Reputation Analyzer

Entrega Final
Informe obligatorio 2024

Grupo 7

Joaquin Campo Nario - 5.280.080-4
Mateo Daneri Dini - 5.660.750-1
Valentin Dutra Claudio - 5.231.787-7

Docente

Libertad Tansini

Contents

1 Presentación del problema

1.1 Motivación

En el año 2024, la competencia electoral en Estados Unidos es intensa y la opinión pública, influenciada por las redes sociales, juega un papel crucial. Sin embargo, no existe una herramienta automatizada y eficiente que permita analizar en tiempo real los sentimientos de los usuarios hacia los candidatos presidenciales. Nuestro proyecto, "Reputation Analyzer", busca solucionar este problema desarrollando una aplicación web que recupere publicaciones del subreddit "r/politics" en Reddit y utilice el modelo "xlm-roberta-base-sentiment" para medir la positividad o negatividad de las mismas. Esto permitirá obtener una métrica precisa de la aceptación o rechazo hacia los candidatos, proporcionando información valiosa y actualizada para analistas políticos y el público en general.

2 Enfoque de la Solución

Nuestra solución se centra en desarrollar una aplicación web, denominada "Reputation Analyzer", que permite analizar la opinión pública sobre los candidatos presidenciales en Estados Unidos utilizando datos de Reddit. El enfoque principal incluyó las siguientes etapas:

2.1 Recuperación de Datos

Integramos la API de Reddit para obtener publicaciones relevantes del subreddit "r/politics", asegurando una fuente de datos rica y actualizada.

2.2 Análisis de Sentimientos

Utilizamos el modelo de aprendizaje automático "xlm-roberta-base-sentiment" para evaluar la positividad o negatividad de las publicaciones. Este modelo nos permitió asignar una métrica de sentimiento a cada publicación, facilitando el análisis cuantitativo de la opinión pública.

2.3 Infraestructura Tecnológica

Implementamos un stack tecnológico que incluye React para el Front-End, Python y FastAPI para el Back-End, y ElasticSearch para la gestión y búsqueda de datos. Esta combinación de tecnologías asegura una aplicación robusta, eficiente y escalable. Además, el back-end está contenido en un contenedor Docker, lo que facilita su despliegue y escalabilidad.

2.4 Visualización de Resultados

Diseñamos una interfaz amigable y visualmente atractiva para presentar los resultados del análisis de sentimientos de manera clara y comprensible para los usuarios.

3 Diseño del Sistema e Implementación

El diseño del sistema y la implementación de "Reputation Analyzer" se estructuraron en varias etapas clave para asegurar una solución robusta y eficiente.

3.1 Arquitectura del Sistema

Optamos por una arquitectura basada en microservicios para separar las responsabilidades y facilitar el mantenimiento y escalabilidad del sistema. Esta arquitectura incluye capas de presentación, servicios, negocio y datos.

3.2 Front-End

React es una biblioteca de JavaScript para construir interfaces de usuario. Utilizamos React para el desarrollo del front-end debido a su capacidad para construir interfaces de usuario interactivas y su compatibilidad con componentes reutilizables. Esto permitió un diseño modular y una experiencia de usuario dinámica y responsiva.

3.3 Back-End

Python es un lenguaje de programación versátil y FastAPI es un framework web moderno y de alto rendimiento para construir APIs con Python. Para el back-end, elegimos Python y FastAPI por su versatilidad y la amplia disponibilidad de bibliotecas y frameworks. FastAPI nos permitió crear una API RESTful que maneja la lógica de negocio y coordina la interacción entre el front-end y las bases de datos. Además, todo el back-end está contenido en un contenedor Docker, lo que mejora la portabilidad y simplifica el despliegue.

3.4 Gestión de Datos

ElasticSearch es un motor de búsqueda y análisis distribuido, diseñado para la búsqueda en tiempo real y el análisis de datos estructurados y no estructurados. Implementamos ElasticSearch tanto para la creación de índices invertidos como para la gestión de la base de datos. Esta elección nos permitió realizar búsquedas eficientes y manejar grandes volúmenes de datos de manera efectiva. A continuación se muestra el código utilizado para crear los índices en ElasticSearch:

```
mapping = {
    "mappings": {
        "properties": {
            "related_entity": {"type": "text"},
            "id": {"type": "keyword"},
            "name": {"type": "text"},
            "title": {"type": "text"},
            "text": {"type": "text"},
```

```
            "date": {"type": "date", "format": "yyyyMMdd"},
            "cant_comments": {"type": "integer"},
            "thumbsup": {"type": "integer"},
            "link": {"type": "text"},
            "subreddit": {"type": "text"},
            "sentiment": {"type": "text"},
            "score": {"type": "integer"}
        }
    }
}
```

Este índice contiene varios campos, como `related_entity`, `id`, `name`, `title`, `text`, `date`, `cant_comments`, `thumbsup`, `link`, `subreddit`, `sentiment` y `score`, permitiendo búsquedas eficientes y análisis detallado de las publicaciones.

```
mapping = {
    "mappings": {
        "properties": {
            "related_entity": {"type": "text"},
            "post_id": {"type": "keyword"}
        }
    }
}
```

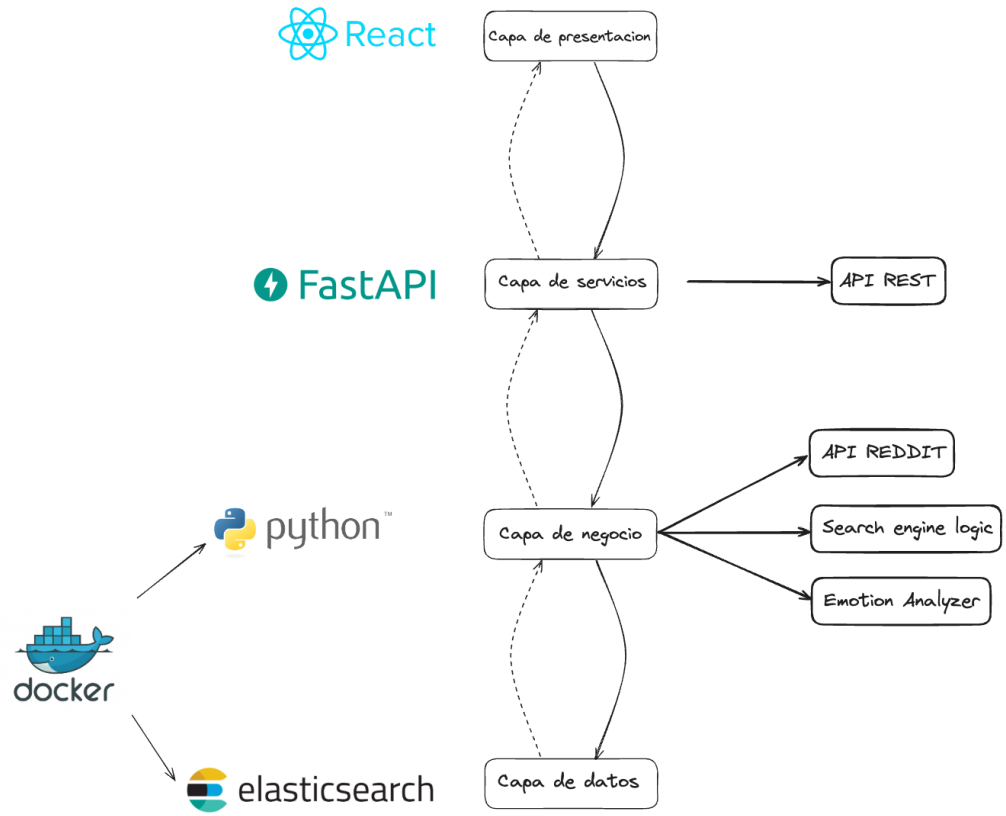
Este índice almacena el identificador del último post recuperado, facilitando la actualización de datos de manera eficiente.

3.5 Integración con Reddit

La API de Reddit permite a los desarrolladores interactuar con la plataforma de Reddit para recuperar y enviar datos. La integración con la API de Reddit fue crucial para recuperar publicaciones del subreddit "r/politics". Este proceso incluyó la autenticación con OAuth y el uso de endpoints específicos para obtener datos relevantes.

3.6 Análisis de Sentimientos

RoBERTa es un modelo de aprendizaje profundo basado en la arquitectura Transformer, preentrenado para diversas tareas de procesamiento del lenguaje natural, incluyendo el análisis de sentimientos. Utilizamos el modelo de aprendizaje automático "xlm-roberta-base-sentiment" para analizar el sentimiento de las publicaciones recuperadas. Este modelo, preentrenado y ajustado para tareas de análisis de sentimientos, permitió clasificar las publicaciones en positivas, negativas o neutras con alta precisión.



3.7 Visualización y Presentación de Datos

Diseñamos una interfaz de usuario intuitiva para presentar los resultados del análisis de sentimientos. Utilizamos gráficos y visualizaciones interactivas para facilitar la interpretación de los datos y proporcionar una experiencia informativa y atractiva para los usuarios.

4 Obstáculos encontrados

En el transcurso del desarrollo, afrontamos diversos contratiempos los cuales nos llevaron a tomar distintas decisiones sobre el diseno de la aplicación. Por ejemplo, a pesar de la rapidez considerable que provee el Emotion Analyzer en su análisis, al tener que considerar muestras de datos grandes y diversas para llegar a un resultado objetivo, ésta llevaba del orden de los 5 a 10 minutos para muestras de al rededor de 1000 publicaciones de Reddit. Para combatirlo, se intentó darle más recursos de Hardware al proceso ejecutando el análisis, investigando incluso en el uso de una GPU, pero sin mayor éxito.

5 Resultados Obtenidos

La implementación de "Reputation Analyzer" ha generado varios resultados significativos, los cuales describiremos a continuacion:

5.1 Rendimiento del Sistema

El uso de ElasticSearch permitió realizar búsquedas y análisis de datos en tiempo real, con tiempos de respuesta del orden de minutos para consultas complejas. La arquitectura basada en microservicios y Docker demostró ser altamente escalable. Además, al recurrir a una solución con datos precargados al sistema, se pueden obtener estimaciones de las reputaciones de los candidatos en el orden de los milisegundos, a costa de un análisis no tan reciente de las mismas.

5.2 Interacción del Usuario

La interfaz desarrollada con React ofreció una experiencia de usuario intuitiva y fluida. Se permite a los usuarios visualizar los resultados del análisis de sentimientos de manera interactiva, mejorando la comprensión y el uso de los datos obtenidos con las gráficas de las estadísticas obtenidas.

6 Conclusiones

La implementación de "Reputation Analyzer" ha demostrado ser una herramienta poderosa y eficiente para el análisis de la opinión pública en el contexto electoral de Estados Unidos en 2024. A continuación, se presentan las conclusiones clave obtenidas a partir del desarrollo y uso de la aplicación:

6.1 Precisión y Eficiencia

El modelo de análisis de sentimientos "xlm-roberta-base-sentiment" ha permitido evaluar con alta precisión la positividad o negatividad de las publicaciones en Reddit. La integración de tecnologías como React, FastAPI, ElasticSearch y Docker ha garantizado una infraestructura robusta, eficiente y escalable, capaz de manejar grandes volúmenes de datos y ofrecer resultados claros.

6.2 Valor para los Usuarios

La herramienta ha proporcionado a los analistas políticos y al público en general una visión clara y actualizada de la opinión pública sobre los candidatos presidenciales. La interfaz intuitiva desarrollada con React ha facilitado la interpretación y visualización de los datos, mejorando la toma de decisiones informadas.

6.3 Escalabilidad y Adaptabilidad

La arquitectura basada en microservicios y el uso de contenedores Docker han permitido una implementación escalable y adaptable, teóricamente capaz de manejar aumentos en la carga de trabajo sin comprometer el rendimiento. Esto asegura que la herramienta puede ser ampliada y mejorada para futuros contextos y aplicaciones, como el análisis de la reputación de marcas y figuras públicas en diversas plataformas sociales.

6.4 Oportunidades Futuras

El éxito de "Reputation Analyzer" abre la puerta a múltiples oportunidades de mejora y expansión. Entre ellas, la posibilidad de integrar datos de otras plataformas sociales como Twitter y Facebook, aplicar filtros geográficos y temporales, y personalizar los análisis según necesidades específicas de los usuarios.

En resumen, "Reputation Analyzer" ha cumplido con los objetivos planteados, demostrando ser una herramienta valiosa para el análisis de sentimientos y la comprensión de la opinión pública en el entorno político actual. Su diseño robusto y escalable asegura su relevancia y utilidad para futuros desarrollos y aplicaciones.

7 Trabajos Futuros

Entre los trabajos futuros que podrían implementarse se encuentran:

7.1 Integración con Más Redes Sociales

Ampliar el alcance de la herramienta para incluir datos de otras plataformas como Twitter, Facebook, Instagram y YouTube, lo que proporcionaría una visión más completa de la opinión pública.

7.2 Filtros Geográficos y Temporales

Implementar filtros que permitan analizar la opinión pública por ubicación geográfica y periodo de tiempo, proporcionando análisis más segmentados y contextuales.

7.3 Análisis de Tendencias

Desarrollar funcionalidades que detecten y analicen tendencias en tiempo real, permitiendo identificar cambios significativos en la opinión pública a lo largo del tiempo.

7.4 Dashboard Personalizado

Crear dashboards personalizables para diferentes tipos de usuarios, como analistas políticos, periodistas y empresas, permitiendo visualizar y explorar datos según sus necesidades específicas.

7.5 Mejora del Modelo de Sentimientos

Entrenar modelos de análisis de sentimientos más avanzados y específicos para distintos dominios (e.g., política, economía, entretenimiento), mejorando la precisión del análisis.

7.6 Feedback y Mejora Continua

Implementar mecanismos de feedback que permitan a los usuarios sugerir mejoras y reportar errores, asegurando la mejora continua de la herramienta basada en las necesidades de los usuarios.

7.7 Predicción de Resultados

Desarrollar modelos predictivos que utilicen los datos de sentimientos para prever resultados de eventos futuros, como elecciones, referendos, o lanzamientos de productos.

7.8 Integración con Herramientas de Visualización

Mejorar las capacidades de visualización de datos mediante la integración con herramientas avanzadas de visualización como D3.js o Tableau, permitiendo análisis más profundos y visuales.

8 Referencias

React: https://es.reactjs.org/
Docker: https://www.docker.com/
Python: https://www.python.org/
ElasticSearch: https://www.elastic.co/
API Reddit: https://www.reddit.com/dev/api/
RoBERTa: https://huggingface.co/transformers/model_doc/roberta.html
Repositorio GitHub: https://github.com/JoaquinCampo/WIR2024