Università degli studi di Milano-Bicocca

Scuola di Economia e Statistica
Corso di Laurea in
STATISTICA E GESTIONE DELLE INFORMAZIONI

Phishing or legitimate?

Sara Borello Matr. N. 882793
Lorenzo Giuliano Matr. N. 870811
Keita Jacopo Viganò Matr. N. 870980

ANNO ACCADEMICO 2023/2024

Introduzione al progetto

Il phishing è riconosciuto come una delle minacce di cybercriminalità più semplici e diffuse. Questa tecnica non richiede agli hacker di decifrare codici complessi o di violare firewall avanzati. Invece, si basa sull'inganno, utilizzando e-mail che sollecitano emotivamente i destinatari a fornire dati sensibili, come credenziali personali, cliccando su link che reindirizzano a siti web falsificati. Questi siti imitano molto fedelmente quelli autentici, ingannando le vittime. Con l'aumento della sofisticazione dei cybercriminali, è emerso che una grande percentuale di siti di phishing utilizza persino la protezione SSL, un tempo distintiva dei soli siti autentici. La frequenza con cui vengono lanciati nuovi siti di phishing evidenzia l'urgenza di studi approfonditi per lo sviluppo di metodi efficaci di rilevamento e prevenzione. Il rilevamento degli attacchi di phishing è principalmente un problema di classificazione, per il quale le tecniche di machine learning sono considerate soluzioni promettenti. Queste tecniche necessitano di un'accurata selezione di classificatori efficienti, l'uso di caratteristiche distintive e la raccolta di campioni di dati rappresentativi per l'addestramento. I sistemi basati su machine learning sviluppati per il rilevamento del phishing si dividono generalmente in due categorie: quelli basati sul contenuto e quelli basati sugli URL. Nel primo caso, il phishing viene rilevato esaminando attivamente o passivamente il contenuto delle pagine web visitate; nel secondo, viene esaminato solo l'URL delle pagine web visitate. In questo contesto, l'obiettivo principale è utilizzare tecniche di Machine Learning per creare un sistema in grado di identificare siti web di phishing in modo preciso. Questo processo coinvolge un'analisi dettagliata sia delle caratteristiche degli URL che del contenuto delle pagine web. Per quanto riguarda le caratteristiche degli URL (IU), vengono estratte informazioni dal testo degli URL stessi. Questa analisi comprende sia elementi strutturali, come il tipo di protocollo utilizzato, il dominio, i sottodomini, il percorso, la porta e il dominio di primo livello, sia caratteristiche statistiche come il conteggio di punti e sottodomini, oltre alla lunghezza delle parole nell'URL. Le caratteristiche basate sul contenuto (IC) vengono estratte dai dati HTML presenti nelle pagine web. Questa estrazione comprende vari aspetti, tra cui il numero e la natura dei collegamenti ipertestuali, nonché la rilevazione di eventuali contenuti o comportamenti sospetti all'interno delle pagine stesse.

L'approccio complessivo mira a fornire una classificazione più precisa dei siti di phishing, poiché combina una serie di indicatori provenienti sia dalle caratteristiche dell'URL che da quelle del contenuto delle pagine web. Questo approccio multidimensionale permette di individuare in modo più accurato potenziali siti di phishing, rendendo più efficace la protezione contro le minacce online.

0.1 Descrizione del dataset

Come mostrato in Figura 1, il diagramma illustra dettagliatamente la procedura adottata per la creazione del dataset.[1].

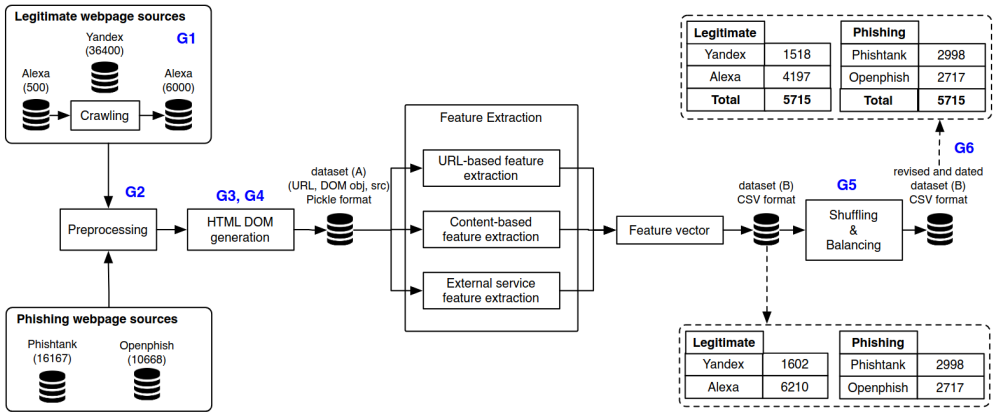


Figura 1: Source: Extract form the paper

Il dataset analizzato comprende 11.430 URL, ognuno caratterizzato da 87 attributi distinti. Questo dataset è stato sviluppato specificatamente come benchmark per sistemi di rilevamento di phishing che impiegano tecniche di apprendimento automatico. È caratterizzato da un equilibrio nella distribuzione degli URL: il 50% degli URL è di phishing e il 50% è legittimo. Va notato che la proporzione reale dei casi di phishing nella popolazione è molto inferiore, stimata intorno allo 0,01% come indicato nella fonte citata[2].

Le caratteristiche incluse nel dataset sono raggruppate in tre categorie principali:

- 56 caratteristiche sono derivate dalla struttura e dalla sintassi degli URL;
- 24 caratteristiche sono estratte analizzando il contenuto delle pagine web;
- 7 caratteristiche sono ottenute interrogando servizi esterni.

Inoltre, la variabile target di questo dataset è *status*, una variabile binaria che indica se un sito web è classificato come legittimo o come sito di phishing. Come illustrato nella tabella di

[1] ≪Cryptography and Security (cs.CR)≫. in: *Engineering Applications of Artificial Intelligence 104C (2021) 104347* (2021). DOI: https://doi.org/10.1016/j.engappai.2021.104347. URL: https://doi.org/10.48550/arXiv.2010.12847.

[2] ≪Cryptography and Security (cs.CR)≫.

distribuzione dello status (vedi Tabella 1), questa variabile è bilanciata, riflettendo la distribuzione equa degli URL di phishing e legittimi nel dataset.

Status	Train	Test
Phishing	3829	1792
Legitimate	3829	1792

Tabella 1: Distribuzione della variabile status nei dataset train e test

0.2 Preprocessing

Trasformazione variabili binarie e creazione dello score set

Prima di cominciare l'analisi è necessario fattorizzare le variabili binarie in quanto R le importa come se fossero delle variabili categoriali quando in realtà sono numeriche. Invece lo score set è necessario per l'ultimo step in cui si fa la previsione di nuove osservazioni di cui non si conosce il valore della variabile target, ovvero status. Il dataset score è stato creato considerando il 5% del test score.

Missing data

I dataset, sia per il train che il test, non presentano alcun valore mancante come evidenziato dai grafici sottostanti(rispettivamente figura 2) e figura 3).

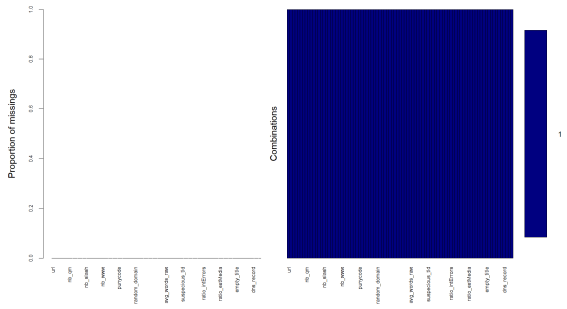


Figura 2: missing train

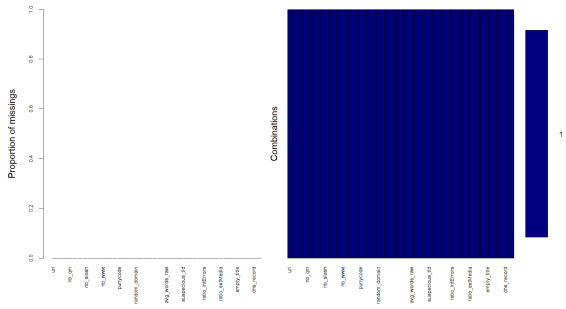


Figura 3: missing test

Rimozione variabili problematiche

Le variabili visualizzate nel grafico riferito dalla Figura 4 sono state eliminate dall'analisi in quanto caratterizzate da un'unica categoria, il che le rende non informative per il modello. Inoltre,

si è riscontrato un problema con la variabile ‘statistical_report‘: sebbene appaia con tre categorie, consultando la pubblicazione originale si è scoperto che dovrebbe essere una variabile binaria. Di conseguenza, i valori in eccesso sono considerati errati e la variabile verrà esclusa dall’insieme dei predittori.

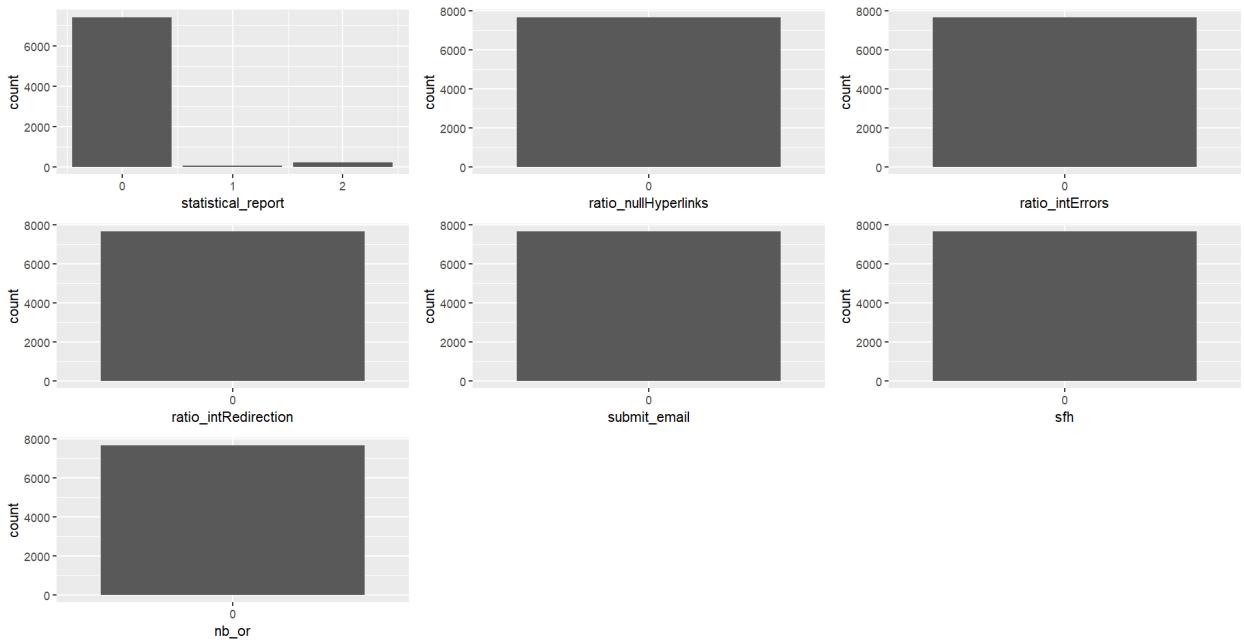


Figura 4: variabili problematiche

Model selection con Boruta

Nel processo di selezione delle variabili per l’analisi, è stato adottato l’algoritmo Boruta per la sua robustezza e capacità di fornire risultati più affidabili rispetto ad altri metodi come il Decision Tree o la Random Forest. Boruta si distingue per la sua metodologia iterativa, che valuta la rilevanza delle variabili attraverso diverse iterazioni, offrendo così una visione più approfondita dell’importanza delle variabili nel dataset. Dopo l’applicazione di Boruta, è stato creato un subset del dataset originale che include solo le 68 variabili identificate come rilevanti dall’algoritmo, inclusa la variabile target. Questo nuovo dataset sarà utilizzato per l’addestramento di modelli di machine learning che necessitano di una fase preliminare di selezione del modello, sfruttando la selezione accurata delle variabili per migliorare l’efficacia del modello.

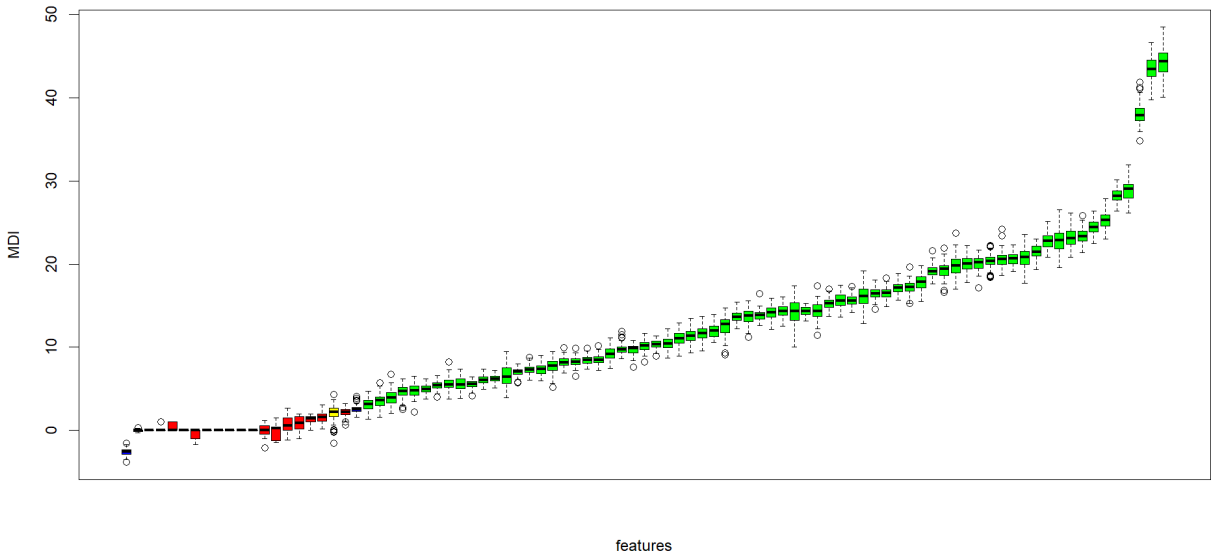


Figura 5: boruta

Si evidenziano, in particolare, tre variabili che si distinguono per importanza: ‘google_index‘, ‘page_rank‘ e ‘nb_hyperlinks‘.

Feature	meanImp	medianImp	minImp	maxImp	decision
google_index	43.3923540	43.407850030	40.473236069	46.661588	Confirmed
page_rank	42.6068700	42.653352067	39.644587797	45.602508	Confirmed
nb_hyperlinks	37.2520280	37.315795790	34.500929467	40.575561	Confirmed
domain_age	28.83370859	28.731570436	25.735769316	31.050372	Confirmed
web_traffic	28.28171636	28.291790564	26.563648663	30.035131 0	Confirmed
nb_www	25.30681692	25.329973224	23.114021043	27.378201	Confirmed

Tabella 2: Importanza delle Variabili secondo Boruta

Verifica separation per variabili problematiche

Durante l’analisi della distribuzione di tutte le variabili in relazione alla variabile obiettivo, è emerso che ‘page_rank‘ e ‘google_index‘ presentano il fenomeno della separation. In particolare, il ‘page_rank‘ riflette il posizionamento di una pagina nei risultati di ricerca di Google: si è osservato che le pagine con un rank molto basso tendono a essere poco referenziate da Google e spesso associabili a siti di phishing, mentre quelle che appaiono nelle prime posizioni sono generalmente

considerate legittime. Per quanto riguarda ‘google_index‘, questo assume valore 1 quando un sito è indicizzato da Google e 0 in caso contrario. I siti non indicizzati sono spesso quelli che Google ha identificato come di phishing e, di conseguenza, non sono visibili nei risultati di ricerca.

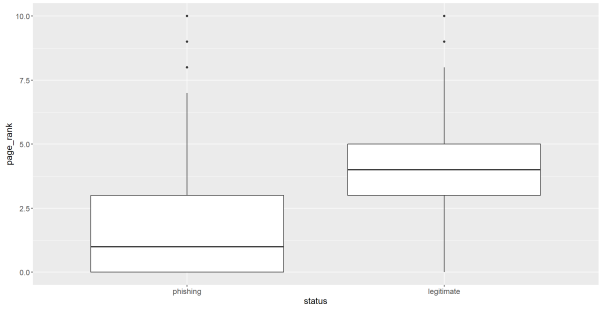


Figura 6: page rank

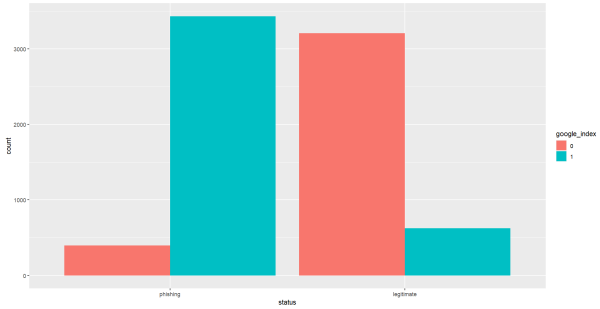


Figura 7: google index

1. Capitolo 1

Nella fase preparatoria all'addestramento di un modello di machine learning, è di cruciale importanza la scelta degli iperparametri ottimali. Tali parametri, definiti prima dell'addestramento, hanno un impatto significativo sul processo di apprendimento e sulle prestazioni del modello. Ciò li distingue dai parametri del modello, che vengono invece appresi direttamente dai dati. Per determinare la migliore combinazione di iperparametri si utilizza un processo noto come Grid Search. Questo metodo esamina sistematicamente tutte le possibili combinazioni di iperparametri all'interno di una griglia di valori predefinita. La tecnica di Validazione Incrociata, in particolare quella k-fold, è un complemento fondamentale alla Ricerca a Griglia. Il dataset viene diviso in k parti di dimensioni uguali. L'addestramento avviene su k-1 di queste parti, mentre la validazione si svolge sulla parte rimanente, ripetendo il processo per k volte. Questa strategia garantisce una valutazione approfondita e solida delle prestazioni del modello, assicurando che ogni set di iperparametri venga testato su diverse suddivisioni del dataset. Il risultato finale è un modello meticolosamente ottimizzato, che mira a garantire prestazioni generalizzate e affidabili su dati mai visti precedentemente.

1.1 Metrica per il Tuning dei modelli

Nella presente analisi, la metrica scelta per il tuning degli iperparametri è la **sensitivity** (nota anche come recall o tasso di veri positivi). Tale scelta risponde all'esigenza di minimizzare i falsi negativi. Nel contesto del rilevamento del phishing, un falso negativo si verifica quando un sito di phishing viene erroneamente classificato come legittimo. Questo errore è potenzialmente molto costoso perché permette ai siti malevoli di rimanere non rilevati, esponendo gli utenti a rischi significativi, come frodi o furto di identità. Al contrario, un falso positivo in questo contesto si verifica quando un sito legittimo viene erroneamente classificato come sito di phishing. Sebbene ciò possa causare disagio e necessiti di ulteriori verifiche, il costo associato a questo tipo di errore è generalmente inferiore rispetto alle gravi conseguenze di un falso negativo. Pertanto, in un ambito dove le conseguenze di non rilevare un sito di phishing sono molto gravi, è strategico orientare il modello a essere "eccessivamente cauto", prediligendo la sensibilità. Ciò significa che il modello sarà più incline a segnalare un sito come potenziale phishing, riducendo così il rischio di lasciare passare siti pericolosi non rilevati, anche a costo di avere un numero maggiore di falsi positivi.

GLM-Logistic Regression

DATASET

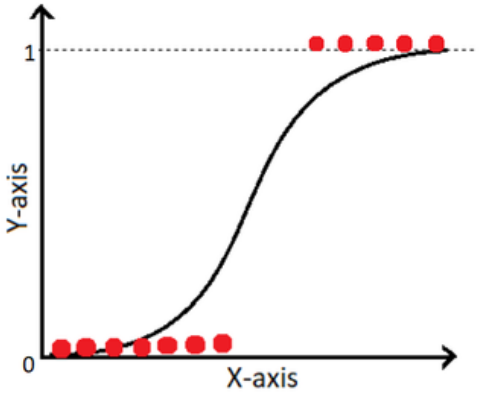
- Dataset: Train Selected
- N° Features: 68 features

PRE-PROCESSING

- Missing Values
- Standardizzazione
- Collinearity

TUNING PARAMETERS

No Tuning Parameters



MODEL EVALUATION

	Phishing	Legitimate
Phishing	32.36	3.48
Legitimate	314.82	3233.34

- **Accuracy Train:** 0.9033
- **Accuracy Test:** 0.9111
- **Sensitivity Test:** 0.9029
- **Specificity Test:** 0.9112

KNN- K Nearest Neighbors

DATASET

- Dataset: Train Selected
- N° Features: 68 features

PRE-PROCESSING

- Missing Values
- Standardization
- Collinearity

TRAIN CONTROL

- Method: Cross-Validation
- N° Fold: 10
- Search: Grid

TUNING PARAMETERS

K= n° neighbors

k	ROC	Sens	Spec
5	0.9660138	0.9028461	0.9396716
8	0.9710350	0.8900489	0.9417611
11	0.9723355	0.8843021	0.9420222
14	0.9722781	0.8741152	0.9456769
17	0.9719753	0.8707223	0.9451547
20	0.9720307	0.8696807	0.9459393

MODEL EVALUATION

	Phishing	Legitimate
Phishing	32.3	3.54
Legitimate	243.54	3304.62

Accuracy Train: 0.9521

Accuracy Test: 0.9310

Sensitivity Test: 0.9012

Specificity Test: 0.9313

Lasso

DATASET

- Dataset: Train
- N° Features: 79 features

PRE-PROCESSING

- Missing Values
- Standardizzazione

TRAIN CONTROL

- Method: Cross-Validation
- N° Fold: 10
- Search: Grid

TUNING PARAMETERS

λ= effect of regularization

λ	ROC	Sens	Spec
0.00	0.9840610	0.9206007	0.9589948
0.01	0.9792895	0.8913531	0.9720531
0.02	0.9748977	0.8592293	0.9772784
0.03	0.9704626	0.8250167	0.9845919
0.04	0.9680024	0.8119585	0.9864196
0.05	0.9673261	0.7962879	0.9924255

MODEL EVALUATION

	Phishing	Legitimate
Phishing	32.36	3.48
Legitimate	314.82	3233.34

Accuracy Train: 0.9030

Accuracy Test: 0.9111

Sensitivity Test: 0.9039

Specificity Test: 0.9112

PLS

DATASET

- Dataset: Train
- N° Features: 79 features

PRE-PROCESSING

- Missing Values
- Standardizzazione

TRAIN CONTROL

- Method: Cross-Validation
- N° Fold: 10
- Search: Grid

TUNING PARAMETERS

K= n° PLS scores

ncomp	ROC	Sens	Spec
1	0.9300563	0.7819276	0.9004962
2	0.9436718	0.8621034	0.8772491
3	0.9607444	0.8793426	0.9109421
4	0.9645633	0.8840430	0.9117241
5	0.9664970	0.8863956	0.9169508
6	0.9659573	0.8863943	0.9156453
7	0.9659282	0.8890066	0.9164279
8	0.9659595	0.8887462	0.9177341
9	0.9661260	0.8874407	0.9182563
10	0.9661601	0.8884851	0.9185174

MODEL EVALUATION

	Phishing	Legitimate
Phishing	31.44	4.4
Legitimate	364.32	3183.84

Accuracy Train: 0.8931

Accuracy Test: 0.8971

Sensitivity Test: 0.8772

Specificity Test: 0.8973

Naive Bayes

DATASET

- Dataset: Train Selected
- N° Features: 68 features

PRE-PROCESSING

- Collynearity
- Zero-Frequency

TRAIN CONTROL

- Method: Cross-Validation
- N° Fold: 10
- Search: Grid

TUNING PARAMETERS

laplace = 0
adjust = 1
usekernel = True

usekernel	ROC	Sens	Spec
FALSE	0.9106876	0.6873819	0.9044120
TRUE	0.9507623	0.6194708	0.9681373

MODEL EVALUATION

	Phishing	Legitimate
Phishing	24.3	11.54
Legitimate	344.32	3203.64

Accuracy Train: 0.8966

Accuracy Test: 0.9029

Sensitivity Test: 0.6780

Specificity Test: 0.9029

Tree

- Dataset: Train
- N° Features: 87 features

DATASET

PRE-PROCESSING

- Look at Strong Predictors but no pre-processing

- Method: Cross-Validation
- N° Fold: 10
- Search: Grid

TRAIN CONTROL

TUNING PARAMETERS

Pruning

N	CP	nsplit	rel error	xerror	xstd
24	0.00104465918	51	0.126664926	0.1619222	0.006234153
25	0.00091407678	65	0.111256203	0.1600940	0.006201942
26	0.00078349438	67	0.109428049	0.1553931	0.006118008
27	0.00065291199	75	0.103160094	0.1556542	0.006122714
28	0.00060938452	77	0.101854270	0.1535649	0.006084926
29	0.00052232959	81	0.099242622	0.1559154	0.006127414
30	0.00045703839	117	0.080438757	0.1559154	0.006127414
31	0.00043527466	121	0.078610603	0.1559154	0.006127414
32	0.00039174719	126	0.076260120	0.1569600	0.006146165
33	0.00037723804	152	0.065552364	0.1593105	0.006188064

	Phishing	Legitimate
Phishing	33.2	2.64
Legitimate	308.88	3239.28

MODEL EVALUATION

Accuracy Train: 0.9457

Accuracy Test: 0.9130

Sensitivity Test: 0.9263

Specificity Test: 0.9129

Bagging

DATASET

- Dataset: Train
- N° Features: 87 features

PRE-PROCESSING

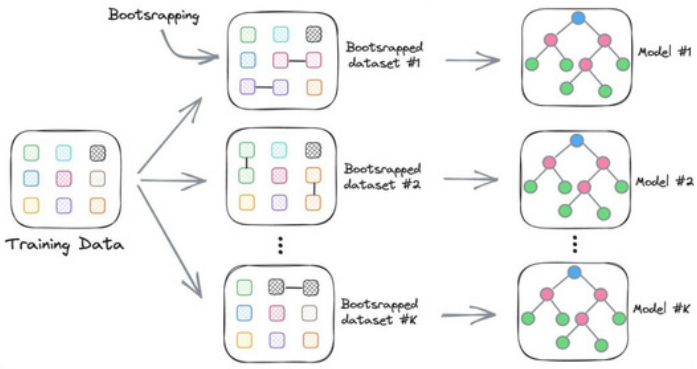
- Look at Strong Predictors but no pre-processing

TRAIN CONTROL

- Method: Cross-Validation
- N° Fold: 10
- Search: Grid

TUNING PARAMETERS

ntree (Number of Trees) = 250



MODEL EVALUATION

	Phishing	Legitimate
Phishing	33.84	2
Legitimate	227.7	3320.46

Accuracy Train: 0.9831

Accuracy Test: 0.9359

Sensitivity Test: 0.9441

Specificity Test: 0.9358

Gradient Boosting

DATASET

- Dataset: Train
- N° Features: 87 features

PRE-PROCESSING

- Look at Strong Predictors but no pre-processing

TRAIN CONTROL

- Method: Cross-Validation
- N° Fold: 10
- Search: Grid

TUNING PARAMETERS

- **n.trees** = **500:** Sets 500 trees in the model.
- **nteraction.depth** = **4:** Limits trees to 4 levels deep.
- **shrinkage** = **0.1:** Controls learning speed.
- **n.minobsinnode** = **10:** Minimum 10 samples per tree node.

MODEL EVALUATION

	Phishing	Legitimate
Phishing	34.5	1.34
Legitimate	180.18	3367.98

Accuracy Train: 0.9796

Accuracy Test: 0.9488

Sensitivity Test: 0.9642

Specificity Test: 0.9486

Random Forest

DATASET

- Dataset: Train
- N° Features: 87 features

PRE-PROCESSING

- Look at Strong Predictors but no pre-processing

TRAIN CONTROL

- Method: Cross-Validation
- N° Fold: 10
- Search: Grid

TUNING PARAMETERS

mtry = n° of features considered for deciding a split at each tree node.

mtry	ROC	Sens	Spec
2	0.9892418	0.9482878	0.9618703
10	0.9928586	0.9623918	0.9673568
19	0.9923899	0.9616085	0.9621335
28	0.9917313	0.9605655	0.9616113
37	0.9913790	0.9600427	0.9613488
45	0.9909784	0.9595205	0.9600433
54	0.9908690	0.9597822	0.9613481
63	0.9903206	0.9582150	0.9600433
72	0.9899625	0.9589983	0.9613481
81	0.9898104	0.9563852	0.9618724

MODEL EVALUATION

	Phishing	Legitimate
Phishing	34.22	1.62
Legitimate	182.16	3366

Accuracy Train: 0.9931

Accuracy Test: 0.9492

Sensitivity Test: 0.9542

Specificity Test: 0.9492

Stacking

DATASET

- Dataset: Train
- N° Features: 87 features

PRE-PROCESSING

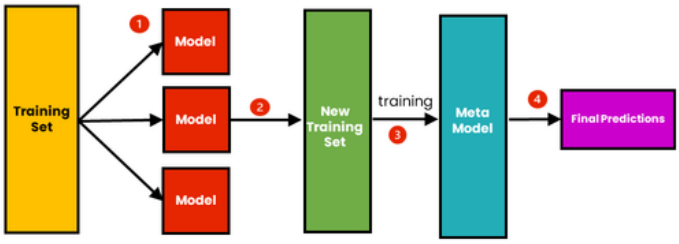
- No pre-processing

TRAIN CONTROL

- Method: Cross-Validation
- N° Fold: 10
- Search: Grid

TUNING PARAMETERS

Meta-Classifier= Logistic Regression



Models: glm, knn, naive_bayes, rf

MODEL EVALUATION

	Phishing	Legitimate
Phishing	34.28	1.56
Legitimate	168.3	3379.86

Accuracy Train: 0.9267

Accuracy Test: 0.9526

Sensitivity Test: 0.9564

Specificity Test: 0.9526

NNet - Neural Network

DATASET

- Dataset: Train Selected
- N° Features: 68 features

PRE-PROCESSING

- Collinearity
- Missing
- Standardisation
- No Zero Variance

TRAIN CONTROL

- Method: Cross-Validation
- N° Fold: 10
- Search: Grid

TUNING PARAMETERS

Single-hidden-layer neural network

Size	Decay	ROC	Sens	Spec
1	0.001	0.9559084	0.8999713	0.8986692
1	0.300	0.9602811	0.8999706	0.8978859
2	0.001	0.9596799	0.8866383	0.9083325
2	0.300	0.9664662	0.9083312	0.9059799
3	0.001	0.9621736	0.9093742	0.8973644
3	0.300	0.9702908	0.9203423	0.9177368
4	0.001	0.9625276	0.9093715	0.9044113
4	0.300	0.9681096	0.8771465	0.9206061
5	0.050	0.9712337	0.9245253	0.9174757
5	0.100	0.9713112	0.9185174	0.9226970
5	0.300	0.9736938	0.9273974	0.9221720

MODEL EVALUATION

	Phishing	Legitimate
Phishing	33.6	2.24
Legitimate	271.26	3276.9

Accuracy Train: 0.9428

Accuracy Test: 0.9237

Sensitivity Test: 0.9375

Specificity Test: 0.9235

2. Assessment

L'obiettivo di questa fase è la valutazione delle performance classificative dei vari classificatori precedentemente addestrati. A tal fine, vengono introdotte misure di valutazione che consentono di confrontare i classificatori, risolvendo il cosiddetto problema della soglia. Non è possibile confrontare direttamente i modelli osservando misure come sensitivity o accuracy, poiché sono calcolate per un singolo valore di soglia, qualsiasi esso sia. In altre parole, le misure di valutazione non devono dipendere dalla soglia.

In prima istanza, viene utilizzato il metodo delle curve ROC. Le curve ROC mostrano come varia la probabilità di corretta classificazione degli eventi (quindi True Positive Rate o sensitivity) al variare della errata classificazione dei non eventi (quindi False Positive Rate o 1-specificity) per ciascuna soglia (point of ROC). La crescita della curva indica la rapidità della corretta classificazione degli eventi, minimizzando gli errori sui non eventi (1-specificity). Si osservi che nella prima parte della curva ROC corrispondono soglie alte delle posteriors, mentre nella seconda parte corrispondono soglie basse delle posteriors.

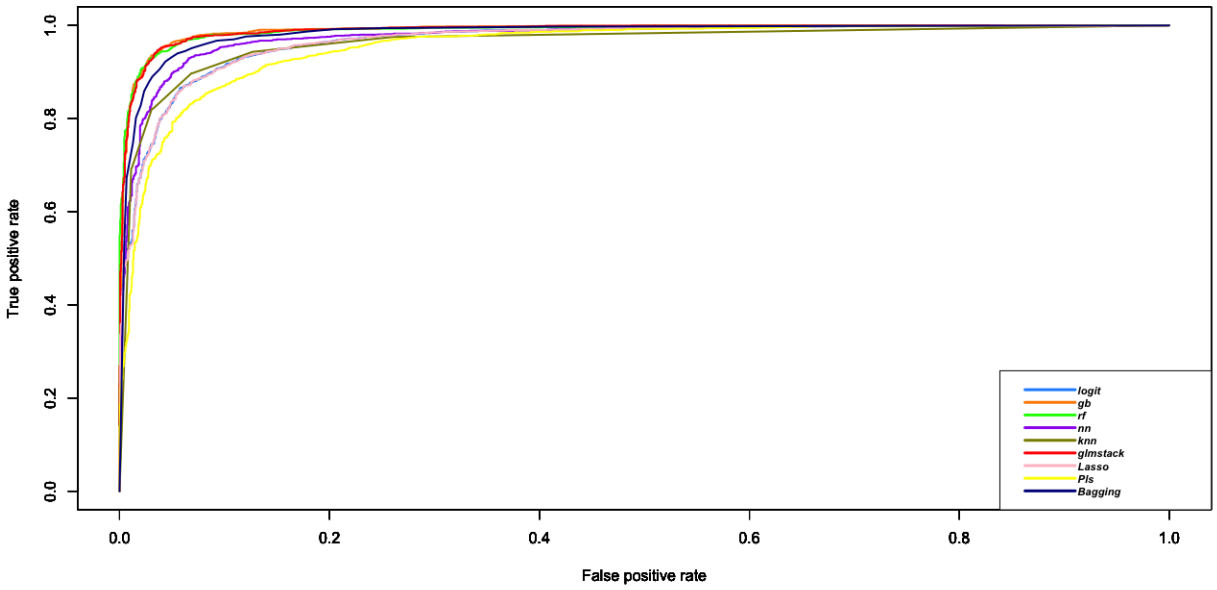


Figura 2.1: ROC Curves

Dall'analisi della figura 2.1, si evince che la maggior parte dei classificatori addestrati mostra una buona o almeno discreta capacità di classificazione degli eventi, con una bassa percentuale di

misclassificazione dei non-eventi. Tuttavia, il grafico non è particolarmente chiaro nel distinguere un modello superiore, poiché nessuna curva specifica di un modello si distingue nettamente sull'intero asse X per tutte le soglie considerate. Per meglio valutare la capacità discriminante dei modelli, si ricorre all'Area Under ROC Curve (AUC), che varia tra 0.5 e 1. La confronto delle AUC, presentato nella tabella 2.1, evidenzia la difficoltà nella selezione di un modello vincente, dato che la maggior parte dei classificatori raggiunge risultati eccellenti anche in termini di AUC.

Algoritmi	**AUC_values**	**Algoritmi**	**AUC_values**
Logit	0.9669143	GLM Stack	0.9892422
Gradient Boosting	0.9879352	Lasso	0.9668068
Random Forest	0.9891493	PLS	0.9545042
Neural Network	0.9751982	Bagging	0.9832734
KNN	0.9631318		

Tabella 2.1: Valori AUC per diversi algoritmi

Attraverso la figura 2.2, è possibile identificare i modelli che sembrerebbero essere in competizione reale. Il grafico ha la finalità di evidenziare tra tutte le curve ROC quelle che presentano le migliori performance. Identifichiamo pertanto tre modelli a "parimerito": Gradient Boosting, Random Forest e Stacking.

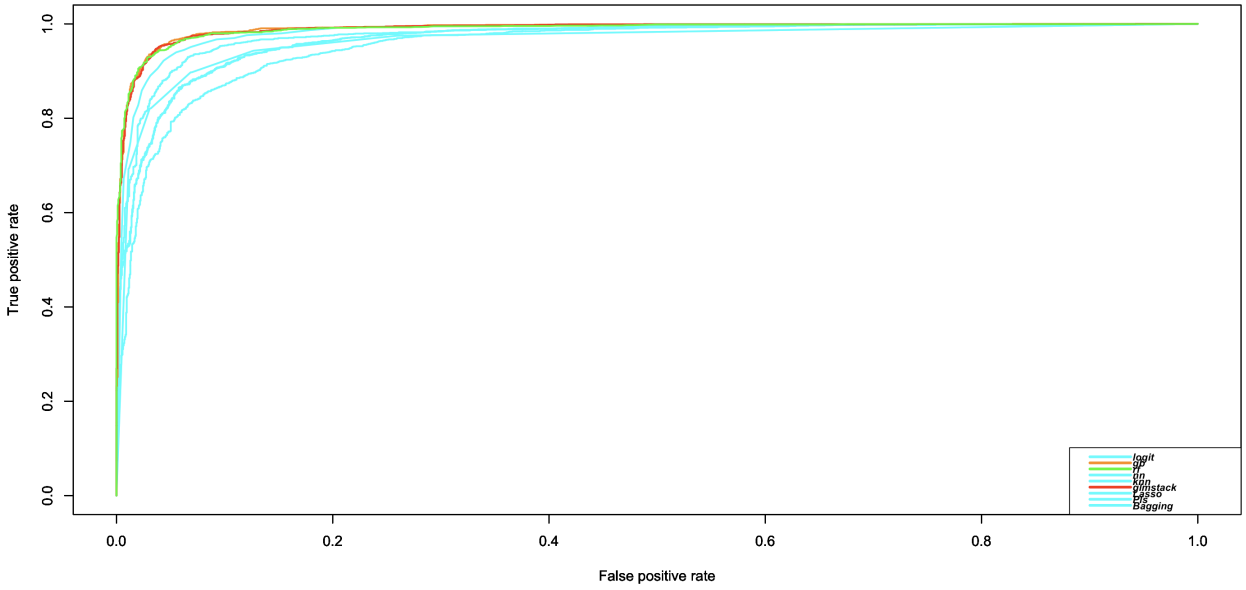


Figura 2.2: Focus on ROC curves

In situazioni di tale natura, si ricorre alle curve Lift. Le curve Lift, costituiscono uno strumento di valutazione delle performance dei modelli di classificazione. In generale, una curva Lift viene generata confrontando la percentuale cumulativa di eventi correttamente classificati dal modello con la percentuale attesa sotto condizioni casuali. Questo rapporto viene rappresentato graficamente in funzione di una variabile, spesso la percentuale di campioni esaminati. Una curva Lift ben posizionata indica che il modello supera una scelta casuale, mentre una curva Lift prossima alla linea di riferimento denota una performance modesta.

- **Curve di Lift - Random Forest:** Nei primi tre decili, il modello cattura il 59.9% dei veri URL di phishing; su tale decile, la curva di Lift presenta un punteggio pari a 2.

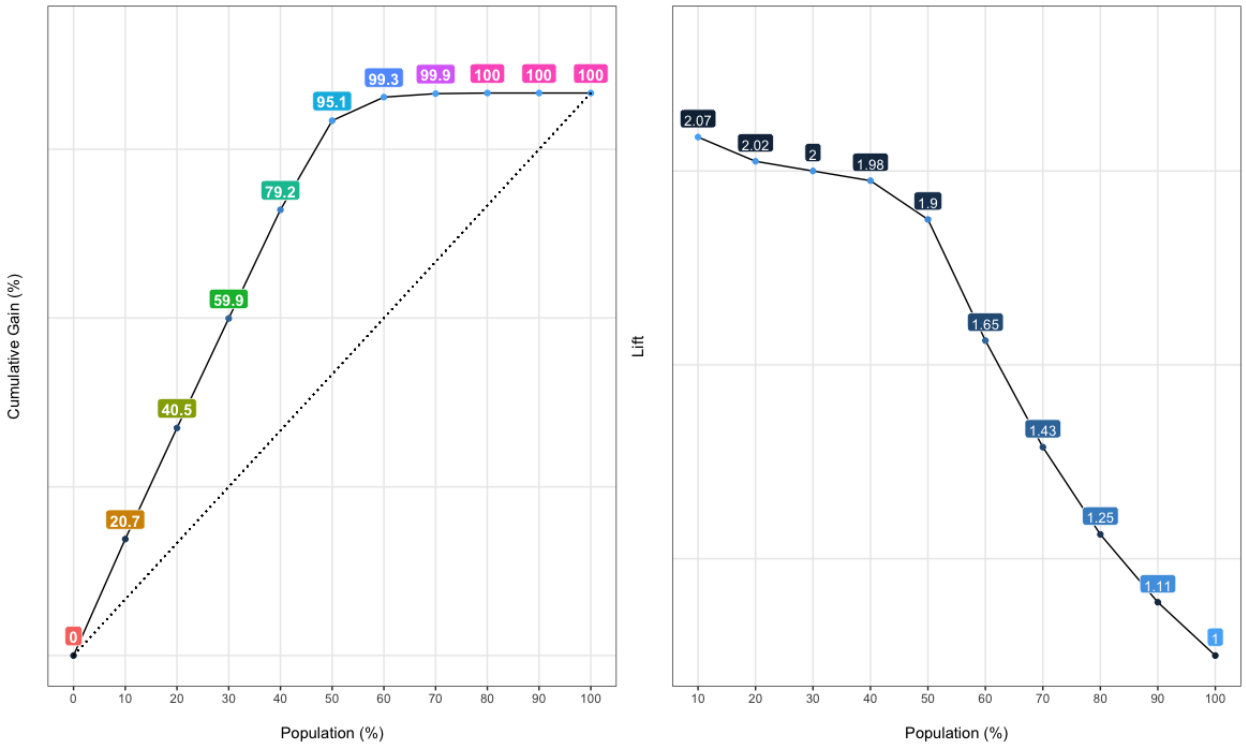


Figura 2.3: LIFT-RF

- **Curve di Lift - Gradient Boosting:** Nei primi tre decili, il modello cattura il 59.8% dei veri URL di phishing; su tale decile, la curva di Lift presenta un punteggio pari a 1.99.

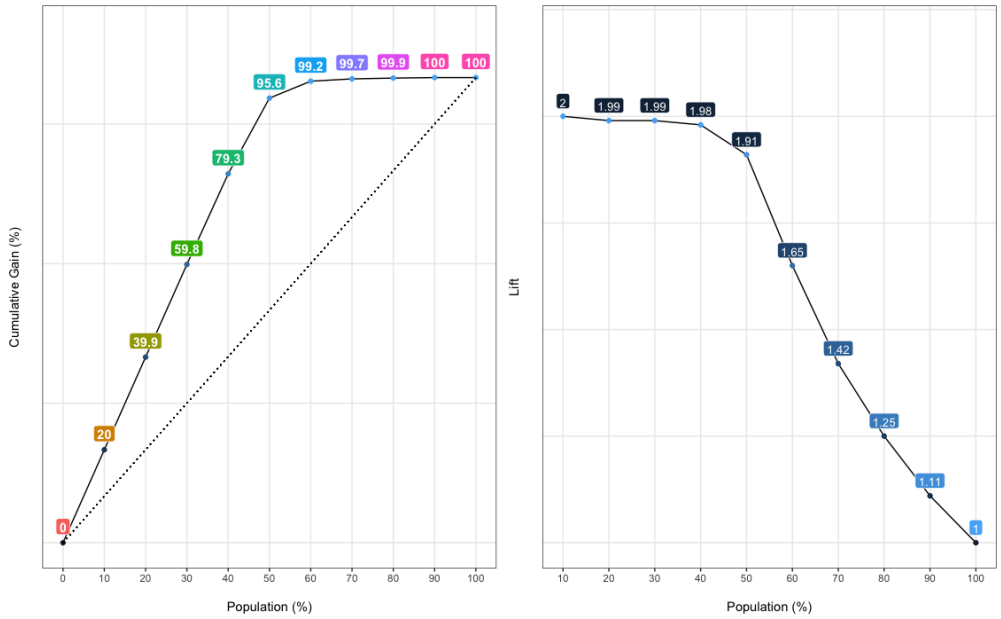


Figura 2.4: LIFT-GB

- **Curve di Lift - Stacking:** Nei primi tre decili, il modello cattura il 59.8% dei veri URL di phishing; su tale decile, la curva di Lift presenta un punteggio pari a 1.99.

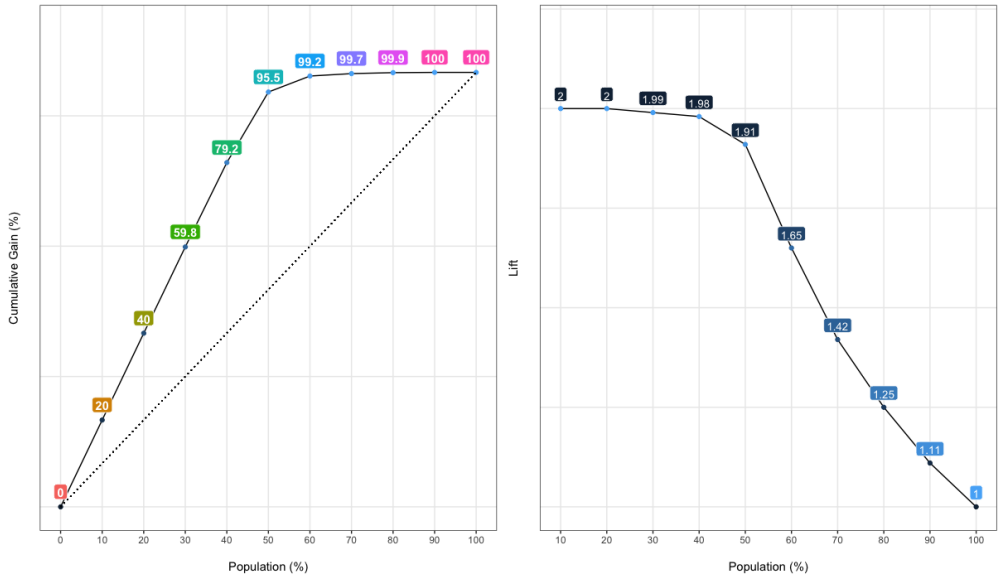


Figura 2.5: LIFT-STACKING

In base ai risultati ottenuti dall'analisi, il modello che evidenzia la migliore capacità di catturare veri URL di phishing nei primi tre decili, risulta essere Random Forest.

3. Scelta della soglia

In questo step, si è proceduto all'analisi delle prestazioni classificative del modello migliore, enfatizzando la necessità di identificare una soglia ottimale per l'applicazione ai nuovi dati. Tale soglia è stata definita con l'intento primario di ottimizzare la sensibilità del modello più performante identificato, la Random Forest. Per il raggiungimento di questo obiettivo, si è ricorso all'impiego di metodologie basate su criteri statistici rigorosi, volti alla massimizzazione della sensibilità all'interno del modello selezionato.

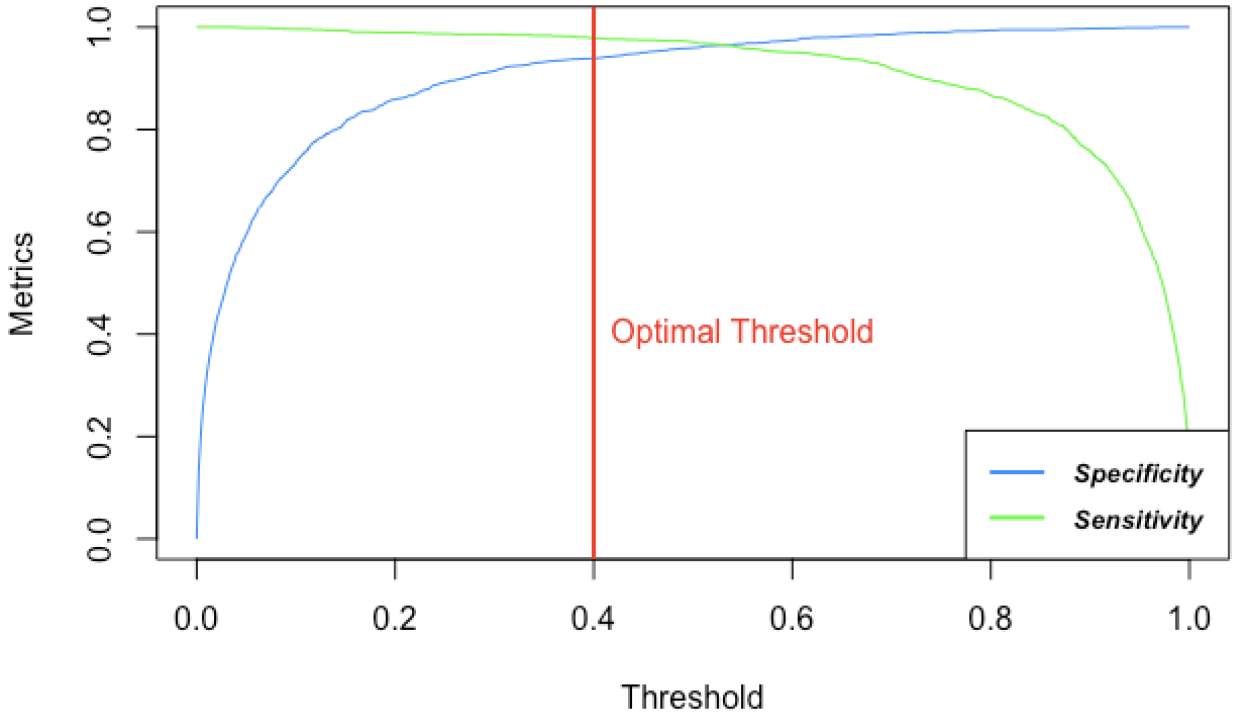


Figura 3.1: scelta della soglia

Dall'analisi del grafico è emerso che il valore di soglia ottimale è 0.4. Questa determinazione si fonda sull'osservazione che tale valore corrisponde a un elevato livello di sensitivity, cruciale per assicurare previsioni accurate dei siti di phishing. In corrispondenza di questa soglia, si notano valori significativi anche per la specificity.

	Predetto Phishing	Predetto Legitimate
Reale Phishing	34.88	0.96
Reale Legitimate	273.24	3274.92

Tabella 3.1: Matrice di confusione

La matrice di confusione, una volta aggiustata con la nuova soglia di 0.4, mostra un incremento nei valori, in particolare per la sensitivity.

Metrica	Valore
Accuracy	0.9235
Sensitivity	0.9732
Specificity	0.9229

Tabella 3.2: Metriche con soglia 0.4

Metrica	Valore
Accuracy	0.9492
Sensitivity	0.9542
Specificity	0.9492

Tabella 3.3: Metriche con soglia 0.5

Variabili più importanti con la Random Forest

Le variabili che si sono rivelate importanti per il modello di Random Forest sono illustrate nella Figura 3.2.

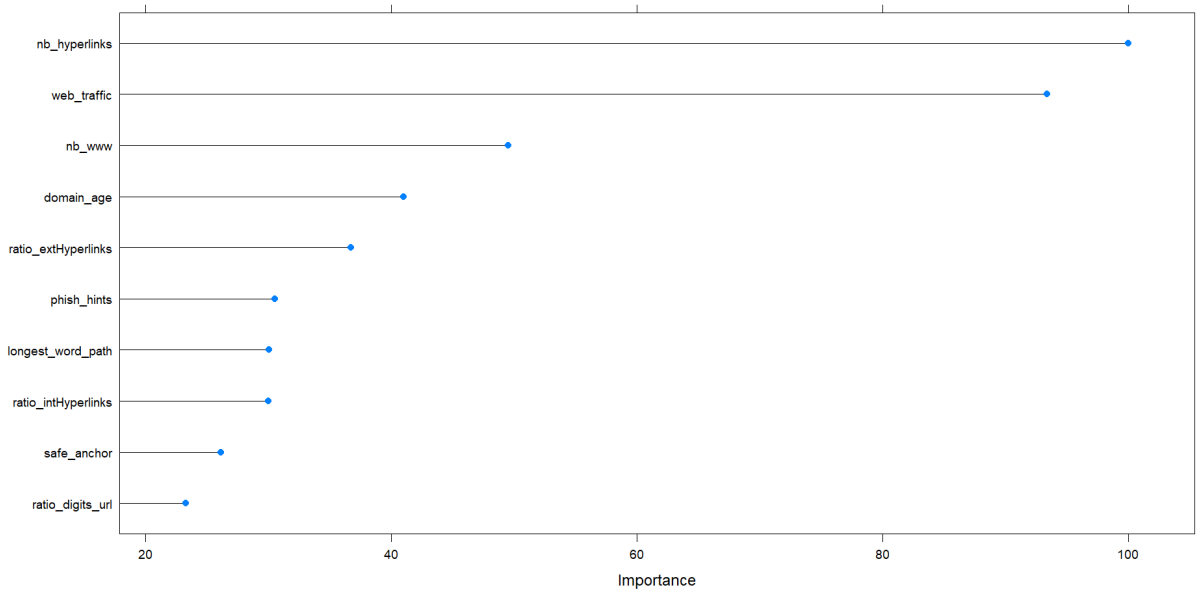


Figura 3.2: Variable importance

4. Score di nuovi casi

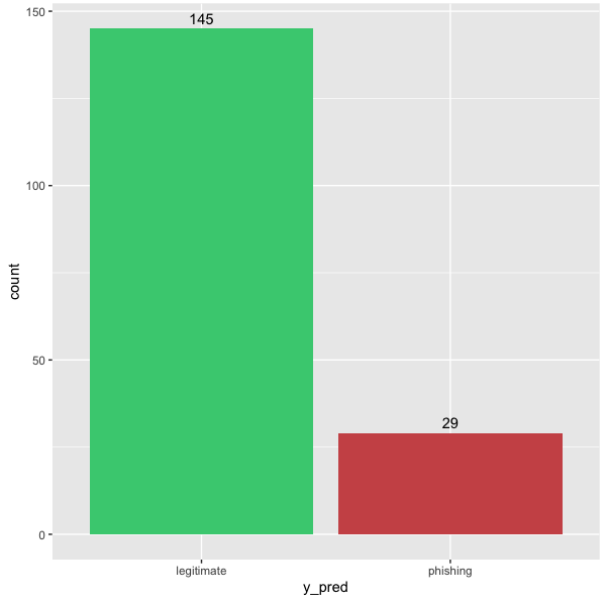


Figura 4.1: Score Set

I risultati dell'analisi nello step precedente sono soddisfacenti, quindi verrà utilizzato lo score set creato all'inizio del progetto composto da 174 osservazioni.

Successivamente, verrà applicata la soglia(0.4) ottenuta allo step precedente ai nuovi casi.

Alla fine su 174 osservazioni 145 sono state classificate come legitimate e 29 come phishing.

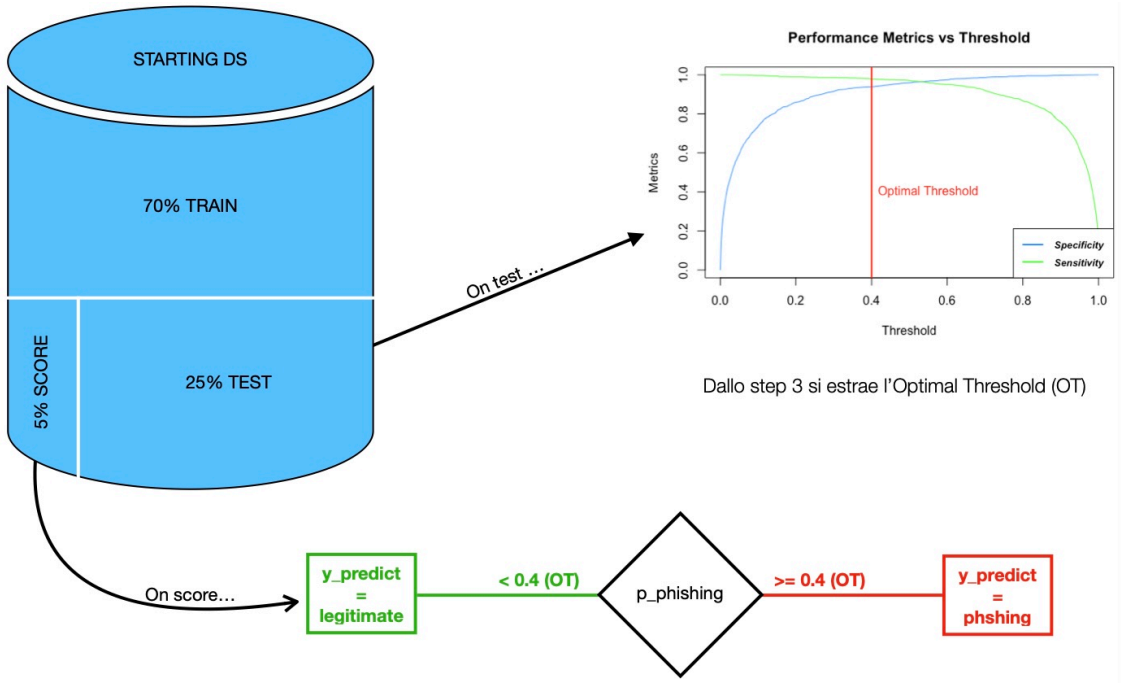


Figura 4.2: Step 4 at Glance

Project check-points

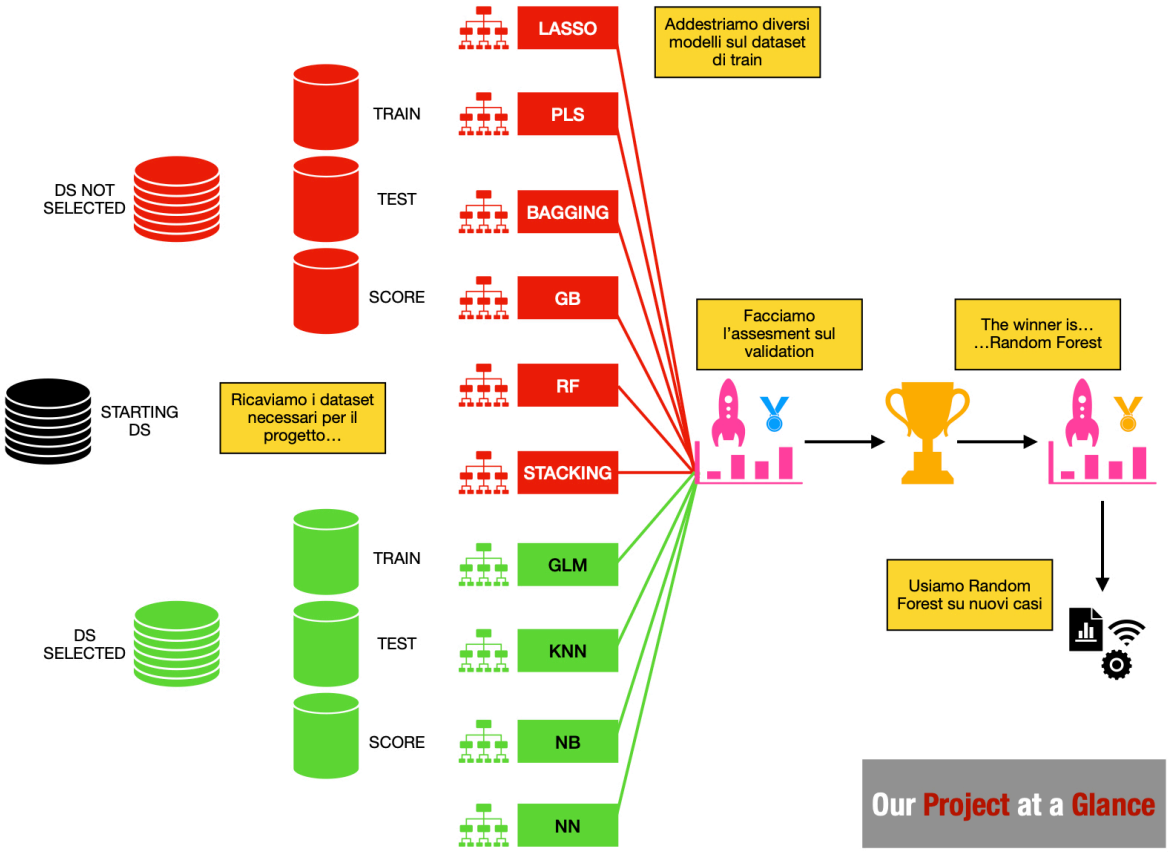


Figura 4.3: Project check-points

L'ipotesi di utilizzare un modello di Random Forest addestrato per l'identificazione di siti di phishing in contesti aziendali potrebbe fornire un contributo significativo al miglioramento della sicurezza informatica. Si potrebbe ipotizzare che, integrando questo modello in un sistema di protezione della posta elettronica, si possa ottenere un metodo efficace per contrastare uno dei vettori di attacco più comuni e insidiosi, ovvero il phishing via email.

In questa ipotetica implementazione, il modello sarebbe integrato con i sistemi di posta elettronica aziendali, fornendo una soluzione proattiva per rilevare e neutralizzare le minacce di phishing. L'analisi in tempo reale dei link contenuti nelle email potrebbe permettere di identificare immediatamente i tentativi di phishing, prevenendo danni prima che possano verificarsi. Questo approccio

si distingue per la sua potenziale efficienza, in quanto ridurrebbe la necessità di interventi manuali da parte del personale IT, automatizzando la rilevazione delle minacce.

Tips per individuare siti di phishing

- Le pagine legittime di solito utilizzano collegamenti ipertestuali con lo stesso dominio di base del sito web, mentre le pagine di phishing utilizzano più collegamenti ipertestuali esterni che puntano a siti web target. Il rapporto tra collegamenti ipertestuali interni ed esterni delle pagine web è considerato un indicatore di phishing.
- I siti web di phishing generalmente hanno un numero minore di visitatori rispetto ai siti legittimi. Alexa viene utilizzato per identificare il traffico web degli URL.
- Poiché i siti web di phishing sono di breve durata, l'età dei domini URL è considerata un indicatore di phishing.
- Si presume che i siti web legittimi consistano di un numero maggiore di pagine rispetto a quelli di phishing. Pertanto, il numero di collegamenti nei contenuti delle pagine web degli URL è considerato per distinguere i siti web di phishing.
- Termini comuni negli URL come 'www' sono utilizzati solo una volta negli URL legittimi, mentre si osserva che sono utilizzati più di una volta negli URL di phishing.

Bibliografia

≪Cryptography and Security (cs.CR)≫. In: *Engineering Applications of Artificial Intelligence 104C (2021) 104347* (2021). DOI: https://doi.org/10.1016/j.engappai.2021.104347. URL: https://doi.org/10.48550/arXiv.2010.12847.