UNIVERSIDAD CATÓLICA BOLIVIANA "SAN PABLO"
UNIDAD ACADÉMICA REGIONAL LA PAZ
DEPARTAMENTO DE POSGRADO
MAESTRÍA EN CIENCIA DE DATOS

PROYECTO 2 - CASO DE ESTUDIO

"Clasificación Automática de Tickets con NLP"

MÓDULO de Machine Learning

Docente: Jheser Guzman Ph.D.

Maestrantes: Virginia Mercedes Fernández Daza
Marco Antonio Velásquez Rocha
Ivan Israel Machicado Quiroga

noviembre, 2024

Índice general

Índice de tablas

Índice de figuras

1. Antecedentes.

Las quejas de los clientes en el sector financiero son un aspecto crucial para la mejora continua de los productos y servicios ofrecidos. Estas quejas no solo reflejan insatisfacciones, sino que también brindan valiosa información sobre posibles áreas de mejora. Según Kumar et al. (2018), la correcta gestión de las quejas no solo resuelve problemas inmediatos, sino que también permite construir una relación más sólida con los clientes, lo que lleva a una mayor lealtad y satisfacción. Resolver quejas de manera eficiente es clave para mantener la competitividad en un mercado tan dinámico como el financiero, donde los clientes buscan una atención al cliente rápida y eficaz.

En el contexto de la empresa financiera, las quejas a menudo se presentan en forma de datos no estructurados en tickets de atención. Estos tickets incluyen una variedad de problemas que los clientes experimentan con los productos, como tarjetas de crédito, servicios bancarios y préstamos. Según Zhang et al. (2017), la presencia de datos textuales no estructurados puede representar un desafío para las empresas, ya que su análisis manual es intensivo en recursos y tiempo, lo que retrasa la capacidad de respuesta. Esto genera una sobrecarga de trabajo para el personal de atención, lo que puede afectar negativamente la eficiencia y la experiencia del cliente.

Con el crecimiento de las empresas y la expansión de su base de clientes, la carga de trabajo relacionada con el manejo de quejas aumenta. En este contexto, el uso de tecnología para automatizar el proceso de clasificación de quejas es fundamental. Pérez et al. (2019) destacan que la implementación de sistemas automatizados que pueden categorizar y priorizar tickets de queja permite a las empresas no solo reducir la carga de trabajo manual, sino también mejorar la precisión y velocidad en la resolución de problemas. La automatización, a través de técnicas como el procesamiento de lenguaje natural (PLN), puede transformar los datos no estructurados en información útil de manera mucho más eficiente que los métodos tradicionales.

Además, la automatización del análisis de quejas tiene el potencial de proporcionar información en tiempo real sobre las áreas de mejora de los productos y servicios. Según Batra et al. (2020), los sistemas automatizados de clasificación de quejas no solo permiten una respuesta más rápida, sino que también generan datos valiosos para la mejora continua. Estos sistemas pueden identificar tendencias en las quejas de los clientes, lo que ayuda a la empresa a anticipar problemas antes de que escalen. De esta manera, la implementación de soluciones basadas en inteligencia artificial y análisis de datos en la gestión de quejas se convierte en un factor clave para el éxito a largo plazo de la empresa financiera.

2. Situación Problema.

La empresa financiera tiene acumulación y el manejo manual de tickets de atención al cliente, estos tickets contienen quejas y solicitudes de los clientes, usualmente redactadas en lenguaje natural y relacionadas con diversos productos y servicios como tarjetas de crédito, banca y préstamos/hipotecas.

El proceso tradicional requiere que múltiples empleados analicen y clasifiquen manualmente cada ticket, lo que resulta en un uso intensivo de recursos humanos, tiempos prolongados de respuesta y un margen de error significativo en la categorización. A medida que la base de clientes crece, la cantidad de tickets aumenta exponencialmente, haciendo que este enfoque manual se vuelva ineficiente y afecte la capacidad de la empresa para atender rápidamente a los clientes, poniendo en riesgo su satisfacción y fidelidad.

Además, la falta de un sistema automatizado limita la capacidad de la empresa para identificar patrones y problemas recurrentes en sus productos y servicios, lo que impide la implementación de mejoras proactivas. Este enfoque reactivo no solo dificulta la resolución ágil de quejas, sino que también reduce la capacidad de la organización para obtener ventajas competitivas mediante la innovación basada en las necesidades de los clientes.

En resumen, la empresa enfrenta un desafío crítico: optimizar el manejo de tickets para garantizar la satisfacción del cliente mientras minimiza costos operativos y mejora su capacidad de respuesta.

3. Objetivos del Proyecto

3.1. Objetivo General.

Desarrollar un modelo basado en técnicas de Procesamiento de Lenguaje Natural, para clasificar de manera eficiente y precisa las quejas de los clientes en cinco categorías principales: tarjetas de crédito/tarjetas de prepago, servicios de cuentas bancarias, reporte de robo/disputa, hipotecas/préstamos y otros, de tal forma que este modelo permitirá optimizar el sistema de tickets de atención al cliente, facilitando la identificación rápida de problemas, mejorando la oferta de servicios y fortaleciendo la capacidad de respuesta de la empresa Financiera.

3.2. Objetivos Específicos.

- Cargar y preparar los datos: Realizar la carga y organización de los datos de texto provenientes de los tickets de atención al cliente, garantizando su correcta disposición para las etapas de procesamiento y análisis.
- Implementar técnicas de limpieza y normalización de texto, como la eliminación de ruido, tokenización, lematización y eliminación de palabras irrelevantes, para asegurar que los datos sean adecuados para su análisis.

- Realizar un análisis exploratorio de datos (EDA): analizar las características de los datos, como la distribución de palabras, frecuencias y patrones, para obtener una comprensión inicial de las quejas y su contenido.
- Aplicar métodos de extracción de características, como la representación TF-IDF o embeddings, para transformar los datos textuales en un formato numérico utilizable en el modelado.
- Utilizar factorización de matrices no negativas (NMF) para identificar y clasificar las quejas en cinco grupos principales relacionados con productos y servicios financieros.
- Construir modelos utilizando aprendizaje supervisado: Diseñar modelos estadísticos y de aprendizaje supervisado que complementen el modelado de temas, permitiendo una clasificación precisa de los tickets.
- Entrenar los modelos propuestos utilizando los datos disponibles, evaluando su desempeño mediante métricas relevantes como precisión, recall y f1-score.

4. Metodología

El enfoque metodológico de este proyecto es de tipo aplicado y exploratorio, basado en la clasificación automática de quejas provenientes de los tickets de atención al cliente. La investigación aplicada se centra en resolver un problema práctico, desarrollando una solución eficiente para la gestión de quejas en un contexto empresarial real (Streefkerk, 2019). En cuanto a la investigación exploratoria, esta se utiliza en las etapas iniciales para entender las características, patrones y relaciones clave en los datos, lo cual es fundamental para orientar las decisiones en las fases posteriores del modelado y la clasificación (George, 2021).

En términos de enfoque metodológico, es cuantitativo, dado que se enfoca en el análisis y procesamiento de datos numéricos y textuales. A través de la extracción de características y el entrenamiento de modelos matemáticos, se busca transformar los datos en información útil para la clasificación automática. Las métricas de desempeño, como la precisión y el F1-score, son esenciales para evaluar los resultados y asegurar que los modelos sean efectivos (Streefkerk, 2019; George, 2021).

También el enfoque utilizado tiene un ingrediente cualitativo representado por la técnica de nube de palabras.

4.1. Recopilación de datos.

El conjunto de datos para el estudio está compuesto por 78,313 registros y 22 columnas, la data esta almacenada en una base de datos en formato JSON, por lo que, para su tratamiento y análisis, necesitamos convertirlos a un formato de dataframe, utilizando para ese fin la librería REQUEST de Python.

Las columnas que contiene el conjunto de datos es el siguiente:

Tabla 1: Variables del conjunto de datos “complaints”

#	Column	Non-Null Count	Dtype
0	_index	78313 non-null	object
1	_type	78313 non-null	object
2	_id	78313 non-null	object
3	_score	78313 non-null	float64
4	_source.tags	10900 non-null	object
5	_source.zip_code	71556 non-null	object
6	_source.complaint_id	78313 non-null	object
7	_source.issue	78313 non-null	object
8	_source.date_received	78313 non-null	object
9	_source.state	76322 non-null	object
10	_source.consumer_disputed	78313 non-null	object
11	_source.product	78313 non-null	object
12	_source.company_response	78313 non-null	object
13	_source.company	78313 non-null	object
14	_source.submitted_via	78313 non-null	object
15	_source.date_sent_to_company	78313 non-null	object
16	_source.company_public_response	4 non-null	object
17	_source.sub_product	67742 non-null	object
18	_source.timely	78313 non-null	object
19	_source.complaint_what_happened	78313 non-null	object
20	_source.sub_issue	32016 non-null	object
21	_source.consumer_consent_provided	77305 non-null	object

dtypes: float64(1), object(21)

Fuente: Elaboración propia, 2024

Análisis de la calidad de los datos:

- ✓ Se han identificado 78,313 filas (entradas) en el DataFrame, numeradas del 0 al 78,312.
- ✓ El DataFrame tiene un total de 22 columnas.
- ✓ En la tabla 1 se puede observar la lista de las columnas en el DataFrame, junto con información sobre cuántos valores no nulos hay en cada columna y el tipo de dato de cada columna o **Non-Null Count**: Muestra cuántos valores no nulos hay en cada columna. Por ejemplo, en la columna _**source.tags**, solo hay 10,900 valores no nulos, lo que significa que hay muchos valores faltantes o **Dtype:** Indica el tipo de dato de cada columna. En este caso, la mayoría son de tipo object (que generalmente representa texto), y hay una columna de tipo float64 (que representa números decimales).:

4.2. Análisis Exploratorio de Datos (EDA).

4.2.1. Renombrar columnas.

Como se observó en el punto anterior, el conjunto de datos dispone de 22 columnas, considerando que muchos de los nombres de estas tienen el prefijo de **"_source."** primero se procederá a cambiar el nombre de las columnas eliminando el mencionado prefijo de las columnas 0 al 21 (Tabla 1), adicionalmente se cambia el nombre de la columna texto que se analiza **'complaint_what_happened'** por **'complaints'** y se obtiene el siguiente resultado:

Figura 1. Cambio de nombre 'complaint_what_happened' por 'complaints'

```
Index(['_index', '_type', '_id', '_score', 'tags', 'zip_code', 'complaint_id',
       'issue', 'date_received', 'state', 'consumer_disputed', 'product',
       'company_response', 'company', 'submitted_via', 'date_sent_to_company',
       'company_public_response', 'sub_product', 'timely', 'complaints',
       'sub_issue', 'consumer_consent_provided'],
      dtype='object')
```

Fuente: Elaboración propia, 2024

4.2.2. Preparación del texto para modelado.

Se elabora una función para que en la columna **"complaint"** del conjunto de datos convierta el texto a minúsculas, eliminando el texto entre corchetes, las palabras que contienen número, los textos que contengan "xx", "xxx" o "xxxx", los signos de puntuación y los espacios en blanco adicionales. A continuación, se muestra un ejemplo de cómo queda el texto después de aplicar los ajustes previamente mencionados.

Figura 2: En la columna "complaint" del conjunto de datos se convierta el texto a minúsculas

```
1     good morning my name is and i appreciate it if...
2     i upgraded my card in and was told by the agen...
10    chase card was reported on however fraudulent ...
11    on while trying to book a ticket i came across...
14    my grand son give me check for i deposit it in...
15                        can you please remove inquiry
17    with out notice jp morgan chase restricted my ...
20    during the summer months i experience a declin...
21    on i made a payment to an online retailer usin...
23    i have a chase credit card which is incorrectl...
Name: complaints, dtype: object
```

Fuente: Elaboración propia, 2024

4.2.3. Lematización y extracción de POS tags.

Se carga un modelo de lenguaje llamado **en_core_web_sm de spaCy**, que es un modelo pre entrenado para el idioma inglés. Este modelo incluye herramientas para analizar y procesar texto, como el análisis sintáctico y la lematización, que es el proceso de reducir las palabras a su forma base. Se crea una función que se encargará de lematizar el texto. Esta función toma un texto como entrada, representado como un **string**. Dentro de la función, el texto se procesa utilizando el modelo de **spaCy**. Este procesamiento tiene como resultado una estructura de datos que contiene información sobre cada palabra del texto, permitiendo acceder a diversos atributos lingüísticos de cada palabra (como su forma base).

Se extraen los lemas de cada palabra procesada. El lema es la forma básica de una palabra. Por ejemplo, los lemas de **"caminando" y "caminé"** serían **"caminar"**. Todos los lemas se agrupan para formar una nueva cadena de texto.

Tabla 2: Tabla comparativa de texto Lematizado

	complaints	complaint_clean	complaint_lemmatized
1	good morning my name is and i appreciate it if...	good morning my name be and I appreciate it if...	good morning my name be and I appreciate it if...
2	i upgraded my card in and was told by the agen...	I upgrade my card in and be tell by the agent ...	I upgrade my card in and be tell by the agent ...
10	chase card was reported on however fraudulent ...	chase card be report on however fraudulent app...	chase card be report on however fraudulent app...
11	on while trying to book a ticket i came across...	on while try to book a ticket I come across an...	on while try to book a ticket I come across an...
14	my grand son give me check for i deposit it in...	my grand son give I check for I deposit it int...	my grand son give I check for I deposit it int...
...	...	...	...
78303	after being a chase card customer for well ove...	after be a chase card customer for well over a...	after be a chase card customer for well over a...
78309	on wednesday i called chas my visa credit card...	on wednesday I call chas my visa credit card p...	on wednesday I call chas my visa credit card p...
78310	i am not familiar with pay and did not underst...	I be not familiar with pay and do not understa...	I be not familiar with pay and do not understa...
78311	i have had flawless credit for yrs ive had cha...	I have have flawless credit for yrs I ve have ...	I have have flawless credit for yrs I ve have ...
78312	roughly years ago i closed out my accounts wit...	roughly year ago I close out my account with j...	roughly year ago I close out my account with j...

21072 rows × 3 columns

Fuente: Elaboración propia, 2024

Una vez que el texto fue lematizado se procedió a la extracción de etiquetas. Nuevamente, con **spacy** se procede a trabajar con el texto y se crea un objeto estructurado para analizar palabras. La función extrae **etiquetas POS** de cada palabra en el texto, las devuelve como una lista y almacena el resultado en una nueva columna, **complaint_pos_tags**, que contendrá las etiquetas gramaticales correspondientes a las quejas lematizadas:

Tabla 3: Tabla de texto aplicando POS tag

	complaint_pos_tags
1	[SPACE, NOUN, ADP, NOUN, ADP, PRON, NOUN]
2	[SPACE, VERB, DET, NOUN, ADP, DET, NOUN, PRON,...
10	[SPACE, ADV, SPACE, ADJ, NOUN, AUX, AUX, VERB,...
11	[SPACE, PRON, AUX, ADV, AUX, VERB, ADP, PRON, ...
14	[SPACE, ADP, NOUN]
...	...
78303	[SPACE, NOUN, CCONJ, VERB, PART, VERB, NOUN, N...
78309	[SPACE, VERB, ADP, PRON, SPACE, CCONJ, PRON, V...
78310	[SPACE, VERB, NOUN, NOUN, ADP, DET, NOUN, SPAC...
78311	[SPACE, PRON, VERB, PRON, ADV, SPACE, NOUN, SP...
78312	[SPACE, ADP, NOUN, PART, VERB, ADP, DET, ADJ, ...

21072 rows × 1 columns

Fuente: Elaboración propia, 2024

A continuación, se realiza un proceso para mantener solo los sustantivos del texto que estamos analizando. En esta etapa se procesa cada queja en la columna **“complaints”** del dataframe utilizando **spacy**. Se filtra todas las palabras conservando los sustantivos lematizados en un nuevo **string**, los resultados se almacenan en una nueva columna **“complaint_POS_removed”.**

En el dataframe resultante se comparan las quejas originales con las quejas lematizadas que contienen solo sustantivos. Esto es muy útil en el procesamiento de texto y análisis de datos, ya que ayuda a centrarse en las palabras clave (sustantivos) de las quejas, lo que puede facilitar un análisis más enfocado.

Tabla 4: Tabla comparativa eliminando NN

	complaints	complaint_clean	complaint_lemmatized	complaint_POS_removed
1	good morning my name is and i appreciate it if...	morning name stop bank cardmember service ask ...	good morning my name be and I appreciate it if...	morning name stop bank cardmember service ask ...
2	i upgraded my card in and was told by the agen...	card agent anniversary date agent information ...	I upgrade my card in and be tell by the agent ...	card agent anniversary date agent information ...
10	chase card was reported on however fraudulent ...	chase card report application identity consent...	chase card be report on however fraudulent app...	chase card report application identity consent...
11	on while trying to book a ticket i came across...	ticket offer ticket reward card information of...	on while try to book a ticket I come across an...	ticket offer ticket reward card information of...
14	my grand son give me check for i deposit it in...	son chase account fund chase bank account mone...	my grand son give I check for I deposit it int...	son chase account fund chase bank account mone...
...	...	...	...	...

Fuente: Elaboración propia, 2024

4.2.4. Análisis exploratorio de datos para familiarizarse con la información.

Para la visualización de los datos según la longitud de los caracteres **"Complaint"**, procedemos a agregar una nueva columna **(complaint_length)** que calcule la longitud de los caracteres, posterior a ello lo ordenamos de forma descendente y el resultado es el siguiente:

Tabla 5: Tabla de longitud de caracteres

	complaint_clean	complaint_length
1096	right experience relate issue place care unit ...	12560
6262	security disclosure identity theft dispute pos...	12376
26169	disclosure identity theft dispute position ide...	10495
31952	response co mortgage reference number document...	10482
48112	majority owner proxy holder head email phone e...	10070
...	...	...
11632	money	5
25414	chase	5
7066	case	4
11806	card	4
4721		0

Fuente: Elaboración propia, 2024

Podemos observar que hay una variación grande entre el máximo tamaño de 12,560 caracteres hasta solo 4 caracteres.

- **Top 40 palabras más frecuentes.**

El código utilizado **tokeniza** el texto en una lista de palabras. Utilizamos counter para contar la frecuencia de cada palabra. Posteriormente, se extrae las 40 palabras más comunes y sus frecuencias.

Siguiendo los siguientes procesos:

Tipos de Tokenización

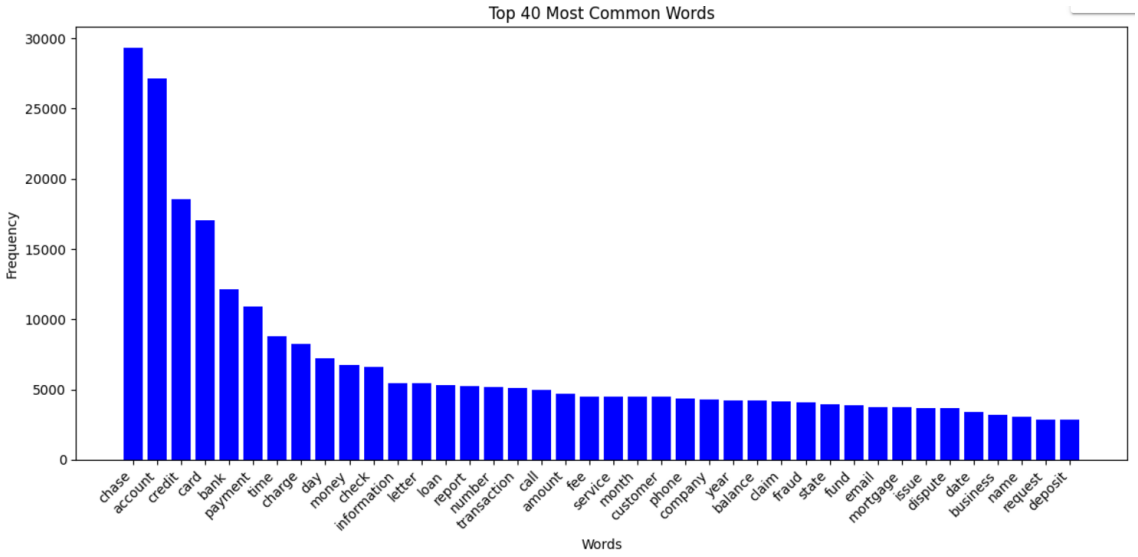
Tokenización basada en palabras: Divide el texto en palabras individuales, ejemplo: "El cliente está satisfecho" se tokeniza como ["El", "cliente", "está", "satisfecho"].

Tokenización basada en caracteres: Divide el texto en caracteres individuales. Ejemplo: "Hola" se tokeniza como ["H", "o", "l", "a"], Tokenización basada en subpalabras: Divide las palabras en unidades más pequeñas, como prefijos, sufijos o raíces, ejemplo: "clasificación" podría dividirse en ["clasific", "ación"].

Tokenización basada en oraciones: Divide el texto en oraciones completas, ejemplo: "Hoy hace sol. Mañana podría llover." se tokeniza como ["Hoy hace sol.", "Mañana podría llover."].

Los datos obtenidos se convierten a un dataframe de pandas para facilitar la visualización. El resultado puede observarse en la siguiente imagen:

Figura 3: Top 40 palabras más frecuentes



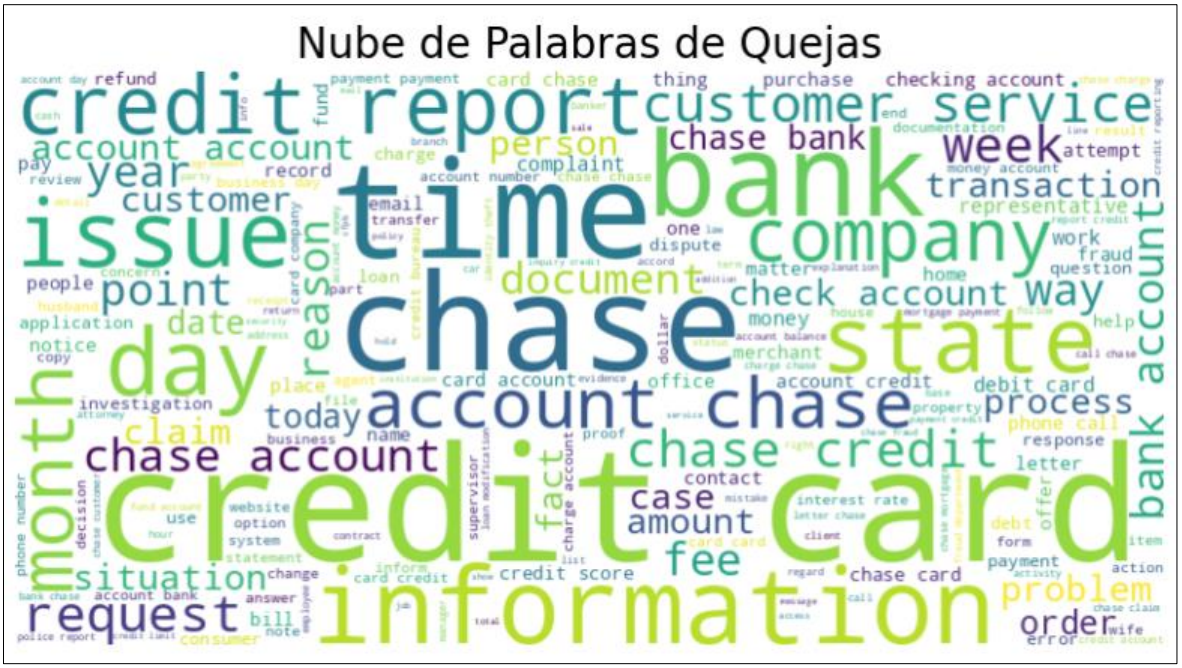
Fuente: Elaboración propia, 2024

Como puede observarse en la imagen anterior, las tres palabras con mayor frecuencia son “chase” (54.326 repeticiones), “account” (47.508 repeticiones) y “credit” (33.512 repeticiones).

- **Nube de palabras.**

Una vez identificadas las frecuencias con WordCloud podemos generar una nube de palabras.

Figura 4: Nube de palabras



Fuente: Elaboración propia, 2024

Con relación a la nube de palabras **"credit"** y **"card"** son prominentes, lo que sugiere que muchas de las quejas están relacionadas con las tarjetas de crédito. Esto puede incluir problemas como cargos incorrectos o dificultades con el uso de la tarjeta.

Asimismo, la presencia notable de **"bank"** y **"chase"** indica que la mayoría de las quejas provienen de clientes de **Chase Bank**, lo que resalta la importancia de esta institución en el contexto de las quejas. Por otro lado, la presencia de la palabra "account" sugiere que los problemas pueden estar relacionados con cuentas bancarias, como acceso, gestión o errores en el saldo.

Otras palabras relevantes son **"issue" y "problem"**, indicando que los clientes están expresando preocupaciones o dificultades específicas, enfatizando la insatisfacción con el servicio. Respecto a **"time," "day," y "month"**, se podría señalar que muchos clientes están reportando problemas relacionados con la duración de los procesos o la tardanza en la resolución de sus quejas. **"customer"** y **"service"**, sugerirían que muchos clientes están insatisfechos con el servicio al cliente, lo que puede indicar una falta de respuesta adecuada o soluciones a sus problemas.

En términos generales, la nube refleja una fuerte conexión entre las quejas y la gestión de cuentas de tarjetas de crédito en Chase Bank. Los clientes parecen tener problemas recurrentes con sus cuentas, la atención al cliente, y la resolución de disputas o cobros. Las palabras relacionadas con el tiempo (como "day" y "month") pueden indicar frustración relacionada con la duración del proceso de resolución de quejas. Frases como "credit report" y "transaction"

sugieren que algunos problemas pueden estar vinculados a la gestión de crédito y transacciones específicas.

- **Unigramas**

Para obtener los unigramas se utilizará la columna de texto **“complaint”**, que contiene las quejas ya limpias (sin caracteres especiales o ruido innecesario). Con la función **join** se combina todas estas quejas en una sola cadena, separando cada queja por un espacio. El resultado es un texto largo que contiene todas las quejas que posteriormente será dividido para obtener palabras individuales. Se utilizará **Counter** de la biblioteca **collections** para contar la frecuencia de cada palabra en la lista unigrams (Counter crea un diccionario donde las claves son las palabras y los valores son las cantidades de veces que cada palabra aparece). El resultado es un objeto **Counter** que refleja la frecuencia de cada unigram, como puede observase en la siguiente imagen.

Por ejemplo:

"El cliente está satisfecho"

Los unigramas serían:

["El", "cliente", "está", "satisfecho"]

Tabla 6: Tabla de Unigrams

	Unigram	Frequency
0	chase	54326
1	account	47508
2	credit	33512
3	card	30315
4	bank	21892
5	payment	21163
6	time	16036
7	charge	14274
8	day	13193
9	check	12288

Fuente: Elaboración propia, 2024

Se puede observar que la palabra **“chase”** es la que más veces se repite (54.326 veces) seguido de **“account”** (47.508 veces) y “credit” (33.512 veces).

- **Bigramas**

Al igual que el anterior punto se utilizará la columna **complaint_clean** y todas las quejas se combinarán en una cadena separadas por espacios; asimismo, su usó el método split() para dividir el texto combinado en palabras individuales (words, será una lista que contiene todas las palabras).

Se utilizó la función **bigrams de nltk**, que toma la lista de palabras words y genera todos los pares adyacentes de palabras (bigramos). Cada **bigrama** es representado como una **tupla de dos palabras**. El resultado es una lista llamada **bigram_list** que contiene todos los bigramas extraídos del texto.

Por ejemplo:

"El cliente está satisfecho"

Los bigramas serían:

[("El", "cliente"), ("cliente", "está"), ("está", "satisfecho")]

En el caso de los bigramas, la combinación “credit, card” es la que más se repite (12.857 veces) seguido de “credit, report” (3.957 veces) y “account, chase” (3.167).

Tabla 7: Tabla de Bigrams

	Bigram	Frequency
0	(credit, card)	12857
1	(credit, report)	3957
2	(account, chase)	3167
3	(customer, service)	3076
4	(chase, credit)	2921
5	(chase, account)	2723
6	(bank, account)	2465
7	(account, account)	2405
8	(check, account)	2302
9	(chase, bank)	2183

Fuente: Elaboración propia, 2024

- **Trigramas**

De forma similar a los anteriores casos, se utilizará la columna complaint_clean y se construirán cadenas de texto para posteriormente construir los trigramas y establecer las frecuencias de estas combinaciones, y se llegan a los siguientes resultados.

La secuencia “chase, credit, card” es la que tiene una mayor frecuencia (1.948 veces), seguido de “credit, card, account” (1.096 veces) y “credit, card, company” (987 veces).

Por ejemplo:

"El cliente está satisfecho"

Los trigramas serían:

[("El", "cliente", "está"), ("cliente", "está", "satisfecho")]

Tabla 8: Tabla de Trigrams

	Trigram	Frequency
0	(chase, credit, card)	1948
1	(credit, card, account)	1096
2	(credit, card, company)	987
3	(credit, card, chase)	744
4	(inquiry, credit, report)	522
5	(credit, card, credit)	521
6	(account, credit, card)	488
7	(card, credit, card)	414
8	(charge, credit, card)	409
9	(chase, customer, service)	392

Fuente: Elaboración propia, 2024

4.3. Modelado mediante Non-Negative Matrix Factorization (NMF).

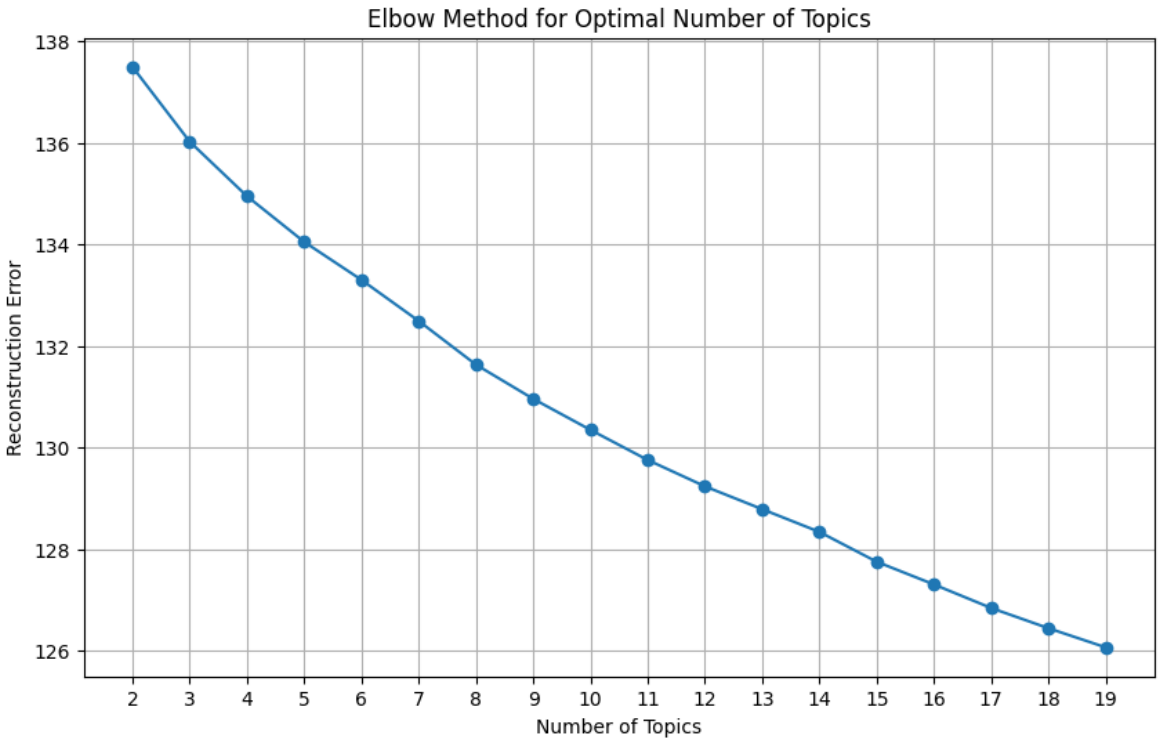
(NMF) es una técnica no supervisada, por lo que no hay etiquetas de temas en los que se entrenará el modelo. La forma en que funciona es que NMF descompone (o factoriza) vectores de alta dimensión en una representación de menor dimensión. Estos vectores de menor dimensión no son negativos, lo que también significa que sus coeficientes no son negativos.

4.3.1. Definición del mejor número de tópicos.

- **Método del Elbow.**

Según este método hay mejora al aumentar la cantidad de los clusters, pero a partir de un cierto punto (9-11), cada tema adicional proporciona una mejora menor en la calidad del modelo. Este es el punto óptimo para decidir cuántos temas utilizar. Así que, aunque puedes seguir añadiendo temas, no necesariamente obtendrás mejoras que justifiquen la complejidad adicional del modelo.

Figura 5: Método del Elbow número óptimo de tópicos



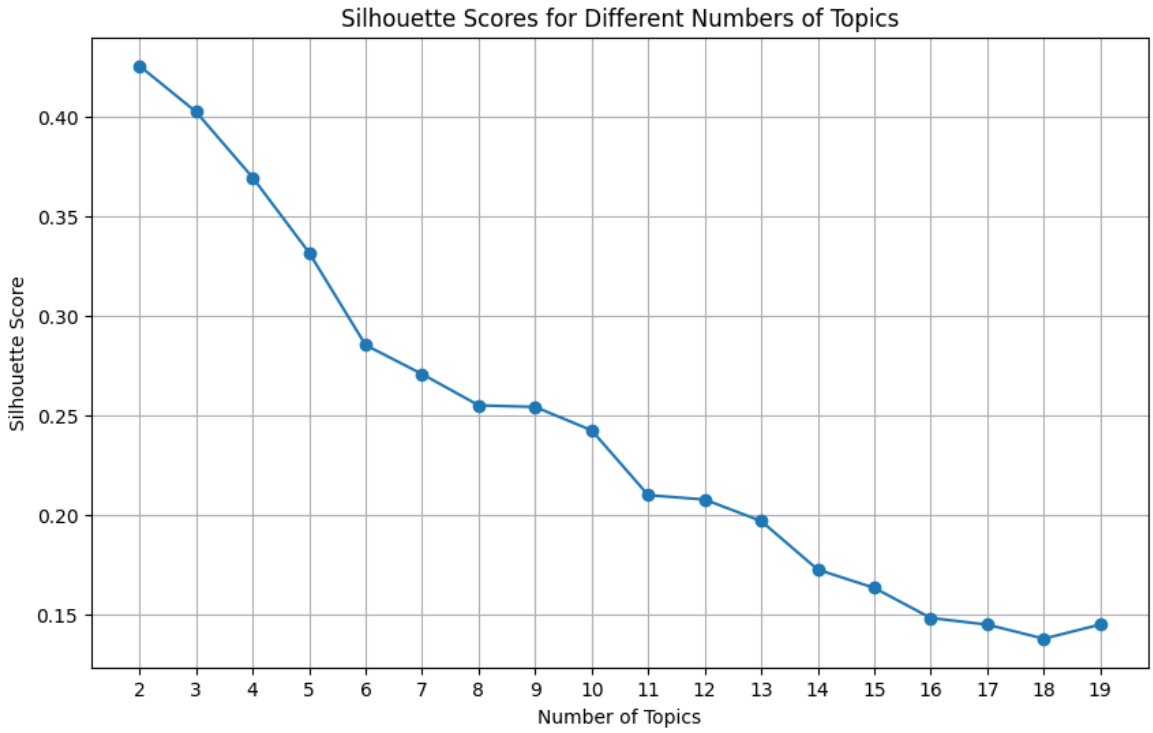
- **Método Silhouette Score.**

Este permite evaluar cómo varía la calidad de la agrupación de documentos en función del número de temas elegidos al aplicar NMF, utilizando el índice de silueta como métrica. Un puntaje de silueta más alto indica una mejor separación entre los grupos formados por el modelo.

A partir de esta gráfica, podría considerarse que entre 3 y 5 temas podría ser el número óptimo para una buena separación de agrupaciones. Más temas parecen complicar la separación.

La disminución general del puntaje sugiere que podría ser útil revisar la calidad de los datos o explorar otras técnicas de modelado y comparación con otros métodos de agrupación.

Figura 6: Método de Silhouette Score número óptimo de tópicos



Contexto Adicional: Sería valioso complementar este análisis con una evaluación cualitativa de los temas generados, para ver si esto coincide con la interpretación humana de los grupos que se están formando.

4.3.2. Tópicos establecidos.

Aplicando NMF y tomando cinco como el número de clusters para el análisis, se presentan los tópicos que ha aprendido un modelo de descomposición de matrices, mostrando las palabras más significativas para cada tópico. Es una forma efectiva de interpretar y visualizar los resultados del modelado de tópicos.

El análisis muestra los siguientes tópicos:

- ✓ **Tópico 1:**

"account check bank money fund chase deposit branch day checking business number transfer customer transaction"

- ✓ **Tópico 2:**

"card credit chase balance account limit score year point offer month interest purchase application reason"

- ✓ **Tópico 3:**

"payment loan mortgage chase home month modification interest time year rate balance property amount statement"

- ✓ **Tópico 4:**

"report inquiry credit company information reporting identity debt account theft score letter date inquire file"

- ✓ **Tópico 5:**

"charge chase dispute transaction claim merchant fraud fee purchase email amount service call refund time"

Asignamos el mejor tópico a cada ticket "complaint", luego asignamos estos nombres a los tópicos relevantes:

- ✓ Bank Account services
- ✓ Credit card or prepaid card
- ✓ Theft/Dispute Reporting
- ✓ Mortgage/Loan
- ✓ Others

5. Descripción de los Modelos Propuestos

5.1. Modelos

- ➢ **Modelo de Regresión Logística.**

  El modelo de Regresión Logística, es un modelo de clasificación supervisado que utiliza una función sigmoide para predecir la probabilidad de que una observación pertenezca a una clase específica. En este caso, la regresión logística asigna probabilidades a cada categoría de queja (tópico) basándose en las características generadas por Non-Negative Matrix Factorization NMF (vectores TF-IDF). Se clasifica un ticket en la categoría con mayor probabilidad. Su simplicidad y eficiencia lo hacen ideal para problemas con relaciones lineales entre las características.

- ➢ **Modelo de Árbol de Decisión.**

  El Árbol de decisión es un modelo basado en reglas jerárquicas que divide iterativamente los datos en subconjuntos basándose en características que maximizan la ganancia de información o reducen la impureza de Gini. En este caso, el árbol de decisión utiliza los vectores TF-IDF preprocesados para clasificar cada ticket en una de las cinco categorías. Este modelo es fácil de interpretar, permitiendo identificar directamente las reglas de clasificación para cada categoría.

- ➢ **Modelo Random Forest.**

  Random forest es un conjunto de árboles de decisión donde cada árbol se entrena en un subconjunto aleatorio de datos y características. Luego, combina los resultados de todos los árboles para tomar una decisión final, lo que mejora la precisión y reduce el sobreajuste. Aquí, el Random Forest predice la categoría de cada ticket al analizar patrones complejos en los datos, siendo especialmente útil en escenarios con interacciones no lineales.

- **Modelo de Naive Bayes.**

  Naive Bayes es un clasificador probabilístico que asume independencia entre las características, utilizando el teorema de Bayes para calcular la probabilidad de que un ticket pertenezca a una categoría dada. En este caso, utiliza los vectores TF-IDF como entradas para calcular la probabilidad de que un ticket sea clasificado en cada tema. Este modelo es rápido y eficiente para datos textuales, como las quejas de clientes.

5.2. Entrenamiento y Validación.

- **Conjuntos de Datos Utilizados:**

  Entrenamiento (70%):

  Usado para ajustar los modelos (`fit(X_train, y_train)`), incluye los datos de quejas procesados y sus categorías.

  Testeo (30%):

  Usado para evaluar el desempeño del modelo con datos no vistos (`predict(X_test)`), mide qué tan bien generaliza el modelo en nuevos tickets.

- **Forma de Aplicación.**

  Preprocesamiento:
  Se procesan las quejas con técnicas de vectorización (TF-IDF) para convertir el texto en datos numéricos.

  Entrenamiento:
  Los modelos aprenden patrones en el conjunto de entrenamiento para asociar palabras y combinaciones de palabras con temas específicos.

  Evaluación:
  Se generan predicciones en el conjunto de prueba, comparándolas con las etiquetas reales (y_test). Métricas como F1-score y precisión ayudan a determinar la eficacia del modelo.

- **Regresión Logística.**

  Entrenamiento y testeo:
  La regresión logística se entrena usando las representaciones TF-IDF de los datos de texto (X_train) como características y las categorías de temas (y_train) como etiquetas. Se utiliza un límite de iteraciones de 1000 (`max_iter=1000`) para asegurar la convergencia.

  Aplicación:

Este modelo predice la probabilidad de que cada ticket pertenezca a un tema específico. En este caso, clasifica el tema con mayor probabilidad como la categoría final para cada ticket.

Evaluación:
Se mide el desempeño en los datos de prueba (X_test) usando métricas como precisión, recall y F1-score calculadas en el reporte de clasificación.

- **Árbol de Decisión.**

Entrenamiento y testeo:
El Árbol de Decisión se ajusta utilizando las mismas entradas y etiquetas que la regresión logística. Se construyen reglas jerárquicas que dividen los datos según las características de los vectores TF-IDF.

Aplicación:
El modelo clasifica un ticket en un tema basándose en las reglas aprendidas. Por ejemplo, si ciertas palabras tienen pesos altos en un tema específico, esas ramas del árbol clasifican automáticamente el ticket en esa categoría.

Evaluación:
Evalúa el modelo en el conjunto de prueba (X_test), y los resultados incluyen métricas clave para analizar su desempeño en términos de exactitud y capacidad predictiva.

- **Random Forest.**

Entrenamiento y testeo:
El Bosque Aleatorio entrena múltiples árboles de decisión en subconjuntos aleatorios de los datos y combina sus predicciones mediante votación para clasificar cada ticket.

Aplicación:
Este modelo maneja bien la variabilidad en los datos, mejorando la capacidad de generalización en comparación con un solo árbol. Los vectores TF-IDF proporcionan las características que el modelo utiliza para detectar patrones complejos en las quejas.

Evaluación:
Utiliza los datos de prueba (X_test) para medir métricas como la F1-score ponderada, que es crucial en este problema ya que las categorías pueden no estar balanceadas.

- **Naive Bayes.**

Entrenamiento y testeo:

Naive Bayes entrena en las características TF-IDF asumidas como independientes entre sí. Utiliza la probabilidad condicional para predecir la categoría más probable para cada ticket.

Aplicación:
Este modelo es computacionalmente eficiente y adecuado para problemas de clasificación de texto. Predice temas analizando la probabilidad de que ciertas palabras o combinaciones estén asociadas con cada categoría.

Evaluación:
Al igual que los otros modelos, se evalúa utilizando el conjunto de prueba (X_test), y los resultados incluyen medidas de precisión, recall y F1-score.

5.3. Selección del Modelo Final.

- **Regresión Logística.**

  Este modelo alcanzó el mejor desempeño global, con una exactitud del 96.58% y métricas asociadas muy consistentes: precisión (96.59%), recall (96.58%) y F1-score (96.57%).

  La alta precisión indica que el modelo clasifica correctamente la mayoría de los tickets y casi no tiene falsos positivos. Además, el alto recall sugiere que la mayoría de los temas relevantes son identificados correctamente. El equilibrio entre estas métricas muestra un excelente rendimiento general.

  Su simplicidad, robustez y capacidad para manejar problemas lineales lo convierten en una opción ideal para datos con características bien separadas, como los vectores TF-IDF preprocesados.

- **Árbol de Decisión.**

  Este modelo tuvo un desempeño moderado, con una exactitud del 82.37%. Sus métricas asociadas (precisión, recall y F1-score) también están alineadas alrededor del 82.3%.

  El modelo puede estar sobreajustado a ciertas características específicas del conjunto de entrenamiento, lo que lo hace menos generalizable. Su desempeño inferior sugiere que los patrones subyacentes en los datos son más complejos de lo que un árbol de decisión simple puede capturar.

- **Random Forest.**

  Este modelo mostró un buen desempeño, con una exactitud del 87.25%, precisión del 87.33%, recall del 87.25% y F1-score del 87.18%.

  Random Forest mejora la generalización al combinar predicciones de múltiples árboles, lo que le permite manejar relaciones no lineales y características complejas mejor que un solo árbol de decisión. El modelo tiene robustez frente a sobreajuste y buena capacidad para manejar datos complejos con interacciones no lineales.

- **Naive Bayes.**

  Este modelo tuvo el desempeño más bajo, con una exactitud del 76.67%. Aunque su precisión es ligeramente más alta (78.08%), sus valores de recall (76.67%) y F1-score (75.63%) son inferiores a los demás modelos.

  Naive Bayes asume independencia entre las características, lo cual puede no ser una suposición válida en datos textuales, donde las palabras suelen estar correlacionadas. Esto explica su desempeño limitado en este caso. Es rápido y eficiente, ideal para datos textuales cuando la independencia entre características es razonable.

6. Resultados y Selección del modelo.

A continuación, un cuadro resumen de las métricas obtenidas de los modelos planteados en el análisis:

Tabla 9: Cuadro Comparativo de Métricas

Modelo	Exactitud	F1-Score
Regresión Logística	96.58%	96.57%
Árbol de Decisión	82.37%	82.36%
Random Forest	87.25%	87.18%
Naive Bayes	76.67%	75.63%

La Regresión Logística fue seleccionada como el mejor modelo debido a su desempeño superior y su adecuación al problema de clasificación de texto preprocesado, respaldado por las siguientes razones:

6.1. Desempeño Métrico Sobresaliente.

Exactitud: Con una precisión del 96.58%, la Regresión Logística supera a los demás modelos. Esto significa que clasifica correctamente la gran mayoría de las quejas en sus categorías correspondientes.

F1-Score: Su F1-score de 96.57% refleja un equilibrio ideal entre precisión y recall, indicando que tanto los falsos positivos como los falsos negativos son mínimos.

Consistencia en las métricas: Las métricas asociadas (precisión, recall y F1-score) están alineadas, lo que confirma un rendimiento robusto y consistente.

6.2. Adecuación al Problema.

La Regresión Logística es un modelo lineal que funciona bien con datos representados en vectores dispersos de alta dimensionalidad, como los obtenidos mediante la transformación TF-IDF en este caso.

La naturaleza del problema (clasificación de texto en categorías bien definidas) es linealmente separable en gran medida, lo que se ajusta perfectamente a las capacidades del modelo.

6.3. Simplicidad y Eficiencia.

Facilidad de implementación: La Regresión Logística es computacionalmente más eficiente que modelos como Random Forest, especialmente en conjuntos de datos grandes o con muchas características, como el texto preprocesado.

Velocidad: Su tiempo de entrenamiento y predicción es más rápido en comparación con modelos más complejos, lo cual es crucial para sistemas en producción donde la velocidad es un factor importante.

Interpretabilidad: Los coeficientes del modelo proporcionan una manera clara de entender la importancia de las características en la clasificación, lo cual facilita la generación de insights adicionales.

6.4. Comparación con otros modelos.

Árbol de decisión: Aunque interpretable, su exactitud es del 82.37%, significativamente más baja. Además, es más propenso al sobreajuste, especialmente con datos textuales complejos.

Random Forest: Aunque robusto y capaz de manejar relaciones no lineales, su exactitud del 87.25% es inferior, y su costo computacional es mayor. La mejora en rendimiento respecto al Árbol de Decisión no justifica su complejidad adicional frente a la Regresión Logística.

Naive Bayes: Este modelo, con una exactitud del 76.67%, tiene un desempeño limitado debido a la suposición de independencia de las características, que no es válida para datos TF-IDF, donde las palabras suelen estar correlacionadas.

6.5. Escalabilidad y Producción.

La Regresión Logística es altamente escalable y fácil de integrar en sistemas productivos, haciendo que sea ideal para aplicaciones de clasificación en tiempo real.

Su capacidad para manejar grandes cantidades de datos y realizar predicciones rápidamente la hace adecuada para escenarios donde el número de tickets puede crecer significativamente.

7. Rendimiento del Modelo Final.

Tras probar múltiples modelos de aprendizaje supervisado, la Regresión Logística fue seleccionada como el modelo final debido a su desempeño superior, eficiencia y escalabilidad.

El modelo de **Regresión Logística** fue evaluado utilizando métricas clave: **exactitud**, **precisión**, **recall** y **F1-Score**. Los resultados obtenidos son los siguientes:

Tabla 10: Rendimiento modelo Regresión Logística

Métrica	Valor
Exactitud	96.58%
Precisión (Promedio Ponderado)	96.59%
Recall (Promedio Ponderado)	96.58%
F1-Score (Promedio Ponderado)	96.57%

- **Exactitud:** El modelo clasifica correctamente el 96.58% de las quejas en las categorías correspondientes.
- **F1-Score:** Indica un excelente equilibrio entre precisión y recall, asegurando que tanto los falsos positivos como los falsos negativos sean mínimos.

8. Conclusiones.

☑ La correcta disposición de los datos de texto provenientes de los tickets de atención al cliente fue un paso fundamental para garantizar la calidad del análisis posterior. La organización estructurada de los datos permitió una transición eficiente hacia las etapas de preprocesamiento y análisis, estableciendo una base sólida para el desarrollo del modelo. Sin embargo es importante considerar que se desecharon casi el 30% de los registros del conjunto de datos porque tenían la columna de análisis “complaint” en blanco.

☑ La implementación de técnicas de limpieza como eliminación de ruido, tokenización, lematización y filtrado de palabras irrelevantes resultó crucial para preparar los datos de texto “complaint” en un formato uniforme y procesable. Este paso aseguró que las características extraídas fueran representativas del contenido de los tickets y no estuvieran sesgadas por información irrelevante o ruido textual.

- ☑ El análisis exploratorio permitió comprender las características principales de las quejas, identificando patrones en la frecuencia de palabras y términos comunes en los datos. Esta etapa proporcionó insights iniciales sobre la naturaleza de los problemas planteados por los clientes, ayudando a alinear los enfoques de modelado con las características del conjunto de datos.

- ☑ El uso de representaciones TF-IDF fue clave para transformar los datos textuales en un formato numérico adecuado para el modelado (vectorización). Esta técnica destacó las palabras más relevantes en cada ticket en relación con su frecuencia global, asegurando que el modelo se enfocara en términos significativos para la clasificación de los temas.

- ☑ La factorización de matrices no negativas (NMF) identificó cinco temas principales relacionados con productos y servicios financieros: tarjetas de crédito/prepago, servicios de cuentas bancarias, reportes de robo, hipotecas/préstamos y otros. Este enfoque no supervisado permitió una segmentación inicial de los tickets, facilitando su posterior clasificación mediante técnicas supervisadas.

- ☑ Se diseñaron y probaron varios modelos de aprendizaje supervisado (Regresión Logística, Árbol de Decisión, Random Forest y Naive Bayes) para clasificar los tickets en las categorías identificadas. Entre estos, la Regresión Logística mostró el mejor rendimiento, alcanzando una exactitud del 96.58%, junto con altos valores de precisión, recall y F1-score.

- ☑ La Regresión Logística fue seleccionada como el modelo final debido a su desempeño superior en todas las métricas relevantes, además de su eficiencia computacional e interpretabilidad. Este modelo demostró ser adecuado para la clasificación de tickets en tiempo real, cumpliendo con el objetivo del proyecto.

- ☑ En caso de que se diera la implementación del modelo propuesto, éste proporcionará a la empresa financiera una herramienta robusta para automatizar la clasificación de quejas de los clientes. Esto optimiza los tiempos de respuesta, reduce la carga de trabajo manual y mejora la experiencia del cliente. Además, el enfoque propuesto puede ser actualizado con nuevos datos para mantener su relevancia y efectividad a lo largo del tiempo.

En resumen, este estudio demuestra cómo las técnicas de procesamiento de lenguaje natural (NLP), combinadas con enfoques de modelado de temas y aprendizaje supervisado, pueden abordar eficazmente el problema de clasificación de tickets en empresas financieras. La solución desarrollada no solo cumple con los objetivos planteados, sino que también establece un marco escalable para abordar futuros desafíos en la gestión de datos de texto.

.

9. Referencias Bibliográficas

Batra, S., Sinha, A., & Sharma, S. (2020). Customer complaint management: A review of contemporary research and future directions. *Journal of Business Research, 111*, 319-328. https://doi.org/10.1016/j.jbusres.2019.08.013

George, T. (2021). Exploratory Research | Definition, Guide, & Examples. *Scribbr*. https://www.scribbr.com

Kumar, A., Shukla, S., & Prasad, S. (2018). Customer satisfaction and loyalty in financial services: An empirical investigation. *International Journal of Bank Marketing, 36*(7), 1232-1249. https://doi.org/10.1108/IJBM-06-2017-0147

Pérez, C., Martínez, M., & Gómez, L. (2019). Automating customer complaint management in financial services: A case study. *Journal of Financial Services Marketing, 24*(4), 25-36. https://doi.org/10.1057/s41264-019-00062-0

Streefkerk, R. (2019). Qualitative vs. Quantitative Research | Differences, Examples & Methods. *Scribbr*. https://www.scribbr.com

Zhang, X., Liu, X., & He, J. (2017). Text mining and machine learning approaches in customer complaint classification. *Journal of Service Research, 20*(3), 334-350. https://doi.org/10.1177/1094670517699823