Universidad del Valle de Guatemala
Security Data Science
Adrian Rodríguez - 21691
Daniel Gómez - 21429

Laboratorio 1 - *Phishing Detection*

Repositorio
https://github.com/adrianrb469/security-ds/tree/main/lab1

Parte 1 - Ingeniería de Características

Exploración

1. Cargue el dataset en un dataframe de pandas, muestre un ejemplo de cinco 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

2. Muestre la cantidad de observaciones etiquetadas en la columna status como “legit” y como “phishing”. ¿Está balanceado el dataset?

```
Number of legitimate websites: 5715
Number of phishing websites: 5715
```

Sí.

Derivación de Características

1. ¿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?

- Velocidad y Eficiencia: No requiere acceder a la página web en sí (lo cual puede ser lento o imposible si la página ya no existe) ni obtener información de terceros (como la edad del dominio). Esto permite la detección en tiempo real (zero-hour detection).

- Independencia del Contenido: El análisis de la URL no depende del contenido de la página web, que puede ser escaso, engañoso o estar oculto intencionalmente por los atacantes.

- Costo: El acceso a la URL es libre, mientras que algunas APIs requieren costos adicionales.

- Disponibilidad: La URL siempre está presente, incluso si el sitio web ya no está activo. Esto permite analizar incluso ataques que ya han sido desactivados.

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

1. Entropía de Shannon de Caracteres No Alfanuméricos: Medida de la aleatoriedad en el uso de caracteres especiales.
2. Entropía Relativa de Distribución de Caracteres: Cuantifica la diferencia entre la distribución de caracteres de la URL y una distribución "normal" de sitios legítimos.
3. Número de Puntos en la URL: Cantidad de "." en la URL; puede indicar subdominios excesivos.
4. Número de Guiones en la URL: Cantidad de "-" en la URL; usado para imitar o confundir.
5. Longitud Total de la URL: Número total de caracteres.
6. Longitud del Nombre de Dominio: Número de caracteres en el dominio.
7. Longitud de la Ruta: Número de caracteres en la ruta de la URL.
8. Presencia del Token "HTTPS": Indica si la URL usa conexión segura.
9. Cantidad de Subdominios: Número de subdominios presentes.
10. Relación Dígitos/Caracteres en el Nombre de Dominio: Proporción de dígitos respecto a letras en el dominio.
11. TLD Conocido (Lista Blanca): Verifica si el dominio de nivel superior es legítimo.
12. Cantidad de "Palabras Sensibles" Presentes: Número de palabras clave sospechosas (ej. "login").
13. Presencia de Dirección IP en el Dominio: Indica si el dominio es una dirección IP.
14. Número de Vocales en la URL: La cantidad de vocales usadas.
15. Cantidad de Caracteres Especiales Usados: El total de carácteres especiales usado.

Hemos seleccionado estas 15 características para la detección de phishing basándonos en la información y conclusiones clave de varios estudios examinados. La longitud total de la URL, la longitud del nombre de dominio, la longitud de la ruta y el número de puntos y guiones en la URL ayudan a detectar el intento de los atacantes de ofuscar la dirección web real, como señalan Calzarossa et al. y Aung & Yamana. Asimismo, se ha incluido la presencia de "HTTPS" para verificar si hay seguridad y también el número de subdominios pues los estafadores suelen usarlos mucho.

Además, la entropía de Shannon y relativa de caracteres no alfanuméricos proporciona una manera de medir la aleatoriedad y el uso inusual de caracteres especiales. Un análisis del nombre de dominio como es el TLD, y la relación dígitos caracteres nos dan seguridad del dominio, y buscamos palabras sensibles y direcciones IP para buscar técnicas de ingeniería social. Combinadas, estas características deberían permitir a nuestros modelos distinguir con mayor precisión entre URLs legítimas y de phishing.

Las funciones para extraer dichas características se encuentran en features.ipynb

Preprocesamiento

El preprocesamiento se realizó seleccionando únicamente la variable objetivo "status" junto con las siete características identificadas como más relevantes: "shannon_entropy", "url_length", "special_chars", "suspicious_words", "domain_length", "vowels_count" y "dots_count". Posteriormente, se separaron las variables predictoras (X) y la respuesta (y), y se aplicó el escalado utilizando StandardScaler, el cual transforma las variables restándoles la media y dividiéndolas por la desviación estándar. Esto estandariza las características para que tengan una media de 0 y una desviación estándar de 1, reduciendo el impacto de las diferencias de escala.. Finalmente, se dividió el conjunto de datos en tres particiones: 55% para entrenamiento, 15% para validación y 30% para prueba, y se guardaron en archivos CSV para su posterior análisis.

status	shannon_entropy	url_length	special_chars	suspicious_words	domain_length	vowels_count	dots_count
0	-0.909906	-0.580937	-0.545208	-0.394903	-0.009295	-0.672349	-0.350946
1	-0.666616	-0.400073	-0.326120	1.262124	-0.380426	-0.496797	0.379180
1	1.655221	1.770300	1.645668	-0.394903	-0.658774	1.785373	-0.350946
1	0.007481	-0.237295	-0.107033	-0.394903	-0.565991	-0.584573	0.379180
0	0.364476	0.142520	0.769318	-0.394903	-0.937123	0.117633	0.379180

Selección de Características

1. ¿Qué columnas o características fueron seleccionadas y por qué?

A partir del análisis se seleccionaron aquellas características que presentaban una varianza adecuada y un buen poder discriminativo, medido a través de la métrica ROC AUC, siendo superiores al umbral de 0.6. Se descartaron aquellas con varianza muy baja, como known_tld, digit_ratio y has_ip, que además estaban altamente correlacionadas entre sí, lo que podría inducir redundancias. Asimismo, se observó que la característica relative_entropy no aportaba discriminación suficiente (ROC AUC aproximadamente 0.53), por lo que se optó por prescindir de ella en beneficio de mantener un conjunto de indicadores robusto y diversificado.

Por otro lado, se priorizaron características que capturan distintos aspectos estructurales y textuales de las URL. Así, shannon_entropy ayuda a identificar la aleatoriedad en el contenido, mientras que url_length y domain_length proporcionan información sobre la extensión de la dirección y el dominio, respectivamente. Las características special_chars, suspicious_words, vowels_count y dots_count complementan el análisis al ofrecer detalles sobre la presencia de caracteres especiales, palabras sospechosas, cantidad de vocales y puntos, lo que puede ser indicativo de patrones anómalos en las URL. **El análisis exploratorio se encuentra en features.ipynb**

Lista final de características:

- shannon_entropy
- url_length

- special_chars
- suspicious_words
- domain_length
- vowels_count
- dots_count

Parte 2: Implementación

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

Puede bloquear el acceso a información o servicios confiables para los usuarios, o bien, generar desconfianza en el sistema, ya que los empleados o clientes pueden sentirse frustrados al encontrarse con bloqueos o alertas erróneas. Potencialmente perjudicar la reputación de la empresa si se invalida el tráfico normal de usuarios o se afecta la percepción de la marca.

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

Clasificar un sitio de phishing como legítimo (falso negativo) tiene un impacto aún más serio, ya que:

1. Permite que el ataque de phishing tenga éxito, exponiendo a los usuarios a robos de información, comprometiendo credenciales y pudiendo causar daños económicos o a la reputación.
2. Los usuarios son engañados para que interactúen con contenidos maliciosos, lo que puede derivar en la propagación de malware o pérdida de datos sensibles.

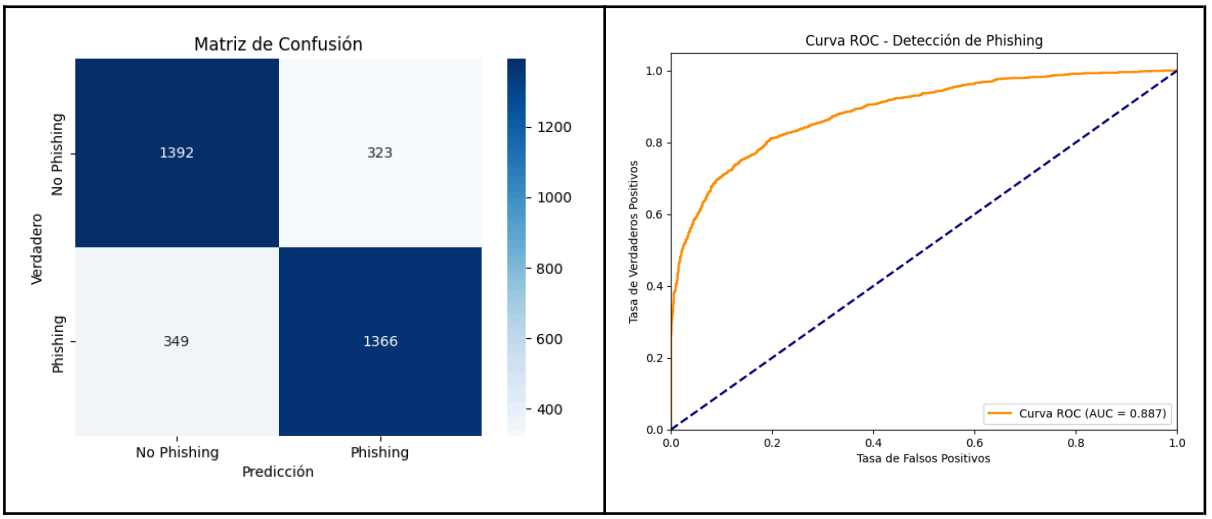
3. En base a las respuestas anteriores, ¿qué métrica elegiría para comparar modelos similares de clasificación de phishing?

Dado que el mayor riesgo radica en no detectar correctamente un sitio de phishing (falso negativo), es fundamental priorizar la sensibilidad (recall) de la clase phishing. Sin embargo, debido a que también es importante evitar la saturación de falsas alarmas, se debe equilibrar con la precisión. Por ello, una métrica combinada como el F1-score para la clase phishing o, de manera más global, el AUC-ROC (que considera el trade-off entre verdaderos positivos y falsos positivos en distintos umbrales) es muy útil para comparar modelos similares.

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

En este caso, el modelo Random Forest fue el mejor candidato. Obtuvo la mayor área bajo la curva ROC (AUC) en validación (0.891) y en el test (0.887). Su robustez y capacidad de capturar relaciones no lineales entre las características permitió un desempeño balanceado tanto en la detección de phishing como en la minimización de falsos positivos. Esto se refleja en las métricas de precisión (~0.81) y recall (~0.80), que indican un rendimiento equilibrado y aceptable para ambos tipos de error.

Cabe resaltar que se evaluaron un total de cuatro modelos: Regresión Logística, Random Forest, Gradient Boosting y SVC. Cada uno de ellos fue optimizado mediante técnicas de validación cruzada (GridSearchCV) para afinar sus hiper parámetros y garantizar la robustez en la detección de phishing.



5. Caso práctico: Aplicación del modelo en una empresa con 50,000 emails

La empresa supone que el 15% de los emails son phishing, es decir:
Emails de phishing: 50,000 × 0.15 = 7,500
Emails legítimos: 50,000 − 7,500 = 42,500

El recall es de aproximadamente 79.65%, lo que implica que el modelo detecta correctamente ≈ 7,500 × 0.80 ≈ 6,000 emails de phishing.
Esto significa que ≈ 1,500 emails phishing pasarían como legítimos (falsos negativos).

Según la matriz de confusión, se clasificaron erróneamente como phishing 323 de 1,715 casos, lo que representa una tasa de falsos positivos de ≈ 18.8%. Aplicando esta tasa a 42,500 emails legítimos, se obtienen ≈ 42,500 × 0.188 ≈ 8,000 falsos positivos.

Por lo tanto:
Alarmas positivas (marcados como phishing): ≈ 6,000 (verdaderos positivos) + 8,000 (falsos positivos) = 14,000 alarmas.
Alarmas negativas (no marcados): ≈ 50,000 − 14,000 = 36,000 emails, entre los cuales habría ≈ 1,500 phishing que no se detectaron.

¿Funciona el modelo para el BR (Business Requirement) propuesto?
Si bien el modelo detecta alrededor del 80% de los emails de phishing, la generación de aproximadamente 8,000 falsas alarmas podría ser problemática para la operativa de la empresa, ya que saturaría a los empleados con alertas en las que la mayoría son erróneas.

Propuesta para reducir las falsas alarmas:

- Ajustar el umbral de decisión: Modificar el umbral de probabilidad para clasificar un email como phishing podría mejorar la precisión (reduciendo los falsos positivos), aun sacrificando algo de recall.

- Implementar un sistema de doble verificación: Usar el modelo actual como filtro inicial y después aplicar una segunda capa de validación (por ejemplo, un análisis manual o un modelo complementario) para confirmar las alarmas.
- Cost-sensitive learning: Ajustar el entrenamiento del modelo para penalizar de forma más intensa los falsos positivos, buscando un mayor equilibrio que minimice las alarmas sin comprometer demasiado la detección de phishing.