MODELO DE LENGUAJE NATURAL PARA EL ANÁLISIS DE CONVERSACIONES DE WHATSAPP ENTRE UN ASESOR DE COBRANZAS Y UN CLIENTE

Jesus David Alvear Corro

Código 8.992.695

Proyecto Aplicado para optar al título de

Magister en Ciencia de Datos

Director(a)

Carlos Ernesto Ramírez Ovalle

Codirector(a)

Abel Álvarez Bustos

FACULTAD DE INGENIERÍA Y CIENCIAS

MAESTRÍA EN CIENCIA DE DATOS

BOGOTÁ, MAYO 19 DE 2025

Reconocimiento o Agradecimientos

Este proyecto representa la culminación de un proceso de formación académica y profesional que no habría sido posible sin el acompañamiento, guía y apoyo de diversas personas e instituciones, a quienes deseo expresar mi más sincero agradecimiento.

En primer lugar, agradezco profundamente a mis directores de trabajo de grado, Carlos Ernesto Ramírez Ovalle y Abel Álvarez Bustos, por su orientación constante, sus valiosos aportes técnicos y metodológicos, y su disposición para acompañar cada etapa de este proyecto con rigor y compromiso.

A la Universidad Javeriana y a la Facultad de Ingeniería y Ciencias, por brindarme las herramientas, el entorno académico y los recursos necesarios para desarrollar una investigación aplicada con impacto real. Su enfoque multidisciplinar y formación rigurosa han sido pilares fundamentales en mi desarrollo como profesional de ciencia de datos.

Extiendo un agradecimiento especial al Banco Finandina, por su respaldo institucional y por facilitar el acceso a los datos y plataformas necesarias para el desarrollo de este trabajo. Su compromiso con la innovación y el uso responsable de la tecnología fueron claves para aplicar este proyecto en un contexto real.

A mis compañeros de la Maestría en Ciencia de Datos, por las conversaciones, aprendizajes compartidos y por ser fuente constante de motivación e inspiración.

A mi familia, especialmente a mis padres y seres queridos, por su apoyo incondicional, su paciencia y por creer en mí incluso en los momentos más exigentes de este proceso.

Finalmente, agradezco a todas las personas que, directa o indirectamente, contribuyeron a la realización de este proyecto. Este logro es también de ustedes.

FICHA RESUMEN

TÍTULO DEL PROYECTO: Modelo de lenguaje natural para el análisis de conversaciones de WhatsApp entre un asesor de cobranzas y un cliente

1. ÁREA DE TRABAJO: Banco Finandina
2. TIPO DE PROYECTO (Aplicado, Innovación, Investigación): Aplicado
3. ESTUDIANTE(S): Jesús David Alvear Corro
4. CORREO ELECTRÓNICO: jesalvc@javerianacali.edu.co
5. DIRECCIÓN Y TELEFONO: Calle 179 # 6 – 41 int 3 apto 502, +57 3168564635
6. DIRECTOR: Carlos Ernesto Ramírez Ovalle
7. VINCULACIÓN DEL DIRECTOR:
8. CORREO ELECTRÓNICO DEL DIRECTOR: carlosovalle@javerianacali.edu.co
9. CO-DIRECTOR (Si aplica): Abel Álvarez Bustos
10. GRUPO O EMPRESA QUE LO AVALA (Si aplica): Banco Finandina
11. OTROS GRUPOS O EMPRESAS:
12. PALABRAS CLAVE (al menos 5): Modelos, Lenguaje, Texto, Conversaciones, Redes
13. FECHA DE INICIO: 15 de Julio de 2024
14. DURACIÓN ESTIMADA (En meses): 8-10 Meses
15. RESUMEN:

El siguiente proyecto aborda la necesidad de mejorar la gestión de cobranzas en entidades bancarias dado el repentino aumento de la cartera vencida, lo que pone en presión la eficacia de las estrategias de cobranzas, el objetivo del proyecto se enfoca en la implementación de un modelo avanzado de procesamiento de lenguaje natural (PLN) como BERT. Este analiza las conversaciones de WhatsApp entre asesores de cobranza y clientes, para mejorar la eficiencia en la gestión de cobranzas identificando patrones de comportamiento y analizando el sentimiento expresado en estas interacciones.

El desarrollo del modelo se estructura en varias fases clave. Primero, se establece una robusta gestión de datos y creación de pipelines para capturar sistemáticamente las conversaciones

de WhatsApp. Esto incluye el almacenamiento, procesamiento, limpieza, normalización y entrenamiento de los datos, asegurando una base sólida para el análisis posterior.

Se implementan modelos avanzados como BERT para evaluar la calidad de las conversaciones y analizar el sentimiento expresado por los clientes. Los modelos BERT se utilizarán para la clasificación de sentimientos, proporcionando así una comprensión profunda de la percepción y la actitud de los clientes frente a sus obligaciones financieras.

La evaluación del modelo es exhaustiva, considerando métricas estándar como precisión, exactitud y F1-score, así como la capacidad del modelo para generalizar a nuevas conversaciones y su interpretabilidad. Esto garantiza que el modelo seleccionado cumpla con los estándares de calidad necesarios para su implementación en un entorno operativo real.

Además de mejorar la eficiencia en la gestión de cobranzas, el proyecto logra proporcionar herramientas para una evaluación temprana del desempeño de los asesores. Esto se logra mediante el análisis automatizado de las interacciones, identificando conversaciones críticas y áreas donde los asesores podrían mejorar en la comunicación con los clientes.

Tabla de contenido

Lista de Figuras

Lista de tablas

1 INTRODUCCIÓN

Tradicionalmente, la gestión de cobro en las entidades bancarias se realizaba a través de llamadas telefónicas. Sin embargo, la tasa de contactabilidad y respuesta ha disminuido debido al auge de las comunicaciones mediante canales digitales como WhatsApp [2]. Lo que ha conllevado a que muchas empresas han adoptado este medio como su principal canal de comunicación con los clientes, lo que ha generado nuevas problemáticas, especialmente en entornos donde se realizan miles de conversaciones de WhatsApp diarias. La evaluación precisa de la gestión, especialmente en áreas críticas como la recuperación de cobros en bancos, se ha vuelto esencial.

El problema que se pretende resolver con la ejecución de este proyecto es la necesidad de analizar y evaluar eficientemente las conversaciones de WhatsApp entre asesores y clientes para identificar las necesidades de los clientes, los motivos de impago, las alternativas ofrecidas y las soluciones propuestas. Dada la gran cantidad de datos generados diariamente, se requiere un modelo de lenguaje natural que permita evaluar estas conversaciones de manera rápida y constante.

La implementación de esta solución está diseñada para mejorar significativamente la operación en múltiples aspectos críticos. En primer lugar, aumentaría la eficiencia operativa dado que la gestión de cobro es un proceso vital para las entidades bancarias y la incapacidad de analizar y responder adecuadamente a las conversaciones de WhatsApp puede incrementar el tiempo y los recursos necesarios para recuperar pagos. En segundo lugar, mejoraría la satisfacción del cliente. Un modelo de lenguaje natural puede identificar y atender rápidamente las necesidades de los clientes, fortaleciendo la relación con ellos.

El propósito fundamental de este proyecto implementando un sistema basado en modelos de procesamiento de lenguaje natural para optimizar la gestión de cobranzas mediante la plataforma WhatsApp. Inicialmente, se desarrolló un flujo de extracción, transformación y carga (ETL) para gestionar eficazmente el almacenamiento y procesamiento de la información pertinente. Posteriormente, se llevará a cabo la tokenización y normalización de los datos con el fin de entrenar un modelo de aprendizaje automático apropiado. El modelo se validó con pruebas de bondad de ajuste, análisis de métricas de clasificación como precisión, exhaustividad y F1-score, y evaluaciones de curvas ROC-AUC. Una vez validado, se implementó para su operacionalización.

El objetivo de este enfoque es mejorar significativamente la eficiencia y efectividad de la comunicación con clientes y asesores, proporcionando a los coordinadores, directores y gerentes una visualización clara y oportuna de la gestión de cobranzas. Esta capacidad facilitará la evaluación de la efectividad de las campañas, la prontitud en la respuesta a mensajes, y, en última instancia, la toma de decisiones informadas que promuevan mejoras operativas y la recuperación de la cartera vencida.

2 CONTEXTUALIZACIÓN DEL PROYECTO

2.1 DEFINICIÓN DEL PROBLEMA

Muchos bancos enfrentan un aumento significativo en su cartera vencida, debido al comportamiento de los clientes que han dejado de pagar sus obligaciones, desde mayo del 2023 hasta abril del 2024 la cartera vencida de los bancos nacionales y extranjeros ha aumentado en promedio 17.5% [1] lo que ha generado una presión adicional en la gestión de cobranzas.

BANCOS	CARTERA_VENCIDAD_MAYO_2023	CARTERA_VENCIDAD_ABRIL_2024	AUMENTO_PORCENTUAL
Banco de Bogotá	$3,471,468.59	$3,731,474.01	14.61%
Banco Popular	$838,025.06	$846,806.18	10.62%
Bancolombia	$7,559,417.63	$8,415,089.08	15.60%
Banco de Occidente	$1,488,153.60	$1,837,115.26	19.12%
Banco Caja Social S.A.	$689,478.39	$802,669.89	18.95%
Banco Davivienda	$6,784,159.87	$7,024,869.29	3.69%
AV Villas	$461,748.06	$500,630.81	9.18%
Bancien	$82,574.12	$70,315.51	-26.48%
Bancamía S.A.	$120,598.99	$160,474.87	71.71%
Banco W S.A.	$87,819.35	$114,172.94	49.33%
Bancoomeva	$239,374.17	$258,380.99	5.99%

Finandina	**$198,684.73**	**$265,286.57**	**41.92%**
Coopcentral	$23,942.60	$21,743.72	-2.54%
Banco Mundo Mujer S.A.	$123,352.09	$162,788.44	45.71%
Mibanco S.A.	$74,512.98	$112,236.66	72.13%
Banco Serfinanza S.A.	$148,032.43	$192,872.94	33.88%
Lulo Bank	$31,731.69	$33,277.80	4.98%
Banco Unión	$31,526.63	$43,481.66	36.12%
Itaú	$847,021.05	$791,192.11	-22.25%
Banco GNB Sudameris	$146,180.31	$139,137.11	-1.59%
BBVA Colombia	$2,429,824.19	$2,992,229.64	26.95%
Scotiabank Colpatria S.A.	$1,369,346.11	$1,537,382.84	14.19%
Banco Falabella S.A.	$715,061.93	$555,028.25	-26.35%
Banco Pichincha S.A.	$190,549.17	$137,900.25	-25.46%
Banco Santander	$208,611.07	$270,678.71	46.49%

Datos en miles de millones

Tabla 1 Estado de la cartera de las entidades bancarias datos entregados por la Superintendencia Financiera de Colombia tomados de [1].

2.1.1 PLANTEAMIENTO DEL PROBLEMA

En el ámbito de las entidades financieras, la eficacia en la recuperación y mejora de la cartera de clientes es crucial para mantener la salud financiera y los indicadores en niveles óptimos. A pesar de aumentar el personal de gestión con el objetivo de incrementar los contactos con los clientes, se ha observado que este aumento no se traduce proporcionalmente en mejoras en la eficiencia operativa ni en la calidad del servicio durante las interacciones mediante conversaciones de WhatsApp.

Al analizar las interacciones entre los asesores y los clientes, se identifican varios síntomas preocupantes: respuestas inadecuadas, tiempos de espera prolongados y oportunidades de negocio desaprovechadas. Estos problemas no solo generan un cuello de botella operativo, sino que también impactan negativamente en la experiencia del cliente y en los resultados financieros de la entidad.

Estos síntomas son alarmantes porque, de persistir la situación actual, se prevé una disminución en la efectividad de la recuperación de la cartera, a pesar del aumento en el personal. Además, se anticipa un incremento en la insatisfacción de los clientes y un creciente cuello de botella debido al incremento de personal sin un seguimiento ni capacitación adecuados.

Adicionalmente, la falta de herramientas efectivas para evaluar y monitorear de manera rápida y precisa las interacciones individuales entre los asesores y los clientes agrava la situación. La ausencia de un sistema robusto de análisis de datos limita la capacidad de la entidad para identificar áreas de mejora específicas, implementar programas de capacitación efectivos y realizar ajustes estratégicos en tiempo real.

Desde la perspectiva de la ciencia de datos, la aplicación avanzada de modelos de procesamiento de lenguaje natural representa una herramienta poderosa para analizar detalladamente las interacciones entre asesores y clientes en las entidades financieras. Estos modelos pueden examinar conversaciones en profundidad, lo que permitiría descubrir hallazgos significativos en la gestión de los asesores y mejorar la recuperación de la cartera.

2.1.2 FORMULACIÓN DEL PROBLEMA

2.1.3 PREGUNTA GENERAL

¿De qué manera los modelos de procesamiento de lenguaje natural pueden aumentar la eficiencia y efectividad de la gestión de cobranzas durante una crisis bancaria, mediante el análisis de conversaciones de WhatsApp entre clientes morosos y asesores de cobranza?

2.1.4 PREGUNTAS DE SISTEMATIZACIÓN

- ¿Cómo se puede medir la calificación individual de cada asesor por su gestión a través de WhatsApp?
- ¿Qué factores en las conversaciones actuales se correlacionan con las oportunidades de negocio perdidas?
- ¿Qué tipos de preguntas y quejas hay más frecuentes de los clientes?
- ¿Cómo se debe almacenar la información para facilitar su análisis?
- ¿De qué manera se pueden incorporar estrategias de análisis de texto en los procesos de cobranzas para mejorar los resultados?
- ¿Qué técnicas de procesamiento de lenguaje natural deberían utilizarse para construir un modelo efectivo?

2.2 OBJETIVOS

2.2.1 OBJETIVO GENERAL

Implementar modelos de lenguaje natural (PNL) existentes, como BERT para evaluar la calidad de las conversaciones entre un asesor y un cliente, y extraer elementos significativos de ellas.

2.2.2 OBJETIVOS ESPECÍFICOS

1. Implementar una herramienta de recolección y adquisición de textos provenientes desde una API de WhatsApp y el CRM que es el software donde la empresa y gestiona sus interacciones y relaciones con los clientes que para nuestro caso es Five9.

2. Etiquetar de manera manual un conjunto de datos del total poblacional desde la formación del experto, para realizar una etapa de transferencia de conocimiento usando modelos XGBoost para lograr etiquetar el resto de las conversaciones.

3. Analizar las interacciones entre asesores y clientes, con el fin de evaluar la calidad de la conversación, identificando patrones claves entre ellas.

4. Evaluar el rendimiento del modelo de PLN mediante métricas estándar como precisión, exactitud (accuracy), F1-score, y ajuste de modelo, optimizando los hiperparámetros para seleccionar el modelo más adecuado que cumpla con los criterios de calidad establecidos.

5. Implementar el modelo en un entorno operativo, que permita un análisis y generación de reportes sobre la gestión.

2.3 MARCO DE REFERENCIA

2.3.1 MARCO TEÓRICO

El procesamiento de Lenguaje Natural (PNL) tiene un inmenso potencial para transformar la atención al cliente. Al implementar técnicas de PNL basadas en redes neuronales como LSTM y modelos como Bert uno de los beneficios más notables es su capacidad para realizar análisis de sentimiento de manera precisa y automatizada [17] que permiten a las compañías identificar clientes insatisfechos, priorizar respuestas y adaptar estrategias de comunicación para mejor el servicio al cliente.

La extracción de información clave es otra área en la que el PNL brilla [10]. Los modelos pueden identificar y extraer automáticamente información relevante de las interacciones con los clientes, como nombres de productos, números de pedido, problemas técnicos o cualquier otro dato que pueda ser útil para mejorar la atención al cliente. Esto permite a las empresas tomar decisiones más informadas y ofrecer soluciones más rápidas y eficientes.

El Procesamiento de Lenguaje Natural (PNL) es una subdisciplina de la inteligencia artificial que se centra en la interacción entre las computadoras y el lenguaje humano. Su principal objetivo es capacitar a las máquinas para analizar el lenguaje humano, permitiéndoles entender, interpretar y generar texto de manera eficaz. Uno de los algoritmos más usados en estos modelos son las Redes Neuronales que buscan representar el funcionamiento Biológico de una "Neurona Humana" [14]. En este proyecto, se implementarán técnicas de PNL desarrolladas bajo redes neuronales como LSTM y otros modelos como BERT para lograr los objetivos planteados.

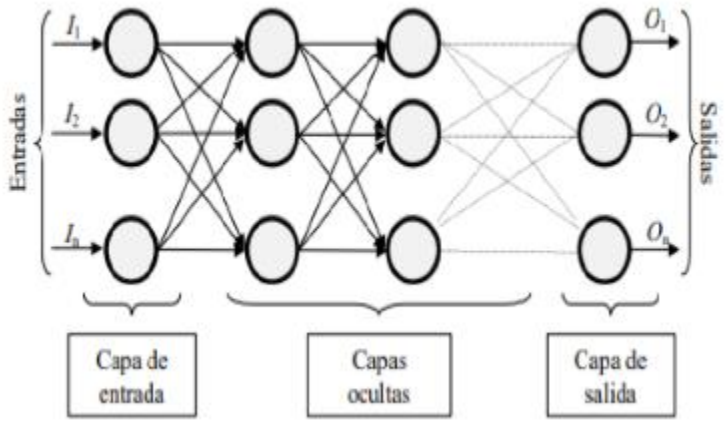

Figura 1Red Neuronal Tradicional tomado de [15]

El procesamiento de lenguaje natural ha tenido hitos en diferentes tipos de modelos. Los modelos más antiguos, como aquellos que utilizan reglas gramaticales explícitas y diccionarios para analizar y generar texto [13]. Sin embargo, los modelos más modernos, como las redes neuronales recurrentes (RNN) y LSTM, manejan secuencias de datos y capturan dependencias a largo plazo en el texto. Estas redes contienen una estructura que permite que la información persista, haciendo posible que una red "recuerde" elementos anteriores a la secuencia, como palabras, contextos, verbos, etc.

Redes Neuronales RNN:

Las redes neuronales RNN son un tipo de red neuronal artificial diseñada para procesar secuencias de datos, como texto, audio o series temporales, donde la posición de los datos es importante. Entre otras aplicaciones los modelos RNN tienen conexiones retroalimentadas que les permiten mantener información previa a medida que procesan nuevas entradas. Este enfoque las hace especiales para tareas donde la historia de los datos es crucial para la predicción o generación de datos. En nuestro caso, una red RNN es valiosa, ya que permitirá entender el contexto de la conversación en cada mensaje [3].

Redes Neuronales LSTM:

Las redes neuronales LSTM es una arquitectura de una RNN que introduce una celda de memoria y mecanismos de puerta que controlan el flujo de la información. Una celda de memoria es un componente central que va reteniendo la información a lo largo de las secuencias. La puerta de entrada, olvido y salida se encarga de controlar cómo se actualiza la celda de memoria y cómo la información se propaga en la red, lo que la hace una mejor red que una RNN estándar. Sin embargo, conserva los fundamentos de una red neuronal recurrente [4].

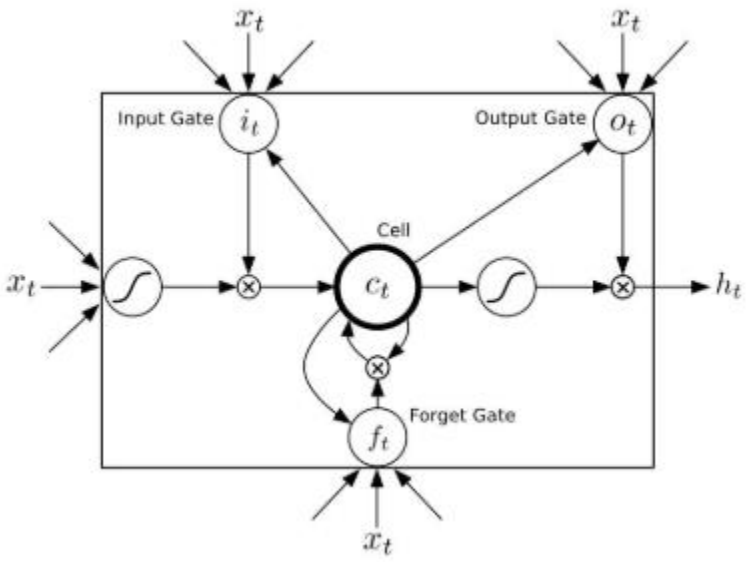

Figura 2 Celda de memoria a corto plazo tomado de [16]

Modelos Transformers:

Los modelos modernos hacen uso de nuevas técnicas como los Transformers una arquitectura revolucionaria son la base de grandes modelos como ChatGPT o Google Bard [5]. Estos modelos, que son la base de un procesamiento paralelo de secuencias de texto. A diferencia de los modelos RNN o LSTM, que procesan la información secuencialmente, los Transformers pueden analizar todos los elementos de la secuencia simultáneamente, lo que mejora la eficiencia y permite modelar relaciones más complejas [6]. El componente más clave de la arquitectura Transformer es el mecanismo de atención, que permite al modelo considerar simultáneamente todas las palabras en una oración para así asignar diferentes niveles de importancia a diferentes palabras en la secuencia lo que lo hace sumamente poderoso en la generación de texto y compresión de lenguaje natural [6].

Modelo XGBoost

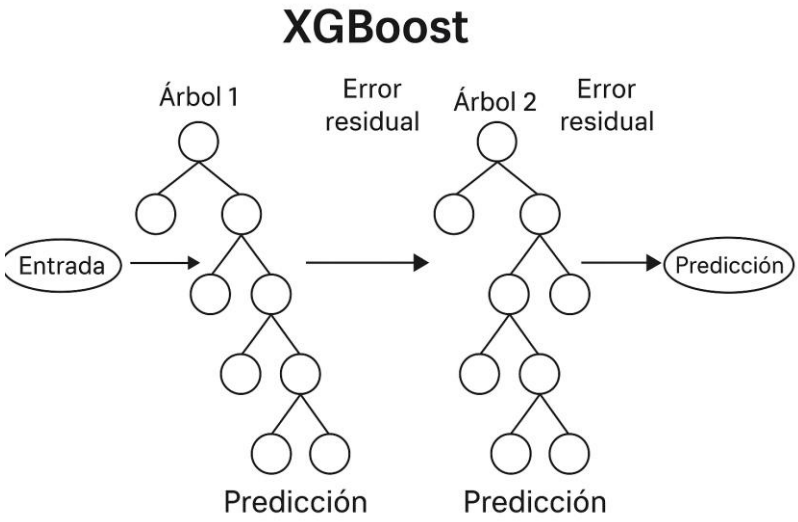

Figura 3 Algoritmo XGBoost

XGBoost (Extreme Gradient Boosting) es una técnica de aprendizaje supervisado basada en árboles de decisión, ampliamente utilizada por su eficiencia y precisión en tareas de clasificación y regresión. A diferencia de algoritmos como Random Forest, donde los árboles se construyen de forma independiente y suelen crecer hasta su máxima profundidad, en XGBoost los árboles son generados de manera secuencial, y el usuario tiene control sobre su profundidad máxima, lo que permite un mayor ajuste y control sobre la complejidad del modelo.

Este algoritmo incorpora múltiples optimizaciones que lo diferencian de otros enfoques de ensamble: utiliza procesamiento en paralelo, realiza poda de árboles durante la construcción para reducir sobreajuste, y maneja eficientemente valores ausentes. Cabe resaltar que, en lugar de predecir directamente el objetivo, cada árbol adicional se enfoca en modelar los errores cometidos previamente. Finalmente, las salidas de todos los árboles se combinan de forma ponderada para generar una predicción final más precisa, lo que refleja la capacidad del modelo para optimizar el rendimiento de manera aditiva y progresiva. Además, incluye términos de regularización en la función objetivo, lo que mejora la capacidad del modelo para generalizar, evitando sesgos y sobreajustes [19]

Gracias a estas características, XGBoost ha sido ampliamente adoptado en aplicaciones reales donde se requiere alto rendimiento predictivo con tiempos de entrenamiento razonables, siendo especialmente competitivo en competiciones y casos donde los datos estructurados son predominantes.

Modelo BERT

BERT (Bidirectional Encoder Representations from Transformers) es un modelo de lenguaje desarrollado por Google en 2018, basado en la arquitectura Transformer propuesta por Vaswani et al. (2017). A diferencia de modelos anteriores como Word2Vec o LSTM, que procesan el texto de forma secuencial (de izquierda a derecha o viceversa), BERT introduce un enfoque bidireccional que permite capturar el contexto completo de una palabra considerando simultáneamente las palabras que la anteceden y las que la siguen. Este mecanismo mejora significativamente la comprensión semántica del lenguaje.

El modelo BERT se construye sobre una arquitectura de redes neuronales conocida como Transformer, introducida por Vaswani et al. en 2017. Esta arquitectura representó un cambio paradigmático en el procesamiento de lenguaje natural (PLN), al abandonar el procesamiento secuencial característico de modelos como las redes recurrentes (RNN) o las redes LSTM, en favor de un procesamiento completamente paralelo que permite analizar todas las palabras de una oración simultáneamente.

El componente central del Transformer es el mecanismo de atención, especialmente la atención múltiple o *multi-head attention*. Este mecanismo permite al modelo asignar diferentes niveles de importancia a cada palabra dentro de una secuencia, considerando el contexto completo. Por ejemplo, en la frase “El cliente no pagó porque perdió su empleo”, el modelo puede identificar que la palabra *“perdió”* tiene una relación directa con *“no pagó”*, a pesar de que estén separadas por otras palabras.

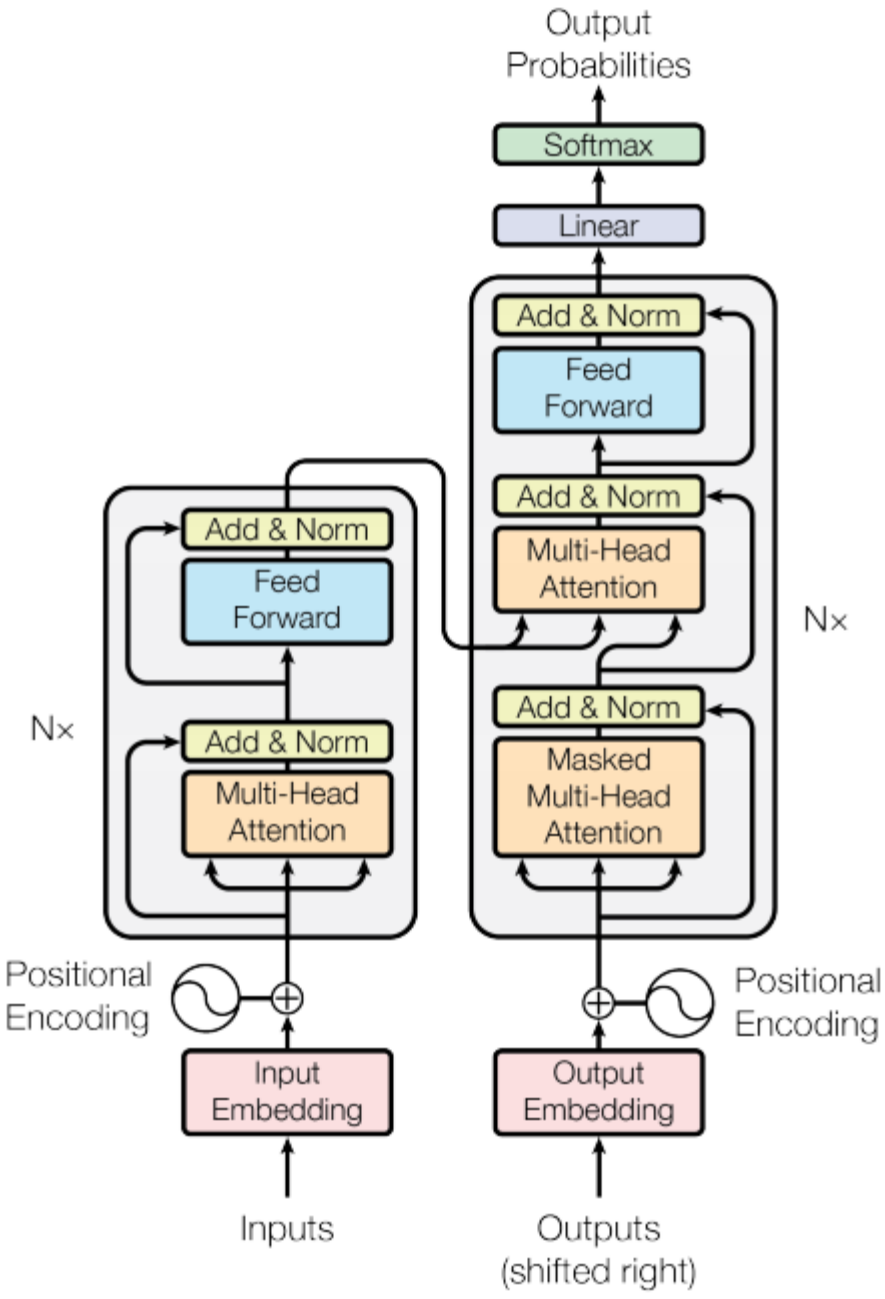

Figura 4 The Transformer Arquitectura modelo

El modelo BERT es una adaptación del Transformer, preentrenado en grandes cantidades de datos utilizando una técnica de entrenamiento no supervisada llamada enmascaramiento de palabras y predicción de la siguiente oración. BERT es un modelo con múltiples capas de encoders bidireccionales, lo que permite analizar el texto en ambas direcciones (de izquierda a derecha y de derecha a izquierda) para capturar el contexto completo de cada palabra es de los modelos más poderosos actualmente [7].

La arquitectura Transformer se compone de los siguientes elementos principales:

- **Capas de codificación (encoders)**: procesan la secuencia de entrada para generar representaciones contextuales de cada palabra.
- **Capas de decodificación (decoders)**: utilizadas principalmente en tareas de generación de texto, como la traducción automática.
- **Capas de atención**: permiten que el modelo enfoque su análisis en las partes más relevantes del texto, a través del cálculo de pesos de atención.

Gracias a esta capacidad de contextualización profunda y procesamiento paralelo, el Transformer ha demostrado ser excepcionalmente eficaz para tareas como clasificación de texto, análisis de sentimiento, respuesta a preguntas, resumen automático y otras aplicaciones de PLN. Modelos como BERT, GPT, T5 y RoBERTa están todos construidos sobre esta arquitectura.

Las dos principales tareas del modelo BERT son:

1. Masked Language Modeling: El objetivo de esta tarea es permitir que el modelo comprenda el contexto bidireccional de las palabras en una oración. Es una técnica donde se selecciona aleatoriamente un porcentaje de palabras del texto de entrada y se enmascaran. La tarea del modelo será predecir estas palabras basándose en el contexto proporcionado por las demás palabras de la oración. Por ejemplo, en la frase "El gato está en el jardín", si se enmascara la palabra "gato", el modelo verá "El [MASK] está en el jardín" y deberá predecir "gato" [7].

2. Next Sentence Prediction: Esta tarea tiene como objetivo ayudar al modelo a entender las relaciones entre oraciones. Dado que BERT es un modelo bidireccional y entiende el contexto de la información se entrena para predecir si una segunda oración sigue lógicamente a una primera por ejemplo 1. “El sol brilla intensamente”, 2.” No hay nubes en el horizonte”, durante el entramiento de BERT aprenderá a predecir si la segunda oración (2) sigue lógicamente a la primera oración (1) [7] dado que en muchos casos una segunda oración proporciona información adicional sobre el evento, en nuestro caso las conversaciones de WhatsApp donde estas relaciones serán más evidentes.

Estas dos técnicas se usan para entrenar modelos con datos no etiquetados. Esto hace que BERT sea fuerte en la clasificación de texto, reconocimiento de sentimientos.

Dada las características de los modelos BERT estos modelos son sumamente útiles en el manejo de auto etiquetado que buscamos en la información dado que funciona con parámetros de incrustación de palabras previamente entrenados a partir de textos sin etiquetar [7]. que tenemos a partir de las conversaciones de WhatsApp adicional tiene un gran rendimiento en el reconocimiento de patrones y sentimientos en las conversaciones.

Preprocesamiento en el lenguaje natural

La ciencia de datos en modelos de lenguaje natural está compuesta por diferentes partes; una de ellas es el procesamiento de datos, momento crucial para la creación de modelos de lenguaje natural. Este proceso prepara el texto para que los algoritmos puedan procesarlo eficientemente. Algunas de las técnicas más comunes incluyen:

- Tokenización: Es el proceso de dividir el texto en unidades más pequeñas llamadas tokens, que pueden ser palabras, caracteres o subpalabras. La tokenización facilita el análisis de texto y la construcción de vocabularios. Por ejemplo, el texto original "El gato negro" después de un proceso de tokenización quedaría como ["El", "gato", "negro"] [8].

- Eliminación de StopWords: Las stopwords son palabras comunes que no aportan mucho significado en la comprensión del texto, como "el", "de", y "y". Eliminarlas reduce el ruido, ya que muchos modelos avanzados actuales no son tan precisos como para procesar estas palabras de manera efectiva. Estos cambios ayudan a mejorar la eficiencia del modelo. Sin embargo, es importante aclarar que las palabras vacías poseen contenido semántico; por ejemplo, "la persona" tiene un significado ligeramente diferente al de "una persona". A pesar de esto, la eliminación de stopwords permite que los modelos se centren en las palabras distintivas, lo que contribuye a un mejor rendimiento [8].

- Lematización: La lematización convierte las palabras a su forma base, considerando el contexto y las reglas gramaticales del lenguaje, esto es vital ya que reduce el número de palabras únicas en el texto por ejemplo “disfruta” y “disfruto” se asignarían al mismo token “disfruta” [8].

- Normalización de texto: Consiste en estandarizar el texto para que sea más coherente y fácil de procesar mediante la eliminación de caracteres especiales, corrección de ortografía por ejemplo “camin” se convertiría en “camión” esto ayuda a reducir significativamente la cantidad de palabras en el análisis, el uso adecuado de las mayúsculas [8] este punto es importante dado que en el contexto de la oración o conversación como es el caso de nuestro proyecto una palabra en mayúscula podría querer trasmitir una emoción diferente.

- Word Embeddings: Los embeddings de palabras son representaciones vectoriales de palabras en un espacio multidimensional, donde cada palabra se asigna a un vector de números reales. Estas representaciones son fundamentales en la creación de modelos de procesamiento natural, ya que capturan el significado semántico y las relaciones entre palabras. Por ejemplo, las palabras "rey" y "reina" tienen una distancia corta, al igual que "hombre" y "mujer", en un espacio vectorial [8].

- Vectorización: Vectorización es el proceso mediante el cual los datos no estructurados (como texto, imágenes o sonido) se transforman en representaciones numéricas o vectores, con el fin de que puedan ser procesados por algoritmos de aprendizaje automático o modelos estadísticos. Esta transformación es fundamental en tareas de clasificación, predicción y análisis automatizado, ya que permite aplicar operaciones matemáticas sobre los datos.[22]

Esto es un preprocesamiento que se deberá realizar con las fuentes de información, con el fin de que el modelo reciba la mejor información posible que permitirá obtener el mejor rendimiento en el entrenamiento del modelo.

SMOTE

SMOTE es una técnica de sobremuestreo sintético desarrollada por Chawla et al. (2002) para abordar el problema del desbalance de clases en conjuntos de datos de aprendizaje automático. Esta técnica se fundamenta en la generación artificial de ejemplos de la clase minoritaria mediante interpolación, en lugar de simplemente duplicar instancias existentes.[20]

El algoritmo SMOTE opera bajo el principio de que las instancias sintéticas creadas en el espacio de características entre ejemplos existentes de la clase minoritaria mantienen las propiedades estadísticas de la distribución original. La técnica utiliza el concepto de vecinos más cercanos (k-NN) para identificar instancias similares y generar nuevos ejemplos a lo largo de los segmentos de línea que conectan estas instancias. [20]

El proceso de SMOTE se ejecuta en las siguientes etapas:

1. **Selección de instancia**: Para cada ejemplo de la clase minoritaria, se identifican sus k vecinos más cercanos dentro de la misma clase.
2. **Generación sintética**: Se selecciona aleatoriamente uno de los k vecinos y se crea un nuevo ejemplo mediante interpolación lineal entre la instancia original y el vecino seleccionado.
3. **Interpolación**: La nueva instancia sintética se genera usando la fórmula: $x\_nuevo = x\_i + \lambda \times (x\_j - x\_i)$ Donde x_i es la instancia original, x_j es el vecino seleccionado, y λ es un número aleatorio entre 0 y 1. [20]

Muestre en Aprendizaje Automatico

El **muestreo** es un procedimiento estadístico mediante el cual se selecciona un subconjunto de elementos (llamado muestra) a partir de una población más amplia, con el fin de inferir características o comportamientos de dicha población.

El propósito principal del muestreo es realizar generalizaciones válidas y precisas, minimizando sesgos y errores, mientras se optimizan recursos como tiempo, dinero y esfuerzo. [21]

Existen diferentes tipos de muestreo

En este proyecto enfocamos en los muestreos probabilísticos

1. Muestreo Aleatorio Simple: Técnica en la cual cada elemento de la población tiene la misma probabilidad de ser seleccionado, y la selección de un elemento no afecta la probabilidad de selección de otros elementos. Se implementa mediante tablas de números aleatorios o generadores pseudoaleatorios. [21]

2. Muestreo sistemático: Consiste en seleccionar elementos de la población siguiendo un intervalo fijo (k) después de un inicio aleatorio. El intervalo se calcula dividiendo el tamaño poblacional entre el tamaño muestral deseado. [21]

3. Muestreo Estratificado: Técnica que divide la población en subgrupos homogéneos (estratos) basados en características relevantes, seleccionando posteriormente muestras independientes de cada estrato. [21]

4. Muestreo por conglomerados: Método que divide la población en grupos naturales (conglomerados) y selecciona aleatoriamente algunos de estos grupos para incluir todos sus elementos en la muestra. [21]

2.3.2 ANTECEDENTES

En el ámbito del procesamiento de lenguaje natural (PLN) aplicado a la gestión de cobranzas y comunicaciones en entornos bancarios, se han realizado diversos estudios y desarrollos que abordan problemas similares. A continuación, se presentan algunos trabajos relevantes que, aunque abordan diferentes problemáticas, utilizan técnicas y estrategias aplicables a nuestro proyecto.

1. “Automated Customer Service Chatbot for Banking Sector” - IEEE Conference, 2020 [10]

Este estudio presenta el desarrollo e implementación de un chatbot automatizado para el sector bancario, utilizando técnicas avanzadas de procesamiento de lenguaje natural, como Transformers. El chatbot fue utilizado para manejar consultas comunes de los clientes y reducir la carga de trabajo del servicio al cliente.

El principal aporte de este modelo fue la demostración de que los modelos NLP pueden ser aplicados para automatizar y mejorar la atención a los clientes en el sector bancario. La principal diferencia entre este proyecto y el nuestro es que ellos se centraron en la creación de un chatbot para manejar consultas generales, mientras que nuestro enfoque está en analizar y evaluar la calidad de las conversaciones de WhatsApp entre asesores y clientes.

2."Natural Language Processing Techniques for Analyzing Financial Sentiment in Social Media" - Journal of Financial Technology, 2021 [11]

Este artículo explora el uso de técnicas de NLP para analizar el sentimiento financiero en las redes sociales. Utiliza modelos como LSTM y BERT para clasificar el sentimiento de las publicaciones y predecir tendencias del mercado basadas en la percepción pública.

El trabajo destaca cómo las técnicas de NLP pueden ser utilizadas para captar y analizar sentimientos financieros, proporcionando una visión clara de las percepciones del mercado. La diferencia con nuestro proyecto radica en el contexto y el propósito: el estudio se centra en redes sociales y análisis de sentimiento financiero, mientras que nuestro proyecto se enfoca en la gestión de cobranzas a través de WhatsApp y la evaluación de la calidad de las conversaciones entre asesores y clientes.

3. "Improving Debt Collection Efficiency Using Machine Learning and Natural Language Processing" - Proceedings of the AAAI Conference on Artificial Intelligence, 2019 [12]

Este estudio investiga el uso de técnicas de aprendizaje automático y PLN para mejorar la eficiencia en la gestión de cobranzas. El enfoque incluye la clasificación de conversaciones de cobranza, identificación de patrones de comportamiento y automatización de respuestas, utilizando modelos como LSTM y técnicas de vectorización de texto.

El principal aporte de este trabajo es la aplicación directa de técnicas de PLN para mejorar la eficiencia en la gestión de cobranzas, alineándose estrechamente con nuestro objetivo. La diferencia clave es que el estudio se centra en la automatización y optimización de procesos específicos de cobranza, mientras que nuestro proyecto también busca evaluar la calidad de las interacciones y proporcionar resultados detallados sobre las necesidades de los clientes y el desempeño de los asesores.

3 METODOLOGIA

3.1 HERRAMIENTAS DE SOFTWARE UTILIZADAS

Five9: Es una plataforma de software en la nube especializada en centros de contacto y atención al cliente. Está diseñada para ayudar a las empresas a gestionar de manera eficiente las interacciones con sus clientes a través de múltiples canales, como teléfono, correo electrónico, chat, redes sociales y SMS. Version 12 utilizada.

Selenium: Es una herramienta de código abierto diseñada para automatizar navegadores web. Permite a los desarrolladores y testers controlar y simular la interacción con páginas web de forma programada, facilitando pruebas automatizadas, extracción de datos y tareas repetitivas en sitios web. Selenium es compatible con múltiples lenguajes de programación, como Python, Java y JavaScript, y funciona con distintos navegadores, lo que la convierte en una solución flexible y muy utilizada en el desarrollo y aseguramiento de la calidad de aplicaciones web versión utilizada 4.32.0.

Python: es un lenguaje de programación de alto nivel, interpretado y de propósito general, conocido por su sintaxis clara y sencilla, que facilita el desarrollo rápido de aplicaciones en diversos campos como la ciencia de datos, desarrollo web, automatización, inteligencia artificial y más versión utilizada Python 3.11.0.

SQL Server: e un sistema de gestión de bases de datos relacional desarrollado por Microsoft, que permite almacenar, administrar y consultar grandes volúmenes de datos de manera segura y eficiente. Es ampliamente utilizado en entornos empresariales para gestionar información,

ejecutar consultas complejas, realizar análisis y apoyar aplicaciones que requieren acceso rápido y confiable a datos estructurados.

Knime: Es una plataforma de software de código abierto diseñada para el análisis de datos, integración, minería y creación de modelos de machine learning mediante flujos de trabajo visuales. Permite a los usuarios construir, ejecutar y automatizar procesos de análisis de datos sin necesidad de programar, usando nodos que representan diferentes pasos del análisis. knime es muy popular en áreas como la ciencia de datos, bioinformática y business intelligence por su facilidad de uso y capacidad para conectar con múltiples fuentes de datos y herramientas versión utilizada 3.0.0.

3.2 DATOS

3.2.1 Automatización de la Recolección y Almacenamiento de Conversaciones

Se logró implementar una herramienta eficiente para la recolección y adquisición de textos provenientes de una API de WhatsApp y del CRM utilizado por la empresa, que en este caso es Five9. Para llevar a cabo este proceso, se utilizó Selenium, una poderosa herramienta de automatización web. Selenium es una biblioteca de código abierto que permite automatizar las interacciones con navegadores web, simulando el comportamiento de un usuario real. Esto incluye tareas como hacer clic, escribir, navegar entre diferentes páginas y descargar datos.

En nuestro caso, Selenium fue empleado para simular el proceso de descarga de las conversaciones, emulando a una persona que interactúa con las interfaces web tanto de la API de WhatsApp como de Five9. Esta técnica permitió extraer los datos de manera continua y confiable, asegurando que se respetaran los tiempos de espera y las restricciones de las plataformas. El proceso de recolección se extendió por aproximadamente cinco días, ya que la simulación de interacciones humanas con Selenium requiere ejecutar el proceso de manera secuencial y paulatina para evitar la sobrecarga de los sistemas y asegurar la integridad de los datos. Los textos extraídos fueron posteriormente almacenados en una base de datos, asegurando su disponibilidad para futuros análisis y gestión de relaciones con los clientes.

Para el almacenamiento de las conversaciones recolectadas, se utilizó código Python. Python es un lenguaje de programación altamente versátil y utilizado en una amplia variedad de aplicaciones, incluyendo la automatización de tareas y la manipulación de datos a gran escala. En este caso, Python fue clave para gestionar la conexión y almacenamiento de las conversaciones extraídas en un servidor SQL Server, un sistema de gestión de bases de datos

relacionales robusto y escalable. En total, se descargaron 60,000 conversaciones, las cuales se almacenaron de forma segura en la base de datos para su posterior uso y análisis en la toma de decisiones empresariales.

El proceso completo, utilizando tanto Selenium como Python, permitió la automatización eficiente de una tarea compleja, asegurando que los datos estuvieran disponibles en tiempo y forma, con la integridad necesaria para el éxito de las operaciones del CRM y la mejora de la gestión de las relaciones con los clientes.

3.2.2 Estructuración de Conversaciones en formato JSON

Los siguientes pasos para seguir en este proyecto implican la conversión de las conversaciones almacenadas en la base de datos en un formato **JSON**. Este formato permitirá estructurar y diferenciar de manera clara los intercambios entre el asesor del banco y el cliente, incluyendo las fechas de los mensajes y las respectivas respuestas de ambas partes. Esta transformación es esencial para preparar los datos para un posterior modelado y análisis, facilitando la interpretación del contexto conversacional mediante técnicas de procesamiento de lenguaje natural (NLP, por sus siglas en inglés).

El formato **JSON** es ampliamente utilizado por su flexibilidad y capacidad para organizar los datos en estructuras de clave-valor, lo que permite representar las conversaciones de manera jerárquica. Para cada conversación, se generará un objeto JSON que contenga:

1. **Identificación del asesor y cliente**: Para saber quién está hablando en cada turno.
2. **Fecha y hora**: Para registrar la temporalidad de cada mensaje.
3. **Contenido del mensaje**: El texto real de la conversación.
4. **Turno conversacional**: Identificar si el mensaje es del asesor o del cliente.

Esto facilitará al modelo identificar patrones conversacionales, como el tipo de respuesta que suele dar el cliente en función de lo que dice el asesor, o el tiempo de respuesta promedio entre ambas partes. |

Preprocesamiento de los datos

El conjunto de datos original fue cargado desde un archivo en formato Excel, que contenía las conversaciones estructuradas en formato JSON y las calificaciones correspondientes en forma textual ("MALA", "BUENA"). Estas calificaciones fueron transformadas a etiquetas numéricas (0, 1) mediante un mapeo directo. Además, se aplicó una función personalizada para extraer los mensajes individuales de cada conversación y concatenarlos en un único texto plano, facilitando así su posterior tokenización.

3.2.3 Vectorización BERT

La representación vectorial de las conversaciones se llevó a cabo utilizando el modelo BERT (Bidirectional Encoder Representations from Transformers), específicamente la versión preentrenada bert-base-uncased. Este modelo permite mapear cada conversación a un espacio denso de alta dimensión, generando una representación semántica global que puede ser utilizada por modelos de clasificación supervisados.

El proceso se compone de varias etapas:

1. Tokenización: Las cadenas de texto (conversaciones) fueron convertidas en secuencias de tokens utilizando el tokenizador oficial de BERT. Este tokenizador aplica la técnica de WordPiece, que fragmenta palabras en subunidades semánticas más pequeñas, permitiendo manejar vocabulario abierto de forma eficiente.
2. Truncamiento y relleno (padding): Para cumplir con las restricciones de entrada del modelo, cada secuencia fue truncada o rellenada hasta un máximo de 512 tokens, que corresponde al límite permitido por la arquitectura base de BERT.
3. Identificación de tokens especiales: Se añadieron los tokens [CLS] (al inicio de cada secuencia) y [SEP] (para delimitar fragmentos), tal como lo requiere la arquitectura para tareas de clasificación.

Una vez transformadas las secuencias, estas fueron procesadas en GPU para maximizar la eficiencia computacional. BERT produce una representación vectorial por cada token, pero en este caso se utilizó exclusivamente el embedding asociado al token [CLS], el cual está diseñado para capturar el significado general de la secuencia completa. Este vector, de 768 dimensiones, actúa como una "firma numérica" de la conversación, encapsulando su contenido semántico de forma compacta.

Esta representación densa se convirtió en la entrada para los modelos de clasificación utilizados en etapas posteriores (como XGBoost o el clasificador BERT ajustado), facilitando la identificación automática de patrones conversacionales relevantes para el análisis de la calidad de la interacción.

3.2.4 Balanceo de las clases

Durante el proceso de entrenamiento del modelo de transferencia de conocimiento, se identificó un desbalance significativo entre las clases etiquetadas como "BUENA" y "MALA". Este desequilibrio puede comprometer seriamente la capacidad del modelo para generalizar correctamente, dado que tiende a favorecer la clase mayoritaria, afectando la precisión de la clase minoritaria y distorsionando las métricas globales.

Para mitigar este problema, se implementó la técnica **SMOTE (Synthetic Minority Over-sampling Technique)**. SMOTE es una técnica de sobremuestreo que **genera nuevas instancias sintéticas** de la clase minoritaria en el espacio de características, en lugar de simplemente replicar ejemplos existentes.

La lógica detrás de SMOTE es la siguiente:

- Para cada observación minoritaria, se identifican sus **k vecinos más cercanos** (generalmente k=5).
- Luego, se crean nuevas observaciones interpolando entre la muestra original y alguno de sus vecinos seleccionados aleatoriamente.
- Estas nuevas instancias no son copias exactas, sino puntos intermedios dentro del espacio de características, lo que mejora la diversidad del conjunto de entrenamiento y reduce el sobreajuste.

En este proyecto, SMOTE fue aplicado sobre las representaciones vectorizadas de las conversaciones obtenidas a través de BERT ([CLS] embedding). El proceso es realizado antes del entrenamiento del modelo **XGBoost** encargado de la transferencia de conocimiento.

3.3 ETIQUETADO MANUAL

3.3.1 Método de selección para la muestra a etiquetar

En esta etapa se desarrolló un proceso de etiquetado manual sobre una muestra representativa del conjunto poblacional, con la intervención de un experto con amplio conocimiento del

dominio del negocio. El objetivo principal fue construir un conjunto de datos etiquetado con altos estándares de calidad, que sirviera como base para la implementación posterior de modelos de procesamiento de lenguaje natural (PLN).

Para garantizar la representatividad y calidad de la muestra, se seleccionaron aleatoriamente conversaciones que reflejaran la diversidad de escenarios y patrones típicos presentes en el total de interacciones. La selección de las 2.522 conversaciones se llevó a cabo mediante un muestreo estratificado basado en la longitud de los textos, medida por el número de palabras. El conjunto poblacional se dividió en tres rangos: conversaciones cortas, medias y largas. Posteriormente, se extrajo una muestra proporcional al número de registros en cada estrato, asegurando así una representatividad estructural del lenguaje.

Dentro de cada grupo, las conversaciones se seleccionaron de forma aleatoria utilizando una semilla predefinida, con el fin de garantizar la reproducibilidad del procedimiento. Este enfoque permitió capturar una diversidad significativa en términos de extensión y complejidad lingüística, aspecto fundamental para un entrenamiento robusto del modelo.

Además, se procuró balancear la muestra con base en las tres categorías principales identificadas en las conversaciones, favoreciendo una transferencia de conocimiento eficaz. En este contexto, dicha transferencia consiste en utilizar un conjunto de datos etiquetado manualmente como punto de partida para entrenar un modelo que pueda generalizar y clasificar automáticamente nuevas interacciones no vistas.

Uno de los principales desafíos encontrados fue la ambigüedad entre ciertas etiquetas, ya que algunas presentaban diferencias superficiales, pero correspondían a comportamientos conversacionales similares, lo que dificultó el aprendizaje del modelo y requirió una revisión cuidadosa por parte del experto.

3.3.2 Información descriptiva sobre la muestra

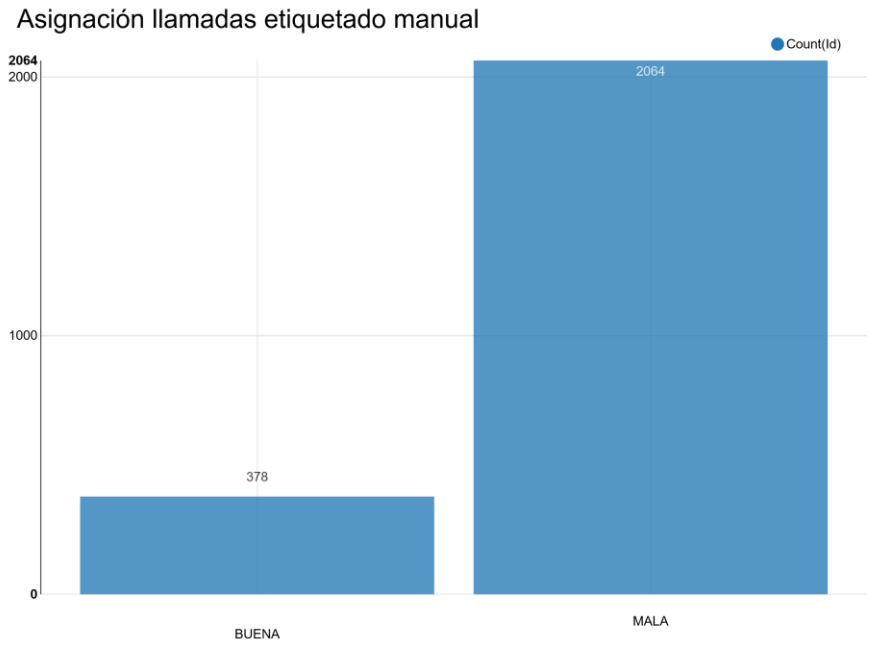

Figura 5 Resultado muestra por calificación etiquetado manual

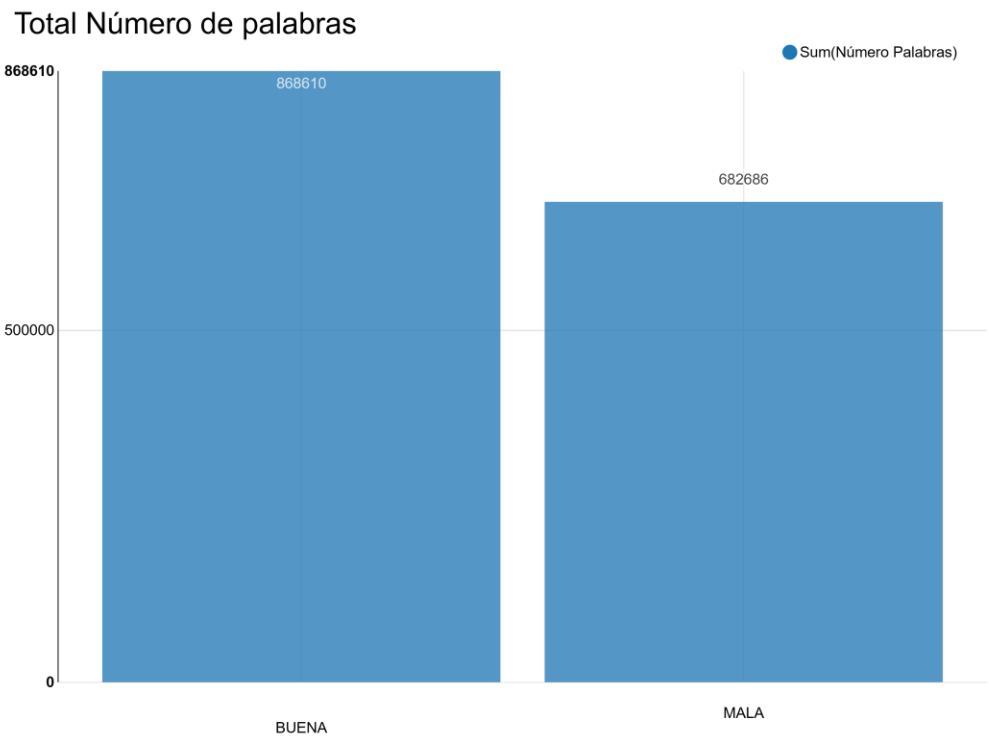

Figura 6 Total Palabras por calificación etiquetado manual

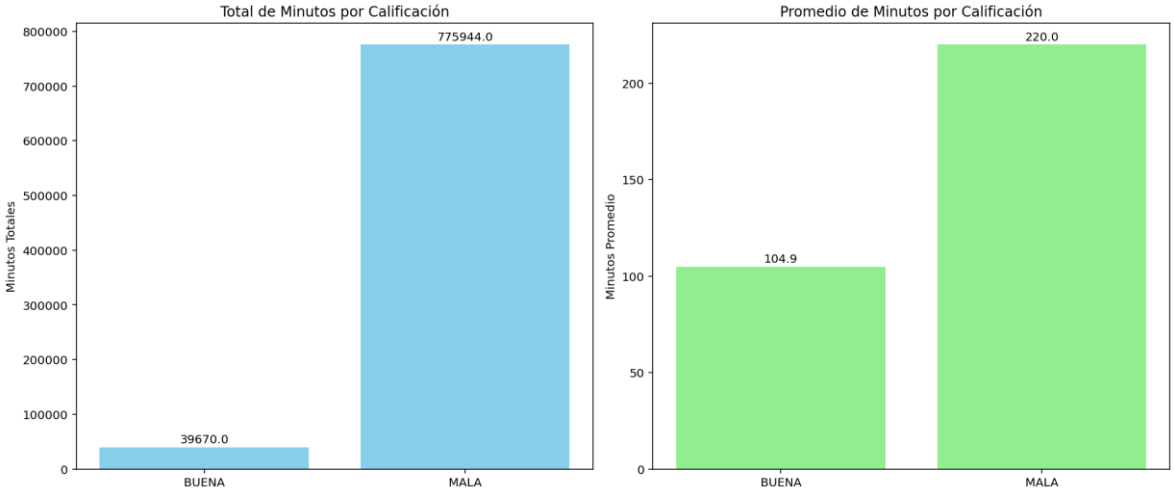

Figura 7 Promedio minutos por calificación etiquetado manual

3.4 DESARROLLO MODELO

3.4.1 Arquitectura general del modelo

Link Arquitectura modelo completa : Diagrama_Modelo

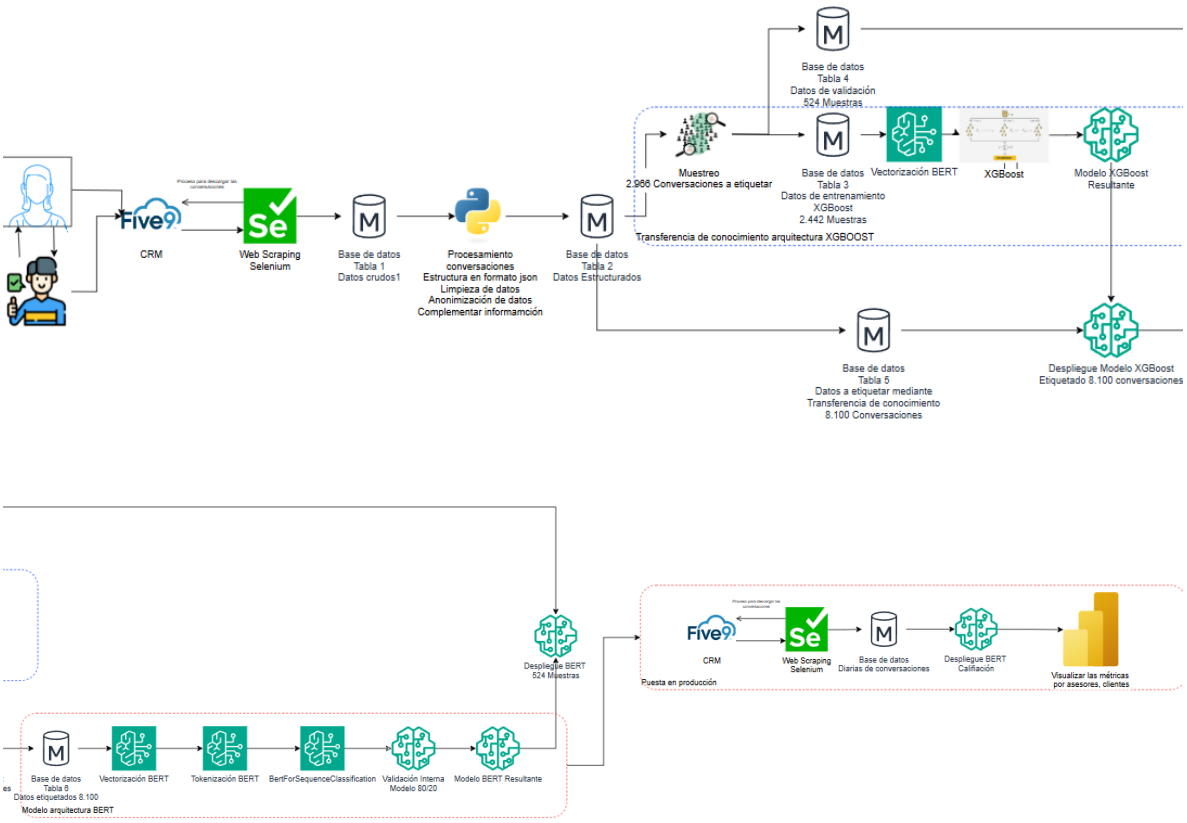

Figura 8 Arquitectura general del modelo

3.4.2 Modelo para desarrollar la transferencia de conocimiento

3.4.2.1 Descripción y enfoque del modelo

La primera etapa del proceso de modelado consistió en implementar una estrategia de transferencia de conocimiento, con el objetivo de entrenar un modelo capaz de clasificar la calidad de las conversaciones entre asesores y clientes. Para ello, se empleó una muestra representativa de 2.966 conversaciones previamente etiquetadas de forma manual por un experto del dominio.

A partir de esta muestra, se abordaron dos enfoques complementarios:

1. El desarrollo de un modelo XGBoost, un algoritmo de aprendizaje supervisado basado en árboles de decisión, que genera múltiples árboles de manera secuencial, donde cada nuevo árbol intenta corregir los errores del árbol anterior. Este algoritmo se fundamenta en el principio de boosting por gradiente, optimizando una función de pérdida mediante adiciones iterativas, lo que lo convierte en una herramienta altamente eficiente y precisa, especialmente en el tratamiento de datos estructurados.
2. El entrenamiento de un modelo BERT de clasificación directa, utilizando un conjunto de 8.100 conversaciones derivadas del despliegue inicial del modelo de transferencia basado en XGBoost.

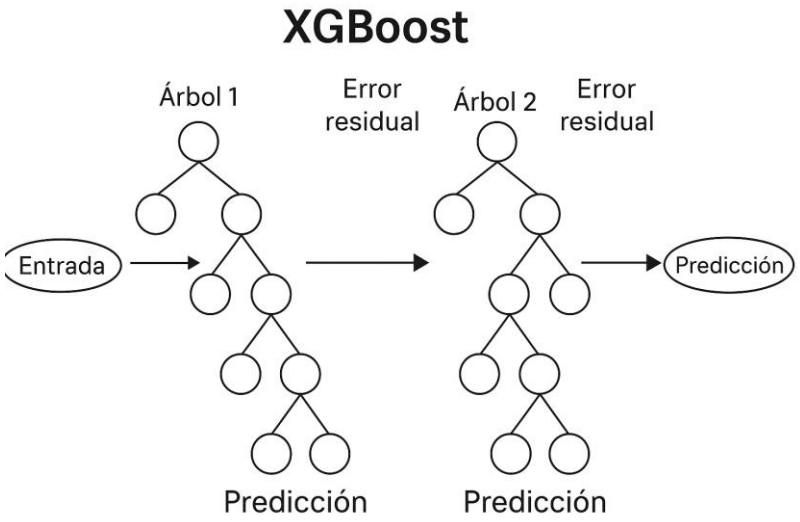

Figura 9 Algoritmo XGBoost

Este modelo esta alimentado con representaciones vectorizadas mediante BERT (utilizando 2.442 muestras) y un modelo de clasificación directa con BERT (entrenado con 8.100 conversaciones derivadas del despliegue del modelo de transferencia de conocimiento).

Entrenamiento XGBoost

Para optimizar el proceso de entrenamiento del modelo de transferencia de conocimiento, se implementó el algoritmo XGBoost con soporte para aceleración mediante GPU. Esta decisión se fundamentó en la necesidad de reducir los tiempos de cómputo, debido al tamaño y complejidad del conjunto de datos vectorizado a partir de representaciones densas de texto (embeddings BERT). El uso de GPU permite procesar grandes volúmenes de datos en paralelo, acelerando la construcción de los árboles de decisión y haciendo viable la exploración exhaustiva del espacio de hiperparámetros.

El modelo fue ajustado a través de una búsqueda exhaustiva de hiperparámetros, considerando una combinación estratégica de valores para las variables que más inciden en el rendimiento del algoritmo. Entre los principales hiperparámetros evaluados se encuentran:

- **Profundidad máxima de los árboles (max_depth):** controla la complejidad del modelo.
- **Tasa de aprendizaje (learning_rate):** regula la contribución de cada árbol nuevo.
- **Número de árboles (n_estimators):** determina cuántos árboles componen el modelo.
- **Proporción de muestras por árbol (subsample):** ayuda a evitar sobreajuste.
- **Proporción de variables por árbol (colsample_bytree):** controla la diversidad de los árboles.
- **Parámetros de regularización L1 y L2 (reg_alpha, reg_lambda):** evitan el sobreajuste penalizando modelos complejos.

Para encontrar la combinación óptima de estos parámetros, se utilizó un proceso de búsqueda en malla (grid search) combinado con validación cruzada. Este enfoque permitió evaluar múltiples configuraciones en particiones distintas del conjunto de datos, garantizando así la robustez del modelo y reduciendo la posibilidad de sobreajuste.

3.4.2.2 Validación del modelo y métricas de evaluación

La validación del modelo se realizó mediante una partición estratificada del conjunto de datos, dividiendo la muestra en un **80% para entrenamiento y 20% para validación**, asegurando la **preservación proporcional de las clases** (BUENA y MALA) en ambas particiones. Esta estrategia garantizó una evaluación confiable de la capacidad del modelo para generalizar más allá de los datos vistos durante el entrenamiento.

Las métricas utilizadas para evaluar el desempeño del modelo fueron:

- **Precisión (Precision)**: Proporción de predicciones positivas correctas entre todas las predicciones positivas realizadas. $Precision = \frac{\text{TP}}{TP+FP}$
- **Exhaustividad (Recall)**: Proporción de verdaderos positivos correctamente identificados entre todos los casos reales positivos. $Recall = \frac{\text{TP}}{\text{TP+FN}}$
- **F1-score**: Media armónica entre precisión y recall, especialmente útil cuando existe un desequilibrio entre clases. $F1Score = 2\ \frac{\text{Precision x Recall}}{\text{Precision + Recall}}$
- **Exactitud (Accuracy)**: Porcentaje de clasificaciones correctas sobre el total de predicciones realizadas. $Accuracy = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$

Adicionalmente, se utilizó la **matriz de confusión**, que permite observar visualmente el rendimiento del modelo en cada clase, identificando los aciertos y errores de clasificación.

Los resultados obtenidos por el modelo XGBoost en esta fase de validación fueron los siguientes:

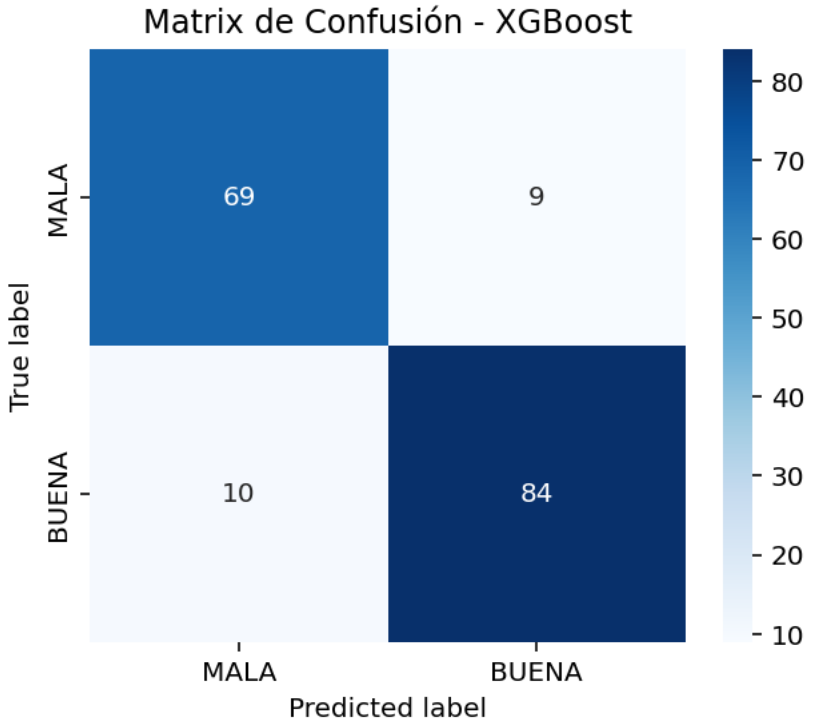

Figura 10 Matrix de confusión Modelo de transferencia de conocimiento XGBoost

Clase	Precisión	Recall	F1-Score	Soporte
MALA	0.873	0.885	0.879	78
BUENA	0.903	0.894	0.898	94
Accuracy			0.889	172
Prom. macro	0.888	0.890	0.889	
Prom. ponderado	0.890	0.890	0.889	

Tabla 2 Métricas de clasificación modelo de transferencia de conocimiento con XGBoost

3.4.2.3 Despliegue del modelo

Una vez validado el rendimiento del modelo, se procedió con su despliegue en un entorno controlado, con el propósito de evaluar su desempeño en condiciones más cercanas a un escenario productivo. El modelo fue implementado sobre un conjunto de datos nuevo, compuesto por 8.100 conversaciones no utilizadas durante las fases de entrenamiento ni validación.

Este despliegue tuvo dos objetivos principales:

1. Clasificación automatizada de nuevas conversaciones, aplicando los conocimientos adquiridos mediante la transferencia desde el conjunto etiquetado manualmente.
2. Generación de insumos para el entrenamiento posterior de modelos más complejos, como el clasificador directo basado en BERT, utilizando las etiquetas producidas por el modelo de transferencia.

Para ello, se diseñó un flujo de procesamiento compuesto por las siguientes etapas:

- Carga y preprocesamiento de los datos: Las conversaciones fueron estructuradas en formato JSON y transformadas en texto plano siguiendo la misma lógica utilizada durante el entrenamiento.
- Tokenización y vectorización: Se reutilizó el esquema de tokenización y extracción del vector [CLS] del modelo BERT para representar las nuevas conversaciones.
- Clasificación: El modelo XGBoost previamente entrenado se utilizó para predecir la categoría (BUENA o MALA) de cada conversación.

- Almacenamiento estructurado: Las predicciones fueron almacenadas junto con los textos originales en una base de datos estructurada, permitiendo su uso posterior para análisis, visualización y entrenamiento adicional.

Este proceso permitió consolidar una nueva base de datos etiquetada de manera automática, que fue utilizada para entrenar el modelo BERT de clasificación directa, el cual se beneficia de la escala del nuevo conjunto, manteniendo la lógica de etiquetado original aprendida del experto.

El despliegue también permitió identificar patrones emergentes en las conversaciones, validar la estabilidad del modelo en escenarios no vistos, y establecer una infraestructura básica para su futura incorporación en el flujo operativo de evaluación de calidad de atención

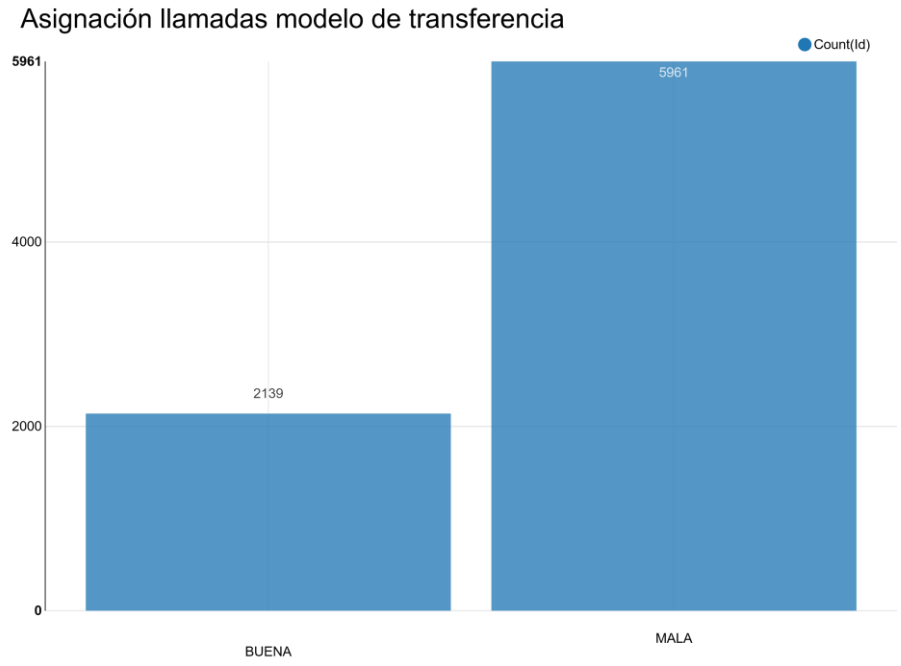

Figura 11 Resultados distribución por calificación de los 8100 registros

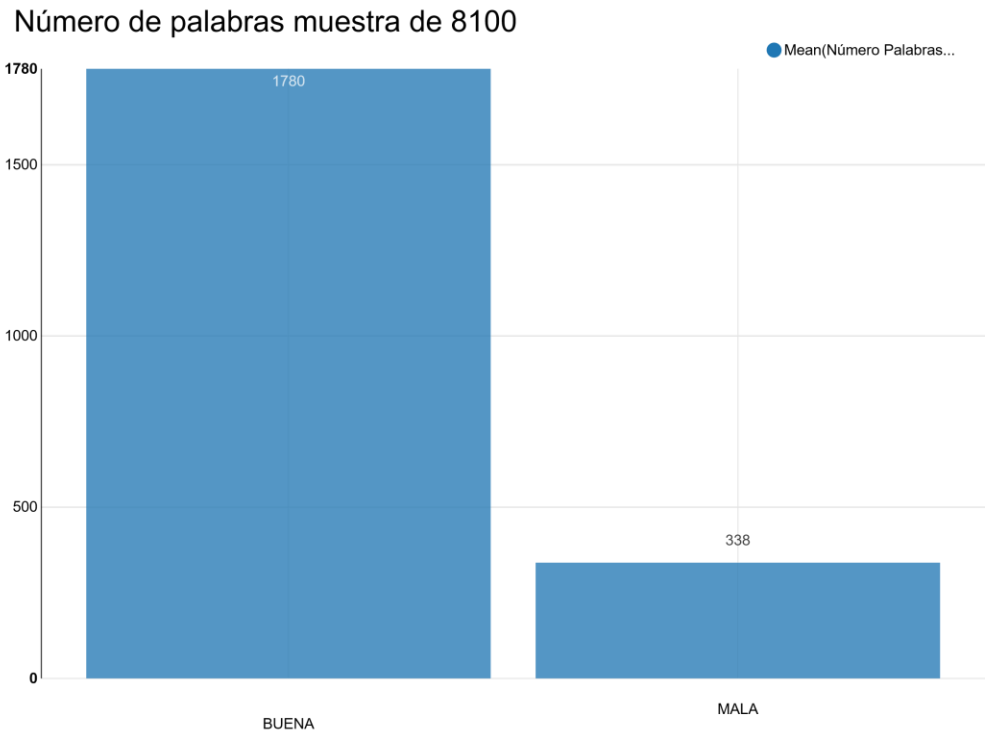

Figura 12 Promedio de palabras por calificación transferencia de 8.100 registros

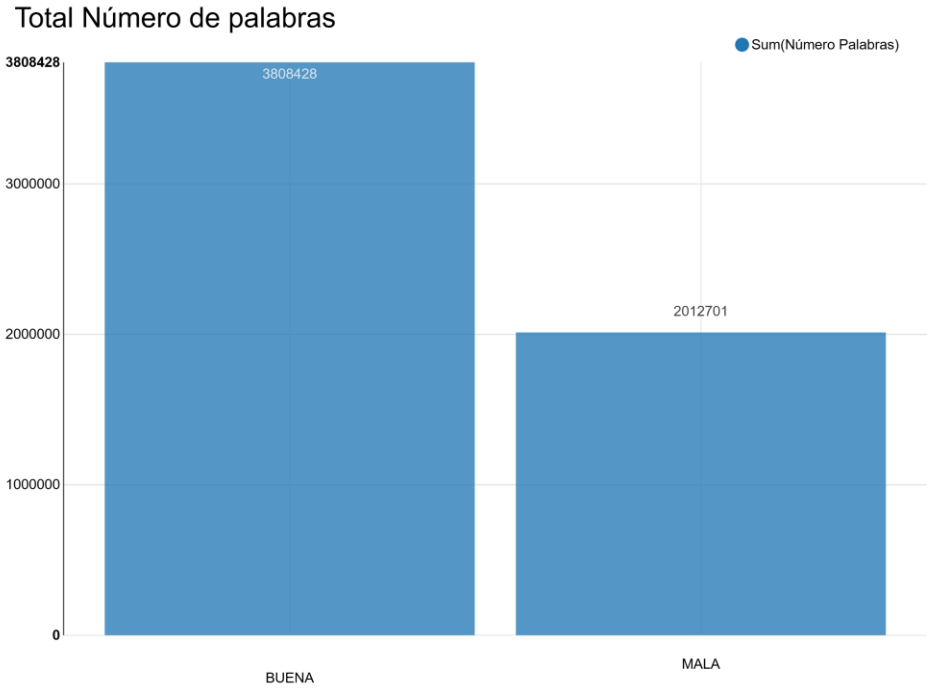

Figura 13 Número de palabras totales por calificación transferencia de 8.100 registros

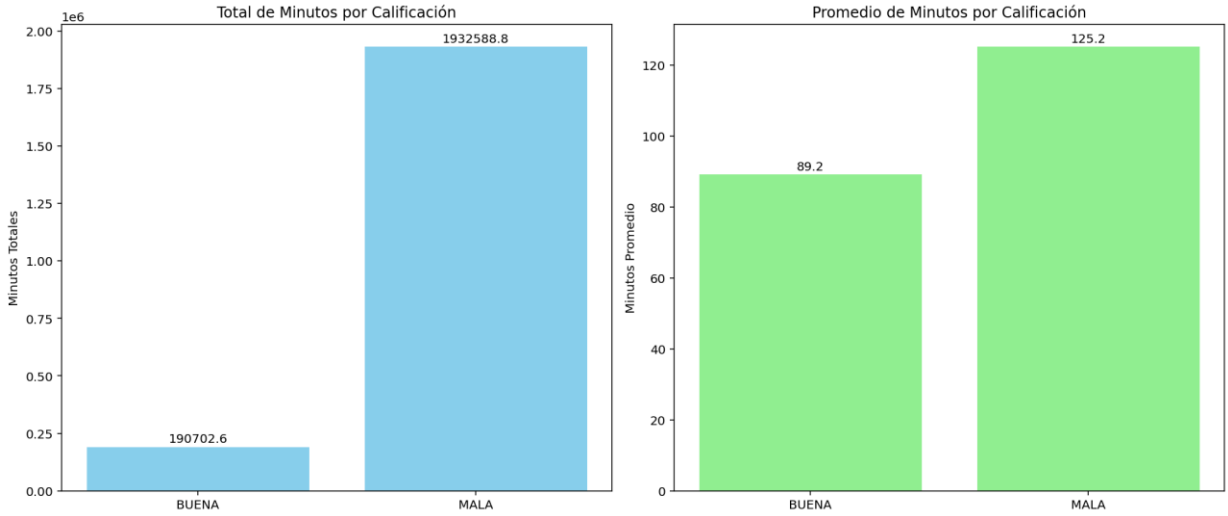

Figura 14 Totales y promedio de minutos por calificación 8.100 registros

3.4.3 Modelo BERT para la clasificación de calificaciones

3.4.3.1 Preprocesamiento y tokenización de conversaciones

El preprocesamiento inició con la lectura de 8.100 registros generados por un modelo previo de transferencia de conocimiento, los cuales contenían conversaciones en formato JSON. Cada conversación fue desestructurada y transformada en una única cadena de texto, concatenando todos los turnos de diálogo. Las calificaciones textuales ("BUENA" y "MALA") fueron mapeadas a valores binarios (1 y 0, respectivamente), y se descartaron aquellos registros con etiquetas inválidas. Este paso es crucial, ya que BERT no opera directamente sobre texto plano, sino sobre tokens numéricos, y requiere etiquetas numéricas en contextos supervisados.

La tokenización se realizó utilizando el tokenizador **BertTokenizer** asociado al modelo seleccionado. Este tokenizador aplica la técnica de WordPiece, que fragmenta palabras en subunidades semánticas y las convierte en identificadores enteros. También incorpora automáticamente los tokens especiales [CLS] y [SEP], y asegura una longitud uniforme de las secuencias mediante truncamiento y relleno (padding).

Para facilitar el entrenamiento, se definió una clase personalizada ConversationDataset compatible con PyTorch. Esta clase encapsula los textos tokenizados junto con sus etiquetas, permitiendo su uso directo dentro del objeto Trainer de la biblioteca HuggingFace.

3.4.3.2 Configuración de entrenamiento

El modelo utilizado fue **BertForSequenceClassification**, una variante de BERT diseñada específicamente para tareas de clasificación. Este modelo incluye:

- Capas de embeddings que combinan información del token, la posición y el segmento.
- Doce capas de codificación (encoders) basadas en atención multi-cabeza (multi-head self-attention), cada una con mecanismos de normalización y redes feed-forward.
- Una capa de clasificación aplicada sobre la representación del token [CLS], que genera la predicción final mediante una capa lineal seguida de una función softmax.

El entrenamiento se realizó utilizando el objeto Trainer, configurado con los siguientes hiperparámetros:

- num_train_epochs = 20: número de épocas de entrenamiento.
- per_device_train_batch_size = 8: tamaño del lote por dispositivo.
- weight_decay = 0.01: regularización L2 para prevenir sobreajuste.
- evaluation_strategy = "epoch": evaluación del modelo al final de cada época.

La optimización se llevó a cabo mediante el algoritmo **AdamW**, una variante del optimizador Adam que incorpora decaimiento de peso, lo cual resulta especialmente efectivo en modelos basados en Transformers.

3.4.3.3 Validación del modelo y métricas de evaluación

Para evaluar el rendimiento del modelo de procesamiento de lenguaje natural (PLN) desarrollado, se emplearon métricas estándar ampliamente aceptadas en la literatura:

Exactitud (accuracy): $Accuracy = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$

Precisión $Precision = \frac{\mathrm{TP}}{TP+FP}$

Recall $Recall = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$

F1-score. $F1Score = 2\ \frac{\text{Precision x Recall}}{\text{Precision + Recall}}$

Donde

TP: Verdaderos positivos

TN: Verdaderos negativos

FP: Falsos positivos

FN: Falsos negativos

Además de las métricas tradicionales, se empleó el coeficiente **Kappa de Cohen** para evaluar el nivel de concordancia entre las etiquetas reales y las predichas por el modelo, controlando el efecto del acuerdo que podría producirse por azar. Esta métrica resulta especialmente útil en contextos de clasificación multiclase y desequilibrio entre clases, donde la exactitud por sí sola puede ofrecer una visión parcial del rendimiento Porque si clasificar fuera tan fácil, incluso al azar se lograría lo mismo.

Estas métricas permiten cuantificar el nivel de acierto del modelo, no solo a nivel general, sino también por clase, considerando la distribución de etiquetas reales y predichas. A continuación, se presentan los resultados derivados del análisis de la matriz de confusión correspondiente a la fase de entrenamiento del modelo.

6.1 Rendimiento del modelo fase de entrenamiento

Para evaluar el rendimiento del modelo **BERT** en la tarea de clasificación de conversaciones como **BUENAS** o **MALAS**, se construyó una **matriz de confusión**, que permite visualizar el número de aciertos y errores del modelo en cada categoría. El conjunto de prueba estuvo conformado por **1.605 conversaciones etiquetadas manualmente**, y el modelo logró clasificar correctamente **1.502** de ellas, lo cual se traduce en una **exactitud (accuracy)** del **93,58%**.

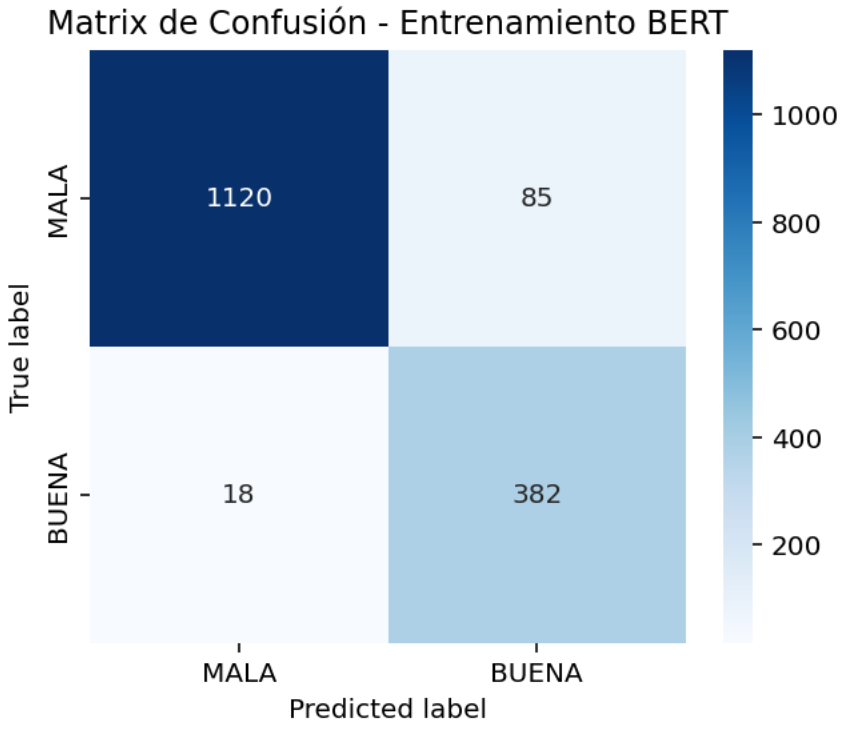

Figura 15 Matrix de confusión entramiento Modelo BERT 8100 registros 20% test

Con el fin de evaluar el rendimiento del modelo de clasificación basado en BERT, se construyó una matriz de confusión que resume los aciertos y errores cometidos por el modelo al momento de clasificar las conversaciones en dos clases: MALA y BUENA. Esta evaluación permite identificar el comportamiento del modelo en función de su capacidad para distinguir entre ambas categorías.

Clase	**Precisión**	**Recall**	**F1-Score**	**Soporte**
MALA	0.984	0.930	0.956	1205
BUENA	0.818	0.955	0.882	400
Accuracy			0.938	1605

Clase	Precisión	Recall	F1-Score	Soporte
Prom. macro	0.901	0.942	0.919	
Prom. ponderado	0.943	0.936	0.937	

Tabla 3 Métricas de clasificación modelo BERT fase de entrenamiento

De un total de **1.605 conversaciones evaluadas**, el modelo clasificó correctamente **1.502**, lo que se traduce en una **exactitud (accuracy) global del 93,58%**. Este valor indica que el modelo tiene una alta capacidad para identificar correctamente la clase correspondiente en la mayoría de los casos.

A partir de los valores individuales de la matriz de confusión, se calcularon las métricas estándar por clase:

Clase MALA:

- Precisión: 98,42%
  De todas las veces que el modelo predijo que una conversación era MALA, acertó el 98,42% de las veces.
- Recall: 92,94%
  De todas las conversaciones que realmente eran MALA, el modelo identificó correctamente el 92,94%.
- F1-score: 95,58%
  Un excelente equilibrio entre precisión y recall para esta clase.

Clase BUENA:

- Precisión: 81,78%
  De todas las veces que el modelo dijo que una conversación era BUENA, acertó el 81,78%.
- Recall: 95,50%
  El modelo detectó correctamente el 95,50% de todas las conversaciones que realmente eran BUENAS.
- F1-score: 88,16%
  Buen desempeño general, aunque con mayor tendencia a equivocarse al predecir BUENA.

El modelo BERT demuestra un comportamiento sólido y balanceado en la tarea de clasificación. La **alta precisión en la clase MALA** indica que rara vez comete errores al señalar conversaciones negativas, lo cual es crucial en contextos donde detectar problemas a tiempo es prioritario. Por otro lado, el **alto recall en la clase BUENA** muestra que el modelo identifica correctamente la gran mayoría de interacciones positivas, aunque con una tasa de error algo mayor al predecir esa categoría.

El **F1-score general** superior al 90% refuerza la conclusión de que el modelo no solo acierta con alta frecuencia, sino que lo hace de forma coherente entre las dos clases. Estos resultados justifican la elección de BERT como una arquitectura adecuada para tareas de análisis automatizado de calidad conversacional.

3.4.3.4 Rendimiento del modelo BERT despliegue

En esta etapa, se utilizaron **524 conversaciones etiquetadas manualmente por un experto** para evaluar el desempeño del modelo en un escenario de despliegue real, donde se enfrenta a datos completamente nuevos, sobre los cuales **no ha recibido ningún tipo de entrenamiento previo**. Este procedimiento tiene como objetivo medir la **capacidad de generalización del modelo**, es decir, su habilidad para realizar predicciones precisas sobre información que no ha sido vista anteriormente.

Además, esta evaluación permite verificar que **no se haya producido sobreajuste (overfitting)** en ninguna de las fases de desarrollo y entrenamiento del modelo, asegurando que el rendimiento observado no se limita únicamente al conjunto de datos de entrenamiento, sino que se extiende de manera robusta a nuevos contextos operativos.

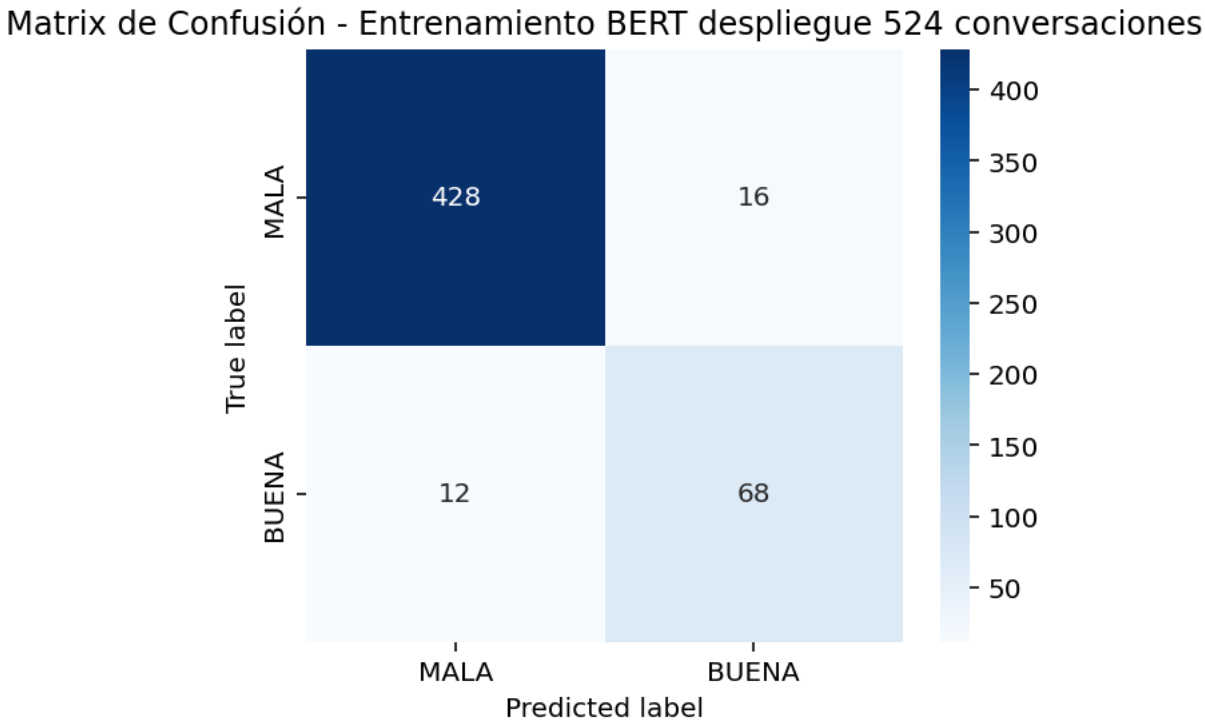

Figura 16 Matrix de confusión despliegue modelo Bert 524 conversaciones

Clase	**Precisión**	**Recall**	**F1-Score**	**Soporte**
MALA	0.973	0.964	0.968	444
BUENA	0.810	0.850	0.829	80
Accuracy			0.945	524
Prom. macro	0.891	0.907	0.899	
Prom. ponderado	0.943	0.945	0.938	

Tabla 4 Métricas de clasificación modelo BERT fase de despliegue

El modelo alcanza un **94.66% de precisión** global y sólo un **5.34% de error**, lo que, a primera vista, sugiere un rendimiento excelente. La Kappa de 0.798 (sustancial) respalda que gran parte de ese acierto se debe a la capacidad real del modelo más que a un acuerdo por azar.

No obstante, la matriz muestra que el conjunto de datos está desequilibrado: hay 444 llamadas MALA frente a sólo 80 BUENA. En consecuencia, el modelo clasifica casi correctamente todas las llamadas MALA (428/444 ≈ **96.4%** correctas) y, aunque también detecta bien las BUENA, su *recall* en esa clase es menor (68/80 = **85%**). Es decir, el modelo falla en el **15%** de los casos BUENA (12 Falsos Negativos) frente a sólo un **3.6%** de falsos positivos en MALA.

En términos prácticos, esto significa que el modelo identifica con alta probabilidad las llamadas malas, pero aún pasa por alto un número moderado de buenas. En el contexto de negocio clasificar erróneamente una llamada buena como mala tiene alto costo, este sesgo debe considerarse. Por otro lado, la precisión global supera holgadamente el nivel que obtendría un clasificador trivial que predijera siempre la clase mayoritaria (por ejemplo, siempre *MALA* daría ≈84.7% de acierto), lo que indica que el modelo añade valor al reconocer correctamente la mayoría de las instancias BUENA.

En resumen, el modelo funciona muy bien en general: identifica correctamente la gran mayoría de instancias (94.66% de acierto) y presenta un acuerdo sólido con la realidad ($\kappa \approx 0.80$). Sin embargo, la comparación con escenarios de referencia trivial muestra que parte de esta alta precisión se debe al sesgo hacia la clase mayoritaria. Además, **una precisión elevada no garantiza por sí sola un desempeño excelente** si no se consideran los tipos de error; por ejemplo, un 99% de precisión podría ocultar falsos negativos críticos en un contexto sensible. En este caso concreto, conviene examinar la tasa de falsos negativos (15% en BUENA) en función de los objetivos y costos de clasificación.

3.5 CONSIDERACIONES ÉTICAS Y LEGALES

3.5.1 Privacidad de los datos

En concordancia con las normativas vigentes en materia de protección de datos personales, y en particular con lo dispuesto en la Ley 1581 de 2012 de Habeas Data en Colombia, el presente proyecto adoptó medidas estrictas para garantizar la privacidad de la información sensible de los clientes. Durante todo el ciclo de desarrollo del modelo, se implementó un proceso riguroso de anonimización de los datos, con el objetivo de evitar cualquier posibilidad de identificación directa o indirecta de los individuos involucrados en las conversaciones analizadas.

Para ello, se eliminaron o transformaron de forma irreversible todos los elementos de identificación personal presentes en los textos, tales como: número de documento, dirección de residencia, número de teléfono, números de tarjetas, número de productos financieros asociados y cualquier otra información que pudiera comprometer la privacidad del cliente. A cada registro conversacional se le asignó un token único anonimizado, que actúa como identificador irrepetible sin referencia directa al cliente real.

3.5.2 Manejo de datos sensibles

De igual forma, se establecieron procedimientos diferenciados para el tratamiento de datos considerados sensibles según los estándares de privacidad y ética en la ciencia de datos. Esto incluyó no solo la anonimización técnica, sino también la definición de políticas de acceso restringido a los datos, trazabilidad en las manipulaciones realizadas, y control de versiones durante todo el proceso de extracción, transformación y análisis.

La muestra utilizada para el etiquetado manual fue gestionada bajo protocolos de confidencialidad firmados por los participantes, y el personal involucrado en el proyecto recibió formación sobre los principios de manejo responsable de la información.

3.5.3 Resguardo de la información

Un principio rector en el desarrollo del proyecto fue la seguridad de la información. En este sentido, ninguno de los datos utilizados fue almacenado, procesado ni expuesto en servicios de nube pública o externa. Toda la operación se llevó a cabo en servidores locales autorizados por la entidad bancaria, que cumplen con los estándares institucionales de seguridad de la información y control de acceso.

Esto garantizó que no existiera riesgo de filtración, fuga o exposición de datos a terceros no autorizados. Asimismo, se deshabilitó cualquier mecanismo de sincronización automatizada o respaldo externo en plataformas que pudieran comprometer la confidencialidad de los datos procesados.

4 CONCLUSIONES

El presente proyecto de grado logró satisfacer de manera integral los objetivos planteados, demostrando la viabilidad y eficacia de aplicar técnicas avanzadas de procesamiento de lenguaje natural (PLN) para analizar conversaciones de WhatsApp en el ámbito de la gestión de cobranzas bancarias. A través de una metodología rigurosa, se desarrolló un flujo automatizado que combinó herramientas como Selenium y Python para la recolección de datos, junto con un sistema de almacenamiento estructurado en SQL Server, garantizando la trazabilidad y calidad de los datos desde su origen.

En la fase de preprocesamiento, se implementaron técnicas de limpieza, normalización y vectorización, donde el uso de embeddings BERT superó significativamente a enfoques tradicionales como TF-IDF, al capturar de manera más efectiva el contexto semántico y las relaciones entre palabras. Este avance fue crucial para abordar la complejidad de las conversaciones, que incluían jerga financiera y expresiones coloquiales. Además, se resolvió el desafío del desequilibrio de clases mediante técnicas como SMOTE, optimizando métricas clave como el F1-score y el recall, lo que permitió un aprendizaje equilibrado y robusto.

La arquitectura del modelo combinó dos enfoques complementarios:

1. XGBoost para la transferencia de conocimiento, destacándose por su eficiencia computacional y rendimiento en la clasificación inicial.
2. BERT para la clasificación directa, aprovechando su capacidad de atención bidireccional para analizar patrones conversacionales complejos.

La implementación de GPU aceleró los procesos de entrenamiento y despliegue, reduciendo tiempos de cómputo y facilitando la escalabilidad del sistema. Los resultados fueron contundentes: el modelo alcanzó una precisión del 94.66% en el despliegue real, con un kappa de Cohen de 0.798, evidenciando una concordancia sustancial con las evaluaciones humanas. Estas métricas validaron no solo la capacidad predictiva del modelo, sino también su utilidad práctica para identificar oportunidades de mejora en la gestión de cobranzas.

Durante la implementación de este proyecto, se identificaron varias limitaciones y aprendizajes clave:

1. Dependencia inicial de etiquetado manual

   La necesidad de un conjunto de datos etiquetado de calidad puso de manifiesto la importancia de contar con expertos en la fase de entrenamiento. Si bien el etiquetado manual asegura datos representativos y relevantes, también es costoso y propenso a sesgos humanos sin embargo un numero alto de datos etiquetados en los modelos suele ser bastante representativo en el rendimiento del mismo. Para mitigar esta limitación, se consideró la posibilidad de emplear técnicas de etiquetado semi-supervisado o active learning en iteraciones futuras.

2. Costo computacional de modelos avanzados como BERT

   Modelos basados en transformers, como BERT, presentan una demanda significativa de recursos computacionales debido a su gran número de parámetros y su arquitectura autoatencional. Esto requirió el uso de hardware especializado (GPUs o TPUs) y estrategias de optimización, como el fine-tuning controlado y la adopción de versiones más ligeras (por ejemplo, DistilBERT o ALBERT) para tareas específicas donde la precisión podía sacrificarse en favor de eficiencia.

3. Vectorización y Representación de Texto

   Una lección importante fue la relevancia de la vectorización para capturar las relaciones semánticas y contextuales en el lenguaje. Técnicas previas, como TF-IDF o Word2Vec, no lograban representar el contexto de las palabras en una oración completa, mientras que modelos como BERT introducen embeddings contextualizados, fundamentales para tareas complejas como clasificación o detección de intenciones.

4. Por qué se eligió BERT sobre otros modelos

   Aunque existen diversos modelos para procesamiento de lenguaje natural (por ejemplo, LSTM, CNN, fastText, Word2Vec), BERT demostró superioridad en tareas que requieren comprensión profunda del contexto. Su arquitectura basada en autoatención bidireccional permite capturar dependencias a largo plazo, superando a modelos que solo consideran el contexto unidireccional o embeddings estáticos. Adicionalmente, se exploraron variantes como LSTM pero BERT mostró el mejor equilibrio entre rendimiento, disponibilidad de modelos preentrenados y facilidad de integración en nuestro flujo de trabajo.

   En síntesis, este trabajo demuestra cómo el análisis automatizado de conversaciones puede cambiar profundamente la forma en que se gestionan los equipos de asesores en el sector financiero. Ahora es posible identificar rápidamente grupos de conversaciones similares, lo que permite evaluar de forma más justa y precisa el desempeño de los asesores, y detectar oportunidades de capacitación y mejora continua.

   Además, esta tecnología abre la puerta a un análisis más estratégico de las interacciones con los clientes, facilitando la identificación de temas recurrentes, necesidades no atendidas y patrones de comportamiento. A futuro, su aplicación podrá ayudar a optimizar procesos clave, como la atención al cliente, la recuperación de cartera y nuevos mecanismos para ayudar a los clientes.

   En definitiva, este proyecto no solo introduce un avance técnico, sino que ofrece una herramienta práctica y escalable que puede transformar la calidad de la atención y la toma de decisiones dentro de la banca.

5 TRABAJOS FUTUROS

A partir de los resultados obtenidos en este proyecto, se abren diversas oportunidades de profundización y mejora, tanto desde la perspectiva metodológica como tecnológica. Las siguientes líneas de investigación se proponen como una evolución natural de este trabajo, con el fin de ampliar su alcance, robustecer su aplicabilidad y alinearse con los avances más recientes en ciencia de datos y procesamiento de lenguaje natural:

1. Expansión del esquema de clasificación a múltiples categorías
   Actualmente, el modelo clasifica las conversaciones en dos clases: BUENA y MALA. Una línea de trabajo inmediata consiste en diseñar un esquema de clasificación más granular, incorporando nuevas categorías que reflejen mejor la diversidad de interacciones, tales como: *neutral*, *potencial de pago*, *cliente en desacuerdo*, *ofrecimiento de alternativas*, entre otras. Esto permitiría no solo evaluar la calidad general de la conversación, sino también caracterizar con mayor precisión las dinámicas de las interacciones y sus implicaciones en la gestión de cobranzas.

2. Aplicación de modelos de lenguaje de última generación (LLMs)
   El uso de modelos como GPT-4, LLaMA 3, Mistral o Claude abre nuevas posibilidades para realizar análisis conversacional más profundo y contextual. Estas arquitecturas permiten tareas avanzadas como resumen de conversaciones, identificación de intenciones múltiples, análisis contextualizado de emociones y detección de conflictos. Incorporar estos modelos, ya sea mediante APIs comerciales o mediante fine-tuning en infraestructura propia, podría mejorar significativamente el rendimiento y la comprensión semántica de las conversaciones.

3. Análisis multilingüe y reconocimiento de variaciones lingüísticas
   En un contexto real donde los clientes pueden comunicarse en diferentes dialectos, jergas o incluso idiomas, una línea de investigación importante sería la incorporación de capacidades multilingües y el reconocimiento de fenómenos lingüísticos propios del español colombiano. Modelos como XLM-RoBERTa o mBERT podrían ser útiles en este propósito.

4. Integración con tecnologías cloud y despliegue escalable
   Un siguiente paso clave es la integración del sistema con arquitecturas en la nube como AWS SageMaker, Google Vertex AI, Azure Machine Learning o Hugging Face Inference Endpoints, lo que permitiría escalar la solución, reducir tiempos de inferencia y facilitar el mantenimiento del sistema. Además, se podría habilitar el monitoreo en tiempo real del rendimiento del modelo y establecer pipelines de actualización continua (MLOps), manteniendo el sistema actualizado frente a cambios en los datos conversacionales.

5. Incorporación de feedback humano en bucle (Human-in-the-loop)
   Para mejorar continuamente el modelo y corregir posibles sesgos, se propone incluir un sistema de retroalimentación por parte de supervisores humanos, que permita refinar las predicciones y ajustar el modelo dinámicamente. Este enfoque híbrido es

especialmente valioso en entornos sensibles como la gestión de cobranzas, donde errores de clasificación pueden tener implicaciones reputacionales o legales.

6. Análisis longitudinal del comportamiento del cliente
   Otra línea de investigación relevante es la exploración del comportamiento conversacional del cliente a lo largo del tiempo. Mediante técnicas de seguimiento secuencial o modelado temporal (por ejemplo, usando redes neuronales temporales o Transformers de series temporales), sería posible predecir la probabilidad de pago o el riesgo de deserción con base en el historial de interacciones del cliente.

7. Desarrollo de tableros interactivos con capacidades explicativas (Explainable AI)
   Implementar dashboards interactivos que integren visualizaciones de métricas, ejemplos representativos de conversaciones clasificadas y explicaciones sobre las decisiones del modelo (utilizando herramientas como LIME o SHAP), permitiría a los usuarios no técnicos entender mejor el funcionamiento del sistema y aumentar su confianza en los resultados.

6 REFERENCIAS BIBLIOGRÁFICAS

[1]https://www.superfinanciera.gov.co/publicaciones/60950/informes-y-cifrascifrasestablecimientos-de-creditoinformacion-periodicamensualevolucion-cartera-de-creditos-60950/. [Accedido: 17-Jun-2024].

[2] Statista, " Number of unique WhatsApp mobile users worldwide from January 2020 to March 2024" 2024. [En línea]. Disponible: https://www.statista.com/statistics/1306022/whatsapp-global-unique-users/. [Accedido: 17-Jun-2024].

[3] J. Schmidhuber, “Deep learning in neural networks: An overview” 8-Oct-2014. [En Línea]. Disponible en https://arxiv.org/pdf/1404.7828

[4] S. Hochreiter y J. Schmidhuber, "Long Short-term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997. [En línea]. Disponible: https://www.researchgate.net/publication/13853244_Long_Short-term_Memory.

[5] José Luis Sarmiento-Ramos, " Applications of neural networks and deep learning to biomedical engineering " 2020. [En línea]. Disponible: https://www.redalyc.org/journal/5537/553768213002/. [Accedido: 17-Jun-2024].

[6] V. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems. Disponible en https://arxiv.org/pdf/1706.03762

[7] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018. Disponible en: https://arxiv.org/abs/1810.04805

[8] C. P. Chai, "Comparison of text preprocessing methods," Natural Language Engineering, vol. 29, (3), pp. 509-553, 2023/05//. Disponible: https://www.proquest.com/scholarly-journals/comparison-text-preprocessing-methods/docview/2815043087/se-2. DOI: https://doi.org/10.1017/S1351324922000213.

[9] Barrett Studdard “Preprocessing Text Data for Machine Learning” 2021 Disponible en: https://datastud.dev/posts/nlp-preprocess

[10] J. Smith, R. Kumar, and A. Patel, "Automated Customer Service Chatbot for Banking Sector," in Proceedings of the IEEE Conference on Artificial Intelligence and Machine Learning, 2020, pp. 789-794.

[11] H. Liu, Z. Wang, and L. Chen, "Natural Language Processing Techniques for Analyzing Financial Sentiment in Social Media," Journal of Financial Technology, vol. 15, no. 2, pp. 134-145, 2021.

[12] P. Johnson and S. Lee, "Improving Debt Collection Efficiency Using Machine Learning and Natural Language Processing," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 1, pp. 3456-3463, 2019.

[13] Deep Talk "Historia y actualidad del procesamiento de lenguaje natural," Deep Talk Blog, 29-Nov-2021. Disponible: https://blog.deep-talk.ai/historia-y-actualidad-del-procesamiento-de-lenguaje-natural-8de41a357ca9.

[14] Néstor Camilo Beltrán, Edda Camila Rodríguez (2021), Procesamiento del lenguaje natural (PLN) - GPT-3, y su aplicación en la Ingeniería de Software. Tecnol.Investig. Academia TIA, ISSN: 2344- 8288, 8 (1), pp. 18-37. Bogotá-Colombia Disponible: https://revistas.udistrital.edu.co/index.php/tia/article/download/17323/17210/104548

[15] P. Yalla y N. Sharma, «Integrating Natural Language Processing and Software Engineering,» International Journal of Software Engineering and Its Applications, vol. 9, nº 11, pp. 127-136, 2015. [11]D. J. Matich, «Redes Neuronales: Conceptos Básicos y Aplicaciones.,» 03 03 2001. [En línea]

[16] Alex Graves, " Generating Sequences With Recurrent Neural Networks " arXiv preprint arXiv:1308.0850v5.

[17] Cambria, E., Poria, S., Gelbukh, A., & Thelwall, M. (2017). Sentiment analysis is a big suitcase. IEEE Intelligent Systems, 32(6), 74-80.

[18] Xu, P., Liu, Q., Qiu, X., & Huang, X. (2016). A novel word embedding model for keyword extraction. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 2136-2145).

[19] Espinosa - Zúñiga, J. J. (2020). Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito. Ingeniería, investigación y tecnología, 21(3), 00002. Epub 02 de diciembre de 2020.https://doi.org/10.22201/fi.25940732e.2020.21.3.022

[20]Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

[21] Otzen, T., & Manterola, C. (2017). Técnicas de muestreo sobre una población a estudio. International Journal of Morphology, 35(1), 227–232. https://doi.org/10.4067/S0717-95022017000100227

[22] Jurafsky, D., & Martin, J. H. (2025). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition with language models (3rd ed.). Stanford University. https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf