Autores: Fabiola Contreras, 22787 María José Villafuerte, 22129	Docente: Jorge Yass Laboratorio 1
Sección: 11	Fecha: 08/02/2026

Laboratorio 1. Detección de phishing

I. Índice

II. Ingeniería de características 1

1. Exploración de datos 1
2. Derivación de características 2

1) ¿Qué ventajas tiene el análisis de una URL contra el análisis de otros datos, cómo el tiempo de vida del dominio, o las características de la página Web? 2

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

3. Preprocesamiento 4
4. Selección de características 5

3) ¿Qué columnas o características fueron seleccionadas y por qué? 5

III. Implementación de modelos 6

1. Separación de datos e Implementación 6
2. Resultados 6
3. Discusión 9

IV. Referencias 15

II. Ingeniería de características

1. Exploración de datos

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

```python
import pandas as pd

# Cargar el archivo CSV
df = pd.read_csv('dataset_pishing.csv')

# Visualizar las primeras filas
print(df.head())
```

✓ 0.0s

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

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

```
# Contar frecuencia de cada valor en la columna 'status'
frecuencias = df['status'].value_counts()
print(frecuencias)
```

✓ 0.0s

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

El dataset sí está balanceado, lo que permite al modelo aprender de forma equitativa tanto patrones de URLs legítimas como de phishing. Esta distribución evita sesgos hacia una clase dominante y facilita el entrenamiento del modelo. Sin embargo, cabe destacar que esta proporción no refleja necesariamente la distribución real del problema, donde los sitios de phishing suelen ser menos frecuentes, por lo que la evaluación del modelo debe considerar métricas adecuadas más allá de la accuracy.

2. Derivación de características

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

    - No requiere descargar el contenido de la página web, lo que evita riesgos de seguridad al acceder a sitios potencialmente maliciosos.
    - No depende de servicios de terceros (WHOIS, DNS, rankings) que pueden introducir latencia significativa.
    - Los estudios muestran que las características basadas en URL son extremadamente rápidas de extraer (~41ms según Hannousse & Yahiouche 2020). En contraste, características basadas en contenido HTML pueden tomar más de 100 segundos (especialmente features como f63 y f65 que verifican cada enlace).
    - No depende de históricos o bases de datos actualizadas, entonces puede detectar sitios de phishing nuevos que aún no están en blacklists.
    - Facilita la implementación de sistemas globales sin necesidad de procesamiento multilingüe.

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

    - Longitud de componentes como el url en su totalidad, el pathname y el dominio. Específicamente en ese orden de importancia según Calzarossa et al. 2023. Una URL más larga permite ocultar el dominio real.
    - La entropía de los caracteres, suelen usarse raras distribuciones de caracteres especiales. Aung & Yamana 2019 reportan mejora de 5-6% en ROC AUC
    - Conteo de símbolos especiales como puntos, guiones, barras.
    - La presencia de nombres de marcas, palabras sensibles (login, secure, account, update, verify, confirm) o palabras aleatorias.

Basado en esto, este es el listado de funciones implementadas para integrar características al dataset:

	Función	Descripción	Artículo de Referencia
1	**url_length**	Mide la longitud total de la URL (característica más discriminativa según estudios)	Calzarossa et al. (2023) - Quality Reliability Eng
2	**domain_length**	Calcula la longitud del nombre de dominio	Calzarossa et al. (2023) - Quality Reliability Eng
3	**path_length**	Calcula la longitud del path/pathname de la URL	Calzarossa et al. (2023) - Quality Reliability Eng
4	**count_dots**	Cuenta el número de puntos en toda la URL (dominios phishing suelen tener más)	Sahingoz et al. (2019) - Machine learning based phishing
5	**count_hyphens**	Cuenta guiones que a menudo se usan para imitar marcas legítimas	Karim et al. (2023) - Phishing Detection System
6	**count_subdomains**	Determina el número de subdominios (phishing tiende a usar más)	Karim et al. (2023) - Phishing Detection System
7	**has_ip_address**	Detecta uso de dirección IP en lugar de nombre de dominio (indicador fuerte)	Karim et al. (2023) - Phishing Detection System
8	**has_at_symbol**	Detecta el símbolo @ que puede indicar redirección maliciosa	Karim et al. (2023) - Phishing Detection System
9	**has_double_slash_redirect**	Identifica "//" en el path usado para redirecciones	Karim et al. (2023) - Phishing Detection System
10	**is_https**	Verifica uso de protocolo HTTPS (aunque 78% de phishing ahora lo usa)	Karim et al. (2023) - Phishing Detection System
11	**count_query_params**	Cuenta parámetros en query string (URLs complejas son sospechosas)	Sahingoz et al. (2019) - Machine learning based phishing
12	**has_sensitive_words**	Detecta palabras sensibles como "login", "verify", "account", etc.	Sahingoz et al. (2019) - Machine learning based phishing
13	**count_special_chars**	Cuenta todos los caracteres especiales/no alfanuméricos	Aung & Yamana (2019) - URL-based Phishing Detection
14	**digit_ratio**	Calcula proporción de dígitos (URLs con muchos números son sospechosas)	Hannousse & Yahiouche (2020) - Benchmark Datasets
15	**count_vowels_in_domain**	Cuenta vocales en el dominio (palabras aleatorias tienen menos)	Calzarossa et al. (2023) - Quality Reliability Eng

16	**has_prefix_suffix**	Detecta guiones en el dominio (técnica común: paypal-secure.com)	Karim et al. (2023) - Phishing Detection System
17	**avg_word_length_path**	Longitud promedio de palabras en el path	Sahingoz et al. (2019) - Machine learning based phishing
18	**count_ampersands**	Cuenta símbolos & (múltiples parámetros)	Calzarossa et al. (2023) - Quality Reliability Eng
19	**count_equals**	Cuenta símbolos = (parámetros)	Calzarossa et al. (2023) - QualityReliability Eng
20	**is_shortened_url**	Detecta servicios de acortamiento de URLs (bit.ly, tinyurl, etc.)	Hannousse & Yahiouche (2020) - Benchmark Datasets
21	**shannon_entropy_feature**	Entropía de Shannon de toda la URL (mide aleatoriedad)	Aung & Yamana (2019) - URL-based Phishing Detection
22	**relative_entropy_feature**	Entropía relativa respecto a distribución de referencia	Aung & Yamana (2019) - URL-based Phishing Detection
23	**count_slashes**	Cuenta barras diagonales (paths profundos son sospechosos)	Calzarossa et al. (2023) - Quality Reliability Eng

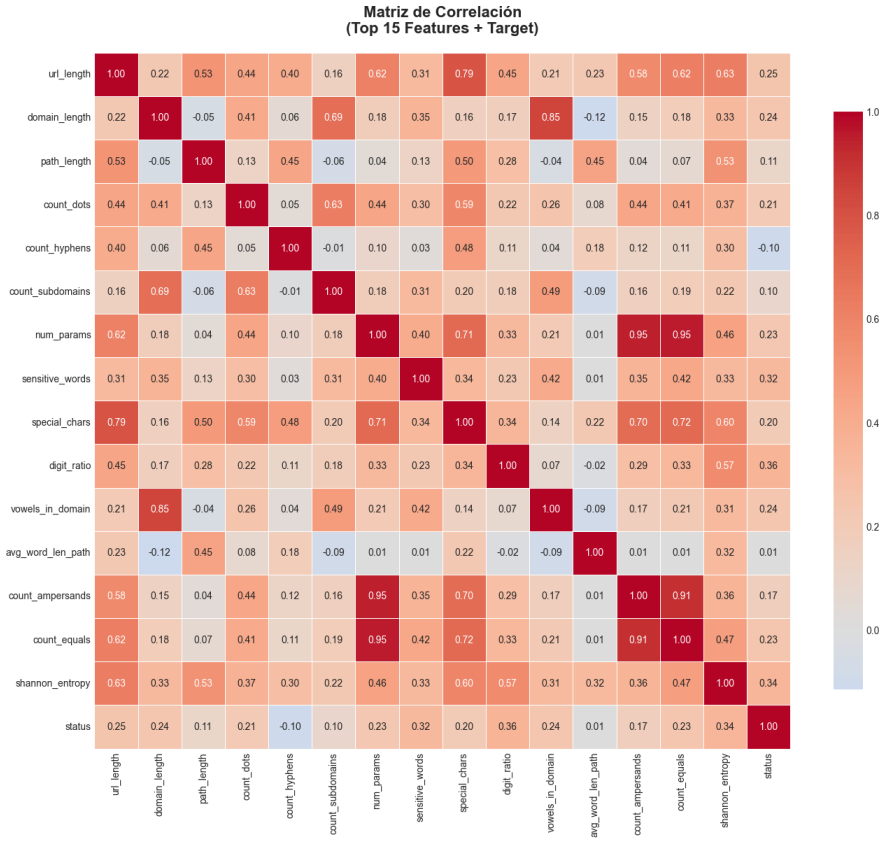
3. Preprocesamiento

En el código se realiza el siguiente preprocesado

- Convierte status a binaria (phishing=1, legitimate=0)
- Elimina la columna url (guardando backup por seguridad)
- Normaliza features continuas con StandardScaler
- Mantiene features binarias sin cambios
- Crea visualizaciones para análisis exploratorio
- Valida que no haya nulos ni infinitos al final

4. Selección de características

Visualizar correlación entre características:



3) ¿Qué columnas o características fueron seleccionadas y por qué?

Se seleccionaron las siguientes características:

- shannon_entropy
- digit_ratio
- sensitive_words
- url_length
- domain_length
- num_params
- special_chars

Y definitivamente se rechazaron las siguientes por su baja correlación:

- avg_word_len_path (~0.01)
- path_length (~0.11)
- count_hyphens (-0.10)

La selección inicial de variables se realizó considerando la correlación con la variable objetivo, evitando redundancia entre variables altamente correlacionadas. Posteriormente, se validó la relevancia mediante métricas de importancia de variables y evaluación del impacto en el desempeño del modelo.

III. Implementación de modelos

1. Separación de datos e Implementación

Puede acceder al código con el análisis realizado en el repositorio:
https://github.com/Fabiola-cc/phishing_detection

2. Resultados

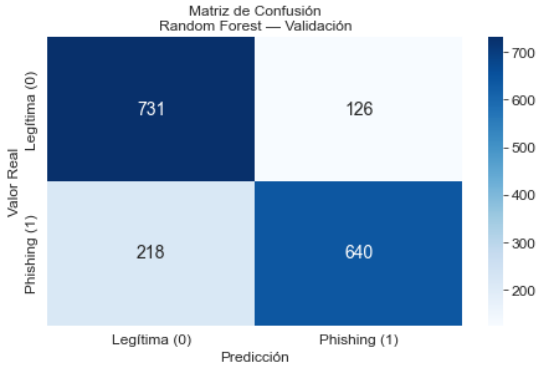
i. Modelo 1: Random Forest

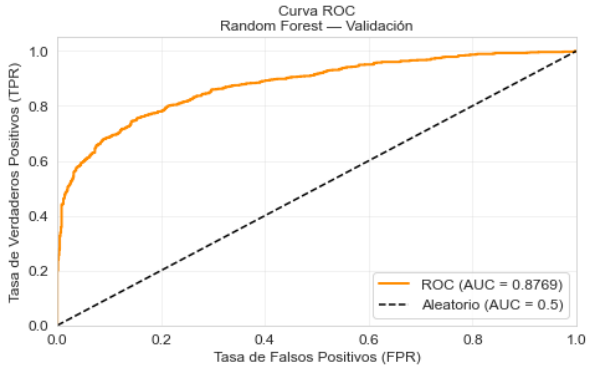
Random Forest es un ensemble de árboles de decisión que combina múltiples clasificadores mediante bagging. Es adecuado para detección de phishing porque:

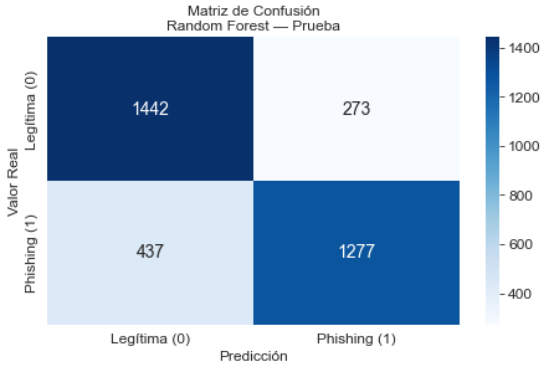
a) Captura relaciones no lineales entre las características de las URLs
b) Es resistente al overfitting gracias al promedio de múltiples árboles
c) Proporciona importancia de variables, útil para interpretar qué características discriminan mejor

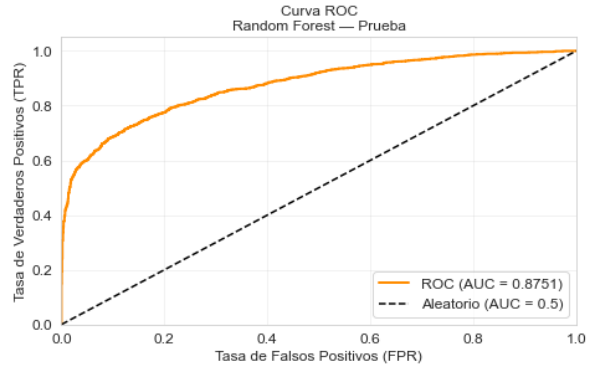
Entrenamiento y Optimización de Hiperparámetros

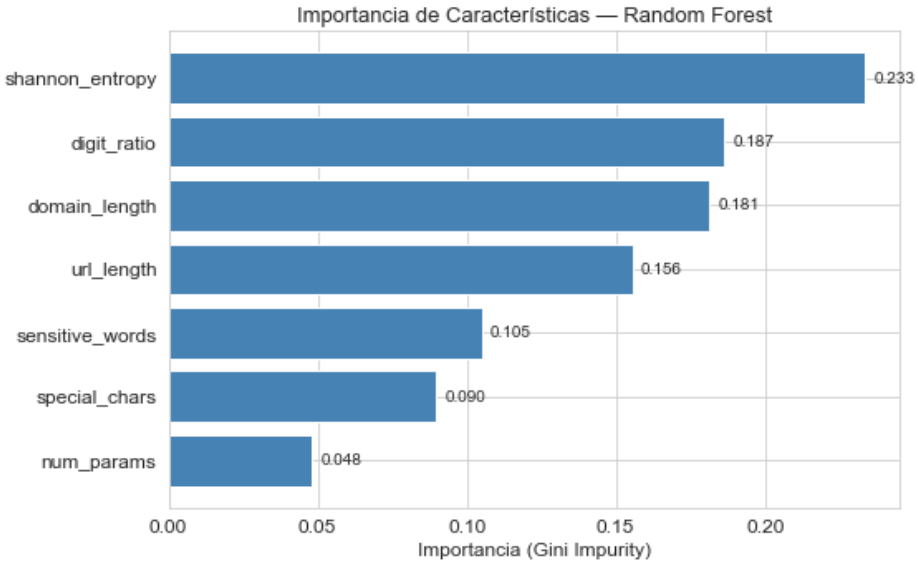
Se utiliza `GridSearchCV` para encontrar la mejor combinación de hiperparámetros, evaluando con F1-Score (métrica que balancea precision y recall).







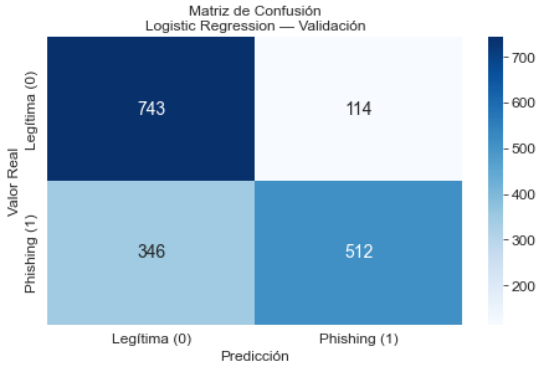


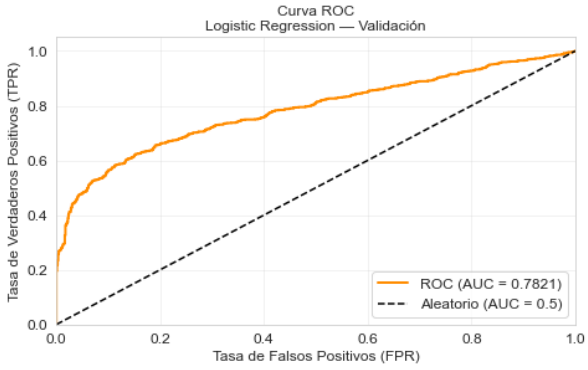


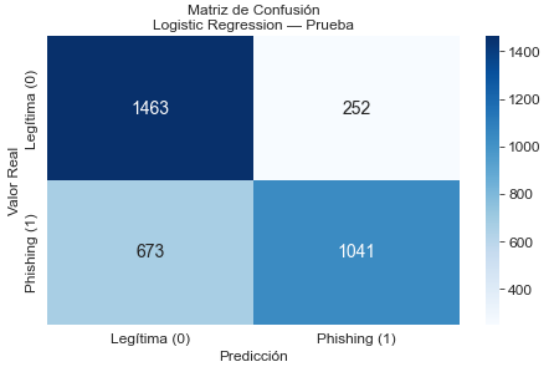
ii. Modelo 2: Logistic Regression

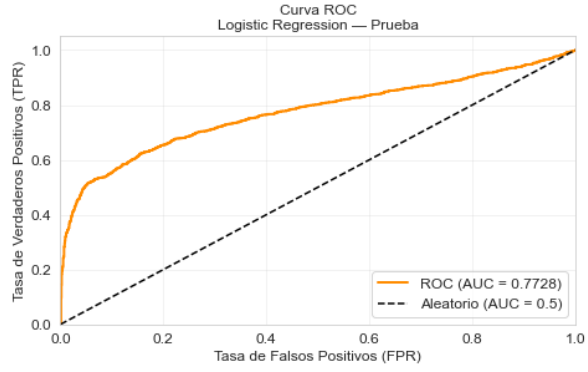
La Regresión Logística estima la probabilidad de que una URL sea phishing mediante una función sigmoide. Sus ventajas incluyen:

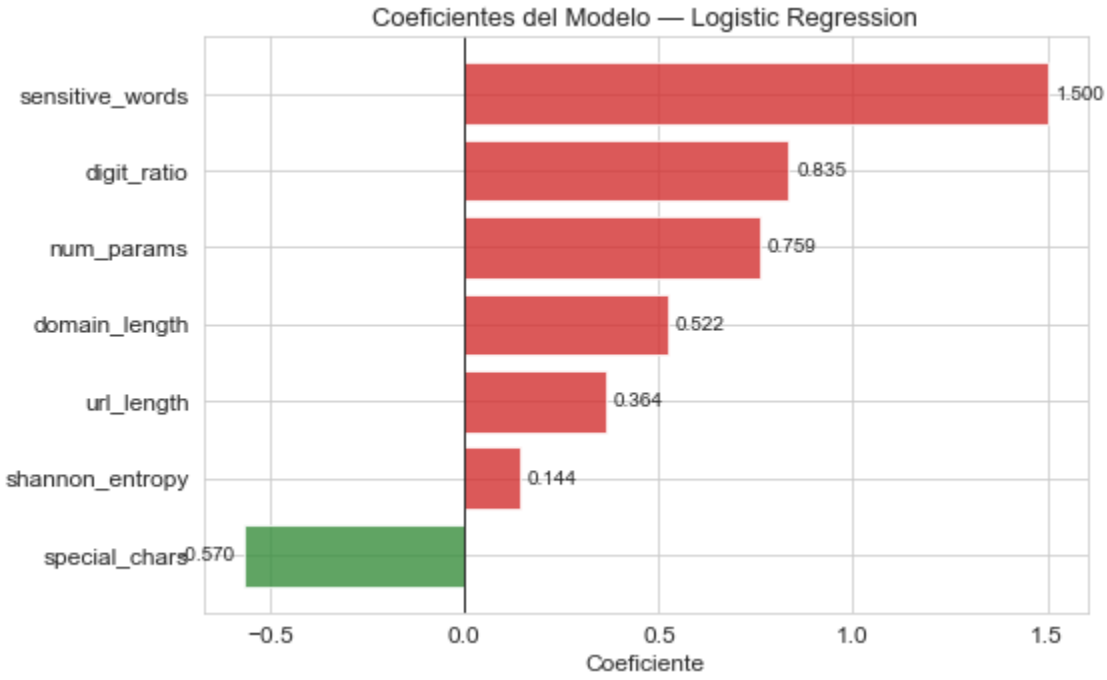
a) Alta interpretabilidad: los coeficientes indican dirección e intensidad del efecto de cada feature
b) Inferencia rápida, ideal para sistemas de detección en tiempo real
c) Probabilidades calibradas que permiten ajustar umbrales de decisión



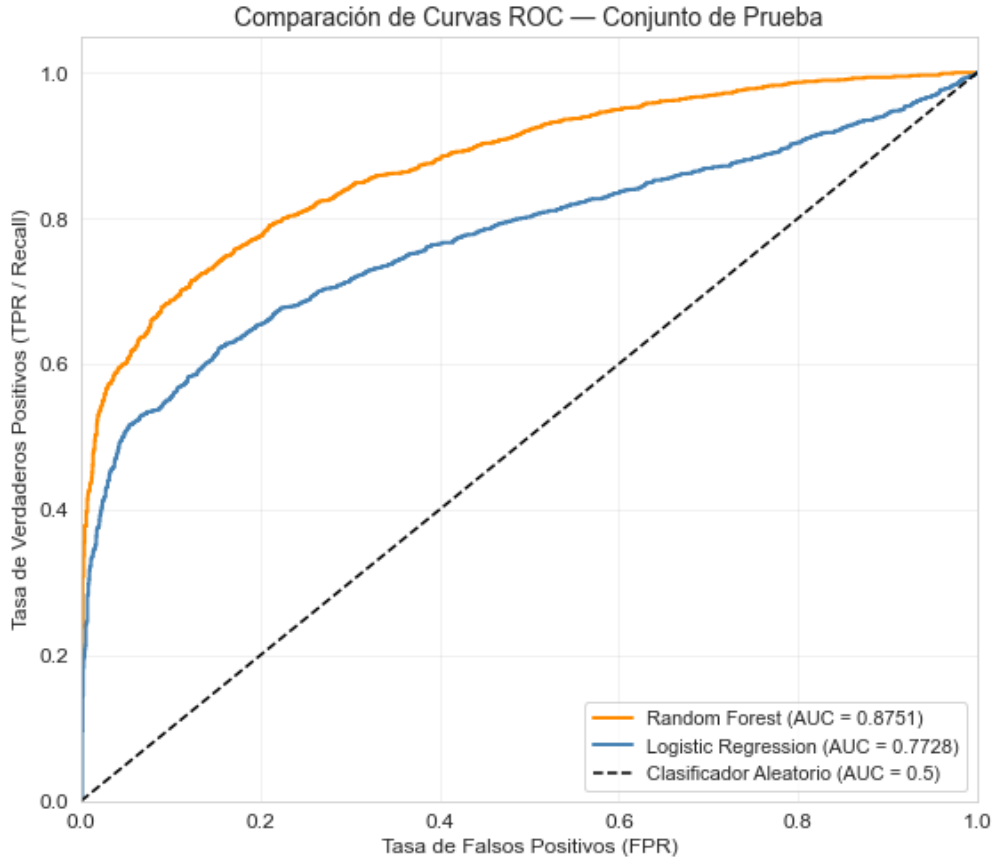


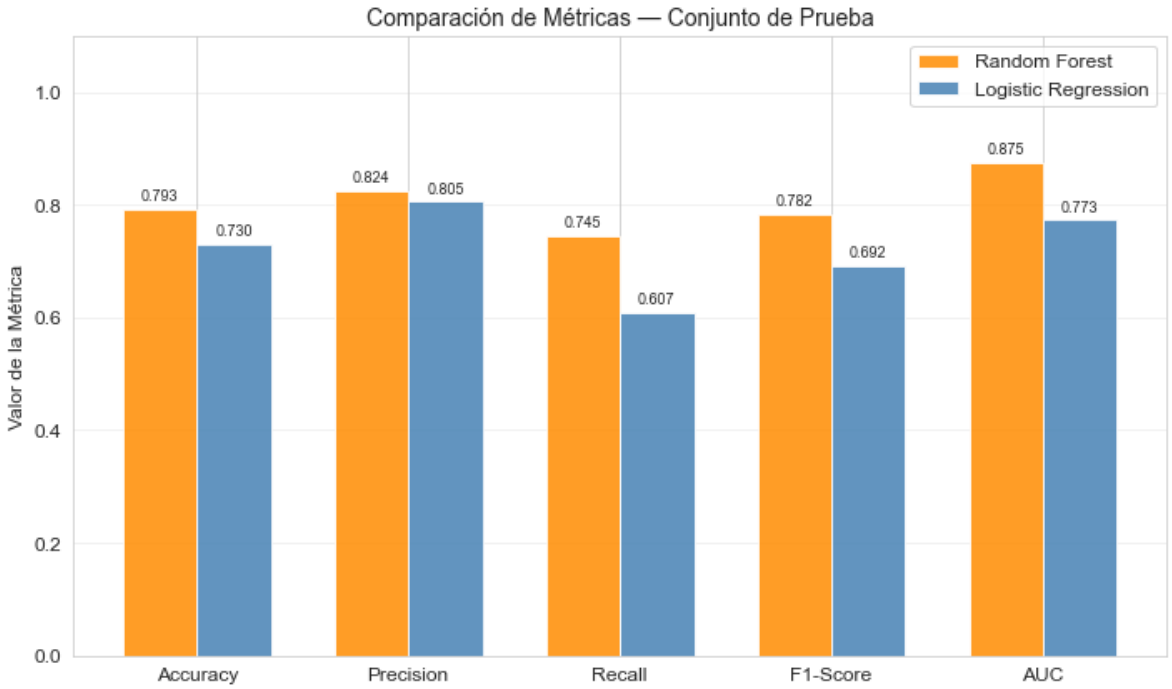






iii. Comparación entre modelos





3. Discusión

4) ¿Cuál es el impacto de clasificar un sitio legítimo como phishing?
Un Falso Positivo (FP) ocurre cuando el modelo clasifica una URL legítima como phishing. Aunque a primera vista podría parecer un error "seguro" (ya que se está siendo conservador), sus consecuencias son significativas en múltiples dimensiones.

Clasificar erróneamente un sitio legítimo como phishing puede generar una pérdida de confianza y reputación tanto para el sitio afectado como para los sistemas de detección que realizan la clasificación. Este tipo de error puede provocar que los usuarios eviten páginas legítimas, afectando la comunicación entre empresas y clientes, reduciendo el tráfico web y, en algunos casos, generando pérdidas económicas. Además, un alto número de falsos positivos puede disminuir la credibilidad del modelo predictivo, haciendo que los usuarios ignoren futuras alertas.

5) ¿Cuál es el impacto de clasificar un sitio de phishing como legítimo?
Un Falso Negativo (FN) ocurre cuando el modelo clasifica una URL de phishing como legítima, permitiendo que el ataque pase desapercibido. Este es el tipo de error más peligroso en el contexto de seguridad informática.

Clasificar un sitio de phishing como legítimo representa un riesgo mucho mayor, ya que expone directamente a los usuarios a estafas, robo de credenciales, información personal y recursos económicos. Este tipo de error puede facilitar ataques de suplantación de identidad, malware y fraude financiero. En el contexto de la seguridad informática, los falsos negativos son especialmente críticos, ya que

comprometen la protección del usuario y pueden tener consecuencias legales y económicas tanto para los usuarios como para las organizaciones involucradas.

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

Usaría Recall (como métrica principal) + F1-Score (como métrica de balance)

En seguridad informática, el principio fundamental es que es preferible tener falsas alarmas (FP) a dejar pasar un ataque real (FN). Un phishing no detectado puede tener consecuencias catastróficas e irreversibles, mientras que una falsa alarma solo genera inconveniencia temporal.

El Recall mide directamente la capacidad de detección:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Un Recall alto garantiza que la mayoría de las URLs de phishing son detectadas, minimizando la probabilidad de que un ataque exitoso pase desapercibido

El F1-Score como métrica complementaria:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Sin embargo, un Recall de 1.0 es trivialmente alcanzable si clasificamos *todo* como phishing (con Precision cercana a 0). Por ello, el **F1-Score** es útil como métrica complementaria, ya que penaliza modelos con Precision excesivamente baja. Al ser la media armónica de Precision y Recall, el F1-Score asegura que ambos estén en niveles razonables.

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

Comparación basada en resultados experimentales

Tabla comparativa — Conjunto de Prueba

Métrica	**Random Forest**	**Logistic Regression**	**Diferencia (RF − LR)**
Accuracy	0.7929	0.7302	+0.0627
Precision	0.8239	0.8051	+0.0188
Recall	0.7450	0.6074	+0.1376
F1-Score	0.7825	0.6924	+0.0901
AUC	0.8751	0.7728	+0.1023

Matriz de confusión — Conjunto de Prueba

	Random Forest	**Logistic Regression**
TP (phishing detectado)	1,277	1,041
TN (legítimo OK)	1,442	1,463
FP (falsa alarma)	273	252
FN (phishing no detectado)	437	673

Modelo seleccionado: Random Forest

Random Forest superó a Logistic Regression en todas las métricas evaluadas sobre el conjunto de prueba. A continuación se detallan las razones:

1. **Recall significativamente superior (+13.76 puntos porcentuales):** Random Forest detecta el 74.50% de las URLs de phishing frente al 60.74% de Logistic Regression. Esto significa que RF deja pasar 437 emails de phishing de cada 1,714, mientras que LR deja pasar 673. En un escenario de seguridad, esta diferencia de **236 ataques adicionales no detectados** por LR es crítica, ya que cada phishing no detectado representa un riesgo real de compromiso.

2. **F1-Score superior (+9.01 puntos porcentuales):** El F1-Score de RF (0.7825) confirma un mejor balance global entre detección y falsas alarmas. LR obtiene un F1 de 0.6924, penalizado por su bajo Recall a pesar de tener una Precision ligeramente competitiva (0.8051).

3. **AUC superior (+10.23 puntos porcentuales):** El AUC de RF (0.8751) indica una capacidad discriminativa general notablemente mayor que LR (0.7728). Esto significa que RF es mejor para distinguir entre phishing y legítimo en cualquier umbral de decisión.

4. **Precision:** Aunque LR tiene ligeramente menos FP (252 vs. 273), la diferencia es mínima (+0.0188). RF genera solo 21 falsas alarmas más, pero a cambio detecta 236 ataques de phishing adicionales — un intercambio claramente favorable.

5. **Capacidad de capturar no linealidades:** Las relaciones entre las características de URLs de phishing no son lineales. Por ejemplo, la combinación de alta shannon_entropy (importancia: 0.233) con alto digit_ratio (importancia: 0.187) puede ser más indicativa de phishing que cada variable por separado. Random Forest captura estas interacciones

de forma natural a través de sus árboles de decisión, mientras que Logistic Regression asume una relación lineal entre las features y el log-odds.

6. **Generalización consistente:** Las métricas entre validación y prueba son consistentes para RF (F1: 0.7882 → 0.7825), lo que indica que el modelo no presenta sobreajuste (*overfitting*) y generaliza adecuadamente a datos no vistos.

Conclusión: Random Forest es el modelo recomendado para la detección de phishing en este laboratorio, con ventajas claras en Recall, F1-Score y AUC.

8) 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 cuantas negativas? ¿Funciona el modelo para el BR propuesto? En caso negativo, ¿qué propone para reducir la cantidad de falsas alarmas?

Escenario

Una empresa recibe **50,000 emails**, de los cuales estima que un **15% son phishing**. Se desea aplicar el mejor modelo (Random Forest) para filtrar estos correos.

Datos del problema

Parámetro	**Valor**
Total de emails	50,000
Emails de phishing (15%)	7,500
Emails legítimos (85%)	42,500
Recall del mejor modelo (RF)	0.7450
FPR del mejor modelo (RF)	0.1592
Precision del mejor modelo (RF)	0.8239

Donde:

Recall (TPR) = TP / (TP + FN) = 1277 / (1277 + 437) = 0.7450 → Proporción de phishing correctamente detectado

FPR = FP / (FP + TN) = 273 / (273 + 1442) = 0.1592 → Proporción de legítimos incorrectamente marcados

Cálculos detallados

Paso 1: Emails de phishing detectados (Verdaderos Positivos)

$$TP = \text{Recall} \times \text{Total phishing} = 0.7450 \times 7{,}500 = 5{,}588$$

Paso 2: Emails de phishing no detectados (Falsos Negativos)

$$FN = \text{Total phishing} - TP = 7{,}500 - 5{,}588 = 1{,}912$$

Paso 3: Emails legítimos marcados incorrectamente como phishing (Falsos Positivos)

$$FP = \text{FPR} \times \text{Total legítimos} = 0.1592 \times 42{,}500 = 6{,}766$$

Paso 4: Emails legítimos correctamente clasificados (Verdaderos Negativos)

$$TN = \text{Total legítimos} - FP = 42{,}500 - 6{,}766 = 35{,}734$$

Resumen de resultados

Clasificación	**Cantidad**	**Descripción**
Verdaderos Positivos (TP)	5,588	Phishing detectado correctamente
Verdaderos Negativos (TN)	35,734	Email legítimo clasificado correctamente
Falsos Positivos (FP)	6,766	Email legítimo marcado como phishing
Falsos Negativos (FN)	1,912	Phishing que pasó como legítimo

¿Cuántas alarmas generaría el modelo?

$$\text{Total alarmas} = TP + FP = 5{,}588 + 6{,}766 = 12{,}354$$

De estas alarmas:

Alarmas positivas (correctas): 5,588 emails que realmente son phishing (45.2% de las alarmas).

Alarmas negativas (falsas alarmas): 6,766 emails legítimos bloqueados innecesariamente (54.8% de las alarmas).

¿Funciona el modelo para este caso de uso?

El modelo presenta limitaciones importantes para este escenario. Aunque logra detectar 5,588 de los 7,500 emails de phishing (74.5%), el análisis revela dos problemas significativos:

El modelo genera 6,766 falsas alarmas, lo que significa que **más de la mitad de las alertas (54.8%) son incorrectas**. Esto implica que el equipo de seguridad debería revisar manualmente casi 12,354 alertas, de las cuales la mayoría son falsas. Esta carga operativa es insostenible para la mayoría de las organizaciones.

1,912 emails de phishing (25.5%) pasan como legítimos. Cada uno de estos representa un riesgo potencial de compromiso para la empresa. En un contexto donde los empleados sufren ataques constantes, dejar pasar casi 2,000 emails maliciosos es un riesgo considerable.

Impacto en la productividad: 6,766 emails legítimos serían bloqueados, afectando la comunicación y operaciones normales de la empresa. Si se asume un tiempo promedio de 2 minutos por alerta para su revisión manual, el equipo de seguridad necesitaría aproximadamente **411 horas** solo para procesar las alertas del modelo.

Propuestas para reducir las falsas alarmas

Dado que el modelo no es suficientemente efectivo para el caso de uso propuesto, se proponen las siguientes estrategias:

En lugar de usar el umbral por defecto de 0.5, se puede aumentar el umbral de clasificación (por ejemplo, a 0.7 o 0.8). Esto significa que el modelo solo clasifica como phishing aquellas URLs con una probabilidad muy alta, reduciendo significativamente los FP a costa de un leve aumento en FN. Se puede optimizar el umbral usando la curva Precision-Recall o el punto óptimo de la curva ROC (punto de Youden).

Implementar un sistema de clasificación en cascada:

- **Nivel 1 (alta sensibilidad):** El modelo actual filtra emails con alta probabilidad de phishing, priorizando Recall.
- **Nivel 2 (alta especificidad):** Un segundo modelo o un analista humano revisa los emails marcados como phishing para confirmar o descartar la alerta, priorizando Precision.

Este enfoque reduce las falsas alarmas que llegan al usuario final sin sacrificar la tasa de detección.

IV. Referencias

Sahingoz et al. (2019): "Machine learning based phishing detection from URLs" - Expert Systems With Applications

Aung & Yamana (2019): "URL-based Phishing Detection using the Entropy of Non-Alphanumeric Characters" - iiWAS2019

Hannousse & Yahiouche (2020): "Towards Benchmark Datasets for Machine Learning Based Website Phishing Detection: An Experimental Study"

Calzarossa et al. (2023): "Explainable machine learning for phishing feature detection" - Quality and Reliability Engineering International

Karim et al. (2023): "Phishing Detection System Through Hybrid Machine Learning Based on URL" - IEEE Access