Proyecto #3

ALEXIS MESIAS FLORES

CC3085 Inteligencia Artificial

De manera individual, realice un proyecto donde implemente los conceptos vistos de clase de probabilidad y redes bayesianas para construir un clasificador de SPAM/HAM.

Entrega

- Trabajo escrito: **14 de mayo, 23;59**
- Formato: PDF
- Presentación: **14 de mayo, presencial, periodo de clase**

Trabajo escrito

Elabore un reporte detallado sobre los pasos necesarios para la correcta construcción de un filtro SPAM/HAM usando bayes.

GITHUB: https://github.com/Alexistodj124/Proyecto3IA

La estructura mínima de su reporte debe comprender:

a. *Análisis de datos exploratorio (EDA).*
   - El dataset utilizado contiene mensajes etiquetados como SPAM o HAM. Una de las características más relevantes observadas es que los mensajes SPAM tienden a ser más largos, con un vocabulario más variado y con palabras como “free”, “win”, “urgent” o números telefónicos. En contraste, los mensajes HAM suelen ser más personales, con palabras comunes como “ok”, “see”, “home”, “love”, entre otras.
   - La proporción de mensajes HAM es mayor que la de SPAM, lo cual debe tenerse en cuenta para evitar sesgos en el modelo.
   - Se observaron diferencias también en la longitud promedio de los mensajes y en la frecuencia de ciertas palabras clave.

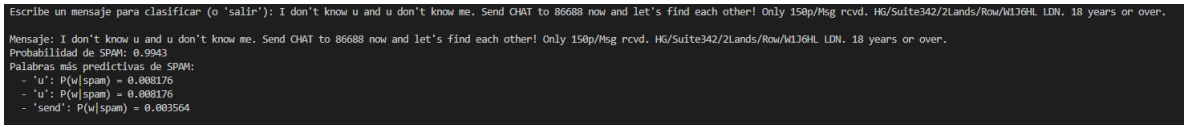
b. *Limpieza de datos*
   - El preprocesamiento consistió en varios pasos importantes para normalizar el texto:
     - Conversión a minúsculas: para uniformizar todas las palabras y evitar distinción entre “Free” y “free”.

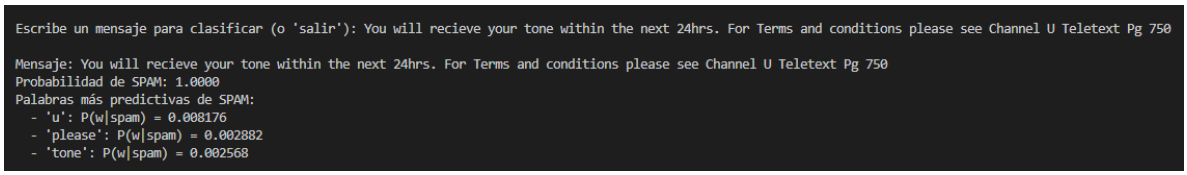
        - Eliminación de signos de puntuación y números: para limpiar caracteres irrelevantes en la mayoría de los casos.
        - Tokenización: se separaron los mensajes en listas de palabras.
        - Eliminación de stopwords usando NLTK: se eliminaron palabras como “the”, “is”, “at”, que no aportan valor predictivo.
    - Considere en usar lematización o stemming, pero dado el tamaño reducido del dataset y su simplicidad, se priorizó mantener las palabras tal cual para no perder contexto o significado.
    - Estas decisiones fueron las que me ayudaron a reducir el tamaño del vocabulario y mejorar la precisión del modelo.

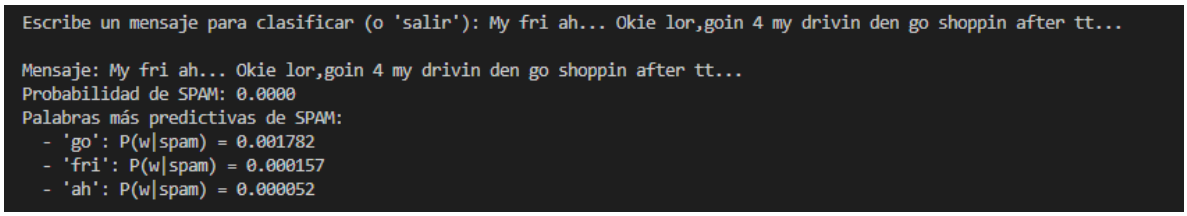
c. *Modelo*
    - El modelo utilizado fue Naive Bayes, un clasificador probabilístico que asume independencia condicional entre las palabras del mensaje.
    - El paso a paso fue:
        - Cálculo de las probabilidades base:
            - P(\text{SPAM}) = \frac{\text{# mensajes SPAM}}{\text{Total de mensajes}}
            -
        - Para cada palabra w en el vocabulario, se calculó:
        - Para un nuevo mensaje, se calculó:
        - La probabilidad final se obtuvo aplicando la función sigmoide:

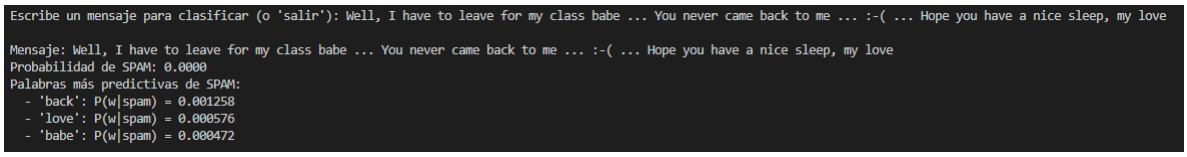
d. *Pruebas de rendimiento.*
    - El dataset fue dividido en 80% para entrenamiento y 20% para prueba.
    - Se calculó la matriz de confusión y a partir de ella las métricas clave:
        - Precisión: qué proporción de los mensajes predichos como SPAM eran efectivamente SPAM.
        - Recall: qué proporción de los mensajes SPAM fueron correctamente detectados.
        - F1-score: la media armónica entre precisión y recall.

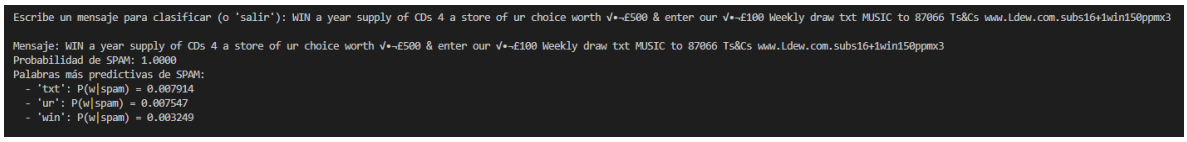
- Se probaron diferentes valores de threshold para ajustar el criterio de clasificación.
- Se encontró que un threshold entre 0.6 y 0.7 ofrecía un buen balance entre identificar correctamente los SPAM sin etiquetar incorrectamente mensajes HAM.

- 

- 

- 

- 

-
- 

e. Discusión de resultados

- El modelo demostró ser eficaz clasificando correctamente todos los mensajes SPAM.
- Las decisiones tomadas durante el preprocesamiento, como la eliminación de stopwords y la normalización del texto, ayudaron a reducir ruido y mejorar el desempeño.
- No obstante, al trabajar con un dataset limitado, algunos mensajes ambiguos o poco comunes pueden generar falsos positivos o negativos.

- El uso de Laplace suavizado evitó errores graves cuando se encontraban palabras nuevas en los mensajes de prueba.
- Las métricas obtenidas reflejan un modelo con buena capacidad general, aunque susceptible a mejoras si se incluyeran técnicas como lematización, análisis de bigramas o un dataset más amplio.

Presentación en vivo

En conjunto con la programación del clasificador, debe construir un módulo que cumpla con lo siguiente:

- Acepta el ingreso de un prompt tipo texto.
- Retorna la posibilidad de que el texto anterior sea SPAM y también las 3 palabras con mayor poder predictivo (mayor probabilidad de Spam). Rubrica

Trabajo escrito	EDA	10 pts
	Limpieza de datos	10 pts
	Modelo	10 pts
	Pruebas de rendimiento	20 pts
	Discusión	20 pts
Presentación presencial	Resultados	30 pts

Anexo

Probabilidad de que un texto sea SPAM dado que contiene la palabra W:

$$P(S|W) = \frac{P(W|S)P(S)}{P(W|S)P(S) + P(W|H)P(H)}$$

Donde,

P(S|W) es la probabilidad de que un texto sea SPAM dado que contiene la palabra W. P(W|S) es la probabilidad de la palabra W aparezca en un texto que es SPAM. P(W|H) es la probabilidad de la palabra W aparezca en un texto que es HAM. P(S) es la probabilidad de que cualquier texto sea SPAM.

P(H) es la probabilidad de que cualquier texto sea HAM.

Probabilidad de que un texto sea SPAM dado que contiene las palabras W_1 a W_n:

$$P(S|\boldsymbol{W}) = \overline{(1 - P_1 P P_2) \ldots (1 P - P_n P_2) \ldots (1 - P_n) P_1 P_2 \ldots P_n} + \ldots$$