Inteligencia de Negocios – ISIS 3301
Departamento de Ingeniería de Sistemas y Computación

Entrega 2: Automatización de analítica de Textos

Grupo 37

Antonin Bouillaud – 202325830

Ernesto José Duarte Mantilla – 202014279

Lina María Gómez Mesa – 201923531

Universidad de Los Andes

Octubre 15 2023

Contenido

Introducción

En el presente trabajo se llevará a cabo una propuesta de clasificación textual de acuerdo con los objetivos 3, 4 y 5 del Desarrollo Sostenible (ODS) como colaboración entre la Universidad de los Andes y la UNFPA. Este texto tiene como objetivo, en primera instancia, explicar la elaboración un análisis automatizado de opiniones que representan la voz de los habitantes locales sobre problemáticas de su entorno particular mediante el uso de un modelo de clasificación de textos supervisado. Para ello, se llevará a cabo un entendimiento preliminar de los datos recopilados, teniendo en cuenta características generales de los datos como idioma, unicidad, duplicidad y completitud en relación con los objetivos y criterios de éxito desde el punto de vista del negocio. A continuación, se realizará la limpieza de los datos teniendo en cuenta el análisis exploratorio inicial para luego presentar los modelos seleccionados (Random Forest, KNN, Multinomial Naive Bayes y Ridge Classifier). Posteriormente, se presentarán los resultados cuantitativos obtenidos y las recomendaciones preliminares para la organización. Finalmente, se describirá el mejor modelo y conclusiones. Todos estos pasos tienen como finalidad determinar oportunidades de éxito para el negocio con el fin de automatizar el proceso de clasificación de textos de acuerdo con los objetivos de desarrollo sostenible seleccionados.

1. Entendimiento del Negocio Y Enfoque Analítico

Los objetivos de desarrollo sostenible fueron adoptados por las Naciones Unidas en el 2015 como un llamado universal para erradicar la pobreza, proteger al planeta y que para en el 2030 se disfrute de mayor prosperidad a nivel mundial [4]. En total son 17 metas que se centran en un área particular para alcanzar tres dimensiones del desarrollo: ámbito económico, social y ambiental. En particular, los objetivos 3,4 y 5 se centran en [5]:

- Salud Y Bienestar (3): Garantizar una vida sana y promover el bienestar de todos a todas las edades. Intenta combatir las desigualdades en acceso a asistencia sanitaria.
- Educación de Calidad (4): Garantizar una educación inclusiva y equitativa de calidad y promover oportunidades de aprendizaje permanentes para todos. Intenta fomentar la tolerancia entre personas y contribuir a sociedades más pacíficas.
- Igualdad de Género (5): Lograr igualdad de género y empoderar a todas las mujeres y las niñas. Intenta ir en contra los prejuicios y las asociaciones implícitas a menudo invisible para la igualdad de oportunidades.

En Colombia, estos objetivos son de suma importancia dado que ayudan a la erradicación de la pobreza, reducción de la mortalidad infantil y mejoramiento de la cobertura en educación [6]. Además, sirven como indicadores de medición del progreso del país hacia el desarrollo y como herramienta para la integración y la coherencia entre las agendas y las políticas públicas. El objetivo de salud (3) se centra especialmente en reducir la mortalidad materna a 32 por cada 100 mil nacidos vivos, acabar con las muertes prevenibles de menores de 5 años, luchar contra enfermedades transmisibles (malaria, VIH, enfermedades tropicales) entre otras. La educación se centra en aumentar la cobertura de educación superior pasando de 49,4% en 2015 a 80,0% en 2030, educación básica y media gratuita, aumentar el número de personas con habilidades relevantes para el éxito financiero, entre otros. Finalmente, el objetivo de igualdad de género se centra en que las mujeres ocuparán el 50% los cargos decisorios dentro del Estado Colombiano, acceso universal a los derechos y salud reproductiva y hacer cumplir la legislación que promueve la igualdad de género.

A. Descripción del enfoque analítico

Uno de los enfoques analíticos que puede ayudar a alcanzar los objetivos del negocio es implementar un sistema de clasificación de textos que permita clasificar los textos de los habitantes locales de acuerdo con los objetivos de desarrollo sostenible 3,4 y 5 de forma automática. Esto se realizará utilizando técnicas de procesamiento de lenguaje natural y aprendizaje automático para entrenar el modelo con un conjunto de datos etiquetados. Una vez implementado el modelo, se utilizarán las métricas de evaluación detalladas en la anterior sección para medir el rendimiento del modelo y ajustarlo en caso de ser necesario.

B. Objetivos del Negocio

Tabla 1. Principales Objetivos del Negocio

Oportunidad/problema Negocio	La UNFPA busca clasificar automáticamente el texto en las categorías de los ODS 3, 4 y 5 para identificar problemas y evaluar soluciones en la información textual recopilada, reduciendo así el esfuerzo y el uso de recursos en la organización.
Enfoque analítico (Descripción del requerimiento desde el punto de vista de aprendizaje automático) e incluya las técnicas y algoritmos que propone utilizar	Implementar un modelo de clasificación de textos utilizando técnicas de procesamiento de lenguaje natural y aprendizaje supervisado para entrenar y evaluar el modelo con un conjunto de datos de textos etiquetados con los ODS 3, 4 y 5.
Organización y rol dentro de ella que se beneficia con la oportunidad definida	La UNFPA es la organización que se beneficia del proyecto ya que es la encargada de “identificar problemas y evaluar soluciones actuales, relacionando la información con los diferentes ODS”. Asimismo, dentro de la organización, los roles que se benefician de este proyecto son los (usuarios) que son participantes internos de la organización como los directores ejecutivos y coordinadores nacionales y asesores de asuntos políticos e intergubernamentales quienes son los encargados de plantear las políticas públicas y planes de mitigación para llegar a los ODS en el 2030 [1,2,3]. Los ciudadanos (habitantes locales) también se beneficiarían de manera indirecta ya que se identificarían de manera más rápida y eficaz las problemáticas de su entorno.
Contacto con experto externo al proyecto	Juan José Suárez jj.suarezm1@uniandes.edu.co Canal Seleccionado: Correo y reunión presencial

2. Automatización del Proceso

A. Preparación de los Datos

Inicialmente se realizaron tres tipos de modificaciones a los datos de entrada: 1) limpieza de los datos, 2) tokenización y 3) normalización. Dentro de la limpieza de los datos se consideró que una posible modificación era eliminar las columnas que no se encontraban en español; es decir, inglés y francés. No obstante, luego de comparar resultados obtenidos entre los modelos con y sin columnas no se evidenció una diferencia significativa entre las métricas (f_1, precisión y recall) por lo que no se realizó.

En la segunda etapa, se eliminaron los símbolos de puntuación o comillas para evitar que este tipo de caracteres interfirieran en la interpretación de palabras por los algoritmos. Luego, se eliminaron los caracteres que no fueran tipo ASCII, se pasaron todos los caracteres a minúscula y se pasaron todos los dígitos o números en su representación de texto en español. Posteriormente, se eliminaron las palabras tipo *stopwords* que no aportan significado al problema de clasificación (a, el, lo... etc).

Finalmente, en la tercera etapa en la normalización de los datos, se realiza la eliminación de prefijos y sufijos, además de realizar una lematización. Esto da como resultado los datos limpios listos para ser convertidos a valores numéricos por uno de los siguientes dos métodos que se describen a continuación:

- **TF-IDF:** Esta es una técnica de procesamiento de lenguaje natural en la que se calcula la importancia de cada uno de los términos en un texto analizando la frecuencia con la que aparecen en el texto. Este considera tanto la frecuencia de una palabra en un documento como la frecuencia de la misma palabra en todo el texto, lo que ayuda a reducir la importancia de palabras comunes. Se eligió esta técnica ya que permite que el modelo sea sensible a las palabras que son más importantes para cada clase.

B. Clasificador Final Seleccionado

Se decidió utilizar un modelo de clasificación lineal como *Ridge Classifier*. Este se caracteriza por ser un modelo lineal discriminativo que penaliza los coeficientes del modelo para evitar el sobreajuste a los datos. Por lo tanto, agrega un término de penalización a la función de costo que usualmente es la suma de los coeficientes al cuadrado de los *features.* Una penalización mayor resulta en una mayor regularización y en valores de coeficientes más pequeños lo cual funciona cuando hay pocos datos como es este caso. Este clasificador tiene un enfoque one-vs- all para problemas de multi-clasificación.

Justificación: Se eligió este clasificador, en particular, ya que como se discutió evita el sobreajuste, penaliza los *features* irrelevantes asignándoles un menor coeficiente y es un método inherentemente interpretable; por lo que, le puede dar indicios al negocio de porque el modelo está clasificando las cosas de cierta manera. Asimismo, obtuvo mejores métricas (f-1, accuracy) de 0.98.

C. Construcción del Pipeline

Para construir nuestro Pipeline, decidimos mantener el preprocesamiento por separado y enfocarnos en la etapa de vectorización junto con el modelo.

En otras palabras, diseñamos el pipeline de manera que la entrada ya se encuentre preprocesada y lista para ser vectorizada y evaluada por el modelo seleccionado, el Ridge Classifier,

que demostró ser la mejor opción en términos de rendimiento en nuestros datos de prueba. Esta elección nos permite una mayor flexibilidad y modularidad en nuestro flujo de trabajo.

El preprocesamiento, que incluye pasos como la limpieza de texto, eliminación de palabras de enlace, puntuación y lematización, se lleva a cabo antes de que los datos ingresen al pipeline. Luego, en el pipeline, nos enfocamos únicamente en la transformación de estos datos preprocesados en características numéricas a través de la vectorización y en la predicción de la clase correspondiente con el modelo Ridge Classifier. Esta separación de responsabilidades hace que nuestro código sea más eficiente y mantenible, ya que permite cambios o mejoras específicas en cada etapa del proceso sin afectar a las demás.

Para configurar este pipeline, comenzamos por reutilizar el conjunto de datos que ya había sido preprocesado, lo que nos permitió volver a entrenar el TF-IDF Vectorizer y crear una instancia optimizada que guardamos en el pipeline. Simultáneamente, extrajimos el mejor modelo previamente entrenado con la biblioteca Joblib. Finalmente, ensamblamos el vectorizador y el modelo en el pipeline. Para garantizar un uso eficiente en el futuro, hemos guardado todo el pipeline con la biblioteca Joblib, lo que facilitará su integración en nuestra API.

D. Persistencia del Modelo Y Acceso por medio de API

A continuación, en la figura 1 y 2 se muestra los modelos resumidos del pipeline final implementado dentro de la aplicación.

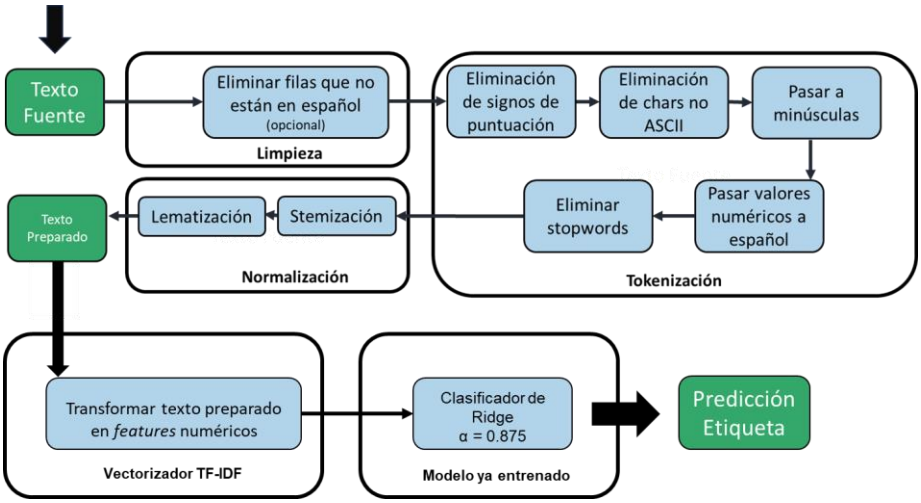

Figura 1. Gráfica Específica del Pipeline

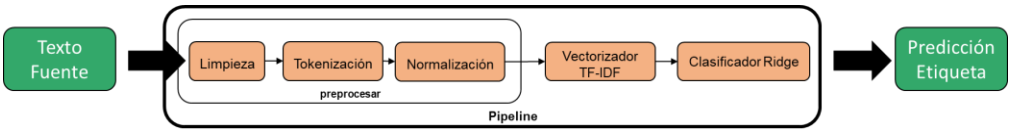

Figura 2. Gráfica General del Pipeline

Este proceso se convirtió dentro de la aplicación en los archivos clean.py. El cual es un pipeline para poder estandarizar y automatizar el proceso de limpieza y transformación de los datos utilizando las clases Clean, TextVectorizer y Pipeline. Clean se encargó del preprocesamiento que se observa en la figura 2 en la sección “preprocesar”, TextVectorizer se encargó de pasar a features_numéricos, y el Pipeline se encargó de juntar todas las clases y convertirlo en un pipeline funcional en el que se pudiera

hacer predict para cualquier dato de entrada en formato dataframe. Se conectó a la API a través de el archivo llamado main.py donde se crea un instancia del modelo y luego en el archivo PredictionModel.py se serializa una instancia de la clase Pipeline la cuál ayuda a predecir todos los textos de entrada del usuario. El código se encuentra en el repositorio de Github el cuál fue adjunto también como link en esta entrega.

3. Desarrollo de la Aplicación y Justificación

A. Descripción del Backend Y FrontEnd

Para el back-end de la aplicación, se decidió utilizar Python con el framework FastAPI. Este framework es conocido por su facilidad de uso, velocidad de ejecución y robustez. FastAPI utiliza el servidor web Uvicorn para ejecutar y exponer el API. La aplicación puede recibir archivos .XLSX que contienen textos de los habitantes en su mayoría en español o un único *input* del usuario de texto. Luego, utiliza un modelo de aprendizaje automático previamente entrenado con los datos etiquetados de los ODS (que se encuentra serializado en Prediction_Model.py y está toda la lógica en clean.py) para predecir si cada texto pertenece al ODS 3, 4 o 5. Las dos principales funcionalidades de la aplicación fueron: **1)** Clasificar un único texto ingresado por un usuario y **2)** Clasificar un archivo excel (.xlsx) con varios textos ingresado por un usuario.

A continuación, se presentan los endpoints utilizados por la aplicación:

- (/): Este primer end-point muestra la página principal de la aplicación. Inicialmente, se muestra una descripción de los ODS, una descripción de cada uno de los ODS seleccionados (3,4 y 5). Y las dos funcionalidades descritas en el párrafo anterior.

- (/predict-text) En este segundo end-point le ingresa un único texto escrito por el usuario y se guarda en formato .XLSX. Además, se despliega la información en la parte web de la aplicación. Este utiliza el método HTTP POST y recibe un dataframe de 1 única fila y predice el resultado. Le muestra al usuario en pantalla la clasificación según el ODS.

- (/predict-file): En el tercer end-point se le ingresa un archivo de .XLSX y para cada línea del archivo Excel el modelo predice los resultados y luego le muestra al usuario dos gráficas. La primera es un pie chart con la cantidad de datos por cada clase y la segunda es un wordcloud con las palabras que fueron más representativas por cada clase. Este endopoint utiliza el método HTTP POST y recibe como entrada un archivo UploadFile que contiene múltiples textos de los habitantes en español. El archivo se lee en memoria, se transforma en un objeto pandas DataFrame y luego se utiliza el modelo para hacer predicciones en el DataFrame completo.

En la parte del front-end se decidió utilizar JinjaX el cual despliega templates. Estos templates contienen html, css y javascript lo cual los hace altamente modificables y editables para futuras iteraciones en caso de quererlas modificar. Únicamente se tienen dos templates: 1) el de la página inicial y 2) el de el despliegue de las gráficas.

B. Persistencia

Como persistencia, la aplicación tiene 3 formas de persistencia. La primera se centra en guardar todo input que el usuario mete en un .log el cual se encuentra en la carpeta Project. En este se guarda cada

input y cada proceso por el que pasa un texto. Así, en caso de que haya un fallo, se puede ver en qué paso del proceso se detuvo la limpieza, vectorización o clasificación del texto. En segundo lugar, la aplicación le permite al usuario descargar sus datos en formato XLSX. Esto lo hace mucho más fácil dado que lo puede utilizar luego para hacer sus propios análisis o tableros de control. Finalmente, también puede descargar las imágenes de las gráficas en casos de que las quiera utilizar.

C. Descripción del usuario/rol de la organización

A continuación, en la tabla 2 se muestra el mapa de actores que se pueden ver beneficiados con la aplicación.

Tabla 2. Principales Objetivos del Negocio

Rol dentro de la Empresa	Tipo de Actor	Beneficio	Riesgo
Asesor y Coordinador de Programas Locales del País	Usuario-cliente	Este actor trabaja para garantizar la cohesión en la ejecución de los distintos programas financiados. Puede ayudarle a llegar a políticas públicas de una manera más rápida y justificar cuantitativamente procesos políticos e intergubernamentales.	Si el modelo tiene métricas bajas puede estar comunicando de manera errónea a demás stakeholders qué políticas son indicadas para implementar o que clases tienen más textos asociados y se deberían enfocar.
Director ejecutivo	Usuario-cliente	Dado que es quien comunica resultados generales, el clasificador puede ayudarle en sus roles de: estrategia al Identificar y contribuir a definir las prioridades estratégicas y operativas globales del UNFPA.	Dado que es quien comunica resultados generales, puede generar desinformación si las métricas son bajas.
Habitante local	Proveedor de los datos / Beneficiador	Apoya con su visión e identifican cuáles son los principales problemas del país para que luego se formulen buenas políticas que los beneficien.	Manejo incorrecto de los datos que lleve a la violación de la privacidad de los datos y planteamiento de políticas que no benefician a la comunidad.

Se considera que la aplicación presentada tiene dos funcionalidades como ya se había mencionado: 1) clasificar un único texto ingresado por un usuario y 2) clasificar múltiples textos. Estas dos funcionalidades sirven para distintos actores. En el caso de la primera funcionalidad se considera que

sirve para un habitante local ya que le permitiría identificar en su localidad, cotidianidad cuáles son los problemas que se presentan y a qué ODS pertenecen. Esta información se puede compartir en juntas de acción y reuniones de participación donde los habitantes pueden formular propuestas de alto impacto que pueden llegar a los representantes en el gobierno local para ser aplicadas.

En segundo lugar, la funcionalidad de subir un archivo de Excel le puede servir a la UNFPA u cualquier otra organización que tenga altos volúmenes de datos para clasificar múltiples textos de encuestas o propuestas de habitantes locales. La App les permite descargar de una vez el Excel clasificado, además de descargar gráficas que resumen la información obtenida y se pueden presentar en reuniones de alta gerencia cómo resumen general. Los procesos de negocio asociados se observan en la tabla 2 para cada actor en la columna beneficio. Los resultados de las gráficas mostradas se observan en la figura 3.

Figura 3. Posibles Gráficas que pueden descargar de la aplicación A) Pie Chart con resultado de porcentaje por cada clase y b) palabras representativas para la clase 3 (salud)

4. Resultados Y Recomendaciones

En general, se observaron los siguientes resultados a lo largo de este proyecto:

- El clasificador de Ridge es un modelo computacionalmente eficiente fácil de ajustar a nuevos datos y entrenar constantemente dada la posibilidad de que nuevos habitantes locales presenten nuevas propuestas u opiniones. Haber hecho que el pipeline fuera un conjunto de clases y que se pudiera guardar en un joblib por proceso de ejecución también facilita su posterior modificación.
- El modelo seleccionado es altamente interpretable. El negocio puede verificar cuáles *features* está utilizando más el modelo para clasificar los datos. Sabiendo, a qué palabras le está poniendo más atención el modelo, se puede volver a reentrenar. Esto lo hace con la segunda funcionalidad de la aplicación como se observa en el wordcloud presentado en este documento lo cuál lo hace fácil de entender para el negocio.
- Es esencial tener en cuenta que el modelo se construye utilizando datos pasados y una pequeña base de datos, lo que significa que su rendimiento puede verse afectado por cambios en el comportamiento de los usuarios y cantidad de datos. Por lo tanto, es fundamental realizar un

seguimiento constante y actualizar el modelo de manera regular para garantizar que siga siendo válido y preciso.

- El tiempo de respuesta de la aplicación (latencia) es demasiado alto. Clasificando únicamente 20 líneas del archivo sin etiquetas se podía demorar más de 30 segundos. Esto es un gran aspecto a mejorar para el negocio ya que puede ser muy estresante para los usuarios.

5. Trabajo en equipo

Se realizó una reunión inicial de lanzamiento y planeación en la que se definieron los roles y forma de trabajo. En esta reunión se definieron los roles y cómo se iba a realizar la primera parte de la entrega del proyecto. Se llegó al acuerdo de que Antonin era el encargado del rol de ingeniero de datos, Ernesto el líder de proyecto y ingeniero de software responsable del diseño de la aplicación y Lina la ingeniera encarda de desarrollar la aplicación final. Adicionalmente, cada integrante estaba encargado de escribir en este informe o hacer el video. Posteriormente, se hicieron dos reuniones subsecuentes: una para verificar el pipeline y una final para verificar el funcionamiento de la aplicación y reunirse con el experto de estadística para que la probara y nos diera su opinión.

Tabla 7. Repartición de Puntuación entre Integrantes

Puntaje Obtenido	**Antonin Bouillaud**	**Ernesto Duarte**	**Lina Gómez**
	0.33	0.33	0.33

En total, cada miembro le dedicó por su parte aproximadamente 8-12 horas al proyecto sin contar las reuniones grupales. Uno de los problemas con los que se enfrentó el grupo fue el tiempo de corrida para buscar los mejores parámetros por lo que tocó hacerlo con antelación para cumplir con la fecha de entrega.

6. Retroalimentación de Estudiante de Estadística

El estudiante de estadística nos comentó de la anterior entrega que luego de hablar con su profesor, no había modificaciones adicionales que realizar dado que no estábamos haciendo generalizaciones con nuestras afirmaciones. Asimismo, nos comentó cuando estaba probando la aplicación que le pareció fácil de utilizar y le gustó mucho. Este aparece en el video publicado en el padlet probando la aplicación. Más a futuro se espera seguir trabajando con el experto de estadística para siguientes proyectos.