Informe de Distribución de roles del Proyecto de Regresión

Participantes: Nicolás Bedoya (202212100), Alejandro Pulido (202215711) y Giuliana Volpi (202123986)

Roles y tareas realizadas

Giuliana: Realizó el Pipeline permitiendo una ejecución secuencial y reutilizable del proyecto. Además, se incluyó un análisis de impacto de palabras por clase y el modelo final fue serializado usando joblib para generar el despliegue y uso en producción.

Nicolás: Apoyó el proceso de realización del Pipeline. Se propuso otro enfoque para que su reentrenamiento fuera factible, revisión de errores, entre otros. Investigación de las formas de reentrenamiento y elección de la misma a partir del contexto del problema.

Alejandro: Desarrolló la aplicación, tanto interfaz como API.

Distribución de puntos entre los integrantes

Dado que el trabajo en equipo fue equitativo y se cumplieron todos los objetivos establecidos dentro de las fechas propuestas se decidió hacer la división de los puntos equitativamente entre los tres integrantes del grupo.

- Giuliana: 33.3 pts
- Nicolás: 33.3 pts
- Alejandro: 33.3 pts

Reuniones del equipo

Reunión 1: Lanzamiento y Planeación

Fecha: miércoles 19 de marzo

Participantes: Giuliana, Nicolás y Alejandro
Objetivo:

- Comprender completamente el enunciado de la Etapa 2.

- Distribuir roles entre los miembros.
- Definir entregables intermedios y establecer responsables

Reunión 2: Seguimiento de Desarrollo Técnico

Fecha: sabado 22 de marzo

Participantes: Giuliana, Nicolás y Alejandro

Objetivo:

- Revisar el avance técnico del pipeline, modelo y API.
- Validar funcionamiento del primer endpoint /predict.
- Comenzar pruebas locales con Postman y conectar frontend básico.

Reunión 3: Revisión Intermedia y Ajustes

Fecha: lunes 24 de marzo

Participantes: Giuliana, Nicolás, Alejandro
Objetivo:

- Validar el funcionamiento completo de ambos endpoints.
- Alinear reentrenamiento con definición elegida.
- Revisión y ajustes del flujo de datos entre frontend y backend.

Reunión 4: Pruebas de Usabilidad

Fecha: miércoles 26 de marzo
Participantes: Giuliana, Nicolás, Alejandro

Objetivo:

- Validar la experiencia de uso de la aplicación.
- Observar interacción con el sistema y recopilar feedback

Reunión 5: Cierre y Consolidación

Fecha: viernes 28 de marzo
Participantes: Giuliana, Nicolás, Alejandro
Objetivo:

- Validar todos los entregables antes del envío.
- Finalizar video y realizar pruebas integradas de todo el sistema.
- Revisar documento de entrega completo, incluyendo Sección 4 (trabajo en equipo).

Proceso de automatización

Para automatizar el proceso de la preparación de datos, construcción y persistencia del modelo se llevó a cabo la creación de un pipeline, dividido en 3 fases fundamentales para que la tarea se realizara correctamente.

- **Cleaner:** Esta es la primera etapa del pipeline, en la cual se llevó a cabo todo el proceso relacionado a la preparación de los datos. Dentro de este paso se realizaron procesos como eliminación de stop words, lematización, conversión de mayúsculas a minúsculas, eliminación de signos de puntuación, entre otros. Con las anteriores operaciones se garantiza que los datos a usar por el modelo estén limpios y no se generen posibles errores o sesgos durante el entrenamiento del mismo.
- **Vectorizer:** Una vez finalizado el proceso de limpieza se continuó con la vectorización de los textos y palabras para que así el modelo sea capaz de reconocer patrones y realizar las operaciones de computación necesarias.
- **Model:** Después de realizar la limpieza y vectorización de los datos se realiza el paso final, el cual está asociado a la creación del modelo y su entrenamiento. Acorde a los resultados obtenidos en la Etapa 1, se hace uso de una regresión logística, con parámetros c=1 y penalidad de l1. Una vez creado el modelo se realiza el entrenamiento del mismo haciendo uso de los datos de la Etapa 1. Finalmente, las métricas solicitadas son reportadas

El pipeline descrito previamente fue convertido en un joblib para que así la API pueda cargarlo fácilmente y hacer uso del mismo.

A continuación, se listan las formas de reentrenamiento planteadas.

- **Full retraining:** Usado para volver a entrenar un modelo desde cero haciendo uso de todos los conjuntos de datos disponibles, se usan tanto datos antiguos como nuevos. Este enfoque es particularmente útil en caso de que los datos cambien drásticamente. Es efectivo pero costoso en términos de tiempo y recursos debido al procesamiento completo de los datos
- **Incremental learning:** Este método consiste en entrenar de forma continua un modelo conforme se dispone de nuevos datos. Este enfoque suele ser usado cuando los datos se generan de forma secuencial o existe una cantidad de datos muy grande, puesto que no es práctico almacenarlos y procesaros todos a la vez.
- **Warm-start training:** Warm-start training o entrenamiento con inicio en caliente consiste en entrenar el modelo haciendo uso de los pesos y parámetros de un modelo que ya se tenía previamente, garantizando que se sigue usando el conocimiento previo que este tenía.

Acorde a la descripción de los métodos se decidió hacer uso del Warm-start training, puesto que permite hacer uso del conocimiento previo del modelo.

Desarrollo de la aplicación

Teniendo en cuenta que el propósito del modelo es clasificar noticias falsas, se determina que un posible usuario serían entidades gubernamentales que busquen la forma de poder determinar si una noticia es falsa o no de forma rápida para así evitar tensiones internacionales. Tomando como un proceso de negocio el hecho de que un funcionario gubernamental de alto rango vaya a realizar un comunicado frente a una noticia reciente, el modelo apoyaría para dar una primera instancia de cómo debería de ser dicho comunicado, evitando que en caso de que la noticia sea falsa, el funcionario cometa un error al asumirla como verdadera. En este mismo punto radica la importancia de la aplicación para este tipo de organizaciones.

Igualmente, ciudadanos del común también podrían llegar a tener interés en esto por lo que también serían usuarios potenciales.

¿Qué recursos informáticos requiere para entrenar, ejecutar, persistir el modelo analítico y desplegar la aplicación?

Es necesario tener acceso a un computador con mínimo 16 gigas de RAM. Se determina esta cantidad teniendo en cuenta el conjunto de datos brindado inicialmente y el tiempo que tomó entrenar, ejecutar y persistir el modelo analítico.

¿Cómo se integrará la aplicación construida a la organización, estará conectada con algún proceso del negocio o cómo se pondrá a disposición del usuario final?

La aplicación funcionará de forma independiente para las personas que deseen usarla, accediendo a ella como si fuera una página web. Para entidades gubernamentales se les daría una versión distinta en la que son capaces de reentrenar el modelo cuando les parezca pertinente. La plataforma podría ser integrada directamente a un departamento de analítica que esté enfocado directamente en el análisis de noticias.

¿Qué riesgos tiene para el usuario final usar la aplicación construida?

Es importante tener en cuenta que por más que se hayan presentado muy buenos resultados respecto al modelo de clasificación, estos no son perfectos. Esto implica que el uso desmedido de la aplicación podría llegar a generar problemas en caso de no ser monitoreada correctamente, puesto que es posible que clasifique una noticia falsa como verdadera y viceversa. Por ello, es importante que las entidades que hagan uso de la aplicación tengan una forma de corroborar que la noticia es falsa (por medio de un analista, por ejemplo) en caso de que la probabilidad asignada no sea tan alta.