Bootcamp Data Science - Ejercicio S21 - Gerardo Rodríguez

Objetivo: Profundizar en el entendimiento del análisis de sentimientos en Python.

1. **Identificación de Modelos de Análisis de Sentimientos**:

   - ¿Qué modelos o herramientas se utilizan en el notebook para realizar el análisis de sentimientos?

     El código que revisamos en clase utiliza dos modelos:

   - **SentimentIntensityAnalyzer** de NLTK: Es un modelo basado en reglas que utiliza un léxico preentrenado (VADER, Valence Aware Dictionary and sEntiment Reasoner) para analizar el sentimiento de los textos.

   - RoBERTa (Robustly optimized BERT approach): Un modelo de lenguaje basado en BERT (Bidirectional Encoder Representations from Transformers), optimizado para un mejor rendimiento en tareas de procesamiento de lenguaje natural.

   - Enuncia las diferencias en su uso.

   - SentimentIntensityAnalyzer (VADER):

   Basado en reglas.

   Utiliza un léxico preentrenado.

   Rápido y eficiente para textos cortos.

   Menos efectivo para textos largos y complejos.

   - RoBERTa:

   Basado en aprendizaje profundo.

   Utiliza técnicas avanzadas de procesamiento de lenguaje natural.

   Más preciso y efectivo para capturar matices complejos del lenguaje.

   Requiere más recursos computacionales.

2. **Interpretación de Resultados**:

    - Explica cómo se pueden interpretar los resultados “positivo”, “negativo” y “neutral” y qué significan en el contexto de las reseñas de Amazon.

- Cada observación es ponderada en porcentajes de acuerdo a la evaluación del algoritmo y dividido en positivo, negativo y neutral, la suma de los resultados suma uno (100%). Ejemplo:

  vader_pos: 0.85

  vader_neu: 0.10

  vader_neg: 0.05

- Positivo: Una puntuación alta en esta categoría indica que la reseña contiene palabras y frases que expresan satisfacción, alegría, aprobación o experiencias positivas.

- Negativo: Una puntuación alta en esta categoría indica que la reseña contiene palabras y frases que expresan insatisfacción, frustración, críticas o experiencias negativas.

- Neutral: Una puntuación alta en esta categoría indica que la reseña es equilibrada, con pocos elementos que expresan fuertes sentimientos positivos o negativos. Estas reseñas suelen ser objetivas o descriptivas sin un fuerte sesgo emocional.

Objetivo: Exploración de otros aspectos del análisis de sentimientos en Python y cómo transformarlos en conclusiones para compartir.

Se sube el código a Colab (Google Drive) y sería visible con el siguiente link:

https://colab.research.google.com/drive/12lCEqL1UpPKvGKgtytZ3SEAwTb-G1QnD?usp=sharing

3. **Modificación y Comparación de Código**:

    a) Modifica el código para obtener los sentimientos relacionados con las cámaras de los teléfonos móviles.

A1. Se agrega librería re para limpiar códigos HTML

```
import re
```

A2. Se agrega función para limpiar código HTML

```
# Función para limpiar los elementos HTML de una cadena de texto
def clean_html(text):
    clean = re.compile('<.*?>')
    return re.sub(clean, '', text)

# Aplicar la función de limpieza a la columna `Text`
try:
    df['Text'] = df['Text'].apply(clean_html)
    print("Limpieza de la columna 'Text' completada.")
except Exception as e:
    print(f"Error al limpiar la columna 'Text': {e}")
```

A3. Ampliar los registros para ampliar los comentarios con cámara de 5000 a 60,000

```
# Read in data
df = pd.read_csv('../input/amazon-fine-food-reviews/Reviews.csv')
print(df.shape)
df = df.head(60000)
print(df.shape)
```

A4. Filtrar los textos con comentarios de cámaras y/o celulares

```
# Filtrar comentarios que mencionan cámaras
camera_keywords = ['camera', 'cameras', 'photo', 'photos', 'picture',
'pictures','phone','cellphone' ]
camera_comments = df[df['Text'].str.contains('|'.join(camera_keywords),
case=False, na=False)]
```

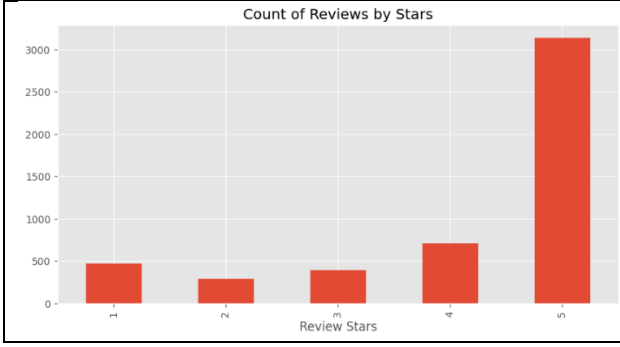
```
# Mostrar la cantidad de comentarios filtrados
print(f"Total de comentarios relacionados con cámaras:
{camera_comments.shape[0]}")

# Mostrar los primeros comentarios filtrados
camera_comments.head()
```

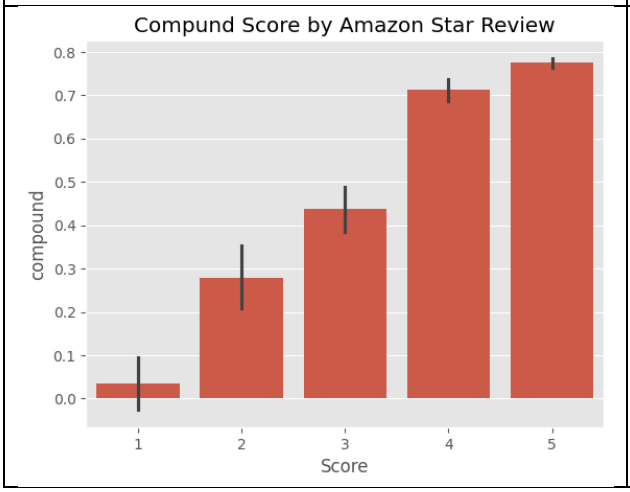
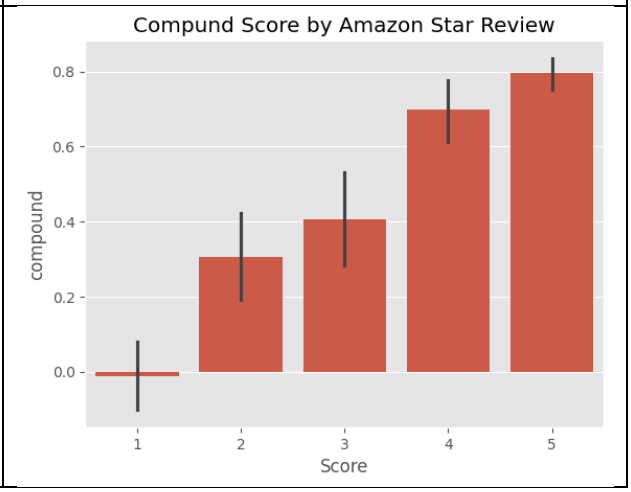
A5. Luego de este ajuste, reutilizaremos el código solo cambiando el dataframe “df” en todo el resto del código por “camera_comments” y volvemos a procesar todo con los nuevos ajustes.

b) ¿Qué tendencias se perciben en estos comentarios?

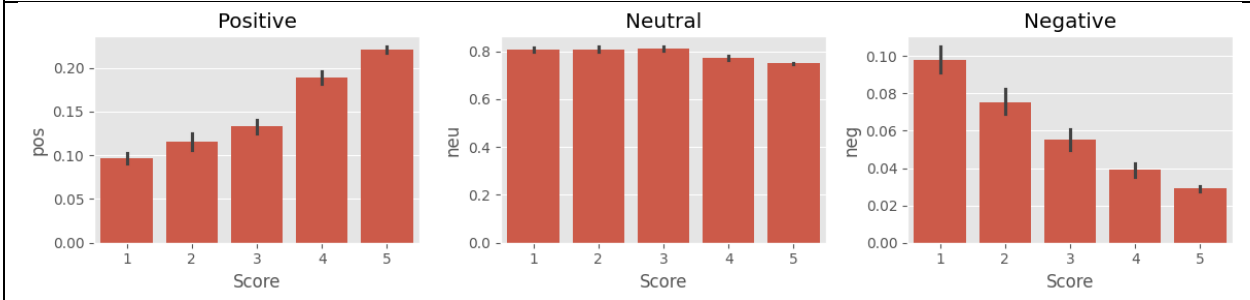
B1. El primer cambio interesante es la distribución de estrellas usando comentarios generales y usando comentarios con palabras relacionadas a cámaras y celulares. Ya que se balancean los comentarios negativos y positivos.

CON TODOS LOS DATOS	SOLO DATOS DE CAMARAS Y CELULARES
	

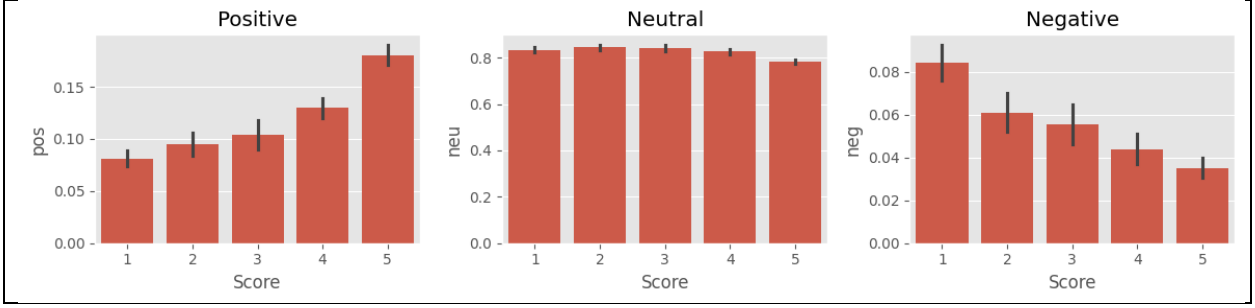
B2. Otro dato interesante es como aumenta el nivel de error, y es lógico al tener una menor muestra de datos, en este caso solo usando comentarios de cámaras y celulares el rango de error es más alto por la variabilidad de datos.

CON TODOS LOS DATOS	SOLO DATOS DE CAMARAS Y CELULARES
	

CON TODOS LOS DATOS



SOLO DATOS DE CAMARAS Y CELULARES



B3. Con el modelo de Roberta se balancean de manera muy diferentes el análisis de sentimientos, si bien la tendencia es la misma, los comentarios son un poco menos negativos.

CON TODOS LOS DATOS	SOLO DATOS DE CAMARAS Y CELULARES

```
[28] # Run for Roberta Model
encoded_text = tokenizer(example, return_tensors='pt')
output = model(**encoded_text)
scores = output[0][0].detach().numpy()
scores = softmax(scores)
scores_dict = {
    'roberta_neg' : scores[0],
    'roberta_neu' : scores[1],
    'roberta_pos' : scores[2]
}
print(scores_dict)

{'roberta_neg': 0.97635514, 'roberta_neu': 0.020687465, 'roberta_pos': 0.0029573692}
```

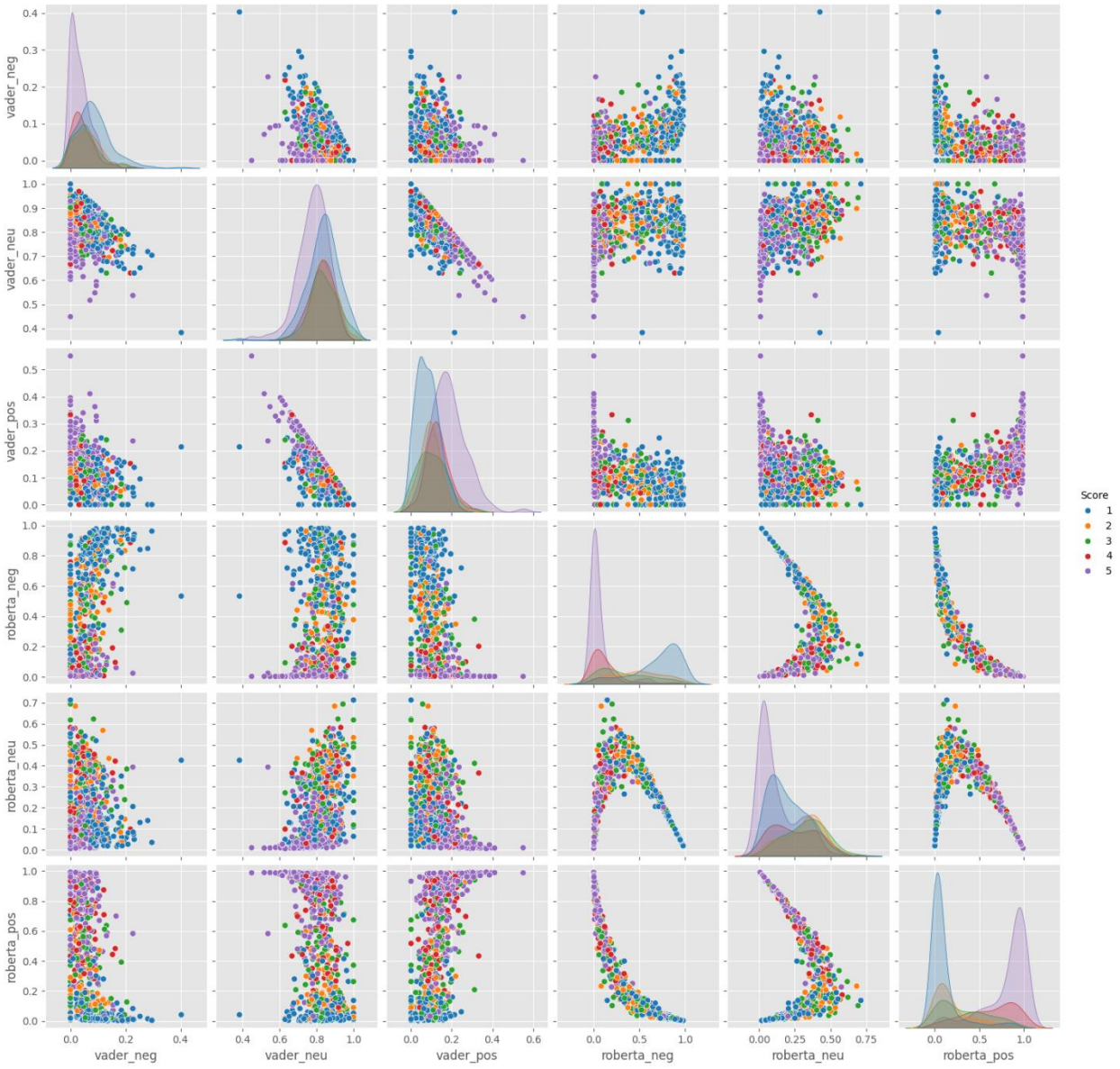
```
# Run for Roberta Model
encoded_text = tokenizer(example, return_tensors='pt')
output = model(**encoded_text)
scores = output[0][0].detach().numpy()
scores = softmax(scores)
scores_dict = {
    'roberta_neg' : scores[0],
    'roberta_neu' : scores[1],
    'roberta_pos' : scores[2]
}
print(scores_dict)

{'roberta_neg': 0.8697886, 'roberta_neu': 0.108836204, 'roberta_pos': 0.021375272}
```

B4. Los datos filtrados tienen una mayor representabilidad de puntajes negativos y neutrales, haciendo mucho más sensible la vinculación de comentarios con bajas estrellas y la relación con las calificaciones negativas de Roberta y Vader, lo mismo con comentarios positivos y estrellas altas.

Esto es relevante, porque este mismo gráfico en el ejercicio de todos los datos era predominantemente positivo.

En otras palabras con esta data el modelo parece ser más eficiente.

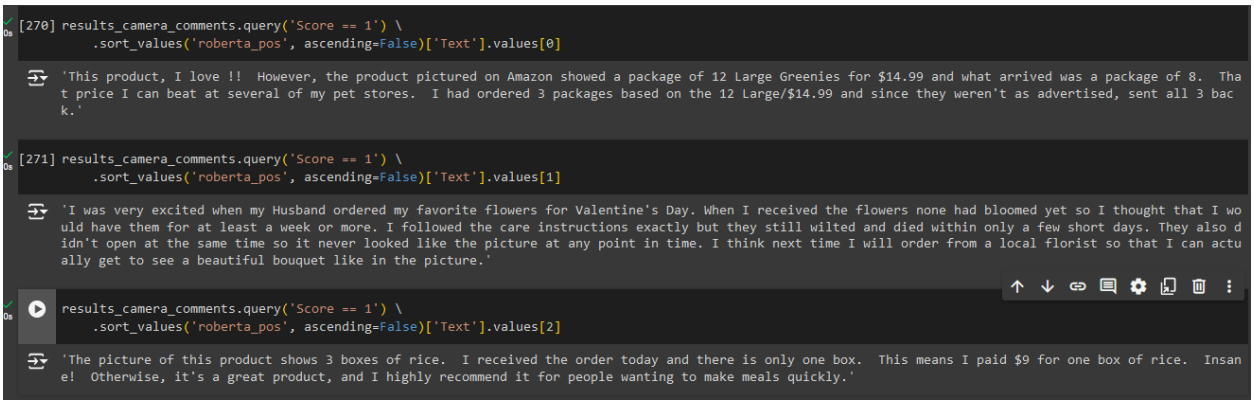

vader_neg
vader_neu
vader_pos
roberta_neg
roberta_neu
roberta_pos
Score
1
2
3
4
5

3. **Identificación de Comentarios Negativos y Propuestas de Mejora**:

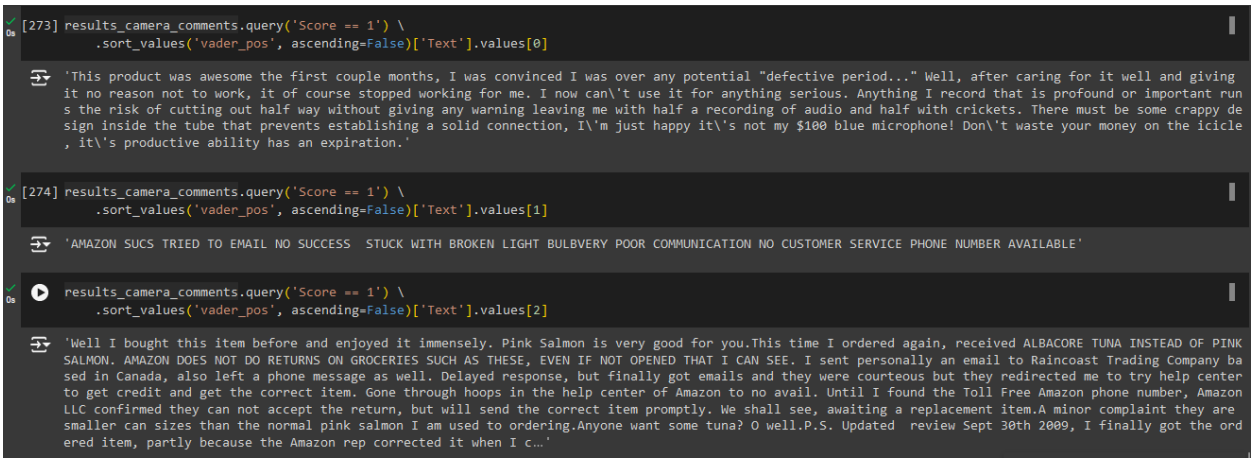
- Identifica los 3 comentarios más negativos utilizando el código original y propón ideas o estrategias sobre cómo un negocio podría utilizar esta información para mejorar sus productos o servicios.

a) 3 Comentarios más negativos

A1. ROBERTA



A2. VADER



Patrón:

Un sentir en común de los usuarios en estos comentarios es sobre las fotografías de los productos, estos se sienten defraudados al haber comprado por las imágenes que vieron en la página, y al final los diferentes productos no cumplieron lo que ofrecieron. En ocasiones recibieron menos producto o el producto no llego a verse como estaba especificado. Se identifica también de manera regular la falta de formas de contactar a Amazon o a sus proveedores.

Propuestas:

- Los proveedores deberían subir solo fotografías que cumplan a cabalidad con lo que el cliente puede obtener al pagar el producto, ya sea cantidad de producto o el resultado final de operar, armar, o procesar el mismo.

- Para los productos que el cliente debe trabajar se deben mandar instrucciones claras para que este pueda cuidar/armar/procesar el producto y obtener el resultado deseado.
- Los productos con riesgo de no alcanzar su máximo potencial, deberían tener clausulas del riesgo que tienen de dañarse para que el consumidor este prevenido y compra bajo su propio riesgo.
- Simplificar el proceso de contacto a personal de Amazon y sus proveedores sería un gran acierto, ya que también identificamos un patrón en los comentarios más negativos.