UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

Guatemala 05 de Mayo de 2024
Manuel Rodas - 21509

Inteligencia Artificial

Proyecto 02

```
Matriz de Confusión:
[[1576   11]
 [  27  225]]
Informe de Clasificación:
              precision    recall  f1-score   support

         ham       0.98      0.99      0.99      1587
        spam       0.95      0.89      0.92       252

    accuracy                           0.98      1839
   macro avg       0.97      0.94      0.96      1839
weighted avg       0.98      0.98      0.98      1839
```

Precisión y Recall:

- HAM: Con una precisión de 0.98 y un recall de 0.99, el modelo clasifica con gran precisión la mayoría de los mensajes HAM y tiene pocos falsos negativos.
- SPAM: Con una precisión de 0.95 y un recall de 0.89, el modelo clasifica la mayoría de los mensajes SPAM correctamente, pero tiene más falsos negativos en comparación con HAM.

F1-Score:

- Los valores de f1-score (0.99 para HAM y 0.92 para SPAM) indican un buen equilibrio entre precisión y recall, pero muestran que el modelo tiene un rendimiento ligeramente menor en la clasificación de SPAM en comparación con HAM.

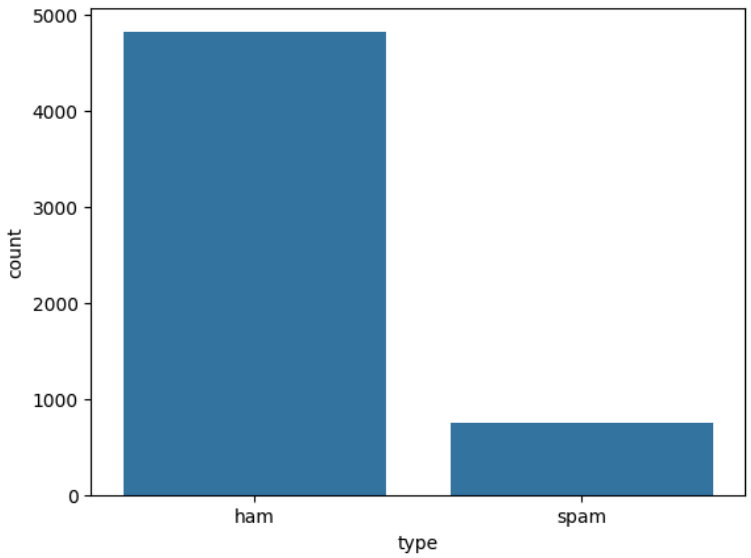
Exactitud:

- La exactitud del modelo es de 0.98, indicando que en general el modelo clasifica correctamente la gran mayoría de los casos.

Con la limpieza del modelo, pudimos llevar el texto a su mini expresion, osea un texto muy básico, que nos facilitaba la clasificación del mismo, ya que nos permite ver más similitudes, además de poder observar así un patrón y una probabilidad más rápida sobre como es el texto spam y como es el texto ham. La limpieza de los datos nos permitió poder generar respuestas más exactas, ya que la limpieza nos permite obtener un modelo más preciso y robusto. Además con la librería nltk, se nos hizo más fácil la interpretación de mensajes, ya que podíamos llevar las palabras a su forma más básica y primitiva, esto con funciones como 'WordNetLemmatizer' y 'PorterStemmer'.

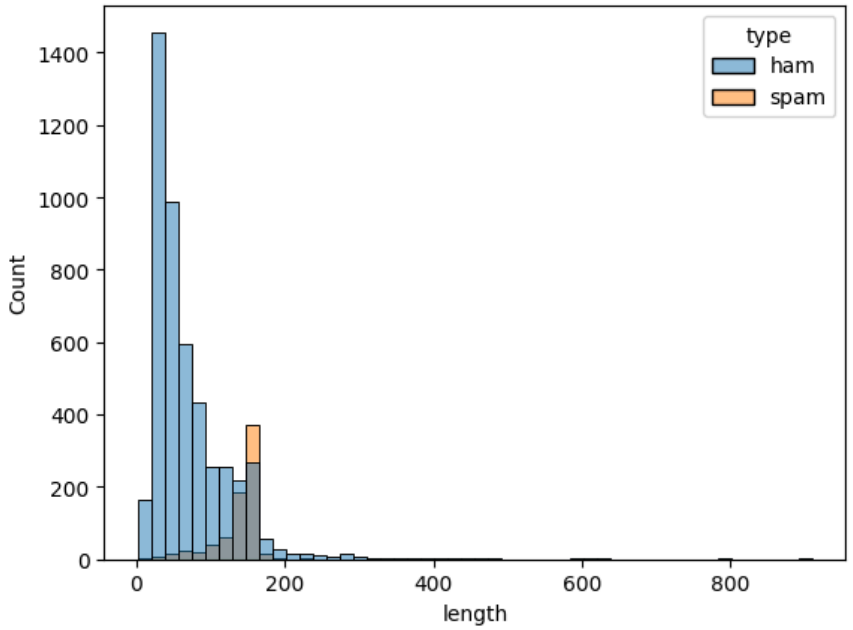
Pero no todo siempre es bueno, ya que a la hora de que en la limpieza se eliminar palabras comunes o eliminar palabras entre comillas, se podía perder mucho el sentido del mensaje, y quiera que no, existen muchos mensajes que se basan de esas palabras que fueron eliminadas, esto llega a crear un hoyo en nuestro programa, ya que son ciertas ocasiones que no se toman en cuenta y que pueden dañar la precisión de nuestras probabilidades y asi nuestro resultado.

Para poder realizar este proyecto, fue necesario cargar primero una base de datos que ya organizaban los mensajes en spam y ham. Luego se realizo cierto analisis exploratorio, que nos proporcionó una idea de como se maneja la base de mensajes que se estaba utilizando.

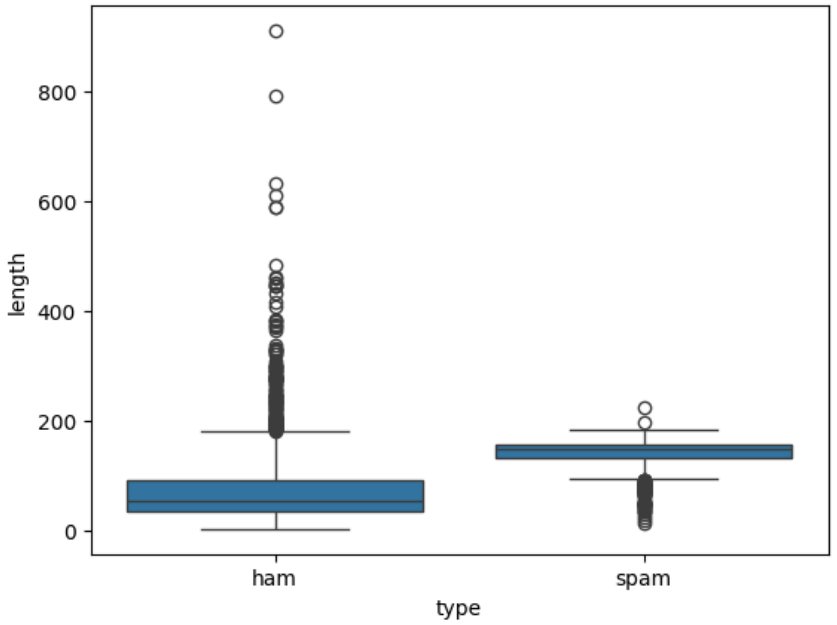


Podemos notar una cantidad más alta de mensajes ham que de mensajes spam. Esto se puede explicar de mejor manera, ya que es mas común recibir mensajes 'normales' que de spam.

Luego se realizó un conteo de lo largo en cuestión de letras que son los tipos de mensajes, esto con el fin de ver si por alguna razón se notaba una tendencia en que algún tipo de mensaje fuera más largo que otro.



Podemos ver como existen mensajes ham, que son más largos que cualquier otro mensaje spam, pero también se llega a notar cierto patrón, que muestra una tendencia en que por porcentaje, la mayoría de los mensajes spam son más largos que la mayoría de mensajes ham.



En conclusión podemos decir que las métricas obtenidas muestran que el modelo tiene un buen rendimiento general, pero podría mejorar en la clasificación de SPAM. Si se desea mejorar aún más el rendimiento, debemos realizar una limpieza aún más exhaustiva, con el fin de poder llevar los mensajes a un estado 'limpio' o a su estado más puro, para poder realizar comparaciones más correctas y precisas.