Análisis de sentimientos en redes sociales

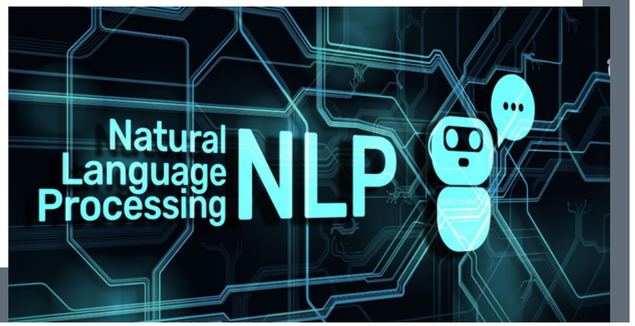
NLTK

Objetivos Generales:

Desarrollar un sistema de análisis de sentimientos en tiempo real para redes sociales, utilizando técnicas de NLP y optimizado para procesamiento paralelo.

Objetivos Específicos:

Extraer y preprocesar datos de redes sociales, implementar modelos de análisis de sentimientos, y paralelizar el proceso para mejorar el rendimiento.



METODOLOGÍA

Enfoque ágil y la danza de los sprints.

Se desarrollo 3 Sprint.
Seguimiento del progreso y resolución de impedimentos
Revisión de Sprint Retrospectiva de Sprint
Evaluación del trabajo completado al final de cada sprint
Mejorar futuras

Herramientas:

- Entorno de Desarrollo:
- Google Colab
- Control de Versiones:
- GitHub
- Extracción y Procesamiento de Datos:
- API de Twitter v2
- Tweepy para interacción con la API
- MongoDB para almacenamiento de datos
- Procesamiento de Lenguaje Natural:
- NLTK para preprocesamiento de texto
- Transformers (Hugging Face) para modelos basados en BERT

Machine Learning:

- scikit-learn para el modelo Naive Bayes y evaluación de modelos
- PyTorch para implementación y fine-tuning de BERT
- Optimización y Escalabilidad:
- multiprocessing para procesamiento paralelo
- Visualización y Presentación:
- Streamlit para el desarrollo del dashboard interactivo
- Matplotlib para generación de gráficos
- Lenguaje de Programación y Bibliotecas Principales:
- Python 3.x
- pandas para manipulación y análisis de datos

Sprint 1

Objetivos

- Configurar y utilizar la API de Twitter para la extracción de datos.
- Implementar un sistema de almacenamiento eficiente para los datos extraídos.
- Desarrollar un pipeline de preprocesamiento de datos robusto.
- Evaluar la calidad de los datos extraídos y preprocesados.

Logros

- Implementación exitosa de la extracción de datos de Twitter utilizando la API v2
- Reducción y refinamiento del dataset de 9,028 a 1,201 tweets, mejorando su calidad para el análisis de sentimientos.

Sprint 2

Objetivos

1. Implementar y entrenar modelos de análisis de sentimientos.
2. Evaluar y comparar el rendimiento de los modelos implementados.
3. Optimizar los modelos para mejorar la precisión en la clasificación de sentimientos.
4. Implementar técnicas de procesamiento paralelo para mejorar la eficiencia.

Logros

- Detección de idioma, tokenización, eliminación de stopwords y generación de n-gramas.
- Implementación de procesamiento paralelo para mejorar la eficiencia.
- Uso de técnicas de paralelización para el entrenamiento y predicción de modelos.
- Reducción significativa en los tiempos de procesamiento (ej. preprocesamiento completado en 23.71 segundos).

Sprint 3

Objetivos

- Realizar un análisis comparativo profundo de los modelos implementados.
- Desarrollar un dashboard interactivo para la visualización y uso de los modelos.
- Implementar funcionalidades de análisis en tiempo real en el dashboard.

Logros

- Evaluación de tamaño en memoria, escala y rendimiento de ambos modelos.
- Implementación de técnicas de carga eficiente de modelos para mejorar el tiempo de respuesta.
- Interfaz de usuario intuitiva para análisis de sentimientos en tiempo real.

RESULTADOS

- **Funcionalidades desarrolladas**

Implementación de modelos de clasificación:
- Modelo BERT (XLM-RoBERTa) para análisis avanzado
- Modelo Naive Bayes como baseline eficiente

Procesamiento paralelo:
- Preprocesamiento de datos en paralelo
- Entrenamiento y predicción paralela para Naive Bayes

Evaluación comparativa de modelos:
- Cálculo de métricas de rendimiento (precisión, recall, F1-score)
- Generación de matrices de confusión

Análisis de escalabilidad y rendimiento:
- Medición de tiempos de procesamiento y predicción
- Comparación de tamaños de modelo y requisitos de recursos

RESULTADOS

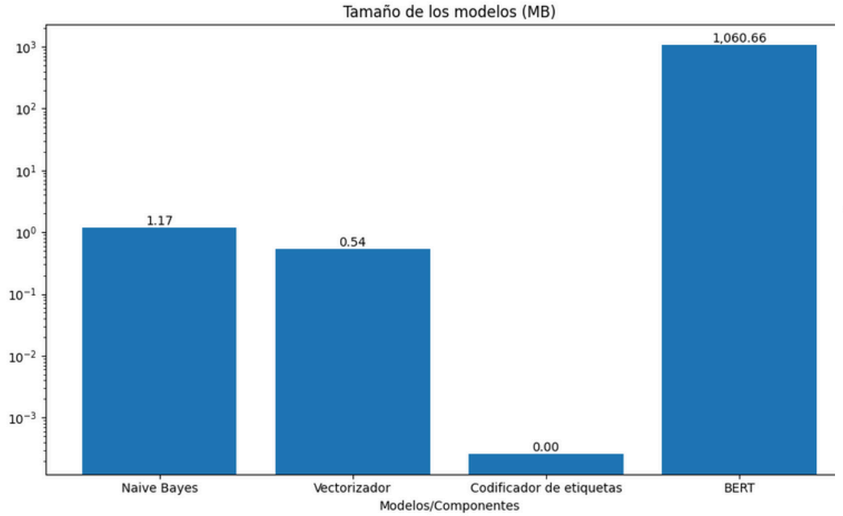
- Resultados de pruebas y análisis de rendimiento

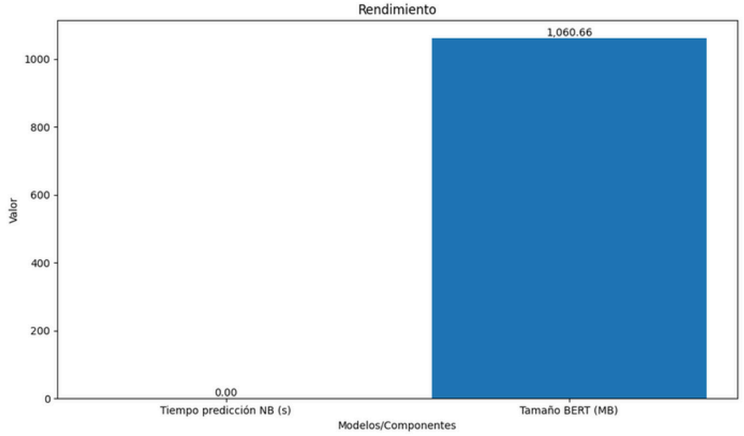
Reporte de clasificación BERT:

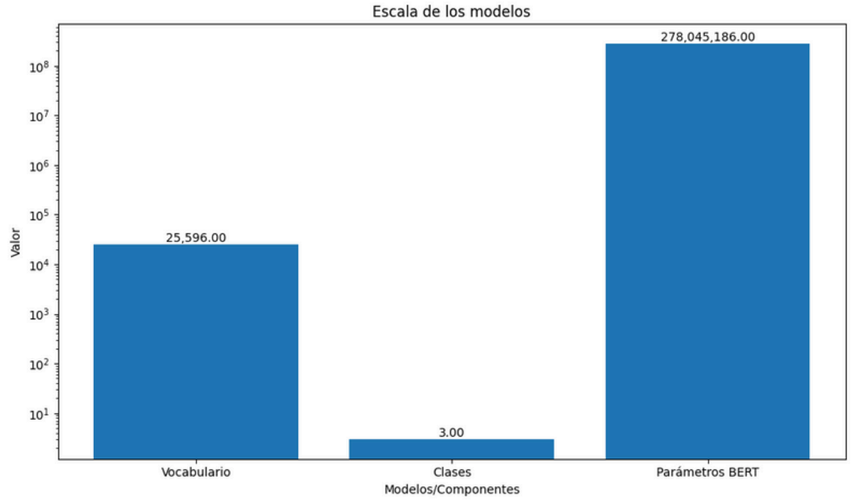
	precision	recall	f1-score	support
0	0.89	0.88	0.88	82
1	0.92	0.83	0.88	59
2	0.87	0.93	0.90	100
accuracy			0.89	241
macro avg	0.89	0.88	0.89	241
weighted avg	0.89	0.89	0.89	241

Reporte de clasificación Naive Bayes:

	precision	recall	f1-score	support
0	0.90	0.73	0.81	82
1	0.78	0.78	0.78	59
2	0.79	0.91	0.85	100
accuracy			0.82	241
macro avg	0.82	0.81	0.81	241
weighted avg	0.82	0.82	0.82	241







DEMOSTRACIÓN EN VIVO

Texto

Te amo y te odio
El servicio al cliente fue terrible, muy decepcionado.
A qualidade é aceitável, mas pode melhorar

Analizar

Línea 1:

Predicción de Naive Bayes: Negativo

Predicción de BERT: Negativo

Línea 2:

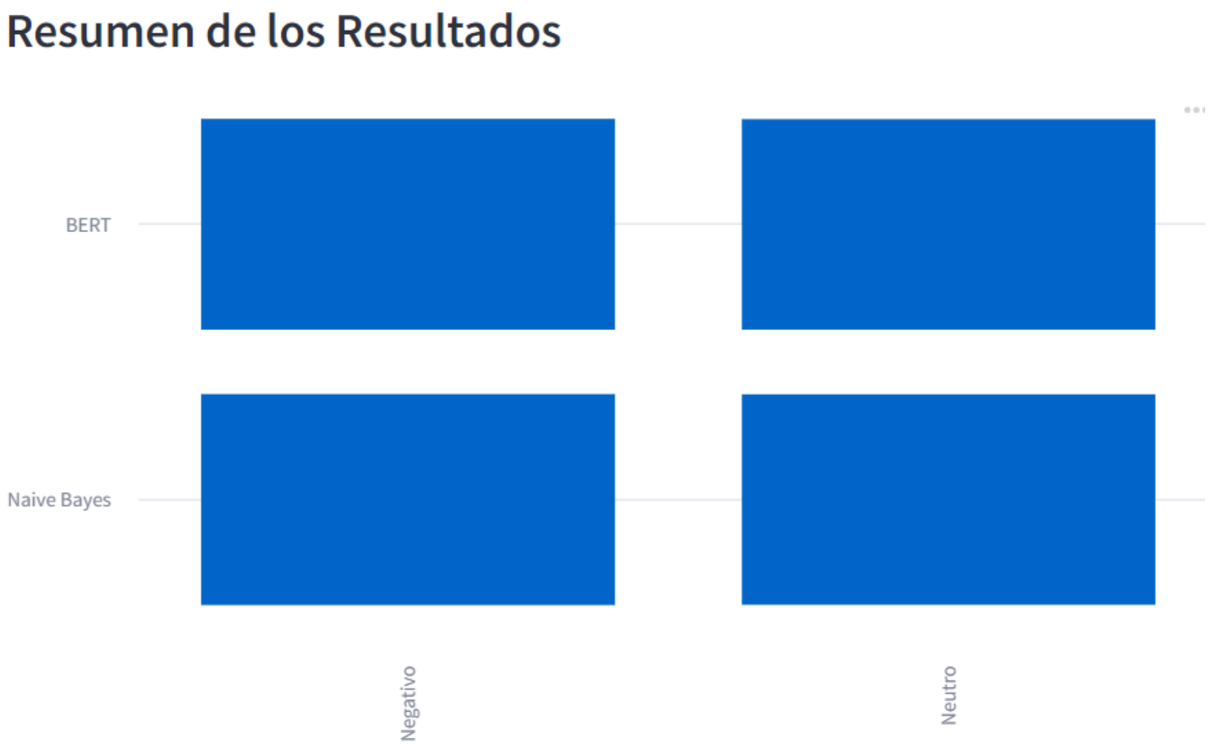
Predicción de Naive Bayes: Negativo

Predicción de BERT: Negativo

Línea 3:

Predicción de Naive Bayes: Neutro

Predicción de BERT: Neutro



ANÁLISIS Y EVALUACIÓN

• Lecciones aprendidas

- Importancia del preprocesamiento de datos: Un preprocesamiento robusto y eficiente es crucial para el rendimiento de los modelos de NLP.
- Valor del enfoque multimodelo: Comparar diferentes enfoques (BERT vs Naive Bayes) proporciona insights valiosos sobre las fortalezas y debilidades de cada técnica.
- Eficacia de la paralelización: Implementar técnicas de procesamiento paralelo puede mejorar significativamente la eficiencia, especialmente en conjuntos de datos grandes.

• Desafíos y soluciones

Manejo de límites de API:

• Desafío: Restricciones en la extracción de datos de Twitter.

• Solución: Implementación de pausas y manejo de excepciones para respetar los límites de tasa.

Manejo de datos multilingües:

• Desafío: Procesar textos en diferentes idiomas.

• Solución: Implementación de detección automática de idioma y uso de modelos preentrenados multilingües como XLM-RoBERTa.

Optimización del rendimiento:

• Desafío: Tiempo de procesamiento largo para grandes volúmenes de datos.

• Solución: Implementación de técnicas de procesamiento paralelo y optimización de la carga de modelos

CONCLUSIÓN Y FUTURO TRABAJO

Resumen de los logros

Implementamos y evaluamos dos modelos:

- Los datos y modelos están preparados para su uso en aplicaciones futuras.
- Optimizamos los modelos para mejorar la precisión en la clasificación de sentimientos.
- Implementamos técnicas de procesamiento paralelo para mejorar la eficiencia.

Posibles mejoras y expansiones futuras

Optimización de Rendimiento

- Investigar técnicas avanzadas de optimización para mejorar el rendimiento.
- Implementar estrategias de compresión de modelos, especialmente para BERT.

Integración de Tecnologías Distribuidas

- Profundizar en la integración de Dask y PySpark para procesamiento distribuido.
- Explorar soluciones de almacenamiento distribuido para manejar volúmenes de datos mayores.
- Aplicar mas paralelismo a los modelos de entrenamiento.