UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Master en Big Data Analytics

2023-2024

TextMining en las Redes Sociales

Memoria proyecto IberAuTexTification

Alumno:

José Carlos Ávila Palazón

Nickname:

Joavpa

Tabla de contenido

1. Introducción del proyecto

En esta memoria vamos a intentar explicar todo el proceso abordado para poder llegar a la solución más optima del modelo.

El objetivo trata de intentar generar un modelo que prediga si un texto esta generado por IA o está escrito por una persona.

Esto nos lleva a realizar un proceso en el que se puede diferenciar en cinco partes principalmente:

1. Análisis de los datos: tratar de sacar la mayor información posible, explorando los datos utilizando visualizaciones.

2. Limpieza de los datos: se deberá quitar NA, revisar caracteres extraños si procede,etc.

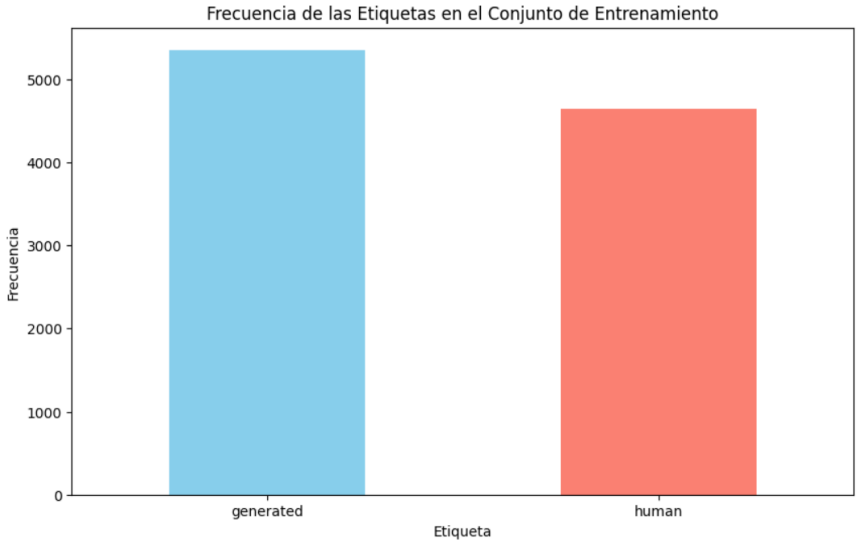
3. Procesamiento de los datos: esta parte aborda teniendo más información por los pasos anteriores, intentar transformarlos utilizando técnicas para aportar al algoritmo más información útil y vectorizar para que el modelo pueda ejecutarse.

4. Entrenamiento del modelo y Evaluación: en este punto se abordan el testing de distintos modelos y el ajuste de distintos hiperparámetros, con el fin de encontrar el mejor rendimiento posible, a su vez una representación de los resultados obtenidos con la predicción.

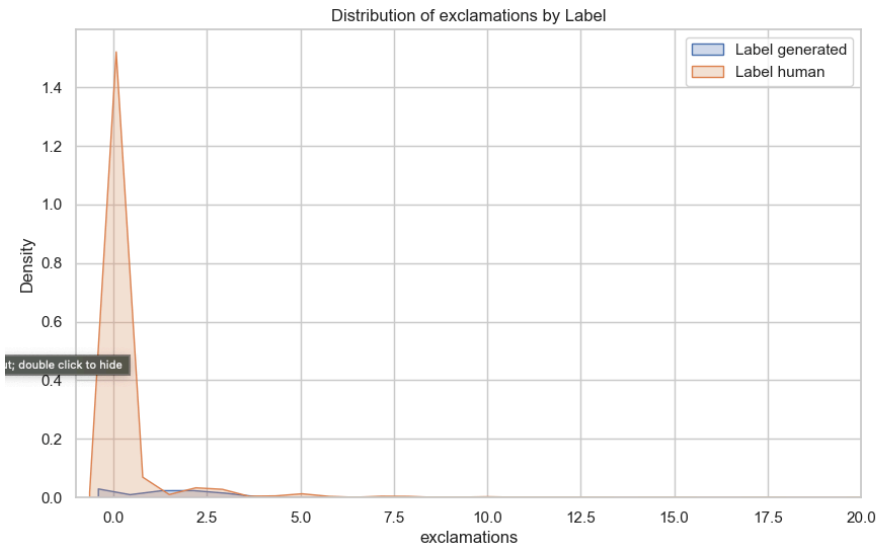
2. Análisis de los datos

En este apartado comentaremos toda la información que hemos llegado a obtener respecto a nuestro dataframe.

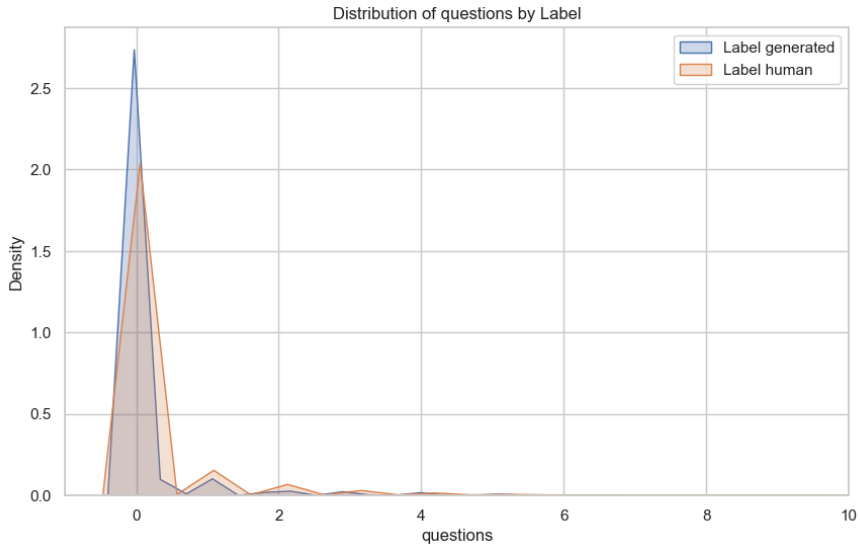
Tenemos para el entrenamiento 100 mil registros aprox, y una distribución del label bastante correcta.



Si revisamos caracteres que pueden ser determinantes como los signos de interrogación o exclamación podemos ver lo siguiente:

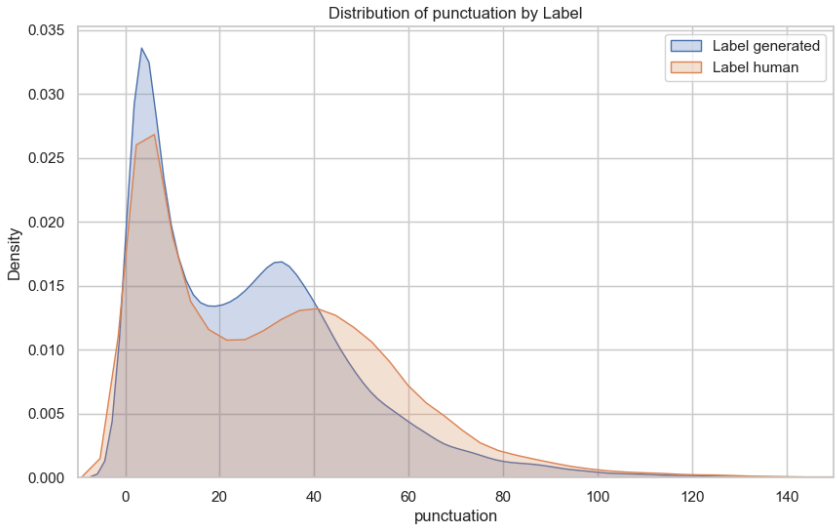


Vemos visiblemente una gran diferencia en las exclamaciones entre los textos escritos por humanos y por IA, esto luego nos puede ayudar mucho a entrenar el modelo.



Por parte de los signos de interrogación lo vemos bastante más equilibrado, pero con mayores picos respecto a los textos generados.

Y por último revisamos los signos de puntuación respecto a los dos label:



Se puede ver que a distribución se asemeja, con mayores picos de densidad por parte de los textos generados.

Con esto sacamos conclusiones de que los textos escritos por humanos pretenden expresar más emociones, enfatizar, etc. Es por esto por lo que se utilizan significativamente más signos de exclamación, no obstante, tienden a utilizar menos signos de puntuación, es decir, frases menos estructuradas y completas, pero no de una forma notablemente considerable.

3. Limpieza y Procesamiento de los datos

En primer lugar, vamos a dropear cualquier columna que aparezcan NA, no nos aporta ningún valor esas columnas.

Luego, caracteres como @ se deben de considerar ya que son textos de redes sociales, por lo tanto, no consideramos limpiar los datos de esos tipos de caracteres.

Se ha considerado también eliminar Números y editar los textos a minúsculas, pero los resultados han disminuido.

1. Procesamiento de los datos

Esta parte vamos a ir paso por paso junto con para cada parte comentarios sobre otras soluciones que se han desconsiderado.

1.1. Detección de errores gramaticales y ortográficos en los textos

La detección de errores de este tipo ha sido posible por la librería language_tool_python y langdetect para la detección de los idiomas.

Simplemente detectamos el idioma y se lo pasamos por parámetro a la función que detecta con ayuda de la librería los errores por texto.

Ventajas: Nos aporta mucha información, ya que los humanos tendemos a hacer errores de escritura sin darnos cuenta, es un aspecto muy diferencial.

Inconvenientes: Tiene un gran coste computacional, demora mucho tiempo de ejecución.

Para nuestro caso que debemos en poco tiempo dar un fichero con nuestros resultados, no es posible poder incorporarlo en nuestro proceso, no obstante, para un proyecto de mayor tiempo es algo muy interesante.

1.2. Detección de caracteres gramaticales como son exclamaciones, interrogaciones y signos de puntuación

Las personas tendemos a expresar efusividad, somos seres emociales y esto nos lleva a utilizar en mayor medida estos caracteres de ¡¿.

Además, somos menos concisos, para explicarnos en función de la persona y de su nivel de expresión en la escritura puede llevar a mayor número de frases y a su vez mas signos de puntuación.

1.3. Ratio de palabras únicas

Se trata de hacer una función que calcule el ratio de palabras únicas, esto nos puede dar mucha información, ya que los humanos tendemos a utilizar palabras iguales porque son con las que más cómodos nos podemos sentir.
La verdad que es la feature que mejor ha funcionado de todas.

1.4. Vectorización

Para poder llevar a cabo la ejecución del modelo debemos vectorizar los textos para ser tramitados, no obstante, hay que tener en cuenta que al algoritmo queremos proporcionarle también toda la información de las otras variables (número de exclamaciones, numero de signos de puntuación...).
La solución es vectorizar por una parte el texto, que tras probar con múltiples técnicas (CountVectorizer, Word enbbedings…) por TF-IDF.

Por otra parte, extraemos del dataFrame las características adicionales (interrogantes, exclamaciones y signos de puntuación) las estandarizamos y las combinamos utilizando hstack, añadiéndolas de forma horizontal una a una.

1.5. Procesos desconsiderados

Entendiendo que nuestro objetivo es encontrar patrones de estructura o diseño que definan si es un humano o generado, no a buscar que el modelo entienda el significado del texto, o lo que quiere decir.

Por ello nos lleva a desestimar técnicas como stopwords (no queremos eliminar esas palabras de nuestros textos), tokenizar y lematizar.

También se ha probado la detección de preposiciones en función del idioma del texto, por lógica los humanos nos vamos más por las ramas y para la construcción de las frases tendemos a utilizar más preposiciones, pero los resultados nos han reflejado que no funciona tan bien (es posible mejorar la detección del idioma y con ello mejorar el rendimiento)

4. Entrenamiento del modelo y Evaluación

Tras considerar distintos algoritmos de clasificación como LogisticRegression, Redes Neuronales, Naive Bayes ,SVM y RandomForest.

Hemos escogido LogicRegression, algoritmo para los run2 y el 3.

RandomForest fue nuestra elección inicial, pero tras ver los resultados reflejados en el run1, es posible que provoque overfitting.

Los hiperparámetros hemos escogido los estándares.

5. Conclusiones

Podemos sacar como conclusión general de la tarea de la asignatura es que el alcance es muy grande, el conocimiento para el uso de estas técnicas debe de ser muy amplio y hay que tener muy en cuenta al analizar textos cual es nuestra finalidad, es decir, saber si nuestro modelo tiene como objetivo sacar información del significado del texto o simplemente se necesita analizar su forma y estructura para clasificar por si es generado por IA o no.

Además, también hay que comentar que el tiempo para la elaboración del modelo es reducido, se necesita más tiempo para poder elaborar un modelo con mayor capacidad y con procesos más elaborados.

Respecto al modelo, se ven bastantes diferencias de puntaje entre nuestra validación y posteriormente de los resultados reflejados, esto puede llevar a un overfitting o mal proceso de entrenamiento en nuestro modelo, se tendrían que revisar con detenimiento cada aspecto que es lo que puede influir a que ocurra.