Interpretación y Relacionamiento de textos con Objetivos de Desarrollo Sostenible

Proyecto desarrollado por:
Sergio Guillen(201912757) - Santiago Mora(201913351) - David Cruz(201912150)

Comprensión del Negocio y Enfoque analitico

La Organización de las Naciones Unidas (ONU) adoptó, el 25 de septiembre del año 2015, la Agenda 2030 para el desarrollo sostenible, cuyo fin es reducir la pobreza, garantizar acceso a la salud y educación, buscar igualdad de género y oportunidades, disminuir el impacto ambiental, entre otros. Esta agenda se basa en 17 objetivos de desarrollo sostenible (ODS) y 169 metas (derivadas de los diferentes ODS). Dentro del trabajo en conjunto de diferentes entes para alcanzar el cumplimiento de los ODS, muchas entidades tienen como enfoque el seguimiento y la evaluación de las políticas públicas y su impacto a nivel social. Este es el caso del Fondo de Poblaciones de las Naciones Unidas (UNFPA) que, junto con entidades públicas y haciendo uso de diferentes herramientas de participación ciudadana, busca identificar problemas y evaluar soluciones actuales, relacionando la información con los diferentes ODS. En este contexto, uno de los procesos que requiere de un mayor esfuerzo es la clasificación de la información textual que es recopilada, ya que es una tarea que consume gran cantidad recursos y para la cual se requiere un experto.

Por lo anterior se vió la oportunidad de crear una serie de modelos de Machine Learning para clasificar y relacionar los datos textuales que se tienen respecto a cada ODS y sus derivados, los cuales, serán utilizados por entidades como UNFPA para tomar y/o implementar las decisiones que consideren necesarias.

Objetivos de negocio:

- Identificar problemáticas a partir de los archivos de texto buscando relacionarlos con algún ODS dentro de la Agenda 2030.

Para lograr el objetivo es necesario tener en cuenta los siguientes requerimientos técnicos:

- Aplicar técnicas de machine learning para realizar un análisis sobre lenguaje natural y poder identificar el ODS que debería estar relacionado con el artículo que se encuentre en revisión.
- Realizar un procesamiento y preparación de los datos para los modelos de machine learning.
- Realizar un proceso de descripción y análisis de los datos para entender el contexto del problema.
- Realizar un análisis detallado de los resultados obtenidos por los algoritmos de machine learning y con base en estos proporcionar métricas al UNFPA.

Tarea de analítica de datos - Machine Learning

Teniendo en cuenta el contexto del problema y la base de datos seleccionada, se identifica que el problema se puede resolver como una tarea de clasificación, donde a partir de un texto plano, se pueda representar en forma de bolsa de palabras para poder ser interpretado por un clasificador. Los datos originalmente tienen no solo el texto plano sino un identificador SDG con rango 3 a 5 indicando al ODS que pertenecen. En este caso en particular se quiere generar etiquetas a los textos por lo tanto el SDG servirá como verificador de estas etiquetas.

Criterios de éxito

Teniendo en cuenta el contexto planteado, se puede establecer como criterio de éxito el tener un rendimiento del modelo de al menos 60% puesto que un porcentaje menor sería muy cercano al 50% lo cuál es dejarlo al azar, así mismo el tiempo de respuesta del modelo no debe ser muy alto por lo tanto se espera que la aplicación sea exitosa no se debe superar los 4 segundos para dar una respuesta.

Tabla Requerimiento de negocio- Requerimiento Machine Learning

Oportunidad o problema de negocio	Clasificar los textos dentro de las ODS 3,4 y 5 para identificar sus problemáticas y posibles soluciones.
Descripción del requerimiento desde el punto de vista de aprendizaje automático	Para una entrada de un texto plano *X*, se quiere encontrar una etiqueta *Y* que corresponde al tipo de ODS (*Y* ={3, 4, 5}) en donde puede ser clasificado el texto.
Organización y rol dentro de ella que se beneficia con la oportunidad definida	La organización beneficiada sería la ONU más específicamente el UNFPA que es el encargado de implementar estrategias para la ciudadanía
Contacto con experto externo al proyecto	Martina Pombo estudiante de estadística para ciencias sociales.

Entendimiento y Preparación de los datos

Los datos son una muestra recopilada en un archivo formato xlsx donde cada fila contiene dos variables si es el dataset etiquetado y una única variable para el dataset no etiquetado. En ambos dataset se tiene el ***Textos_espanol*** que es la variable con el texto plano a clasificar. En el caso de ser etiquetado se tiene el ***SDG***

el cual puede tomar valores de 3 a 5 para indicar el tipo de ***ODS*** al que pertenece el texto.
Aclaración: Aunque SDG y ODS significan lo mismo. En nuestro caso nos referiremos a SDG como la columna y ODS al significado de su valor.)

Para la preparación de los datos se comenzó por utilizar las librerías de ***spacy*** y ***stanza*** como lematizador de palabras en español y el stemmer de nltk para aplicar stemming. Se verifica si la tokenización y los métodos de preprocesamiento son los apropiados para los textos que se van a manejar en el dataset en español.
En el perfilamiento de los datos procedemos a importar los datos de los archivos xlsx para poder visualizarlos, comenzando por los datos etiquetados primero verificamos el estado de los textos y los valores que se tienen en las etiquetas; una vez con esto verificamos que las etiquetas estén dentro del rango de 3 a 5 que es el establecido y para los textos verificamos los posibles caracteres que se tienen y los que puedan dar problemas por el tipo de **codificación** que se tiene. Se observa que los datos,las filas y columnas, tienen valores completos los cuales son de alrededor de 3000 entradas. En cuanto a los textos se observa que hay problemas en la codificación de algunos caracteres especiales como lo son las tildes. Identificados estos problemas se procede con la limpieza empezando por corregir aquellas palabras que están mal codificadas, después se eliminan aquellas que no pertenecen a la tabla **ASCII** luego se remueve todos los signos de puntuación, se reemplazan los números por palabras, las palabras se dejan todas en minúscula y se remueven los artículos para así eliminar el ruido que existe en el texto.
Al haber realizado la limpieza se procede a tokenizar las palabras y normalizarlas para su clasificación.

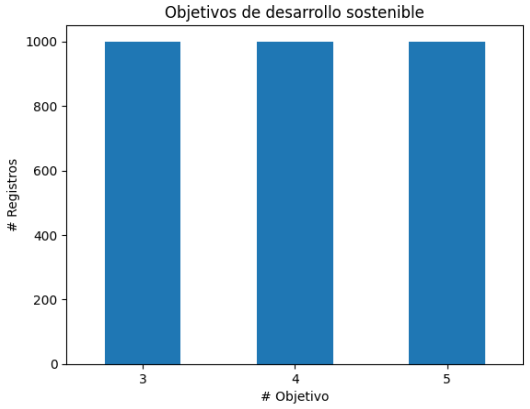


Figura 1: Distribución de los ODS en el conjunto de datos

	Textos_espanol	sdg	texto_final
0	Por ejemplo, el nÃºmero de consultas externas ...	3	[dat especial ocde receta existent gast mayor ...
1	En 2007, el gobierno central financió directam...	3	[pobre realiz nivel ofert 56 baj gobiern trans...
2	Claramente, hay muchos otros factores en juego...	3	[cronic jueg muj aborto selectivo reproduct pr...
3	Por ejemplo, el estado australiano de Victoria...	3	[especial ingreso fij dinamarca embargo financ...
4	El consumo anual de alcohol se estima en 15,7 ...	3	[ocde obeso relat ayudar hombr obesity bien ue...
...	...	...	...
2995	Un caucus efectivo se basa en fuertes vínculos...	5	[objet interinstitucional nivel muj federal co...
2996	Por el contrario, el porcentaje de hogares en ...	5	[muj panel espana hogar descens ahora trabajar...
2997	El análisis utilizará una gama de medidas que ...	5	[commonwealth movimiento econom mucho trabajar...
2998	La capacitación económica, el apoyo y, a veces...	5	[econom zomba altern incentivo social alternat...
2999	Esto ha sido reconocido por los comitÃ©s en la...	5	[muj garantiz aborto reproduct embarazo ayudar...

3000 rows × 3 columns

Figura 2: Conjunto de datos de textos en español, su ODS y su representación tokenizada y normalizada, Stemming y Lematización en español

Dado que los datos están balanceados, no se tomó ninguna medida en particular. En esta etapa los datos ya se encuentran preparados por lo cual se procede a eliminar la variable ***Textos_espanol*** puesto que el clasificador solo necesita palabras, por lo tanto se deben unir nuevamente los tokens o palabras generadas por los procesos de Normalización (stemming y lematización) para posteriormente poder aplicar algún método de Vectorización ya sea **countVectorizer** o **TfidfVectorizer.** Una vez realizada la división de conjuntos de datos y pruebas se procede a aplicar countVectorizer y TfidVectorizer.
Se evidencia que para cada columna se tomaron todas las palabras y se transformaron a numéricas, en donde palabras tenía un total de 15948 palabras y 2250 registros diferentes en nuestra Bolsa de palabras

También se pudo ver que dado que este dataframe fue generado con el uso de tfidf los valores de las columnas no son binarios. Existen valores entre 0 y 1.

Modelado y Evaluación

Para el modelado se utilizaron 3 algoritmos diferentes. Random Forest Classifier, Naive Bayes y Regresión logística debido a que sirven para clasificar los conjuntos de datos.

Random Forest Classifier - Sergio Guillen (201912757)

En nuestro modelo utilizaremos el algoritmo Random Forest Classifier. Si bien este algoritmo sirve para hacer tanto regresión como clasificación, lo utilizaremos de esta manera ya que es nuestra tarea de modelado.

(Nota: Los pasos detallados del modelado se encuentran en el notebook)

Para nuestro modelado con Random Forest Classifier lo primero que se intentó fue utilizar el conjunto de datos de entrenamiento y prueba divididos anteriormente aunque se

pudo utilizar otro cargado de la importación del data frame preparado anteriormente. Haciendo uso de la librería de RandomForest Classifier ajustamos los parámetros del algoritmo como lo son el número de estimadores y la profundidad máxima en un rango de [100,300] y [8,15] respectivamente, ya que son los valores recomendados para un buen modelado estándar con el algoritmo.

Aplicamos RandomForest Classifier a dos tipos de datos entrenamiento y prueba distintos. Estos son los vectorizados con **count Vectorizer** y **tfidfVectorizer** para poder ver con cual método el algoritmo obtiene mejores resultados.

Utilizamos un Grid Search para ambos conjuntos. Grid Search utiliza KFold el cual hace k divisiones, evalúa el rendimiento del modelo para cada división para finalmente validar y entregarnos el modelo que objetivamente puede predecir mejor el resultado de datos desconocidos.

Una vez obtenidos los mejores parámetros para el RFC aplicamos el algoritmo a los conjuntos de datos de prueba y entrenamiento, y posteriormente predecimos el conjunto de prueba. En nuestro caso la columna **SDG.**

Naive Bayes Classifier (MultinomialNB) - David Cruz (201912150)

El **MultinomialNB** es un clasificador basado en el teorema de Bayes el cual utiliza algoritmos de tipo Naive o ingenuos porque asumen que las variables predictoras son independientes entre sí; con esto en mente se utiliza el algoritmo multinomial que es especialmente bueno para clasificar textos dependiendo de las regresiones.
Para este modelo y como se ha explicado en los anteriores se creó un conjunto de datos de prueba y de entrenamiento del modelo y se procede a utilizar la librería de MultinomialNB disponible en Scikit-learn.

Se aplica el algoritmo tanto a los datos de prueba como a los de entrenamiento que ya se encontraban vectorizados con **countVectorizer** y **tfidVectorizer** para comprobar cuál de los dos métodos el algoritmo obtiene mejores resultados.

Utilizando GridSearch sobre ambos conjuntos utilizando KFold el cuál hace k, divisiones evaluando el rendimiento del modelo para cada división y finalmente se valida y entrega el mejor modelo que objetivamente puede predecir el mejor resultado de los datos desconocidos.

Con los pasos anteriores se obtienen los mejores parámetros para el NB y aplicando el algoritmo a los datos de prueba y entrenamiento, se hace la predicción de los conjuntos sobre la variable objetivo SDG.

Regresión Logística - Santiago Mora (201913351)

El LogisticRegression es un modelo de clasificación que se basa en la ecuación logística que se usa en estadística, este modelo se considera bastante bueno al clasificar textos, y, si bien se considera mejor cuando los datos que se quieren predecir son binarios, se decidió utilizar este modelo dado que es útil para clasificar textos.

En esta ocasión, se entrenó el algoritmo sobre los datos destinados para ello que ya estaban vectorizados anteriormente, si bien se utilizaron dos métodos de vectorización, para este modelo solo se tomó en cuenta la que usó tfidVectorizer. Dado que para este algoritmo no se requiere buscar los mejores parámetros, el modelo se entrenó directamente sobre los datos e inmediatamente después se procedió a hacer la predicción correspondiente.

Análisis de Resultados

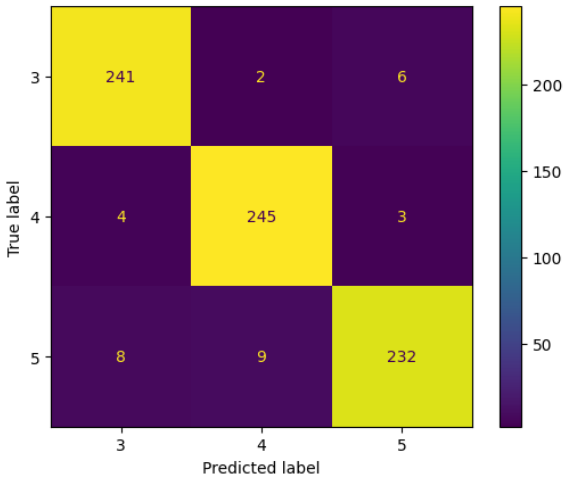
Random Forest Classifier - Sergio Guillen (201912757)

En este algoritmo se pudo evidenciar como al hacer la división temprana de los datos de prueba y entrenamiento, el metodo de vectoriación que nos arroja los mejores resultados sobre el conjunto de prueba es **TFIDF.** *(Tf-idf, frecuencia de término – frecuencia inversa de documento, es una medida numérica que expresa cuán relevante es una palabra para un documento en una colección)*

La razón por la cual **TFIDF** tuvo un mejor rendimiento que **countVectorizer** se debe principalmente a que **TFIDF** no solo se centra la frecuencia de las palabra sino en la importancia de cada una. Esto explica ademas el hecho de que nos dio una menor bolsa de palabras que con countVectorizer.

Con una exactitud del 96% sobre conjuntos de prueba, este algoritmo fue el escogido para la generación y aplicación de un **Pipeline** ajustado a RandomForestClassifier y TFIDF sobre el conjunto de textos sin etiquetar a entregar.

```
countVectorizer
RFC: Exactitud sobre entrenamiento: 0.51
RFC: Exactitud sobre test: 0.38
TFID Vectorizer
RFC: Exactitud sobre entrenamiento: 1.00
RFC: Exactitud sobre test: 0.96
```

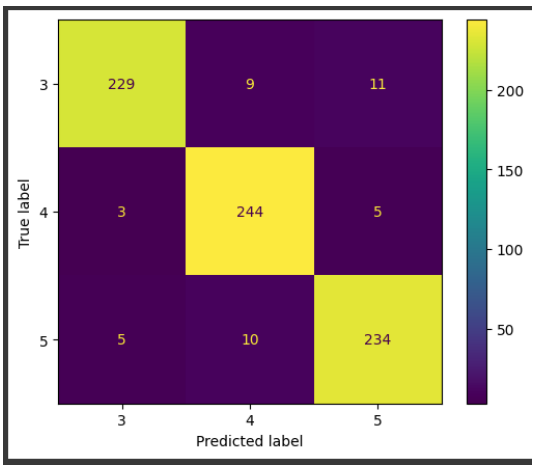


	precision	recall	f1-score	support
3	0.95	0.97	0.96	249
4	0.96	0.97	0.96	252
5	0.96	0.93	0.95	249
accuracy			0.96	750
macro avg	0.96	0.96	0.96	750
weighted avg	0.96	0.96	0.96	750

Naive Bayes Classifier - David Cruz (201912150)

Para este algoritmo se con las divisiones de los conjuntos de datos de prueba y entrenamiento se observó que el conjunto que tiene mejor rendimiento es el **TFIDF** (*Tf-idf, frecuencia de término – frecuencia inversa de documento, es una medida numérica que expresa cuán relevante es una palabra para un documento en una colección*). Se observó que al utilizar **TFIDF** el cuál no solo se centra en la frecuencia de aparición de las palabras sino en la importancia de cada una de ellas arroja los mejores resultados proporcionando así una mejor bolsa de palabras y en este caso en particular una más pequeña.
Siendo así como se observa en la figura se tiene un f1-score del 94% calificando así como un modelo cercano a la perfección.

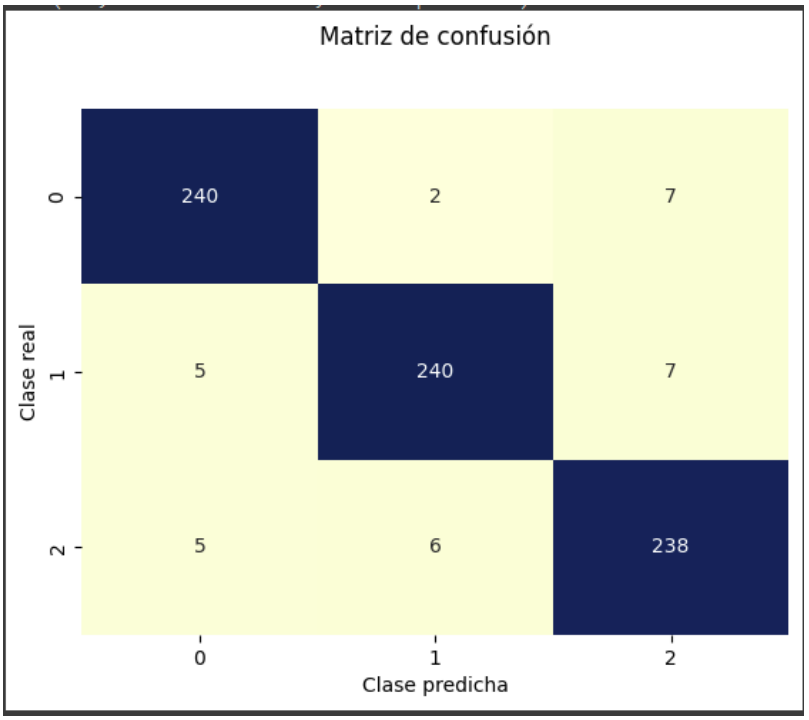
	precision	recall	f1-score	support
3	0.97	0.92	0.94	249
4	0.93	0.97	0.95	252
5	0.94	0.94	0.94	249
accuracy			0.94	750
macro avg	0.94	0.94	0.94	750
weighted avg	0.94	0.94	0.94	750



Logistic Regression - Santiago Mora (201913351)

Llegados hasta este punto, con el grupo se llegó a la conclusión de que el tfidVectorizer garantiza una mejor bolsa de palabras, por lo cual la predicción nuevamente se hizo sobre el conjunto de datos vectorizado por este método.

Dicho lo anterior, el modelo logró una precisión del 96%, lo cual lo hace un modelo bastante bueno para predecir a qué ODS pertenece un texto y por lo tanto lo hace un modelo bastante útil .



	precision	recall	f1-score	support
3	0.96	0.96	0.96	250
4	0.95	0.97	0.96	248
5	0.96	0.94	0.95	252
accuracy			0.96	750
macro avg	0.96	0.96	0.96	750
weighted avg	0.96	0.96	0.96	750

Mapa de actores relacionado con un producto de datos creado con el modelo analítico construido

Rol dentro de la Empresa	**Tipo de actor**	**Beneficio**	**Riesgo**
Habitantes Ayudados (Locales)	Beneficiado	Su opinión es escuchada y tomada en cuenta para el enfoque de los ODS	Si el modelo tiene un mal desempeño, puede hacer que las opiniones de los locales se interpreten mal o directamente no

			sean escuchadas
UNFPA	Usuario - Cliente	Ayuda a darle un enfoque correcto y a saber cómo van las implementaciones de los ODS	Si el modelo no tiene buen desempeño, se le dará un enfoque incorrecto a la implementación de los ODS al no tomar bien en cuenta las opiniones de los locales
UN (Filial)	Financiador	Colaboración en el logro de los ODS	Si el modelo falla de alguna forma, el dinero invertido en este proyecto se perdería

Trabajo en equipo:

- **Líder de proyecto:** Sergio Guillen - Está a cargo de la gestión del proyecto. Define las fechas de reuniones, pre-entregables del grupo y verifica las asignaciones de tareas para que la carga sea equitativa. Se encarga de subir la entrega del grupo. Si no hay consenso sobre algunas decisiones, tiene la última palabra.
- **Líder de negocio:** David Cruz - Es responsable de velar por resolver el problema o la oportunidad identificada y estar alineado con la estrategia del negocio para el cual se plantea el proyecto. Debe garantizar que el producto se puede comunicar de forma apropiada. Debe encargarse de contactar al grupo de expertos de estadística para determinar la fecha en la que se van a reunir para revisar los resultados de esta etapa e iniciar el trabajo de la etapa 2.
- **Líder de datos:** - Santiago Mora - Se encarga de gestionar los datos que se van a usar en el proyecto y de las asignaciones de tareas sobre datos. Debe dejarlos disponibles para todo el grupo.
- **Líder de analítica:** Santiago Mora | Sergio Guillen| Martina Pombo - Se encarga de gestionar las tareas de analítica del grupo. Se encarga de verificar que los entregables cumplen con los estándares de análisis y que se tiene el “mejor modelo” según las restricciones existentes.

Para repartir los puntos se usó la descripción sugerida por el enunciado del proyecto.

En ese sentido se le asignará la puntuación promedio del resto de miembros del equipo asignando de 0 a 100 puntos posibles que tan bien se realizó la descripción de cada rol del proyecto.

- **Líder de proyecto:** Sergio Guillen
- **Puntuaciones:** 90-90-85
- **Puntuación final: 88/100**

- **Líder de negocio:** David Cruz
- **Puntuaciones:** 80 - 95 - 90
- **Puntuación final: 88/100**

- **Líder de datos:** Santiago Mora
- **Puntuaciones:** 90 - 90 - 80
- **Puntuación final: 86/100**

- **Líder de analitica:** Santiago Mora | Sergio Guillen | Martina Pombo
- **Puntuaciones:** 85 - 90 - 100
- **Puntuación final: 92/100**

Repositorio GitHub: https://github.com/saguillen/BI_Proyecto1.git