermedio_meanV.csv`, `intermedio_paper.csv`, `KNN.R`, `knnresultfiltraggio.png`, `model_performance_plot`, `ollama_gemma2_decompose_try.R`, `Paper_Phishing.pdf`, `Phishing_URL_Dataset_4.csv`, `Progetto.Rproj`, `README.md`, `statistica_descrittiva_dataset_com...`, `statistica_descrittiva_plot`, `statistica_descrittiva_plot_sintetico`, `statistica_descrittiva.R`, `synthetic_dataset`, and `test_noisson.R`.

Analisi Dataset Sintetici

Andremo ad analizzare i risultati ottenuti
dai due dataset sintetici

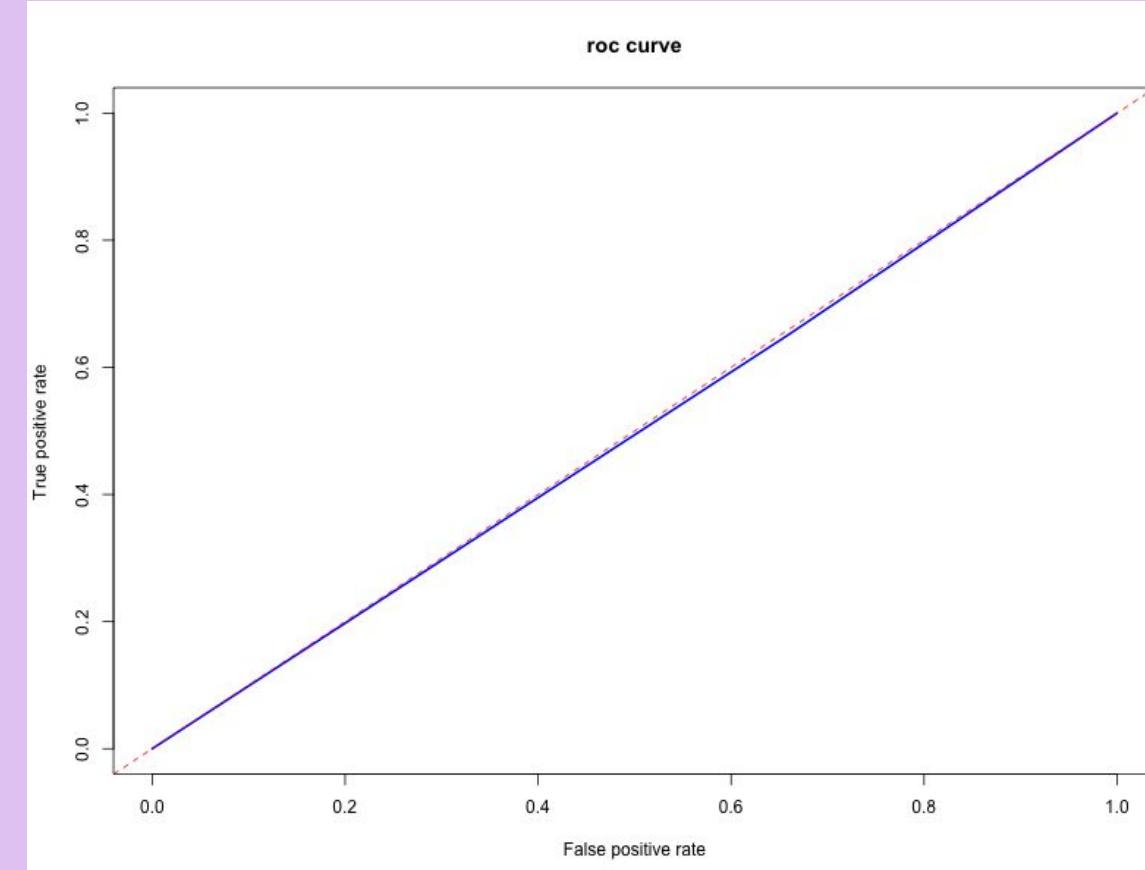


Statistica descrittiva Prompt1



KNN e Roc Curve Prompt1

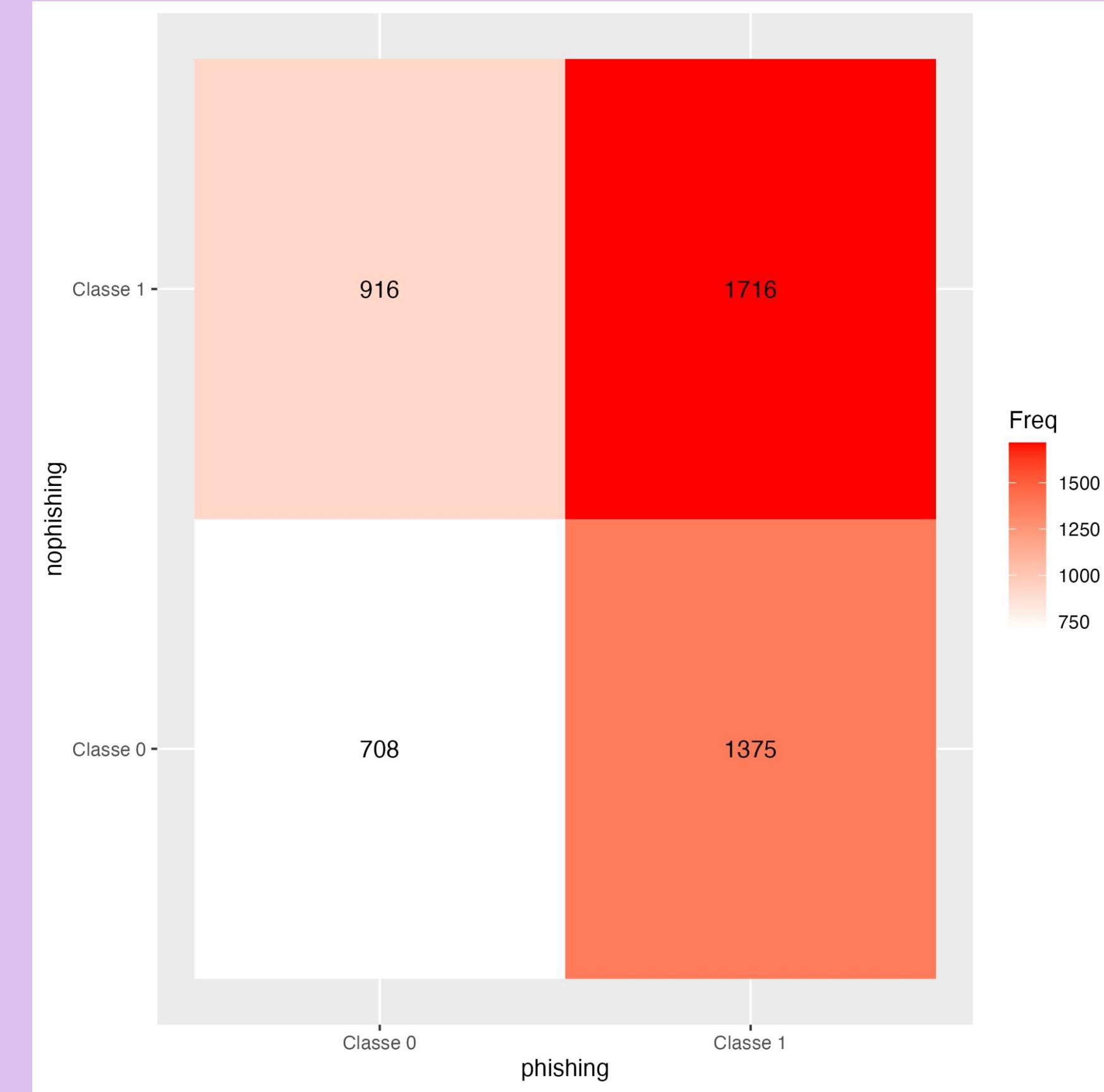
Da questa prima analisi dei risultati del KNN possiamo notare una differenza sostanziale rispetto al dataset filtrato originario. Il valore della sensitivity, ad esempio, è molto bassa, suggerendo che il modello non è molto bravo a rilevare i positivi. La precisione è anch'essa relativamente bassa (circa 43%), indicando che quando il modello non è molto affidabile. La Roc Curve invece 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.



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



Confusion Matrix Prompt1



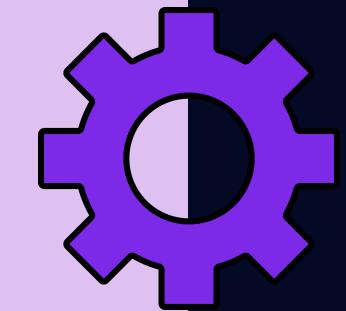
Indice di sintesi prompt1

Media, Moda, Mediana

Da questa tabella notiamo che 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**.

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

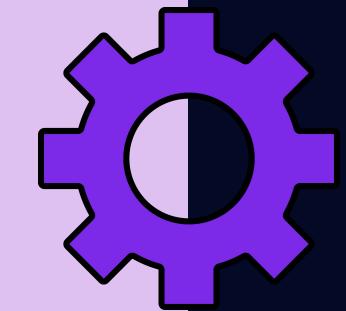
Table 9: Risultati di Media, Moda e Mediana



Skewness e Kurtosis Prompt1

Da questa tabella, possiamo notare un'enorme differenza dei valori di Skewness e Kurtosis rispetto al dataset originario. **DomainLength**, ad esempio ha subito una riduzione del valore di Skewness di circa il 2.30, ed una riduzione di Kurtosis scendendo ad un valore negativo. Abbiamo quindi ottenuto dei valori molto sbilanciati e condizionati da outlier a causa di un prompt incorretto o impreciso.

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



Test del Chi Quadrato Prompt1

Test Normale, Binomiale e Poisson

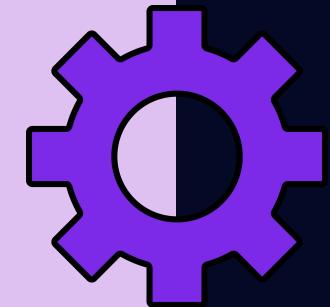
Come per il dataset originario abbiamo effettuato tre test sulle osservazioni del dataset: **Normale, Binomiale e Poisson**.

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

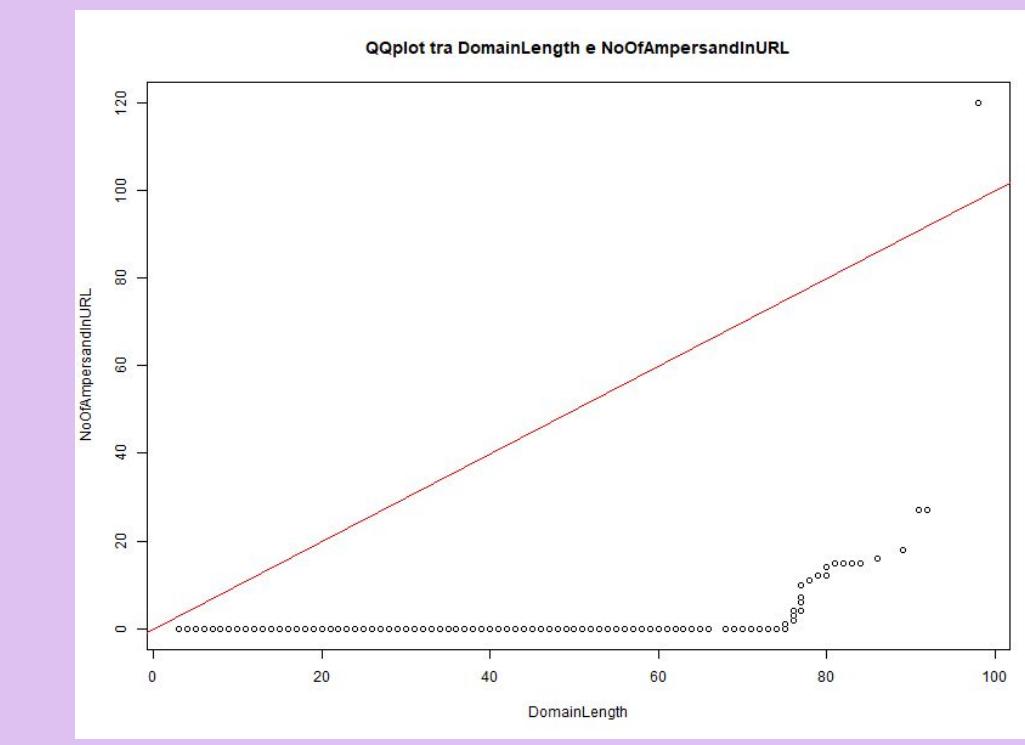
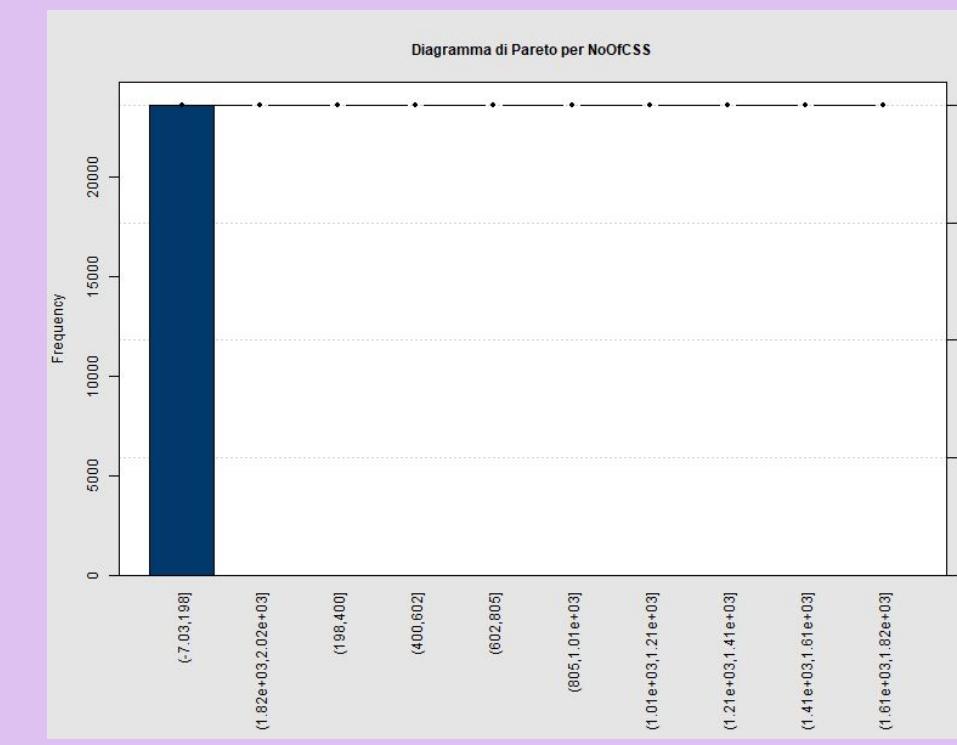
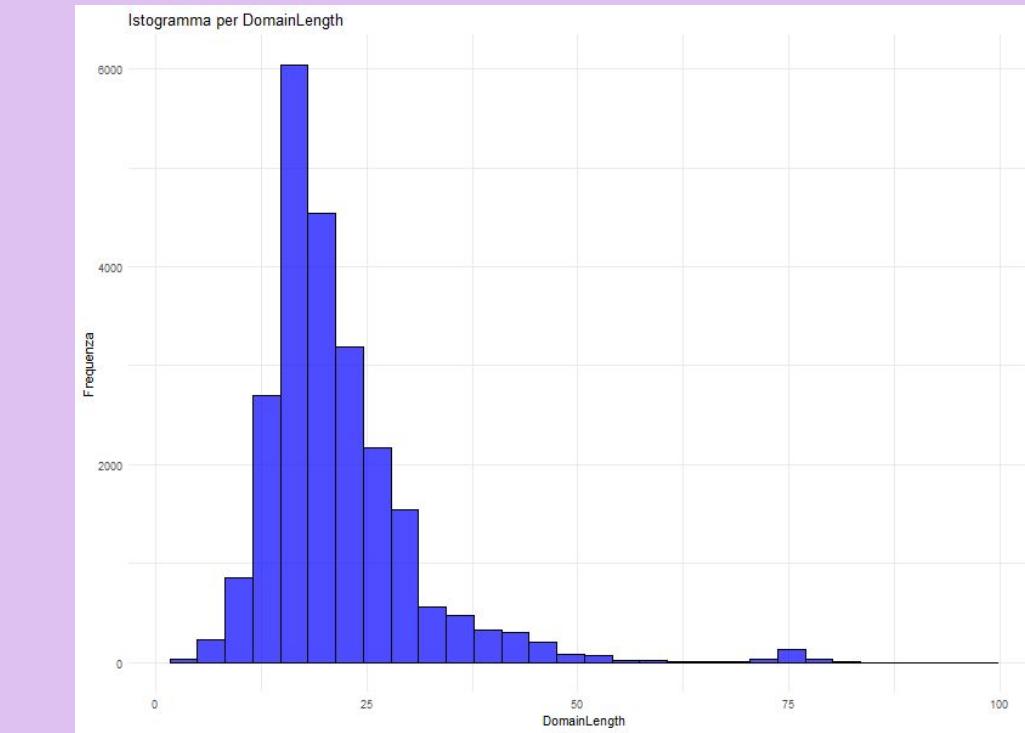
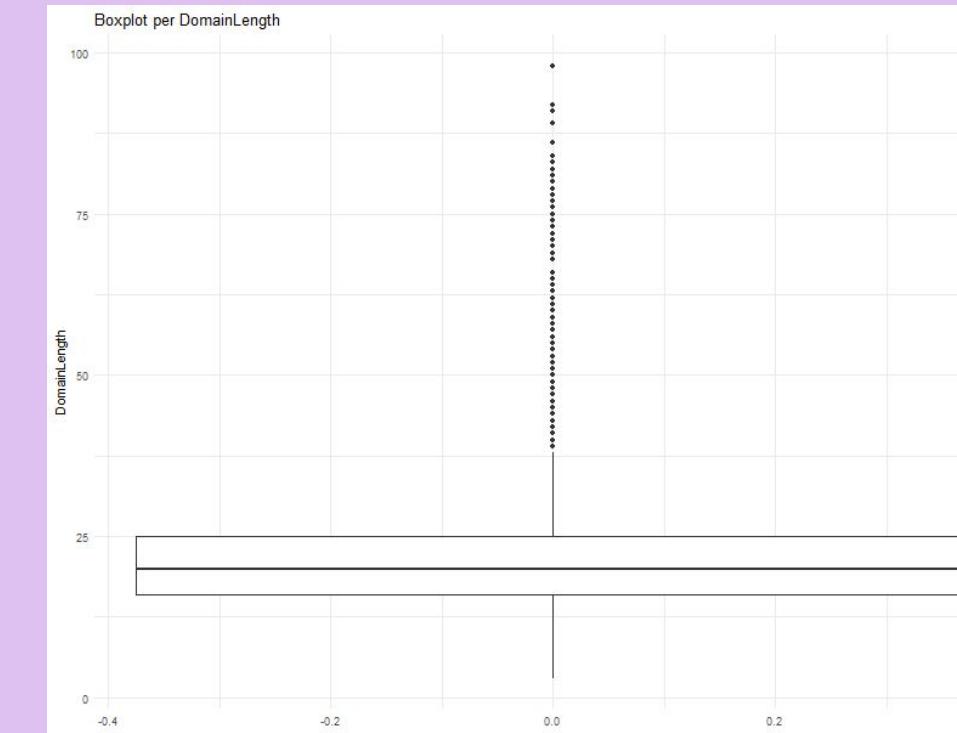
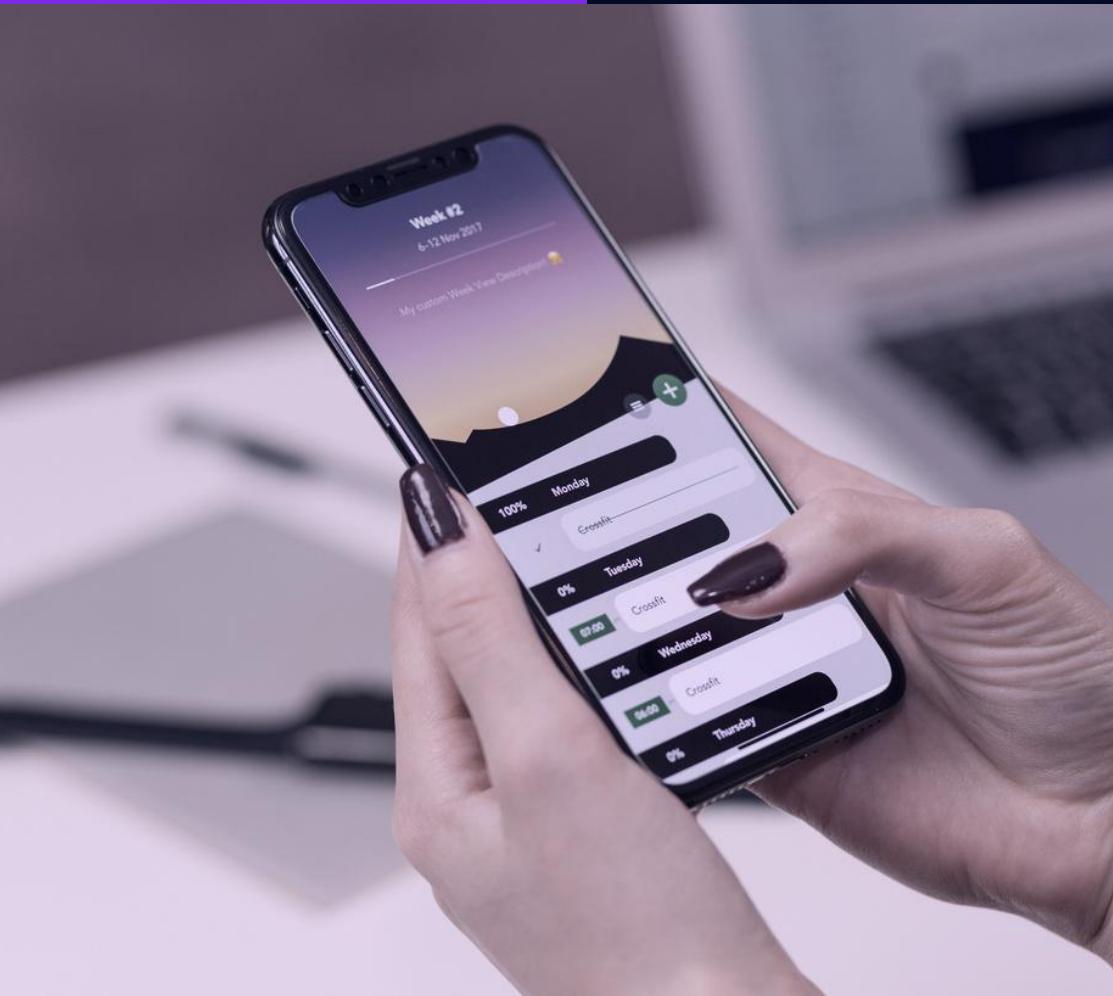
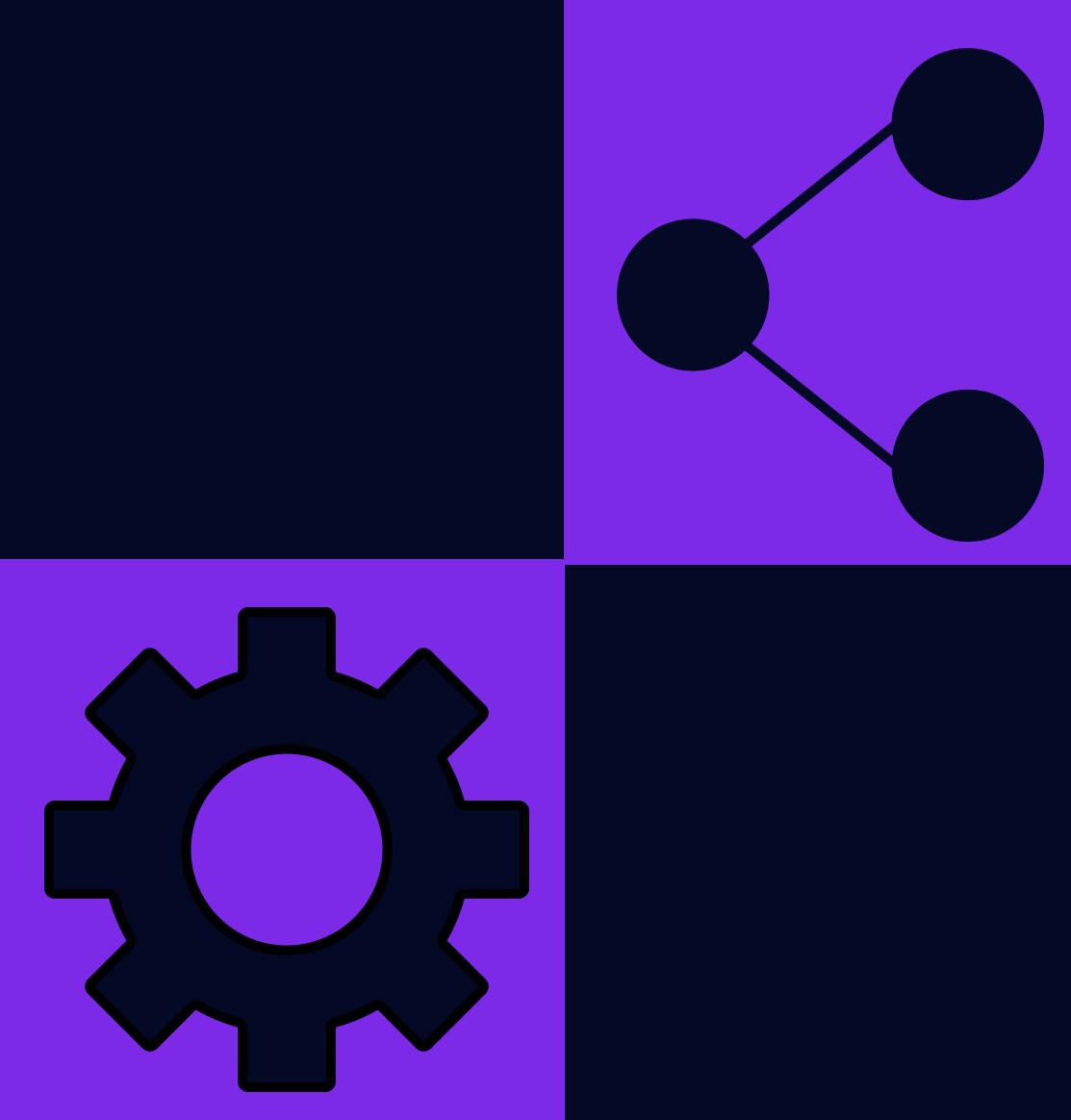
Osservazione	Chi2	First	Last	NObs
IsDomainIP	804.6949	0.0009820691	5.023886	0 23580
HasFavicon	0.01084553	0.0009820691	5.023886	14376 9204
NoOfURLRedirect	0.04491974	0.0009820691	5.023886	20243 3337
Bank	3.268503e-05	0.0009820691	5.023886	20397 3183
HasTitle	1.245781e-06	0.0009820691	5.023886	3372 20208
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

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

Dalla prima tabella possiamo analizzare i valori ottenuti dal test della normale. Nessuno dei dati ha passato il test, poiché il numero di osservazioni attese per l'intervallo non è rispettato (valori troppo elevati. Analogamente possiamo notare dalla seconda tabella (**binomiale**) che **HasTitle**, **Bank** e **IsDomainIP** non hanno passato il test. Crypto, invece, ha passato il test rispetto al dataset originario. Nella terza tabella abbiamo eseguito il test di **Poisson** sui dati che non avevano passato il test della normale; anche qui ottenendo tutti fallimenti così come nel dataset originario.

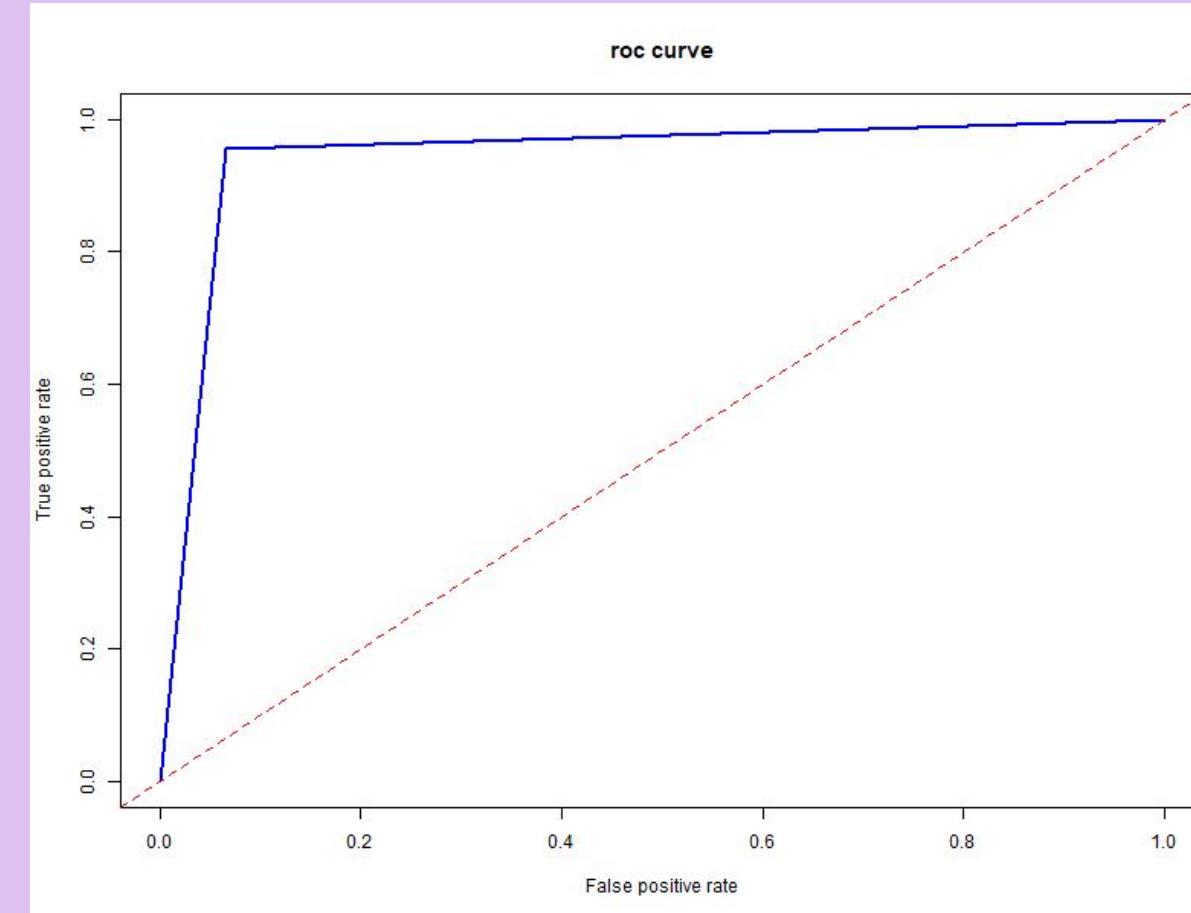


Statistica descrittiva Dataset Sintetico Finale



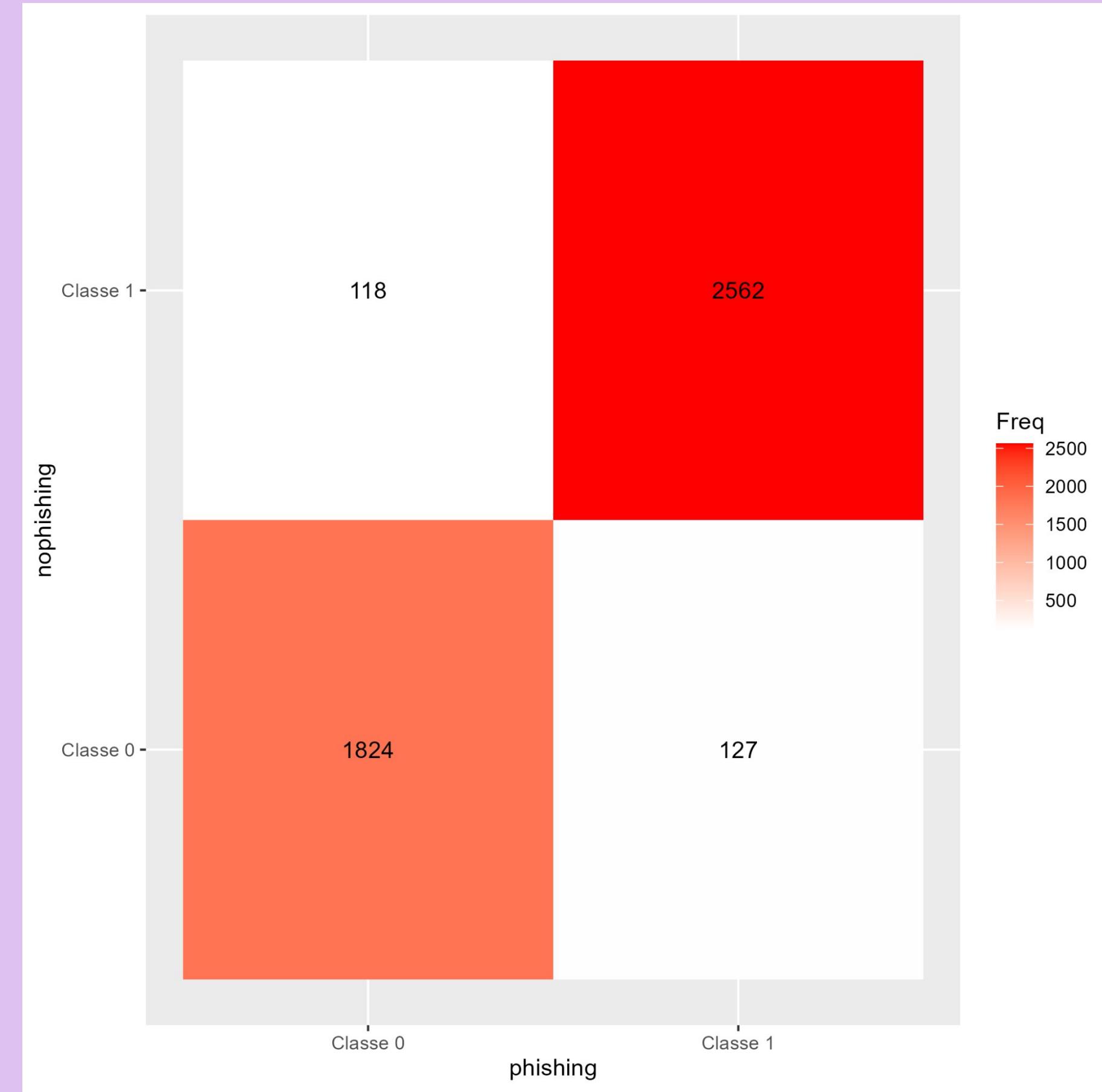
KNN e Roc Curve Sintetico Finale

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. La **Roc Curve** mostra che teoricamente il modello potrebbe separare bene le due classi.



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

Confusion Matrix Sintetico Finale

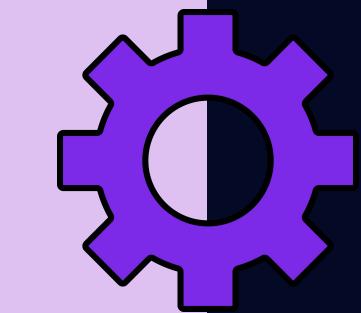


Indice di sintesi sintetico finale

Media, Moda, 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. Il prompt dettagliato ha dato riscontro a delle migliorie del dataset finale, che ha portato il tutto ad un improvement sostanziale.

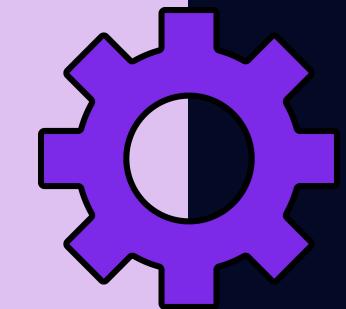
Osservazione	Media	Moda	Mediana
DomainLength	21.57	19	20.00
IsDomainIP	0.002417	0	0.000000
TLD	148.9	96	100.0
TLDLength	2.767	3	3.000
NoOfSubDomain	1.163	1	1.000
NoOfAmpersandInURL	0.02774	0	0.00000
HasTitle	0.8637	1	1.0000
HasFavicon	0.3581	0	0.0000
Robots	0.2681	0	0.0000
NoOfURLRedirect	0.1332	0	0.00000
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



Skewness e Kurtosis Sintetico Finale

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.

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



Test del Chi Quadrato Sintetico Finale

Test Normale, Binomiale e Poisson

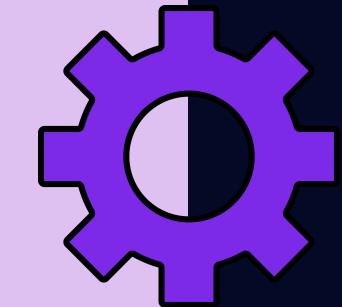
Come per il dataset originario abbiamo effettuato tre test sulle osservazioni del dataset: **Normale, Binomiale e Poisson**.

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

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

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

Dalla prima tabella possiamo analizzare i valori ottenuti dal test della normale. Nessuno dei dati ha passato il test, poiché il numero di osservazioni attese per l'intervallo non è rispettato (valori troppo elevati). Analogamente possiamo notare dalla seconda tabella (**binomiale**) che **HasTitle** e **Bank** non hanno passato il test. Crypto, invece, ha passato il test rispetto al dataset originario. Nella terza tabella abbiamo eseguito il test di **Poisson** sui dati che non avevano passato il test della normale; anche qui ottenendo tutti fallimenti così come nel dataset originario, per via di valori molto elevati.



Conclusioni e Sviluppi futuri (1/2)

Dopo l'analisi precedente possiamo in sintesi rispondere alla domande poste nella sezione LLM:

- **ARQ1:** Il primo prompt non migliora il modello, mentre quello finale invece lo migliora in maniera considerevole.
- **ARQ2:** Per nessun dataset si è riusciti tramite il test del chiquadro a trovare una distribuzione nota.
- **ARQ3:** il dataset prodotto dal prompt finale risulta essere il più instabile ma ciò serve per generare dati di qualità che migliorino il modello.

Conclusioni e Sviluppi futuri (2/2)

Concludiamo suggerendo alcuni spunti che possono migliorare il lavoro già svolto:

- 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.
- 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.

Grazie per l'attenzione!

