Informe - Detección Automática de Sitios de Phishing

Introducción

El presente trabajo tiene como objetivo el desarrollo, entrenamiento y evaluación de un modelo de clasificación capaz de detectar de manera automática sitios web fraudulentos de tipo phishing. Este tipo de ataques representa una de las amenazas más frecuentes en el ámbito de la ciberseguridad, ya que buscan engañar a los usuarios para obtener información confidencial, como credenciales o datos financieros. Detectar estos sitios de manera temprana y precisa resulta fundamental para minimizar riesgos y evitar posibles fraudes.

El conjunto de datos utilizado corresponde al Phishing Websites Dataset del repositorio UCI, el cual contiene un conjunto de características numéricas codificadas en {-1, 0, 1} que describen propiedades estructurales y de comportamiento de los sitios web. La variable objetivo, denominada Result, se codifica como –1 para phishing y 1 para sitios legítimos. La evaluación se centró en maximizar el rendimiento sobre la clase phishing, considerando que los falsos negativos (sitios fraudulentos no detectados) constituyen el mayor riesgo operativo. Por este motivo, se priorizó el F1-score de la clase phishing y un recall elevado, manteniendo una precisión razonable para evitar falsas alarmas excesivas.

Metodología y diseño experimental

Se trabajó con una partición estratificada de los datos en proporciones de 70 % para entrenamiento, 15 % para validación y 15 % para prueba, asegurando la preservación de la proporción original de clases. El conjunto presenta una distribución de clases balanceada, lo que permitió aplicar técnicas estándar de partición sin necesidad de correcciones adicionales por desbalance. Aun así, se priorizaron métricas sensibles a posibles desviaciones de la proporción ideal, como el recall y el F1-score por clase, para garantizar una evaluación justa del desempeño.

Para garantizar la reproducibilidad de los experimentos, se estableció una semilla aleatoria fija en todos los procesos que involucran aleatoriedad, como la partición de datos y la validación cruzada.

El desarrollo se estructuró en dos líneas experimentales complementarias. El Script A (notebookSinSeleccionColumnas.ipynb) implementó un modelo Gaussian Naive Bayes sin selección de características, utilizado como modelo base. Sobre esta configuración se aplicó una búsqueda de hiperparámetros mediante GridSearchCV, optimizando el parámetro var_smoothing con el F1-score de la clase phishing como métrica principal. Además, se analizó el impacto del ajuste del umbral de decisión, explorando valores alternativos al estándar (0.5) para maximizar el F1 en el conjunto de validación.

Por su parte, el Script B (notebook.ipynb) incorporó una etapa de selección de características dentro de un pipeline que combinó SelectKBest(mutual_info_classif) con GaussianNB. Este enfoque permitió reducir la redundancia y eliminar atributos irrelevantes sin riesgo de fuga de información, ya que la selección se ejecutó dentro del proceso de validación cruzada. En este caso, se optimizaron simultáneamente los hiperparámetros selector__k (número de características seleccionadas) y clf__var_smoothing, también priorizando el F1-score como criterio de evaluación. Finalmente, se ajustó

el umbral de clasificación mediante el análisis de la curva Precisión-Recall para identificar el punto que maximizaba el equilibrio entre ambas métricas.

Dado que el dataset es de tamaño moderado, la validación cruzada de 5 particiones permitió estimar de forma más estable la capacidad de generalización sin aumentar el costo computacional.

Resultados y análisis

Los resultados obtenidos se resumen en la siguiente tabla, donde se reportan las métricas sobre el conjunto de prueba:

Experimento	Precision	Recall	F1
Script A — Base (0.5)	0.5288	0.9986	0.6915
Script A — Optimizado HP	0.8979	0.8857	0.8918
Script A — Umbral 0.95	0.5930	0.9973	0.7438
Script B — Pipeline (0.5)	0.8245	0.9333	0.8756
Script B — Umbral 0.9	0.9086	0.8789	0.8935

En el modelo base de Script A, el recall extremadamente alto (≈ 1.0) indica que casi todos los sitios de phishing fueron detectados, aunque a costa de una baja precisión debido a la gran cantidad de falsos positivos. Tras la optimización de hiperparámetros, el modelo alcanzó un equilibrio más adecuado (F1 ≈ 0.89), reduciendo falsos positivos sin sacrificar excesivamente la sensibilidad. Sin embargo, al incrementar el umbral a 0.95, si bien se mantuvo un recall muy alto, el F1-score cayó a 0.74, reflejando una pérdida de equilibrio en el compromiso precisión–recall.

El modelo del Script B mostró un rendimiento más consistente y estable tanto en validación como en prueba. Con el umbral estándar (0.5) alcanzó un F1 de 0.876, mientras que el ajuste a 0.9 permitió mejorar la precisión hasta 0.91 y mantener un F1 global de 0.894, ligeramente superior al del Script A optimizado. La coherencia entre los resultados de validación y prueba sugiere una buena capacidad de generalización y menor riesgo de sobreajuste.

Discusión

El análisis comparativo entre ambos experimentos permite entender las causas del mejor desempeño del Pipeline del Script B. La incorporación de una etapa de selección de características mediante Información Mutua (MI) concentró el aprendizaje en las variables más relevantes, reduciendo redundancia y ruido. Esto mejoró la estabilidad del clasificador y disminuyó su varianza sin aumentar el sesgo, favoreciendo un comportamiento más consistente entre validación y prueba.

En cuanto al modelo, Gaussian Naive Bayes logró adaptarse adecuadamente a las variables discretas del dataset, aunque su rendimiento resultó sensible al parámetro var_smoothing, que estabiliza las estimaciones de varianza. La optimización de este hiperparámetro mediante GridSearchCV fue clave para reducir falsos positivos y mejorar la precisión sin comprometer la detección de phishing.

El ajuste del umbral de decisión mostró cómo varía el equilibrio entre precisión y recall: valores bajos (0.5) maximizan la detección, mientras que umbrales más altos (0.9) reducen falsas alarmas

manteniendo un recall elevado. Esta capacidad de ajuste confirma la flexibilidad operativa del modelo según el nivel de riesgo aceptable.

Finalmente, la consistencia del Script B frente a la mayor variabilidad observada en el Script A refleja un mejor equilibrio sesgo–varianza y una metodología más sólida. En conjunto, los resultados demuestran que la mejora de rendimiento no se debe solo a ajustes numéricos, sino a un diseño experimental riguroso que integra selección de atributos, control de aleatoriedad y validación reproducible. Esta robustez metodológica constituye el principal aporte del trabajo y sienta las bases para futuros desarrollos más complejos.

Conclusiones

El trabajo logró implementar un sistema de clasificación eficaz para la detección automática de sitios de phishing, cumpliendo con los objetivos planteados en la consigna. El modelo final, correspondiente al Script B, alcanzó un F1-score cercano a 0.89 y un recall superior al 0.87, lo que demuestra una alta capacidad para identificar sitios fraudulentos con un bajo margen de error. En términos prácticos, esto implica que el modelo dejaría escapar menos del 13 % de los casos de phishing, un resultado especialmente favorable en contextos donde la seguridad es prioritaria.

Al ajustar el umbral a 0.9, la precisión se elevó al 0.91, indicando que la mayoría de las alertas emitidas serían efectivamente correctas. Este equilibrio entre protección efectiva (alto recall) y usabilidad (alta precisión) confirma que el modelo logra un compromiso adecuado entre ambos extremos del problema, manteniendo alta sensibilidad sin generar un exceso de falsas alarmas.

Comparado con el modelo base del Script A, el enfoque del Script B mostró mayor estabilidad y generalización, gracias a la selección de características mediante Información Mutua y a la integración de todas las etapas dentro de un pipeline reproducible. Estas decisiones metodológicas evitaron fuga de información y redujeron la varianza, lo que explica su rendimiento más consistente en validación y prueba.

El modelo resultante presenta además una flexibilidad operativa valiosa, ya que el umbral de decisión puede adaptarse según el nivel de riesgo tolerado: mayor sensibilidad para entornos críticos como sistemas bancarios, o mayor precisión para aplicaciones orientadas al usuario final. En conjunto, los resultados demuestran que incluso un modelo estadísticamente simple como Gaussian Naive Bayes puede alcanzar un desempeño competitivo cuando se aplican buenas prácticas de validación, selección de variables y calibración de umbral. Esto confirma que la rigurosidad metodológica y la comprensión del problema son tan determinantes como la complejidad del algoritmo en sí.