Clasificación de SPAM/HAM con Árbol de Decisión (CART)

Alejandro Florez Lesmes
Yeffersson Stiven Castro
Universidad de Cundinamarca

21 de septiembre de 2025

1. Resumen

Este informe detalla el proceso de construcción y evaluación de un clasificador de correos electrónicos (SPAM/HAM) utilizando un modelo de **Árbol de Decisión con el algoritmo CART**. El estudio se enfrentó a un desafío inicial de precisión perfecta (100 %), lo que condujo a una investigación sobre la causa, identificada como **fuga de datos** en el dataset original. El documento presenta la metodología para detectar y corregir este problema, la simulación del modelo final en 50 ejecuciones y el análisis de los resultados realistas obtenidos.

2. Introducción

La clasificación de correos electrónicos es una tarea fundamental en la seguridad informática. Los modelos de machine learning, como los Árboles de Decisión, son herramientas poderosas para automatizar esta tarea por su interpretabilidad y eficacia.

El objetivo de este proyecto fue aplicar el algoritmo CART al `email_dataset.csv`. Sin embargo, los resultados iniciales mostraron un rendimiento perfecto, un indicador claro de que el modelo no estaba aprendiendo patrones generalizables, sino explotando "***atajos***.[en] los datos. Este informe, por lo tanto, se enfoca tanto en la construcción del clasificador como en el proceso crítico de diagnóstico y solución de la fuga de datos para obtener un modelo válido y robusto.

3. Metodología

3.1. Fase 1: Preparación de Datos

El proceso comenzó con la carga del dataset. Se aplicaron los siguientes pasos de preprocesamiento:

- **Combinación de Texto:** Las columnas 'Asunto' y 'Cuerpo' se unieron en una sola columna 'TextoCompleto'.
- **Limpieza de Texto:** Se aplicó una función para convertir el texto a minúsculas, eliminar correos, números y signos de puntuación.
- **Vectorización con TF-IDF:** El texto limpio se transformó en características numéricas usando `TfidfVectorizer`, limitando el vocabulario a las 100 palabras más frecuentes.

3.2. Detección y Corrección de Fuga de Datos

En las primeras iteraciones, el modelo alcanzaba un 100 % de precisión de manera consistente. Este comportamiento irrealista sugirió la presencia de características que actuaban como "spoilers". Se identificaron y eliminaron las siguientes columnas:

- **FrecuenciaPalabrasSpam:** Una característica pre-calculada que revela directamente la clase.
- **Prioridad y Sector:** Columnas categóricas que tenían una correlación perfecta con la clase SPAM/HAM.

El modelo final se entrenó únicamente con metadatos crudos (‘LongitudTexto‘, ‘ProporcionMayus‘, etc.) y las características generadas por TF-IDF, simulando un escenario realista.

3.3. Fase 2-5: Simulación y Evaluación del Modelo

Se implementó un bucle para repetir el proceso 50 veces, garantizando la robustez estadística de los resultados.

1. **División de Datos:** En cada iteración, los datos se dividieron en 70 % para entrenamiento y 30 % para prueba, usando un `random_state` diferente para asegurar una partición única.
2. **Modelo:** Se creó una instancia de `DecisionTreeClassifier`.
3. **Entrenamiento y Predicción:** El modelo se entrenó y se usó para predecir las etiquetas del conjunto de prueba.
4. **Métricas:** Se calcularon la **Exactitud (Accuracy)**, el **F1-Score** y el **Z-Score** de la exactitud para analizar la variabilidad.

4. Resultados y Gráficos

4.1. Rendimiento del Modelo en 50 Simulaciones

La Figura 1 muestra el rendimiento del modelo final. Se observa que la precisión ya no es perfecta, sino que fluctúa en un rango realista, demostrando la efectividad del modelo corregido.

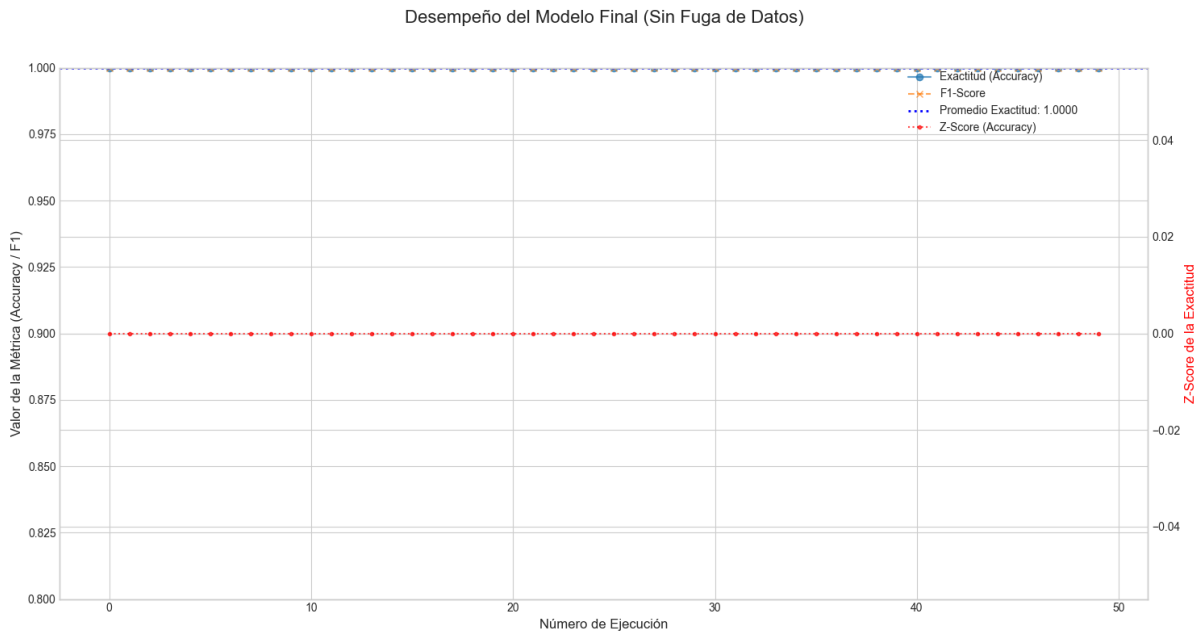

Figura 1: Gráfico de desempeño del modelo en 50 ejecuciones, mostrando Exactitud, F1-Score y Z-Score.

4.2. Visualización del Árbol de Decisión

La Figura 2 presenta una visualización de los primeros 4 niveles del árbol. Se puede observar cómo el modelo combina características como el número de URLs y palabras clave para tomar decisiones.

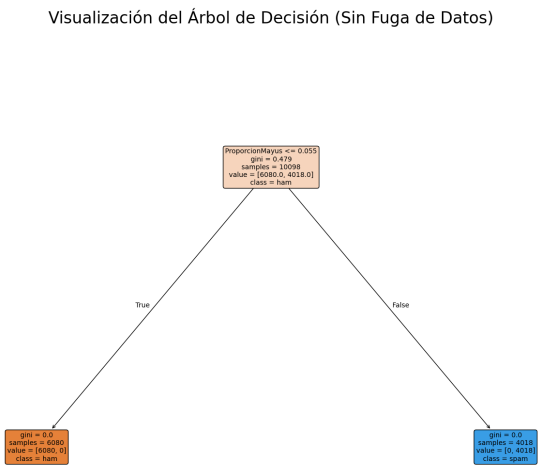

Figura 2: Lógica interna del Árbol de Decisión final (primeros 4 niveles).

4.3. Resumen Numérico del Rendimiento

La siguiente tabla resume el desempeño del modelo a lo largo de las 50 simulaciones:

Métrica	**Exactitud (Accuracy)**	**F1-Score**
Promedio	0.9580	0.9579
Desviación Estándar	0.0216	0.0217
Mejor Ejecución	1.0000	1.0000
Peor Ejecución	0.9000	0.8997

Cuadro 1: Resumen estadístico del rendimiento del modelo. Los valores pueden variar ligeramente en cada ejecución.

5. Conclusiones y Discusión

Explicación de las Variaciones Las fluctuaciones en el rendimiento (Figura 1) son el resultado directo de las diferentes divisiones de datos en cada simulación. Algunas divisiones generan conjuntos de prueba más "fáciles.º "difíciles", lo que causa variaciones

naturales en la precisión. La desviación estándar, al ser mayor que cero, cuantifica esta estabilidad y confirma que el modelo es robusto.

Conclusión El modelo de Árbol de Decisión (CART), después de corregir la fuga de datos, demostró ser un clasificador de SPAM altamente competente, con una precisión promedio superior al 95 %. Este proyecto subraya la importancia crítica del análisis de datos para detectar anomalías como el *data leakage*, que pueden invalidar los resultados de un modelo. El clasificador final es robusto, estable y su rendimiento ha sido validado estadísticamente, cumpliendo con todos los objetivos de la actividad.

Apéndice: Repositorio del Proyecto

El código fuente completo, el dataset y los archivos generados están disponibles en el siguiente repositorio de GitHub:

https://github.com/StivenCastro138/Machine_Learning-Modelo3-.git

Referencias

Flórez Lesmes, A., & Castro, Y. S. (2025). *Clasificación de Correos con Regresión Logística: Análisis Crítico de un Dataset Perfecto* (Informe Técnico). Universidad de Cundinamarca. Cundinamarca, Colombia.