Universidad de los Andes

Facultad de Ingeniería
Departamento de Ingeniería de Sistemas
Inteligencia de negocios
2023-20

Proyecto 1 – Etapa 1

Daniel Fernando Gomez Barrera - 201728920
Lindsay Vanessa Pinto Morato – 202023138
Yei Hong Zhang Cárdenas – 201823976

15 de octubre de 2023
Bogotá D.C.

Contenido

1 Entendimiento del negocio y enfoque analítico. 3
2 Entendimiento y preparación de los datos 4
2.1 Entendimiento de los datos 4
2.2 Preparación de los datos 5
Limpieza de datos 5
Tokenización 6
Vectorización 6
3 Modelado y evaluación 6
3.1 Naive Bayes [Implementado por Lindsay Pinto] 6
3.2 Regresión logística [Implementado por Daniel Gomez] 7
3.3 Random Forest [Implementado por Yei Zhang] 8
4 Resultados 9
5 Mapa de actores relacionado con un producto de datos creado con el modelo analítico construido 10
6 Trabajo en equipo 11
6.1 Puntos a mejorar para la siguiente entrega 12
Bibliografía 12

1 Entendimiento del negocio y enfoque analítico.

Objetivos del Negocio:

- El objetivo principal del proyecto es desarrollar un modelo de clasificación basado en técnicas de aprendizaje automático que permita relacionar textos con los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030 de la ONU.

- Crear una aplicación que facilite la interacción con los resultados del modelo, permitiendo a UNFPA interpretar y analizar la información textual recopilada en procesos de planeación participativa para el desarrollo a nivel territorial.

Criterios de Éxito:

- El modelo debe tener una alta precisión y recall en la clasificación de textos según los ODS, con métricas de evaluación superiores al 90%.

- La aplicación desarrollada debe ser intuitiva y eficiente en la presentación de resultados, lo que se reflejará en una alta tasa de adopción y satisfacción por parte de los usuarios de UNFPA.

Descripción de los ODS involucrados y su impacto en Colombia

En el contexto de la Agenda 2030, los Objetivos de Desarrollo Sostenible (ODS) son metas globales que buscan abordar desafíos sociales, económicos y ambientales para lograr un mundo más equitativo y sostenible. Para este proyecto, es esencial identificar los ODS específicos que se abordarán:

ODS 6 - Agua Limpia y Saneamiento: Este ODS se centra en garantizar la disponibilidad y gestión sostenible del agua y el saneamiento para todos. En Colombia, abordar este objetivo puede tener un impacto significativo en la calidad de vida de las comunidades, especialmente en áreas donde el acceso al agua potable y saneamiento básico es limitado.

ODS 7 - Energía Asequible y No Contaminante: Este ODS busca asegurar el acceso a una energía asequible, fiable, sostenible y moderna para todos. En Colombia, abordar este objetivo puede contribuir a la mejora de la infraestructura energética y a la reducción de la dependencia de fuentes contaminantes.

ODS 16 - Paz, Justicia e Instituciones Sólidas: Este ODS tiene como objetivo promover sociedades pacíficas e inclusivas para el desarrollo sostenible, proporcionar acceso a la justicia para todos y construir instituciones eficaces, responsables e inclusivas a todos los niveles. En Colombia, este ODS es especialmente relevante para fortalecer el sistema de justicia y promover la paz en áreas afectadas por conflictos.

Oportunidad/ problema Negocio	El problema de negocio radica en la necesidad de UNFPA de automatizar el proceso de clasificación de textos que representan la opinión de habitantes locales sobre problemáticas específicas. La clasificación manual consume recursos significativos y requiere de expertos. La oportunidad radica en el desarrollo de un modelo de clasificación automática que optimice este proceso, permitiendo a UNFPA analizar de manera efectiva la información textual recopilada y relacionarla con los ODS de la Agenda 2030. Esto puede tener un impacto positivo en la eficiencia y eficacia de los esfuerzos de desarrollo sostenible en Colombia.
Enfoque analítico	• **Fuente de Datos:** Los datos provienen de textos recopilados a través de diferentes fuentes por UNFPA, que representan opiniones de habitantes locales sobre problemáticas específicas de sus entornos.

	• **Preprocesamiento de Datos:** Se requiere una etapa de preprocesamiento para limpiar y preparar los textos antes de la clasificación. Esto incluirá la tokenización, eliminación de stop words y la vectorización de texto. • **Selección de Características:** Se aplicará un enfoque basado en procesamiento de lenguaje natural (NLP) para identificar características relevantes en los textos. • **Modelo de Clasificación:** Se utilizarán técnicas de aprendizaje automático para desarrollar un modelo que clasifique automáticamente los textos según los ODS. Esto podría implicar el uso de algoritmos de clasificación como Naive Bayes, Regresión logística o modelos de clasificación basados en árboles. • **Entrenamiento del Modelo:** Se utilizarán conjuntos de datos etiquetados previamente para entrenar el modelo. Se aplicarán técnicas de validación cruzada para evaluar su rendimiento. • **Evaluación y Métricas:** Las métricas de evaluación incluirán precision, recall, F1-score.
Organización que se beneficiaría con la oportunidad	El Banco de Desarrollo de América Latina y el Caribe mediante esta propuesta puede verse beneficiado en sus procesos de: • Optimizar el Proceso de Evaluación y Selección de Proyectos • Mejorar el Monitoreo y Evaluación del Impacto de Proyectos Financiados • Facilitar la Identificación de Áreas de Inversión Estratégica • Promover Colaboraciones Efectivas
Contacto con experto externo al proyecto	Isabella Gonzalez Castellanos i.gonzalezc23@uniandes.edu.co Fecha de reunión: 13 de octubre de 2023 Canal: Para la primera reunión se usó Zoom donde se habló el contexto del problema y se resolvieron dudas.

2 Entendimiento y preparación de los datos

2.1 Entendimiento de los datos

Frecuencia de Palabras:
Las palabras más comunes en el texto son las "stopwords", siendo "de" la más repetida.

Unicidad de Filas:
Cada fila en el dataframe es única, lo que indica que no hay duplicados en el conjunto de datos evaluado.

Longitud de Texto:
La longitud de los textos varía entre 143 y 1616 caracteres, con un promedio de aproximadamente 770 caracteres.

Organización de los SGD:
Cada SGD está compuesto por 1000 filas.

Caracteres Especiales:
Se han identificado caracteres como las vocales con tilde, representando alrededor del 88.4% del total. También se han detectado otros caracteres menos comunes que requerirán un tratamiento especial.

Signos de Puntuación:

Se han identificado signos de puntuación como comillas, guiones, puntos, comas y puntos suspensivos, los cuales necesitarán ser abordados en el proceso de análisis posterior.

Stop Words:
Hay un total de 158 stopwords diferentes en los datos suministrados.

Predominancia de Neutralidad:
Es notable que la categoría más frecuente es "Neutral", lo cual indica que una gran proporción de textos evaluados no exhiben un sentimiento claramente positivo o negativo.

Predominancia de idioma español:
Al evaluar los idiomas presentes en el dataframe de entrenamiento se puede evidenciar que más del 99% de los registros están escritos en español.

Balance de Registros:
En la tarea de clasificación uno de los aspectos a tener en cuenta es el balance en los datos de entrenamiento, esto debido a que el modelo podría quedar sesgado e inclinar el resultado de una clase, solo por el hecho de tener más registros. En este dataset los datos se encuentran balanceados.

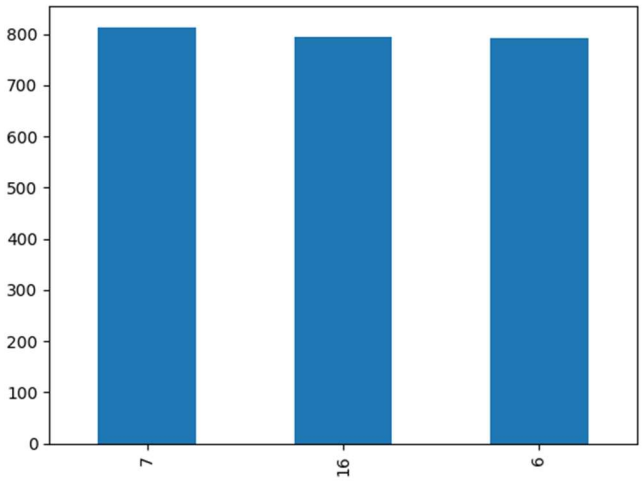

Ilustración 1. Balance de Clases en Registros de Entrenamiento

2.2 Preparación de los datos

Limpieza de datos

Se realiza una función por cada transformación. La primera cambia los caracteres dependiendo del encoding. Revisando el dataset notamos que hay tanto latin-1 encodig así como utf-8. Por lo que por cada palabra se puede dejar como está o intentar codificar por latin-1 si al codificar y decodificar en utf-8 aparece un acento aceptado en utf-8 se mantiene la palabra transformada. Esta medida asegura la uniformidad en la representación de caracteres.

La siguiente función elimina los acentos de las palabras y posteriormente se define otra función para convertir a minúsculas todas las palabras. Esta acción es esencial para evitar discrepancias en comparaciones posteriores, garantizando que las palabras escritas en mayúsculas o minúsculas sean tratadas de manera uniforme.

A continuación, se remueven los signos de puntuación lo cual simplifica el procesamiento subsiguiente y reduce el ruido en el texto presentes en el dataframe. Luego, se eliminan las palabras de parada detectadas en el análisis de los datos. Esta decisión se basa en un análisis previo de los datos, donde se identificaron términos frecuentes pero poco informativos. Además, se retiran una serie de caracteres especiales previamente identificados en el dataframe, lo que asegura la eliminación de elementos superfluos que no contribuyen al análisis.

Finalmente se crea una función que encapsula todas estas funciones y será la encargada del preprocesamiento.

Tokenización

En el proceso de tokenización, se optó por emplear el SnowballStemmer de la biblioteca nltk. Esta herramienta es reconocida por su eficacia en la reducción de palabras a su forma base o raíz, lo que facilita la identificación de similitudes semánticas entre diferentes formas de una misma palabra. Esta elección se basa en la necesidad de simplificar el análisis al trabajar con una versión más compacta y representativa del texto original. El SnowballStemmer es particularmente útil en este contexto, ya que permite reducir palabras a sus formas fundamentales, lo que a su vez simplifica la comparación y el análisis de similitudes semánticas entre diferentes documentos.

Vectorización

En la fase de vectorización, se emplea la técnica $tf.idf$. que tiene en cuenta tanto la frecuencia de la palabra en el documento, así como la discriminación de la palabra en el corpus. Se considera efectiva en el procesamiento de texto, ya que permite destacar términos relevantes para el análisis, a la vez que atenúa el impacto de palabras extremadamente comunes.

3 Modelado y evaluación

3.1 Naive Bayes [Implementado por Lindsay Pinto]

El código realiza una búsqueda de hiperparámetros para el modelo Naive Bayes Multinomial. Se establece el parámetro de alpha en valores de 0.1 a 1.0 en incrementos de 0.1. Esto se hace para evaluar cómo diferentes valores de alpha afectan el rendimiento del modelo. Alpha es un hiperparámetro de suavizado de Laplace que controla la regularización en el modelo Naive Bayes. Se prueban múltiples valores para determinar cuál proporciona el mejor rendimiento.

No se ajustaron más hiperparámetros, ya que se optó por utilizar los valores predeterminados proporcionados por la documentación oficial del modelo. Esto se debe a que los valores predeterminados suelen estar configurados de manera que funcionen bien en una amplia variedad de casos.El parámetro force_alpha que es permite controlar cómo se maneja el alpha si es extremadamente pequeño tampoco fue modificado en el código y se optó por el parámetro por defecto. Finalmente, mediante un ciclo se escoge el mejor parámetro de alpha y este será el valor puesto en el pipeline, para este caso fue 0.3.

Como resultado se puede concluir que el modelo muestra una capacidad sólida para clasificar correctamente las instancias en las categorías evaluadas. Es particularmente eficaz en la categoría 16, donde alcanza un rendimiento casi perfecto. Aunque se observan pequeñas áreas de mejora, como el recall en las categorías 6 y 7, el modelo logra un excelente equilibrio entre precisión y recall en general. La alta precisión global del 97% sugiere que el modelo es confiable en sus predicciones. Los promedios macro y ponderado indican que el modelo es consistente en su rendimiento a través de las

diferentes categorías, lo que es un signo alentador. En general, el modelo parece ser adecuado para el problema, pero es importante continuar monitoreando su rendimiento y considerar ajustes si es necesario en el futuro.

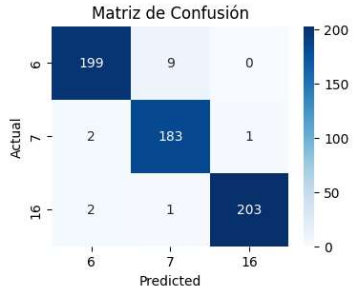

Figura 2. Matriz de Confusión NB

	precision	recall	f1-score	support
6	0.95673	0.98030	0.96837	203
7	0.98387	0.94819	0.96570	193
16	0.98544	0.99510	0.99024	204
accuracy			0.97500	600
macro avg	0.97535	0.97453	0.97477	600
weighted avg	0.97522	0.97500	0.97495	600

Figura 3. Resultados NB

```
Hyperparameter Search
Best Alpha: 0.3
```

Figura 4. Mejores Hiperparámetros NB

3.2 Regresión logística [Implementado por Daniel Gómez]

Se utilizó como segundo algoritmo el modelo de regresión logística. Este modelo puede incurrir en overfitting dado que a diferencia de la regresión lineal la solución no es determinística, sino que es un algoritmo iterativo que para cada ciclo cambia pesos en búsqueda de la convergencia del modelo[1].

Por esta razón se plantea una búsqueda de hiperparámetros que implemente un 10-fold cross validation. Esta validación divide los datos de entrenamiento en 10 subconjuntos y ajusta el modelo para cada uno, en cada entrenamiento valida el modelo con los otros 9 subconjuntos, como siempre hay un conjunto de validación para evaluar se reduce el error por overfitting. Ahora, para cada una de las combinaciones de parámetros obtiene un score, los mejores parámetros serán aquellos para los que el modelo tuvo un mejor score. En nuestro caso seleccionamos f1-micro como el score.

Los hiperparámetros[2] en el primer tipo de regresión logística utilizada fueron penalty (modelo de penalización de pesos grandes), fit_intercept (en caso de ser verdadero se incluye un término de bias) y max_iter (cantidad máxima de iteraciones para determinar convergencia del modelo). Tanto el modelo de penalización como max_iter son parámetros que influyen en el overfitting (Maina, 2021). Entre más iteraciones el modelo disminuye el train_loss, pero si se excede el número el pierde la capacidad de generalizar y no es bueno con registros que el modelo no han visto.

El segundo modelo de regresión logística utiliza SDG (stochasting gradient descent) como algoritmo de búsqueda de pesos. Aunque su nombre cambia en sklearn, sigue siendo una regresión logística (Dubey, 2018). Para este modelo hay dos parámetros adicionales learning_rate (permite la selección de modelo de selección de tasa de aprendizaje) y eta0 (tasa de aprendizaje inicial), la tasa de aprendizaje indica que tanto el modelo mueve los pesos en cada iteración.

Los mejores resultados se obtuvieron con el primer modelo de regresión lineal que utiliza como algoritmo de optimización LBFGS. Como se puede notar en la tabla agregada de métricas están por encima del 98%. El f1-score es sobresaliente en el objetivo de desarrollo sostenible 16 que tiene una precisión de 100% (todas las etiquetas existentes fueron clasificadas en el objetivo 16) y una

[1] En la sección de resultados está una explicación más extensa de la arquitectura del modelo
[2] En el notebook hay una explicación más extensa de cada uno de los hiperparámetros utilizados en la búsqueda.

sensibilidad de 99.5% (de todas las etiquetas clasificadas en el objetivo 16 99.5% estaban correctamente clasificadas). Lo mismo para el objetivo 6 y 7, tienen un alto f-score porque las sensibilidad y precisión son altas.[3]

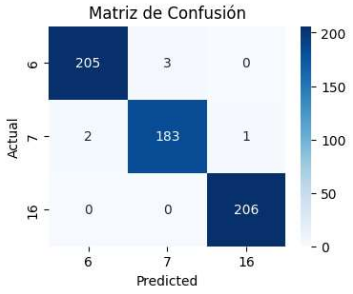

Figura 5. Matriz de Confusión LR

	precision	recall	f1-score	support
6	0.98558	0.99034	0.98795	207
7	0.98387	0.98387	0.98387	186
16	1.00000	0.99517	0.99758	207
accuracy			0.99000	600
macro avg	0.98982	0.98979	0.98980	600
weighted avg	0.99002	0.99000	0.99001	600

Figura 6. Resultados LR

```
Modelo 1 (LBFGS)
Best Training Score: 0.9854166666666668
Best Hyperparameters: {'fit_intercept': False, 'max_iter': 50, 'penalty': 'l2'}
Modelo 2 (SDG)
Best Score: 0.9887499999999999
Best Hyperparameters: {'eta0': 0.001, 'fit_intercept': True, 'learning_rate': 'optimal', 'max_iter': 1000,
'penalty': 'l1'}
```

Figura 7. Mejores Hiperparámetros LR

3.3 Random Forest [Implementado por Yei Zhang]

Para el último modelo se uso el algoritmo de Random Forest, este algoritmo puede ser utilizado como regresión o clasificación. Para este contexto claramente se usó el Random Forest Classification que tiene como bases los árboles de decisión. Los árboles de decisión son modelos de machine learning para toma de decisiones y predicciones en problemas de clasificación y regresión (IBM, s.f.). Se llaman así ya que tienen una representación gráfica de un árbol donde cada nodo es una característica o atributo, cada rama es una regla de decisión basada en dicha característica, y cada hoja es un resultado.

En este caso, el Random Forest crea varios árboles que emitirán una decisión de clasificación, utilizando la base de arboles de decisión anteriormente mencionados. Al final, todo el conjunto de todos los árboles elegirá con la Regla de la mayoría que indica cual fue la decisión que más votos tiene en todo el bosque. Las desventajas de los árboles de decisión son que pueden generar sesgo en sus modelos y tener una alta varianza, problema que RFC solución con la creación de muchos más árboles mejora la varianza del modelo y también reduce el sobreajuste que se pueda tener.

Para lograr esto, lo primero que se hizo un pipeline con todo el preprocesamiento de datos desde la limpieza de datos, la tokenización y vectorización de los mismo, finalizando con el clasificador Random Forest Classifier de Sklearn. Se corrió el modelo y se produjo la matriz de confusión que se muestra en la figura 8. Sin embargo, se hizo un GridSearchCV para buscar los mejores hiperparámetros, que en este caso son n_estimators, max_depth y criterion, lo que dio como resultado que los mejores fueron 100, 75 y gini respectivamente. Con lo anterior se creo otro pipeline con los mejores parámetros y se compararon ambos pipelines principalmente con la métrica f1 score, concluyendo que ambos modelos resultaban muy similares y que sus diferencias eran insignificativas.

[3] En la sección de resultados se añaden más detalles de los resultados de este modelo.

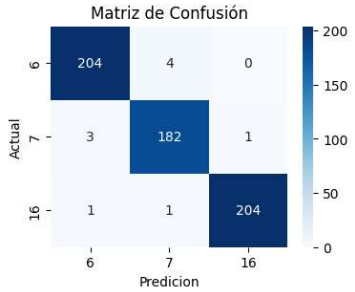

Figura 8. Matriz de Confusión RF

	precision	recall	f1-score	support
6	0.98077	0.98077	0.98077	208
7	0.97849	0.97326	0.97587	187
16	0.99029	0.99512	0.99270	205
accuracy			0.98333	600
macro avg	0.98319	0.98305	0.98311	600
weighted avg	0.98331	0.98333	0.98332	600

Figura 9. Resultados RF

```
Best Score: 0.9829166666666665
Best Hyperparameters: {'criterion': 'gini', 'max_depth': 75, 'n_estimators': 100}
```

Figura 10. Mejores Hiperparámetros RF

4 Resultados

El modelo seleccionado es *Regresión Logística* (sklearn.linear_model.LogisticRegression) utilizando el algoritmo de LBFGS y los hiperparámetros *max_iter = 50, 'fit_intercept': False, 'max_iter': 50 y 'penalty': 'l2'.* En la sección 3.2 se explica la implementación.

El objetivo de nuestro algoritmo es automatizar la clasificación de documentos. La arquitectura de la regresión logística (Figura 11) permite entender como a partir de la entrada de palabras se clasifican los textos y así evaluar cualitativamente la efectividad del clasificador de documentos en un ambiente real.

Para esto vamos a emplear un análisis de características, cada componente del vector de entrada corresponde a una palabra. Ahora bien, el conocimiento adquirido por el modelo se encuentra en los pesos. Hay un peso para cada clase y cada token $w_{clase,token}$, es decir hay $|V| * 3 + 0$ pesos dónde |V| es el tamaño del vocabulario. Entre mayor sea el peso más importancia tiene ese token (en caso de aparecer) en determinar la clase, con la suma de los pesos multiplicado por las componentes del vector se obtiene la probabilidad de que el texto corresponda a esa clase.

Por lo tanto, se encontraron los tokens con más influencia a clasificación del algoritmo. Se tomaron las 5 palabras con más peso y las 5 palabras con menos peso. Para el objetivo 6 se encontró que el token más influyente es “agu” que viene de la palabra agua (Figura 12). Esto tiene mucho sentido dado que este objetivo es “Agua Limpia y Saneamiento”. Igual que “hidric”, “cuenc”, “rio” y “rieg”. Ahora bien, si queremos hacer un análisis exhaustivo se podrían analizar más tokens y encontrar relaciones menos obvias, pero estos resultados nos indican que el modelo si le está dando importancia a los tokens más relevantes.

Lo mismo sucede con los objetivos 7 y 16, los tokens con más importancia vienen de palabras estrechamente relacionadas con la definición del objetivo.

Ahora bien, una observación importante se obtiene a partir de los tokens con pesos menores, que para estos objetivos son negativos. Es decir, si aparecen en el texto le restan a la sumatoria disminuyendo la probabilidad de pertenecer a esa clase. En muchos casos los tokens con pesos negativos en una clase son para otra clase los pesos mayores, esto se debe a que como solo existen tres clases si hay una palabra que aparece mucho en una clase y muy poco en las otras tiene una mayor discriminación, por ejemplo, *energet* es negativo para el objetivo 6 y 16 pero positiva para el 7. Esto no sucede con todos, por ejemplo, aunque el token “agu” es muy relevante para el ODS 6 seguramente también aparece en los otros objetivos con alta frecuencia. De aquí sale una conclusión importante: el modelo

con alta certeza clasifica correctamente un texto si pertenece a una de esas 3 clases, si se le entrega un texto de otro objetivo desarrollo sostenible no es fiable en determinar a cuál de los tres objetivos de desarrollo sostenible se parece más.

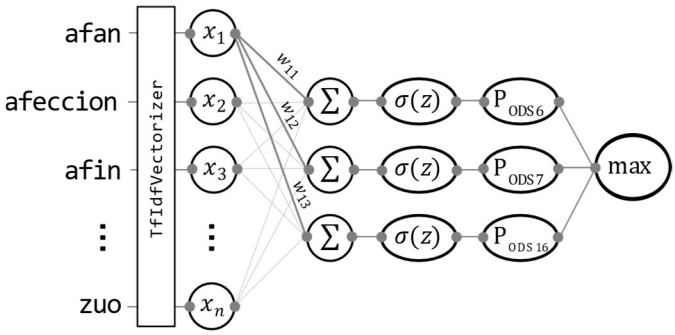

Figura 11. Arquitectura LR. Elaboración Propia.

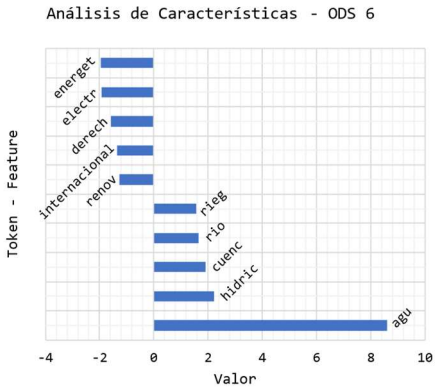

Figura 12. Análisis de Características ODS 6

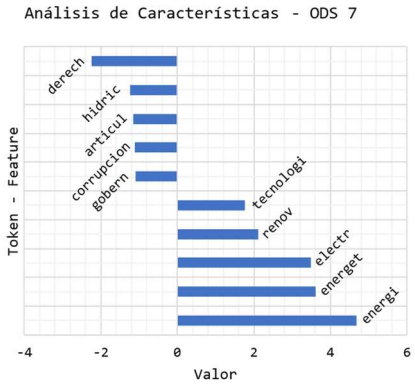

Figura 13. Análisis de Características ODS 6

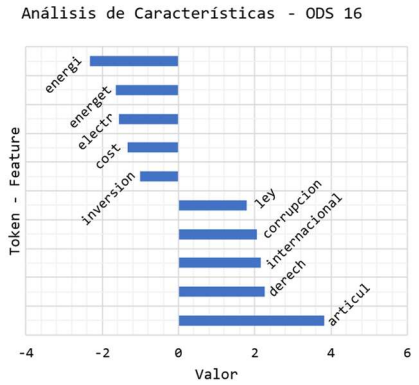

Figura 14. Análisis de Características ODS 6

Este algoritmo se seleccionó entre los tres porque contaba con los mejores scores en todas las métricas[4]. En la Figura 5 y 6 se pueden observar los resultados. El accuracy para este set de datos (dado que está balanceado) es un un buen indicador y está en el 99%. Por lo que se concluye que el modelo es capaz de clasificar textos pertenecientes a estos tres objetivos de desarrollo sostenible con alta confiabilidad cumpliendo con el objetivo del negocio.

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

Organización: Banco de Desarrollo de América Latina y el Caribe (también conocido como CAF - Banco de Desarrollo de América Latina) es una institución financiera multilateral que tiene como objetivo promover el desarrollo sostenible y la integración regional en América Latina y el Caribe.

Rol dentro de la empresa	Tipo de actor	Beneficio	Riesgo
Presidente Ejecutivo de CAF	Líder de la Institución	Tomar decisiones estratégicas basadas en datos con respecto a la financiación de proyectos y programas.	Posible desafío en la implementación y aceptación del modelo.

[4] En la sección 3.2 se comenta sobre el significado de estas métricas.

Equipo de Evaluación de Proyectos	Empleados de CAF	Evaluar y seleccionar proyectos que estén alineados con los ODS de manera más efectiva y objetiva.	Evaluar y seleccionar proyectos que estén alineados con los ODS de manera más efectiva y objetiva.
Gobiernos Nacionales	Cliente	Ayudar a los gobiernos a seleccionar proyectos que tengan un impacto positivo en los ODS.	Dependencia del modelo sin tener en cuenta las particularidades locales y las necesidades específicas de cada país.
Beneficiarios de Proyectos Financiados por CAF	Beneficiario	Asegurar que los proyectos financiados tengan un impacto significativo en términos de desarrollo sostenible.	Posible desvinculación de la participación comunitaria y de la adaptación a las necesidades reales de las comunidades.
ONGs (Socios de CAF)	Socio Externo	Ayudar a las ONGs a seleccionar y diseñar proyectos que estén alineados con los ODS.	Posible falta de confianza en los resultados del modelo y su aplicabilidad en contextos locales específicos

6 Trabajo en equipo

Estudiante	Rol o Roles	Trabajo realizado	Retos y cómo se resolvieron	Puntos asignados	Número de horas
Daniel Gomez	Líder de proyecto Líder de datos	**Algoritmo realizado:** regresión Logística **Trabajo:** - Realización de tokenización y vectorización - Pregunta 3 y 4 del informe - Pregunta 6 del informe - Edición del video	Reto: Analizar cualitativamente el algoritmo y dar conclusiones importantes para el objetivo del negocio. Solución: Investigar sobre el algoritmo y apoyarme de conocimiento fuera del contenido de la clase en internet	35	12 Horas
Lindsay Pinto	Líder de datos Líder de negocio	**Algoritmo realizado:** Naive Bayes **Trabajo:** - Entendimiento y limpieza de datos - Pregunta 2 del informe - Pregunta 3 y 4 del informe	Reto: Enfrentarse a una nueva tarea de aprendizaje de la cual no tenía mucho conocimiento Solución: Buscar documentación y apoyo con mis compañeros	34	12 Horas
Yei Hong Zhang	Líder de analítica Líder de negocio	**Algoritmo realizado**: Random Forest Classifier **Trabajo:** - Limpieza de datos	Reto: Establecer horarios de trabajo para investigación y creación del modelo.	31	10 Horas

		- Contacto con persona de estadística - Pregunta 1 y 5 del informe - Pregunta 3 y 4 del informe	Solución: Crear checklist y seguir el seguimiento planteado por el equipo.		

6.1 Puntos que mejorar para la siguiente entrega

Respecto al modelo a pesar de implementar un stemmer que en algunos casos convierte las palabras de diferentes idiomas con el mismo significado en el mismo token, no sucede en todos los casos lo que puede tener implicaciones negativas en la efectividad del modelo. En la siguiente entrega se podría intentar implementar un paso adicional en el tratamiento de datos para normalizar estas palabras.

Uno de los puntos a mejorar para la siguiente entrega es la revisión de los entregables y del proyecto, dándole más tiempo y programando una reunión para complementar la comprensión de cada uno de nuestros compañeros sobre su entrega.

Debemos continuar con el apoyo que nos hemos brindado, pero mejorar la comunicación para poder responder de manera más rápida a las inquietudes de nuestros compañeros.

Bibliografía

Dubey, A. (7 de Septiembre de 2018). *What is the difference between SGD classifier and the Logisitc regression?* Obtenido de DataScience StackExchange: https://datascience.stackexchange.com/questions/37941/what-is-the-difference-between-sgd-classifier-and-the-logisitc-regression

Keepcoding. (17 de 02 de 2023). *Keepcoding*. Obtenido de Qué es el TF-IDF Vectorizer: https://keepcoding.io/blog/que-es-el-algoritmo-tf-idf-vectorizer/#:~:text=TF%2DIDF%20Vectorizer%20(Term%20Frequency,forma%20parte%20de%20un%20corpus.

Maina, S. (25 de Febrero de 2021). *Lasso, Ridge, and Elastic-net Regularization For Preventing Overfitting in Machine Learning*. Obtenido de Towards Datascience: https://towardsdatascience.com/preventing-overfitting-with-lasso-ridge-and-elastic-net-regularization-in-machine-learning-d1799b05d382

NLTK. (2023). *NLTK*. Obtenido de nltk.stem.snowball module: https://www.nltk.org/api/nltk.stem.snowball.html

¿Qué es un árbol de decisión? | IBM. (s. f.). https://www.ibm.com/es-es/topics/decision-trees