Responde a las siguientes preguntas:

**1. Preprocesamiento de datos:**

**1. ¿Como limpiaría y preprocesaría los datos para un análisis más detallado?**

Para datos no textuales, primero identificaría las variables entre numéricas y categóricas, exploraría si existen missings no declarados, el número de nulos existentes en cada variable, así como de outliers. Para tratamiento de outliers, habría que determinar en función de las distribuciones de las variables si winsorizar o imputar a missings. Por otro lado, para la gestión de missings, para aquellas variables como images que tiene un 96% de datos faltantes se eliminarían por la problemática de imputar un porcentaje tan alto; y para aquellas con una incidencia aproximada de menos del 50% intentaría imputarlas (habría que estudiar con que método: media, mediana, moda, KNN, Cadenas de Markov…)

Para datos textuales, la limpieza pasaría por un análisis preliminar (wordcloud y palabras más usadas), transformar los textos a minúsculas y eliminar las stopwords, eliminar signos de puntuación y tokenizarlos.

**2. ¿Como normalizarías los datos?**

Una vez tokenizados, pasaría a lematizar mis tokens, normalizaría números y URLs generando un token para su respectiva representación textual (98 🡪 “NUM” // https://...... 🡪 “URL”), identificaría emojis para generar una nueva variable que los recoja como información adicional (posteriormente intentaría extraer su sentimiento para introducirlo al modelo), así como extraería la polarity y subjetivity de los textos para generar características adicionales que aporten a la detección de sentimientos en el modelo.

En resumen, si tuviese que sacar los puntos principales del proceso a seguir del preprocesado y normalización de textos sería:

* Eliminar espacios extra entre términos.
* Transformar a minúsculas.
* Reemplazar contracciones del inglés.
* Normalizar URLs y números por tokens comunes.
* Lematizar tokens, eliminar stopwords y eliminar sígnos de puntuación.

**2. Análisis exploratorio de los datos:**

**1. ¿Cuáles son los términos relevantes más frecuentes en los comentarios? ¿Encuentras diferencias significativas entre la distribución de términos provenientes de comentarios positivos y negativos? Para esta sección se entiende comentario positivo por aquel que termina recomendando el producto**

No he encontrado una gran diferencia entre términos previo a normalizar mis textos, incluso ya normalizados hay ligeras diferencias, pero no muy significativas. Los tokens “game” y “NUM” son altamente frecuentes en ambas categorías. Sin embargo, las principales diferencias las encontramos en la aparición de palabras como “time” o “work” en la categoría negativa, tal vez debido al uso de expresiones como que el producto es una pérdida de tiempo o que sencillamente no funciona. Por otro lado, en la categoría positiva podemos apreciar un uso extendido de los tokens “great”, “good” y “promotion”, normalmente asociados a conceptos o sentimientos positivos. Cabe también recalcar que un dataset con más registros y sin tanto desbalance entre clases, tal vez la diferencia entre los tokens de las categorías fuese más evidente y el modelo clasificaría mejor los sentimientos (registros positivos: 4000 // registros negativos: 400)

**2. Muestre una visualización básica de los datos**

Negativo:

Un letrero de color blanco

Descripción generada automáticamente con confianza baja

Neutro:

Imagen que contiene Texto

Descripción generada automáticamente

Positivo:

Texto

Descripción generada automáticamente

Gráfico, Gráfico de barras

Descripción generada automáticamente

**3. Análisis de sentimientos:**

**1. Entrene uno o varios modelos para clasificar los comentarios en las siguientes categorías: positivos (rating 4 - 5), neutrales (rating 3) y negativos. Justifica qué métrica de evaluación has elegido para determinar el modelo que pondrías en producción**

Calendario

Descripción generada automáticamente

En base a lo analizado en el notebook, entre todos los modelos realizados me quedaría con el XGBoost con oversampling sobre los datos de training a las clases minoritarias (0 y 1).

Como métrica de evaluación escogería el f1-score, ya que es realmente útil para datos desbalanceados en problemas de clasificación, siendo este la media armónica de precision y recall, ponderando ambos valores en un solo estimador.

Antes de analizar los resultados, hay que contar con que el dataset no tiene los datos suficientes como para elaborar un modelo de clasificación basado en análisis de sentimiento debido a que solo presenta unos 4600 registros aproximadamente, sumado al hecho de que el desbalanceo de las clases es demasiado exagerado, quedando 4000 datos en la clase positiva y los apenas restantes repartidos entre las clases negativas y neutras.

Teniendo esto en cuenta, y aplicando oversampling así como la incorporación de características adicionales al texto, he conseguido un f1-score del 0,65 y 0,95 para la clase 0 y 2 respectivamente (negativo y positivo). De la clase mayoritaria era de esperar ya que contábamos con los suficientes registros, pero gracias al oversampling se ha podido llegar a un f1 decente para los pocos datos disponibles en esta clase. Sin embargo, y como era de esperar, la clase 1 (neutro) cuenta con menos registros que la negativa, por lo que el modelo le cuesta de por si identificar los comentarios, ya que además comparte similitudes en los tokens junto con la clase positiva (en la matriz de confusión se puede ver como la mayoría de los mal clasificados de esta clase se encuentran en la clase 2), llegando únicamente a un f1-score del 0,20.

En resumen, el modelo es capaz de clasificar de forma relativamente aceptable entre las clases 0 y 2, con una clara tendencia sobre la clase mayoritaria, pero muestra unas capacidades pobres hacia la clase neutral. Seguramente con un volumen de datos mayor y no tan desbalanceado, este modelo pueda ser mejorado y predecir correctamente las 3 clases.