PROYECTO DE CLASIFICACIÓN

Tema: Implementación y evaluación de clasificación de sitios que realizan Phishing

Conjunto de datos: https://archive.ics.uci.edu/dataset/327/phishing+websites

Fecha de entrega: 10 de noviembre

Entregables

1. Parte 1 — Desarrollo del Script
2. Parte 2 — Informe Breve (máx. 3 hojas)Desarrollo del Script

Desarrollo del Script

Para comenzar, se deberá cargar el dataset localmente, en formato `.arff` o `.csv`, e identificar claramente la variable objetivo, denominada `Result`. Se espera que el estudiante explore los datos iniciales mostrando la forma del dataset y los tipos de cada columna, con el fin de comprender la estructura y las características disponibles.

Luego, se realizará un preprocesamiento adecuado de los datos. En esta etapa, se confirmará que los valores de la variable `Result` tengan una codificación binaria coherente con el problema (por ejemplo, `1` para phishing y `-1` para sitios legítimos), se analizará la distribución de clases mediante una gráfica y se verificará si existen valores faltantes que requieran tratamiento.

El trabajo deberá garantizar la reproducibilidad de los resultados estableciendo una semilla de aleatoriedad (`random_state`) en todas las funciones que involucren procesos estocásticos, como la partición de datos o la validación cruzada.

A continuación, los datos se dividirán en conjuntos de entrenamiento y prueba, utilizando una partición estratificada para mantener la proporción original de clases. El estudiante deberá seleccionar y justificar la estrategia de validación empleada: una validación simple o una validación cruzada (por ejemplo, mediante `Stratified K-Fold`).

Con la división de datos definida, se entrenará un modelo base utilizando `GaussianNB()` sin ajustes iniciales. El desempeño de este modelo se evaluará con métricas adecuadas para

clasificación binaria, especialmente considerando los riesgos asociados a los falsos negativos en phishing. En el contexto de phishing:

- FN (phishing no detectado) → mayor riesgo de fraude
- FP (sitio legítimo marcado como phishing) → molestia para el usuario

Precision indica qué proporción de las predicciones positivas realizadas por el modelo son correctas.

$$Precision = \frac{TP}{TP + FP}$$

Donde:

- **TP**: Verdaderos Positivos
- **FP**: Falsos Positivos

Alta precisión implica pocos falsos positivos.

Recall mide cuántos de los positivos reales fueron correctamente identificados:

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

Donde:

- **FN**: Falsos Negativos

Un alto recall implica que el modelo deja pasar pocos positivos sin detectar.

Por eso, es importante equilibrar ambas métricas, lo que justifica el uso del F1-score como métrica de optimización.

Posteriormente, se avanzará con la optimización de hiperparámetros. Para ello, será necesario seleccionar al menos un hiperparámetro que pueda influir significativamente en los resultados —como `var_smoothing`— y definir un espacio de búsqueda razonable. Se recomienda utilizar técnicas como `GridSearchCV` o `RandomizedSearchCV`, priorizando el F1-score como criterio de evaluación debido al contexto de fraude, donde el equilibrio entre precisión y recall es relevante.

Finalizado el proceso de optimización, se realizará una comparación entre el modelo base y el modelo ajustado utilizando métricas tales como Accuracy, Precision, Recall, F1-score y matriz de confusión. De manera opcional, se puede incluir el análisis mediante curva ROC y el cálculo de AUC.

Finalmente, se deberá incluir al menos una visualización relevante de los resultados del modelo, como la matriz de confusión o la curva ROC, que permita interpretar de forma clara el desempeño alcanzado.

Informe Breve (máx. 3 hojas)

El informe debe incluir:

1. Descripción del problema
    a. ¿Qué es el phishing?
    b. ¿Por qué es importante detectarlo automáticamente?
2. Análisis de la distribución de clases
    a. Mostrar la proporción de clases 1 y -1
    b. Comentar si existe desbalance y cómo podría influir en:
        i. Precisión (accuracy)
        ii. Recall de clase positiva (detección de phishing)
        iii. Potencial riesgo en un sistema real
3. Reproducibilidad en los experimentos. Responder brevemente:
    a. ¿Qué significa usar una semilla de aleatoriedad (random_state)?
    b. ¿Por qué es importante que los resultados puedan repetirse exactamente?
    c. ¿Qué operaciones del script requieren control de aleatoriedad? Ejemplos: división train/test, validación cruzada.
4. Elección del método de validación. Explicar y justificar:
    a. ¿Por qué se usa validación estratificada?
    b. ¿Cuándo sería más conveniente usar validación cruzada?
    c. ¿Cómo influye el tamaño del dataset en esa decisión?
5. Selección y optimización de hiperparámetros. El informe debe incluir:
    a. ¿Qué hiperparámetros posee el modelo elegido?
    b. ¿Cuáles pueden influir más en la calidad de predicción? ¿Por qué?
    c. ¿Qué espacio de búsqueda se definió?
    d. ¿Qué criterio se usó para la optimización?
    e. (Ej: F1-score para priorizar detección de phishing)
    f. Tabla con resultados del modelo base vs. modelo optimizado
6. Resultados y conclusiones
    a. Incluir la tabla exportada con métricas
    b. Comparar modelo base vs modelo optimizado
    c. Interpretar al menos 2 métricas relevantes
    d. Referirse a la matriz de confusión:
        i. ¿Qué tipo de error fue más frecuente?
        ii. ¿Por qué es delicado fallar en phishing?
    e. ¿El modelo alcanzó desempeño suficiente? ¿Por qué?