

Actividad Práctica Integradora

Procesamiento Natural del Lenguaje (NPL)

Actividad 1

Marce Martinez.

Contexto

La concesionaria AutoFlex, donde trabajamos como parte de una célula Data Science, está preocupada por la aceptación de la nueva versión de su marca estrella. Se han recibido muchos comentarios en redes sociales al respecto, y claramente se quiere optimizar el costo de procesamiento de esas opiniones. Para ello, desea interpretar esos comentarios utilizando el preprocesamiento de texto e implementar un modelo para resumir las opiniones de manera automática. El equipo de trabajo propone realizar un análisis de sentimientos que clasifique cada texto escrito, ya que la gerencia comercial tiene un conjunto de comentarios clasificados como "bueno", "malo" o "info" (necesita más información). Debemos realizar, primero, una descarga de la base, y terminar de sustentar metodológicamente el proyecto.

Consignas

1. En primer lugar, debemos sustentar metodológicamente:
 - a. ¿Qué tipo de aplicación es un análisis de sentimientos? Por favor, en un párrafo de no más 6 líneas, explique en qué consiste, de manera que la gerencia de analítica lo comprenda fácilmente.

Actividad Práctica Integradora

- b. ¿Qué tipo de procesamiento es necesario realizar primero? Exponga en un dibujo los pasos que va a realizar sobre los comentarios con el fin de convertirlos en data estructurada.
2. La gerencia de *marketing* le ha entregado una tabla con diferentes comentarios de los clientes, la cual se llama "comentarios.csv". Primeramente, realice una lectura de los datos. Para ello, use el método `read_csv` en vez de `read_table`; utilice como separador la coma (,). Indique cuántos registros tiene la tabla y cuántas columnas; visualice los 20 primeros registros.
 3. Realice un análisis exploratorio de esta data encontrando el porcentaje de tipo de comentarios que han sido clasificados como malos, buenos o información, aplique el código Python que considere necesario y exprese el resultado en una tabla.
 4. Diseñe un patrón de expresión regular para utilizarlo como tokenizador más adelante, que además de las palabras en idioma español, lea los emojis como 🥰❤️. También considere la expresión como ":" como un solo token, pero que excluya la puntuación punto ".", coma "," y punto y coma ";".
 5. Defina el conjunto X como los comentarios del *data frame*, y el *target*, y como la columna tipo.

Formato de entrega

Se debe entregar esta parte en formato *notebook* de Jupyter. Escriba el código de cada parte solicitada en una celda "code", y mantenga las salidas dentro del mismo documento. Sus comentarios y respuestas textuales las puede escribir en una celda tipo "markdown".

Actividad Práctica Integradora

Cómo cargar la actividad

¡Felicitaciones por llegar al final de esta actividad!

Para concluir tu entrega sigue estos pasos:

1. Haz clic en el botón “Enviar tareas” para comenzar el proceso de carga de tu archivo con las respuestas.
2. Adjunta el archivo que contiene tus respuestas a través de la interfaz proporcionada.
3. Una vez que hayas adjuntado el archivo, confirma tu entrega haciendo clic en el botón correspondiente.

Recuerda completar todos los puntos y tener cuenta las condiciones de entrega.

¡Éxitos!

1. ¿Qué tipo de aplicación es un análisis de sentimientos?

El análisis de sentimientos es una aplicación de procesamiento de lenguaje natural (PNL) que utiliza algoritmos para interpretar textos y determinar su tono emocional como positivo, negativo o neutral. Es ampliamente utilizado en diversas industrias para analizar comentarios de redes sociales, reseñas o encuestas, permitiendo tomar decisiones estratégicas basadas en opiniones colectivas.

2. ¿Qué tipo de procesamiento es necesario realizar primero?

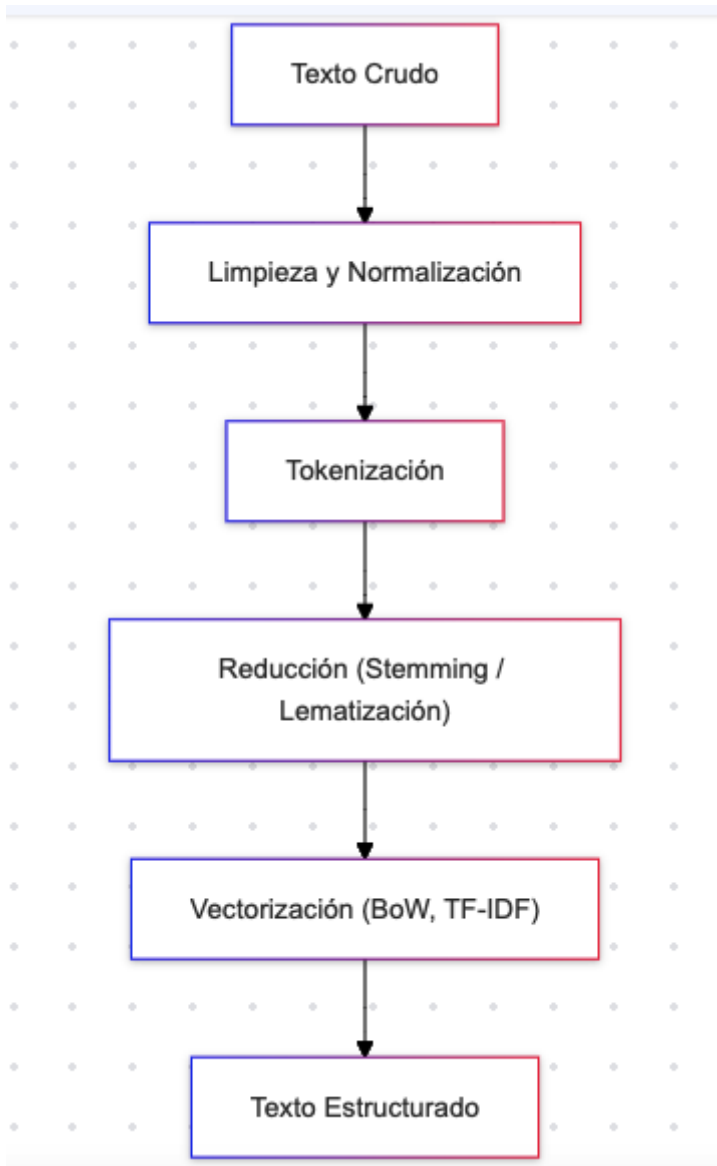
El procesamiento inicial consiste en transformar texto no estructurado en datos estructurados. Esto incluye:

1. **Limpieza y normalización:** Eliminar ruido (HTML, espacios innecesarios, etc.).
2. **Segmentación/tokenización:** Dividir texto en unidades significativas como palabras o frases.
3. **Reducción (stemming/lematización):** Simplificar palabras a su forma raíz o lema.

Actividad Práctica Integradora

4. Representación vectorial: Convertir texto en vectores (e.g., Bag of Words o TF-IDF).

Diagrama:



3. Lectura de datos en "comentarios.csv":

- Usar el método `read_csv` con separador coma (,).
- Contar registros y columnas.
- Mostrar los primeros 20 registros.

Actividad Práctica Integradora

Código en Python:

```
import pandas as pd

# Lectura del archivo

comentarios = pd.read_csv("comentarios.csv", sep=",")

# Cantidad de registros y columnas

num_registros, num_columnas = comentarios.shape

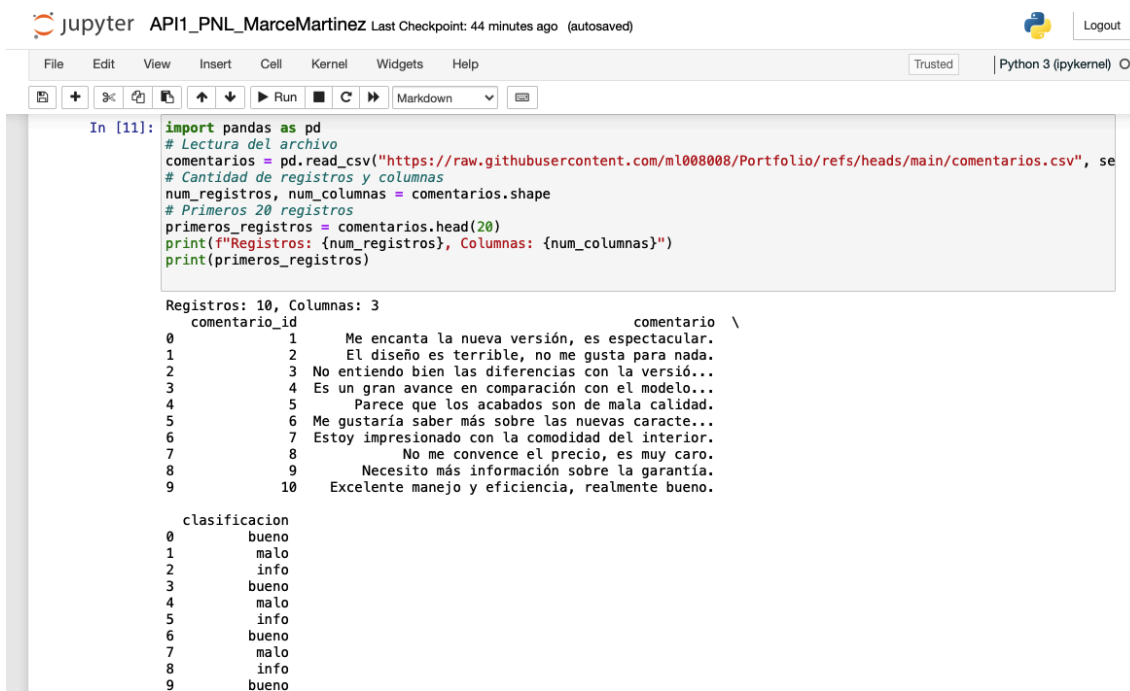
# Primeros 20 registros

primeros_registros = comentarios.head(20)

print(f"Registros: {num_registros}, Columnas: {num_columnas}")

print(primeros_registros)

#Favor de revisar codigo en archivo adjunto (Jupyter Notebook )
```



The screenshot shows a Jupyter Notebook interface. The top bar includes the Jupyter logo, the username 'API1_PNL_MarceMartinez', and the last checkpoint time 'Last Checkpoint: 44 minutes ago (autosaved)'. The interface has a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and markdown. The main area displays a code cell with the following Python code:

```
In [11]: import pandas as pd
# Lectura del archivo
comentarios = pd.read_csv("https://raw.githubusercontent.com/ml008008/Portfolio/refs/heads/main/comentarios.csv", se
# Cantidad de registros y columnas
num_registros, num_columnas = comentarios.shape
# Primeros 20 registros
primeros_registros = comentarios.head(20)
print(f"Registros: {num_registros}, Columnas: {num_columnas}")
print(primeros_registros)
```

The output of the code is displayed below the cell:

```
Registros: 10, Columnas: 3
comentario_id      comentario \
0      1  Me encanta la nueva versión, es espectacular.
1      2  El diseño es terrible, no me gusta para nada.
2      3  No entiendo bien las diferencias con la versió...
3      4  Es un gran avance en comparación con el modelo...
4      5  Parece que los acabados son de mala calidad.
5      6  Me gustaría saber más sobre las nuevas caracte...
6      7  Estoy impresionado con la comodidad del interior.
7      8  No me convence el precio, es muy caro.
8      9  Necesito más información sobre la garantía.
9     10  Excelente manejo y eficiencia, realmente bueno.

clasificacion
0      bueno
1      malo
2      info
3      bueno
4      malo
5      info
6      bueno
7      malo
8      info
9      bueno
```

4. Análisis exploratorio: Porcentaje de comentarios clasificados.

Crear una tabla que muestre los porcentajes de cada tipo de comentario (bueno, malo, info).

Código en Python

```
# Conteo de tipos de comentarios

conteo_tipos = comentarios['tipo'].value_counts()
```

Actividad Práctica Integradora

Calcular porcentajes

```
porcentajes = (conteo_tipos / num_registros) * 100
```

Crear una tabla resumen

```
tabla_resumen = porcentajes.reset_index()
```

```
tabla_resumen.columns = ['Tipo', 'Porcentaje']
```

```
print(tabla_resumen)
```

```
In [13]: # Conteo de tipos de comentarios
conteo_tipos = comentarios['clasificacion'].value_counts()
# Calcular porcentajes
porcentajes = (conteo_tipos / num_registros) * 100
# Crear una tabla resumen
tabla_resumen = porcentajes.reset_index()
tabla_resumen.columns = ['clasificacion', 'Porcentaje']
print(tabla_resumen)
```

	clasificacion	Porcentaje
0	bueno	40.0
1	malo	30.0
2	info	30.0

Importante:
Columna clasificacion = Tipo

5. Patrón de expresión regular para tokenización.

El patrón debe:

- Incluir palabras en español y emojis como 😊❤️.
- Tratar expresiones como ":" como un solo token.
- Excluir puntuaciones como . , ;.

Patrón en Regex:

```
r"[\wáéíóúñ]+|[:()]\s|[\p{Emoji}]"
```

```
In [17]: r"[\wáéíóúñ]+|[:()]\s|[\p{Emoji}]" r"[\wáéíóúñ]+|[:\]|[\u2764\u263a\u2765]"
```

```
Out[17]: ' [\wáéíóúñ]+|[:()]\s|[\p{Emoji}]" r"[\wáéíóúñ]+|[:\]|[\u2764\u263a\u2765]"
```

Importante: El código unicode hace referencia a los emojis.

6. Definición del conjunto X e Y. X: Comentarios del DataFrame (columna de texto). Y: Target o tipo de comentario.

Actividad Práctica Integradora

6. Definición del conjunto X e Y.

- **X:** Comentarios del DataFrame (columna de texto).
- **Y:** Target o tipo de comentario.

Código en Python:

X = comentarios['texto']

y = comentarios['tipo']

```
In [20]: X = comentarios['comentario']  
y = comentarios['clasificacion']
```

```
In [25]: print(X,y)  
  
0      Me encanta la nueva versión, es espectacular.  
1      El diseño es terrible, no me gusta para nada.  
2      No entiendo bien las diferencias con la versió...  
3      Es un gran avance en comparación con el modelo...  
4      Parece que los acabados son de mala calidad.  
5      Me gustaría saber más sobre las nuevas caracte...  
6      Estoy impresionado con la comodidad del interior.  
7      No me convence el precio, es muy caro.  
8      Necesito más información sobre la garantía.  
9      Excelente manejo y eficiencia, realmente bueno.  
Name: comentario, dtype: object 0      bueno  
1      malo  
2      info  
3      bueno  
4      malo  
5      info  
6      bueno  
7      malo  
8      info  
9      bueno  
Name: clasificacion, dtype: object
```

Gracias!

Actividad Práctica Integradora