**Trabajo en equipo - Descripción de roles y tareas**

**Integrantes:**

* **Andrés Pérez**
* **Juana Morales**
* **John Serrato**

**Descripción del código:**

Este código implementa una API utilizando FastAPI para un modelo analítico basado en aprendizaje automático que clasifica textos en categorías de los Objetivos de Desarrollo Sostenible (ODS). La API tiene dos funcionalidades principales:

1. **Predicción de categorías ODS**:
   * El usuario puede cargar un archivo CSV con textos en español.
   * El código preprocesa los textos (normalización y eliminación de caracteres especiales).
   * Luego, el modelo entrenado (SVM) predice la categoría ODS para cada texto y devuelve la predicción y su probabilidad en porcentaje.
2. **Reentrenamiento del modelo**:
   * El usuario puede cargar nuevos datos para reentrenar el modelo.
   * Los nuevos datos se combinan con los datos anteriores para actualizar el modelo.
   * Tras el reentrenamiento, se calculan y devuelven métricas de desempeño como precisión, recall, F1-score y exactitud.

Adicionalmente, el código:

* Utiliza un pipeline de preprocesamiento con TF-IDF y un clasificador SVM.
* Implementa un flujo de trabajo para preprocesar textos, manejar la carga de archivos, y actualizar el modelo.
* Utiliza la librería joblib para persistir el modelo entrenado y uvicorn para ejecutar la API.

**Definiciones de reentrenamiento:**

1. **Reentrenamiento incremental:**

Este método actualiza el modelo solo con los nuevos datos proporcionados, sin empezar desde cero. Es eficiente en términos de tiempo y recursos, ya que no se necesita procesar todo el conjunto de datos desde el principio.

* **Ventajas**: Su eficiencia permite actualizaciones rápidas y frecuentes, ideal para sistemas de producción que requieren adaptarse rápidamente a nuevos datos.
* **Desventajas**: Puede generar un desbalance si los nuevos datos no son representativos de la distribución completa o son muy diferentes de los datos originales.

1. **Reentrenamiento completo:**

El modelo se entrena nuevamente desde cero, usando tanto los datos antiguos como los nuevos, procesando todo el conjunto de datos desde el principio.

* **Ventajas**: Asegura que el modelo tenga una visión completa de los datos, evitando sesgos hacia los datos nuevos y corrigiendo posibles errores acumulados.
* **Desventajas**: Es más costoso en términos de tiempo y recursos computacionales, ya que requiere un entrenamiento completo cada vez que se añaden nuevos datos.

1. **Reentrenamiento con selección de muestras:**

Se selecciona una muestra representativa de los datos antiguos junto con los nuevos para reentrenar el modelo, manteniendo un balance en la distribución de los datos.

* **Ventajas**: Permite un reentrenamiento más rápido que el completo, manteniendo una representación equilibrada de los datos originales y los nuevos.
* **Desventajas**: La selección de las muestras puede no siempre ser representativa, lo que podría afectar la precisión del modelo.

**Decisión de uso:**

En esta etapa, se implementó el reentrenamiento incremental por su eficiencia en términos de recursos y tiempo, lo cual es clave en sistemas de producción que requieren actualizaciones frecuentes con nuevas instancias de datos.

**2. Descripción usuario/rol:**

El usuario es un ingeniero de sostenibilidad o un especialista en desarrollo sostenible dentrode una organización. Su función principal es alinear las operaciones y proyectos de la organización con los Objetivos de Desarrollo Sostenible (ODS).

**Conexión con el proceso de negocio:**

Este usuario trabaja para garantizar que las actividades y proyectos de la organización sean sostenibles y cumplan con los ODS. La aplicación les ayuda a analizar grandes volúmenes de texto, como informes de sostenibilidad y evaluaciones de impacto, identificando automáticamente los ODS relevantes.

**Importancia de la aplicación:**

Esta herramienta es crucial para los ingenieros de sostenibilidad, ya que automatiza el análisis y la clasificación de textos según los ODS. Esto reduce significativamente la necesidad de trabajo manual y agiliza la toma de decisiones basadas en datos.

**Beneficios:**

El uso de la aplicación permite tomar decisiones más informadas sobre proyectos prioritarios y asignación de recursos. Además, ayuda a asegurar que la organización esté en el camino correcto para cumplir con sus metas de sostenibilidad de manera efectiva y eficiente.

**3. Roles asignados:**

1. **Líder de proyecto (Andrés Pérez)**

**Responsabilidades:** Definió las fechas clave, como reuniones y avances. Adicional a ello, realizó la gestión de asignación de tareas asegurando una carga de trabajo equilibrada entre todos los integrantes. Coordinó las reuniones del equipo y toma de decisiones en situaciones donde no se llegara a un consenso. Por último se aseguró de la subida del proyecto final.

**Tiempo dedicado:** 1 hora y media

**Retos enfrentados:** Gestionar la disponibilidad de los integrantes y ajustar las tareas cuando algunos miembros enfrentaban dificultades técnicas, como con Google Colab.

**Solución:** Se ajustaron las tareas y tiempos para que cada integrante pudiera cumplir con sus responsabilidades.

1. **Ingeniero de datos (Juana Morales):**

**Responsabilidades:** realizó la supervisión de la calidad en la automatización del proceso relacionado con la construcción del modelo analítico. Adicional a ello, aportó en el preprocesamiento de los datos para asegurar que el modelo analítico funcione correctamente. Posteriormente, desarrollo de la primera parte del API junto a Andrés. Junto con la implementación de los pipelines necesarios para garantizar un flujo de trabajo eficiente.

**Tiempo dedicado:** 2 horas y media

**Retos enfrentados:** Problemas con la implementación de la API en Google Colab debido a errores constantes.

**Solución:** Optó por cambiar el entorno de trabajo y colaboró en la implementación del API en Visual Studio, mejorando la estabilidad del proyecto.

1. **Ingeniero de software - Diseño de la aplicación y resultados (Andrés Pérez)**

**Responsabilidades:** Diseño la aplicación web a nivel general junto a Juana. Adicional a ello, realizó la implementación de la funcionalidad de la API para hacer predicciones basadas en los datos introducidos por los usuarios. También, desarrollo de la interfaz de usuario para la interacción con los resultados del modelo analítico. Y realizó las debidas pruebas que evidenciaran la correcta funcionalidad.

**Tiempo dedicado:** 3 horas

**Retos enfrentados:** Dificultades técnicas en la integración del API con la interfaz.

**Solución:** Ajustes en los endpoints y en la estructura del código para garantizar una correcta comunicación entre la API y la aplicación.

1. **Ingeniero de software - Desarrollo de la aplicación final (John Serrato):**

**Responsabilidades:** Pulió la parte final de la aplicación, asegurando la funcionalidad completa de la API. También realizó del video de presentación de resultados. Adicionalmente, realizó nuevas pruebas de facilidad de uso para garantizar la fluidez de la interacción con la aplicación. Y por último, hizo una simulación de dos acciones que el usuario final podría realizar con la aplicación, explicando cómo los resultados del modelo aportan a la toma de decisiones.

**Tiempo dedicado:** 2 horas y media

**Retos enfrentados:** No hubo fallas significativas, solo al inicio hubo pequeños errores en el uso de comandos para la ejecución de app.

**Solución:** Se adaptó el entorno de desarrollo a Visual Studio con la instalación de las librerías correspondientes y que funcionará todo adecuadamente.

**Reuniones del grupo:**

1. **Reunión de lanzamiento y planeación:** se definieron los roles y forma de trabajo, así como brainstorming de ideas para abordar el proyecto. Como resultado, se distribuyeron los roles de acuerdo con las fortalezas de cada integrante, estableciendo plazos internos para los entregables. En total duro aproximadamente 1 hora.
2. **Reunión de seguimiento:** esto lo realizamos vía mensaje (WhatsApp), con el objetivo de revisa e informar avances, identificar obstáculos y ajustar tareas según el progreso de cada integrante. Se realizó de manera constante (a diario).
3. **Reunión de finalización:** se consolidó el trabajo final y se verificó cada parte de la entrega, y se analizaron los puntos de mejora para el próximo proyecto. Tuvo una duración de 2 horas.

**Distribución de puntos:**

Basándonos en la contribución de cada miembro, la siguiente es la distribución de los 100 puntos del proyecto:

* **Andrés Pérez:** 33,3 puntos
* **Juana Morales:** 33,3 puntos
* **John Serrato:** 33,3 puntos

**Reflexión y puntos de mejora:**

En futuros proyectos, sería útil dividir más equitativamente el trabajo relacionado con la parte técnica y la generación de resultados, para evitar sobrecargar a ciertos miembros. Resaltando de igual forma, que para esta entrega se trató de mejorar significativamente ello y todo el trabajo fluyó de manera más acertada.