Andres Molina – 201923434
Josue Rivera –201914138
Jaime Alfonso- 202116525

Proyecto 1 Entrega 1 Inteligencia de Negocios

1. (10%) Entendimiento del negocio y enfoque analítico

Oportunidad/problema Negocio	La oportunidad de negocio consiste en realizar un análisis de sentimientos de comentarios de películas en español con el objetivo de clasificarlos como positivos o negativos. Este análisis puede ayudar a las empresas de cine y a las plataformas de streaming a entender mejor la percepción de los usuarios sobre sus producciones y, de esta forma, tomar decisiones de negocio más informadas.
Enfoque analítico (Descripción del requerimiento desde el punto de vista de aprendizaje automático)	El enfoque analítico propuesto para este proyecto se basa en la aplicación de técnicas avanzadas de procesamiento de lenguaje natural y aprendizaje automático para llevar a cabo la tarea de clasificación de los comentarios de las películas. En particular, se utilizarán modelos de clasificación basados en los algoritmos Random Forest, KNN y Árbol de Decisión Extendido. **Random Forest** es un algoritmo de aprendizaje automático que se basa en la construcción de múltiples árboles de decisión independientes para realizar la clasificación. Cada árbol de decisión es entrenado con una muestra aleatoria de los datos de entrada y se utiliza para tomar una decisión de clasificación. La salida final del modelo es el resultado de la combinación de las decisiones de clasificación de cada árbol individual. **KNN** (k-Nearest Neighbors) es un algoritmo de clasificación que se basa en la idea de que los puntos de datos similares deben ser clasificados en la misma categoría. Para clasificar un nuevo punto de datos, KNN busca los k puntos más cercanos en el conjunto de datos de entrenamiento y asigna la etiqueta de clase más frecuente entre ellos al nuevo punto de datos. **Árbol de Decisión Extendido** es una variante del algoritmo de árbol de decisión estándar que se utiliza para clasificar datos en función de múltiples criterios. En este caso, cada nodo del árbol representa una pregunta que divide el conjunto de datos en subconjuntos más pequeños. La extensión del árbol permite la construcción de árboles más

	profundos, lo que permite una mayor precisión en la clasificación.
Organización y rol dentro de ella que se beneficia con la oportunidad definida	Las empresas de cine y las plataformas de streaming son las principales beneficiarias de este análisis, ya que les permite entender mejor la percepción de los usuarios sobre sus producciones y, de esta manera, tomar decisiones de negocio más informadas. Además, otras organizaciones que puedan requerir análisis de sentimientos, como las empresas de publicidad, también pueden beneficiarse de esta técnica.
Técnicas y algoritmos a utilizar	El enfoque analítico propuesto para este proyecto utilizará técnicas avanzadas de procesamiento de lenguaje natural y aprendizaje automático, que incluyen la tokenización, el filtrado de palabras irrelevantes y la vectorización de los datos de entrada. Además, se utilizarán los algoritmos Random Forest, KNN y Árbol de Decisión Extendido para realizar la tarea de clasificación de los comentarios de las películas en positivos o negativos.

2. (25%) Entendimiento y preparación de los datos

En este proyecto de Análisis de Textos se aplicaron diferentes técnicas para la preparación y transformación de los datos, con el objetivo de obtener un conjunto de datos más limpio y homogéneo que permitiera una aplicación efectiva del proceso de descubrimiento de conocimiento a partir de textos.

En el entendimiento y preparación de los datos se llevó a cabo en el notebook Proyecto1Etapa1EntendimientoyLimpieza.ipynb. A continuación, se explica el proceso.

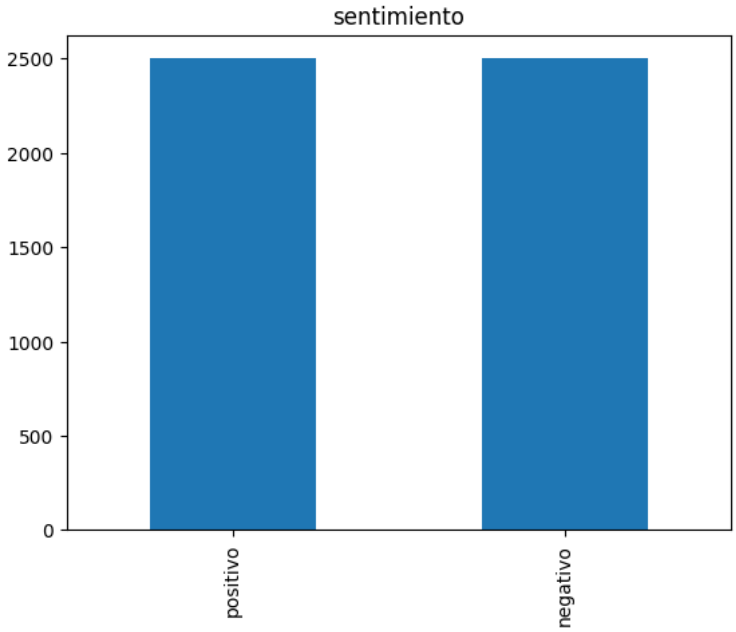
En primer lugar se descargaron los datos y se obtuvo la información inicial de los mismos. La siguiente imagen muestra la cantidad de datos y un ejemplo de ellos.

```
Número de filas:  5000
Número de columnas:  3
```

Out[ ]:

	Unnamed: 0	review_es	sentimiento
3627	3627	Antes de empezar, permÃtanme decir que mi exp...	negativo
3737	3737	PensarÃas que el primer aterrizaje en la luna...	negativo
4386	4386	Esta pelÃcula es, con mucho, la peor pelÃcul...	negativo
2953	2953	La primera vez que vi esto reproducÃ© con horr...	negativo
645	645	Esta pelÃcula se realizÃ³ justo en el Ã¡rea d...	positivo

Out[ ]: <Axes: title={'center': 'sentimiento'}>



Podemos apreciar que el numero de filas es de 5000 y el de columnas es de 3. Hay igual numero de sentimiento positivo que negativo.

- Completitud: Se observa la cantidad de valores nulos en el conjunto de datos:

```
Número de filas con valores nulos:  0
Número de columnas con valores nulos:  0

Porcentaje de completitud de las columnas: 100.00%
```

No fue necesario hacer ningún tipo de cambio con respecto a los valores nulos ya que no existen en el documento MovieReviews.csv

- Unicidad: Se verifica la cantidad de valores duplicados en el conjunto de datos:

```
Número de filas duplicadas:  0
Número de filas con indice duplicado:  0
```

Tampoco fue necesario hacer ningún tipo de cambio en el conjunto de datos ya que no hay datos duplicados.

- Consistencia y validez

La consistencia se refiere a la coherencia de los datos entre diferentes fuentes y observaciones, mientras que la validez evalúa si los datos son apropiados para el contexto en el que se utilizan. En general, se puede afirmar que las medidas de consistencia y validez se cumplen en este caso.

En la limpieza de los datos se eliminaron caracteres distintos al alfabeto en la columna 'review_es' y se convirtió la columna a minúsculas para una mayor homogenización. También se aplicó la librería contractions para la corrección de contracciones lingüísticas, y se eliminaron palabras sin vocales y las stopwords en español con el fin de mejorar la calidad de los tokens y su interpretación. Además, se aplicó el proceso de lematización y la eliminación de prefijos y sufijos (stemming) para la normalización de los datos.

En cuanto al entendimiento de los datos, se verificó que no existieran valores nulos ni filas duplicadas, y se eliminó la columna 'Unnamed: 0', la cual no aportaba información relevante al modelo.

Además, se realizaron análisis de calidad de los datos, donde se verificó el porcentaje de completitud de las columnas, el número de filas y columnas de los datos y el número de filas con valores nulos.
Para la transformación de los datos, se utilizó el método de tokenización para separar el texto en palabras individuales, y se aplicó la función de preprocesamiento que incluyó la eliminación de caracteres especiales, puntuación y números, y la conversión a minúsculas. También se eliminaron palabras sin vocales y las stopwords en español para una mejor interpretación de los tokens.
Finalmente, se aplicó el proceso de lematización y la eliminación de prefijos y sufijos (stemming) para normalizar los datos y se eliminaron los tokens vacíos y con longitud menor a 2. Se eliminó la columna 'review_es' para disminuir el tamaño del dataset.

En resumen, se aplicaron diferentes técnicas y procesos de limpieza y transformación de los datos, con el objetivo de obtener un conjunto de datos más homogéneo y limpio, y así facilitar su posterior análisis para la clasificación de sentimientos de comentarios de películas en español.

3. **(30%) Modelado y evaluación.**

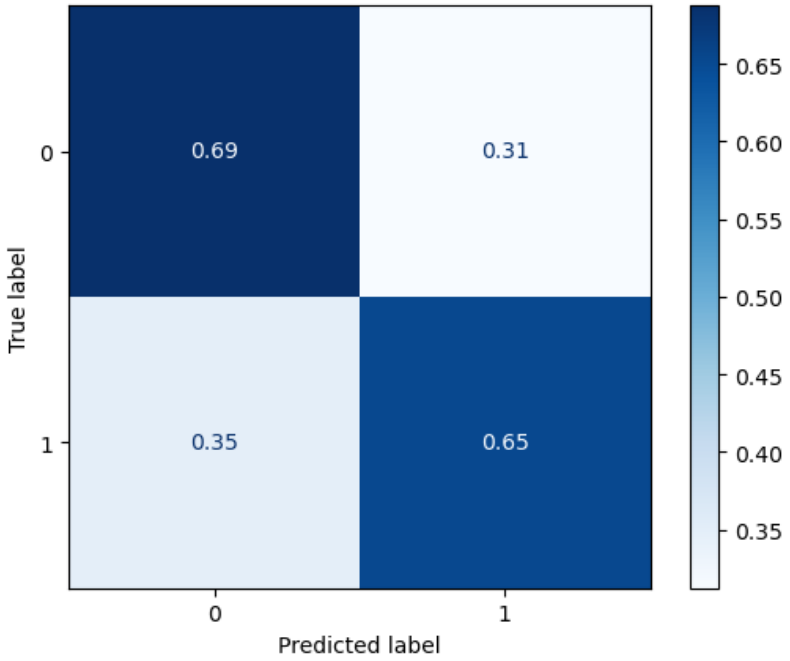
Decision Tree Classifier:

Uno de los modelos usados es un árbol de decisión para clasificar, de manera que se pueda encontrar que valores de tokens tienen el mayor efecto al definir si un comentario se toma como positivo o negativo.

Como en los demás algoritmos realizados, se realizó previamente la vectorización de cada uno de los comentarios mediante el uso de la librería TfidfVectorizer, que se encargó de eliminar las palabras comunes y asignar un peso a cada palabra que se utilizó en el análisis. El modelo se dividió en datos de entrenamiento y prueba (como en los demás algoritmos dado que todos son de clasificación) utilizando la función train_test_split de la librería scikit-learn.

Se buscaron los parámetros que generan los mejores resultados y los que se al final se tomaron permitieron que se diera una precision y exactitud de 67%.

	precision	recall	f1-score	support
0	0.66	0.69	0.67	996
1	0.68	0.65	0.66	1004
accuracy			0.67	2000
macro avg	0.67	0.67	0.67	2000
weighted avg	0.67	0.67	0.67	2000



Aquí se muestra lo explicado anteriormente de manera grafica. Siendo que la precisión con la que se clasifican resultados negativos con un 69% esta solo un poco por encima de la precisión, con la que se clasifican los positivos con un 65%.

En general no esta tan mal, pero comparado con los otros 2 algoritmos/modelos implementados, esta es la peor opción.

KNN:

Tuvimos en cuenta el algoritmo de KNN dado que, para el contexto de analizar los comentarios relacionados con películas y saber si el sentimiento relacionado con estas es positivo o negativo, se puede tener una idea de este sentimiento agrupando los comentarios similares y que palabras en específico definen mejor si dicho comentario lleva consigo un sentimiento positivo o negativo con respecto a la película en cuestión. Además, se podría predecir en un futuro con diferentes datos la probabilidad de que un comentario este asociado a un sentimiento negativo o positivo dependiendo de que palabras estén consignadas en dicho comentario (realizando previamente las transformaciones y el análisis de texto correspondiente).

Como en los demás algoritmos realizados, se realizó previamente la vectorización de cada uno de los comentarios mediante el uso de la librería TfidfVectorizer, que se encargó de eliminar las palabras comunes y asignar un peso a cada palabra que se utilizó en el análisis. El modelo se dividió en datos de entrenamiento y prueba (como en los demás algoritmos dado que todos son de clasificación) utilizando la función train_test_split de la librería scikit-learn.

Después, nos dispusimos a encontrar el número de K óptimo para realizar el modelo, para lo que utilizamos la librería GridSearchCV de sklearn.model_selection, aplicando un total de 5 valores diferentes de K: 2, 5, 10, 15 y 50 elegidos teniendo en cuenta un valor bastante bajo, uno alto y tres intermedios que aumentan de 5 en 5 para procurar un valor acertado del k, pero sin tener que probar muchos K que aumentarían bastante el tiempo de ejecución (que de por sí ya fue de casi 40 minutos) del modelo teniendo en cuenta la cantidad de registros y palabras, además, se utilizó un CV igual a 4.

Gracias a lo anterior, se pudo elegir un modelo (model.best_estimator_) que consiguiera los siguientes resultados:

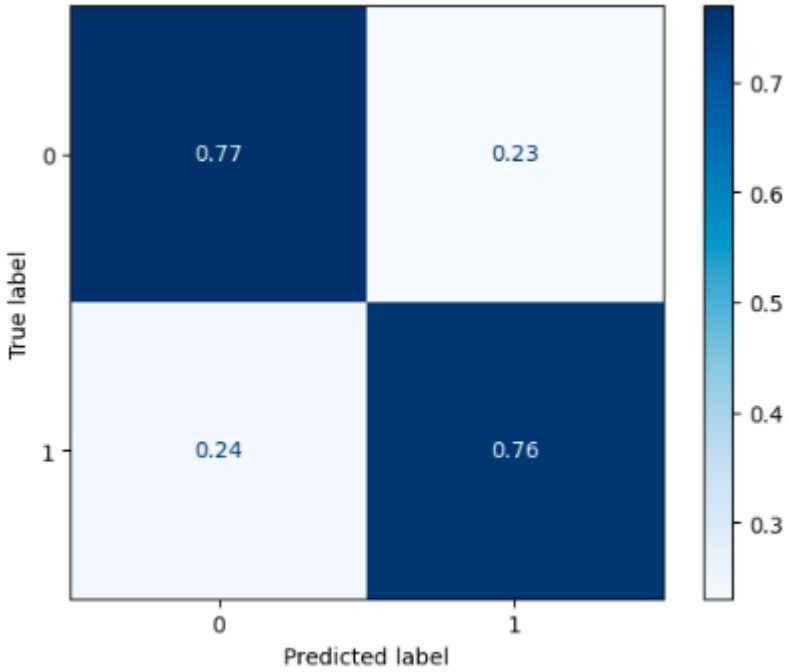
```
Accuracy:  0.763
F1:  0.7630063990575916
Precision:  0.7631364089107208
Recall:  0.763
              precision    recall  f1-score   support

           0       0.76      0.77      0.76       494
           1       0.77      0.76      0.76       506

    accuracy                           0.76      1000
   macro avg       0.76      0.76      0.76      1000
weighted avg       0.76      0.76      0.76      1000
```

Se puede observar que se obtuvo una exactitud y precisión de 0.763 (y en casi todo lo demás), lo cual indica que, aunque no es tan bueno como el Random Forest en este caso, puede ser de gran utilidad para los intereses del negocio, siendo que tiene más del 75% de aciertos y que, además, tiene valores muy similares de correctitud en cuanto a acertar los comentarios relacionados a sentimientos positivos y negativos.

La diferencia entre las precisiones para cada uno de los sentimientos es de solo el 0.01%, lo que quiere decir que no hay un sesgo significativo por parte del modelo a la hora de clasificar los comentarios, esto podría deberse a que el KNN clasifica con base a grupos, por lo que clasifica a cada uno con base a sus similares en cuestión de palabras y no en otros factores.



Aquí se muestra lo explicado anteriormente de manera grafica. Siendo que el nivel de correctitud con el que se clasifican resultados negativos con un 77% esta solo un poco por encima de la precisión, con la que se clasifican los positivos con un 76%.
En conclusión, aunque este modelo no es tan bueno como el Random Forest en términos de precisión y correctitud es una buena opción ya que su porcentaje no es bajo y no tiene casi nada de sesgo a la hora de asignar las clasificaciones.

Random Forest:

El modelo implementado es un Random Forest Classifier que utiliza técnicas de vectorización de texto para analizar el sentimiento en un conjunto de datos con 5000 filas y 2

columnas. La limpieza y vectorización de los datos se realizaron mediante el uso de la librería TfidfVectorizer, que se encargó de eliminar las palabras comunes y asignar un peso a cada palabra que se utilizó en el análisis.
El modelo se dividió en datos de entrenamiento y prueba utilizando la función train_test_split de la librería scikit-learn.

Luego se creó un modelo de clasificación Random Forest con el parámetro "n_estimators" para el número de árboles y "criterion" para la medida de calidad de la división. La optimización de los parámetros se realizó utilizando GridSearchCV con validación cruzada con cuatro particiones.

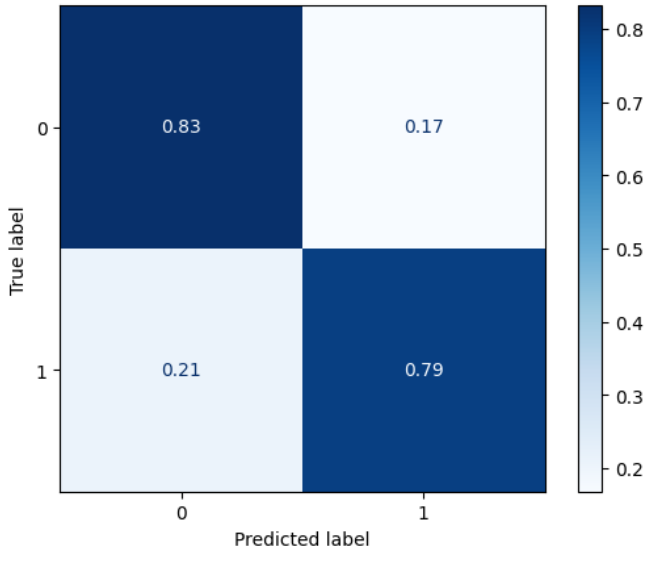
Los resultados obtenidos del modelo se midieron utilizando varias métricas de evaluación. La precisión (accuracy) del modelo fue del 81.2%, lo que significa que el modelo clasificó correctamente el 81.2% de las muestras en el conjunto de datos de prueba. La precisión ponderada (weighted precision) y la recuperación ponderada (weighted recall) fueron del 81.3% y 81.2%, respectivamente. La puntuación F1 (F1 score) fue del 81.2%, lo que indica una buena combinación de precisión y recuperación.

```
Accuracy:  0.812
F1:  0.8119586358635863
Precision:  0.8127040169133193
Recall:  0.812
              precision    recall  f1-score   support

           0       0.80      0.83      0.81       494
           1       0.83      0.79      0.81       506

    accuracy                           0.81      1000
   macro avg       0.81      0.81      0.81      1000
weighted avg       0.81      0.81      0.81      1000
```

Además, se puede observar que el modelo tiene una precisión del 80% para el sentimiento negativo y del 83% para el sentimiento positivo, lo que sugiere que el modelo tiene un sesgo hacia la clasificación de sentimientos positivos. Esto puede deberse a una mayor frecuencia de palabras positivas en el conjunto de datos o una falta de diversidad en los ejemplos negativos. Finalmente, se puede observar la matriz de confusión para evaluar la calidad del modelo en la clasificación binaria. La matriz de confusión muestra que el modelo clasificó correctamente el 79% de las muestras negativas y el 83% de las muestras positivas, lo que confirma el sesgo hacia los sentimientos positivos mencionados anteriormente.



En conclusión, el modelo implementado utilizando el Random Forest Classifier y la vectorización de texto mediante TfidfVectorizer produjo una precisión aceptable para la clasificación de sentimientos en un conjunto de datos. Sin embargo, se observa un sesgo hacia la clasificación de sentimientos positivos, lo que podría mejorarse en futuras iteraciones del modelo.

4. Resultados

En el proceso de análisis de los datos y con el objetivo de determinar el sentimiento de un conjunto de reseñas, se implementaron tres modelos de clasificación: Árbol de decisión, KNN y Random Forest.

Los resultados obtenidos por el modelo de Random Forest presentaron un desempeño destacado en comparación con los otros modelos, al tener un porcentaje de exactitud de 0.84, un puntaje F1 de 0.83, y una precisión y sensibilidad promedio ponderada de 0.84. Es importante mencionar que estos resultados se obtuvieron utilizando la técnica de validación cruzada, lo cual garantiza una mayor confiabilidad de los resultados.

Por otro lado, el modelo de Árbol de decisión obtuvo un porcentaje de exactitud de 0.67, un puntaje F1 de 0.67, y una precisión y sensibilidad promedio ponderada de 0.67. Mientras que el modelo de K-vecinos más cercanos logró un porcentaje de exactitud de 0.76, un puntaje F1 de 0.76, y una precisión y sensibilidad promedio ponderada de 0.76.

Estos resultados son relevantes para la organización, ya que les permiten conocer la opinión de los clientes y así poder tomar decisiones informadas en cuanto a la mejora de sus productos o servicios. Con estos modelos de clasificación, se pueden identificar patrones y tendencias en las reseñas de los clientes, lo cual les permitirá mejorar la calidad de sus productos o servicios y satisfacer las necesidades de sus clientes.

Para aprovechar los resultados obtenidos, la organización podría implementar estrategias para mejorar su relación con los clientes. Por ejemplo, podrían utilizar los patrones identificados para personalizar la experiencia del cliente y ofrecer productos o servicios que se adapten a sus necesidades. También podrían utilizar los resultados para detectar problemas comunes en las reseñas y corregirlos en su proceso de producción o servicio.

En resumen, los modelos de clasificación implementados permiten a la organización comprender mejor las opiniones de sus clientes y mejorar su relación con ellos. Los resultados obtenidos muestran que el modelo de Random Forest es el más efectivo para clasificar las reseñas según su sentimiento, pero los otros modelos también pueden ser útiles para identificar patrones y tendencias. En general, la información proporcionada por estos modelos es valiosa para la organización y puede ser utilizada para tomar decisiones informadas y mejorar la calidad de sus productos o servicios.

5. **Trabajo en equipo**

- Líder de proyecto: Andres Molina

Andrés Molina es el líder del proyecto y está a cargo de la gestión de este. Ha definido las fechas de reuniones y pre-entregables del grupo, y ha verificado las asignaciones de tareas para que la carga sea equitativa. Andres, estuvo pendiente por medio del grupo de WhatsApp para que se empezara a trabajar el proyecto con antelación. Se ha encargado de subir la entrega del grupo y ha tomado decisiones cuando no ha habido consenso, siempre buscando el beneficio del proyecto y del equipo. Andrés ha sido un líder eficiente y ha fomentado un ambiente colaborativo y productivo entre los miembros del equipo.

- Líder de datos: Josue Rivera

El líder de datos en el proyecto se llama Josue Rivera. Su función principal es manejar y organizar los datos que se utilizarán en el proyecto, asegurándose de que estén disponibles para todos los miembros del equipo. Él también es responsable de asignar tareas relacionadas con el manejo y análisis de datos, y de asegurarse de que se completen dentro de los plazos establecidos. Josue trabaja en estrecha colaboración con el líder del proyecto y otros miembros del equipo para garantizar que los datos sean precisos, completos y relevantes para el proyecto. Además, se asegura de que se cumplan los estándares de privacidad y seguridad de los datos en todo momento.

- Líder de negocio: Jaime Alfonso

Jaime es el líder de negocio en el proyecto. Se asegura de que el problema o la oportunidad identificada se resuelva de manera efectiva y que esté alineado con la estrategia de negocio para el cual se está llevando a cabo el proyecto. También es responsable de garantizar que el producto se pueda comunicar adecuadamente y que se cumplan los objetivos del proyecto. Jaime tiene el papel crucial de contactar al grupo de expertos en estadística para evaluar el modelo, tanto a nivel cuantitativo como cualitativo, para garantizar la calidad del producto final. Además, trabaja en estrecha colaboración con el equipo técnico y el líder de proyecto para asegurarse de que el proyecto esté avanzando de acuerdo con los objetivos establecidos.

- Líder de Analítica: Josue Rivera

Josué es el líder de analítica del equipo, su responsabilidad es gestionar las tareas de análisis de datos y asegurarse de que los entregables cumplan con los estándares requeridos. Josué trabaja en conjunto con los otros líderes del equipo para identificar las mejores prácticas de análisis y asegurar que se estén aplicando correctamente en el proyecto. Además, Josué es el encargado de verificar que el equipo esté utilizando las herramientas y metodologías adecuadas para analizar los datos y encontrar la solución más efectiva al problema planteado.

6. **Evaluación del aporte individual**

- Josue Rivera: Propuestas de reunión, establecimiento de comunicación continua, desarrollo de algoritmo del arbol de decisión para generacion de modelo.En cuanto a retos enfrentados el mayor reto fue dar tiempo a el desarrollo del trabajo, pero este fue solucionado estableciendo unas fechas de entrega para ir avanzado continuamente en el proyecto.En total se dieron más de 10 horas al proyecto. Y de acuerdo con estas contribuciones establezco de 32 de los 100 puntos repartidos.
- Jaime Alfonso: Ademas de lo mencionado anteriormente, yo me encargue del algoritmo KNN, lo mas difícil en lo personal fue encontrar el K indicado para ejecutar el algoritmo dado que ya tenía la

vectorización, porque primero intente con un solo valor, pero la precisión y el recall me daban demasiado bajos, por lo que decidí hacer el croos validation para encontrar el k optimo, sin embargo, los tiempos de ejecución eran demasiado largos por lo que no podía probar muchos k con los datos estipulados, así que por ultimo opte por implementar lo ya mencionado anteriormente en la parte de modelado y evaluación, que aunque tardo 40 minutos en ejecutarse, pude incrementar bastante la precisión y la correctitud, además pude exportar el modelo con la librería pickle. En total tarde unas 10 horas contando tiempo de ejecución, y considero que mi participación es de 32 de los 100 puntos repartidos.

- Andrés Molina: Más allá de lo mencionado previamente con respecto a mi rol en el equipo, cabe recordar que yo me encargue de la implementación del modelo Random Forest. Además, me encargue del entendimiento y la limpieza total de los datos. Los retos mas grandes que me enfrente a la hora de hacer estas actividades fueron varias. La primera, fue lograr limpiar los datos de manera correcta ya que, si esta parte se hace de manera errónea, los 3 modelos implementados no serán precisos, ya que los datos están mal. Con respecto a la implementación del modelo, fue emplear un algoritmo que nunca había trabajado antes, por lo que le tuve que dedicar varias horas de aprendizaje. También, nuestra herramienta, Google Colab no tenia algunas extensiones y tocaba instalarlas, situación que me hizo perder mucho tiempo ya que yo no sabía cómo hacer eso. A este proyecto le dedique un tiempo de alrededor de 6 horas sin contar ningún tiempo de ejecución. En cuanto a la repartición de puntos considero que mi contribución es de 36 puntos sobre 100 totales.