Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Integrantes:

José Jesús Ramírez Cruz 202052478

Gabriel Reyes Leal 202053516

Alfonso Saldaña Campos 202056307

Periodo: Primavera 2024

Materia: Minería de Datos

Reporte: Predicción objetiva de las elecciones México 2024

Nombre del Docente: María Beatriz Bernabé Loranca

ÍNDICE

RESUMEN

Este proyecto se centra en el análisis de sentimientos y la extracción de palabras comunes de tweets mediante técnicas de procesamiento de lenguaje natural (NLP). Se emplean varias herramientas y bibliotecas de Python, incluyendo `nltk` para el preprocesamiento del texto y `scikit-learn` para el entrenamiento de un modelo de clasificación de sentimientos.

El proceso comienza con la lectura de tweets desde un archivo de Excel y su posterior preprocesamiento para eliminar elementos no deseados como URL, menciones y caracteres especiales, además de convertir el texto a minúsculas y lematizar las palabras. Posteriormente, los tweets se clasifican en sentimientos positivos, negativos o neutrales utilizando un diccionario predefinido y un modelo de Naive Bayes entrenado con datos etiquetados manualmente.

Además del análisis de sentimientos, se realiza una limpieza de texto para extraer y analizar las palabras más comunes en los tweets, lo cual permite identificar tendencias y patrones en el lenguaje utilizado. Los resultados se visualizan mediante gráficos y se documentan en archivos CSV para su posterior análisis.

Este reporte describe en detalle las metodologías utilizadas, las etapas de preprocesamiento y modelado, y los pasos seguidos para la visualización y análisis de los resultados. El objetivo principal es proporcionar una visión comprensiva del flujo de trabajo y las técnicas aplicadas en el análisis de sentimientos y la extracción de palabras en los tweets.

INTRODUCCIÓN

A lo largo de las sesiones que hemos tenido, referentes a la asignatura de Minería de Datos, hemos tenido la oportunidad de abordar y discutir ciertos trabajos en los que nuestra docente, la profesora María Beatriz Bernabé Loranca, ha tenido participación o incluso ha desarrollado de manera individual diferentes proyectos y trabajos que van de la mano con la ya mencionada asignatura.

En base a ello, el siguiente trabajo tiene la intención de poder hallar una predicción objetiva para las principales candidatas a la presidencia de nuestro país, México. Este será un acontecimiento de enorme peso para todos los ciudadanos que habitamos en él, por lo que con nuestro trabajo, esperamos dar perspectiva a cualquiera que desee leer este artículo, no solo por el mero interés político del país, sino que, de igual manera se busca que se comprenda el método que se utilizó para poder llegar a ciertas conclusiones y tener una auténtica fuente de confianza para que llegado el momento, se demuestre la transparencia del proceso democrático y quede de cierta manera en evidencia la coherencia y relación que tendrá la opinión pública con los resultados finales de las elecciones.

ANTECEDENTES

Este artículo fue posible en gran medida gracias al material proporcionado por nuestra docente ya antes mencionada, ya que por ello fue posible hacernos a la idea de cómo llevar a cabo nuestros análisis y estructuración de los mismos.

Tal como se ha discutido en nuestras sesiones de la asignatura, hemos buscado la mejor manera de conseguir los datos más relevantes para el presente trabajo, haciendo uso de herramientas mencionadas durante estas.

En sesiones anteriores se ha mencionado el análisis de sentimientos como método u opción principal para este trabajo, puesto que posiblemente es el mejor camino para llegar a lo que buscamos, la predicción objetiva, por lo que fue algo que decidimos rescatar de aquellas discusiones, ya que se nos fue mostrado en otros trabajos de la misma índole.

OBJETIVO

El proyecto tiene como objetivo proporcionar una herramienta automatizada para analizar el sentimiento de los tweets, lo cual para nuestro caso en particular, nos será de mucha utilidad al momento de querer hacer una predicción objetiva de nuestras elecciones presidenciales en México para este 2024.

Con este reporte se busca clarificar el procedimiento que se llevo a cabo al momento de realizar nuestro código.

HERRAMIENTAS Y TÉCNICAS

En este proyecto, se emplean diversas herramientas y técnicas para llevar a cabo el análisis de sentimientos de los tweets recolectados. Entre ellas se incluyen:

- NLTK (Natural Language Toolkit): Una biblioteca de Python ampliamente utilizada en PLN que proporciona herramientas para tokenización, lematización y análisis de texto.
- Transformers de Hugging Face: Se emplea el tokenizador BERT pre-entrenado para el español proporcionado por Hugging Face, el cual permite tokenizar los tweets y prepararlos para el análisis de sentimientos.
- BERT: La tokenización en BERT, a través del uso de WordPiece, es una parte crítica de su capacidad para entender el lenguaje de manera eficiente y efectiva. Facilita el manejo de un amplio vocabulario, incluidas palabras nuevas o raras, sin comprometer significativamente la eficiencia computacional o la capacidad de generalización del modelo. Este enfoque de tokenización, combinado con la

arquitectura de modelado de atención bidireccional de BERT, permite una comprensión profunda del contexto de las palabras y mejora significativamente el rendimiento en una amplia gama de tareas de procesamiento de lenguaje natural.

- Openpyxl: Para leer y escribir archivos de Excel.
- Sklearn.feature_extraction.text: Para vectorizar texto (CountVectorizer).
- Sklearn.naive_bayes: Para el clasificador Naive Bayes.
    - Clasificación de texto: Sklearn.naive_bayes es un algoritmo de aprendizaje automático que te permite clasificar texto en diferentes categorías. En este caso, lo utilizaste para clasificar tweets como positivos, negativos o neutrales.
    - Probabilidades de clasificación: El algoritmo no solo te dice si un tweet es positivo, negativo o neutral, sino que también te da la probabilidad de que pertenezca a cada categoría. Esto te permite tener una idea más clara de la confianza que puedes tener en la clasificación.
    - Facilidad de uso: Sklearn.naive_bayes es un algoritmo relativamente simple y fácil de usar, lo que lo hace una buena opción para principiantes en el aprendizaje automático.
    - Eficiencia: El algoritmo es computacionalmente eficiente, lo que significa que puede clasificar grandes cantidades de texto rápidamente.
- Joblib:
    - Guardar modelos: Joblib es una biblioteca de Python que te permite guardar modelos de aprendizaje automático entrenados, como el modelo Naive Bayes que utilizaste para clasificar los tweets. Esto te permite reutilizar el modelo en el futuro sin tener que entrenarlo nuevamente.
    - Cargar modelos: Cuando necesites usar el modelo entrenado para clasificar nuevos tweets, Joblib te permite cargarlo fácilmente desde el archivo donde lo guardaste. Esto te ahorra tiempo y esfuerzo, especialmente si el modelo es complejo y requiere mucho tiempo para entrenarse.
    - Guardar vectorizadores: Joblib también te permite guardar los vectorizadores que utilizaste para preparar el texto para el modelo Naive Bayes. Esto es importante porque el vectorizador es necesario para convertir el texto nuevo en el formato que el modelo puede entender.
    - Cargar vectorizadores: Al igual que con los modelos, Joblib te permite cargar los vectorizadores guardados cuando necesites clasificar nuevos tweets. Esto te evita tener que crear un nuevo vectorizador cada vez.
- Matplotlib.pyplot: Para crear gráficos de barras.
- Numpy: Para operaciones matemáticas (cálculo de porcentajes).

METODOLOGÍA

Iniciamos creando un diccionario pequeño de manera manual, se tomó la decisión de hacerlo manualmente puesto que servirá de base para el aprendizaje de nuestro algoritmo, de otra manera no tendría punto de referencia o de partida.

-*Minar Datos*

Lo primero que necesitamos para empezar con el proceso de minar datos es identificar los tipos de datos que queremos minar con esto nos referimos a como de donde va enfocado nuestro problema, en este caso específico va hacia las elecciones presidenciales 2024 identificando esto nosotros decidimos minar datos de 2 videos de entrevistas de mismo canal echo por reporte López Doriga.

Para minar los datos usamos la herramienta octoparse.

Usando la opcion de autonos crea una funcion para que nuestro bot pueda minar datos despues exportamos los datos con la estructura que nosotros queramos par a

Despues eslegir los datos de vitalimportancia para nuestro analisis.

Teniendo este ya establecido, pudimos avanzar con la lectura del archivo denominado “TOKENS Y ETIQUETADO_ELECCIONES PRESIDENCIALES 2024” proporcionado por la docente, tras terminarlo pudimos determinar y entender el objetivo principal de este, enseñarnos cómo ampliar un diccionario limitado de palabras con etiquetas de sentimiento ("positivo", "negativo", "neutral") utilizando técnicas de procesamiento del lenguaje natural (PLN) y aprendizaje automático.

El proyecto se compone de las siguientes funcionalidades y componentes:

-Preprocesamiento de Texto:

Esta función toma un texto de entrada, elimina enlaces y menciones de redes sociales, y elimina caracteres que no sean letras. Luego, convierte el texto a minúsculas, tokeniza el texto en palabras, elimina las palabras de parada (stopwords) en español, y

finalmente lematiza las palabras (es decir, las convierte a su forma base). El texto preprocesado se devuelve como una cadena.

-Obtención de etiquetas de sentimientos:

Esta función analiza un texto y cuenta cuántas palabras en el texto están en las listas de palabras muy positivas y muy negativas. Si hay más palabras positivas, devuelve "bueno", si hay más palabras negativas, devuelve "malo", y si están iguales, devuelve "neutral". Esta función sirve como un análisis básico de sentimiento utilizando el diccionario principal.

-Calcular porcentajes de sentimientos:

Esta función toma una lista de tweets, los preprocesa usando la función definida previamente, y luego aplica la función de etiquetado de sentimiento para obtener las etiquetas de sentimiento de los tweets.

-Visualizar comparación de sentimientos:

Dado un conjunto de porcentajes de sentimientos, esta función visualiza una comparación de los sentimientos de dos personas (o dos conjuntos de tweets) utilizando un gráfico de barras. Además, crea una tabla para mostrar los totales de tweets y el número absoluto de tweets positivos, negativos y neutrales.

-Entrenar modelo Naive Bayes:

Esta función toma una lista de tweets y etiquetas, vectoriza las características de los tweets utilizando CountVectorizer, y entrena un modelo Naive Bayes con los datos. Guarda el modelo entrenado y el vectorizador en archivos para uso posterior.

FRAGMENTOS RELEVANTES DEL CÓDIGO

Leer Tweets desde un Archivo de Excel

```
import pandas as pd

def read_tweets_from_excel(file_path):
    df = pd.read_excel(file_path)
    tweets = df['Tweets'].dropna().tolist()
    return tweets
```

Esta función lee los tweets desde un archivo de Excel y los retorna en una lista.

Preprocesar el Texto

```
import re
import unidecode
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

stop_words = set(stopwords.words('spanish'))
lemmatizer = WordNetLemmatizer()

def preprocess_text(text):
    text = unidecode.unidecode(text)
    text = re.sub(r'http\S+', '', text)
    text = re.sub(r'@\w+', '', text)
    text = re.sub(r'[^a-zA-ZáéíóúÁÉÍÓÚñÑ\s]', '', text)
    text = text.lower()
    tokens = text.split()
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_wo
    return ' '.join(tokens)
```

Esta función limpia y normaliza el texto de los tweets, eliminando URL, menciones, caracteres especiales y palabras vacías, y aplica lematización.

Etiquetar Sentimientos

```
diccionario_principal = {
    "positivo": ["bueno", "excelente", "positivo", "feliz"],
    "negativo": ["malo", "terrible", "negativo", "triste"]
}

def get_sentiment_labels(text):
    words = text.split()
    positive_count = sum(1 for word in words if word in diccionario_principal["pos
    negative_count = sum(1 for word in words if word in diccionario_principal["neg

    if positive_count > negative_count:
        return "positivo"
    elif negative_count > positive_count:
        return "negativo"
    else:
        return "neutral"
```

Esta función determina el sentimiento de un tweet basado en la presencia de palabras positivas o negativas de un diccionario predefinido.

Entrenar el Modelo Naive Bayes

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline
import joblib

def train_naive_bayes_model(X, y):
    vectorizer = CountVectorizer()
    model = MultinomialNB()
    pipeline = make_pipeline(vectorizer, model)
    pipeline.fit(X, y)
    joblib.dump(pipeline, 'naive_bayes_model.pkl')
    joblib.dump(vectorizer, 'count_vectorizer.pkl')
```

Esta función entrena un modelo Naive Bayes utilizando un pipeline que incluye un vectorizador de conteo y guarda el modelo entrenado.

Limpieza de Texto para Análisis de Palabras

```
def clean_text(text):
    text = unidecode.unidecode(text)
    text = re.sub(r'http\S+', '', text)
    text = re.sub(r'@\w+', '', text)
    text = re.sub(r'[^a-zA-ZáéíóúÁÉÍÓÚñÑ\s]', '', text)
    text = text.lower()
    return text
```

Esta función limpia el texto eliminando URL, menciones y caracteres especiales, y convirtiéndolo a minúsculas.

Extraer Palabras de un Archivo de Excel

```
def extract_words_from_excel(input_file):
    df = pd.read_excel(input_file)
    words = []
    for tweet in df['Tweets']:
        cleaned_text = clean_text(str(tweet))
        words.extend(cleaned_text.split())
    return words
```

Esta función extrae y limpia las palabras de los tweets de un archivo de Excel, retornando una lista de palabras.

Escribir Palabras Comunes a CSV

```
import csv
from collections import Counter

def write_top_words_to_csv(words, output_file):
    counter = Counter(words)
    most_common_words = counter.most_common()
    with open(output_file, 'w', newline='', encoding='utf-8') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(['Word', 'Frequency'])
        for word, frequency in most_common_words:
            writer.writerow([word, frequency])
```

Esta función cuenta la frecuencia de las palabras en una lista y escribe las más comunes a un archivo CSV.

RESULTADOS

Dado que el presente reporte aún se encuentra en desarrollo, de momento documentaremos los resultados que se han obtenido hasta la fecha del presente dia 25 de abril de 2024, el presente código ha logrado usar las herramientas ya antes mencionadas para poder tokenizar, etiquetar y comenzar a entrenar nuestro modelo para la predicción y análisis de sentimientos de los respectivos tweets.

Después de definir estas funciones, el código carga tweets de dos archivos de Excel y los procesa para obtener las etiquetas de sentimiento. Calcula y visualiza los

porcentajes de sentimientos para comparar los sentimientos hacia dos personas, en este caso de Claudia y Xochitl.

Luego, el código entrena un modelo Naive Bayes utilizando los tweets y etiquetas de sentimiento y guarda el modelo y el vectorizador para su uso posterior.

Por último, el script carga los archivos de Excel de otro tiempo, carga el modelo y vectorizador guardados, predice los sentimientos de los tweets de ese tiempo utilizando el modelo y el vectorizador, y calcula y visualiza los porcentajes de sentimientos para comparar los sentimientos de las dos personas.

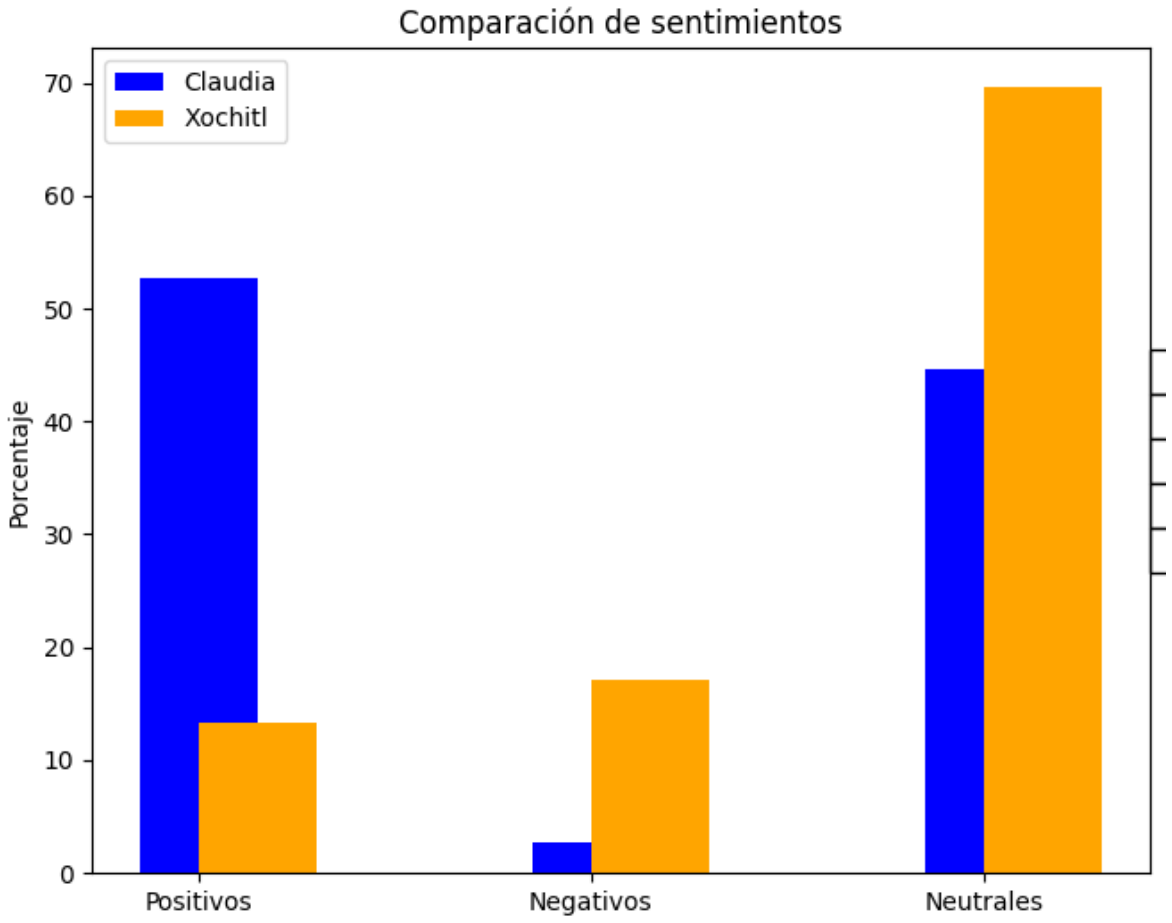
En resumen, el código realiza un análisis de sentimientos de tweets, primero utilizando palabras clave para un análisis básico, y luego utiliza un modelo Naive Bayes para un análisis más avanzado y comparativo de sentimientos entre diferentes conjuntos de tweets.

Actualización tras el último análisis ya incluyendo los tiempos 1 y 2:

Se obtuvieron primeramente ciertos resultados mas inclinados hacia lo positivo para Claudia, mientras que para Xóchitl demostraban mas inclinación a lo negativo y lo neutral, desgraciadamente se sospecha que se hizo bastante uso de bots para muchos de los comentarios positivos hacia Xóchitl, esto en materia del primer análisis pre-debate que se dio antes de la fecha 7 de abril de 2024.

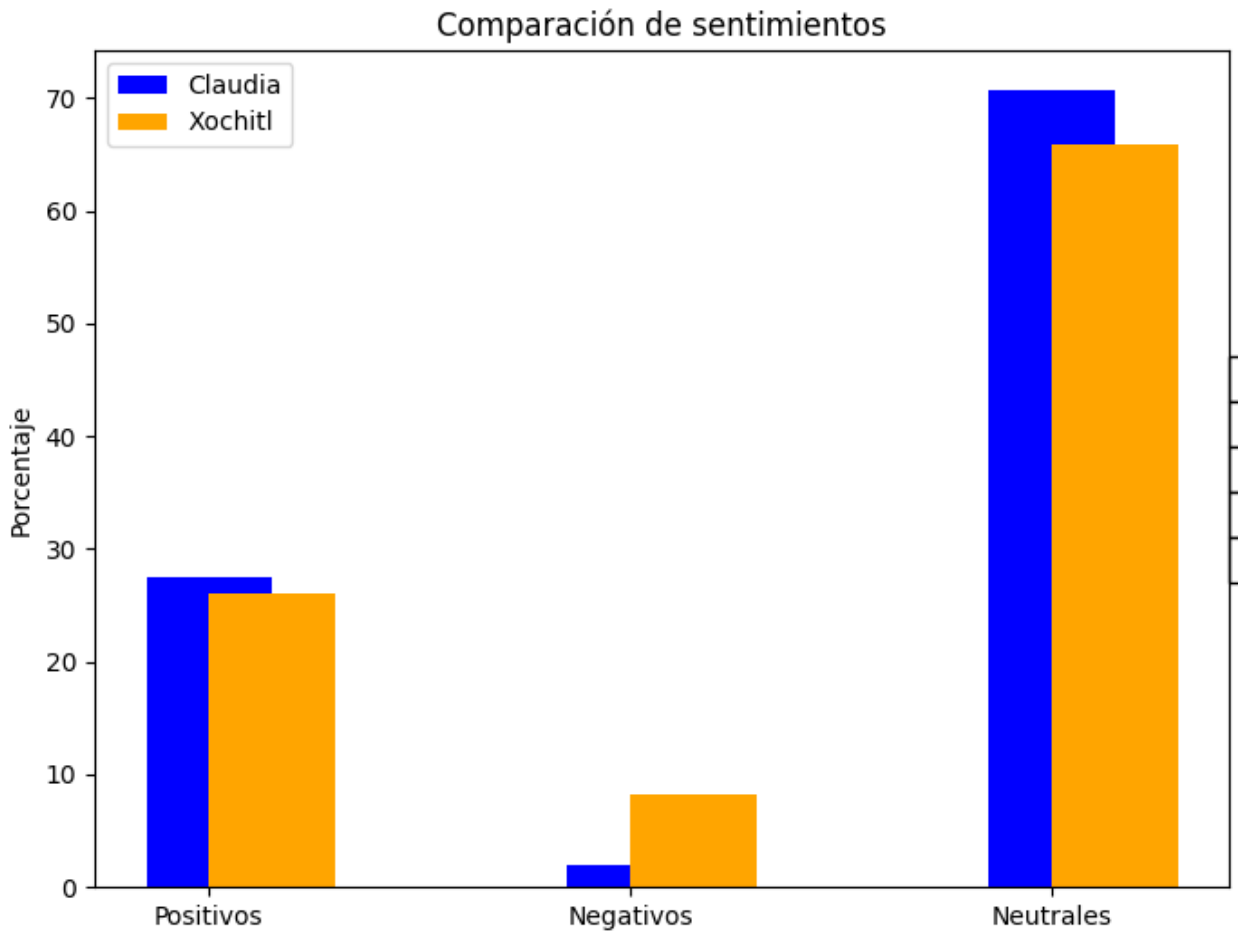
Para el análisis post-debate se decidió esperar un aproximado de 14 días antes de comenzar a minar tweets de nuevo, una vez transcurrido el tiempo y tras haber sido obtenida la nueva información se obtuvieron nuevos resultados que al inicio eran hasta cierto punto un tanto desconcertantes, puesto que la balanza se inclinaba demasiado a lo neutral para ambas candidatas, lo que no hacia sentido al inicio, pero después de debatirlo en sesiones, pudimos deducir el incremento de estos, dándole como explicación el aumento de dudas y comentarios alusivos al debate, puesto que la gente tenia mas dudas en los comentarios que textos de apoyo o desaprobación hacia ambas candidatas en cuestión, dándole fin de este modo a nuestro análisis de sentimientos en los que en nuestro caso se dio una mayor inclinación hacia la candidata Claudia Sheinbaum Pardo.

Gráfica Pre-Debate



	Claudia	Xochitl
Total Tweets	1448	1146
Positivos	762	153
Negativos	40	195
Neutrales	646	798

Gráfica Post-Debate



	Claudia	Xochitl
Total Tweets	1061	943
Positivos	291	245
Negativos	20	77
Neutrales	750	621

CONCLUSIÓN

El análisis de sentimientos basado en procesamiento de lenguaje natural y técnicas de machine learning ofrece una poderosa herramienta para entender las emociones y opiniones expresadas en grandes volúmenes de texto, como los tweets. A lo largo de este script, hemos implementado un flujo de trabajo completo que abarca desde la preparación y limpieza de datos hasta el etiquetado y visualización de sentimientos. El uso de librerías como nltk y scikit-learn ha permitido desarrollar un sistema robusto que puede preprocesar texto de manera efectiva, eliminar el ruido y resaltar las palabras clave que indican sentimientos positivos y negativos.

La capacidad de este sistema para procesar archivos de Excel y convertir textos en una forma adecuada para el análisis es crucial, especialmente en escenarios donde los datos se almacenan en formatos estructurados. La integración de funciones auxiliares para la limpieza y extracción de palabras asegura que los datos estén listos para el análisis, mejorando la precisión y confiabilidad del modelo.

El entrenamiento de un modelo de clasificación Naive Bayes añade un nivel adicional de sofisticación, permitiendo predicciones más precisas sobre el sentimiento de nuevos textos. La posibilidad de guardar y reutilizar el modelo entrenado facilita la escalabilidad y la eficiencia en el análisis de futuros conjuntos de datos.

Finalmente, la visualización de los resultados a través de gráficos de barras y tablas proporciona una forma intuitiva de interpretar y comunicar los hallazgos. La comparación de sentimientos entre diferentes conjuntos de tweets revela patrones y tendencias que pueden ser fundamentales para la toma de decisiones informadas en diversos contextos, desde el análisis de mercado hasta la monitorización de la opinión pública.

BIBLIOGRAFÍA

- Schuster, M., & Nakajima, K. (2016). WordPiece tokenization. arXiv preprint arXiv:1609.08242. https://arxiv.org/abs/1609.08242
- Sobrino, J. (2018). Analísis de sentiemientos en twiiter (Universitat Oberta de Catalunya, Vol. 1) [Electronico].
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. https://arxiv.org/abs/1810.04805
- NLTK : Natural Language Toolkit. (s. f.). https://www.nltk.org/
- Transformers. (s. f.). https://huggingface.co/docs/transformers/index