INTELIGENCIA ARTIFICIAL

PROYECTO FINAL

ADRIAN MADRID ROMERO

18-12-2024

New Technology School

Tokio.

New
Technology
School

PROYECTO: DETECTOR DE SPAM EN SMS DE TEXTO

APARTADOS

1. **Objetivo del proyecto**
2. **Obtención de los datos y Data argumentación**
3. **Preprocesamiento de los datos y Feature Engineering**
4. **Estrategias con diferentes modelos**
5. **Mejor modelo y resultados**
6. **SHAP con el mejor modelo**
7. **Script de Python con prompt para usar el modelo**
8. **Conclusiones**

1. Objetivo del Proyecto

El objetivo principal de este proyecto fue desarrollar un algoritmo capaz de predecir si un mensaje de texto (SMS) es **SPAM** o NO **SPAM**, utilizando técnicas de preprocesamiento de datos y diversos modelos de **Machine learning** y **Deep learning**. La idea era crear una herramienta práctica y efectiva que pudiera abordar un problema real: la detección de mensajes no deseados que interrumpen la comunicación y, en ocasiones, representan riesgos de seguridad para los usuarios.

Desafíos y Soluciones

El mayor desafío encontrado durante el desarrollo fue la escasez de datos y su distribución altamente desbalanceada en el dataset inicial, donde los mensajes etiquetados **como NO SPAM** eran considerablemente más numerosos que los de **SPAM**. Este desbalance podía sesgar el modelo y afectar su capacidad de identificar correctamente mensajes fraudulentos.

Para superar esta limitación, se implementaron técnicas de data augmentation para generar mensajes **SPAM** adicionales, logrando un conjunto de datos más equilibrado. Aunque se exploraron otros datasets en busca de más datos, no se consideraron apropiados debido a la falta de representatividad o a la baja calidad de sus mensajes. Este enfoque permitió entrenar el modelo en un dataset más robusto, representativo y alineado con los objetivos del proyecto.

En resumen, este proyecto no solo busca desarrollar un detector de **SPAM** para **SMS**, sino también enfrentar y resolver limitaciones propias del mundo real, como la falta de datos o problemas de desbalanceo, asegurando un enfoque sólido y bien fundamentado.

2. Obtención de los datos y Data Argumentación

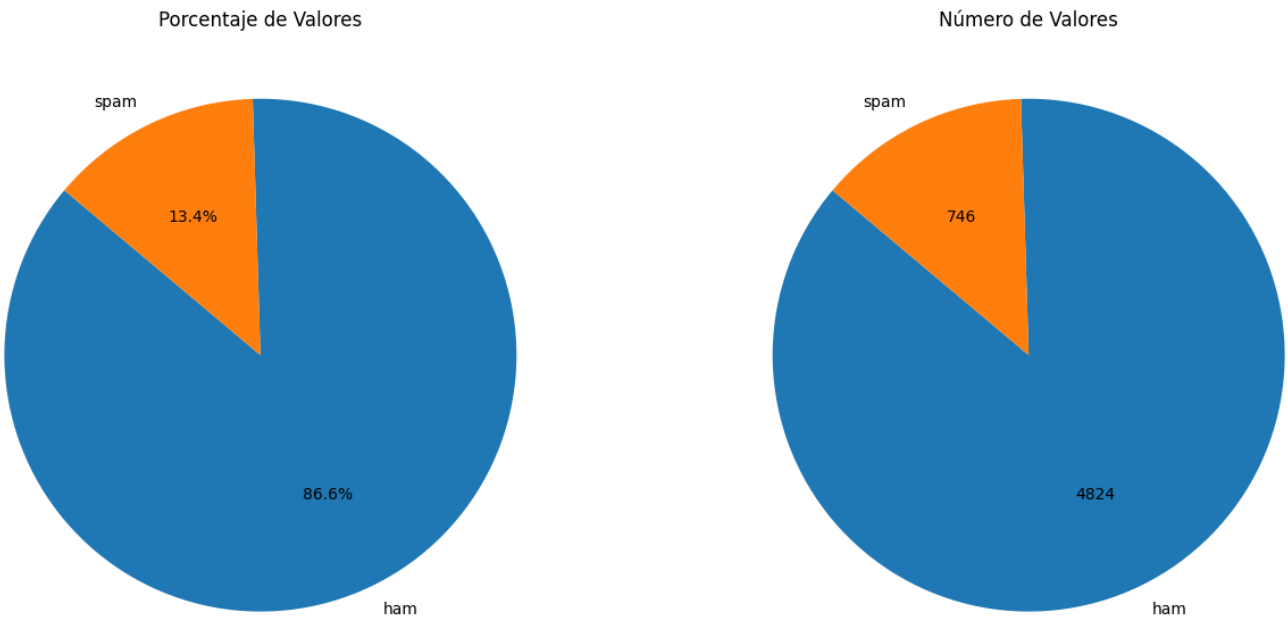
2.1 Análisis inicial del dataset:

Se comenzó el proceso cargando y analizando el dataset original utilizando las librerías Pandas y Matplotlib. El objetivo inicial era identificar posibles desbalances en los datos y entender su distribución.

Para esto, se ejecutaron las siguientes acciones:

- Se cargó el dataset y se analizaron las etiquetas (clases) existentes para detectar desbalance entre las clases "spam" y "no spam".
- Se visualizaron los datos usando gráficos de barras y otros métodos de visualización.

A continuación, se presenta un gráfico que muestra el desbalance identificado en el dataset original:



2.2 Generación de mensajes spam ficticios

Dado que el dataset original presentaba un desbalance significativo, se procedió a generar un total de 4000 mensajes ficticios de spam para equilibrar las clases. Este proceso fue cuidadosamente diseñado para garantizar realismo y diversidad en los mensajes creados.

Estrategia de generación

1. Construcción de listas de frases:
    - Se crearon listas de frases segmentadas en tres partes:
        - Entradas: frases iniciales que captan la atención del usuario (ej.: "Tu regalo especial te espera!").
        - Temas y acciones: frases centrales que describen la acción o promesa del mensaje (ej.: "Gana hasta 1000$ respondiendo ahora").
        - Cierres: frases finales que invitan a la acción (ej.: "Oferta válida hasta la medianoche. No lo pierdas!").
2. Generadores de URLs y números:
    - Se crearon funciones para generar URLs ficticias, agregando realismo al incluir palabras clave comúnmente usadas en spam (ej.: www.claimoffer4u-special.net).

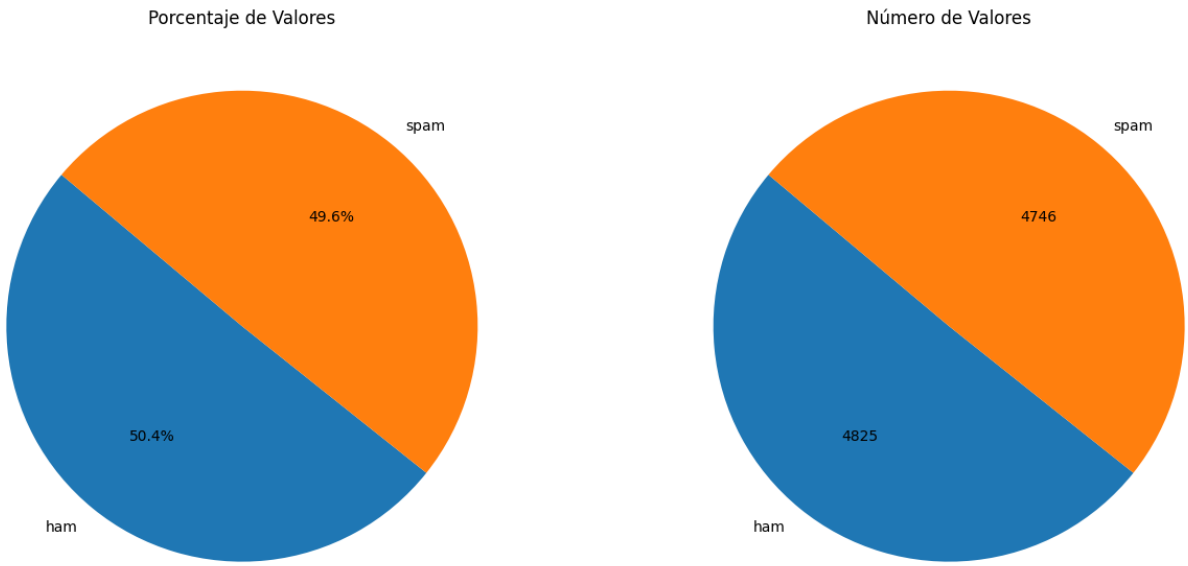
- También se incluyeron números aleatorios y símbolos como %, $ y mayúsculas en palabras clave para imitar patrones comunes.

3. Combinación de frases y elementos:
   - Se combinaron frases de las listas con URLs y símbolos especiales de forma aleatoria para crear mensajes únicos.
4. Control de duplicados:
   - Para garantizar que no se repitieran mensajes, se utilizó un conjunto (set) que almacenaba cada mensaje generado y evitaba duplicados.
5. Diversidad y realismo:
   - Se implementaron sinónimos, variaciones en estilo y el uso de diferentes estructuras para asegurar la mayor diversidad posible.

Ejemplos de mensajes generados:

```
Your gift inside! Your loyalty earned you a £150 discount! . Restrictions may apply.
Your number won a prize in our weekly draw! Don't miss out! . First come, first served.
great opportunity! Loan approved! Call now to finalize up to £50000. . available for a limited time.
Win now! Your number won a prize in our weekly draw! . First come, first served.
Redeem your special prize today! Flash sale: Visit www.claimoffer4u-special.net. Offer valid for 24 hours only. (One per user)
Claim your FREE mobile upgrade. Exclusive offer for loyal users! attention! . Valid until midnight.
Unlock a 40 percent discount today. Top priority: . available for a limited time. (For users 18+)
Limited time only: Join our exclusive prize draw! Limited-time offer: Visit www.exclusiveprizes.com. exclusive access for a limited time only.
Top priority: UpgrdCentre: FREE Camera Phone upgrade for valued customers . Don’t miss out!
Exciting news. limited time only: join our exclusive prize draw! . Offer ends soon!
immediate attention required Win £1000 instantly by replying now! . Act fast to avoid missing out.
great opportunity! Act NOW. Last chance to redeem your prize. Visit www.freesample4u-special.net. One-time opportunity.
Your gift inside! Special cinema pass for 2, valid for 1 year. Call now! . Available for a limited time.
```

Después del balanceo:



De esta forma, se pudo eliminar el problema de ese desbalanceo de los datos que nos iba a dar problemas en el futuro con el modelo ya que seguramente pueda predecir mejor los casos de no spam que de spam, y la idea es que sepa generalizar bien.

3. Preprocesamiento de los datos y Feature Engineering

En este apartado se realizó un proceso de **preprocesamiento** y **Feature Engineering** para enriquecer el análisis de los datos y mejorar el rendimiento del modelo.

Ingeniería de Características

Se agregaron nuevas características al dataset a partir del análisis de los textos. Estas características se diseñaron para capturar patrones específicos que son comunes en mensajes de spam:

1. **special_char_count**:
    - Representa el número de caracteres especiales en el mensaje, como @, $, % o &, que suelen ser comunes en mensajes de spam.
2. **exclamation_count**:
    - Cuenta la cantidad de signos de exclamación ! presentes en el texto, ya que los mensajes de spam tienden a usar exclamaciones para generar urgencia o llamar la atención.
3. **number_count**:
    - Calcula el número de dígitos en el mensaje. Los mensajes de spam incluyen números para montos de dinero, códigos o teléfonos.
4. **emoticon_count**:
    - Identifica la cantidad de emoticonos en el texto, más comunes en mensajes personales o no spam.
5. **message_length**:
    - Longitud total del mensaje en caracteres. Los mensajes de spam suelen ser más breves y directos, a diferencia de los mensajes no spam.

6. **spam_word_count**:
    - Cuenta las palabras típicas de spam, como *"win"*, *"free"*, *"offer"*, entre otras. Esto permite identificar la frecuencia de términos asociados a mensajes de spam.
7. **stop_word_ratio**:
    - Proporción de palabras vacías (stop words) en relación con el total de palabras del mensaje. Los mensajes no spam suelen contener más palabras vacías debido a un lenguaje más natural.
8. **uppercase_word_count**:
    - Número de palabras escritas completamente en mayúsculas, como *"WIN"* o *"OFFER"*. Estas palabras se utilizan frecuentemente para destacar promociones o mensajes urgentes.
9. **url_count**:
    - Cantidad de URLs incluidas en el mensaje. Los mensajes de spam suelen incorporar enlaces para redirigir a sitios externos.

Así sería un ejemplo de cómo quedaría el dataset con las nuevas características:

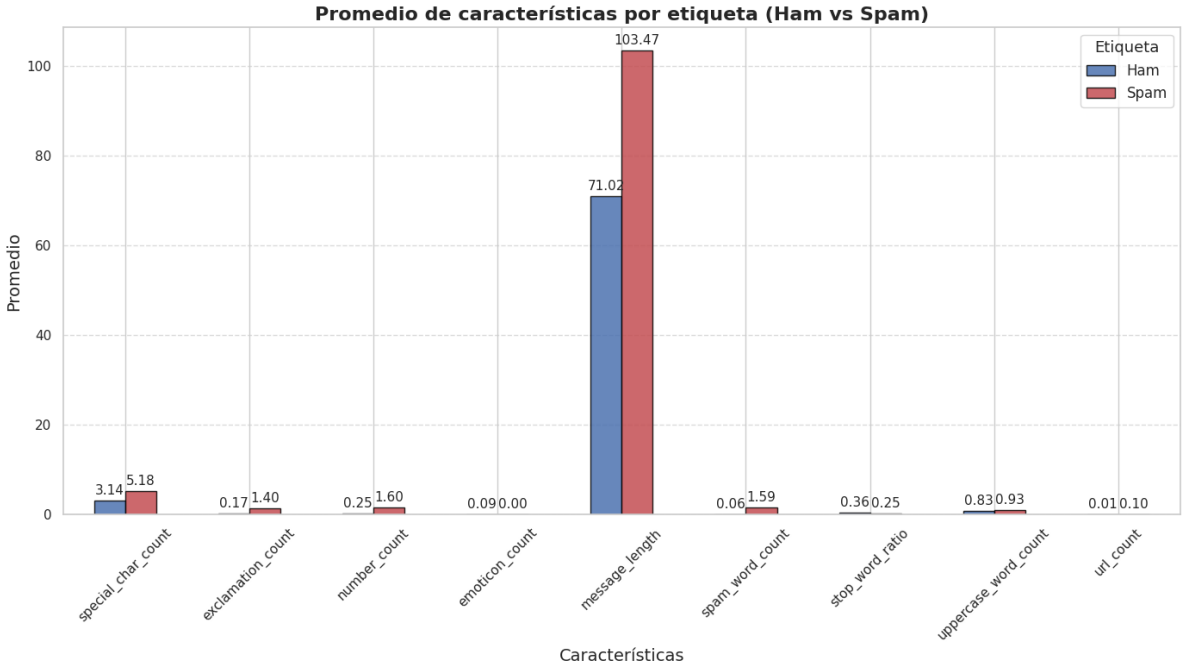
Antes:

	etiqueta	sms	origen
0	ham	Go until jurong point, crazy.. Available only ...	original
1	ham	Ok lar... Joking wif u oni...	original
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	original
3	ham	U dun say so early hor... U c already then say...	original
4	ham	Nah I don't think he goes to usf, he lives aro...	original

Después:

	etiqueta	sms	origen	special_char_count	exclamation_count	number_count	emoticon_count	message_length	spam_word_count	stop_word_ratio	uppercase_word_count	url_count
0	ham	Go until jurong point, crazy.. Available only ...	original	8.0	0.0	0.0	0.0	111.0	0.0	0.200000	0.0	0.0
1	ham	Ok lar... Joking wif u oni...	original	6.0	0.0	0.0	0.0	29.0	0.0	0.000000	0.0	0.0
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	original	4.0	0.0	6.0	0.0	155.0	2.0	0.178571	2.0	0.0

Aquí muestro una gráfica muy interesante de como se comporta cada característica respecto a los mensajes de SPAM y NO SPAM.



RESUMEN DE LA GRÁFICA

La gráfica muestra las diferencias promedio de características entre mensajes **SPAM** y **NO SPAM**. Los mensajes SPAM tienen más **caracteres especiales**, **signos de exclamación**, **números**, **palabras en mayúsculas** y **URLs**, además de ser más largos y contener más palabras asociadas al SPAM. En cambio, los mensajes NO SPAM presentan un mayor uso de **emoticonos** y **stop words**. Estas diferencias son clave para identificar patrones y mejorar el rendimiento del modelo de detección.

Preprocesamiento de los Datos

Además de la creación de características, se realizó un **preprocesamiento de los textos** para preparar los datos:

1. **Limpieza del texto**:
    - Se eliminaron caracteres especiales innecesarios, signos de puntuación y números no relevantes.
    - Se aplicó conversión a minúsculas para normalizar las palabras.

2. **Eliminación de palabras con menos de dos caracteres**:
    - Tras realizar un análisis de frecuencias (word count), se observó la presencia de palabras "raras" o poco informativas, por lo que se eliminaron aquellas con menos de dos caracteres.
3. **Tokenización y análisis de palabras clave**:
    - Se procesó el texto en tokens (palabras) para realizar un conteo de palabras frecuentes y enriquecer el análisis del contenido.

Aquí muestro mensajes antes del preprocesamiento y después de la limpieza:

```
SMS 1 Original: URGENT! You have won a 1 week FREE membership in our â£100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18
SMS 1 Limpio: urgent win number week free membership numbernumber prize jackpot txt word claim number tc url lccltd pobox 4403ldnw1a7rw18

SMS 2 Original: I only haf msn. It's yijue@hotmail.com
SMS 2 Limpio: haf msn yijuehotmailcom

SMS 3 Original: -PLS STOP bootydelious (32/F) is inviting you to be her friend. Reply YES-434 or NO-434 See her: www.SMS.ac/u/bootydelious STOP? Send STOP FRND to 62468
SMS 3 Limpio: pls stop bootydelious numberf invite friend reply yesnumber nonumber url stop send stop frnd number

SMS 4 Original: Are you unique enough? Find out from 30th August. www.areyouunique.co.uk
SMS 4 Limpio: unique find 30th august url

SMS 5 Original: 500 New Mobiles from 2004, MUST GO! Txt: NOKIA to No: 89545 & collect yours today!From ONLY å£1 www.4-tc.biz 2optout 087187262701.50gbp/mtmsg18
SMS 5 Limpio: number new mobile number txt nokia number   collect todayfrom number url 2optout number50gbpmtmsg18

SMS 6 Original: Congratulations ur awarded 500 of CD vouchers or 125gift guaranteed & Free entry 2 100 wkly draw txt MUSIC to 87066 TnCs www.Ldew.com1win150ppmx3age16
SMS 6 Limpio: congratulation ur award number cd voucher 125gift guaranteed   free entry number number wkly draw txt music number tnc url

SMS 7 Original: Ur ringtone service has changed! 25 Free credits! Go to club4mobiles.com to choose content now! Stop? txt CLUB STOP to 87070. 150p/wk Club4 PO Box1146 MK45 2WT
SMS 7 Limpio: ur ringtone service change number free credit club4mobilescom choose content stop txt club stop number 150pwk club4 po box1146 mk45 2wt

SMS 8 Original: HMV BONUS SPECIAL 500 pounds of genuine HMV vouchers to be won. Just answer 4 easy questions. Play Now! Send HMV to 86688 More info:www.100percent-real.com
SMS 8 Limpio: hmv bonus special number pound genuine hmv voucher win answer number easy question play send hmv number infourl
```

En resumen, se obtuvieron nuevas características basadas en los textos que aportan valor al modelo y se prepararon los datos de manera eficiente para la etapa de modelado.

4. Estrategias con diferentes modelos

En este apartado se implementaron diversas estrategias y modelos de aprendizaje automático para la clasificación de mensajes spam y no spam, utilizando un enfoque progresivo de complejidad. A continuación, se describen las estrategias aplicadas:

Para la división del conjunto de datos:
Se realizó una división en entrenamiento y prueba para garantizar una evaluación adecuada:

- Entrenamiento: Incluye mensajes spam ficticios, el 70% de los no spam originales y un 10-20% de los spam originales.
- Prueba: Compuesta exclusivamente por mensajes spam originales y no spam originales para evaluar la capacidad de generalización del modelo.

Es importante recalcar no usar los mensajes ficticios como test ya que al ser inventados no serían unas predicciones reales, por eso era interesante incluir solo mensajes originales en el test para ver realmente si de verdad cumple el modelo.

Los datos fueron preprocesados, normalizando las características numéricas mediante **MinMaxScaler** y reemplazando mensajes vacíos por el valor ***"mensaje_vacio"***.

1. ***TF-IDF y Gradient Boosting***:
   En primer lugar, se aplicó la técnica TF-IDF para vectorizar los textos, evaluando el rendimiento con y sin características adicionales numéricas. La inclusión de las nuevas características resultó en mejores resultados. Posteriormente, se implementó el modelo Gradient Boosting, cuya optimización se realizó mediante GridSearchCV, obteniendo los siguientes hiperparámetros óptimos:
   - learning_rate: 0.1, max_depth: 5, n_estimators: 150, min_samples_leaf: 2.

Estos fueron los resultados:

Resultados en el conjunto de prueba:

	precision	recall	f1-score	support
0	0.93	0.99	0.96	1448
1	0.97	0.70	0.81	374
accuracy			0.93	1822
macro avg	0.95	0.85	0.89	1822
weighted avg	0.94	0.93	0.93	1822

2. ***Red neuronal con embeddings aleatorios***:
   Se construyó una red neuronal combinando dos tipos de entradas:
   - Texto vectorizado con secuencias y embeddings aleatorios.
   - Características numéricas preprocesadas.

El modelo obtuvo una precisión del 96% en el conjunto de prueba y un F1-Score para Spam de 0.89, mejorando respecto al Gradient Boosting.

Reporte de clasificación:

	precision	recall	f1-score	support
No Spam	0.97	0.98	0.97	1448
Spam	0.92	0.86	0.89	374
accuracy			0.96	1822
macro avg	0.94	0.92	0.93	1822
weighted avg	0.96	0.96	0.96	1822

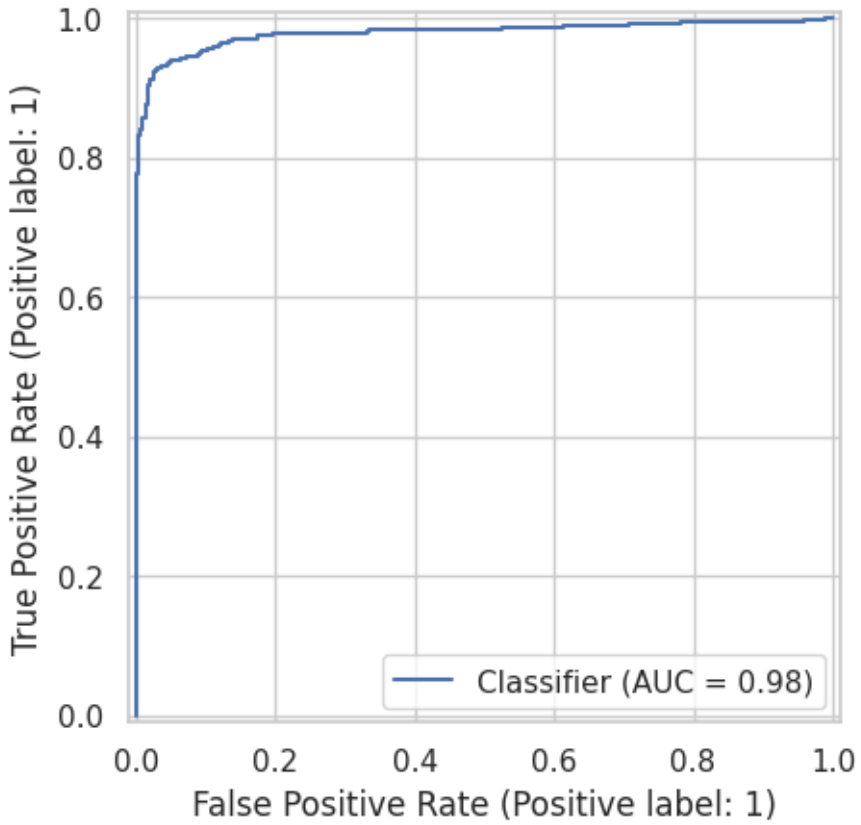
3. ***Red neuronal con embeddings preentrenados (GloVe):***
   Finalmente, se integraron embeddings preentrenados de GloVe (100 dimensiones), congelando los pesos y combinándolos nuevamente con las características numéricas. La arquitectura de la red se ajustó con regularizaciones (Dropout y BatchNormalization), y se utilizó *EarlyStopping* para evitar el sobreajuste.

Este modelo obtuvo los mejores resultados:

```
Reporte de clasificación:
              precision    recall  f1-score   support

     No Spam       0.97      0.98      0.98      1448
        Spam       0.93      0.89      0.91       374

    accuracy                           0.96      1822
   macro avg       0.95      0.94      0.94      1822
weighted avg       0.96      0.96      0.96      1822
```



Métricas de evaluación:

La métrica principal seleccionada fue el F1-Score, al ser la más adecuada para un problema de clasificación con clases desbalanceadas. A pesar de que previamente se balancearon los mensajes pareció más correcto usar F1 score ya que el objetivo final era que el modelo generalizara bien en el conjunto de prueba desbalanceado, donde un modelo podría fallar en detectar spam si se enfocara en la precisión general. Al usar F1-Score se asegura uno de tener una métrica robusta y equilibrada para medir el rendimiento global del modelo.

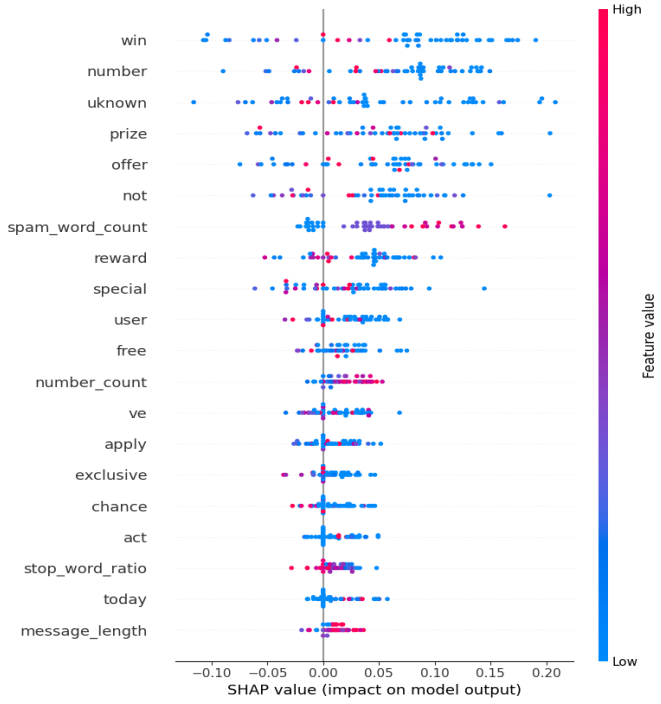
---

5. Mejor modelo y resultados

Tras evaluar distintas estrategias y modelos, el mejor rendimiento se obtuvo con la **red neuronal utilizando embeddings preentrenados (GloVe)**. Este modelo alcanzó un **F1-Score** del **91%** para la clase "Spam" y una **precisión general** del **96%** en el conjunto de prueba. Además, logró un **AUC** de **0.9789**, lo que refleja su capacidad para discriminar entre mensajes "Spam" y "No Spam" de manera efectiva.

El modelo fue guardado para ser utilizado en **futuras predicciones**, permitiendo automatizar el proceso de clasificación de mensajes en tiempo real o en nuevos lotes de datos. Esto ofrece la posibilidad de realizar ajustes incrementales conforme se incorporen nuevos datos, mejorando así su desempeño a lo largo del tiempo.

6. SHAP con el mejor modelo



Se utilizó la librería **SHAP aprendida en el módulo de IA** para entender qué características influyeron más en las predicciones del modelo. En el gráfico:

- **Eje vertical**: características del modelo (palabras y métricas).
- **Eje horizontal**: el impacto de cada característica en la predicción del modelo (positivo o negativo).
- **Colores**:
  - **Azul**: valores bajos de la característica.
  - **Rojo**: valores altos de la característica.

Los resultados muestran que palabras como **"win"**, **"prize"**, **"offer"** y métricas como **"spam_word_count"** y **"number_count"** tienen un impacto significativo, ya que están asociadas con mensajes "spam". Características como **"message_length"** y **"stop_word_ratio"** también influyen, pero en menor medida.

Este análisis fue útil e interesante para confirmar que el modelo se enfocó en patrones esperados y relevantes para detectar spam, como palabras clave comunes y características numéricas específicas. Muy interesante de cara a mejorar el modelo también si se ve que está dándole importancia a palabras o características que no lo son tanto.

7. Script de Python con prompt para usar el modelo

Para facilitar el uso del modelo, se desarrollo un **script en Python** organizado en varios archivos utilizando **Sublime Text**. La estructura incluye:

1. **Entorno virtual**: Se creó un entorno virtual para evitar conflictos de librerías.
2. **Una carpeta de varios archivos para organizar el código donde algunos de ellos son**:
   - Un archivo que contiene las **funciones de preprocesamiento** e **ingeniería de características**.
   - Otro archivo realiza la **división del texto** y prepara los datos.
   - Un archivo almacena los **datos limpios** para su reutilización.
   - Otro archivo realiza un **testeo del modelo** con una lista de SMS de prueba.
   - Finalmente, un archivo incluye el **prompt** para ingresar un SMS individual. Este paso:
     - Traduce el mensaje al **inglés** (independientemente del idioma original).
     - Procesa y clasifica el mensaje como **spam o no spam** utilizando el modelo entrenado.

8. Conclusiones

Finalmente, y ahora hablo en primera persona, ha sido un proyecto que dentro de lo simple que podía parecer a priori, me encontré con una serie de dificultades, entre otras la de los pocos datos y estar tan desbalanceados. Por eso quise enfocarme sobre todo en hacer un buen preprocesamiento de los mismos para compensar esa falta de datos.

Este proyecto fue desarrollado en Google Colab debido a que tiene herramientas de gpu y tpu que permiten un entrenamiento y operaciones matemáticas más rápidas. Un proyecto bastante completo, ya que cubre ML y DL, con elementos de minería de datos en el preprocesamiento.

Me he quedado realmente sorprendido con los resultados del modelo ya que siendo entrenado con 4000 mensajes ficticios, a la hora de sacar las predicciones con mensajes reales de spam a estado bastante acertado y eso quiere decir que se hizo un buen trabajo con los datos y fue realmente la parte más tediosa pero imprescindible para obtener esos resultados.

Probé utilizando Transformers como Bert porque quería seguir mejorando los resultados pero no fue así, y entiendo que puede llegar a ser por la falta de datos ya que esos modelos necesitan muchos datos y si les das pocos tienden al sobreajuste y no funcionan tan bien como modelos algo más simples.

De cara al futuro, si que convendría para mejorar aun más los resultados obviamente entrenar con más datos y que sean reales.

Como broche final quise crear un prompt en un script porque me parecía que podía ser dinámico y divertido el hecho de tener el modelo, fruto de mi trabajo de esta manera. Además de tenerlo guardado y poder usarlo en el futuro, incluso mejorarlo.

INTELIGENCIA ARTIFICIAL
PROYECTO FINAL
Nombre y Apellidos