Markdown para regresión logística

Juan Camilo Calderon Rivera

Resumen / propósito

El presente documento explica paso a paso la implementación de un modelo de **regresión logística** utilizando Python y librerías de *Machine Learning*. El objetivo es **clasificar correos electrónicos** en dos categorías principales:

- **HAM (0):** correos normales, no considerados spam.
- **SPAM (1):** correos sospechosos o no deseados.

El flujo de trabajo abarca desde la **carga y preparación del dataset**, pasando por el **entrenamiento del modelo**, hasta la **evaluación de resultados con métricas y gráficas**. Finalmente, se analizan los **coeficientes del modelo** para interpretar la importancia de cada variable.

1. Cargar dataset

Se inicia importando librerías esenciales:

- **pandas**: manejo de datos en DataFrame.
- **matplotlib y seaborn**: visualización gráfica.
- **scikit-learn**: para preprocesamiento, entrenamiento y métricas.

Archivo utilizado: correos_features.csv

Variables:

- **Variable objetivo (target):**
    - es_largo → indica si el correo es largo (1) o no (0).
- **Variables eliminadas:**
    - remitente, asunto, es_largo (ya que no aportan directamente al modelo).
- **Variables predictoras (features):**
    - Todas las demás columnas del dataset.

Además, se aplica **StandardScaler** para normalizar los datos, garantizando que todas las variables tengan la misma escala y evitando que aquellas con valores más grandes dominen el modelo.

2. Entrenamiento del modelo

El dataset se divide en dos subconjuntos:

- **80% Entrenamiento** → usado para ajustar el modelo.
- **20% Prueba** → usado para evaluar el desempeño en datos no vistos.

Se utiliza el modelo:

- **Regresión Logística (LogisticRegression)**
    - max_iter = 1000 → asegura que el modelo converge incluso en datasets grandes.
    - random_state = 42 → garantiza reproducibilidad.

El modelo ajusta una función lineal que estima la probabilidad de que un correo pertenezca a la clase **SPAM (1)**.

3. Evaluación de resultados

Una vez entrenado el modelo, se calculan distintas métricas para valorar su desempeño:

1. **F1-Score**
    - Métrica balanceada entre *precisión* y *recall*.
    - Útil en problemas de clasificación binaria con clases desbalanceadas.
2. **Reporte de clasificación**
    - Proporciona valores de precisión, recall y f1-score por clase:
        - **HAM (0)**
        - **SPAM (1)**
3. **Matriz de confusión**
    - Representa los valores predichos frente a los reales:
        - Verdaderos Positivos (VP)
        - Falsos Positivos (FP)
        - Verdaderos Negativos (VN)
        - Falsos Negativos (FN)

4. **Curva ROC y AUC**
    - ROC → gráfico que mide la sensibilidad frente a la especificidad.
    - AUC (Área bajo la curva) → mide la calidad global del modelo.

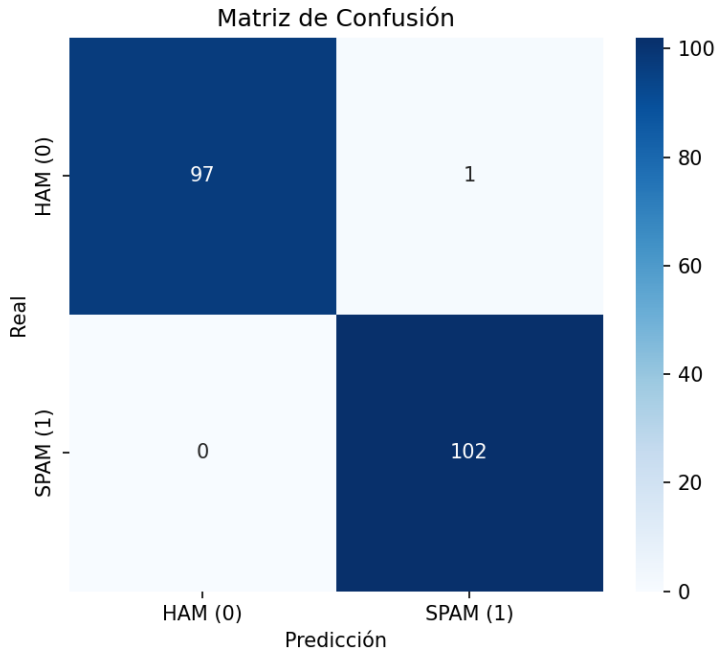
Espacio reservado para gráficas:

- [Inserte aquí matriz de confusión]
- [Inserte aquí curva ROC con valor AUC]

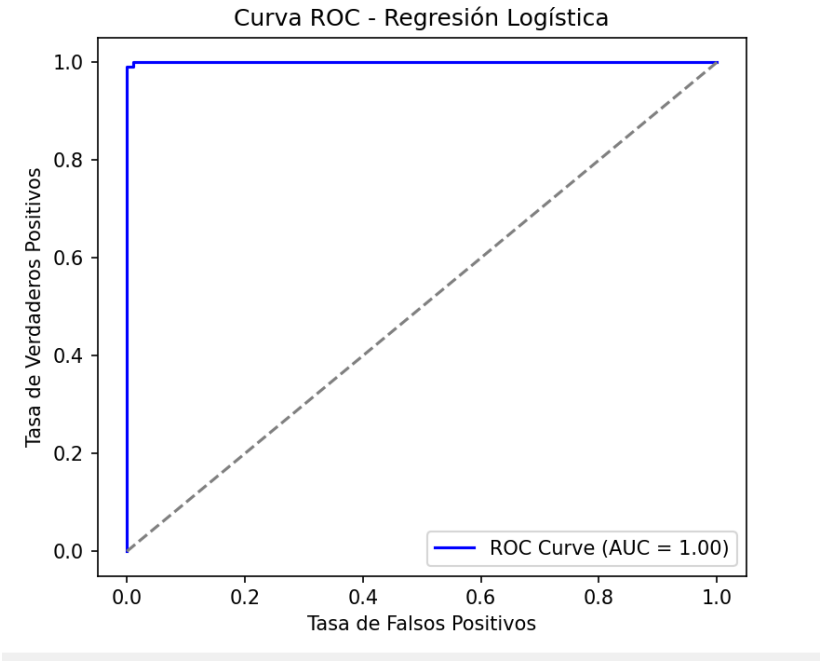
4. Visualización gráfica

El código genera automáticamente dos visualizaciones principales:

- **Matriz de confusión:** heatmap con anotaciones de los valores.



- **Curva ROC:** línea azul que representa el rendimiento, junto con el AUC.



Ambas gráficas permiten comprender mejor el desempeño del modelo más allá de las métricas numéricas.

5. Coeficientes del modelo

La regresión logística permite analizar la importancia de cada variable predictora.

Interpretación de coeficientes:

- **Coeficientes positivos (🔺):** incrementan la probabilidad de que el correo sea **SPAM (1)**.
- **Coeficientes negativos (🔻):** disminuyen la probabilidad de que el correo sea **SPAM (1)**.

El código genera una tabla con las siguientes columnas:

- Feature → nombre de la variable.
- Coeficiente → valor calculado por el modelo.
- Importancia → valor absoluto del coeficiente.
- Signo → dirección del impacto (positivo o negativo).

Espacio reservado para tabla de coeficientes:

```
============================================================
INFLUENCIA DE CADA FEATURE EN EL MODELO
============================================================
           Feature  Coeficiente  Importancia Signo
     num_palabras       7.3073       7.3073     ▲
     tiene_premio      -0.1616       0.1616     ▼
     tiene_saludo       0.1529       0.1529     ▲
     tiene_dinero       0.0971       0.0971     ▲
       tiene_link      -0.0888       0.0888     ▼
 mayusculas_ratio       0.0810       0.0810     ▲
         longitud      -0.0680       0.0680     ▼
signos_exclamacion       0.0625       0.0625     ▲
     num_adjuntos       0.0281       0.0281     ▲

Intercepto (bias): 0.5016
```

6. Ecuación del modelo

La regresión logística se representa con la ecuación:

logit(p)=(coef1×feature1)+(coef2×feature2)+…+(bias)logit(p) = (coef_1 × feature_1) + (coef_2 × feature_2) + … + (bias)logit(p)=(coef1×feature1)+(coef2×feature2)+…+(bias)

Donde:

- **logit(p):** combinación lineal de las variables.
- **p = 1 / (1 + e^(-logit(p)))** representa la probabilidad de que un correo sea **SPAM (1)**.

Esto permite no solo predecir la clase, sino también la **probabilidad estimada** de que un correo pertenezca a SPAM.

---

7. Gráfico de importancia de features

Se genera un gráfico de barras horizontales:

- Barras hacia la derecha (**positivas**) → aumentan la probabilidad de SPAM.
- Barras hacia la izquierda (**negativas**) → reducen la probabilidad de SPAM.

Espacio reservado para gráfico:

Conclusiones

- La regresión logística es un modelo interpretable y adecuado para la clasificación de correos.
- El análisis de coeficientes permite comprender qué características tienen mayor influencia en la clasificación.
- Métricas como F1 y AUC complementan la evaluación más allá de la simple precisión.
- La visualización gráfica refuerza la interpretación y facilita la comunicación de resultados.