Actividad práctica 5.2: Análisis de Datos Textuales

Vamos a adaptar el ejercicio de análisis de datos textuales utilizando un conjunto de datos sobre **álbumes de todos los tiempos**, donde el campo de interés será **"Average Rating"**. Vamos a realizar un análisis de sentimiento sobre las reseñas de los álbums y visualizar los resultados. A continuación, te explico cómo hacerlo paso a paso.

Paso 1: Instalación de librerías necesarias

Antes de comenzar, asegúrate de tener las librerías necesarias instaladas:

```
pip install nltk pandas matplotlib seaborn
```

Paso 2: Importar las librerías necesarias

Comenzamos importando las librerías necesarias:

```
import nltk
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from nltk.sentiment import SentimentIntensityAnalyzer
```

Paso 3: Cargar el conjunto de datos

Suponemos que tenemos un conjunto de datos sobre **álbumes musicales** con las siguientes columnas relevantes:

- `album_name`: Nombre del álbum
- `artist`: Artista del álbum
- `review`: Reseña del álbum (texto)
- `average_rating`: Puntuación promedio del álbum (número)

Cargamos el conjunto de datos usando pandas:

```
# Cargar el conjunto de datos
df = pd.read_csv('albums_all_time.csv')

# Ver las primeras filas del dataset
df.head()
```

Paso 4: Preprocesar los datos

4.1: Tokenización

Primero, tokenizamos las reseñas de los álbumes. La tokenización es el proceso de dividir el texto en palabras individuales o "tokens". En este caso, dividimos la columna `review` en palabras:

```
# Descargar los recursos de NLTK necesarios
nltk.download('punkt')

# Tokenizar las reseñas
df['tokens'] = df['review'].apply(lambda x: word_tokenize(x.lower()))  # Convertir todo a minúsculas
```

4.2: Eliminación de stopwords

Eliminamos las *stopwords* (palabras vacías) que no aportan valor en el análisis, como "el", "la", "y", etc.

```
# Descargar el conjunto de stopwords de NLTK
nltk.download('stopwords')
stop_words = set(stopwords.words('spanish'))  # Usamos el idioma español si las reseñas están en ese idioma

# Eliminar las stopwords de los tokens
df['tokens_sin_stopwords'] = df['tokens'].apply(lambda x: [word for word in x if word not in stop_words])
```

4.3: Lematización

Aplicamos lematización para reducir las palabras a su forma base. Por ejemplo, "correré" se convierte en "correr".

```
# Descargar los recursos necesarios para lematización
nltk.download('wordnet')
lemmatizer = WordNetLemmatizer()

# Lematizar las palabras
df['tokens_lemmatizados'] = df['tokens_sin_stopwords'].apply(lambda x:
[lemmatizer.lemmatize(word) for word in x])
```

Paso 5: Análisis de sentimiento

Realizamos el análisis de sentimiento sobre las reseñas utilizando el **SentimentIntensityAnalyzer** de NLTK, que nos da una puntuación de sentimiento de cada reseña. Este valor va de -1 (sentimiento negativo) a 1 (sentimiento positivo).

```
# Inicializar el analizador de sentimiento
sia = SentimentIntensityAnalyzer()

# Calcular el puntaje de sentimiento para cada reseña
df['sentimiento'] = df['review'].apply(lambda x:
sia.polarity_scores(x)['compound'])  # 'compound' es un valor entre -1 y 1
```

Paso 6: Relación entre sentimiento y puntuación promedio

A continuación, vamos a visualizar cómo el sentimiento de las reseñas se relaciona con la puntuación promedio de cada álbum.

6.1: Gráfico de dispersión de sentimiento vs. puntuación promedio

Vamos a crear un gráfico de dispersión para ver si existe alguna relación entre el sentimiento de las reseñas y la puntuación promedio de los álbumes.

```
# Gráfico de dispersión: Sentimiento vs. Puntuación promedio
plt.figure(figsize=(10, 6))
sns.scatterplot(x='average_rating', y='sentimiento', data=df, color='green')
plt.title('Sentimiento vs. Puntuación Promedio de Álbumes')
plt.xlabel('Puntuación Promedio')
plt.ylabel('Sentimiento')
plt.show()
```

6.2: Histograma de Sentimientos

Podemos crear un histograma para ver la distribución de los sentimientos de las reseñas.

```
# Crear un histograma de la distribución de sentimientos
plt.figure(figsize=(8, 6))
sns.histplot(df['sentimiento'], kde=True, color='blue', bins=30)
plt.title('Distribución de Sentimientos en las Reseñas')
plt.xlabel('Sentimiento')
plt.ylabel('Frecuencia')
plt.show()
```

Paso 7: Clasificación de Sentimientos

Clasificamos las reseñas en positivas, negativas y neutrales según el puntaje de sentimiento. Utilizamos un umbral para determinar estas categorías.

```
# Clasificar las reseñas según el sentimiento
df['clasificacion_sentimiento'] = df['sentimiento'].apply(lambda x: 'Positivo' if x
> 0 else ('Negativo' if x < 0 else 'Neutral'))

# Ver las frecuencias de cada clase
df['clasificacion_sentimiento'].value_counts()
```

7.1: Visualización de la clasificación de sentimientos

Creamos un gráfico para ver la cantidad de reseñas positivas, negativas y neutrales.

```
# Visualizar las clasificaciones de sentimiento
plt.figure(figsize=(8, 6))
sns.countplot(x='clasificacion_sentimiento', data=df, palette='coolwarm')
plt.title('Clasificación de Sentimientos de las Reseñas')
plt.xlabel('Clasificación de Sentimiento')
plt.ylabel('Frecuencia')
plt.show()
```

Paso 8: Conclusiones

- **Distribución de sentimientos**: El histograma nos muestra si las reseñas son mayormente positivas, negativas o neutrales. Si la mayoría de las reseñas son positivas, podríamos investigar qué características del álbum generan ese sentimiento.
- **Relación entre sentimiento y puntuación**: El gráfico de dispersión nos ayuda a ver si hay alguna correlación entre la puntuación promedio y el sentimiento de las reseñas. Por ejemplo, los álbumes con puntuaciones altas pueden tener sentimientos más positivos.
- **Clasificación de sentimiento**: Al clasificar las reseñas como positivas, negativas o neutrales, podemos analizar las reseñas para cada categoría y detectar tendencias o patrones en los comentarios.

Resumen

En este ejercicio, hemos realizado un análisis de sentimientos sobre las reseñas de álbumes utilizando el paquete **nltk** para procesamiento de texto (tokenización, eliminación de stopwords y lematización) y el **SentimentIntensityAnalyzer** para calcular los puntajes de sentimiento. Finalmente, visualizamos los resultados para comprender mejor el sentimiento detrás de las reseñas y su relación con la puntuación promedio de cada álbum.