INTELIGENCIA DE NEGOCIOS ISIS3304
"Proyecto 1.2 Analítica de textos"
Profesor: Fabian Peña
Grupo 3
Daniela Uribe – d.uribe1
Valentina Calderón v.calderonm
Santiago Forero – s.forerog2

Parte 2

1. Proceso de automatización del proceso de preparación de datos, construcción del modelo, persistencia del modelo y acceso por medio de API:
2. Desarrollo de la aplicación y justificación

   2.1 Descripción del usuario/rol de la organización que va a utilizar la aplicación

   2.2 Conexión entre esa aplicación y el proceso de negocio que va a apoyar

   2.3 Importancia que tiene para ese rol la existencia de esta aplicación

   2.4 Ajustar tabla de actores

   2.5 Opciones al momento de definir y desarrollar la aplicación

   2.6 Aporte dado por la persona de estadística

   2.7 Evidencia de las mejoras con la interacción

3. Resultados
   3.1 Descripción y visualización en la aplicación de los resultados del modelo

   3.2 Simular la interacción del usuario final con la aplicación

   3.3 Describir dos acciones que puede realizar como resultado de dicha interacción

   3.4 Mostrar evidencias de las mejoras

   3.5 Revisar la validez de los resultados

4. Trabajo en equipo

   4.1 Roles

   4.2 Tareas de cada integrante

Parte 2

1. Proceso de automatización del proceso de preparación de datos, construcción del modelo, persistencia del modelo y acceso por medio de API:

El proceso de preparación de datos se realizó mayormente a nivel de la aplicación, por lo que para la automatización del proceso se trataron los datos como datos ya preparados previamente. Para la automatización del proceso se comenzó importando SKLearn para la creación del Pipeline que se usará en el proceso. Posteriormente se construye el pipeline que consta de dos etapas:

1. La vectorización de texto utilizando TF-IDF con tokenización, eliminación de palabras de parada (stop words) definidas previamente por lo construido por el científico de datos y conversión a minúsculas.

2. El clasificador de Bosque Aleatorio (Random Forest) con un valor de semilla aleatoria establecido en 4.

Después, se configura una búsqueda aleatoria de hiperparámetros para encontrar las mejores configuraciones de hiperparámetros para el pipeline. Incluyendo como algunos de sus parámetros:

- pipeline: El pipeline que se acabó de crear anteriormente, que incluye la vectorización y el clasificador.
- param_grid: Un diccionario que especifica las combinaciones de hiperparámetros que se probarán durante la búsqueda. Donde se tienen las combinaciones de vectorizador (CountVectorizer o TfidfVectorizer), si se convierte todo a minúscula o no, el número de estimadores del clasificador (50, 100), el criterion (entropy, gini) y la profundidad máxima (25, 50, 75, 100)
- n_iter: El número de iteraciones de búsqueda aleatoria.
- scoring: Las métricas que se utilizarán para evaluar el rendimiento del modelo, en este caso, precisión, recuperación y puntuación F1.
- refit: Indica que se seleccionará el modelo con la mejor puntuación F1.

A continuación, los datos se dividen en conjuntos de entrenamiento y prueba utilizando “train_test_split” con un porcentaje de entrenamiento del 30%. Luego, el modelo se ajusta a los datos de entrenamiento utilizando la búsqueda aleatoria.

De forma que el modelo se encuentra en la variable “pipe”, terminando con la construcción del modelo. Ahora para persistirlo primero, se guarda el modelo entrenado en un archivo llamado ‘model.joblib’ utilizando la librería ‘joblib’, facilitando así la reutilización del modelo previamente entrenado con “dump(pipe, ‘model.loblib’)

Para el acceso por medio de API, la aplicación implementa una API web utilizando el framework FastAPI para permitir el acceso al modelo creado. Los usuarios pueden interactuar con la API a través de dos mecanismos: un formulario en una página HTML o mediante la carga de archivos CSV. Para el caso del texto ingresado

manualmente en la interfaz, se usa la función “handle_form” donde los datos de entrada en forma de texto se obtienen del formulario HTML por el parámetro “cajita”. Después se crea un DataFrame de Pandas con la entrada de texto que se ingresa al pipeline previamente cargado “model.joblib”. Para el caso donde se ingresa un archivo csv se usa la función “predict_from_file” donde inicialmente se verifica que la extensión del archivo sí sea csv. En caso de que sí se crea un DataFrame de Pandas con el archivo csv que se ingresa al pipeline previamente cargado “model.joblib”. Cuando los datos se envían a la API, el modelo realiza predicciones y devuelve resultados en forma de mensaje o archivo CSV descargable (dependiendo del método en el que se ingresó la información). Esto lo hace con base en el ‘model.joblib’ recién creado que se guarda en una carpeta dentro de la aplicación web. Donde se obtienen los resultados por medio de:

```
pipeline_loaded = load("html/static/assets/model.joblib")
df["sdg"] = pipeline_loaded.predict(df["Textos_espanol"])
```

Donde se pueden realizar predicciones de lo que ingrese el usuario.

2. Desarrollo de la aplicación y justificación

2.1 Descripción del usuario/rol de la organización que va a utilizar la aplicación:

En un inicio, la aplicación se diseñó para brindar apoyo al Fondo de Población de las Naciones Unidas (UNFPA) mediante la creación de un modelo de clasificación basado en técnicas de aprendizaje automático. El objetivo primordial de este modelo es permitir la vinculación automática de texto con los Objetivos de Desarrollo Sostenible (ODS) y la aplicación tiene el propósito de simplificar la interacción con los resultados generados por dicho modelo.

Sin embargo, es importante destacar que la utilidad de la aplicación no se limita exclusivamente a UNFPA. Cualquier organización o individuo con interés en los ODS puede aprovecharla. Por ejemplo, la Organización de las Naciones Unidas (ONU) misma podría beneficiarse de esta herramienta para analizar el impacto de los objetivos en sus países miembros. Del mismo modo, gobiernos de diversas naciones también pueden utilizarla para evaluar cómo los ODS influyen en sus respectivos países. La versatilidad de la aplicación la convierte en una valiosa herramienta para múltiples actores interesados en el seguimiento y análisis de los Objetivos de Desarrollo Sostenible.

2.2 Conexión entre esa aplicación y el proceso de negocio que va a apoyar (si aplica)

La aplicación está estrechamente vinculada al proceso de negocio de planificación participativa para el desarrollo a nivel territorial. Su función principal es automatizar la clasificación de texto en relación con los Objetivos de Desarrollo Sostenible (ODS). Esta automatización agiliza el proceso y permite a UNFPA y otras organizaciones

analizar la información de manera más eficiente, mejorando así la toma de decisiones y el avance hacia el desarrollo sostenible.

2.3 Importancia que tiene para ese rol la existencia de esta aplicación

La aplicación beneficia en cuatro puntos clave: 1) Eficiencia: La aplicación automatiza la clasificación de texto según los ODS, lo que ahorra tiempo y recursos que de otro modo se invertirían en tareas manuales intensivas. 2) Precisión: La aplicación brinda resultados precisos y consistentes en la clasificación de texto, reduciendo el riesgo de errores humanos y garantizando la calidad de la información analizada. 3) Toma de decisiones informadas: Al facilitar la identificación rápida de cómo se relaciona la información con los ODS, la aplicación permite tomar decisiones más fundamentadas en la planificación participativa. 4) Versatilidad: La aplicación no se limita a una sola organización, lo que la hace valiosa para múltiples partes interesadas, como la ONU y gobiernos de diferentes países, ampliando su utilidad y alcance.

2.4 Ajustar tabla de actores

Rol dentro de la empresa	Tipo de actor	Beneficio	Riesgo
Fondo de Poblaciones de las Naciones Unidas (UNFPA)	Usuario-cliente	Facilita la evaluación y seguimiento de políticas públicas relacionadas con los ODS, permitiendo una toma de decisiones más informada y basada en datos, lo que mejora la eficacia de sus proyectos y actividades	Si el modelo de clasificación no funciona correctamente, podría llevar a interpretaciones erróneas de la información recopilada y decisiones inadecuadas basadas en esos datos
Entidades públicas y gubernamentales	Financiador	Obtienen información más rápida y precisa sobre cómo sus políticas impactan en la sociedad en relación con los ODS, lo que puede ayudar en la toma de decisiones más efectivas.	En caso de que el modelo no funcione es dinero mal invertido que pudo usarse mejor. Además, depender demasiado del modelo automatizado podría no capturar matices importantes o tener sesgos, lo que podría llevar a decisiones incorrectas.
Universidad de los Andes (entidad colaboradora)	Proveedor	Contribuye al desarrollo de tecnología y soluciones que tienen un impacto significativo en la mejora de políticas públicas y la calidad de vida de las comunidades locales.	Responsabilidad en la calidad y ética de los modelos de clasificación, ya que pueden afectar decisiones importantes en políticas públicas.

Habitantes locales	Beneficiado	Sus opiniones y preocupaciones se ven reflejadas en las políticas y acciones de desarrollo sostenible, lo que potencialmente podría mejorar su calidad de vida.	Vulneración de la privacidad y seguridad de los datos personales si la información se maneja de forma inadecuada.

2.5 Opciones al momento de definir y desarrollar la aplicación

Desde el inicio, nuestro objetivo principal fue desarrollar una aplicación que fuese accesible y amigable para todo tipo de usuario. Nuestra visión era que cualquier persona, sin importar su nivel de experiencia, pudiera beneficiarse de la aplicación de manera sencilla y sin complicaciones. Para lograr esto, diseñamos la aplicación de tal manera que los usuarios tuvieran dos opciones simples y de fácil uso: Cargar un archivo .csv o escribir directamente un texto. En ambos casos, la aplicación proporciona resultados precisos con la clasificación de los textos según los Objetivos de Desarrollo Sostenible (ODS).

2.6 Aporte dado por la persona de estadística

Su contribución se enfocó en el aspecto visual de la aplicación, sugiriéndonos la inclusión de los 12 Objetivos de Desarrollo Sostenible, con el propósito de proporcionar un contexto claro a los usuarios. También recomendó la incorporación de los símbolos de las Naciones Unidas. Además, expresó su satisfacción con la precisión del modelo, destacando que las métricas arrojadas eran realmente favorables.

2.7 Evidencia de las mejoras con la interacción

A continuación, se presenta la primera versión y la versión final de la aplicación, en la que se han considerado los comentarios realizados. Ahora podemos observar una aplicación que ofrece a los usuarios un contexto más claro y una experiencia visual más sencilla de utilizar.

3. Resultados

3.1 Descripción y visualización en la aplicación de los resultados del modelo

3.2 Simular la interacción del usuario final con la aplicación

3.3 Describir dos acciones que puede realizar como resultado de dicha interacción: forma como el resultado del modelo aporta en esas acciones

3.4 Mostrar evidencias de las mejoras

3.5 Revisar la validez de los resultados obtenidos con los modelos analíticos y garantizar que el uso que le están dando es correcto

El video se adjunta en la carpeta.

4. Trabajo en equipo

4.1 Roles

Líder del proyecto: Daniela Uribe

Ingeniero de datos: Santiago Forero

Ingeniero de software responsable del diseño de la aplicación y resultados: Valentina Calderón

Ingeniero de software responsable de desarrollar la aplicación final: Daniela Uribe

4.2 Tareas de cada integrante

Líder de proyecto: Daniela Uribe – Tiempo total: 2 horas

- Definir fechas reuniones
- Pre entregables
- Verificar asignaciones de tareas
- Subir entrega

Ing. de datos: Santiago Forero – Tiempo total: 3 horas

- Verificar la calidad del proceso de automatización relacionado con la construcción del modelo analítico.

Ing. de software responsable del diseño de la aplicación y resultados: Valentina Calderón - Tiempo total: 3 horas

- Liderar el diseño de la aplicación y de la generación del video con los resultados obtenidos.

Ing. de software responsable de desarrollar la aplicación final: Daniela Uribe – Tiempo total: 3 horas

- Gestionar el proceso de construcción de la aplicación.

4.3 Reuniones

- Reunión de lanzamiento y planeación: Realizada el 19 de octubre

- Reuniones de seguimiento: Realizada el 21 de octubre
- Reunión de finalización: Realizada el 23 de octubre

4.4 Retos

- Elección de herramienta para desarrollar la presentación de resultados, debido a que había muchas opciones posibles que traían distintas ventajas y desventajas. Se decidió la implementación final debido a su facilidad de implementar y facilidad a modificaciones futuras.
- Conexión del modelo con API de forma que los usuarios puedan usar el modelo de forma fácil e intuitiva. Se decidió una interfaz donde se puedan ingresar tanto archivos csv como texto plano con el fin de facilitar el uso por los usuarios potenciales.
- Manejo de carga de archivos csv a un navegador. Se usaron clases de forms y file para permitir que el usuario pudiese subir un documento de su elección al oprimir un botón.

4.5 Cómo mejorar para la siguiente entrega:

- Darle más prioridad al enunciado desde el inicio para tenerlo en cuenta durante todo el desarrollo del proyecto
- Comunicarnos activamente entre nosotros más allá de las reuniones en casos de necesidad
- Apoyarnos más en profesores y monitores para algunas preguntas difíciles

4.6 Repartición 100 puntos

Daniela Uribe: 34 puntos Valentina Calderón: 33 puntos Santiago Forero: 33 puntos