

# Laboratorio 5

Presentado por:

**Nicolas Morales Galindo.**

**Juan Sebastian Sanchez.**

**Julian Cifuentes.**

**Javier Valderrama.**

## Objetivo del Laboratorio

Este laboratorio se desarrolla con la intención de fortalecer el aprendizaje con respecto a técnicas de análisis para sentimiento a partir de texto como también el uso de librerías convencionales de python.

## Introducción

En este laboratorio se aborda la librería `transformers`, la idea principal es realizar las pruebas vistas en clase a un conjunto de datos entregados por el profesor a cargo.

## Desarrollo

1. Se realiza la carga de librerías necesarias para el desarrollo de la actividad.

```
from google.colab import drive
drive.mount('/content/drive')
!pip install pysentimiento
import pandas as pd
from textblob import TextBlob
import nltk
nltk.download('punkt')
nltk.download('stopwords')
```

2. Se procede a realizar el importe de la data, desde google drive, posteriormente se obtiene la columna contenido de dicho archivo para realizar seguidamente la limpieza de la data, esto se consigue gracias a funciones que trabajan de la mano a expresiones regulares, lo cual buscará caracteres no convencionales y los retirará de la cadena de texto.

```

# Carga del archivo Excel
noticias = pd.read_excel('/content/drive/MyDrive/Seminario/Sentiments/Noticias2ver.xlsx')

# Extracción de la columna 'contenido'
contenido = noticias['contenido'].dropna() # Elimina filas sin datos

# Limpieza de texto
import re
from nltk.corpus import stopwords

def limpiar_texto(texto):
    texto = re.sub(r'http\S+|www\S+|https\S+', '', texto, flags=re.MULTILINE)
    texto = re.sub(r'\@w+|\#','', texto)
    texto = re.sub(r'^\w\s', '', texto)
    texto = " ".join(word for word in texto.split() if word not in stopwords.words('spanish'))
    return texto

# Aplicar la función de limpieza a cada texto
noticias['contenido_limpio'] = contenido.apply(limpiar_texto)
contenido_limpio = contenido.apply(limpiar_texto)

```

### 3. Procedemos a importar la librería Transformers.

```

from pysentimiento import create_analyzer
import transformers

transformers.logging.set_verbosity(transformers.logging.ERROR)

```

### 4. En este paso se procede a implementar la primera técnica, análisis de sentimiento , este análisis lo que ayuda a determinar que tanta contenido negativo, positivo y neutro maneja la data en análisis, posteriormente se procede a pintar una tabla la cual indica que proporciones se encontraron .

```

import pandas as pd
import matplotlib.pyplot as plt

analyzer = create_analyzer(task="sentiment", lang="es")
sentimientos = contenido_limpio.apply(lambda x: analyzer.predict(x).output)

# Convertir la Serie a DataFrame
df = pd.DataFrame(sentimientos)

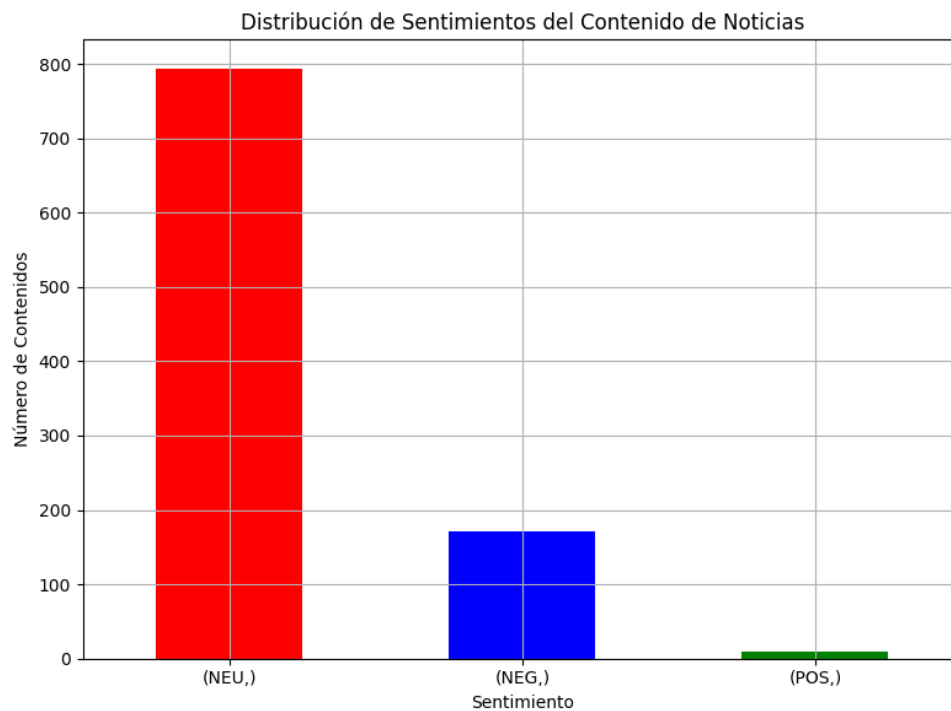
# Mostrar el DataFrame como tabla usando display()
display(df)

# Contar las frecuencias de cada categoría
conteo_sentimientos = df.value_counts()

# Preparar el gráfico de barras
plt.figure(figsize=(8, 6))
conteo_sentimientos.plot(kind='bar', color=['red', 'blue', 'green'])

# Configuraciones del gráfico
plt.title('Distribución de Sentimientos del Contenido de Noticias')
plt.xlabel('Sentimiento')
plt.ylabel('Número de Contenidos')
plt.xticks(rotation=0) # Rotar etiquetas del eje x si es necesario
plt.grid(True)
plt.tight_layout()

```



- En este paso se procede a implementar la segunda técnica, análisis de emociones, este análisis lo que ayuda a determinar es qué tanta proporción hay entre “disgusto” y “otros sentimientos”. En este paso se trabajó con solo 10 registros, posteriormente se procede a pintar la información en una tabla.

```
emotion_analyzer = create_analyzer(task="emotion", lang="en")
emociones = contenido_limpio.head(10).apply(lambda x: emotion_analyzer.predict(x).output)

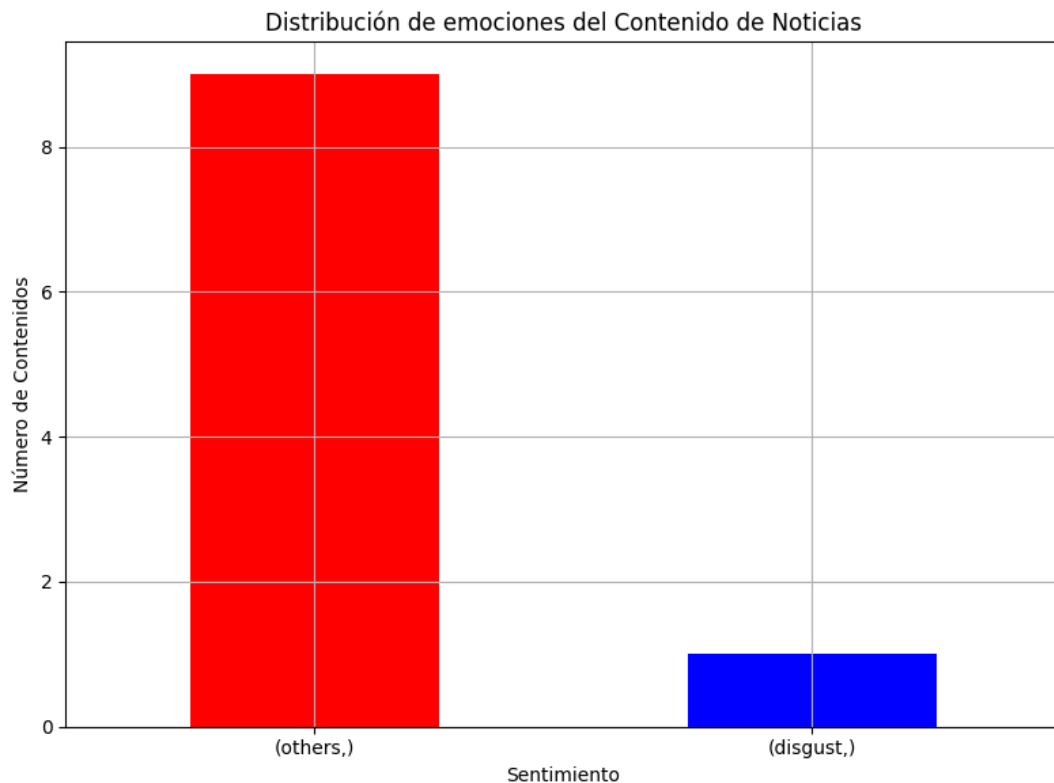
# Convertir la Serie a DataFrame
dfemocion = pd.DataFrame(emociones)

# Mostrar el DataFrame como tabla usando display()
display(dfemocion)

# Contar las frecuencias de cada categoría
conteo_emociones = dfemocion.value_counts()

# Preparar el gráfico de barras
plt.figure(figsize=(8, 6))
conteo_emociones.plot(kind='bar', color=['red', 'blue', 'green'])

# Configuraciones del gráfico
plt.title('Distribución de emociones del Contenido de Noticias')
plt.xlabel('Sentimiento')
plt.ylabel('Número de Contenidos')
plt.xticks(rotation=0) # Rotar etiquetas del eje x si es necesario
plt.grid(True)
plt.tight_layout()
```



- Para ser más claros organizamos los errores en una lista con la intención de reflejar de manera más detallada los resultados obtenidos en el paso 5

```
import pandas as pd
import matplotlib.pyplot as plt

# Suponiendo que `contenido_limpio` contiene tus datos y `emotion_analyzer` es tu analizador de emociones

# Obtener las probabilidades de emociones para los primeros 10 contenidos
emociones = contenido_limpio.head(10).apply(lambda x: emotion_analyzer.predict(x).probas)

# Convertir la Serie de emociones a DataFrame
dfemocion = pd.DataFrame(list(emociones))

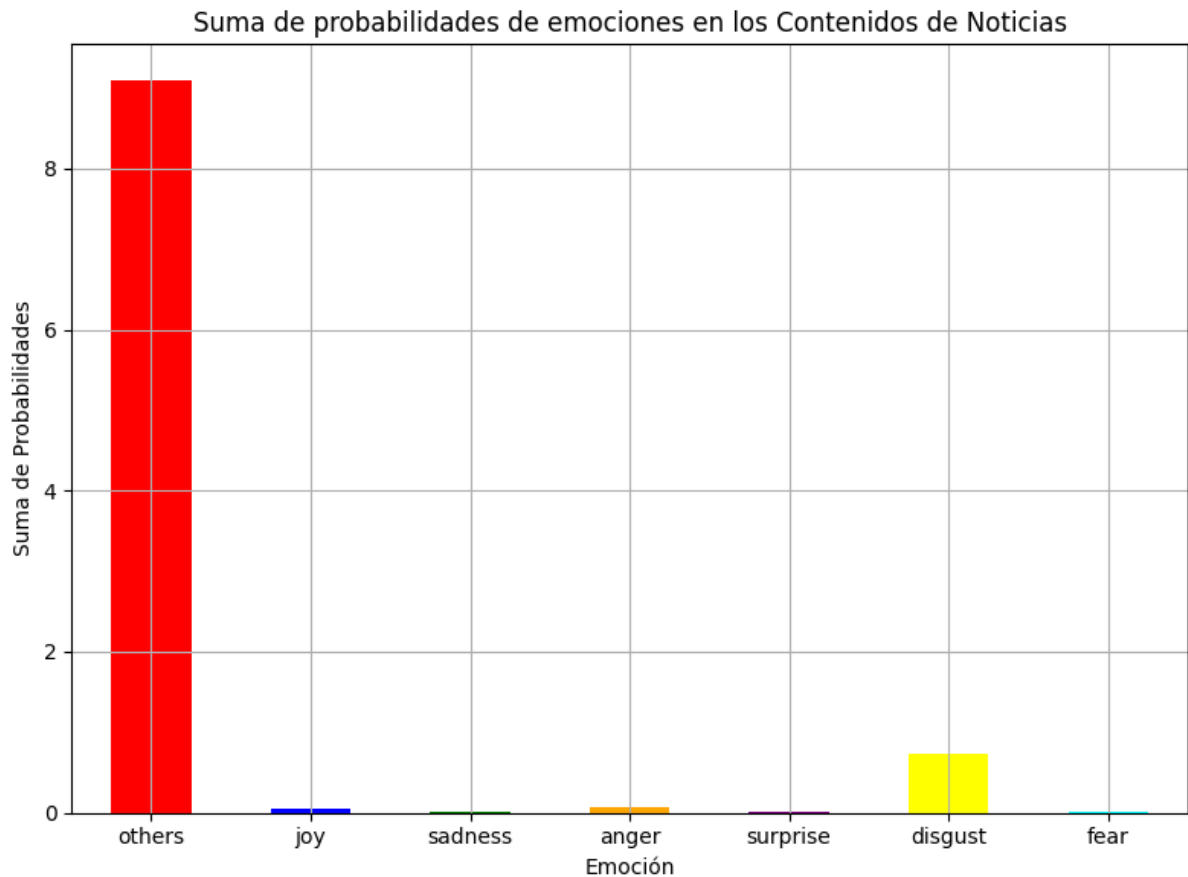
# Sumar las probabilidades de cada emoción para obtener la frecuencia total
suma_emociones = dfemocion.sum(axis=0)

# Preparar el gráfico de barras
plt.figure(figsize=(8, 6))
suma_emociones.plot(kind='bar', color=['red', 'blue', 'green', 'orange', 'purple', 'yellow', 'cyan'])

# Configuraciones del gráfico
plt.title('Suma de probabilidades de emociones en los Contenidos de Noticias')
plt.xlabel('Emoción')
plt.ylabel('Suma de Probabilidades')
plt.xticks(rotation=0) # Rotar etiquetas del eje x si es necesario
plt.grid(True)
plt.tight_layout()

# Mostrar el gráfico
plt.show()
```

- En este paso se procede a implementar la tercera técnica, análisis de odio, este análisis lo que ayuda es a identificar contenido textual que incita al odio ,violencia y



objetividad racial manejada en la data, posteriormente se procede a pintar una tabla la cual indica que proporciones porcentuales se encontraron en 10 registros del lote.

```
import pandas as pd
import matplotlib.pyplot as plt

# Suponiendo que `contenido_limpio` contiene tus datos y `hate_speech_analyzer` es tu analizador de hate speech

# Obtener las salidas y probabilidades de hate speech para los primeros 10 contenidos
hates = contenido_limpio.head(10).apply(lambda x: hate_speech_analyzer.predict(x))

# Inicializar un diccionario para acumular las probabilidades por categoría
acumulado_probab = {'hateful': 0.0, 'targeted': 0.0, 'aggressive': 0.0}

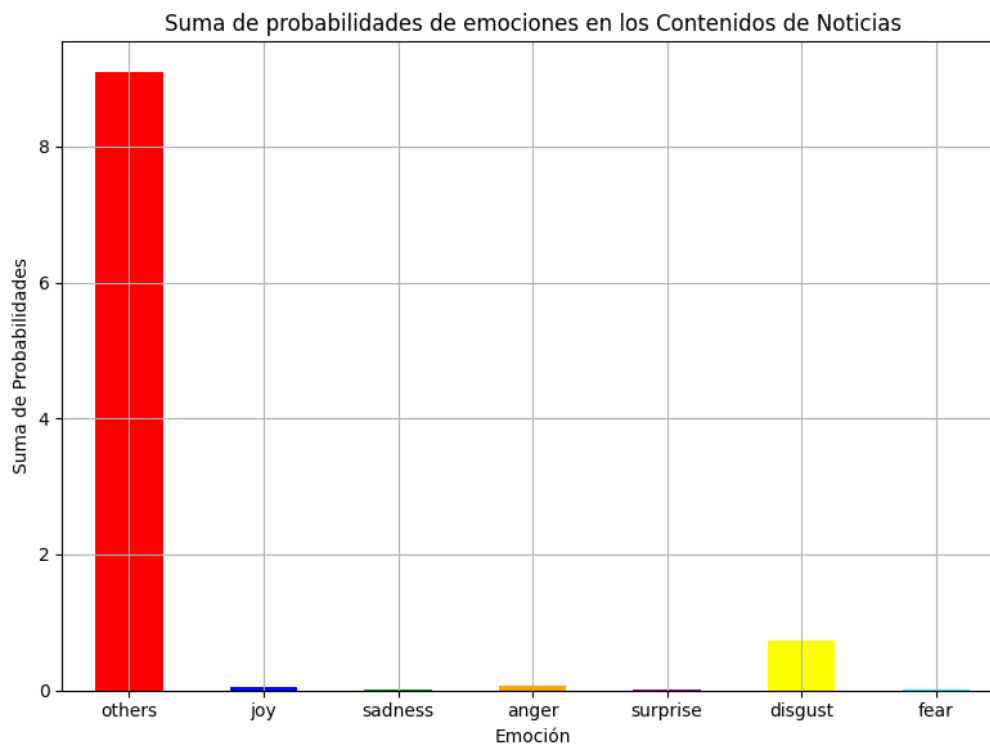
# Sumar las probabilidades por categoría para todos los contenidos
for hate in hates:
    for categoria, probabilidad in hate.probas.items():
        acumulado_probab[categoria] += probabilidad

# Convertir el diccionario a DataFrame para el gráfico de barras
df_acumulado_probab = pd.DataFrame(list(acumulado_probab.items()), columns=['Categoría', 'Suma de Probabilidades'])

# Preparar el gráfico de barras
plt.figure(figsize=(8, 6))
df_acumulado_probab.sort_values(by='Suma de Probabilidades', ascending=False).plot(x='Categoría', y='Suma de Probabilidades', kind='bar', color=['red', 'blue', 'green'])

# Configuraciones del gráfico
plt.title('Suma de Probabilidades por Categoría de Hate Speech en los Contenidos de Noticias')
plt.xlabel('Categoría de Hate Speech')
plt.ylabel('Suma de Probabilidades')
plt.xticks(rotation=0)
plt.grid(True)
plt.tight_layout()

# Mostrar el gráfico
plt.show()
```



## Conclusión

En la resolución de este laboratorio esto nos ayudo ha aplicar y consolidar conocimientos sobre técnicas avanzadas de procesamiento de texto y análisis de sentimiento utilizando la librería Transformers en Python. En el desarrollo de esta práctica, se implementaron tres técnicas principales: análisis de sentimiento, análisis de emociones y análisis de odio.

1. Análisis de Sentimiento: Se logró identificar la proporción de contenido negativo, positivo y neutro en el conjunto de datos, proporcionando una visión general sobre el tono del texto analizado.
2. Análisis de Emociones: Esta técnica permitió diferenciar entre sentimientos de disgusto y otros sentimientos, aunque se limitó a una muestra de 10 registros para una evaluación más manejable.
3. Análisis de Odio: Finalmente, se evaluó la presencia de contenido que incita al odio, violencia y objetividad racial, mostrando las proporciones encontradas en los datos.

Estos análisis fueron respaldados con gráficos y tablas que facilitarán la interpretación de los resultados. La práctica no solo nos ayudará el manejo de las herramientas de procesamiento de texto, sino que también se destaca la importancia de técnicas de limpieza de datos y el uso de expresiones regulares para preparar el texto para su análisis.