Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de la Computación
Security Data Science

08/02/2025
Eunice Mata, 21231
Semestre II - 2025

Laboratorio 1
Detección de Phishing

Parte 1

Exploración de datos

5 Observaciones

	url	status
0	http://www.crestonwood.com/router.php	legitimate
1	http://shadetreetechnology.com/V4/validation/a...	phishing
2	https://support-appleId.com.secureupdate.duila...	phishing
3	http://rgipt.ac.in	legitimate
4	http://www.iracing.com/tracks/gateway-motorspo...	legitimate

Observaciones de las etiquetas

```
status
legitimate    5715
phishing      5715
Name: count, dtype: int64
status
legitimate    0.5
phishing      0.5
```

La cantidad de tipo de datos están balanceada entre los 2 tipos (legitimate y phishing)

Derivación de características

¿Qué ventajas tiene el análisis de una URL contra el análisis de otros datos, como el tiempo de vida del dominio, o las características de la página Web?

La URL es un dato ya está disponible antes de que el usuario haga clic en el enlace, permitiendo detección de un dominio phishing. No requiere cargar la página web completa y posiblemente exponerse a una descarga no autorizada o evita ejecutar código malicioso que podría estar en la página y protege el dispositivo de malware embebido.

Al tener la URL no dependemos de que el sitio esté activo o sea accesible, ya que los sitios de phishing suelen tener vida corta y pueden estar caídos al momento de querer realizar un análisis. Además, los atacantes pueden registrar dominios con anticipación para "envejecerlos" o usar dominios comprometidos, por lo que

el tiempo de vida de un dominio no necesariamente garantiza la legitimidad del mismo.

¿Qué características de una URL son más prometedoras para la detección de phishing?

Según los artículos analizados, las características más prometedoras son cosas como el uso de una dirección IP en lugar de un dominio, el largo del URL (las de phishing tienden a ser más largas), cantidad de puntos tanto en la url en general, en el dominio en sí y subdominios, existencia de frecuente de caracteres especiales, aleatoriedad de los caracteres, paths muy largos con múltiples parámetros son sospechosos y top level domais gratuitos comunes en phishing.

Atributos del URL a agregar

Generales

- length_url - Longitud total de la URL
- num_dots_url - Cantidad de puntos "." en la URL
- num_hyph_url - Cantidad de guiones "-" en la URL
- num_slash_url - Cantidad de barras "/" en la URL
- num_special_chars_url - Total de caracteres especiales (@, ?, =, &, !, %, $, #, +, *, ~)

Específicas del dominio

- length_dom - Longitud del dominio
- num_dots_dom - Cantidad de puntos en el dominio (subdominios)
- dom_in_ip - Si el dominio está en formato IP (binaria)
- num_tld_url - Longitud del TLD (top level domain)

Específicas del pathname

- length_path - Longitud del path/pathname
- num_hyph_path - Cantidad de guiones en el path

De parámetros

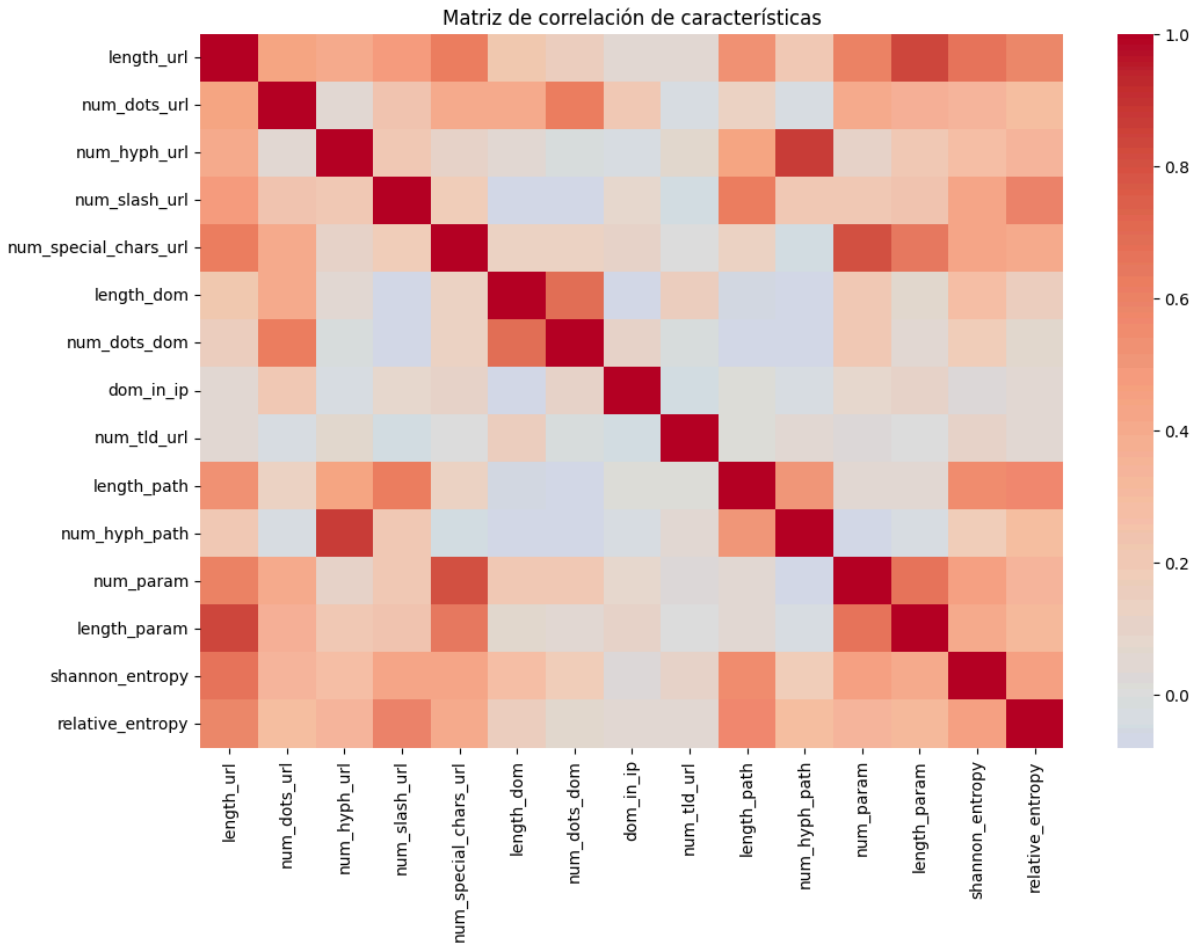
- num_param - Número de parámetros (después de "?")
- length_param - Longitud total de los parámetros

Características de Entropía

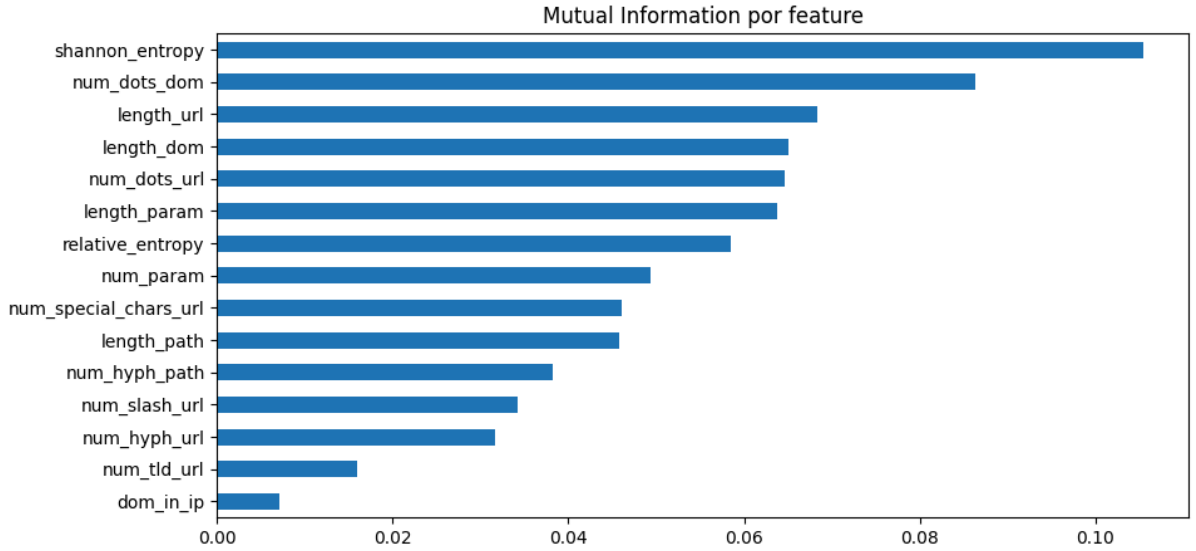
- shannon_entropy - Entropía de Shannon de la URL
- relative_entropy - Entropía relativa (KL divergence)

Preprocesamiento y Selección de Características

Para el preprocesamiento del dataset, se realizaron funciones de lectura del url para obtener 15 atributos que se determinaron anteriormente. Y se pensaba eliminar los que tuvieran un 0.90 o más de correlación con otra variable, para evitar redundancia, pero no se encontró mayor relación entre valores.



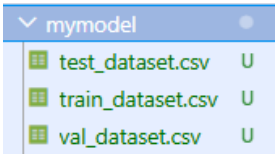
Y se usó mutual information respecto a la variable objetivo, ya que capturan relaciones no lineales comunes en URLs phishing y se eligieron las primera 13 variables en orden.



Parte 2

Separación de datos

```
Train: (6285, 13)
Val: (1715, 13)
Test: (3429, 13)
```



Implementación

Anteriormente se realizó la definición de 1 como phishing y 0 como legitimate. Y La matriz tiene esta forma:

$$\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

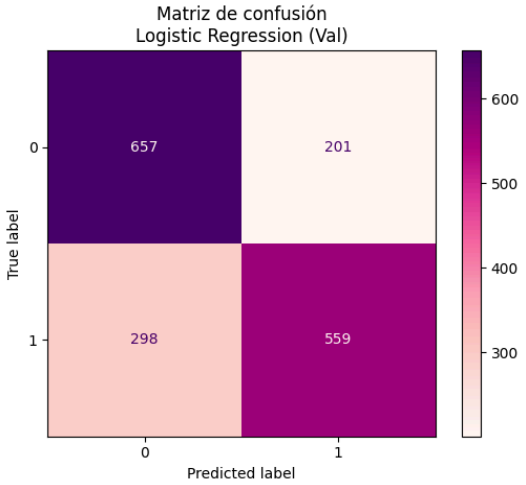
Donde:

- TN: legítimos bien clasificados
- FP: legítimos clasificados como phishing (falsas alarmas)
- FN: phishing clasificados como legítimos (peligroso)
- TP: phishing bien detectados

Resultados

Validación de Regresión Logística

Matriz de confusión de validación de Regresión Logística



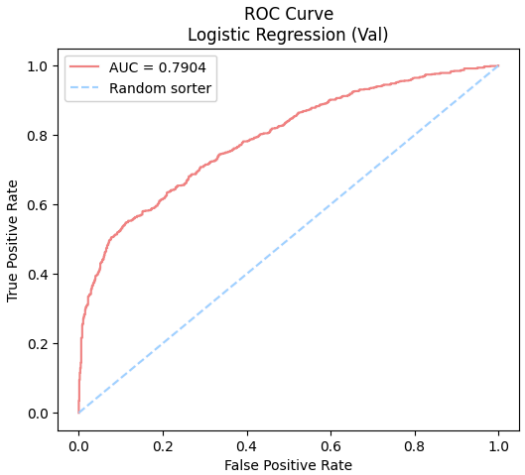
- TN = 657
- FP = 201
- FN = 298
- TP = 559

Precisión: 0.7355

Recall: 0.6523

De todas las URLs que el modelo predijo como phishing, el 73.6% realmente eran phishing. Entonces cada vez que la alarma suena 1 de entre 4 casos el modelo se equivoca. De todos los phishing reales, el modelo detectó el 65.2%, implica que 34.8% de phishing se le escapan.

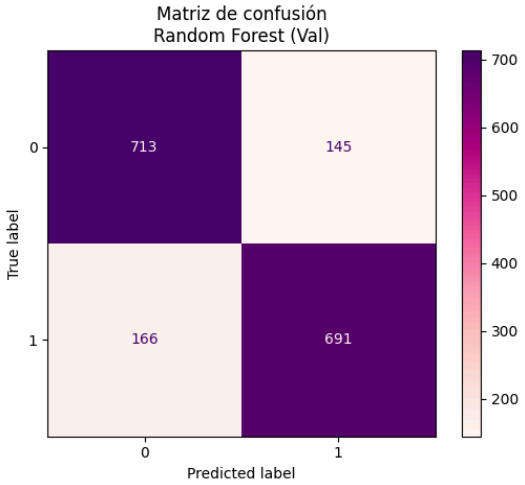
ROC de la validación de Logistic Regression



El ROC AUC es de 0.79, lo cual es decente y es mejor que el modelo que clasifica al azar, pero no se usaría en casos reales.

Validación de Random Forest

Matriz de confusión de validación de Random Forest

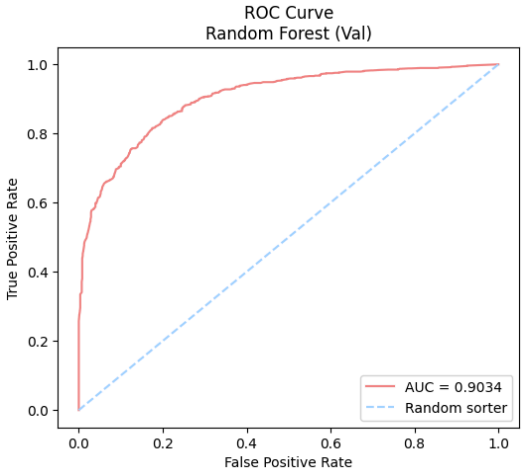


- TN = 713
- FP = 145
- FN = 166
- TP = 691

Precisión: 0.8266
Recall: 0.8063

Cuando el modelo predice phishing, acierta el 82.7%, menos que la regresión lineal. Detecta el 80.6% del phishing real, lo cual es una mejora significativa.

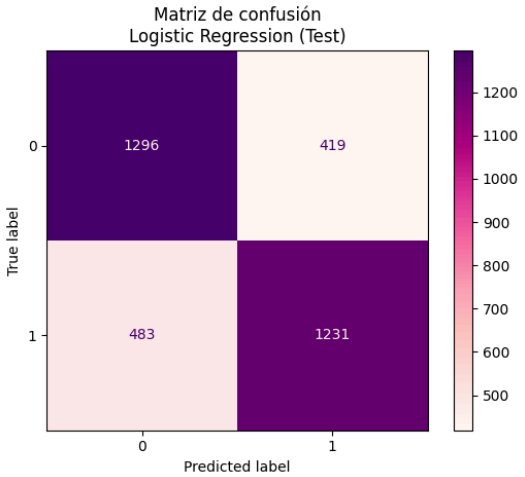
ROC de la validación de Random Forest



Con un ROC AUC de 0.90 es buen desempeño, y generalmente significa que el modelo separa bastante bien URLs legítimas vs phishing

Prueba de Regresión Logística

Matriz de confusión de la Regresión Logística

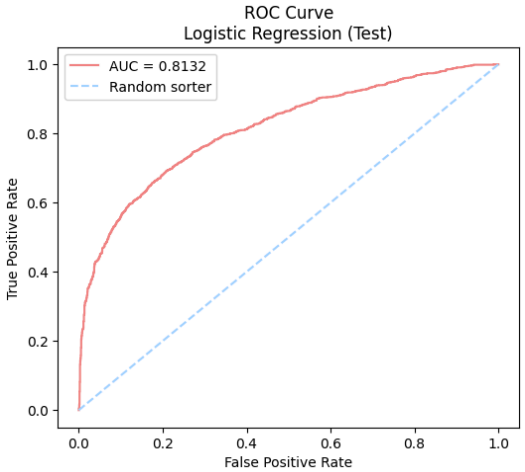


- TN = 1296
- FP = 419
- FN = 483
- TP = 1231

Precisión: 0.7461
Recall: 0.7182

De todas las alarmas de phishing, el 74.6% eran correctas y el modelo detectó el 71.8% del phishing real. Aún se le escapa casi 28% del phishing, que sigue siendo bastante.

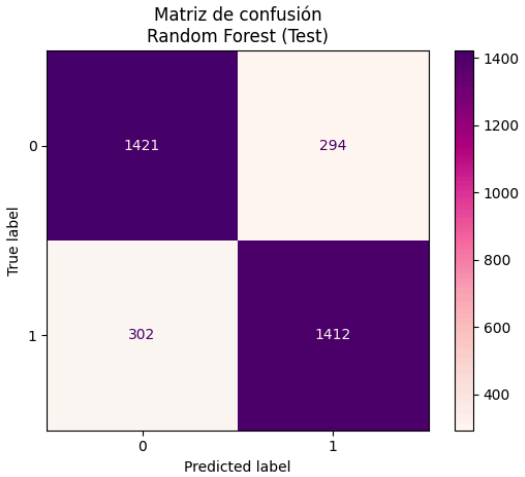
ROC de prueba de Regresión Logística



El ROC es mejor que el de validación, pero todavía está por debajo del Random Forest tanto en la validación como en la prueba.

Prueba de Random Forest

Matriz de confusión de la prueba de Random Forest

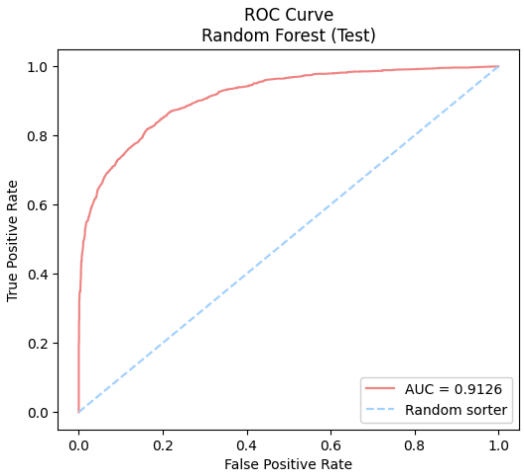


- TN = 1421
- FP = 294
- FN = 302
- TP = 1412

Precisión: 0.8277
Recall: 0.8238

Cuando Random Forest predice phishing, acierta un 82.8% del tiempo y detecta el 82.4% del phishing real. Lo que indica que tan mejor es Random Forest.

ROC de la prueba de Random Forest



Con un ROC AUC de 0.9126 se confirma que el modelo tiene una separación entre phishing y real bastante buena.

Discusión

¿Cuál es el impacto de clasificar un sitio legítimo como phishing?

El impacto real que esto puede provocar es que un usuario no pueda acceder a un sitio legítimo, puede causar pérdida de productividad en empresas y generar fatigas de alertas, si hay muchas alarmas falsas, los usuarios empiezan a ignorarlas. Estos falsos positivos podrían causar que el modelo sea ignorado.

¿Cuál es el impacto de clasificar un sitio de phishing como legítimo?

El impacto real, el cual es bastante peligroso, puede ser que los usuarios entreguen sus credenciales, se comprometan cuentas, que los usuarios sean víctimas de fraude o robo de identidad, y haya una posible entrada a ataques mayores como descarga malware o ransomware.

En base a las respuestas anteriores, ¿Qué métrica elegiría para comparar modelos similares de clasificación de phishing?

Para minimizar los falsos negativos (que phishing sea detectado como legítimo) se compararía el recall de cada modelo. Porque un recall alto significa que el modelo detecta más phishing real. Aún así, tampoco se quieren tantos falsos positivos, entonces se debe de tomar en cuenta precisión, para determinar qué tan eficiente es el modelo.

¿Qué modelo funcionó mejor para la clasificación de phishing? ¿Por qué?

El modelo que funcionó mejor fue Random Forest, debido a los siguientes valores:

Valores en la prueba	Random Forest	Regresión Logística
Recall	0.8238	0.7182
Precisión	0.8277	0.7461

AUC	0.9126	0.8132
FN	294	419
FP	302	483

Y la razón por la que es mejor es debido a que Random Forest captura patrones no lineales y combinaciones entre features, por ejemplo entropía + número de parámetros + longitud del dominio, mientras que Logistic Regression es lineal.

Una empresa desea utilizar su mejor modelo, debido a que sus empleados sufren constantes ataques de phishing mediante e-mail. La empresa estima que, de un total de 50,000 emails, un 15% son phishing. ¿Qué cantidad de alarmas generaría su modelo? ¿Cuántas positivas y cuántas negativas? ¿Funciona el modelo para el BR propuesto? En caso negativo, ¿qué propone para reducir la cantidad de falsas alarmas?

Cantidades reales

- Phishing reales
  $50000 * 0.15 = 7500$
- Legítimos reales
  $50000 - 7500 = 42500$

Recall = 0.8238

$$TP \approx 0.8238 * 7500 = 6178.5 \approx 6179$$
$$FN = 7500 - 6179 = 1321$$

Se detectarían aproximadamente 6,179 phishing y se ne escaparían 1,321 phishing, lo cual todavía es bastante.

Precisión = 0.8277
Precision = TP / (TP + FP)

$$TP + FP = \frac{TP}{Precisión}$$
$$TP + FP = \frac{6179}{0.8277} \approx 7465$$
$$FP = 7465 - 6179 = 1286$$

Generaría un aproximado de 1,286 falsas alarmas, con un total de 7465 alarmas.

De 50,000 emails, tendría los siguientes resultados:
Predicho phishing:

- TP = 6,179 correctas
- FP = 1,286 falsas

Alarmas negativas:

- FN = 1,321 phishing que pasan
- TN = 41,214 legítimos correctos

El modelo no funciona para el BR propuesto. Esto es porque un total de 1,286 falsas alarmas en 50,000 emails puede ser pesado para la persona que los revisa. Además, 1,321 phishing escapados siguen siendo alto para una empresa.

Para reducir la cantidad de falsas alarmas se podría ajustar el threshold, si se sube el threshold, se tendría menos FP pero más FN, si se baja se detectaría más phishing pero sube también el FP, la decisión dependería de la empresa. Usar más modelos, un segundo modelo que filtre el FP para reducirlo. Entrenar con un dataset más realista o más grande, ya que el dataset actual tiene un la misma cantidad de urls phishing y legítimas.