

UNIVERSIDAD DE LOS ANDES
DEPARTAMENTO DE INGENIERÍA DE
SISTEMAS Y COMPUTACIÓN



PROYECTO: ETAPA 1

ISIS 3301 – Inteligencia de Negocios

Carlos Diaz - 202210262

Juan Esteban Quiroga - 202013216

Andres Ortiz - 201727662













Tabla de contenido:

Objetivo.....	3
1. Documentación del proceso de aprendizaje automático.....	3
2. Entendimiento y preparación de los datos.....	4
2.1. Calidad de texto:.....	4
2.2. Tokenización:.....	5
2.3 Vectorización y balanceo de datos.....	6
3. Modelado y evaluación.....	6
3.1. Objetivo y tarea de aprendizaje.....	6
3.2. Búsqueda de hiperparámetros.....	6
3.3 Evaluación de modelos.....	7
4. Resultados.....	8
4.1. Palabras clave.....	8
4.2. Resultados cuantitativos.....	9
4.3. Resultados cualitativos.....	9
5. Trabajo en equipo.....	10

Objetivo

El objetivo de este proyecto es ayudar a la UNFPA a desarrollar una aplicación analítica capaz de categorizar las opiniones de la ciudadanía en tres ODS: **1 (Fin de la pobreza)**, **2 (Salud y bienestar)**, y **3 (Educación de calidad)**. El modelo debe ser integrado en una aplicación web o móvil y que pueda ser utilizada de forma continua y que sea re-entrenable con nuevos textos naturales.

1. Documentación del proceso de aprendizaje automático

TAREA DE APRENDIZAJE  Tipo de tarea: Aprendizaje supervisado de clasificación Predicción: Clasificar opiniones de ciudadanos de acuerdo a los términos ODS (tipo 1, tipo 3, y tipo 4) relacionados con "Fin de la pobreza", "Salud y bienestar" y "Educación de calidad" Resultados: La categorización de cada opinión en uno de los términos ODS Observar: El resultado se obtiene en segundos (dependiendo de la cantidad de datos)	DECISIONES  Las predicciones del modelo permiten generar recomendaciones que vinculan opiniones de la ciudadanía con políticas correspondientes a cada término ODS. Esto puede facilitar la toma de decisiones en la planificación de estrategias para atacar las iniciativas.	PROPUESTA DE VALOR  El beneficiario final es la UNFPA y las organizaciones que trabajan para lograr las metas de la Agenda 2030. El problema es que los datos proporcionados por los ciudadanos son muy grandes lo que causa que el análisis y clasificación manual requiere muchos recursos y expertos especializados. El modelo puede automatizar esta labor y reducir el tiempo de análisis considerablemente pero siempre habrá riesgo de que algunas opiniones no queden clasificadas correctamente.	RECOLECCIÓN DE DATOS  	FUENTES DE DATOS  Los datos que se utilizarán en este proyecto vienen de opiniones textuales proporcionados por los ciudadanos respecto ODS 1,3, y 4.
SIMULACIÓN DE IMPACTO  La automatización del análisis textual reduce el tiempo de labor humano y ayuda a disminuir recursos ayudando a que se puedan destinar para otros propósitos. Sin embargo, el costo es que puedan haber clasificaciones erróneas por parte del modelo que pueden afectar la asignación de recursos públicos para este proyecto. Para medir el éxito del modelo, usaremos las métricas F1-score ya que refleja equilibrio entre precisión y recall.	APRENDIZAJE (USO DEL MODELO)  Si tenemos en cuenta que el ciudadano proporciona su respuesta y que esa sea su única función, podríamos decir que el usuario final en realidad es el funcionario y no el ciudadano. En este caso, el modelo utilizará procesamiento por lotes para poder entregar informes agregados. El modelo se puede ejecutar periódicamente ya que la clasificación en tiempo real de cada respuesta no es una prioridad dado que las decisiones del gobierno no se basan en un solo texto sino en tendencias agregadas de varios ciudadanos. El uso sería por medio de una aplicación web donde ciudadanos tendrían una ventana de tiempo (tal vez en una escala de tiempo semanal)		CONSTRUCCIÓN DE MODELOS  Este proyecto hará uso de 3 algoritmos diferentes en la fase experimental para clasificar texto: 1. Regresión logística 2. Máquina de soporte vectorial 3. Naive Bayes En la app final, solo uno será seleccionado haciendo una evaluación comparativa entre el mejor F1-score. Para generar el modelo toca procesar datos, vectorizar, entrenar, y comparar. Este tiempo está en escala de días pero en un escenario real no debería superar una escala de horas.	INGENIERÍA DE CARACTERÍSTICAS  Para poder analizar los textos toca realizar procesos de tokenización, lematización, y eliminación de stop words. Después de este preprocesamiento toca vectorizar el texto con TF-IDF. Con esto podemos hacer el análisis de palabras más influyentes categorizadas por ODS.
	MONITOREO 			

2. Entendimiento y preparación de los datos

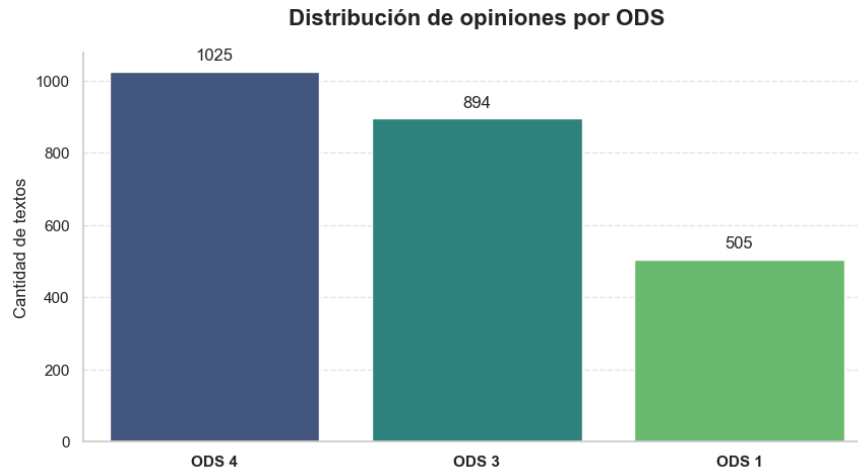
Se inició importando librerías como pandas, seaborn, matplotlib, nltk, re, spacy, sklearn, imblearn, entre otras, para procesos como la creación de gráficos, procesamiento de lenguaje, preparación, modelamiento y evaluación del modelo. Lo siguiente fue definir la ruta del archivo, donde se hizo una lectura de la misma con un manejo de errores, usando try para importar el archivo si la ruta es correcta y except para detectar e imprimir un mensaje de advertencia si la ruta no lo es. Después de eso se incluye un if para imprimir los primeros 5 datos del dataset e información parcial con la función info, a partir de lo cual se deduce lo siguiente:

Tamaño del dataset: 2424

Total de variables: 2

- **textos (object):** Opiniones de la ciudadanía
- **labels (int64):** Objetivos de Desarrollo Sostenible u ODS (*1: Fin de la pobreza, 3: Salud y bienestar, 4: Educación de calidad*)

Con las librerías de seaborn y matplotlib se creó un gráfico para observar la distribución de los textos por objetivo.



Se observa como el objetivo 4, relacionado con educación, tiene asociados la mayor cantidad de textos, más de 1000, que representa el 42.2% de todos los textos recolectados para el proyecto. Mientras que el objetivo 1, relacionado con el fin

de la pobreza, fue el que menos textos tenía. Esto propone un desbalance en las clases de, por lo que más adelante se puede ser problemático para la clasificación.

2.1. Calidad de texto:

De entrada se observó dos aspectos de la calidad de datos, la nulidad y la duplicidad, con las funciones isnull y duplicate se hizo un conteo de los datos nulos y duplicados tanto en conjunto como por columna, donde no se encontró ninguna anomalía.

2.2. Tokenización:

Se inició descargando la lista de stopwords, donde posteriormente se creó la función limpiar_texto para hacer los siguientes cambios: pasar todo el texto a minúscula, eliminar signos de puntuación, tildes, espacios repetidos. También con la función token.lenma_ se incluyó la lematización de los textos omitiendo espacios, stopwords y que los textos tengan más de la palabra. Imprimiendo un nuevo dataset con 4 columnas (labels, texto, texto_limpio, tokens) y se construyó un nuevo dataset con base en la columna tokens para redistribuir cada elemento de la lista de la columna tokens

en una fila independiente.

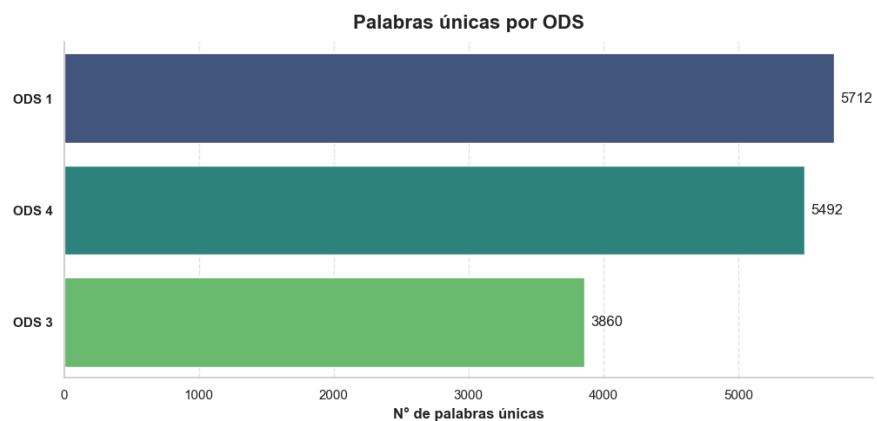
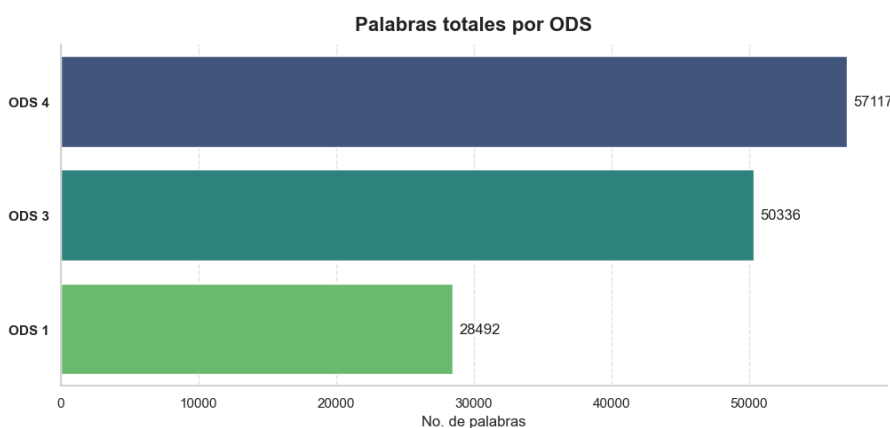
Luego se construyeron los siguientes 3 gráficos que resumen la exploración de los datos:

Es este primer gráfico se observa como los textos referentes al

objetivo 4 son los que contienen mayor cantidad de palabras, con 57117, seguido del ODS 3 con 50336 y por último el ODS 1 con casi 3000, lo que indica que se mantiene la proporción de palabras con respecto a la cantidad de textos por objetivo.

El siguiente gráfico, por el contrario, muestra la cantidad de palabras únicas por cada ODS, y del cual se establece que los textos relacionados con el objetivo 1 a pesar de que son los que menos palabras usan, a la vez son los que menos palabras repiten.

Otro de los aspectos que se revisó, fue la longitud de los textos. Se encontró que los clasificados en el ODS 1 tienen una media de 56,42 palabras, en el ODS 3 el promedio



es de 56,3 y en el ODS 4 de 55,72 palabras, y todos cuentan con una desviación estándar aproximadamente de 18, lo que indica que no hay mucha variación entre el largo de cada texto.

	labels		token	count
labels				
1	2834	1	pobreza	1004
	3386	1	ser	777
	2016	1	ingreso	363
	2739	1	país	351
	2832	1	pobre	313
3	8859	3	ser	1079
	8770	3	salud	1063
	4330	3	atención	820
	6476	3	haber	546
	8866	3	servicio	464
4	14357	4	ser	1165
	11559	4	escuela	888
	11346	4	educación	886
	11659	4	estudiante	758
	12112	4	haber	567

También, se hizo un conteo y se obtuvo el top 5 de palabras más usadas en cada ODS. En el ODS 1 sobresalen términos como *pobreza*, *ingreso* y *país*, asociados a problemáticas económicas y sociales. Para el ODS 3 destacan *salud*, *atención* y *servicio*, que reflejan discusiones en torno al acceso y calidad del sistema sanitario. Mientras que en el ODS 4 aparecen con mayor frecuencia *escuela*, *educación* y *estudiante*, vinculados al ámbito académico. Lo anterior infiere como los textos

concuerdan con las temáticas de cada ODS.

2.3 Vectorización y balanceo de datos

El texto en cada fila del dataset fue transformado en vectores numéricos por medio de TF-IDF con un vocabulario de las 5000 palabras más frecuentes que nos permitió representar la relevancia de las palabras y combinaciones para cada uno de los ODS.

Luego, se dividieron los datos con 80% para entrenamiento y 20% para prueba y finalmente se aplicó SMOTE al conjunto de entrenamiento para poder lograr un balance entre las palabras con el fin de evitar favoritismo del modelo hacia las mayorías. Esto logra la capacidad de generalización del modelo.

3. Modelado y evaluación

3.1. Objetivo y tarea de aprendizaje

El objetivo de estos modelos es realizar tareas de clasificación sobre las opiniones de los ciudadanos en las tres diferentes categorías ODS. Las tareas son supervisadas, ya que partimos desde un set de datos cuyos labels se conocen. → La salida de las tareas es un label discreto {1, 3, 4} (mencionado en la sección previa).

3.2. Búsqueda de hiperparámetros

Para optimizar el desempeño de los modelos, aplicamos GridSearchCV con 5-fold cross-validation y escogimos F1-weighted como la métrica de puntuación para cada modelo. Esto aseguró que la búsqueda tuviera un balance entre precisión y recall entre las categorías. Además, cada pipeline hizo uso de SMOTE para manejar categorías minoritarias.

Por simplicidad, solamente exploramos un hiperparámetro para cada modelo:

- **Regresión Logística:** C (inversa de regularization strength). Controla que tanto el modelo penaliza coeficientes grandes. Un valor bajo de C implica regularización más fuerte para un modelo más simple con menos sobreajuste mientras que un valor alto de C implica regularización más débil para un modelo más flexible pero con más riesgo de sobreajuste. **Grid: [0.1, 1, 10, 100]**
- **Naïve Bayes:** α (Laplace smoothing). Determina qué tanta probabilidad se le agrega a palabras poco vistas en las respuestas de los ciudadanos. Un valor muy bajo de α corre el riesgo de sobreajuste y probabilidades nulas, mientras que un valor muy alto de α causa que el modelo sea muy uniforme, causando la pérdida de poder predictivo. Para este proyecto, este modelo requiere un α ajustado tal que balancea palabras exóticas sin aplastar tanto la frecuencia de palabras comunes. **Grid: [0.1, 0.5, 1.0]**
- **Linear Support Vector Classifier (LinearSVC):** C (parámetro de penalización de clasificación errónea). Este controla el beneficio entre maximizar el margen y permitir errores de clasificación. Un valor bajo de C permite un margen más grande, tiene mayor sesgo, pero mejor generalización. Un valor muy alto de C tiene un margen de error más pequeño, tiene menor sesgo, pero con riesgo de sobreajuste. **Grid: [0.1, 1, 10, 100]**

Abajo se detallan los hiperparámetros elegidos:

```
Fitting 5 folds for each of 4 candidates, totalling 20 fits
Mejor F1-score para LR: 0.9762
Mejores parámetros: {'classifier__C': 100}

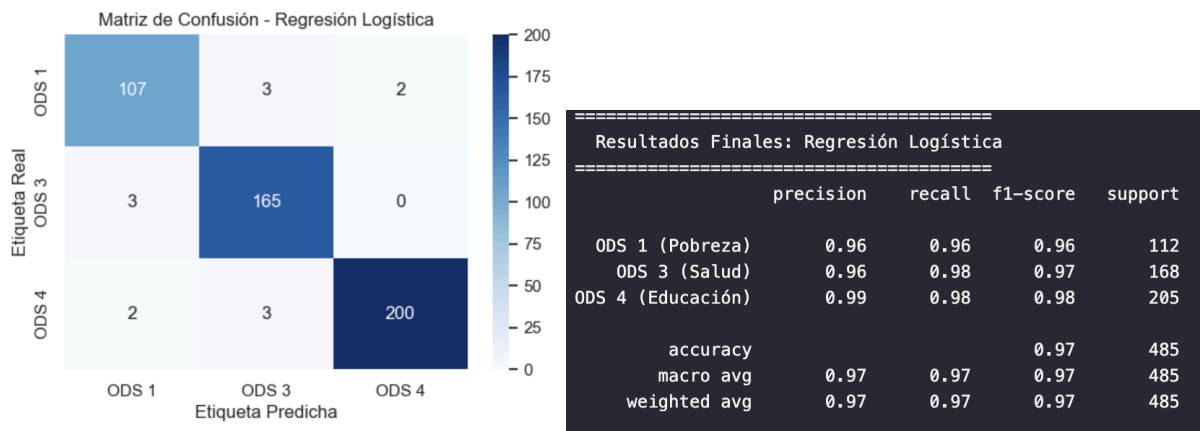
--- Iniciando GridSearchCV para Naive Bayes ---
Fitting 5 folds for each of 3 candidates, totalling 15 fits
Mejor F1-score para NB: 0.9683
Mejores parámetros: {'classifier__alpha': 0.5}

--- Iniciando GridSearchCV para SVM (LinearSVC) ---
Fitting 5 folds for each of 4 candidates, totalling 20 fits
Mejor F1-score para SVM: 0.9788
Mejores parámetros: {'classifier__C': 1}
```

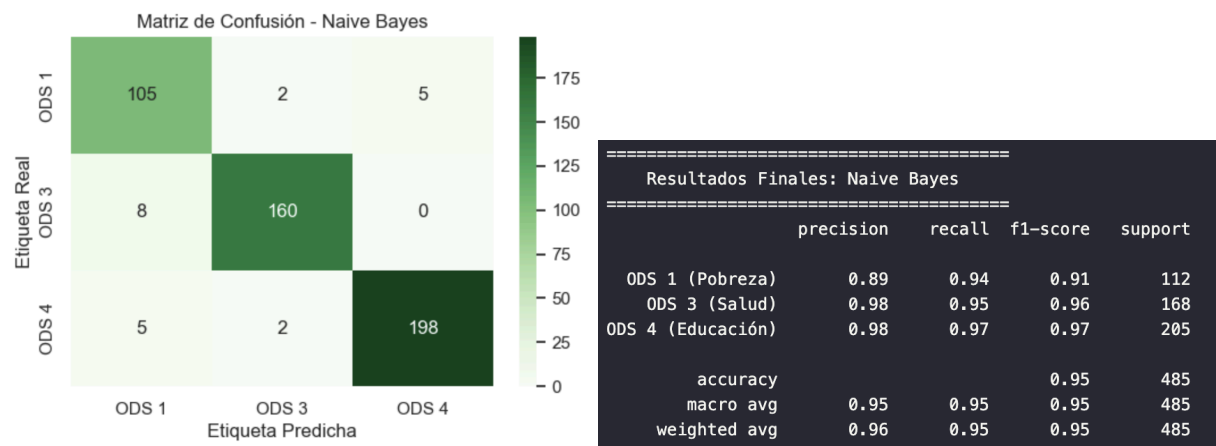
3.3 Evaluación de modelos

Se entrenaron y se evaluaron los tres modelos con SMOTE aplicado en los folds de entrenamiento para manejar datos no balanceados y estos fueron los resultados para cada modelo:

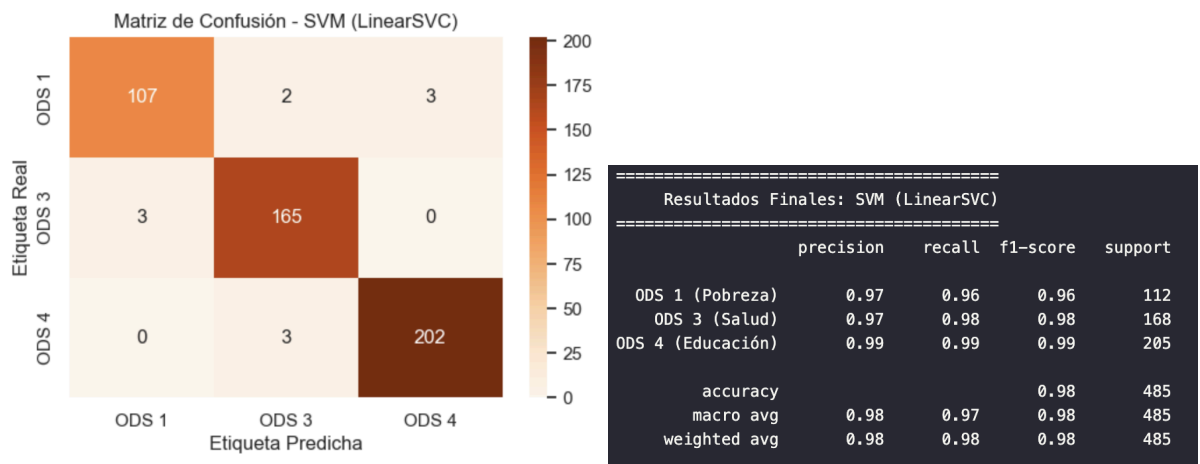
Regresion Logistica:



Naïve Bayes:



SVM (LinearSVC)



LinearSVC logró el F1-score más alto, demostrando que es un modelo robusto y el mejor candidato para el modelo final.

4. Resultados

A continuación vamos a analizar las características del mejor modelo encontrado **LinearSVC**. Características como su score F1, y también vamos a analizar las palabras "clave" o aquellas que el modelo reconoce como más influyentes para la clasificación de cada texto.

4.1. Palabras clave

Ahora pues, veremos las palabras clave más influyentes para la clasificación de los textos en cada uno de los objetivos 1, 3 y 4 de ODS de las Naciones Unidas.

Estas palabras están en orden de mayor a menor importancia:

- **ODS 1:** *pobreza, pobre, hogar, ingreso, social, privación, protección social, protección, empleo, crecimiento, niño, transferencia, umbral, vivir, familia, multidimensional, urbano, pobreza infantil, trabajo, vivienda.*
- **ODS 3:** *salud, atención, médico, enfermedad, paciente, sanitario, mental, mortalidad, tratamiento, alcohol, hospital, droga, servicio, medicamento, consumo, atención primario, salud mental, hospitalario, clínico, muerte.*
- **ODS 4:** *educación, escuela, estudiante, educativo, docente, aprendizaje, escolar, alumno, profesor, habilidad, enseñanza, evaluación, maestro, formación, superior, universidad, sistema educativo, personal, profesional, pisa.*

Esta lista de palabras muestra un correcto funcionamiento del modelo, al no mostrar como palabras clave *stopwords*, y también al mostrar con mucho sentido que para la categoría *ODS 1 (Fin de la pobreza)* relaciona como más relevantes palabras como **pobreza** e **ingreso**, mientras que para la categoría *ODS 3 (Salud y bienestar)* identifica palabras como **salud**, **médico** y **paciente** como aquellas con una mayor relevancia para hacer esta categorización.

No solo muestra que procesos como *Tagging* (asignarle a cada palabra una categoría gramatical para aumentar la información que se tiene acerca de su propósito y sentido en la frase) no serían necesarios, pues el modelo no lo necesita; solo necesita entender a qué categoría pertenecen los textos.

4.2. Resultados cuantitativos

He aquí volveremos a mostrar cuál fue el desempeño del modelo respecto a las métricas accuracy, precision, recall y F1.

	Precisión	Recall	Recall	Soporte
ODS 1 (Pobreza)	0.96	0.96	0.96	101
ODS 3 (Salud)	0.98	0.97	0.97	179

ODS 4 (Educación)	0.97	0.98	0.98	205
*Accuracy			*0.97*	485
*Macro avg	0.97	0.97	0.97	485
*Weighted avg	0.97	0.97	0.97	485

Aquí podemos ver la alta efectividad de nuestro modelo para la clasificación de las tres categorías del ODS que nos aplican a nuestro caso. Mostrando así que este modelo podría ser puesto en producción del negocio si así lo eligiera.

4.3. Resultados cualitativos

En esta sección analizaremos algunos textos en los que el modelo puso erróneamente la categoría para ver sus limitaciones y puntos destacables.

- *“La educación médica continua (CME) y el desarrollo profesional continuo son los mecanismos más conocidos, pero los países de la OCDE también han introducido diferentes formas de evaluación profesional, revisión por pares...”*

Para este caso, incluso un humano vería con dificultad si clasificarlo dentro del ODS 3 (Salud) o ODS 4 (Educación), pues habla bastante sobre educación y muchas de las palabras están en efecto muy relacionadas a educación. Nuestro modelo la clasificó en ODS 4, aunque la etiqueta real era ODS 3.

- *“En el sur de Europa, la República Checa y Polonia, más del 30 % de los cuidadores familiares prestan cuidados intensivos, y el porcentaje es aún mayor en España (más del 50 %) y Corea (más del 60 %). Aunque los cuidados pueden aliviar el riesgo de pobreza...”*

En este caso, la etiqueta real era ODS 1 (Pobreza), pero el modelo la clasificó en ODS 3 (Salud). Si leemos estas primeras líneas, vemos que es muy difícil distinguir las categorías, incluso para un humano.

De estos ejemplos podemos concluir que el modelo puede tener dificultades para clasificar textos donde se menciona más de una categoría. Esto es consistente, puesto que existe una alta correlación entre **educación y bienestar**, o entre **pobreza y acceso a la educación**. Para enfrentar este problema se podría pensar en un clasificador *One-vs-One*, que daría un mayor control o certeza frente a estas situaciones.

5. Trabajo en equipo

Roles y tareas:

- **Líder de proyecto:** Juan Esteban Quiroga
 - Coordinó reuniones, cronograma y entrega final.
 - Horas: 10

- Algoritmo: Naïve Bayes
- Retos: Gestión de tiempo
- Uso de ChatGPT: Apoyo en Redacción
- **Líder de negocio:** Andres Ortiz
 - Aseguró alineación con los ODS y la claridad de los resultados
 - Horas: 8
 - Algoritmo: Regresión logística
 - Retos: Interpretación de resultados
 - Uso de ChatGPT: Validación conceptual, apoyo en informe final, teoría del modelo
- **Líder de datos:** Carlos Diaz
 - Preparó y gestionó los datos, evaluó los modelos, y comparó métricas
 - Horas: 12
 - Algoritmo: LinearSVC
 - Uso de ChatGPT: Explicación de métricas, apoyo en desarrollo de funciones auxiliares

Distribución de puntos:

- Carlos Díaz: 35
- Andres Ortiz: 35
- Juan Esteban Quiroga: 35

Reuniones realizadas:

- Lanzamiento y planeación (x1)
- Seguimiento semanal (x1)
- Entrega final (x1)

Puntos de mejora:

- Mejorar gestión de tiempos con más antelación, documentación más detallada, y mayor interacción entre roles.