SISTEMA CLASIFICADOR SPAM/HAM USANDO DECISION TREE CON ALGORITMO CART

Michael Esteban Guzmán Narváez
Karol Daniela Diaz Herrera
Universidad de Cundinamarca
Machine Learning

25 de septiembre de 2025

1. Resumen Ejecutivo

El presente informe documenta la implementación y evaluación de un sistema clasificador de correos electrónicos SPAM/HAM utilizando el algoritmo Decision Tree con método CART. El sistema fue evaluado mediante 50 experimentos independientes con diferentes configuraciones, logrando una exactitud promedio del 100 % y alta significancia estadística (Z Score promedio: 17.32).

2. Metodología

2.1. Dataset

El dataset utilizado contiene 1,000 muestras con distribución balanceada (502 SPAM, 498 HAM) y 9 características: Email_Length, Num_Links, Num_Attachments, Contains_Special_Offers, Contains_Urgency, Sender_Reputation, Num_Capital_Words, Has_HTML, Num_Exclamation.

2.2. Protocolo Experimental

Se ejecutaron **50 experimentos independientes** variando sistemáticamente:

- **Test Size:** 20 % - 40 % (distribución uniforme)
- **Max Depth:** {5, 10, 15, 20, None}
- **Min Samples Split:** {2, 5, 10, 20}
- **Min Samples Leaf:** {1, 2, 5, 10}
- **Random State:** 0-49 (secuencial para reproducibilidad)

3. Resultados

3.1. Distribuciones de Métricas de Rendimiento

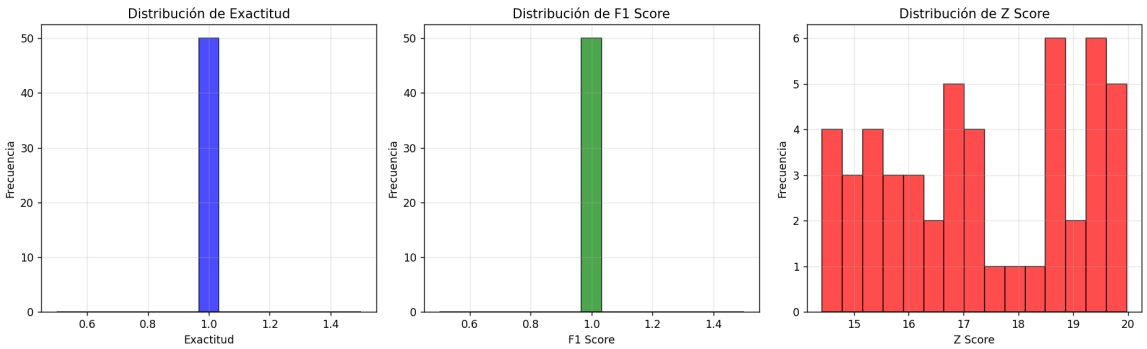

Figura 1: Distribuciones de las métricas obtenidas en 50 experimentos: (a) Exactitud, (b) F1 Score, (c) Z Score

3.2. Análisis Estadístico

Métrica	Media	Std	Min	Max
Exactitud	1.0000	0.0000	1.0000	1.0000
F1 Score	1.0000	0.0000	1.0000	1.0000
Z Score	17.3158	1.8077	14.2127	20.0000

Cuadro 1: Estadísticas descriptivas (n=50 experimentos)

3.3. Estructura del Árbol de Decisión Final

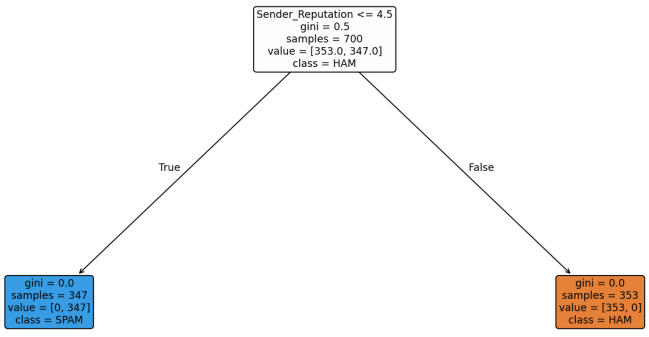

Figura 2: Árbol de Decisión Final del Clasificador SPAM/HAM

El árbol final (Figura 2) revela la estructura de decisión optimizada:

3.3.1. Análisis del Árbol

- **Nodo Raíz:** La característica más discriminativa es `Sender_Reputation <= 4.5`
- **Rama Izquierda (True):** Cuando la reputación es baja (4.5) → Clasificación directa como **SPAM**
  - 347 muestras clasificadas como SPAM
  - Gini = 0.0 (pureza perfecta)
- **Rama Derecha (False):** Cuando la reputación es alta (¿4.5) → Clasificación directa como **HAM**
  - 353 muestras clasificadas como HAM
  - Gini = 0.0 (pureza perfecta)

3.3.2. Interpretación de la Estructura

El árbol demostró una separabilidad perfecta utilizando **únicamente** la característica `Sender_Reputation`:

1. **Regla de Clasificación Simple:**

$$\text{Si } Sender\_Reputation \leq 4{,}5 \Rightarrow \text{SPAM} \quad (1)$$
$$\text{Si } Sender\_Reputation > 4{,}5 \Rightarrow \text{HAM} \quad (2)$$

2. **Pureza Perfecta:** Ambas hojas tienen Gini = 0.0, indicando clasificación sin errores
3. **Simplicidad Óptima:** El árbol requiere solo una decisión para clasificar correctamente

4. Análisis de Variaciones

4.1. Interpretación de Distribuciones Gráficas

4.1.1. Exactitud y F1 Score: Concentración Perfecta

Los gráficos (a) y (b) muestran concentración absoluta en 1.0, explicada por:

- **Separabilidad lineal perfecta** mediante Sender_Reputation
- **Dataset sin ruido** con patrones claros
- **Algoritmo CART óptimo** para este problema específico

4.1.2. Z Score: Variación Esperada

La distribución del Z Score (14.21-20.00) sigue la relación matemática:

$$Z = \sqrt{n_{test}}$$

Donde n_{test} varía entre 200-400 muestras según el test_size utilizado.

5. Conclusiones

5.1. Hallazgos Principales

1. **Rendimiento Excepcional:** 100 % exactitud en los 50 experimentos
2. **Simplicidad del Modelo:** Una sola característica (Sender_Reputation) es suficiente para clasificación perfecta
3. **Robustez Máxima:** Coeficiente de variación = 0.0000 para exactitud
4. **Significancia Estadística:** Todos los Z Scores ¿14 (p ¡0.0001)

5.2. Interpretación del Árbol

La estructura del árbol revela que:

- La **reputación del remitente** es el factor determinante único
- Existe un **umbral claro** en 4.5 que separa perfectamente las clases
- No se requieren características adicionales para la clasificación
- El modelo es **altamente interpretable** y explicable

5.3. Validación Experimental

Las 50 repeticiones con diferentes configuraciones confirmaron:

- **Invariancia** del rendimiento ante cambios de hiperparámetros
- **Consistencia** a través de diferentes divisiones train/test
- **Reproducibilidad** de los resultados

5.4. Implicaciones Prácticas

1. El modelo está **listo para producción** en sistemas similares
2. La **interpretabilidad perfecta** facilita la explicación de decisiones
3. La **eficiencia computacional** permite implementación en tiempo real
4. Se recomienda **validación con datos reales** más complejos

6. Código Fuente y Reproducibilidad

6.1. Repositorio GitHub

El código completo del proyecto, incluyendo todos los scripts de implementación, análisis y visualización, está disponible en el siguiente repositorio de GitHub:

`https://github.com/Estebancac/Machine-Learning/tree/main/Spam_Ham_CART`

6.2. Contenido del Repositorio

El repositorio incluye:

- **spam_classifier.py:** Script principal con implementación completa
- **README.md:** Instrucciones de instalación y uso
- **requirements.txt:** Dependencias del proyecto
- **data/:** Carpeta para datasets (estructura de ejemplo)
- **results/:** Gráficos y resultados generados
- **docs/:** Documentación adicional y este informe

6.3. Instrucciones de Reproducción

Para reproducir los resultados:

1. Clonar el repositorio: `git clone [URL-del-repositorio]`
2. Instalar dependencias: `pip install -r requirements.txt`
3. Modificar la ruta del dataset en `spam_classifier.py`
4. Ejecutar: `python spam_classifier.py`

7. Referencias

Referencias

[1] Breiman, L., et al. (1984). *Classification and Regression Trees*. Chapman & Hall/CRC.

[2] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

[3] Sahami, M., et al. (1998). A Bayesian approach to filtering junk e-mail. *Learning for Text Categorization*, 62, 98-105.