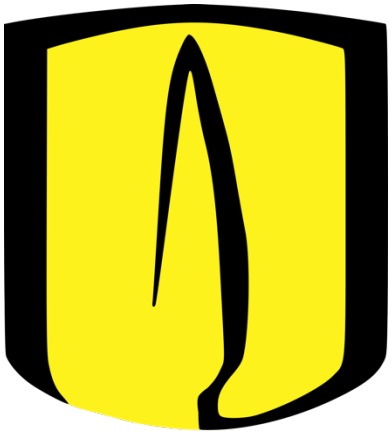
Inteligencia de Negocios – ISIS3301



Proyecto 1 – Etapa 1
Analítica de textos

Grupo 6
Valentina Lara - 201912967
Manuela Lizcano–202122826
Natalia Ricaurte Pacheco – 201914101

Table of Contents

1. Entendimiento del negocio y enfoque analítico. 3

2. Entendimiento y preparación de los datos..... 3

2.1. Entendimiento 3

2.2. Preparación de los Datos 3

3. Modelado y evaluación. 4

3.1. Modelo KNN Vecinos (Valentina Lara) 4

3.2. Modelo Árboles de Decisión (Manuela Lizcano)..... 5

3.3. Modelo Random Forest (Natalia Ricaurte)..... 5

3.4. Modelo Support Vector Machines (Natalia Ricaurte) 6

3.5. Modelo de Regresión Logística (Manuela Lizcano) 7

3.6. Modelo MultinomialNB (Valentina Lara) 7

3.7. Evaluación de Métricas de manera generalizada..... 8

4. Resultados..... 8

4.1. Descripción de los resultados: métricas de calidad y objetivos del negocio. 8

4.2. Análisis de palabras identificadas y posibles estrategias que la organización.....10

5. Mapa de actores relacionado con el producto de datos creado.....11

6. Trabajo en equipo 11

7. Referencias.....13

Proyecto Etapa 1

1. Entendimiento del negocio y enfoque analítico.

Oportunidad/problema Negocio	Analizar información textual recopilada dado los grandes volúmenes de opiniones ciudadanas y los pocos recursos para identificar y relacionar de manera automática estas mismas con los ODS,3,4,5.
Objetivos y criterios de éxito desde el punto de vista del negocio.	Optimizar el proceso y minimizar los tiempos de análisis de las opiniones ciudadanas.
	Automatizar las opiniones con la identificación de los ODS relacionados para incrementar la precisión.
	Incremento en la velocidad de procesamiento de información textual.
Organización y rol dentro de ella que se beneficia con la oportunidad definida	La organización es el Fondo de Poblaciones de las Naciones Unidas. Esta se beneficia a través de sus analistas de datos ya que son los responsables de interpretar y actuar sobre las opiniones ciudadanas. Por lo que, los gestores de los programas tendrán una mayor precisión en los datos y nuevos recursos para en análisis de información textual
Impacto que puede tener en Colombia este proyecto.	Este proyecto abarca múltiples factores de los cuales Colombia puede verse beneficiado. En primer lugar, estos modelos permiten desarrollar una optimización de procesos gubernamentales y sociales relacionado con los objetivos de desarrollo sostenible: salud y bienestar, educación de calidad y igualdad de género. Este tipo de herramientas permite lograr garantizar una mejor conexión con los ciudadanos. Por último, estos modelos analíticos son una herramienta de gran ayuda para interpretar grandes volúmenes de texto, para facilitar la toma de decisiones.
Enfoque analítico. Descripción de la categoría de análisis, tipo y tarea de aprendizaje e incluya las técnicas y algoritmos que propone utilizar.	La categoría de análisis del contexto dado por el proyecto hace parte del análisis descriptivo, considerando que es necesario identificar: clasificar las opiniones sobre los ODS, temas predominantes, etc. El tipo de aprendizaje es un aprendizaje automático, considerando que los datos son estructuras con datos no etiquetados. Las tareas de aprendizaje es la clasificación, que asigna etiquetas a los comentarios según su contenido.

2. Entendimiento y preparación de los datos.

2.1. Entendimiento

En primer lugar, se realizó una exploración inicial del DataFrame utilizando la función datos.info(), lo cual permitió identificar la estructura de los datos. El DataFrame contenía 4049 filas y 2 columnas: Textos_espanol (de tipo object) y sdg (de tipo int64). Ninguna de las columnas presentaba valores nulos, lo que indicaba que los datos estaban completos. Posteriormente, se utilizó datos.describe() para obtener un resumen estadístico de la columna sdg, que clasifica los textos dentro de los Objetivos de Desarrollo Sostenible (ODS). El análisis mostró una media de 4.05 y un rango de valores entre 3 y 5, lo que indicaba que los textos estaban asociados a un conjunto limitado de ODS. Para un análisis más detallado de los textos, se añadieron tres columnas adicionales:

Conteo: Se calculó el número de caracteres en cada texto utilizando len(x) para cada entrada en la columna Textos_espanol.
Max: Se determinó la longitud de la palabra más larga en cada texto, separando las palabras de cada entrada y encontrando la longitud máxima.
Min: De manera similar, se calculó la longitud de la palabra más corta en cada texto

2.2. Preparación de los Datos

Para la preparación de los datos, se comenzó eliminando filas duplicadas y confirmando que no había ninguna en el conjunto. Se verificó la unicidad de los valores en cada columna,

encontrando que la columna `Textos_espanol` contenía 4049 textos únicos y la columna `sdg` tenía 3 categorías correspondientes a los Objetivos de Desarrollo Sostenible. Además, se realizó un análisis de balanceo de clases para asegurar que cada clase estuviera representada de manera equitativa, evitando así que el modelo se sesgara hacia las clases más frecuentes. Se observó que los datos estaban balanceados. Por otra parte, la limpieza de los textos incluyó la conversión de caracteres a minúsculas y la eliminación de signos de puntuación para reducir el ruido. Se procedió a la tokenización de los textos, separando cada uno en palabras individuales, y se aplicó lematización para reducir las palabras a su forma base. Además, se desarrolló una función para eliminar stopwords, utilizando un conjunto de palabras comunes en español que no aportan significado relevante para el análisis. La función `remove_stopwords` filtra estas palabras, mejorando la calidad del texto procesado al enfocarse en términos más significativos.

Después del preprocesamiento, se transformaron las palabras tokenizadas en cadenas de texto unidas, lo que facilitó el trabajo con los textos procesados. Se separaron las variables independientes (`words1`) y dependientes (`sdg`) para su análisis. Finalmente, se aplicó la técnica de vectorización de textos, comenzando con `CountVectorizer` para convertir las palabras en una matriz de frecuencias de término, asegurando que los datos estuvieran listos para un análisis más detallado.

3. Modelado y evaluación.

3.1. Modelo KNN Vecinos (Valentina Lara)

En el proceso de modelado, se empleó el clasificador KNN vecinos para la tarea de clasificación. El método de vecinos más cercanos (KNN) se basa en encontrar las muestras de entrenamiento más próximas a un nuevo punto para predecir su etiqueta. Es un algoritmo no paramétrico que funciona tanto para clasificación (etiquetas discretas) como para regresión (etiquetas continuas). KNN puede usar un número fijo de vecinos o ajustarse según la densidad de los datos (Scikit learn, s.f.).

Primero, se extrajeron las etiquetas de los datos (sdg) y se preparó la matriz TF-IDF (X) como las características de entrada. Luego, se dividió el conjunto de datos en entrenamiento y prueba utilizando una proporción del 80% para entrenamiento y el 20% restante para prueba, con un random_state fijado para reproducibilidad.Se entrenó el clasificador KNN con los datos de entrenamiento (X_train y y_train) y se realizaron predicciones sobre el conjunto de prueba (X_test). La calidad del modelo se evaluó mediante la matriz de confusión, que se visualizó usando ConfusionMatrixDisplay, para observar el desempeño del clasificador en términos de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. (Matriz de confusión en 4. Resultados). Para optimizar el modelo, se realizó un ajuste de hiperparámetros utilizando GridSearchCV. En particular se ajustó el número de vecinos. Este ajuste permitió seleccionar la combinación de hiperparámetros que ofreciera el mejor rendimiento en términos de precisión. Mejores hiperparámetros encontrados: {'n_neighbors': 13}. Resultados obtenidos de KNN y KNN ajuste hiperparámetros:

	KNN-VECINOS	KNN-VECINOS AJUSTE HIPERPARAMETROS
Exactitud	0,950	0,960
Recall	0,949	0,962
Precisión	0,949	0,962
Puntuación F1	0,940	0,962

3.2. Modelo Árboles de Decisión (Manuela Lizcano)

El funcionamiento del modelo de clasificación por medio de árboles de decisión se realiza tomando como analogía la forma de un árbol. En primer lugar, igual que las ramas de un árbol, los datos recursivamente se separan en múltiples clases, esto teniendo en cuenta las características de los datos. Esta transformación de los datos se hace utilizando una medida de impureza, para este contexto usaremos entropía. Después de construir el árbol, se debe realizar la clasificación de los datos, donde cada nueva instancia inicia desde la raíz del árbol, y después cada instancia sigue un camino hasta la hoja (clase asignada). Viendo este modelo aplicado al contexto del proyecto, iniciamos definiendo la variable objetivo, la cual queremos llegar a predecir. En nuestro caso es sdg. Seguido de esto, segmentamos los datos en dos conjuntos: conjunto entrenamiento y conjunto de prueba. El árbol inicialmente es construido con la medida de impureza de entropía, y se debe definir la semilla que establece la aleatoriedad del árbol.

El árbol creado es entrenado por medio del conjunto de entrenamiento (X_train y y_train), los cuales ajustan el modelo, con el objetivo de que los datos se clasifiquen en las hojas (clases) del árbol. Para poder visualizar el comportamiento de clasificación del árbol, construimos herramientas visuales como la matriz de confusión la cual nos permite identificar el desempeño general del modelo, por medio de la comparación de la forma en que el árbol clasifico las predicciones y las etiquetas que fueron usadas para entrenar al modelo. (Matriz de confusión en 4. Resultados). Además, para mejorar el rendimiento del modelo, se realizó un ajuste de hiperparámetros mediante GridSearchCV. En particular, se ajustaron parámetros como el criterio de impureza y la profundidad máxima del árbol. El ajuste permitió encontrar los mejores hiperparámetos para el modelo {'criterion': 'entropy', 'max_depth': 25}. Lo que ayudo a maximizar la precisión y mejorar el rendimiento general del árbol en la tarea de clasificación. Resultados obtenidos de Árboles de decisión y Árboles de decisión ajuste hiperparámetros:

	ARBOLES DE DECISIÓN	ARBOLES DE DECISIÓN AJUSTE HIPERPARAMETROS
Exactitud	0,920	0,920
Recall	0,922	0,921
Precisión	0,922	0,921
Puntuación F1	0,922	0,921

3.3. Modelo Random Forest (Natalia Ricaurte)

El modelo Random Forest es un algoritmo de ensamblado que se enfoca en la creación de múltiples árboles de decisión para realizar tareas de clasificación o, ciertos casos, de regresión. Este fue un modelo de selección debido a su capacidad de mejorar la precisión y la robustez mediante la unión de distintos arboles de decisión independientes. A diferencia de un único árbol de decisión el cual es un modelo de aprendizaje automático que utiliza

divisiones binarias para basar las reglas de decisión de los datos que puede ser propenso al sobreajuste, esto es significativo ya que en Random Forest reduce el riesgo de promediar las predicciones de varios árboles, lo que resulta en un modelo más estable y preciso, pues este aprovecha la estructura básica del árbol, pero expandiéndolo a un bosque de decisiones. (scikit-learn,s.f.)

En Random Forest, cada árbol hace una predicción y la clase final se selecciona por votación de mayoría, lo que asegura que los errores de un solo árbol no afecten demasiado el análisis de decisiones final. En el contexto del proyecto se utiliza para predecir la variable objetivo que representa los ODS. En este modelo cada árbol del bosque es entrenado con una muestra aleatoria de datos de entrenamiento, donde se seleccionó el modelo debido a que es robusto ante el sobreajuste y maneja bien los conjuntos de datos complejos, este se crea a partir RandomForestClassifier(), a partir de este se entrena el modelo con los datos de entrenamiento(x_train y y_train) y luego para hacer predicciones sobre el conjunto de prueba y de esta forma evaluar el modelo con las métricas de rendimiento y la visualización de la matriz de confusión para observar cómo el modelo clasificaba correctamente o incorrectamente las diferentes clases. Finalmente, se realizó un ajuste de hiperparámetros utilizando GridSearchCV, lo que permitió encontrar la mejor combinación de parámetros para optimizar el rendimiento del modelo. En este caso los hiperparámetros ajustados incluyeron el número de árboles en el bosque y la profundidad máxima de los árboles los valores fueron: {'max_depth': 50, 'n_estimators': 300}. Resultado:

	RANDOM FOREST	RANDOM FOREST AJUSTE HIPERPARAMETROS
Exactitud	0,980	0,980
Recall	0,975	0,975
Precisión	0,975	0,975
Puntuación F1	0,975	0,975

3.4. Modelo Support Vector Machines (Natalia Ricaurte)

Es una técnica de clasificación que busca encontrar un hiperplano que separe las clases de datos de la mejor manera posible. Para este proyecto SVM. Se le seleccionaron por la capacidad de manejar problemas de clasificación complejos y de alta dimensionalidad. Este modelo se enfoca en maximizar el margen entre las clases, esto se explica a través de la selección de hiperplano que tenga mayor distancia entre las instancias distintas, lo que genera robustez en cuanto a datos ruidosos y difíciles de separar. En este caso para brindar una mayor precisión en la clasificación, ya que se centraliza en los resultados de alta exactitud (Pérez, 2024).

Para el proyecto, el modelo SVM se utilizó para predecir la variable objetivo sdg a partir del conjunto de datos de entrenamiento a partir de la optimización de la ubicación del hiperplano que separa las clases de ODS. Para mejorar los resultados obtenidos y la precisión de las predicciones se empleó GridSearchCV para ajustar los hiperparámetros, en este caso el parámetro de regularización y el tipo de kernel lo que mejoro los resultados de las métricas de calidad maximizando la capacidad del modelo para separar correctamente las clases en este problema de clasificación multiclase. Hiperparámetros: {'C': 10, 'kernel': 'rbf'}

	SVM	SVM AJUSTE HIPERPARAMETROS
Exactitud	0,980	0,980
Recall	0,980	0,982
Precisión	0,975	0,982
Puntuación F1	0,980	0,982

3.5. Modelo de Regresión Logística (Manuela Lizcano)

El modelo de regresión logística utiliza la probabilidad durante su ejecución de clasificación. Este modelo usa una combinación lineal de características de los datos para clasificar cada dato en una clase especifica. El resultado de esta combinación lineal es directamente la probabilidad de pertinencia a la clase. Esta probabilidad se obtiene con la función de sigmoide. Considerando que este problema de clasificación tiene 3 clases, escogimos usar el modelo de regresión logística, ya que este modelo es especial para uno con múltiples clases.

Para implementar este modelo, primero iniciamos dividiendo el conjunto de entrenamiento y el conjunto de prueba de los datos. Después de esto creamos la regresión logística, en donde definimos como parámetro de máximo número de iteraciones: 1000, para lograr que el modelo tenga suficientes iteraciones que permitan que converja. Después de este paso, entrenamos el modelo, usando los datos de entrenamiento. En este proceso lo que está tratando de hacer el algoritmo es ajustar los coeficientes del modelo. Los coeficientes se definen bajo un rango de 0 a 1. Sin embargo, durante el entrenamiento, el objetivo es lograr ajustar correctamente los coeficientes para minimizar la diferencia de clasificación entre las predicciones y los datos reales.

Después del entrenamiento, el modelo se utilizó para hacer predicciones en el conjunto de prueba y su rendimiento se evaluó utilizando métricas de clasificación y matriz de confusión para visualizar el desempeño del modelo en términos de verdaderos positivos, negativos y falsos positivos y negativos para proporcionar la capacidad del modelo. Finalmente se realizó el rendimiento del modelo a través de ajuste de hiperparámetros mediante GridSearchCV lo que permitió encontrar la mejor combinación de valores para parámetros como la regularización y el tipo de solver. Mejores hiperparámetros encontrados: {'C': 10, 'solver': 'liblinear'}

	REGRESIÓN LOGÍSTICA	REGRESIÓN LOGÍSTICA AJUSTE HIPERPARAMETROS
Exactitud	0,978	0,979
Recall	0,978	0,979
Precisión	0,978	0,979
Puntuación F1	0,978	0,979

3.6. Modelo MultinomialNB (Valentina Lara)

En el análisis realizado, se empleó el modelo de clasificación Naive Bayes multinomial para predecir las etiquetas de los Objetivos de Desarrollo Sostenible (ODS) a partir de una representación TF-IDF de textos. Este método suele trabajar con recuentos de palabras o vectores tf-idf. Cada clase está representada por vectores que indican la probabilidad de que una característica (como una palabra) aparezca en una muestra de esa clase. Los parámetros se estiman usando una versión suavizada de máxima verosimilitud, basada en el recuento de frecuencia relativa (scikit-learn,s.f.). Inicialmente, el modelo se entrenó y

evaluó en un conjunto de datos de prueba, obteniendo métricas de rendimiento como exactitud, recall, precisión y puntuación F1. Posteriormente, se llevó a cabo un ajuste de hiperparámetros utilizando `GridSearchCV` para optimizar el parámetro de suavizado `alpha`, que controla la regularización en el modelo Naive Bayes. La búsqueda exhaustiva de hiperparámetros permitió identificar el mejor valor para `alpha`, que resultó ser 0.1.

	MULTINOMIALNB	MULTINOMIALNB AJUSTE HIPERPARAMETROS
Exactitud	0,960	0,970
Recall	0,962	0,967
Precisión	0,964	0,967
Puntuación F1	0,963	0,967

3.7. Evaluación de Métricas de manera generalizada

	KNN-VECINOS AJUSTE HIPERPARAMETROS	ARBOLES DE DECISIÓN AJUSTE HIPERPARAMETROS	RANDOM FOREST AJUSTE HIPERPARAMETROS	SVM AJUSTE HIPERPARAMETROS	REGRESIÓN LOGÍSTICA AJUSTE HIPERPARAMETROS	MULTINOMIALNB AJUSTE HIPERPARAMETROS
Exactitud	0,960	0,920	0,980	0,980	0,979	0,970
Recall	0,962	0,921	0,975	0,982	0,979	0,967
Precisión	0,962	0,921	0,975	0,982	0,979	0,967
Puntuación F1	0,962	0,921	0,975	0,982	0,979	0,967

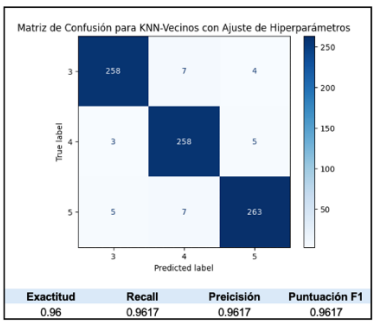
La evaluación de los modelos muestra que los algoritmos Random Forest y SVM, ambos con una exactitud del 98% son los mejores en términos de rendimiento global superando también con otras métricas como la precisión y recall a los demás modelos. Aunque KNN, la Regresión Logística y MultinomialNB ofrecen resultados competitivos, pero no los mejores y en cuanto a los árboles de decisión son el desempeño más bajo. Teniendo esto en cuenta, SVM parece ser el modelo más adecuado para poder maximizar las métricas de calidad y rendimiento lo que convierte la mejor opción para el problema de clasificación de ODS.

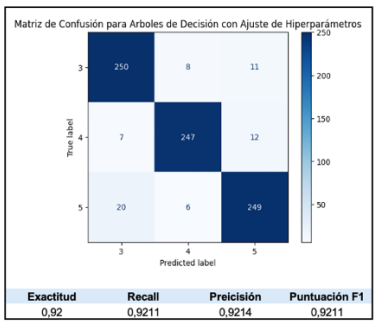
4. Resultados

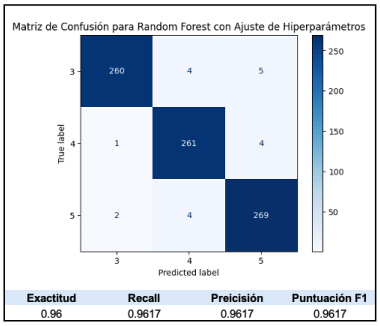
4.1. Descripción de los resultados: métricas de calidad y objetivos del negocio.

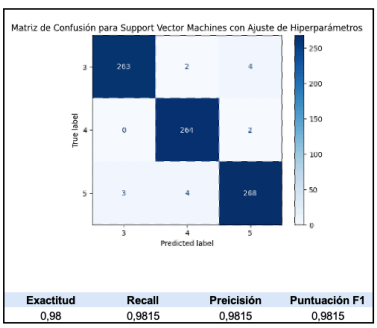
Para analizar de manera respetiva las métricas de calidad se puede observar a continuación las métricas claves, de exactitud, recall, precisión y puntuación F1. A partir de estos resultados se busca alinear con los objetivos del negocio el mejor modelo a ser seleccionado. Teniendo en cuenta que los objetivos principales se enfocan en la precisión de las categorías de los Objetivos de Desarrollo Sostenible (ODS) a partir de datos textuales dado por las opiniones que se reciben en el diario vivir.

Comparación modelos:

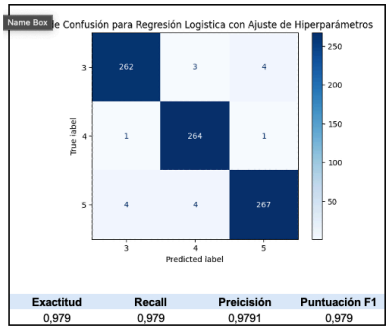




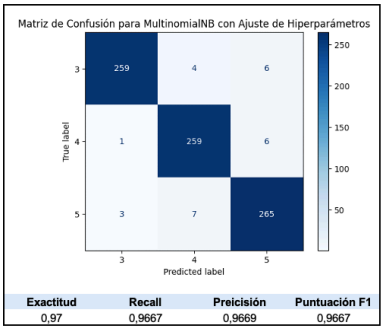




Exactitud	Recall	Preicisión	Puntuación F1
0,98	0,9815	0,9815	0,9815



Exactitud	Recall	Preicisión	Puntuación F1
0,979	0,979	0,9791	0,979



Exactitud	Recall	Preicisión	Puntuación F1
0,97	0,9667	0,9669	0,9667

En las imágenes anteriores se puede analizar el comportamiento de los modelos en donde de manera general son modelos con gran desempeño. En cuanto al modelo KNN (K-vecinos más cercanos) se puede interpretar que hay un buen desempeño con resultados equilibrados en todas las métricas, sin embargo, es un modelo que suele ser menos eficiente en termino de tiempo de ejecución lo cual puede limitar la evaluación de mayores parámetros. En cuanto al árbol de decisión es un modelo con la exactitud más baja en comparación con otros, lo cual es lo menos ideal para el objetivo del proyecto donde la precisión es crítica. Por otro lado, Random Forest es un modelo robusto y de gran rendimiento es donde es menos susceptible al sobreajuste debido al enfoque de ensamblaje manejando bien los conjuntos de datos complejos con una exactitud del 98% y otras métricas de evaluación por encima del 96%. Asimismo, en cuanto al modelo SVM uno de los modelos más fuertes en cuanto a la precisión de los problemas de clasificación complejos donde se busca maximizar el margen entre todas las clases es uno de los mejores de evaluaciones métricas todas por encima de un 98%. Finalmente encontramos los modelos de regresión logística y multinomialNB donde las métricas se encuentran por debajo del 97.9% donde son eficaces, pero pueden no ser los mejores modelos en cuanto a las relaciones complejas entre características ya que la precisión es uno de los factores que más afecta.

En cuantos a los modelos y las matrices de confusión donde podemos analizar la asertividad de cada uno de ellos. Se observa un bajo número de opiniones incorrectamente clasificadas, en cuanto a los modelos el árbol de decisión es uno de los modelos con mayor número de errores, donde las predicciones incorrectas aparecen más notables en ciertas clases lo que indica que este modelo tiene dificultades para la precisión y otros factores en cuanto a la predicción de este proyecto, siguiendo de esta manera sigue el algoritmo de KNN y multinnominal donde hay mayores errores en comparación con los otros modelos, lo que sugiere que no son modelos tan efectivos para manejar las relaciones complejas entre las características del texto y las clases ODS. En cuanto a los modelos de SVM, Random Forest y regresión logística son de los modelos más fuertes que logramos a identificar en el desarrollo del proyecto con ese orden de precisión ye exactitud.

Selección del mejor modelo

Teniendo en cuenta el objetivo del negocio que enfatiza sobre la precisión de las opiniones de los ciudadanos relacionados con los ODS, es el modelo SVM (Support Vector Machines). Tiene la mejor precisión, recall y puntuación F1 lo que garantiza una mejor exactitud y se minimicen los errores de clasificación de opiniones en las diferentes categorías de ODS.

Este es un modelo útil para maneja datos de alta dimensionalidad, como los obtenidos a partir de la representación de los textos.

Objetivos del negocio
Los objetivos encontrados se centralizan en la optimización del proceso en cuanto a la reducción de tiempos en los análisis de las opiniones de los ciudadanos permitiendo un procesamiento más ágil y eficiente. Al igual que a través de la automatización de implementar un sistema que identifique las opiniones y las asocie con la clase ODS mejora la precisión de análisis. Finalmente, en cuanto a la velocidad de procesamiento al tener un modelo efectivo se aumenta la velocidad de identificación de información textual para manejar grandes volúmenes de opiniones de forma eficiente. Por lo que se puede concluir que el modelo Suppport Vector Machines es el mejor que cumple con las expectativas del negocio ya que ofrece el mejor rendimiento en término de precisión y calidad de predicciones lo cual es significativo en la capacidad del Fondo de Poblaciones de las Naciones Unidas para interpretar de manera eficiente grandes volúmenes de datos textuales y tomar decisiones estratégicas basadas en un análisis preciso y automatizado. Eso no solo mejora la gestión de las opiniones ciudadanas, sino que también fortalece la relación entre la organización y los ciudadanos al garantizar un mejor servicio.

4.2. Análisis de palabras identificadas y posibles estrategias que la organización

Para analizar las palabras relacionadas con los ODS y desarrollar estrategias efectivas, se realizó un proceso en varias etapas. Primero, se contaron los registros en los datos asociados a cada ODS (3, 4 y 5), proporcionando una visión general de la cantidad de opiniones relacionadas con cada objetivo. A continuación, se generaron nubes de palabras para cada ODS utilizando `WordCloud`, lo que permitió visualizar las palabras más frecuentes en los textos asociados a cada objetivo. Esta visualización ayudó a identificar los términos predominantes de manera intuitiva. También se aplicó `CountVectorizer` para extraer las palabras más frecuentes en los textos de cada ODS, proporcionando una lista detallada de los términos más comunes junto con su frecuencia de aparición.

Para el ODS 3 (Salud), los términos predominantes como "salud", "atención" y "servicios" sugieren que las opiniones se centran en la calidad y el acceso a los servicios de salud. Esto indica que podría haber áreas de mejora en la prestación de servicios de salud, especialmente en la atención primaria y la salud mental. Las estrategias podrían enfocarse en mejorar la accesibilidad y la calidad de la atención sanitaria para abordar estas preocupaciones.

En el ODS 4 (Educación), las palabras clave como "educación", "estudiantes" y "escuelas" reflejan un interés significativo en la calidad educativa y la infraestructura escolar. La prominencia de términos como "evaluación" y "aprendizaje" sugiere que hay un énfasis en la efectividad del sistema educativo y la importancia de la evaluación. Esto sugiere que las estrategias deben centrarse en mejorar la infraestructura educativa, el apoyo a los estudiantes y la capacitación de los docentes, así como en la implementación de métodos de evaluación efectivos para asegurar una educación de calidad.

Para el ODS 5 (Igualdad de Género), los términos "mujeres", "género" e "igualdad" indican una fuerte preocupación por la igualdad de género y los derechos de las mujeres. La alta frecuencia de términos relacionados con "violencia" y "políticas" revela una posible

necesidad de abordar la violencia de género y de fortalecer las políticas de igualdad. Las estrategias deben enfocarse en la creación e implementación de políticas que promuevan la igualdad de género, el apoyo a las víctimas de violencia y el fomento de un entorno más equitativo para todos.

Nube de palabras por ODS 3,4,5 Resultados

5. Mapa de actores relacionado con el producto de datos creado.

Rol	Actor	Beneficio	Riesgo
Organizador y gestor financiero del proyecto	Fondo de Población de las Naciones Unidas	Con la construcción de este modelo, se evidencia una mejora en la identificación de los ODS. En general, este modelo permite aumentar la eficiencia del proceso de recolección y agrupación de opiniones ciudadanas en las categorías de salud y bienestar, educación de calidad y igualdad de género.	El principal riesgo al cual se enfrenta el UNFPA, se basa en que esto generaría una dependencia en la precisión, eficacia y comportamiento del modelo. Por esto mismo, es necesario implementar correctamente la clasificación del algoritmo, para lograr un f1-score beneficioso con el modelo.
Organizaciones enfocadas en servicios sociales	OSC	Estas organizaciones se pueden beneficiar el proceso de recolección de datos de necesidades y peticiones que tiene la población. Por medio de estas encuestas, las organizaciones pueden tener contacto directo con la población, que tiene información valiosa ante problemáticas.	Si la clasificación del modelo no es la más efectiva, se aumentaría la desconfianza de la población ante las labores de estas organizaciones enfocadas al servicio social.
Generadores de información y data	Personas	Esta herramienta permite conectar directamente con las opiniones de los ciudadanos, para gestionar soluciones ante ciertos problemas diarios.	Si le modelo no es preciso, esto puede generar desconfianza de los ciudadanos. Considerando que sus opiniones no se consideran.
Gestores de implementación de modelo	Entidades gubernamentales	Este tipo de herramientas les pueden brindar información relevante para la toma de decisiones que apoye al desarrollo y progreso de comunidades, municipios, pueblos, etc. Esta herramienta puede ayudar a mejorar la planificación de implementación de nuevas políticas públicas, según comentarios e información relevante según las necesidades de la población.	El riesgo más notable, es que este modelo puede llegar a no se exacto o preciso con algún comentario de los ciudadanos, esto puede llegar a tener repercusiones negativas, especialmente si la ciudadanía empieza a presentar cierto nivel de desconfianza ante el uso de este modelo, considerando que su opinión no haya sido tenida en cuenta.

6. Trabajo en equipo

Para organizar las tareas de la implementación del proyecto, decidimos establecer ciertos roles que nos permitieron completar los requisitos de esta primera etapa del proyecto. En primer lugar, Natalia Ricaurte, fue la encargada de liderar el proyecto. En primer lugar,

realizo una gestión de los tiempos que cada uno debía dedicarle al proyecto con el objetivo de acabarla para antes del 7 de septiembre a las 8 pm. Igualmente, ella coordinaba los momentos en que nos reuniríamos con todos los miembros del grupo. En estas reuniones, ella se encargó de asignarle tareas individuales a cada miembro del equipo, y revisando constantemente el progreso que tenía. La persona encargada de ser el líder del negocio fue Valentina Lara, ella se encargó de que todas las métricas del resultado final del proyecto lograran cumplir con las necesidades del contexto del problema. Ella tuvo que alinear nuestros esfuerzos hacia lo que quería lograr para asegurar el mejor resultado final del proyecto. De igual forma, se encargó de crear gráficos relevantes al contexto de la problemática, para identificar si nuestros modelos comunicaban correctamente el objetivo del proyecto. La persona encarga de ser líder de los datos fue Natalia Ricaurte. Su principal labor fue realizar la preparación de los datos para poder diseñar los 6 modelos que diseñamos en el grupo. Finalmente, Manuela Lizcano, líder de analítica, se encargó de seleccionar el modelo que lograra una mejor clasificación con mayor f1-score. Esto para garantizar el funcionamiento del proyecto planteado por el fondo de poblaciones de las Naciones Unidas y entidades públicas. Ella también se aseguró que se analizarían los datos en los 6 modelos, para identificar los beneficios y falencias que cada modelo. Esto fue mostrado en el documento y en la presentación. Los tiempos dedicados para elaborar la etapa 1 del proyecto se dividieron en dos partes. El tiempo que cada miembro del equipo le dedico en su tiempo libre y de igual las reuniones que realizamos todos los días de lunes a viernes, dos horas diarias. Los modelos que fueron implementados para verificar las métricas de clasificación fueron: regresión logística, arboles de decisión, KNN, Random Forest, SVM y MultinomialNB. Considerando que implementamos 6 modelos diferentes, cada uno de los miembros se encargó de dos.

Natalia Ricaurte: Random Forest y SVM
Valentina Lara: MultinomialNB y KNN
Manuela Lizcano: Arboles de decisión y Regresión logística

Los principales retos que enfrentamos durante la realización de proyecto fueron que al final, no tuvimos mucho tiempo de clase para aclarar las dudas que surgieron durante la realización del proyecto. La forma en que las solucionamos fue hablando con otros grupos que se habían encontrado con los mismos problemas y estar tener un contacto continuo con slack. Para la siguiente entrega del proyecto avanzaremos antes de la fecha de entrega, para aprovechar mejor el espacio de en clase para aclarar dudas.

Reflexión de repartición de puntos: Natalia Ricaurte: 33.33%, Valentina Lara: 33.33%, Manuela Lizcano: 33.33%

7. Referencias

- Built In. (s.f.). How to build a logistic regression model for classification. https://builtin.com/articles/logistic-classifier
- DataScientest. (s.f.). Random Forest: Definición y Funcionamiento. https://datascientest.com/es/random-forest-bosque-aleatorio-definicion-y-funcionamiento
- Pérez, 2024. El algoritmo SVM y sus aplicaciones empresariales. https://www.obsbusiness.school/blog/el-algoritmo-svm-y-sus-aplicaciones-empresariales
- Scikit-learn. (s.f.). 1.10. Decision trees. https://scikit-learn.org/stable/modules/tree.html
- Scikit-learn. (s.f.). MultinomialNB. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#multinomialnb
- Scikit-learn. (s.f.). Nearest Neighbors. https://scikit-learn.org/stable/modules/neighbors.html
- Scikit-learn. (s.f.). Ensemble methods - Random Forests. https://scikit-learn.org/stable/modules/ensemble.html#forest
- Scikit-learn. (s.f.). Support Vector Machines. https://scikit-learn.org/stable/modules/svm.html