

Inteligencia de Negocios

ISIS 3301

Proyecto 1 – Construcción de Modelos de Analítica de Datos

Wilmer Arévalo

202214720

Nicolás Saavedra

202112963

Juan David Castillo

202210669

Universidad de los Andes

2024

Sección 1: Entendimiento del negocio y enfoque analítico

| | |
|--|--|
| Oportunidad/problema de negocio | <p>El Fondo de Poblaciones de las Naciones Unidas (UNFPA) enfrenta el desafío de analizar grandes volúmenes de opiniones textuales de ciudadanos, las cuales son críticas para evaluar y desarrollar soluciones que se alineen con los Objetivos de Desarrollo Sostenible (ODS) 3, 4 y 5. Actualmente, este proceso consume recursos significativos, incluyendo la participación de expertos, lo que limita la capacidad de respuesta y la eficiencia del análisis.</p> |
| Objetivos y criterios de éxito desde el punto de vista del negocio | <ul style="list-style-type: none">- Objetivo: Desarrollar un modelo analítico automatizado que clasifique y relacione opiniones de ciudadanos con los ODS 3 (Salud y bienestar), 4 (Educación de calidad) y 5 (Igualdad de género).- Criterio de éxito: El modelo debe ser capaz de clasificar las opiniones con una precisión superior al 85%, lo que permitirá al UNFPA y a las entidades públicas tomar decisiones informadas con mayor rapidez y menor costo. |
| Organización y rol dentro de ella que se beneficia con la oportunidad definida | <ul style="list-style-type: none">- Organización: Fondo de Poblaciones de las Naciones Unidas (UNFPA).- Rol beneficiado: Equipos de análisis de datos y formuladores de políticas públicas que utilizan los resultados del análisis de opiniones para diseñar intervenciones alineadas con los ODS mencionados. Adicionalmente, los tomadores de decisiones en el UNFPA podrán priorizar recursos hacia |

| | |
|---|---|
| | áreas críticas identificadas a partir del análisis de datos. |
| Impacto que puede tener en Colombia este proyecto | Este proyecto tiene el potencial de mejorar significativamente la capacidad del gobierno y las organizaciones no gubernamentales en Colombia para responder a las necesidades de la población en áreas clave como salud, educación e igualdad de género. Al automatizar el análisis de grandes volúmenes de datos textuales, se espera un impacto positivo en la eficiencia de la implementación de políticas públicas, contribuyendo al avance de los ODS en el país. |
| Enfoque analítico | <ul style="list-style-type: none"> - Categoría de análisis: Predictivo - Tipo de aprendizaje: Supervisado - Tarea de aprendizaje: Clasificación de textos - Técnicas y algoritmos propuestos: Para alcanzar los objetivos planteados, se utilizarán modelos de aprendizaje automático, tales como <i>Logistic Regression</i>, <i>K-Nearest Neighbors (KNN)</i> y <i>Random Forest</i>, todos bien conocidos por su efectividad en la clasificación de texto. Además, se explorará el uso de técnicas avanzadas para capturar patrones complejos en las opiniones textuales. |

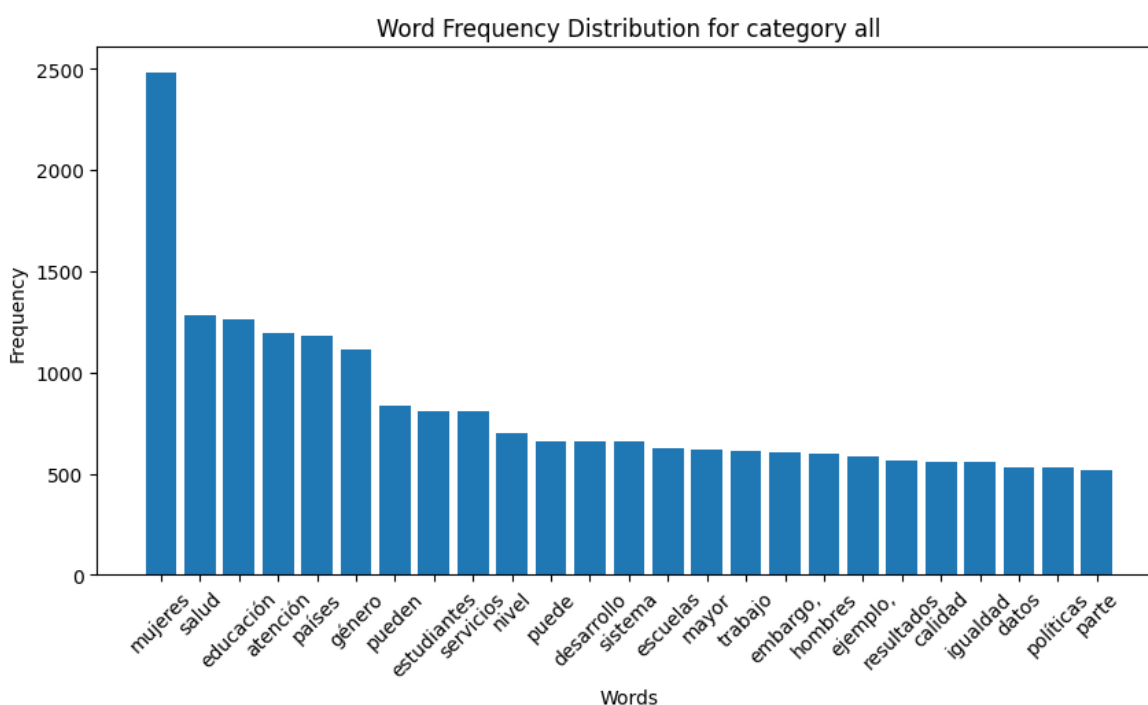
Tabla 1. Entendimiento y enfoque analítico

Sección 2: Entendimiento y preparación de datos

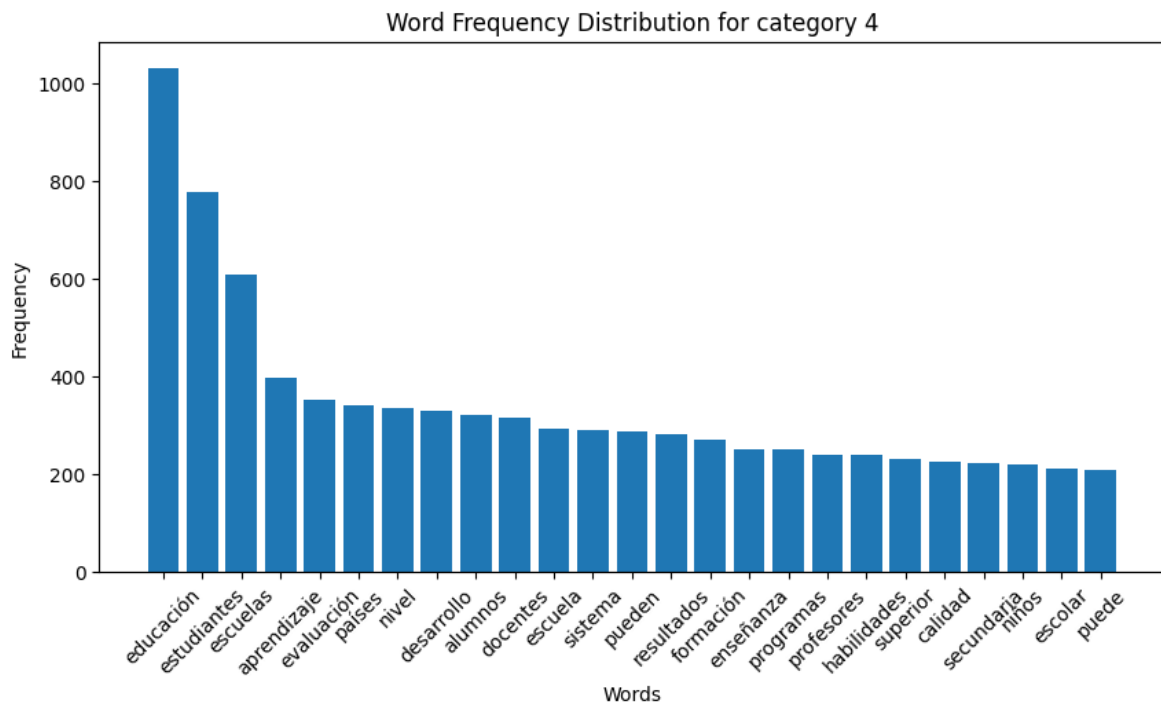
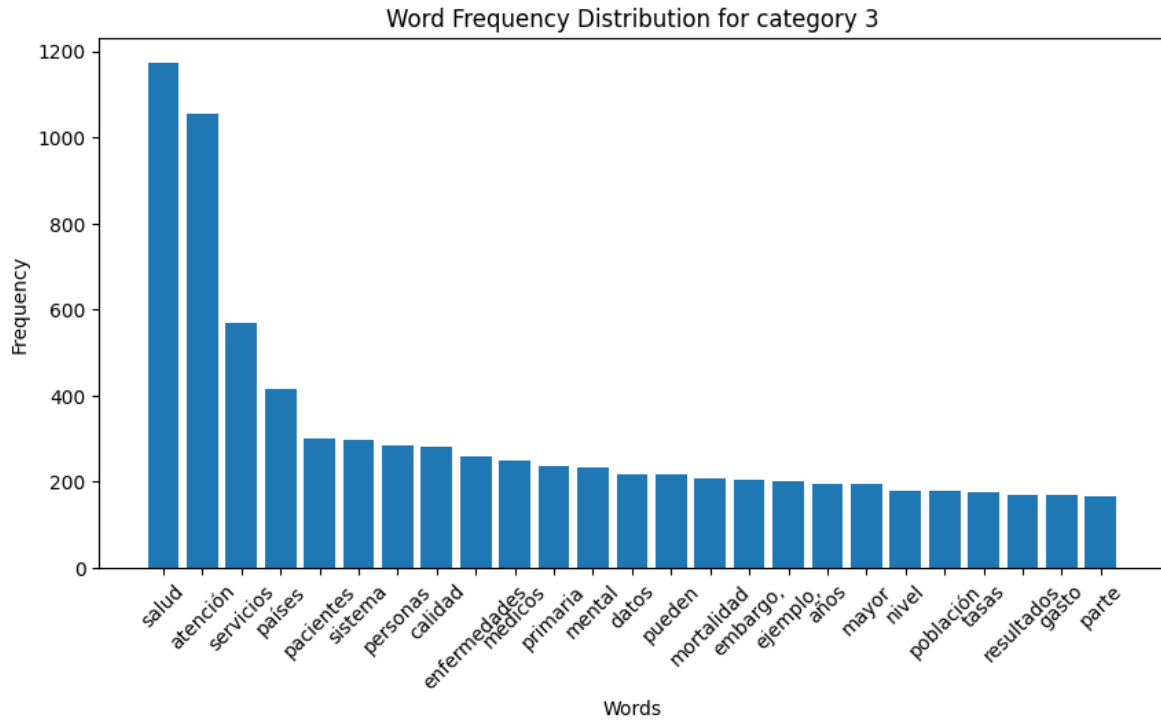
Para el proceso de entendimiento de datos, debemos tener en cuenta que tan viable va a ser un modelo para hacer predicciones basado en una vectorización de TFIDF, la cual se asemeja

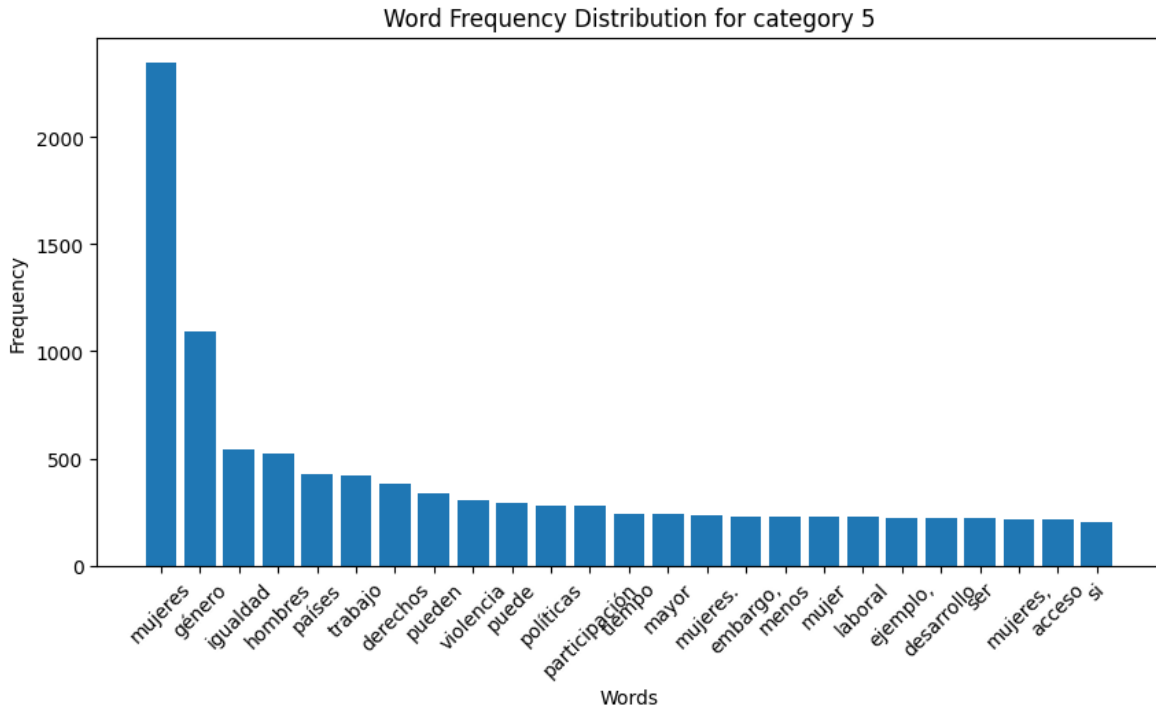
a *BoW*, en el hecho de vectorizar basado en frecuencia de palabras, junto con combinaciones como bigramas y trigramas.

De acuerdo con eso, se escogió el corpus de texto en general del *dataset* y se analizaron las primeras 50 palabras que aparecían tanto en el corpus general como en cada una de las categorías individuales:



Podemos observar que las palabras con mayor frecuencia en el corpus general son palabras relacionadas con el tópico de la categoría, puesto que se puede observar la aparición de palabras como mujeres (para categoría 5), educación (para categoría 4) y salud (para categoría 3). Esto nos da un indicador positivo que un algoritmo como TFIDF puede identificar la categoría de los textos de forma precisa.





Podemos observar haciendo un análisis más a fondo que la frecuencia de palabras es directamente proporcional a la frecuencia de la categorización en el *dataset*. Esto nos permite intuir que TFIDF será un algoritmo de vectorización adecuado para nuestro caso de estudio. Sin embargo, también podemos observar la presencia de puntuación y simbología como tildes en el texto, lo cual tendremos que filtrar.

Ya con esto en mente, pasamos al proceso de tokenización, el cual dividimos en varios filtros diferentes:

- Lowercase

Coge el *dataset* y le aplica un *toLowerCase()* básico para dejar todas las palabras en minúscula.

- Number to text

Toma el *dataset* y convierte todos los caracteres individuales que son numéricos en su versión textual en español.

- Remove punctuation

Toma el *dataset* y retira los símbolos de puntuación y demás componentes textuales que no son considerados letras o números.

- Fix partial encoding

Toma el *dataset* y palabra por palabra, revisa la presencia de un byte que solo aparece en el encoding ISO-8559-1, ya que las tildes del texto están parcialmente en UTF-8 y parcialmente en ISO-8559, por lo tanto, toca hacer esta conversión palabra por palabra para razonar un formato común.

- Stem words

Toma el *dataset* y aplica *stemming* a las palabras para normalizarlas y reducir el tamaño del resultado del paso de vectorización más adelante.

Todos estos pasos son aplicados al *dataset* después de ejecutar un *word_tokenize*, con el fin de separar palabras, para convertir las entradas de texto de cada fila en una lista de tokens que puede ser vectorizada más adelante usando TFIDF y trigramas. Todos estos pasos son importantes, ya que permiten que varias palabras con similar significado intrínseco puedan ser interpretadas de la misma forma en el modelo y pueda realizar mejores asociaciones con menos datos.

El paso de vectorización, como fue mencionado previamente, se realizó usando TFIDF y trigramas. Esto se hizo pensando en la importancia de contexto que una palabra puede tener para la clasificación, puesto que la combinación de palabras importantes como “mujeres” y “genero” le puede dar un índice más fuerte al modelo que un trozo de texto pertenece a una categoría (en este caso, la 5).

Sección 3: Modelado y evaluación

Para nuestro proyecto implementamos 3 principales algoritmos para la generación de los modelos. *Random Forest*, *Logistic Regression* y por último *KNN*.

En primer lugar, los random forest usan la idea de árboles de decisión para construir modelos de clasificación de regresión ensamblando varios árboles de decisión. Esto permite dar una serie de condiciones desde un punto de entrada, que nos lleva a un punto de salida final dependiendo en cada decisión tomada desde cada vértice del camino. Aunque el problema es que pueden sesgar la función objetivo, entre más decisiones, más se complejiza. Además, puede tener una alta varianza, esto quiere decir que nos puede llegar a representar datos alejados entre sí. En caso de tener una variable categórica se busca la mayoría de las características o si es una variable de regresión, es decir un número, se muestra el promedio de los diferentes modelos generados en los bosques aleatorios. Uno de los parámetros aplicados en nuestro caso es el número de estimadores o el número de árboles (número entre 100 y 500) y el *min_leaf = mínima cantidad de muestras en una división para que se permita (número entre 1 y 5)*. Además de esto, el random forest quisimos implementar un *RandomSearch* para intentar refinar sus parámetros de una mejor manera.

Por otro lado, *Logistic regression* es una técnica probabilística de clasificación discriminante que permite predecir la probabilidad de obtener una variable categórica dada una combinación lineal en una entrada. Se tiene una función de probabilidad donde existen los parámetros que constituyen el modelo, características definidas como variables independientes de entrada y salidas categóricas. Entra un valor para cambiarlo a una salida entre 0 y 1. La idea en este modelo es optimizar la función de verosimilitud. Para nuestro caso, no le pusimos restricciones en los parámetros de la función de Python, por lo que le permitimos al modelo trabajar de manera natural. Esto permitió que consiguiéramos las mejores métricas entre los 3 algoritmos, es por esto por lo que lo dejamos como nuestro modelo principal.

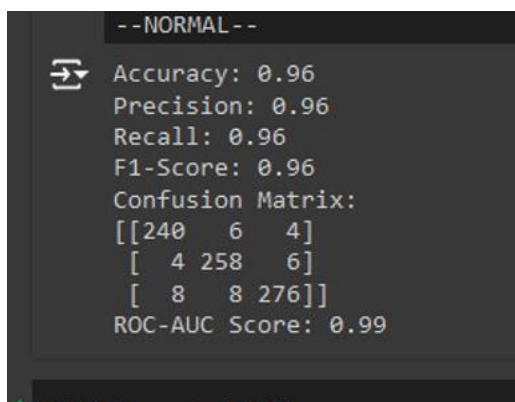
Por último, KNN es un modelo que comúnmente se utiliza para clasificar instancias. Por medio de graficas podemos determinar y predecir algún comportamiento final de uno de nuestros sujetos en los datos, esto lo convierte en un modelo muy intuitivo. Sin embargo, una de sus desventajas se caracteriza cuando tenemos muchas columnas, debido a que el rendimiento del modelo disminuye. Este trabaja con el comportamiento promedio dependiendo del rango de características, esta información nos ayuda a predecir o determinar si un escenario específico dentro de los datos puede comportarse igual que sus vecinos. Para

nuestro caso, determinamos solo tener en cuenta 10 vecinos por cada escenario para recolectar la información.

Sección 4: Resultados

4.a

KNN: En primer lugar, KNN es un modelo que en un principio se podría considerar bueno para la clasificación de los datos, debido a que no cuenta con muchas columnas así que permite dar mayor facilidad y precisión para el modelo. Aunque, el algoritmo representa el modelo más demorado en cuanto a eficiencia, pero no en cuanto a entrenamiento.



```
--NORMAL--  
Accuracy: 0.96  
Precision: 0.96  
Recall: 0.96  
F1-Score: 0.96  
Confusion Matrix:  
[[240  6  4]  
 [  4 258  6]  
 [  8  8 276]]  
ROC-AUC Score: 0.99
```

Random Forest: Para el caso de random forest obtuvimos resultados bastante buenos, que nos acercaron a un buen modelo en primer lugar. Sin embargo, al ejecutar este algoritmo, nos entrega un modelo que se demora mucho tiempo en terminar de ejecutar, por lo cual no lo tomamos como el modelo preferido para la clasificación de nuestros datos. Al contrario del KNN, el random forest representa el algoritmo más demorado en cuanto a entrenamiento, más no en eficiencia.

```
Accuracy: 0.97
Precision: 0.97
Recall: 0.97
F1-Score: 0.97
Confusion Matrix:
[[238  5  7]
 [  1 264  3]
 [  3  8 281]]
ROC-AUC Score: 1.00
```

Logistic regression: Por último, tenemos nuestro algoritmo de preferencia, debido a que además de darnos los resultados en un tiempo considerable, también nos muestra el valor mejor representado en las métricas dadas. Es el algoritmo con mejor velocidad, tanto en eficiencia y entrenamiento de los tres modelos.

```
Accuracy: 0.98
Precision: 0.98
Recall: 0.98
F1-Score: 0.98
Confusion Matrix:
[[246  2  2]
 [  2 262  4]
 [  0  8 284]]
ROC-AUC Score: 1.00
```

4.c Revisar *dataset* dentro del repositorio con título *TestODScat_345_filled.xlsx*

4.d <https://drive.google.com/file/d/1mWW7L6-5SogPv6kzhI0RCOFhuvxqBlFo/view?usp=sharing>

Sección 5: Mapa de actores relacionado con el producto de datos creado

Organización: Fondo de Poblaciones de las Naciones Unidas (UNFPA)

Contexto: El UNFPA trabaja en colaboración con entidades públicas y utilizando herramientas de participación ciudadana para identificar problemas y evaluar soluciones en el marco de los ODS. El modelo desarrollado, que utiliza algoritmos de *Logistic Regression*,

K-Nearest Neighbors (KNN) y *Random Forest* para la clasificación de opiniones, puede ser una herramienta clave para mejorar la eficiencia y precisión de estos análisis.

| Rol dentro de la organización | Tipo de actor | Beneficio | Riesgo |
|---|---------------------|---|--|
| Tomadores de decisiones en políticas públicas | Usuario-cliente | Apoyo en la formulación de políticas basadas en datos precisos y actualizados sobre las preocupaciones ciudadanas alineadas con los ODS. | Si el modelo no se ajusta bien a nuevos datos, podría conducir a decisiones basadas en información incorrecta. |
| Analistas de datos del UNFPA | Usuario interno | Herramienta que automatiza el análisis de textos, liberando recursos y tiempo que pueden ser redirigidos a tareas más estratégicas. | Dependencia excesiva del modelo, lo que podría limitar la intervención manual en casos complejos o excepcionales. |
| Entidades gubernamentales asociadas | Cliente-colaborador | Recepción de informes detallados y basados en datos sobre las áreas de intervención prioritarias según las opiniones ciudadanas, mejorando la eficiencia gubernamental. | Posible resistencia al cambio si los resultados del modelo contradicen políticas o enfoques previos. |
| Ciudadanos participantes en la consulta | Beneficiado | Mayor posibilidad de que sus opiniones sean tenidas en cuenta en la formulación de políticas, lo que podría llevar a un | Riesgo de desconfianza si se percibe que el modelo no refleja adecuadamente las opiniones emitidas o es parcial en sus resultados. |

| | | | |
|--|--|--|--|
| | | aumento en la confianza en las instituciones. | |
|--|--|--|--|

Tabla 2. Mapa de actores relacionado con el modelo analítico planteado.

Beneficio general del proyecto

El modelo analítico propuesto no solo aumenta la eficiencia en el proceso de análisis de opiniones ciudadanas, sino que también mejora la capacidad del UNFPA y de las entidades gubernamentales para responder de manera ágil y precisa a las necesidades de la población, alineándose con los objetivos de los ODS 3, 4 y 5.

Riesgo general del proyecto

El principal riesgo radica en la posibilidad de que el modelo no se adapte bien a datos futuros o a variaciones en el tipo de lenguaje utilizado por los ciudadanos, lo que podría reducir la precisión y la utilidad de los resultados. Es esencial continuar reentrenando el modelo con datos nuevos y variados para mitigar este riesgo.

Sección 6: Trabajo en equipo

Descripción de roles y tareas realizadas por cada integrante del grupo:

- Nicolás Saavedra

Rol: Líder de analítica y desarrollo.

Tareas realizadas: Nicolás asumió la responsabilidad principal en la implementación y programación de los modelos analíticos. Se encargó de la preparación de datos, la elección de los algoritmos y la ejecución del modelado utilizando Logistic Regression, K-Nearest Neighbors (KNN) y Random Forest. Además, se aseguró de que los modelos fueran evaluados de manera rigurosa para seleccionar el más efectivo, que resultó ser Logistic Regression.

Horas dedicadas: Aproximadamente 30 horas en total, distribuidas en la implementación de modelos, pruebas y ajustes de hiperparámetros.

Retos enfrentados: Uno de los mayores desafíos fue garantizar que los modelos manejaran adecuadamente las variaciones en los datos textuales y evitar el sobreajuste. Nicolás resolvió este reto mediante una cuidadosa validación cruzada y la selección de los mejores hiperparámetros.

Soluciones planteadas: Para mejorar la robustez del modelo, se implementaron técnicas de regularización y se realizaron múltiples iteraciones de prueba para optimizar el rendimiento.

- Wilmer Arévalo

Rol: Líder de documentación y presentación.

Tareas realizadas: Wilmer se enfocó en la documentación del proyecto, asegurando que cada fase del proceso estuviera claramente explicada y alineada con los objetivos del negocio. Además, lideró la preparación de la presentación final, diseñando la estructura y el contenido del video explicativo.

Horas dedicadas: Aproximadamente 20 horas en total, distribuidas entre la redacción de la documentación y la preparación de la presentación.

Retos enfrentados: El principal reto fue condensar la complejidad técnica del proyecto en un formato comprensible y atractivo para la presentación. Wilmer trabajó estrechamente con Nicolás y Juan para asegurarse de que la documentación reflejara con precisión los aspectos técnicos y estratégicos del proyecto.

Soluciones planteadas: Para abordar este reto, Wilmer utilizó herramientas visuales y ejemplos concretos para facilitar la comprensión de los conceptos más complejos.

- Juan Castillo

Rol: Apoyo en documentación técnica y presentación a nivel de código.

Tareas realizadas: Juan complementó el trabajo de Wilmer al documentar aspectos técnicos específicos relacionados con el código y las implementaciones realizadas por

Nicolás. Además, revisó y validó la coherencia del código y su alineación con la documentación y la presentación final.

Horas dedicadas: Aproximadamente 18 horas en total, distribuidas en la revisión de código, documentación técnica y apoyo en la presentación.

Retos enfrentados: Juan encontró que uno de los mayores retos fue garantizar que el código estuviera bien comentado y organizado para facilitar su comprensión y presentación.

Soluciones planteadas: Se implementaron comentarios detallados en el código y se utilizaron ejemplos prácticos para mostrar cómo las diferentes partes del código contribuían al resultado final.

Reflexión sobre la distribución de puntos y puntos a mejorar:

En función del trabajo realizado, se ha decidido repartir los 100 puntos de la siguiente manera: 40 puntos para Nicolás Saavedra, 30 puntos para Wilmer Arévalo y 30 puntos para Juan Castillo. Aunque el equipo funcionó de manera eficiente, se identificó que en futuras entregas sería beneficioso aumentar la cantidad de reuniones de seguimiento para asegurar una mejor distribución del trabajo y evitar la concentración de tareas en un solo integrante.