Universidad de los Andes **Universidad de los Andes** **Inteligencia de negocios**

Proyecto 1

Etapa 2

Samuel Freire – 202111460

Juan Felipe Garcia – 202014961

Lucciano Franco Márquez – 202111458

Contenido

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 automatización del flujo de trabajo se inicia con la definición de una estructura de datos estandarizada a través de la clase DataModel. Esta clase establece un modelo de datos uniforme para las solicitudes al API, garantizando que los datos de entrada cumplan con un formato específico. Una vez que los datos llegan a la API, se desencadena un proceso de transformación de texto mediante la clase TransformacionesPD, que se encarga de limpiar y normalizar el texto, incluyendo la eliminación de caracteres no ASCII, conversión a minúsculas, eliminación de puntuación y stopwords en español, entre otras operaciones.

El siguiente paso crucial es la construcción del modelo de procesamiento de lenguaje natural mediante la clase PrediccionesModel. Este modelo se carga desde un archivo Joblib y se utiliza para realizar predicciones basadas en el texto de entrada. El pipeline de procesamiento de texto, que incluye el algoritmo de vectorización y un algoritmo de clasificación, se aplica para clasificar el texto según los Objetivos de Desarrollo Sostenible (SDG). La automatización de este proceso garantiza la consistencia en la predicción de resultados y acelera el flujo de trabajo.

La persistencia del modelo en un archivo .joblib asegura que el modelo se pueda cargar de manera eficiente cada vez que se realice una predicción, minimizando el tiempo de inicio. Finalmente, el acceso al sistema se facilita a través de un API desarrollado con FastAPI. La ruta /predict permite a los usuarios enviar datos en formato JSON, que se someten al proceso automatizado de preparación y predicción. El resultado se devuelve como una respuesta en formato JSON, brindando acceso rápido y sencillo a las predicciones basadas en el modelo de NLP. Este enfoque de automatización del flujo de trabajo garantiza eficiencia, coherencia y accesibilidad en todo el proceso de procesamiento de texto y predicción de resultados.

A continuación, se detallará lo mencionado anteriormente:

Pipeline:

El .joblib contenido en assets viene siendo la pipeline desarrollada en la parte del notebook de proyecto 1, esta pipeline lo que realiza es la ejecución de los algoritmos de vectorización y clasificación, para que de este modo una vez que se le dé al botón de cargar pipeline en la interfaz gráfica este utilice el modelo decidido para realizar las predicciones, de este modo la pipeline es creada del mismo modo que la persistencia de este modelo por lo que ahora cualquier Excel que tenga el objetivo de predecir la categoría de los objetivos puede ser utilizado a partir de esta pipeline creada. Esta Pipeline es fundamental en la creación de la página web y el funcionamiento del código ya que a partir de esta se pueden automatizar varios procesos.

API:

Las tres clases que se mencionaran a continuación son la base para la creación de la API que estará conectada y será la base para la página web que servirá como interfaz grafica para realizar las predicciones:

Clase DataModel:

La clase DataModel es parte esencial del sistema, ya que define la estructura y el formato de los datos que se esperan recibir en las solicitudes POST del API. En su interior, se declara una única variable miembro, Textos_espanol, que representa el texto en español proporcionado en la solicitud. Esta clase actúa como una plantilla o modelo para estandarizar la información de entrada, asegurando que todos los JSON recibidos cumplan con el formato predefinido. La simplicidad de esta clase y su enfoque en un único campo facilitan el manejo coherente de los datos a lo largo de las peticiones, lo que es fundamental para el procesamiento exitoso en el sistema.

Clase PrediccionesModel:

La clase PrediccionesModel desempeña un papel fundamental en el sistema al encargarse de la realización de predicciones basadas en el procesamiento de lenguaje natural. En su constructor, carga un pipeline de procesamiento previamente entrenada desde un archivo Joblib que es la pipeline mencionado anteriormente (tratamientoD.joblib). El método clave, make_predictions, aplica este pipeline a los datos de entrada. En primer lugar, realiza transformaciones del algoritmo de vectorización sobre el texto proporcionado. Luego, utiliza un algoritmo de clasificación para predecir las etiquetas de los Objetivos de Desarrollo Sostenible (SDG) relacionadas con el texto. Finalmente, agrega las predicciones como una nueva columna en el conjunto de datos de entrada y devuelve los resultados. Esta clase encapsula la lógica de predicción y la carga del modelo, permitiendo una integración sencilla en el sistema de API para generar predicciones precisas y eficientes.

Clase TransformacionesPD:

Su función principal es aplicar diversas transformaciones al texto de entrada antes de que se realicen las predicciones. La clase contiene una serie de métodos para llevar a cabo estas transformaciones, incluyendo la eliminación de caracteres no ASCII, la conversión del texto a minúsculas, la eliminación de puntuación, la sustitución de números por representaciones en palabras y la eliminación de palabras vacías o "stopwords" en español. Estas transformaciones tienen como objetivo limpiar y estandarizar el texto, lo que facilita el procesamiento por parte del modelo de procesamiento de lenguaje natural prácticamente es el tratamiento de datos realizado en la etapa 1.

El método preprocessing de la clase aplica todas estas transformaciones en secuencia al texto de entrada. Además, la clase implementa las interfaces BaseEstimator y TransformerMixin, lo que la hace compatible con pipelines de transformación de datos. En resumen, TransformacionesPD realiza la preparación de los datos de texto para su posterior procesamiento y predicción, asegurando que los datos estén en un formato adecuado y limpio para obtener resultados precisos.

Main:

El "main" está destinado a la creación de un API utilizando la biblioteca FastAPI. La API define una única ruta, /predict, que acepta solicitudes POST que contienen datos en formato JSON. La función asociada a esta ruta, make_predictions, procesa las solicitudes, aplica transformaciones al texto de entrada mediante la clase TransformacionesPD, y posteriormente realiza predicciones utilizando la clase Model, que carga un modelo de procesamiento de lenguaje natural desde un archivo Joblib. Finalmente, las predicciones se devuelven como una respuesta en formato JSON para su consumo. Este Main constituye la base del servicio web para la realización de predicciones basadas en los objetivos proporcionados por el excel.

2. Desarrollo aplicación y justificación

Antes de empezar a definir los objetivos, limitaciones y proceso que se siguió para lograr mejorar la aplicación. Se va a definir cómo se puede ejecutar y como se puede ver en funcionamiento para cualquier usuario que lo requiera. Este tutorial, va a ser muy breve, dado que se buscó que la aplicación fuera lo más accesible posible. Esto con la idea que cualquier usuario que en algún momento necesite la aplicación, puede que se describa a continuación o no, pero se busca que todos puedan usar la aplicación de alguna manera. Luego de haber realizado este aviso, vamos a empezar con el proceso para ejecutar la aplicación. Cabe aclarar que este tutorial esta realizado en VisualStudio code, por lo que alguna duda escribir al grupo:

Para empezar, es necesario acceder al repositorio que contiene el proyecto: https://github.com/jf-garciam1/Lab-BI.git . Luego, se clona el proyecto en el entorno de ejecución. Dentro de este repositorio es necesario verificar si se está en la rama llamada "ProyectoTerminado", de no ser el caso ingresar a esta rama según el entorno de ejecución. Luego dentro de este, realizar el siguiente comando en la línea de comandos del computador (**Windows + r**, luego escribir **cmd** y dar enter): **pip install -r requirements.txt**. Luego de realizar esto, es necesario hacer el comando, en la línea de comandos del entorno de ejecución, ejecutar la siguiente lista de comandos: **cd Proyecto1, cd Entrega2, cd Entrega2, cd API**. En este punto realizar en esa misma línea de comandos el siguiente comando: **python -m uvicorn principal:app –reload.** Ya en este punto debería aparecer algo parecido a la siguiente imagen:

```
PS C:\Users\Samue\Lab2\Lab-BI> cd Proyecto1
PS C:\Users\Samue\Lab2\Lab-BI\Proyecto1> cd Entrega2
PS C:\Users\Samue\Lab2\Lab-BI\Proyecto1\Entrega2> cd Entrega2
PS C:\Users\Samue\Lab2\Lab-BI\Proyecto1\Entrega2\Entrega2> cd API
PS C:\Users\Samue\Lab2\Lab-BI\Proyecto1\Entrega2\Entrega2\API> python -m uvicorn principal:app --reload
INFO:     Will watch for changes in these directories: ['C:\\Users\\Samue\\Lab2\\Lab-BI\\Proyecto1\\Entrega2\\Entrega2\\API']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [11996] using WatchFiles
INFO:     Started server process [23296]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

Imagen 1. Resultado ejecución API

Luego, de esto vamos a buscar el archivo llamado PaginaWeb.py. Este archivo se encuentra en la ruta LAB-BI/Proyecto1/Entrega2/Entrega2/PaginaWebBi. En este caso, lo único a realizar es ejecutar el archivo python, si se encuentra en visual darla al botón de ejecutar. Luego de realizar esto, debería estar viendo algo como esto:

```
PS C:\Users\Samue\Lab2\Lab-BI> & C:/Users/Samue/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Samue/Lab2/Lab-BI/Proyecto1/Entrega2/Entrega2/PaginaWebBi/PaginaWeb.py
 * Serving Flask app 'PaginaWeb'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 991-983-310
```

Imagen 2. Resultado ejecución Sitio Web

En este caso, debería tener dos terminales ejecutándose a la vez, es decir, por separado una de la otra.

Ahora bien, luego de realizar esto, va a copiar la url, que le aparece cuando ejecuta el archivo PaginaWeb.py. La cual se puede ver en la imagen 2, que dice "Running on http://127.0.0.1:5000" esta es la url, según le salga a usted, la que tiene que copiar en su navegador para lograr ver la página web.

Ahora bien, apenas, realiza esto va a estar viendo el sitio web, el cual se tiene que ver así:

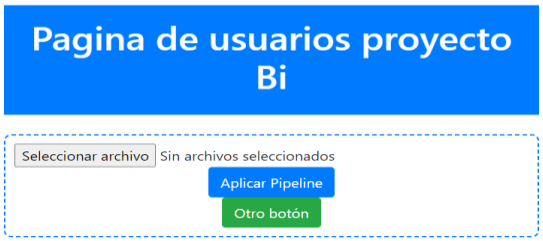

Imagen 3. Sitio Web

En esta pantalla, que es la principal, se pueden tres botones. El primero es el de seleccionar archivo, el cual va a ser útil, para que el usuario pueda subir, el archivo xlsx, el cual quiere predecir los objetivos de desarrollo sostenible. Luego, el segundo botón es el de aplicar Pipeline. Este va a ser útil, para luego de cargar el archivo, se realice la ejecución del proceso para calcular los resultados. Luego, el tercer botón que dice otro botón, esta para un contraste de color. Luego, de cargar el archivo y darle en el botón de aplicar pipeline, debería ver algo similar a lo siguiente:

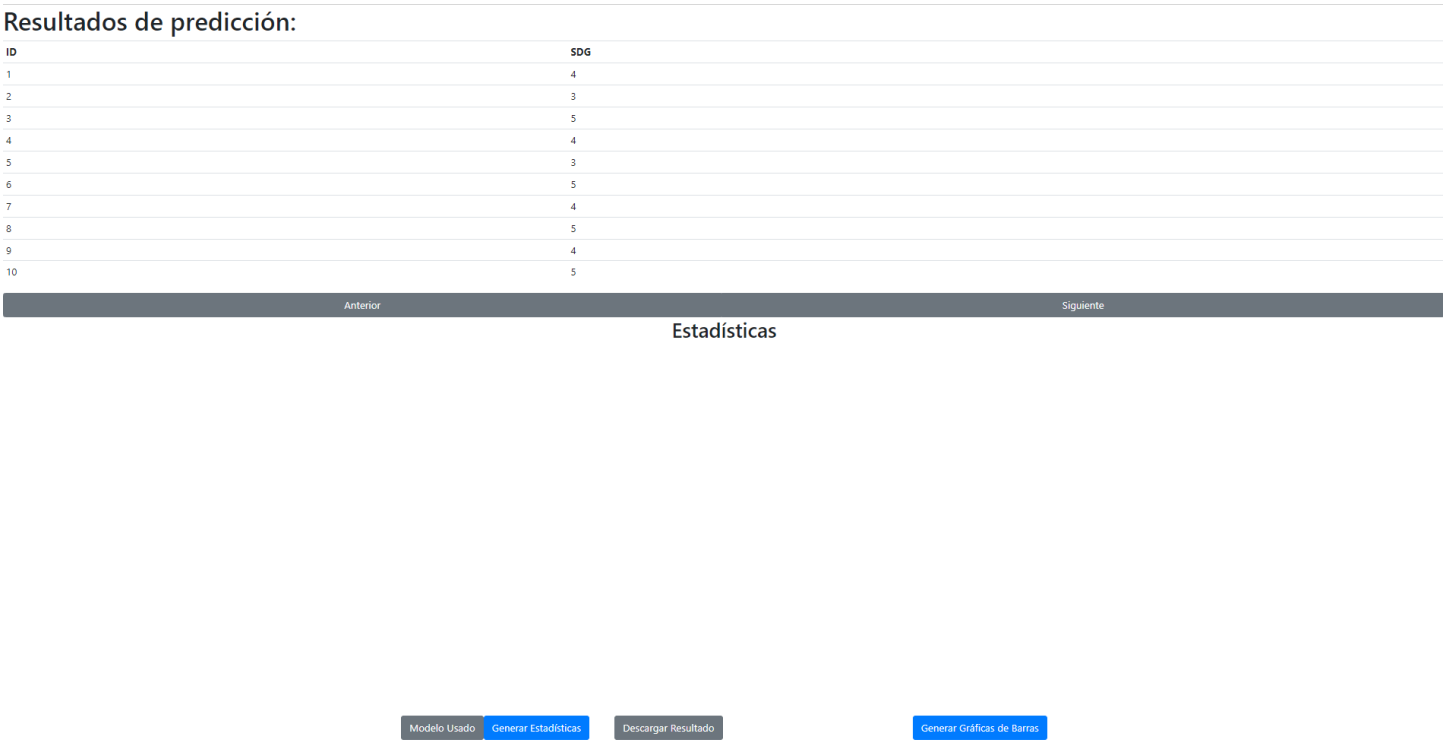

Imagen 4. Resultado cargar archivo

Lo primero que ve el usuario luego de cargar el archivo y darle a aplicar pipeline, es la imagen anterior. Donde tenemos dos zonas realmente. La primera es la zona donde se presenta una tabla, donde se enumeran los textos entregados, del 1 a la cantidad que hubo. Además, de el objetivo de desarrollo sostenible asignado a ese texto. Se tiene dos botones: Anterior y Siguiente. Estos botones sirven para navegar dentro de todos los datos o textos que fueron analizados. Es decir, pasamos de la 1-10 a la 11-20 si se dé al siguiente y si le damos al anterior pasa del 11-20 al 1-10. Luego, tenemos la segunda zona con 4 botones. El primer de ellos es el llamado: “ Modelo Usado”. El cual como su nombre indica nos va a mostrar información sobre el modelo utilizado en esta predicción. Principalmente sus métricas y resultados de estas. Luego, el siguiente botón llamado:”"Generar Estadísticas”, nos genera, como su nombre indica, estadísticas de los datos, principalmente el objetivo de desarrollo sostenible con más apariciones, y con cuanta frecuencia apareció. Luego, el siguiente botón llamado: “Descargar resultados”, le va a permitir al usuario descargar en formato .csv los resultados los datos que se obtuvieron. Luego, el ultimo botón llamado: “Generar Grafica de barras”. Le va a ser útil para generar un gráfico de barras, donde se muestran la cantidad de datos con los objetivos de desarrollo sostenible. Luego, como dato adicional si clicka sobre cualquier fila de la tabla superior, va a aparecer una caja con información relevante sobre el objetivo de desarrollo sostenible de esa fila.

3. Mapa de actores relacionado con un producto de datos creado con el modelo analítico construido (Actualizado)

Rol dentro de la empresa	Tipo de actor	Beneficio	Riesgo
UNFPA	Cliente y usuario	Generación de iniciativas coherentes y efectivas sobre las problemáticas encontradas	Puede generar estrategias que no tengan un sentido lógico, puede estar clasificando problemas en áreas o en objetivos que no son.
Personas (victimas)	Beneficiado	Recibe soluciones a tiempo, adaptadas a su contexto	Pérdida de confianza en la institución, o en cualquier organismo publico
ONU	Cliente y usuario	Apoyar labores sociales preventivas de problemáticas	Puede generar una confusión, en cuanto, a como está el estado actual, como van las iniciativas generadas…
Departamento de servicios de tecnologías de UNFPA	Proveedor	Garantiza la generación de un sistema apto para la toma de decisión y evaluación de iniciativas	Generación incorrecta de análisis y resultados erróneos. Además, de la afectación sobre los datos
Departamento de analítica UNFPA	Financiador	Automatización de la toma de decisiones, con base en opiniones de las víctimas.	Puede ser un proyecto fallido, en el cual se pierde dinero y recursos para no generar el resultado esperado.
Entidades publicas	Cliente y usuario	Generación de pautas para generar soluciones coherentes y efectivas	Pueden no estar al tanto del manejo de información, amenaza de datos, y falta de entendimiento con respecto al formato del proyecto y su objetivo
Investigadores	Cliente y usuario	Análisis de datos para generación de reportes generales de interés	Generación de pautas para soluciones coherentes y efectivas: El riesgo aquí es que las entidades públicas pueden no tener una comprensión completa de lo que necesitan o cómo deben definirse las pautas. Esto podría llevar a un enfoque inadecuado o la generación de pautas poco efectivas.
Analista de datos	Consultor o asesor externo	Los consultores externos suelen aportar experiencia y conocimientos especializados en áreas específicas, lo que puede enriquecer la toma de decisiones y la ejecución de proyectos.	Dependencia excesiva de consultores externos puede llevar a la falta de desarrollo de conocimiento y experiencia interna en la empresa. A largo plazo, esto puede ser perjudicial.
Miembro de la junta directiva	Financiador y proveedor	Los miembros de la junta directiva tienen la capacidad de influir en la dirección estratégica de la empresa. Pueden contribuir a la formulación de políticas, la planificación estratégica y la toma de decisiones clave.	Los miembros de la junta tienen una responsabilidad legal y fiduciaria para actuar en el mejor interés de la empresa y sus accionistas. Esto significa que pueden ser personalmente responsables en caso de mala gestión, conflictos de interés o violaciones de la ley.

4. Resultados

Los resultados del proceso de análisis se comentarán en un vídeo que se publicará en el Padlet especificado. El vídeo proporcionará una experiencia de interacción detallada y fácil de entender entre el usuario y la aplicación desarrollada. Esta representación visual le permite ver de forma clara y precisa cómo los usuarios navegan y utilizan la aplicación para ingresar sus solicitudes en el modelo analítico. Este video demostrará efectivamente cómo la aplicación responde y muestra los resultados que produce el modelo, brindando una comprensión integral del rendimiento y la usabilidad de la solución de análisis implementada en el proyecto. Esta demostración audiovisual servirá como una herramienta valiosa para todas las partes interesadas y miembros del equipo, permitiéndoles comprender de manera integral y detallada las características y beneficios de la aplicación desarrollada. Además, el video no solo se limitará a presentar la funcionalidad de la aplicación y los resultados del modelo analítico, sino que también destacará su impacto en el contexto del proyecto y la organización en su conjunto. Los comentarios y testimonios de los usuarios, junto con métricas de rendimiento clave, ayudarán a ilustrar cómo la aplicación ha mejorado la eficiencia, la toma de decisiones y la satisfacción de los usuarios. Este enfoque integral proporcionará una visión completa de la aplicación y su relevancia para el éxito del proyecto. Al brindar una representación visual detallada y comprensible, el video se convierte en una herramienta esencial para comunicar el valor de la solución de análisis implementada a todas las partes interesadas, asegurando una comprensión profunda de sus características y beneficios.

5. Trabajo en equipo

Integrante del Grupo	Rol	Tareas Realizadas	Tiempo Dedicado (horas)	Retos Enfrentados	Formas de Resolver los Retos
Lucciano Franco	Líder de Proyecto	Gestión y coordinación del proyecto.	15	Coordinar las reuniones de seguimiento.	Establecimos un calendario fijo y utilizamos herramientas de planificación
		Definición de fechas de reuniones y entregables.		Toma de decisiones en situaciones de desacuerdo.	Facilitar espacios para la expresión de opiniones y consensuar decisiones cruciales.
		Verificación de asignaciones.			
Felipe García	Líder de Datos	Garantizar calidad de automatización.	20	Optimizar el proceso de automatización	Revisamos y mejoramos el código para aumentar la eficiencia.
		Validación de resultados analíticos.			
Samuel Freire	Líder de Analítica	Liderar diseño de la aplicación y video.	18	Integrar de manera efectiva los resultados en el diseño.	Realizamos pruebas iterativas para asegurarnos de que

					la interfaz sea intuitiva y funcional.
		Gestionar construcción de la aplicación		Optimizar el rendimiento de la aplicación.	Identificamos y corregimos cuellos de botella en el código.

Distribucion:

- Samuel Freire: 35 puntos
- Felipe García: 35 puntos
- Lucciano Franco: 30 puntos

Reuniones:

- Reunión de Lanzamiento y Planeación: Definición de la organización/empresa/institución beneficiada y sus roles. Acuerdo sobre la forma de trabajo, momentos de reunión y canal de comunicación. Se insta a contactar al estudiante de estadística asignado lo antes posible para asegurar el éxito del proyecto.
- Reuniones de Seguimiento: Se realizan al menos dos reuniones semanales cortas o comunicación vía correo. Se utilizan herramientas como Trello para mantener un tablero de control de tareas y actualizaciones.
- Reunión de Finalización: Consolidación del trabajo final y revisión del desempeño del grupo. Análisis de puntos a mejorar para la próxima etapa del proyecto.